MEC140x/1x

Keyboard and Embedded Controller Products for Notebook PC

Common Features

- 3.3V Operation
- ACPI 3.0 Compliant
- PC2001 compliant
- VTR (standby) and VBAT Power Planes
  - Low Standby Current in Sleep Mode
- Connected Standby Support
- 32kHz Clock Source
  - Internal 32kHz Oscillator
  - External 32kHz Clock Source
    - 32kHz Crystal (XTAL) Supported
    - Single-Ended 32kHz Clock Source
- LPC Host Interface
  - LPC Specification 1.1 Compatible
  - LPC I/O and Memory Cycles Decoded
  - Supports optional signals: CLKRUN#, LPCPD#, SERIRQ, SMI#, EC_SCI# (ACPI PME Event)
  - Supports 19.2 MHz to 33 MHz nominal bus clock speeds
- Configuration Register Set
  - Compatible with ISA Plug-and-Play Standard
  - EC-Programmable Base Address
- 8042 Emulated Keyboard Controller
  - 8042 Style Host Interface
  - Port 92 Legacy A20M Support
  - Fast GATEA20 & Fast CPU_RESET
- System to EC Message Interface
  - One Embedded Memory Interface
    - Host Serial or Parallel IRQ Source
    - Provides Two Windows to On-Chip SRAM for Host Access
    - Two Register Mailbox Command Interface
  - Mailbox Registers Interface
    - Thirty-two 8-Bit Scratch Registers
    - Two Register Mailbox Command Interface
    - Two Register SMI Source Interface
  - Five ACPI Embedded Controller Interfaces
    - Four EC Interfaces
    - One Power Management Interface
- MIPS32® M14K™ Microcontroller Core
  - microMIPS-Compatible Instruction Set
  - High-performance Multiply/Divide Unit
  - Programmable clock frequencies: 48MHz, 12MHz, 3MHz, and 1MHz
  - Sleep mode
  - 2-wire Debug Interface (ICSP)
    - 6 Breakpoints (4-instruction; 2-data)
    - Enhanced to Support Debug in Heavy and Deep Sleep States
- Trace FIFO Debug Port (TFDP)
- Internal DMA Controller
  - Hardware or Firmware Flow Control
  - Firmware Initiated Memory-to-Memory transfers
  - 7-Hardware DMA Channels support three SMBus Master/Slave Controllers and one SPI Controller
  - Hardware CRC-32 Generator on Channel 0
- Secure Boot ROM Loader
  - 4 Code Images in Shared Flash Supported
  - Crisis Recovery over Keyboard matrix Scan Pins
  - Supports CRC-32 and AES-128 Encryption
- Vectored Interrupt Controller
  - Maskable Interrupt controller
  - Maskable Hardware Wake-Up Events
  - Supports legacy aggregated mode
  - Supports Vector Generation per Status Bit
- Programmable 16-bit Counter/Timer Interface
  - Four 16-bit Auto-reloading Counter/Timer Instances
  - Two Operating Modes per Instance: Timer and One-shot.
- 32-bit RTOS Timer
  - Runs Off 32kHz Clock Source
  - Continues Counting in all the Chip Sleep States Regardless of Processor Sleep State
  - Counter is Halted when Embedded Controller is Halted (e.g., JTAG debugger active, break points)
  - Generates wake-capable interrupt event
- Watch Dog Timer (WDT)
- Hibernation Timer Interface
  - One 32.768 KHz Driven Timer
  - Programmable Wake-up from 0.5ms to 128 Minutes
- Week Timer
  - System Power Present Input Pin

  - Week Alarm Event only generated when System Power is Available
  - Power-up Event
  - Week Alarm Interrupt with 1 Second to 8.5 Year Time-out
  - Sub-Week Alarm Interrupt with 0.50 Seconds - 72.67 hours time-out
  - 1 Second and Sub-second Interrupts
- Battery-Powered General Purpose Output (BGPO)
- VBAT-Powered Control Interface (VCI)
  - 2 Active-low VCI Inputs
  - 1 Active-high VCI Input
  - 1 Active-high VCI Output Pin
  - Optional filter and latching
- Power-Fail Status Register
- Port 80 BIOS Debug Port
  - Two Ports, Assignable to Any LPC IO Address
  - 24-bit Timestamp with Adjustable Timebase
  - 16-Entry FIFO
- PECI Interface 3.0
- Two Programmable Comparators
  - 8 Bit Resolution
  - Independent Outputs per Comparator
  - Option to Use Pin or Programmable Voltage Reference Input
  - Can be used for Thermistor Voltage Sensing
- Integrated Standby Power Reset Generator
- XNOR Test Mode

Product Dependent Features

- Enhanced Serial Peripheral Interface (eSPI)
  - Intel eSPI Specification compliant
  - Supports four channels/interfaces:
    - Peripheral channel Interface
    - Virtual Wire Interface
    - Out of Band Channel Interface
    - Flash Channel Interface
  - Supports EC Bus Master to Host Memory
- Internal Memory
  - Boot ROM
  - 32 kB Data Optimized SRAM
  - Code Optimized SRAM Options from 96 kB to 160 kB
  - 64 Bytes Battery Powered SRAM
- Keyboard Matrix Scan Controller
  - Supports 18x8 Matrix
  - Pre-Drive Mode Supported
- Up To Three EC-based SMBus 2.0 Host Controllers
  - Allows Master or Dual Slave Operation
  - Controllers are Fully Operational on Standby Power
  - I^2C Datalink Compatibility Mode
  - Multi-Master Capable
  - Supports Clock Stretching
  - Programmable Bus Speeds
  - 1 MHz Capable
  - SMBus Time-outs Interface
  - Up to 6 Port Flexible Multiplexing
    - Up to 5 ports with 1.8V or 3.3V Configurable Input Threshold
    - 1 port with VTT level signaling (i.e., AMD SB-TSI Port)
  - Supports DMA Network Layer
- Up To Two PS/2 Controllers
  - Independent Hardware Driven PS/2 Ports
  - Fully functional on Main and/or Suspend Power
  - PS/2 Edge Wake Capable
  - 3.6V Tolerant I/O Suitable for Internal Board Routing
- General Purpose I/O Pins
  - Inputs
    - Asynchronous rising and falling edge wakeup detection Interrupt High or Low Level
  - Outputs:
    - Push Pull or Open Drain output
    - Programmable power well emulation
  - Pull up or pull down resistor control
    - Automatically disabling pull-up resistors when output driven low
    - Automatically disabling pull-down resistors when output driven high
  - Group- or individual control of GPIO data.
- Up To Three LEDs
  - Programmable Blink Rates
  - Piecewise Linear Breathing LED Output Controller
    - Provides for programmable rise and fall waveforms
  - Operational in EC Sleep States
- One Serial Peripheral Interface (SPI) Controller
  - Master Only SPI Controller
  - Mappable to three ports (only 1 port active at a time)
    - 1 shared SPI Interface.
    - 1 General Purpose SPI Interface (package dependent)
    - 1 Crisis recovery SPI Interface (located on Keyboard Matrix Scan connector)
  - Dual and Quad I/O Support

  - Flexible Clock Rates
  - SPI Burst Capable
  - SPI Controller Operates with Internal DMA Controller with CRC Generation
- Up To Two BC-Link Interconnection Bus
- ADC Interface
  - Up to 8 Channels
  - 10-bit Conversion in 10µs
  - Integral Non-Linearity of ±0.5 LSB; Differential Non-Linearity of ±0.5 LSB
  - External Analog Voltage Reference
- DAC Interface
  - Up to 2 Channels
  - 8 Bit Resolution
  - External Analog Voltage Reference
- FAN Support
  - Up to 8 Programmable Pulse-Width Modulator (PWM) Outputs, for Fan or General Use
    - Multiple Clock Rates
    - 16-Bit ON & 16-Bit OFF Counters
  - Up to Two Fan Tachometer Inputs,
    - 16 Bit Resolution
- Universal Asynchronous Receiver Transmitter (UART)
  - Full function Serial Port or 2-Pin Debug Port (product dependent)
  - High Speed NS16C550A Compatible UART with Send/Receive 16-Byte FIFOs
  - Accessible from Host and EC
  - Full Duplex Operation
  - Programmable Input/output Pin Polarity Inversion
  - Programmable Main Power or Standby Power Functionality
  - Standard Baud Rates to 115.2 Kbps, Custom Baud Rates to 1.5 Mbps
- Package
  - 128 VTQFP RoHS Compliant Package
  - 144 WFBGA RoHS Compliant Package

Products

Note: This table shows the total number of instances available per product. However, not all features may be used simultaneously since they are multiplexed on the same pins. See the Pin Description chapter to determine specific chip configuration options.

Catalog Part Number	Package	Host Interfaces	SRAM Memory (Code + Data)	Keyboard Matrix Scan Controller	SMBus 2.0 Ports	PS/2 Controllers	GPIOs	SPI Interfaces	BC-Link Interfaces	ADCs	DAC	PWMs	TACHs	UART
MEC1404-NU	128-VTQFP	• LPC • I2C	128 kB	Yes	6	2	106	3	2	8	2	8	2	full
MEC1404-SZ	144-WFBGA													
MEC1406-NU	128-VTQFP	• LPC • I2C	160 kB	Yes	6	2	106	3	2	8	2	8	2	full
MEC1406-SZ	144-WFBGA													
MEC1408-NU	128-VTQFP	• LPC • I2C	192 kB	Yes	6	2	106	3	2	8	2	8	2	full
MEC1408-SZ	144-WFBGA													
MEC1414-NU	128-VTQFP	• LPC • I2C • eSPI	128 kB	Yes	6	2	106	3	2	8	2	8	2	full
MEC1414-SZ	144-WFBGA													
MEC1416-NU	128-VTQFP	• LPC • I2C • eSPI	160 kB	Yes	6	2	106	3	2	8	2	8	2	full
MEC1416-SZ	144-WFBGA													
MEC1418-NU	128-VTQFP	• LPC • I2C • eSPI	192 kB	Yes	6	2	106	3	2	8	2	8	2	full
MEC1418-SZ	144-WFBGA													

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include -literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

Table Of Contents

1.0 GENERAL DESCRIPTION

The MEC140x/1x is a family of keyboard and embedded controller designs customized for notebooks and tablet platforms. The MEC140x/1x family is a highly-configurable, mixed signal, advanced I/O controller architecture. Every device in the family incorporates a 32-bit MIPS32 M14K Microcontroller core with a closely-coupled SRAM for code and data. A secure boot-loader is used to download the custom firmware image from the system's shared SPI Flash device, thereby allowing system designers to customize the device's behavior.

The MEC140x/1x products may be configured to communicate with the system host through one of three host interfaces: Intel Low Pin Count (LPC), eSPI, or I2C. Note that this functionality is product dependent. To see which features apply to a specific part in the family see Products on page 3. The document defines the features for all devices in the family.

The MEC140x/1x products are designed to operate as either a stand-alone I/O device or as an EC Base Component of a split-architecture Advanced I/O Controller system which uses BC-Link communication protocol to access up to two BC bus companion components. The BC-Link protocol is peer-to-peer providing communication between the MEC140x/1x embedded controller and registers located in a companion device.

The MEC140x/1x is directly powered by a minimum of two separate suspend supply planes (VBAT and VTR) and senses a third runtime power plane (VCC) to provide "instant on' and system power management functions. In addition, this family of products has the option to connect the VTR_33_18 power pin to either a 3.3V VTR power supply or a 1.8V power supply. This option may only be used with the eSPI Host Interface or the I2C Host Interface. In systems using the I2C Host Interface, ten GPIOs are powered by VTR_33_18, thereby allowing them to operate at either 3.3V or 1.8V. All the devices are equipped with a Power Management Interface that supports low-power states and are capable of operating in a Connected Standby system.

The MEC140x/1x family of devices offer a software development system interface that includes a Trace FIFO Debug port, a host accessible serial debug port with a 16C550A register interface, a Port 80 BIOS Debug Port, and an In-circuit Serial Programming (ICSP) interface.

1.1 Boot ROM

Following the release of the EC_PROC_RESET# signal, the processor will start executing code in the Boot ROM. The Boot ROM executes the SPI Flash Loader, which downloads User Code from an external SPI Flash and stores it in the internal Code RAM. Upon completion, the Boot ROM jumps into the User Code and starts executing.

1.2 Initialize Host Interface

By default, this device powers up all the interfaces, except the VBAT powered interfaces and select signals, to GPIO inputs. The Boot ROM is used to download code from an external flash via either the Shared Flash Interface, the eSPI flash channel or the Private Flash Interface. The downloaded code must configure the device's pins according to the platform's needs. This includes initializing the Host Interface.

Once the device is configured for operation, the downloaded code must deassert the system's RSMRST# (Resume Reset) signal. Any GPIO may be selected for the RSMRST# function. This is up to the system board designer. The only requirement is that the board designer attach an external pull-down on the GPIO pin being used for the RSMRST# function. This will ensure the RSMRST# pin is asserted low by default and does not glitch during power-up.

This family of devices has up to three Host Interface options. It may be configured as an LPC Device, an eSPI Device, or I2C device. See Products on page 3 for the features supported in each device.

On a VTR POR, all the host interface pins default to GPIO inputs.

1.2.1 CONFIGURE LPC INTERFACE

The downloaded firmware must configure the GPIO Pin Control registers for the LPC alternate function, configure the LPC Base Address Register (BAR), and activate the LPC block.

Example:

- GPIO034 Pin Control Register = 0x1000; //ALT FUNC1 – PCI_CLK
- GPIO040 Pin Control Register = 0x1000; //ALT FUNC1 – LAD0
- GPIO041 Pin Control Register = 0x1000; //ALT FUNC1 – LAD1
- GPIO042 Pin Control Register = 0x1000; //ALT FUNC1 – LAD2
- GPIO043 Pin Control Register = 0x1000; //ALT FUNC1 – LAD3
- GPIO044 Pin Control Register = 0x1000; //ALT FUNC1 – LFRAME_N

- GPIO061 Pin Control Register = 0x1000; //ALT FUNC1 – LPC_PD_N
- GPIO063 Pin Control Register = 0x1000; //ALT FUNC1 – SER_IRQ
- GPIO064 Pin Control Register = 0x1000; //ALT FUNC1 – PCI_RESET
- GPIO067 Pin Control Register = 0x1000; //ALT FUNC1 – CLKRUN
- LPC Interface (Configuration Port) BAR = 0x002E_8C01; //set bit 15
- LPC Activate Register = 0x01;

1.2.2 CONFIGURE ESPI INTERFACE

The downloaded firmware must configure the GPIO Pin Control registers for the eSPI alternate function, configure the eSPI I/O Component (Configuration Port) Base Address Register (BAR), and activate the eSPI block.

Example:

- GPIO034 Pin Control Register = 0x2000; //ALT FUNC2 – ESPI_CLK
- GPIO044 Pin Control Register = 0x2000; //ALT FUNC2 – ESPI_CS#
- GPIO040 Pin Control Register = 0x2000; //ALT FUNC2 – ESPI_IO0
- GPIO041 Pin Control Register = 0x2000; //ALT FUNC2 – ESPI_IO1
- GPIO042 Pin Control Register = 0x2000; //ALT FUNC2 – ESPI_IO2
- GPIO043 Pin Control Register = 0x2000; //ALT FUNC2 – ESPI_IO3
- GPIO063 Pin Control Register = 0x2000; //ALT FUNC2 – ESPI_ALERT#
- GPIO061 Pin Control Register = 0x2000; //ALT FUNC2 – ESPI_RESET#
- eSPI I/O Component (Configuration Port) BAR = 0x002E_0001; //set bit 15
- eSPI Activate Register = 0x01;

1.2.3 CONFIGURE I2C INTERFACE

Similar to the LPC and eSPI interfaces, the downloaded firmware must configure the GPIO Pin Control registers for the SMBus alternate function and activate the associated SMB/I2C Controller.

1.3 Initialize Peripheral Interfaces

This will be system dependent, however, this section outlines some recommendations when enabling certain interfaces.

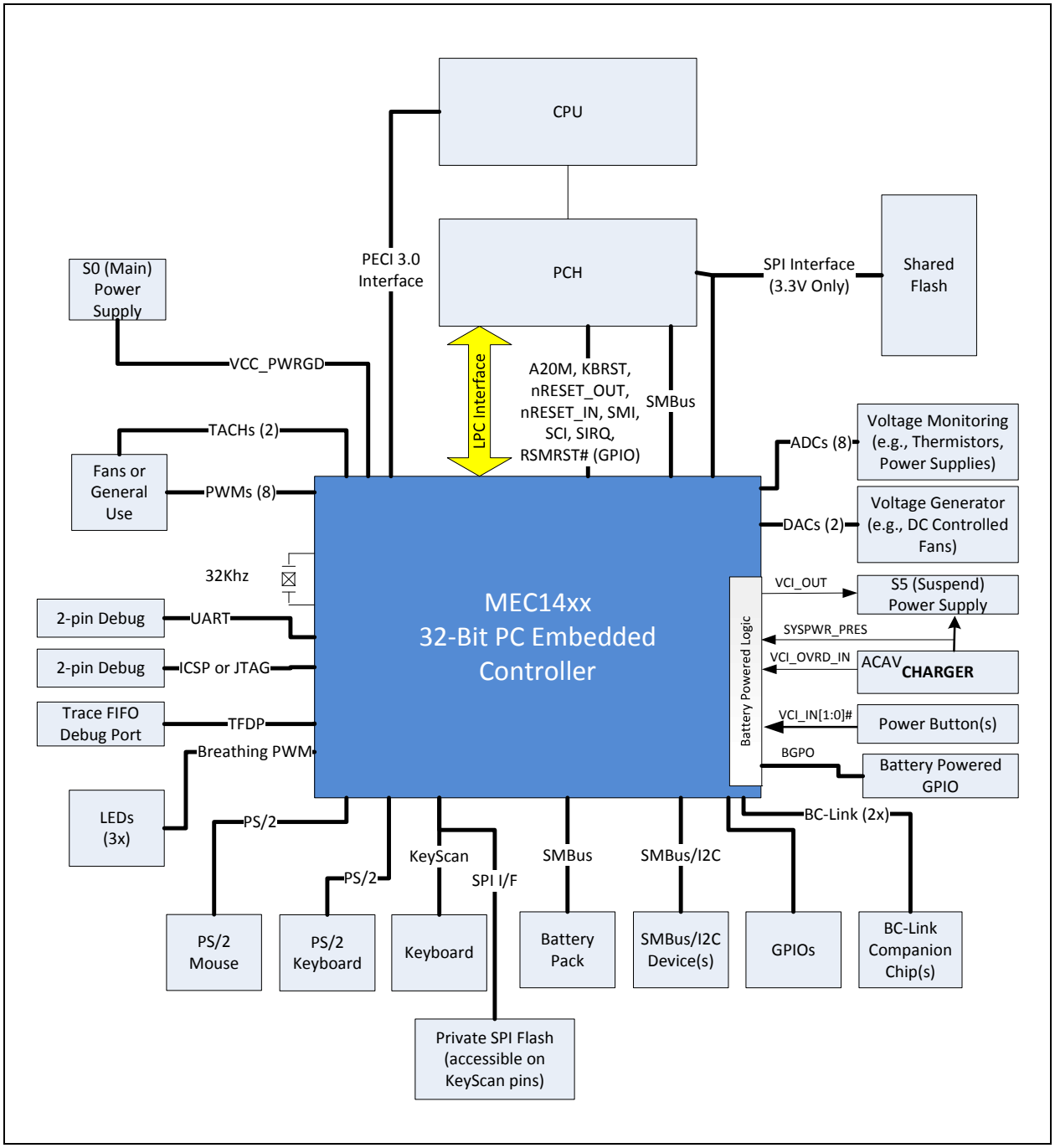
1.3.1 KEYBOARD SCAN INTERFACE

The Keyboard Scan Interface has been multiplexed onto GPIO pins. Internal pull-up resistors, enabled via the GPIO Pin Control Registers", may be used on the KSI and KSO pins instead of external pull-ups. However, if internal pull-ups are used then the PreDrive Mode must be enabled. The GPIO Pin Control register format is defined in Section 22.6.1.1, "Pin Control Register," on page 329. The PreDrive Mode is defined in Section 30.10.2, "PreDrive Mode," on page 406.

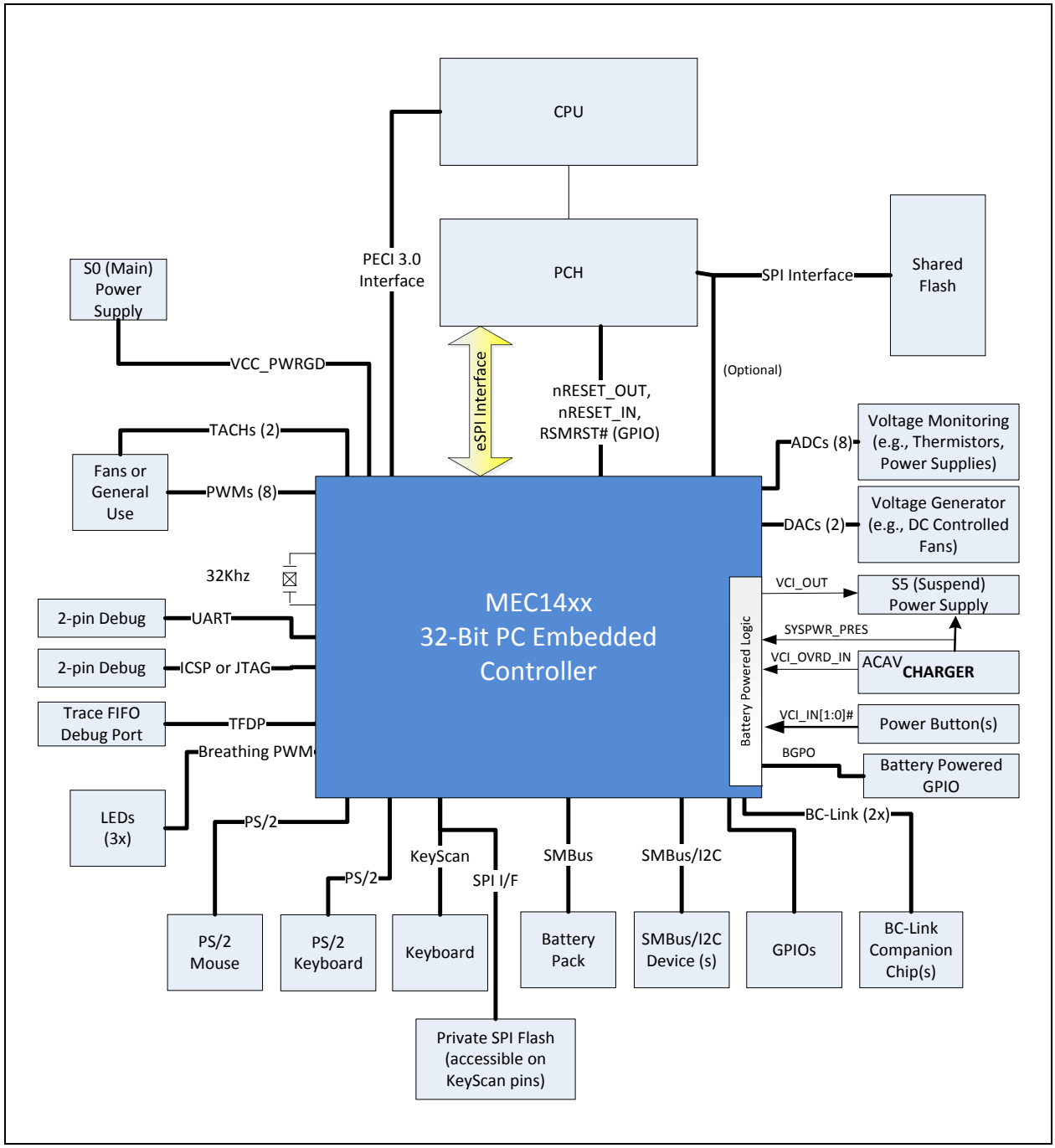
1.4 System Block Diagrams

Note: Not all features shown are available on all devices. Refer to Products on page 3 for a list of the features by device.

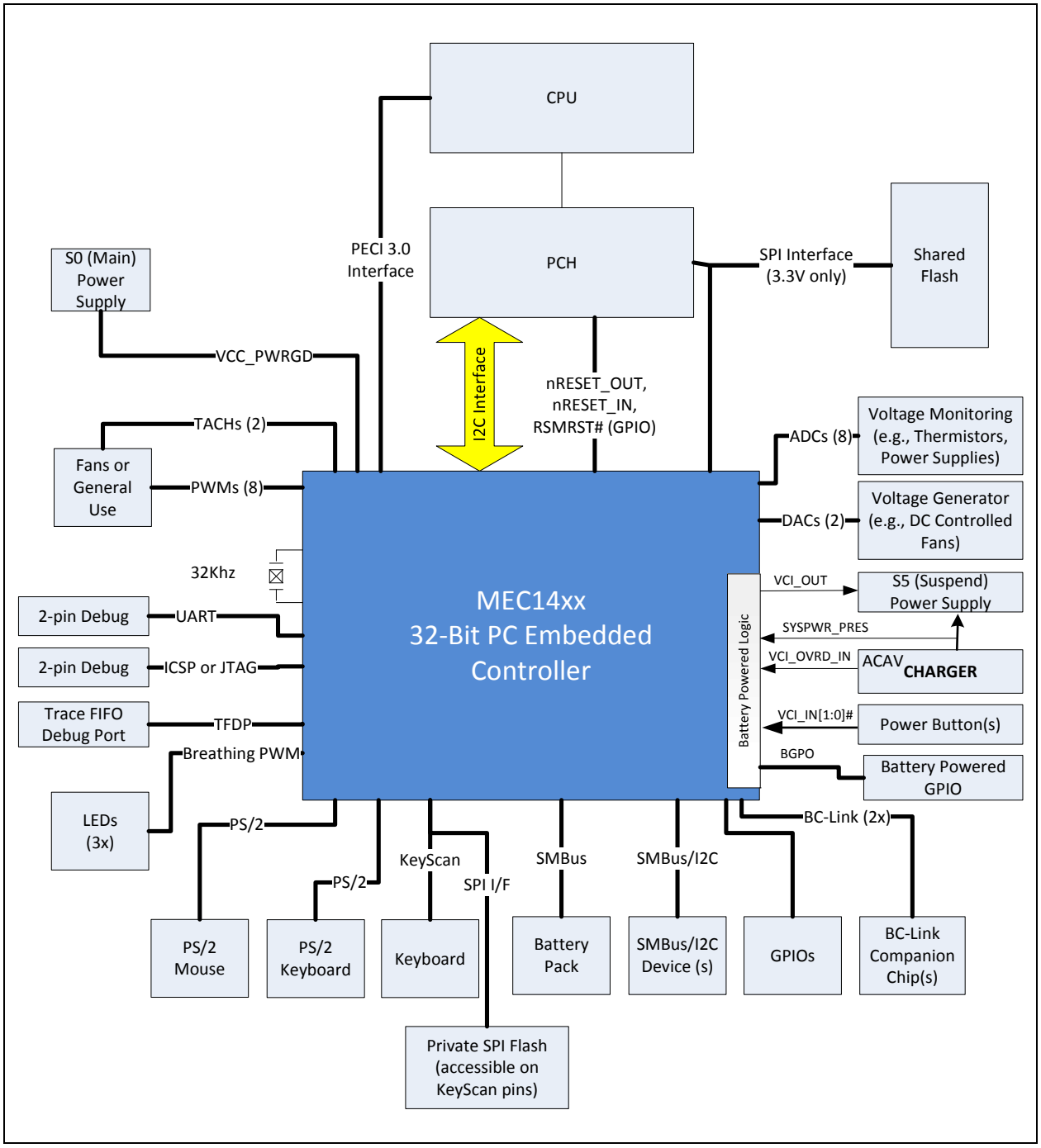
1.4.1 LPC HOST SYSTEM BLOCK DIAGRAM



1.4.2 ESPI HOST SYSTEM BLOCK DIAGRAM



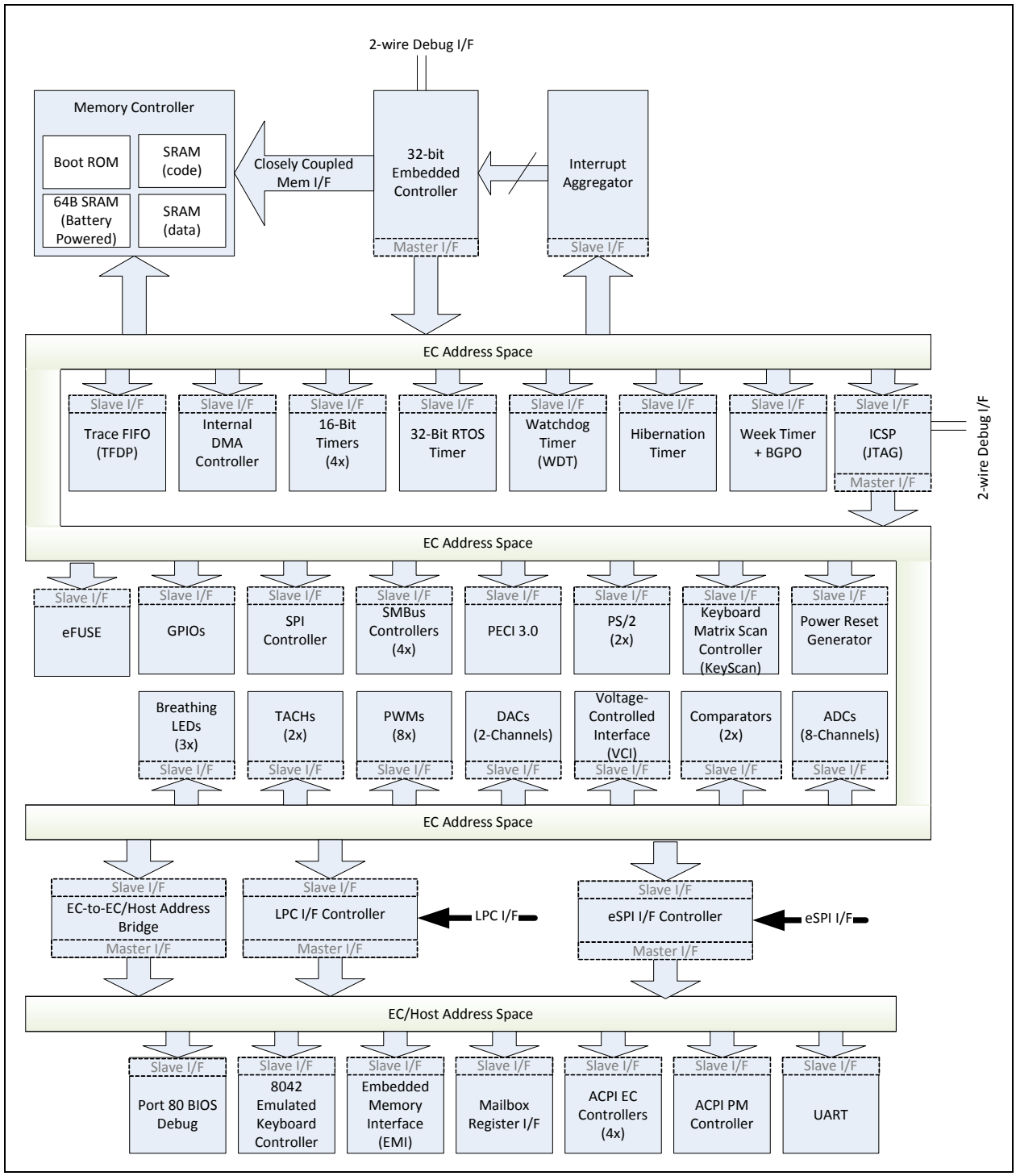
1.4.3 I2C HOST SYSTEM BLOCK DIAGRAM



1.5 MEC140x Internal Address Spaces

The Internal Embedded Controller can access any register in the EC Address Space or Host Address Space. The LPC and eSPI Host Controllers can directly access peripheral registers in the Host Address Space. If the I2C interface is used as the Host Interface, access to all the IP Peripherals is dependent on the EC firmware.

Note:	The eSPI and LPC Host Controllers also have access to the SRAM data space via the SRAM Memory BARs, which is not illustrated below.



Note: Not all features shown are available on all devices. Refer to Products on page 3 for a list of the features by device.

2.0 PIN CONFIGURATION

2.1 Description

The Pin Configuration chapter includes Pin Lists, Pin Description, Pin Multiplexing, Notes for Tables in this Chapter, Pin States After VTR Power-On, and Packages.

2.2 Terminology and Symbols for Pins/Buffers

2.2.1 BUFFER TERMINOLOGY

Term	Definition
Pin Ref. Number	There is a unique reference number for each pin name.
#	The '#' sign at the end of a signal name indicates an active-low signal
n	The lowercase 'n' preceding a signal name indicates an active-low signal
PWR	Power
I	Digital Input
IS	Input with Schmitt Trigger
I_AN	Analog Input
O	Push-Pull Output
OD	Open Drain Output
IO	Bi-directional pin
IOD	Bi-directional pin with Open Drain Output
PIO	Programmable as Input, Output, Open Drain Output, Bi-directional or Bi-directional with Open Drain Output.
PCI_I	Input. These pins meet the PCI 3.3V AC and DC Characteristics. (Note 2-1)
PCI_O	Output. These pins meet the PCI 3.3V AC and DC Characteristics. (Note 2-1)
PCI_OD	Open Drain Output. These pins meet the PCI 3.3V AC and DC Characteristics. (Note 2-1)
PCI_IO	Input/Output These pins meet the PCI 3.3V AC and DC Characteristics. (Note 2-1)
PCI_ICLK	Clock Input. These pins meet the PCI 3.3V AC and DC Characteristics and timing. (Note 2-2)
PCI_PIO	Programmable as Input, Output, Open Drain Output, Bi-directional or Bi-directional with Open Drain Output. These pins meet the PCI 3.3V AC and DC Characteristics. (Note 2-1).
PECI_IO	PECI Input/Output. These pins operate at the processor voltage level (VREF_CPU)
SB-TSI	SB-TSI Input/Output. These pins operate at the processor voltage level (VREF_CPU)

Note 2-1 See the "PCI Local Bus Specification," Revision 2.1, Section 4.2.2.

Note 2-2 See the "PCI Local Bus Specification," Revision 2.1, Section 4.2.2 and 4.2.3.

2.2.2 PIN NAMING CONVENTIONS

1. Pin Name is composed of the multiplexed options separated by '/'. E.g., GPIOxxxx/SignalA/SignalB.
2. The first signal shown in a pin name is the default signal. E.g., GPIOxxxx/SignalA/SignalB means the GPIO is the default signal.
3. Parenthesis '()' are used to list aliases or alternate functionality for a single mux option. E.g. GPIOxxx(Alias)/SignalA/SignalB. The Alias is the intended usage for a specific GPIO. E.g., GPIOxxx(ICSP_DATA) is intended to indicate that ICSP_DATA signal may come out on this pin when the Mux Control is set for GPIOxxx. In this case, enabling the test mode takes precedence over the Mux Control selection.
4. Square brackets '[ ]' are used to indicate there is a Strap Option on a pin. This is always shown as the last signal on the Pin Name.
5. Signal Names appended with a numeric value indicates the Instance Number, except for SMBus Pins. E.g., PWM0, PWM1, etc. indicates that PWM0 is the PWM output for PWM Instance 0, PWM1 is the PWM output for PWM Instance 1, etc. Note that this same instance number is shown in the Register Base Address tables linking

the specific PWM block instance to a specific signal on the pinout. The instance number may be omitted if there in only one instance of the IP block implemented.

Note: The numeric value appended to the end of the SMBus pins indicates they are 1.8V I/O signaling. E.g. SMB03_DATA vs SMB03_DATA18. The SMB03_DATA signal uses standard 3.3V I/O signaling. The SMB03_DATA18 signal operates at 1.8V I/O signaling levels.

6. SMBus Port pins can be mapped to any SMB Controller. The number in the SMBus signal names (SMBxx_DATA) indicates the port value. E.g. SMB01_DATA represents SMBus Data Port 1

2.3 Notes for Tables in this Chapter

Note	Description
Note 1	The LAD and SER_IRQ pins require an external weak pull-up resistor of 10k-100k ohms.
Note 2	The ICSP_MCLR pin is used to enable JTAG. There is an internal pull-up on this pin to keep it from entering debug mode. When debug mode is entered the ICSP_DATA and ICSP_CLOCK signals are automatically enabled on their respective pins. The System Board Designer should leave the ICSP_MCLR pin as a no-connect.
Note 3	An external cap must be connected as close to the CAP pin/ball as possible with a routing resistance and CAP ESR of less than 100mohms. The capacitor value is 1uF and must be ceramic with X5R or X7R dielectric. The cap pin/ball should remain on the top layer of the PCB and traced to the CAP. Avoid adding vias to other layers to minimize inductance.
Note 4	This SMBus ports supports 1 Mbps operation as defined by I2C. For 1 Mbps I2C recommended capacitance/pull-up relationships from Intel, refer to the Shark Bay platform guide, Intel ref number 486714. Refer to the PCH - SMBus 2.0/SMLink Interface Design Guidelines, Table 20-5 Bus Capacitance/Pull-Up Resistor Relationship.
Note 5	RESET_OUT# pin must be pulled to ground via an external 8.2k ohm resistor. This will ensure the glitch-free tristate GPIO input will not glitch high on a power on reset (POR) event.
Note 6	In order to achieve the lowest leakage current when both PECI and SB TSI are not used, set the VREF_CPU Disable bit to 1.
Note 7	The BC DAT pin requires a weak pull up resistor (100 K Ohms).
Note 8	The voltage on the ADC pins must not exceed 3.6 V or damage to the device will occur.
Note 9	The XTAL1 pin should be left floating when using the XTAL2 pin for the single ended clock input.
Note 10	The Boot ROM manipulates the pins associated with the Shared SPI interface and the Private SPI interface to access the external flash. Before exiting, the Boot ROM tristates these interfaces by returning them to their default hardware state (i.e., GPIO input).
Note 11	When the SMBxx_xxxx18 functions are selected, the pins operate at 1.8V I/O signal levels.
Note 12	The GPIO assignment on this pin only provides interrupt and wakeup capability. This is provided by the Interrupt Detection field in the Pin Control register. The Mux control field in the Pin Control Register should not be set to 00 = GPIO or undesirable results may occur. In order to emphasize the prohibition on using the GPIO Signal Pin Function, the Pin Chapter does not list the GPIO signal pin function assigned to this pin; however, the GPIO chapter does so the interrupt can be used.
Note 13	This signal is a test signal used to detect when the internal 48MHz clock is toggling or stopped in heavy and deepest sleep modes.
Note 14	The VCI pins may be used as GPIOs. The VCI input signals are not gated by selecting the GPIO alternate function. Firmware must disable (i.e., gate) these inputs by writing the bits in the VCI Input Enable Register when the GPIO function is enabled.

Note	Description
Note 15	The KSI and KSO Key Scan pins require pull-up resistors. The system designer may opt to use either use the internal pull-up resistors or populate external pull-up resistors.
Note 16	If the eSPI Flash Channel is used for booting, the GPIO123/SHD_CS# pin must be used as RSMRST#. This pin will be driven high by the boot ROM code in order to activate the eSPI flash channel. If the SHD_SPI port is used for booting, then any unused GPIO may be used for RSMRST#.
Note 17	If the eSPI Flash Channel is used for booting, the GPIO135/SHD_IO2 pin must be used to determine that the primary power rails are stable before RSMRST# can be de-asserted. See the MEC140X/1X eSPI Addendum document for more details.
Note 18	If certain blocks are not used, then the associated voltage reference pin may be connected to ground, as follows: • if the ADC is not used and the block is disabled, ADC_VREF can be connected to VSS • if the DAC is not used and the block is disabled, DAC_VREF can be connected to VSS • if both PECI and SB TSI are not used and the GPIO033/PECI_DAT/SB_TSI_DAT and GPIO035/ SB-TSI_CLK pins are configured as GPIOs, then VREF_CPU can be connected to VSS.

2.4 Pin Lists

Note: The GPIO Pin Control registers for the Pads that are not bonded out to pins or balls in the smaller package have been defaulted to their inactive state and are read-only. These pins cannot be modified by the downloaded firmware located in SRAM. No special handling required.

2.4.1 MEC140X PIN LIST

MEC140x		
128-pin VTQFP	**144-pin WFBGA**	**Pin Name**
1	L10	GPIO157/LED0/TST_CLK_OUT
2	N13	GPIO027/KSO00/PVT_IO1
3	M12	GPIO001/SPI_CS#/32KHZ_OUT
4	M10	GPIO002/PWM7
5	G5	VTR
6	M13	GPIO005/SMB00_DATA/SMB00_DATA18/KSI2
7	L12	GPIO006/SMB00_CLK/SMB00_CLK18/KSI3
8	K11	GPIO007/SMB01_DATA/SMB01_DATA18
9	J11	GPIO010/SMB01_CLK/SMB01_CLK18
10	G9	GPIO011/nSMI/nEMI_INT
11	J7	GPIO012/SMB02_DATA/SMB02_DATA18
12	H12	GPIO013/SMB02_CLK/SMB02_CLK18
13	H8	nRESET_IN/GPIO014
14	L11	GPIO015/KSO01/PVT_CS#
15	H11	GPIO016/KSO02/PVT_SCLK
16	J12	GPIO017/KSO03/PVT_IO0
17	C9	VSS
18	F1	VR_CAP
19	H5	VTR
20	G11	GPIO020/CMP_VIN0

MEC140x		
128-pin VTQFP	**144-pin WFBGA**	**Pin Name**
21	H13	GPIO021/CMP_VIN1
22	G12	DAC_VREF
23	G13	GPIO160/DAC_0
24	F12	GPIO161/DAC_1
25	F11	GPIO165/CMP_VREF0
26	E11	GPIO166/CMP_VREF1/UART_CLK
27	F13	GPIO123/SHD_CS#
28	E12	GPIO133/SHD_IO0
29	D12	GPIO134/SHD_IO1
30	E13	GPIO135/SHD_IO2
31	C11	GPIO136/SHD_IO3
32	D13	GPIO126/SHD_SCLK
33	D11	GPIO062/SPI_IO3
34	C12	GPIO030/BCM_INT0#/PWM4
35	C13	GPIO031/BCM_DAT0/PWM5
36	B13	GPIO032/BCM_CLK0/PWM6
37	B11	GPIO045/BCM_INT1#/KSO04
38	B12	GPIO046/BCM_DAT1/KSO05
39	B10	GPIO047/BCM_CLK1/KSO06
40	A13	GPIO050/TACH0
41	A12	GPIO051/TACH1
42	A11	GPIO052/SPI_IO2
43	H6	VTR
44	C8	GPIO053/PWM0
45	B9	GPIO054/PWM1
46	A10	GPIO055/PWM2/KSO08/PVT_IO3
47	A9	GPIO056/PWM3
48	B8	GPIO057/VCC_PWRGD
49	B7	GPIO060/KBRST
50	A8	GPIO025/KSO07/PVT_IO2
51	C10	VSS
52	C7	GPIO026/PS2_CLK1B
53	A7	GPIO061/LPCPD#
54	H7	VTR_33_18
55	C6	GPIO063/SER_IRQ
56	B6	GPIO064/LRESET#
57	A6	GPIO034/PCI_CLK
58	B5	GPIO044/LFRAME#
59	A5	GPIO040/LAD0
60	A4	GPIO041/LAD1
61	C5	GPIO042/LAD2
62	C4	GPIO043/LAD3
63	B4	GPIO067/CLKRUN#
64	D1	VSS

MEC140x		
128-pin VTQFP	**144-pin WFBGA**	**Pin Name**
65	J5	VTR
66	C3	GPIO100/nEC_SCI
67	C2	GPIO101/SPI_CLK
68	A3	GPIO102/KSO09[CR_STRAP]
69	B3	GPIO103/SPI_IO0
70	A2	GPIO104/LED2
71	E2	GPIO105/SPI_IO1
72	C1	GPIO106/KSO10
73	D2	GPIO107/nRESET_OUT
74	B2	GPIO110/KSO11
75	F2	GPIO111/KSO12
76	A1	GPIO112/PS2_CLK1A/KSO13
77	G3	GPIO113/PS2_DAT1A/KSO14
78	E1	GPIO114/PS2_CLK0
79	B1	GPIO115/PS2_DAT0
80	G1	GPIO116/TFDP_DATA/UART_RX
81	G2	GPIO117/TFDP_CLK/UART_TX
82	J6	VTR
83	H2	GPIO120/CMP_VOUT1
84	D3	VSS
85	H1	GPIO124/CMP_VOUT0
86	H3	GPIO125/KSO15
87	K1	ICSP_MCLR
88	J1	GPIO127/PS2_DAT1B
89	K2	GPIO130/SMB03_DATA/SMB03_DATA18
90	J2	GPIO035/SB-TSI_CLK
91	L1	GPIO131/SMB03_CLK/SMB03_CLK18
92	M1	GPIO132/KSO16
93	N1	GPIO140/KSO17
94	K3	GPIO033/PECI_DAT/SB_TSI_DAT
95	L5	VREF_CPU
96	J3	GPIO141/SMB04_DATA/SMB04_DATA18
97	L3	GPIO142/SMB04_CLK/SMB04_CLK18
98	L4	GPIO143/KSI0/DTR#
99	L2	GPIO144/KSI1/DCD#
100	F3	VSS
101	M2	GPIO145(ICSP_CLOCK)
102	M3	GPIO146(ICSP_DATA)
103	G6	VTR
104	N2	GPIO147/KSI4/DSR#
105	M4	GPIO150/KSI5/RI#
106	N3	GPIO156/LED1
107	N4	GPIO151/KSI6/RTS#
108	N5	GPIO152/KSI7/CTS#

MEC140x		
128-pin VTQFP	**144-pin WFBGA**	**Pin Name**
109	N6	GPIO153/ADC4
110	L7	GPIO154/ADC3
111	M6	GPIO155/ADC2
112	M7	AVSS
113	L6	GPIO122/ADC1
114	N7	GPIO121/ADC0
115	M5	ADC_VREF
116	N8	GPIO022/ADC5
117	L9	GPIO023/ADC6/A20M
118	N9	GPIO024/ADC7
119	N10	BGPO/GPIO004
120	M9	SYSPWR_PRES/GPIO003
121	M8	VCI_OUT/GPIO036
122	K12	VBAT
123	J13	XTAL1
124	E3	VSS_VBAT
125	L13	XTAL2
126	N12	VCI_IN1#/GPIO162
127	N11	VCI_IN0#/GPIO163
128	M11	VCI_OVRD_IN/GPIO164
	H9	VSS
	J8	VSS
	J9	VSS
	K13	VSS
	E5	No Connect
	E6	No Connect
	E7	No Connect
	E8	No Connect
	E9	No Connect
	F5	No Connect
	F6	No Connect
	F7	No Connect
	F8	No Connect
	F9	No Connect
	G8	No Connect
	L8	No Connect

2.4.2 MEC141X PIN LIST

MEC141x		
128-pin VTQFP	**144-pin WFBGA**	**Pin Name**
1	L10	GPIO157/LED0/TST_CLK_OUT
2	N13	GPIO027/KSO00/PVT_IO1
3	M12	GPIO001/SPI_CS#/32KHZ_OUT
4	M10	GPIO002/PWM7
5	G5	VTR
6	M13	GPIO005/SMB00_DATA/SMB00_DATA18/KSI2
7	L12	GPIO006/SMB00_CLK/SMB00_CLK18/KSI3
8	K11	GPIO007/SMB01_DATA/SMB01_DATA18
9	J11	GPIO010/SMB01_CLK/SMB01_CLK18
10	G9	GPIO011/nSMI/nEMI_INT
11	J7	GPIO012/SMB02_DATA/SMB02_DATA18
12	H12	GPIO013/SMB02_CLK/SMB02_CLK18
13	H8	nRESET_IN/GPIO014
14	L11	GPIO015/KSO01/PVT_CS#
15	H11	GPIO016/KSO02/PVT_SCLK
16	J12	GPIO017/KSO03/PVT_IO0
17	C9	VSS
18	F1	VR_CAP
19	H5	VTR
20	G11	GPIO020/CMP_VIN0
21	H13	GPIO021/CMP_VIN1
22	G12	DAC_VREF
23	G13	GPIO160/DAC_0
24	F12	GPIO161/DAC_1
25	F11	GPIO165/CMP_VREF0
26	E11	GPIO166/CMP_VREF1/UART_CLK
27	F13	GPIO123/SHD_CS# [BSS_STRAP]
28	E12	GPIO133/SHD_IO0
29	D12	GPIO134/SHD_IO1
30	E13	GPIO135/SHD_IO2
31	C11	GPIO136/SHD_IO3
32	D13	GPIO126/SHD_SCLK
33	D11	GPIO062/SPI_IO3
34	C12	GPIO030/BCM_INT0#/PWM4
35	C13	GPIO031/BCM_DAT0/PWM5
36	B13	GPIO032/BCM_CLK0/PWM6
37	B11	GPIO045/BCM_INT1#/KSO04
38	B12	GPIO046/BCM_DAT1/KSO05
39	B10	GPIO047/BCM_CLK1/KSO06
40	A13	GPIO050/TACH0
41	A12	GPIO051/TACH1
42	A11	GPIO052/SPI_IO2

MEC141x		
128-pin VTQFP	**144-pin WFBGA**	**Pin Name**
43	H6	VTR
44	C8	GPIO053/PWM0
45	B9	GPIO054/PWM1
46	A10	GPIO055/PWM2/KSO08/PVT_IO3
47	A9	GPIO056/PWM3
48	B8	GPIO057/VCC_PWRGD
49	B7	GPIO060/KBRST
50	A8	GPIO025/KSO07/PVT_IO2
51	C10	VSS
52	C7	GPIO026/PS2_CLK1B
53	A7	GPIO061/LPCPD#/ESPI_RESET#
54	H7	VTR_33_18
55	C6	GPIO063/SER_IRQ/ESPI_ALERT#
56	B6	GPIO064/LRESET#
57	A6	GPIO034/PCI_CLK/ESPI_CLK
58	B5	GPIO044/LFRAME#/ESPI_CS#
59	A5	GPIO040/LAD0/ESPI_IO0
60	A4	GPIO041/LAD1/ESPI_IO1
61	C5	GPIO042/LAD2/ESPI_IO2
62	C4	GPIO043/LAD3/ESPI_IO3
63	B4	GPIO067/CLKRUN#
64	D1	VSS
65	J5	VTR
66	C3	GPIO100/nEC_SCI
67	C2	GPIO101/SPI_CLK
68	A3	GPIO102/KSO09[CR_STRAP]
69	B3	GPIO103/SPI_IO0
70	A2	GPIO104/LED2
71	E2	GPIO105/SPI_IO1
72	C1	GPIO106/KSO10
73	D2	GPIO107/nRESET_OUT
74	B2	GPIO110/KSO11
75	F2	GPIO111/KSO12
76	A1	GPIO112/PS2_CLK1A/KSO13
77	G3	GPIO113/PS2_DAT1A/KSO14
78	E1	GPIO114/PS2_CLK0
79	B1	GPIO115/PS2_DAT0
80	G1	GPIO116/TFDP_DATA/UART_RX
81	G2	GPIO117/TFDP_CLK/UART_TX
82	J6	VTR
83	H2	GPIO120/CMP_VOUT1
84	D3	VSS
85	H1	GPIO124/CMP_VOUT0
86	H3	GPIO125/KSO15

MEC141x		
128-pin VTQFP	**144-pin WFBGA**	**Pin Name**
87	K1	ICSP_MCLR
88	J1	GPIO127/PS2_DAT1B
89	K2	GPIO130/SMB03_DATA/SMB03_DATA18
90	J2	GPIO035/SB-TSI_CLK
91	L1	GPIO131/SMB03_CLK/SMB03_CLK18
92	M1	GPIO132/KSO16
93	N1	GPIO140/KSO17
94	K3	GPIO033/PECI_DAT/SB_TSI_DAT
95	L5	VREF_CPU
96	J3	GPIO141/SMB04_DATA/SMB04_DATA18
97	L3	GPIO142/SMB04_CLK/SMB04_CLK18
98	L4	GPIO143/KSI0/DTR#
99	L2	GPIO144/KSI1/DCD#
100	F3	VSS
101	M2	GPIO145(ICSP_CLOCK)
102	M3	GPIO146(ICSP_DATA)
103	G6	VTR
104	N2	GPIO147/KSI4/DSR#
105	M4	GPIO150/KSI5/RI#
106	N3	GPIO156/LED1
107	N4	GPIO151/KSI6/RTS#
108	N5	GPIO152/KSI7/CTS#
109	N6	GPIO153/ADC4
110	L7	GPIO154/ADC3
111	M6	GPIO155/ADC2
112	M7	AVSS
113	L6	GPIO122/ADC1
114	N7	GPIO121/ADC0
115	M5	ADC_VREF
116	N8	GPIO022/ADC5
117	L9	GPIO023/ADC6/A20M
118	N9	GPIO024/ADC7
119	N10	BGPO/GPIO004
120	M9	SYSPWR_PRES/GPIO003
121	M8	VCI_OUT/GPIO036
122	K12	VBAT
123	J13	XTAL1
124	E3	VSS_VBAT
125	L13	XTAL2
126	N12	VCI_IN1#/GPIO162
127	N11	VCI_IN0#/GPIO163
128	M11	VCI_OVRD_IN/GPIO164
	H9	VSS
	J8	VSS

MEC141x		
128-pin VTQFP	**144-pin WFBGA**	**Pin Name**
	J9	VSS
	K13	VSS
	E5	No Connect
	E6	No Connect
	E7	No Connect
	E8	No Connect
	E9	No Connect
	F5	No Connect
	F6	No Connect
	F7	No Connect
	F8	No Connect
	F9	No Connect
	G8	No Connect
	L8	No Connect

2.5 Non 5 Volt Tolerant Pins

There are no 5 Volt tolerant pins in the MEC140x/1x.

2.6 1.8V or 3.3V I/O Pins

The following signals are powered by the VTR_33_18 power supply. This supply determines the operating voltage range for these signals.

Note: The LPC Interface signals require the VTR_33_18 power pin to be connected to the 3.3V VTR rail. The eSPI Interface signals require the VTR_33_18 power pin to be connected to the 1.8V rail. The GPIO signals on these pins may operate at either 1.8V or 3.3V.

- GPIO061/LPCPD#/ESPI_RESET#
- VTR_33_18
- GPIO063/SER_IRQ/ESPI_ALERT#
- GPIO064/LRESET#
- GPIO034/PCI_CLK/ESPI_CLK
- GPIO044/LFRAME#/ESPI_CS#
- GPIO040/LAD0/ESPI_IO0
- GPIO041/LAD1/ESPI_IO1
- GPIO042/LAD2/ESPI_IO2
- GPIO043/LAD3/ESPI_IO3
- GPIO067/CLKRUN#

2.7 POR Glitch Protected Pins

All pins have POR output glitch protection. POR output glitch protection ensures that pins will have a steady-state output during a VTR POR.

2.8 Non Backdrive Protected Pins

TABLE 2-1: lists pins which do not have backdrive protection. If the power supply used to power the buffer of the pin (VTR or VTR_33_18) is off none of these pins are allowed to be above 0V to prevent back-drive onto the associated power supply. The Power Supply used to power the buffer is shown in the Signal Power Well column of the Pin Multiplexing Tables in **Section 2.0 "Pin Configuration"**.

TABLE 2-1: MEC140X/1X NON BACKDRIVE PROTECTED PINS

Pin Name
DAC_VREF
GPIO160/DAC_0
GPIO161/DAC_1
GPIO165/CMP_VREF0
GPIO166/CMP_VREF1/UART_CLK
GPIO020/CMP_VIN0
GPIO021/CMP_VIN1
GPIO035/SB-TSI_CLK
GPIO033/PECI_DAT/SB_TSI_DAT
VREF_CPU
ADC_VREF
GPIO153/ADC4
GPIO154/ADC3
GPIO155/ADC2
GPIO122/ADC1
GPIO121/ADC0
GPIO022/ADC5
GPIO023/ADC6/A20M
GPIO024/ADC7
GPIO040/LAD0
GPIO041/LAD1
GPIO042/LAD2
GPIO043/LAD3
GPIO063/SER_IRQ
XTAL1
XTAL2

2.9 Pin Description

Note: See Section 2.3, "Notes for Tables in this Chapter," on page 13 for notes that are referenced in the Pin Description table.

Interface	Signal Name	Description	Notes
Analog Data Acquisition Interface	ADC0	ADC channel 0	Note 8
Analog Data Acquisition Interface	ADC1	ADC channel 1	Note 8
Analog Data Acquisition Interface	ADC2	ADC channel 2	Note 8

Interface	Signal Name	Description	Notes
Analog Data Acquisition Interface	ADC3	ADC channel 3	Note 8
Analog Data Acquisition Interface	ADC4	ADC channel 4	Note 8
Analog Data Acquisition Interface	ADC5	ADC channel 5	Note 8
Analog Data Acquisition Interface	ADC6	ADC channel 6	Note 8
Analog Data Acquisition Interface	ADC7	ADC channel 7	Note 8
BC-Link Interface	BCM_CLK0	BC-Link Master clock	
BC-Link Interface	BCM_CLK1	BC-Link Master clock	
BC-Link Interface	BCM_DAT0	BC-Link Master data I/O	Note 7
BC-Link Interface	BCM_DAT1	BC-Link Master data I/O	Note 7
BC-Link Interface	BCM_INT0#	BC-Link Master interrupt	
BC-Link Interface	BCM_INT1#	BC-Link Master interrupt	
Comparator Interface	CMP_VIN0	Comparator 0 Positive Input	
Comparator Interface	CMP_VIN1	Comparator 1 Positive Input	
Comparator Interface	CMP_VOUT0	Comparator 0 Output	
Comparator Interface	CMP_VOUT1	Comparator 1 Output	
Comparator Interface	CMP_VREF0	Comparator 0 Negative Input	
Comparator Interface	CMP_VREF1	Comparator 1 Negative Input	
Digatal to Analog (DAC) Interface	DAC_0	DAC channel 0	
Digatal to Analog (DAC) Interface	DAC_1	DAC channel 1	
eSPI HOST INTERFACE	ESPI_ALERT#	eSPI Alert	
eSPI HOST INTERFACE	ESPI_CLK	eSPI Clock	
eSPI HOST INTERFACE	ESPI_CS#	eSPI Chip Select	
eSPI HOST INTERFACE	ESPI_IO0	eSPI Data Pin 0	
eSPI HOST INTERFACE	ESPI_IO1	eSPI Data Pin 1	
eSPI HOST INTERFACE	ESPI_IO2	eSPI Data Pin 2	
eSPI HOST INTERFACE	ESPI_IO3	eSPI Data Pin 3	
eSPI HOST INTERFACE	ESPI_RESET#	eSPI Reset	
GPIO Interface	GPIO	General Purpose Input Output Pins	
ICSP Interface	ICSP_CLOCK	2-Wire Debug Clock	
ICSP Interface	ICSP_DATA	2-Wire Debug Data	
ICSP Interface	ICSP_MCLR	2-Wire Debug Master Reset	Note 2
Keyboard Scan Interface	KSI0	Keyboard Scan Matrix Input 0	Note 15
Keyboard Scan Interface	KSI1	Keyboard Scan Matrix Input 1	Note 15
Keyboard Scan Interface	KSI2	Keyboard Scan Matrix Input 2	Note 15
Keyboard Scan Interface	KSI3	Keyboard Scan Matrix Input 3	Note 15
Keyboard Scan Interface	KSI4	Keyboard Scan Matrix Input 4	Note 15
Keyboard Scan Interface	KSI5	Keyboard Scan Matrix Input 5	Note 15
Keyboard Scan Interface	KSI6	Keyboard Scan Matrix Input 6	Note 15
Keyboard Scan Interface	KSI7	Keyboard Scan Matrix Input 7	Note 15
Keyboard Scan Interface	KSO00	Keyboard Scan Matrix Output 0	Note 15

Interface	Signal Name	Description	Notes
Keyboard Scan Interface	KSO01	Keyboard Scan Matrix Output 1	Note 15
Keyboard Scan Interface	KSO02	Keyboard Scan Matrix Output 2	Note 15
Keyboard Scan Interface	KSO03	Keyboard Scan Matrix Output 3	Note 15
Keyboard Scan Interface	KSO04	Keyboard Scan Matrix Output 4	Note 15
Keyboard Scan Interface	KSO05	Keyboard Scan Matrix Output 5	Note 15
Keyboard Scan Interface	KSO06	Keyboard Scan Matrix Output 6	Note 15
Keyboard Scan Interface	KSO07	Keyboard Scan Matrix Output 7	Note 15
Keyboard Scan Interface	KSO08	Keyboard Scan Matrix Output 8	Note 15
Keyboard Scan Interface	KSO09	Keyboard Scan Matrix Output 9	Note 15
Keyboard Scan Interface	KSO10	Keyboard Scan Matrix Output 10	Note 15
Keyboard Scan Interface	KSO11	Keyboard Scan Matrix Output 11	Note 15
Keyboard Scan Interface	KSO12	Keyboard Scan Matrix Output 12	Note 15
Keyboard Scan Interface	KSO13	Keyboard Scan Matrix Output 13	Note 15
Keyboard Scan Interface	KSO14	Keyboard Scan Matrix Output 14	Note 15
Keyboard Scan Interface	KSO15	Keyboard Scan Matrix Output 15	Note 15
Keyboard Scan Interface	KSO16	Keyboard Scan Matrix Output 16	Note 15
Keyboard Scan Interface	KSO17	Keyboard Scan Matrix Output 17	Note 15
LPC HOST INTERFACE	CLKRUN#	PCI Clock Control	
LPC HOST INTERFACE	LAD0	LPC Multiplexed command, address and data bus Bit 0.	Note 1
LPC HOST INTERFACE	LAD1	LPC Multiplexed command, address and data bus Bit 1.	Note 1
LPC HOST INTERFACE	LAD2	LPC Multiplexed command, address and data bus Bit 2.	Note 1
LPC HOST INTERFACE	LAD3	LPC Multiplexed command, address and data bus Bit 3.	Note 1
LPC HOST INTERFACE	LFRAME#	Frame signal. Indicates start of new cycle and termination of broken cycle	
LPC HOST INTERFACE	LPCPD#	LPC Power Down	
LPC HOST INTERFACE	LRESET#	LPC Reset. LRESET# is the same as the system PCI reset, PCIRST#	
LPC HOST INTERFACE	nEC_SCI	Power Management Event	
LPC HOST INTERFACE	nEMI_INT	EMI Interrupt Output	
LPC HOST INTERFACE	nSMI	SMI Output	
LPC HOST INTERFACE	PCI_CLK	PCI Clock	
LPC HOST INTERFACE	SER_IRQ	Serial IRQ	Note 1
Master Clock Interface	XTAL1	32.768 KHz Crystal Output	
Master Clock Interface	XTAL2	32.768 KHz Crystal Input (single-ended 32.768 KHz clock input)	
MISC Functions	32KHZ_OUT	32.768 KHz Digital Output	
MISC Functions	A20M	KBD GATEA20 Output	
MISC Functions	KBRST	CPU_RESET	
MISC Functions	LED0	LED (Bllinking/Breathing PWM) PWM Output 0	
MISC Functions	LED1	LED (Bllinking/Breathing PWM) PWM Output 1	

Interface	Signal Name	Description	Notes
MISC Functions	LED2	LED (Bllinking/Breathing PWM) PWM Output 2	
MISC Functions	nRESET_IN	External System Reset Input	
MISC Functions	nRESET_OUT	EC-driven External System Reset Output	Note 5
MISC Functions	TFDP_CLK	Trace FIFO debug port - clock	
MISC Functions	TFDP_DATA	Trace FIFO debug port - data	
MISC Functions	VCC_PWRGD	System Main Power Indication	
MISC Functions	XNOR	Test Output	
PECI Interface	PECI_DAT	PECI Bus	Note 12
Power Interface	ADC_VREF	ADC Reference Voltage	Note 18
Power Interface	AVSS	Analog ADC supply associated ground	
Power Interface	DAC_VREF	DAC Reference Voltage	Note 18
Power Interface	VBAT	VBAT supply	
Power Interface	VR_CAP	Internal Voltage Regulator Capacitor	Note 3
Power Interface	VREF_CPU	Processor Interface Voltage Reference	Note 6, Note 18
Power Interface	VSS	VTR associated ground	
Power Interface	VSS_VBAT	VBAT associated ground	
Power Interface	VTR	VTR Suspend Power Supply	
Power Interface	VTR_33_18	Host Interface Power Supply	
PS/2 Interface	PS2_CLK0	PS/2 clock 0 (PS2_CLK)	
PS/2 Interface	PS2_CLK1A	PS/2 clock 1 - Port A (PS2_CLK)	
PS/2 Interface	PS2_CLK1B	PS/2 clock 1 - Port B (PS2_CLK)	
PS/2 Interface	PS2_DAT0	PS/2 data 0 (PS2_DAT)	
PS/2 Interface	PS2_DAT1A	PS/2 data 1 - Port A (PS2_DAT)	
PS/2 Interface	PS2_DAT1B	PS/2 data 1 - Port B (PS2_DAT)	
PWM	PWM0	Pulse Width Modulator Output 0	
PWM	PWM1	Pulse Width Modulator Output 1	
PWM	PWM2	Pulse Width Modulator Output 2	
PWM	PWM3	Pulse Width Modulator Output 3	
PWM	PWM4	Pulse Width Modulator Output 4	
PWM	PWM5	Pulse Width Modulator Output 5	
PWM	PWM6	Pulse Width Modulator Output 6	
PWM	PWM7	Pulse Width Modulator Output 7	
Tachometer	TACH0	Fan Tachometer Input 0	
Tachometer	TACH1	Fan Tachometer Input 1	
SMBus Interface	SB_TSI_DAT	SMBus Controller AMD-TSI Port Data	Note 12
SMBus Interface	SB-TSI_CLK	SMBus Controller AMD-TSI Port Clock	
SMBus Interface	SMB00_CLK	SMBus Controller Port 0 Clock	Note 4, Note 11
SMBus Interface	SMB00_DATA	SMBus Controller Port 0 Data	Note 4, Note 11

Interface	Signal Name	Description	Notes
SMBus Interface	SMB01_CLK	SMBus Controller Port 1 Clock	Note 4, Note 11
SMBus Interface	SMB01_DATA	SMBus Controller Port 1 Data	Note 4, Note 11
SMBus Interface	SMB02_CLK	SMBus Controller Port 2 Clock	Note 4, Note 11
SMBus Interface	SMB02_DATA	SMBus Controller Port 2 Data	Note 4, Note 11
SMBus Interface	SMB03_CLK	SMBus Controller Port 3 Clock	Note 4, Note 11
SMBus Interface	SMB03_DATA	SMBus Controller Port 3 Data	Note 4, Note 11
SMBus Interface	SMB04_CLK	SMBus Controller Port 4 Clock	Note 4, Note 11
SMBus Interface	SMB04_DATA	SMBus Controller Port 4 Data	Note 4, Note 11
Quad SPI Master Controller Interface	PVT_CS#	Private SPI Chip Select (SPI_CS#)	
Quad SPI Master Controller Interface	PVT_IO0	Private SPI Data 0 (SPI_IO0)	Note 10
Quad SPI Master Controller Interface	PVT_IO1	Private SPI Data 1 (SPI_IO1)	Note 10
Quad SPI Master Controller Interface	PVT_IO2	Private SPI Data 2 (SPI_IO2)	Note 10
Quad SPI Master Controller Interface	PVT_IO3	Private SPI Data 3 (SPI_IO3)	Note 10
Quad SPI Master Controller Interface	PVT_SCLK	Private SPI Clock (SPI_CLK)	Note 10
Quad SPI Master Controller Interface	SHD_CS#	Shared SPI Chip Select (SPI_CS#)	
Quad SPI Master Controller Interface	SHD_IO0	Shared SPI Data 0 (SPI_IO0)	Note 10
Quad SPI Master Controller Interface	SHD_IO1	Shared SPI Data 1 (SPI_IO1)	Note 10
Quad SPI Master Controller Interface	SHD_IO2	Shared SPI Data 2 (SPI_IO2)	Note 10
Quad SPI Master Controller Interface	SHD_IO3	Shared SPI Data 3 (SPI_IO3)	Note 10
Quad SPI Master Controller Interface	SHD_SCLK	Shared SPI Clock (SPI_CLK)	Note 10
Quad SPI Master Controller Interface	SPI_CLK	General Purpose SPI Clock (SPI_-CLK)	
Quad SPI Master Controller Interface	SPI_CS#	General Purpose SPI Chip Select (SPI_CS#)	
Quad SPI Master Controller Interface	SPI_IO0	General Purpose SPI Data 0 (SPI_IO0)	
Quad SPI Master Controller Interface	SPI_IO1	General Purpose SPI Data 1 (SPI_IO1)	
Quad SPI Master Controller Interface	SPI_IO2	General Purpose SPI Data 2 (SPI_IO2)	

Interface	Signal Name	Description	Notes
Quad SPI Master Controller Interface	SPI_IO3	General Purpose SPI Data 3 (SPI_IO3)	
UART Port	CTS#	Clear to Send Input	
UART Port	DCD#	Data Carrier Detect Input	
UART Port	DSR#	Data Set Ready Input	
UART Port	DTR#	Data Terminal Ready Output	
UART Port	RI#	Ring Indicator Input	
UART Port	RTS#	Request to Send Output	
UART Port	UART_CLK	UART Baud Clock Input	
UART Port	UART_RX	UART Receive Data (RXD)	
UART Port	UART_TX	UART Transmit Data (TXD)	
VBAT-Powered Control Interface	BGPO	Battery Powered General Purpose Output	
VBAT-Powered Control Interface	SYSPWR_PRES	Battery Powered System Power Present Input	Note 12
VBAT-Powered Control Interface	VCI_IN0#	Input can cause wakeup or interrupt event	Note 14
VBAT-Powered Control Interface	VCI_IN1#	Input can cause wakeup or interrupt event	Note 14
VBAT-Powered Control Interface	VCI_OUT	Output from combinatorial logic and/or EC	
VBAT-Powered Control Interface	VCI_OVRD_IN	Input can cause wakeup or interrupt event	Note 14

2.10 Pin Multiplexing

Multifunction Pin Multiplexing in the MEC140x/1x is controlled by the GPIO Interface and illustrated in the Pin Multiplexing Table in this section. See Section 2.3, "Notes for Tables in this Chapter," on page 13 for notes that are referenced in the Pin Multiplexing Table. See Pin Control Register on page 329 for Pin Multiplexing programming details.

Pin signal functions that exhibit power domain emulation (see Pin Multiplexing Table below) have a different power supply designation in the “Emulated Power Well” column and “Signal Power Well“ columns.

2.10.1 VCC POWER DOMAIN EMULATION

The System Runtime Supply power VCC is not connected to the MEC140x/1x. The VCC_PWRGD signal is used to indicate when power is applied to the System Runtime Supply.

Pin signal functions with VCC power domain emulation are documented in the Pin Multiplexing Table as “Signal Power Well“= VTR and “Emulated Power Well” = VCC. These pins are powered by VTR and controlled by the VCC_PWRGD signal input. Outputs on VCC power domain emulation pin signal functions are tri-stated when VCC_PWRGD is not asserted and are functional when VCC_PWRGD is active. Inputs on VCC power domain emulation pin signal functions are gated according as defined by the Gated State column in the following tables.

Power well emulation for GPIOs and for signals that are multiplexed with GPIO signals is controlled by the Power Gating Signals field in the GPIO Pin Control Register.

2.10.2 PIN MULTIPLEXING TABLE

In the following table, the columns have the following meanings:

MUX

If the pin has an associated GPIO, then the MUX column refers to the Mux Control field in the GPIO Pin Control Register. Setting the Mux Control field to value listed in the row will configure the pin for the signal listed in the Signal column on the same row. The row marked “Default” is the setting that is assigned on system reset.

If there is no GPIO associated with a pin, then the pin has a single function.

SIGNAL

This column lists the signals that can appear on each pin, as configured by the MUX control.

BUFFER TYPE

Pin buffer types are defined in Table 42-5, “DC Electrical Characteristics,” on page 491.

Note that all GPIO pins are of buffer type PIO, which may be configured as input/output, push-pull/OD etc. via the GPIO Pin Control Register and Pin Control Register 2. There are some pins where the buffer type is configured by the alternate function selection, in which case that buffer type is shown in this column.

DEFAULT BUFFER OPERATION

This column gives the pin behavior following the power-up of VTR. All GPIO pins are programmable after this event. This default pin behavior corresponds to the row marked “Default” in the MUX column.

SIGNAL POWER WELL

This column defines the power well that powers the pin.

EMULATED POWER WELL

Power well emulation for GPIOs and for signals that are multiplexed with GPIO signals is controlled by the Power Gating Signals field in the GPIO Pin Control Register. Power well emulation for signals that are not multiplexed with GPIO signals is defined by the entries in this column. See Section 2.10.1, "VCC Power Domain Emulation".

GATED STATE

This column defines the internal value of an input signal when either its emulated power well is inactive or it is not selected by the GPIO alternate function MUX. A value of “No Gate” means that the internal signal always follows the pin even when the emulated power well is inactive.

Note:	Gated state is only meaningful to the operation of input signals. A gated state on an output pin defines the internal behavior of the GPIO MUX and does not imply pin behavior.

TABLE 2-2: MEC140X PIN MULTIPLEXING

MEC140x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
1	Default: 0	GPIO157	PIO	I-4	VTR	VTR/VCC	No Gate	
1	1	LED0	PIO		VTR	VTR	Reserved	
1	2	TST_CLK_OUT	PIO		VTR	VTR	Reserved	Note 13
1	3	Reserved	Reserved		Reserved	Reserved		
1	Strap							
2	Default: 0	GPIO027	PIO	I-4	VTR	VTR/VCC	No Gate	
2	1	KSO00	PIO		VTR	VTR	Reserved	Note 15
2	2	PVT_IO1	PIO		VTR	VTR	Low	Note 10

MEC140x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
2	3	Reserved	Reserved		Reserved	Reserved		
2	Strap							
3	Default: 0	GPIO001	PIO	I-4	VTR	VTR/VCC	No Gate	
3	1	SPI_CS#	PIO		VTR	VTR	Reserved	
3	2	32KHZ_OUT	PIO		VTR	VTR	Reserved	
3	3	Reserved	Reserved		Reserved	Reserved		
3	Strap							
4	Default: 0	GPIO002	PIO	I-4	VTR	VTR/VCC	No Gate	
4	1	PWM7	PIO		VTR	VTR	Reserved	
4	2	Reserved	Reserved		Reserved	Reserved		
4	3	Reserved	Reserved		Reserved	Reserved		
4	Strap							
5		VTR	PWR		PWR	PWR		
5								
5								
5								
5	Strap							
6	Default: 0	GPIO005	PIO	I-4	VTR	VTR/VCC	No Gate	
6	1	SMB00_DATA	PIO		VTR	VTR	High	Note 4
6	2	SMB00_DATA18	PIO		VTR	VTR	High	Note 11
6	3	KSI2	PIO		VTR	VTR	Low	Note 15
6	Strap							
7	Default: 0	GPIO006	PIO	I-4	VTR	VTR/VCC	No Gate	
7	1	SMB00_CLK	PIO		VTR	VTR	High	Note 4
7	2	SMB00_CLK18	PIO		VTR	VTR	High	Note 11
7	3	KSI3	PIO		VTR	VTR	Low	Note 15
7	Strap							
8	Default: 0	GPIO007	PIO	I-4	VTR	VTR/VCC	No Gate	
8	1	SMB01_DATA	PIO		VTR	VTR	High	Note 4
8	2	SMB01_DATA18	PIO		VTR	VTR	High	Note 11
8	3	Reserved	Reserved		Reserved	Reserved		
8	Strap							
9	Default: 0	GPIO010	PIO	I-4	VTR	VTR/VCC	No Gate	
9	1	SMB01_CLK	PIO		VTR	VTR	High	Note 4
9	2	SMB01_CLK18	PIO		VTR	VTR	High	Note 11
9	3	Reserved	Reserved		Reserved	Reserved		
9	Strap							
10	Default: 0	GPIO011	PIO	I-4	VTR	VTR/VCC	No Gate	
10	1	nSMI	PIO		VTR	VTR	Reserved	
10	2	nEMI_INT	PIO		VTR	VTR	Reserved	
10	3	Reserved	Reserved		Reserved	Reserved		
10	Strap							

MEC140x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
11	Default: 0	GPIO012	PIO	I-4	VTR	VTR/VCC	No Gate	
11	1	SMB02_DATA	PIO		VTR	VTR	High	Note 4
11	2	SMB02_DATA18	PIO		VTR	VTR	High	Note 11
11	3	Reserved	Reserved		Reserved	Reserved		
11	Strap							
12	Default: 0	GPIO013	PIO	I-4	VTR	VTR/VCC	No Gate	
12	1	SMB02_CLK	PIO		VTR	VTR	High	Note 4
12	2	SMB02_CLK18	PIO		VTR	VTR	High	Note 11
12	3	Reserved	Reserved		Reserved	Reserved		
12	Strap							
13	0	GPIO014	PIO		VTR	VTR/VCC	No Gate	
13	Default: 1	nRESET_IN	PIO	I-4	VTR	VTR	High	
13	2	Reserved	Reserved		Reserved	Reserved		
13	3	Reserved	Reserved		Reserved	Reserved		
13	Strap							
14	Default: 0	GPIO015	PIO	I-4	VTR	VTR/VCC	No Gate	
14	1	KSO01	PIO		VTR	VTR	Reserved	Note 15
14	2	PVT_CS#	PIO		VTR	VTR	Reserved	Note 10
14	3	Reserved	Reserved		Reserved	Reserved		
14	Strap							
15	Default: 0	GPIO016	PIO	I-4	VTR	VTR/VCC	No Gate	
15	1	KSO02	PIO		VTR	VTR	Reserved	Note 15
15	2	PVT_SCLK	PIO		VTR	VTR	Reserved	Note 10
15	3	Reserved	Reserved		Reserved	Reserved		
15	Strap							
16	Default: 0	GPIO017	PIO	I-4	VTR	VTR/VCC	No Gate	
16	1	KSO03	PIO		VTR	VTR	Reserved	Note 15
16	2	PVT_IO0	PIO		VTR	VTR	Low	Note 10
16	3	Reserved	Reserved		Reserved	Reserved		
16	Strap							
17		VSS	PWR		PWR	PWR		
17								
17								
17								
17	Strap							
18		VR_CAP	PWR		PWR	PWR		Note 3
18								
18								
18								
18	Strap							

MEC140x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
19		VTR	PWR		PWR	PWR		
19								
19								
19								
19	Strap							
20	Default: 0	GPIO020	PIO	I-4	VTR	VTR/VCC	No Gate	
20	1	CMP_VIN0	I_AN		I_AN	I_AN	No Gate	
20	2	Reserved	Reserved		Reserved	Reserved		
20	3	Reserved	Reserved		Reserved	Reserved		
20	Strap							
21	Default: 0	GPIO021	PIO	I-4	PWR	VTR/VCC	No Gate	
21	1	CMP_VIN1	I_AN		I_AN	I_AN	No Gate	
21	2	Reserved	Reserved		Reserved	Reserved		
21	3	Reserved	Reserved		Reserved	Reserved		
21	Strap							
22	0	Reserved	Reserved		Reserved	Reserved		
22	Default: 1	DAC_VREF	DAC_VREF		DAC_VREF	DAC_VREF	No Gate	Note 18
22	2	Reserved	Reserved		Reserved	Reserved		
22	3	Reserved	Reserved		Reserved	Reserved		
22	Strap							
23	Default: 0	GPIO160	PIO	I-4	VTR	VTR/VCC	No Gate	
23	1	DAC_0	O_AN		VTR	VTR	Reserved	
23	2	Reserved	Reserved		Reserved	Reserved		
23	3	Reserved	Reserved		Reserved	Reserved		
23	Strap							
24	Default: 0	GPIO161	PIO	I-4	VTR	VTR/VCC	No Gate	
24	1	DAC_1	O_AN		VTR	VTR	Reserved	
24	2	Reserved	Reserved		Reserved	Reserved		
24	3	Reserved	Reserved		Reserved	Reserved		
24	Strap							
25	Default: 0	GPIO165	PIO	I-4	VTR	VTR/VCC	No Gate	
25	1	CMP_VREF0	CMP_VREF		CMP_VREF	CMP_VREF	No Gate	
25	2	Reserved	Reserved		Reserved	Reserved		
25	3	Reserved	Reserved		Reserved	Reserved		
25	Strap							
26	Default: 0	GPIO166	PIO	I-4	VTR	VTR/VCC	No Gate	
26	1	CMP_VREF1	CMP_VREF		CMP_VREF	CMP_VREF	No Gate	
26	2	UART_CLK	PIO		VTR	VTR/VCC	Low	
26	3	Reserved	Reserved		Reserved	Reserved		
26	Strap							
27	Default: 0	GPIO123	PIO	I-4	VTR	VTR/VCC	No Gate	
27	1	SHD_CS#	PIO		VTR	VTR	Reserved	Note 10

MEC140x								
VTQFP Pin#	Mux	Signal Name	Buffer Type	Default Buffer Operation	Signal Power Well	Emulated Power Well	Gated State	Notes
27	2	Reserved	Reserved		Reserved	Reserved		
27	3	Reserved	Reserved		Reserved	Reserved		
27	Strap							
28	Default: 0	GPIO133	PIO	I-4	VTR	VTR/VCC	No Gate	
28	1	SHD_IO0	PIO		VTR	VTR	Low	Note 10
28	2	Reserved	Reserved		Reserved	Reserved		
28	3	Reserved	Reserved		Reserved	Reserved		
28	Strap							
29	Default: 0	GPIO134	PIO	I-4	VTR	VTR/VCC	No Gate	
29	1	SHD_IO1	PIO		VTR	VTR	Low	Note 10
29	2	Reserved	Reserved		Reserved	Reserved		
29	3	Reserved	Reserved		Reserved	Reserved		
29	Strap							
30	Default: 0	GPIO135	PIO	I-4	VTR	VTR/VCC	No Gate	
30	1	SHD_IO2	PIO		VTR	VTR	Low	Note 10
30	2	Reserved	Reserved		Reserved	Reserved		
30	3	Reserved	Reserved		Reserved	Reserved		
30	Strap							
31	Default: 0	GPIO136	PIO	I-4	VTR	VTR/VCC	No Gate	
31	1	SHD_IO3	PIO		VTR	VTR	Low	Note 10
31	2	Reserved	Reserved		Reserved	Reserved		
31	3	Reserved	Reserved		Reserved	Reserved		
31	Strap							
32	Default: 0	GPIO126	PIO	I-4	VTR	VTR/VCC	No Gate	
32	1	SHD_SCLK	PIO		VTR	VTR	Reserved	Note 10
32	2	Reserved	Reserved		Reserved	Reserved		
32	3	Reserved	Reserved		Reserved	Reserved		
32	Strap							
33	Default: 0	GPIO062	PIO	I-4	VTR	VTR/VCC	No Gate	
33	1	SPI_IO3	PIO		VTR	VTR	Low	
33	2	Reserved	Reserved		Reserved	Reserved		
33	3	Reserved	Reserved		Reserved	Reserved		
33	Strap							
34	Default: 0	GPIO030	PIO	I-4	VTR	VTR/VCC	No Gate	
34	1	BCM_INT0#	PIO		VTR	VTR	High	
34	2	PWM4	PIO		VTR	VTR	Reserved	
34	3	Reserved	Reserved		Reserved	Reserved		
34	Strap							
35	Default: 0	GPIO031	PIO	I-4	VTR	VTR/VCC	No Gate	
35	1	BCM_DAT0	PIO		VTR	VTR	Low	Note 7
35	2	PWM5	PIO		VTR	VTR	Reserved	
35	3	Reserved	Reserved		Reserved	Reserved		

MEC140x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
35	Strap							
36	Default: 0	GPIO032	PIO	I-4	VTR	VTR/VCC	No Gate	
36	1	BCM_CLK0	PIO		VTR	VTR	Reserved	
36	2	PWM6	PIO		VTR	VTR	Reserved	
36	3	Reserved	Reserved		Reserved	Reserved		
36	Strap							
37	Default: 0	GPIO045	PIO	I-4	VTR	VTR/VCC	No Gate	
37	1	BCM_INT1#	PIO		VTR	VTR	High	
37	2	KSO04	PIO		VTR	VTR	Reserved	Note 15
37	3	Reserved	Reserved		Reserved	Reserved		
37	Strap							
38	Default: 0	GPIO046	PIO	I-4	VTR	VTR/VCC	No Gate	
38	1	BCM_DAT1	PIO		VTR	VTR	Low	Note 7
38	2	KSO05	PIO		VTR	VTR	Reserved	Note 15
38	3	Reserved	Reserved		Reserved	Reserved		
38	Strap							
39	Default: 0	GPIO047	PIO	I-4	VTR	VTR/VCC	No Gate	
39	1	BCM_CLK1	PIO		VTR	VTR	Reserved	
39	2	KSO06	PIO		VTR	VTR	Reserved	Note 15
39	3	Reserved	Reserved		Reserved	Reserved		
39	Strap							
40	Default: 0	GPIO050	PIO	I-4	VTR	VTR/VCC	No Gate	
40	1	TACH0	PIO		VTR	VTR	Low	
40	2	Reserved	Reserved		Reserved	Reserved		
40	3	Reserved	Reserved		Reserved	Reserved		
40	Strap							
41	Default: 0	GPIO051	PIO	I-4	VTR	VTR/VCC	No Gate	
41	1	TACH1	PIO		VTR	VTR	Low	
41	2	Reserved	Reserved		Reserved	Reserved		
41	3	Reserved	Reserved		Reserved	Reserved		
41	Strap							
42	Default: 0	GPIO052	PIO	I-4	VTR	VTR/VCC	No Gate	
42	1	SPI_IO2	PIO		VTR	VTR	Low	
42	2	Reserved	Reserved		Reserved	Reserved		
42	3	Reserved	Reserved		Reserved	Reserved		
42	Strap							
43		VTR	PWR		PWR	PWR		
43								
43								
43								
43	Strap							
44	Default: 0	GPIO053	PIO	I-4	VTR	VTR/VCC	No Gate	

MEC140x								
VTQFP Pin#	Mux	Signal Name	Buffer Type	Default Buffer Operation	Signal Power Well	Emulated Power Well	Gated State	Notes
44	1	PWM0	PIO		VTR	VTR	Reserved	
44	2	Reserved	Reserved		Reserved	Reserved		
44	3	Reserved	Reserved		Reserved	Reserved		
44	Strap							
45	Default: 0	GPIO054	PIO	I-4	VTR	VTR/VCC	No Gate	
45	1	PWM1	PIO		VTR	VTR	Reserved	
45	2	Reserved	Reserved		Reserved	Reserved		
45	3	Reserved	Reserved		Reserved	Reserved		
45	Strap							
46	Default: 0	GPIO055	PIO	I-4	VTR	VTR/VCC	No Gate	
46	1	PWM2	PIO		VTR	VTR	Reserved	
46	2	KSO08	PIO		VTR	VTR	Reserved	Note 15
46	3	PVT_IO3	PIO		VTR	VTR	Low	Note 10
46	Strap							
47	Default: 0	GPIO056	PIO	I-4	VTR	VTR/VCC	No Gate	
47	1	PWM3	PIO		VTR	VTR	Reserved	
47	2	Reserved	Reserved		Reserved	Reserved		
47	3	Reserved	Reserved		Reserved	Reserved		
47	Strap							
48	Default: 0	GPIO057	PIO	I-4	VTR	VTR/VCC	No Gate	
48	1	VCC_PWRGD	PIO		VTR	VTR	High	
48	2	Reserved	Reserved		Reserved	Reserved		
48	3	Reserved	Reserved		Reserved	Reserved		
48	Strap							
49	Default: 0	GPIO060	PIO	I-4	VTR	VTR/VCC	No Gate	
49	1	KBRST	PIO		VTR	VCC	Reserved	
49	2	Reserved	Reserved		Reserved	Reserved		
49	3	Reserved	Reserved		Reserved	Reserved		
49	Strap							
50	Default: 0	GPIO025	PIO	I-4	VTR	VTR/VCC	No Gate	
50	1	KSO07	PIO		VTR	VTR	Reserved	Note 15
50	2	PVT_IO2	PIO		VTR	VTR	Low	Note 10
50	3	Reserved	Reserved		Reserved	Reserved		
50	Strap							
51		VSS	PWR		PWR	PWR		
51								
51								
51								
51	Strap							
52	Default: 0	GPIO026	PIO	I-4	VTR	VTR/VCC	No Gate	
52	1	PS2_CLK1B	PIO		VTR	VTR/VCC	Low	
52	2	Reserved	Reserved		Reserved	Reserved		

MEC140x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
52	3	Reserved	Reserved		Reserved	Reserved		
52	Strap							
53	Default: 0	GPIO061	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
53	1	LPCPD#	PCI_IO		VTR	VCC	High	
53	2	Reserved	Reserved		Reserved	Reserved		
53	3	Reserved	Reserved		Reserved	Reserved		
53	Strap							
54		VTR_33_18	PWR		PWR	VTR		
54								
54								
54								
54	Strap							
55	Default: 0	GPIO063	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
55	1	SER_IRQ	PCI_IO		VTR	VCC	High	Note 1
55	2	Reserved	Reserved		Reserved	Reserved		
55	3	Reserved	Reserved		Reserved	Reserved		
55	Strap							
56	Default: 0	GPIO064	PCI_PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
56	1	LRESET#	PCI_IO		VTR	VCC	Low	
56	2	Reserved	Reserved		Reserved	Reserved		
56	3	Reserved	Reserved		Reserved	Reserved		
56	Strap							
57	Default: 0	GPIO034	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
57	1	PCI_CLK	PCI_IO		VTR	VCC	Low	
57	2	Reserved	Reserved		Reserved	Reserved		
57	3	Reserved	Reserved		Reserved	Reserved		
57	Strap							
58	Default: 0	GPIO044	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
58	1	LFRAME#	PCI_IO		VTR	VCC	High	
58	2	Reserved	Reserved		Reserved	Reserved		
58	3	Reserved	Reserved		Reserved	Reserved		
58	Strap							
59	Default: 0	GPIO040	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
59	1	LAD0	PCI_IO		VTR	VCC	High	Note 1
59	2	Reserved	Reserved		Reserved	Reserved		
59	3	Reserved	Reserved		Reserved	Reserved		
59	Strap							
60	Default: 0	GPIO041	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
60	1	LAD1	PCI_IO		VTR	VCC	High	Note 1
60	2	Reserved	Reserved		Reserved	Reserved		
60	3	Reserved	Reserved		Reserved	Reserved		
60	Strap							

MEC140x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
61	Default: 0	GPIO042	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
61	1	LAD2	PCI_IO		VTR	VCC	High	Note 1
61	2	Reserved	Reserved		Reserved	Reserved		
61	3	Reserved	Reserved		Reserved	Reserved		
61	Strap							
62	Default: 0	GPIO043	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
62	1	LAD3	PCI_IO		VTR	VCC	High	Note 1
62	2	Reserved	Reserved		Reserved	Reserved		
62	3	Reserved	Reserved		Reserved	Reserved		
62	Strap							
63	Default: 0	GPIO067	PCI_PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
63	1	CLKRUN#	PCI_IO		VTR	VCC	Low	
63	2	Reserved	Reserved		Reserved	Reserved		
63	3	Reserved	Reserved		Reserved	Reserved		
63	Strap							
64		VSS	PWR		PWR	PWR		
64								
64								
64								
64	Strap							
65		VTR	PWR		PWR	PWR		
65								
65								
65								
65	Strap							
66	Default: 0	GPIO100	PIO	I-4	VTR	VTR/VCC	No Gate	
66	1	nEC_SCI	PIO		VTR	VTR	Reserved	
66	2	Reserved	Reserved		Reserved	Reserved		
66	3	Reserved	Reserved		Reserved	Reserved		
66	Strap							
67	Default: 0	GPIO101	PIO	I-4	VTR	VTR/VCC	No Gate	
67	1	SPI_CLK	PIO		VTR	VTR	Reserved	
67	2	Reserved	Reserved		Reserved	Reserved		
67	3	Reserved	Reserved		Reserved	Reserved		
67	Strap							
68	Default: 0	GPIO102	PIO	I-4	VTR	VTR/VCC	No Gate	
68	1	KSO09	PIO		VTR	VTR	Reserved	Note 15
68	2	Reserved	Reserved		Reserved	Reserved		
68	3	Reserved	Reserved		Reserved	Reserved		
68	Strap	CR_STRAP						
69	Default: 0	GPIO103	PIO	I-4	VTR	VTR/VCC	No Gate	
69	1	SPI_IO0	PIO		VTR	VTR	Low	

MEC140x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
69	2	Reserved	Reserved		Reserved	Reserved		
69	3	Reserved	Reserved		Reserved	Reserved		
69	Strap							
70	Default: 0	GPIO104	PIO	I-4	VTR	VTR/VCC	No Gate	
70	1	LED2	PIO		VTR	VTR	Reserved	
70	2	Reserved	Reserved		Reserved	Reserved		
70	3	Reserved	Reserved		Reserved	Reserved		
70	Strap							
71	Default: 0	GPIO105	PIO	I-4	VTR	VTR/VCC	No Gate	
71	1	SPI_IO1	PIO		VTR	VTR	Low	
71	2	Reserved	Reserved		Reserved	Reserved		
71	3	Reserved	Reserved		Reserved	Reserved		
71	Strap							
72	Default: 0	GPIO106	PIO	I-4	VTR	VTR/VCC	No Gate	
72	1	KSO10	PIO		VTR	VTR	Reserved	Note 15
72	2	Reserved	Reserved		Reserved	Reserved		
72	3	Reserved	Reserved		Reserved	Reserved		
72	Strap							
73	Default: 0	GPIO107	PIO	I-4	VTR	VTR/VCC	No Gate	Note 5
73	1	nRESET_OUT	PIO		VTR	VTR	Reserved	Note 5
73	2	Reserved	Reserved		Reserved	Reserved		
73	3	Reserved	Reserved		Reserved	Reserved		
73	Strap							
74	Default: 0	GPIO110	PIO	I-4	VTR	VTR/VCC	No Gate	
74	1	KSO11	PIO		VTR	VTR	Reserved	Note 15
74	2	Reserved	Reserved		Reserved	Reserved		
74	3	Reserved	Reserved		Reserved	Reserved		
74	Strap							
75	Default: 0	GPIO111	PIO	I-4	VTR	VTR/VCC	No Gate	
75	1	KSO12	PIO		VTR	VTR	Reserved	Note 15
75	2	Reserved	Reserved		Reserved	Reserved		
75	3	Reserved	Reserved		Reserved	Reserved		
75	Strap							
76	Default: 0	GPIO112	PIO	I-4	VTR	VTR/VCC	No Gate	
76	1	PS2_CLK1A	PIO		VTR	VTR/VCC	Low	
76	2	KSO13	PIO		VTR	VTR	Reserved	Note 15
76	3	Reserved	Reserved		Reserved	Reserved		
76	Strap							
77	Default: 0	GPIO113	PIO	I-4	VTR	VTR/VCC	No Gate	
77	1	PS2_DAT1A	PIO		VTR	VTR/VCC	Low	
77	2	KSO14	PIO		VTR	VTR	Reserved	Note 15
77	3	Reserved	Reserved		Reserved	Reserved		

MEC140x								
VTQFP Pin#	Mux	Signal Name	Buffer Type	Default Buffer Operation	Signal Power Well	Emulated Power Well	Gated State	Notes
77	Strap							
78	Default: 0	GPIO114	PIO	I-4	VTR	VTR/VCC	No Gate	
78	1	PS2_CLK0	PIO		VTR	VTR/VCC	Low	
78	2	Reserved	Reserved		Reserved	Reserved		
78	3	Reserved	Reserved		Reserved	Reserved		
78	Strap							
79	Default: 0	GPIO115	PIO	I-4	VTR	VTR/VCC	No Gate	
79	1	PS2_DAT0	PIO		VTR	VTR/VCC	Low	
79	2	Reserved	Reserved		Reserved	Reserved		
79	3	Reserved	Reserved		Reserved	Reserved		
79	Strap							
80	Default: 0	GPIO116	PIO	I-4	VTR	VTR/VCC	No Gate	
80	1	TFDP_DATA	PIO		VTR	VTR	Reserved	
80	2	UART_RX	PIO		VTR	VTR	Low	
80	3	Reserved	Reserved		Reserved	Reserved		
80	Strap							
81	Default: 0	GPIO117	PIO	I-4	VTR	VTR/VCC	No Gate	
81	1	TFDP_CLK	PIO		VTR	VTR	Reserved	
81	2	UART_TX	PIO		VTR	VTR	Reserved	
81	3	Reserved	Reserved		Reserved	Reserved		
81	Strap							
82		VTR	PWR		PWR	PWR		
82								
82								
82								
82	Strap							
83	Default: 0	GPIO120	PIO	I-4	VTR	VTR/VCC	No Gate	
83	1	CMP_VOUT1	PIO		VTR	VTR	Reserved	
83	2	Reserved	Reserved		Reserved	Reserved		
83	3	Reserved	Reserved		Reserved	Reserved		
83	Strap							
84		VSS	PWR		PWR	PWR		
84								
84								
84								
84	Strap							
85	Default: 0	GPIO124	PIO	I-4	VTR	VTR/VCC	No Gate	
85	1	CMP_VOUT0	PIO		VTR	VTR	Reserved	
85	2	Reserved	Reserved		Reserved	Reserved		
85	3	Reserved	Reserved		Reserved	Reserved		
85	Strap							
86	Default: 0	GPIO125	PIO	I-4	VTR	VTR/VCC	No Gate	

MEC140x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
86	1	KSO15	PIO		VTR	VTR	Reserved	Note 15
86	2	Reserved	Reserved		Reserved	Reserved		
86	3	Reserved	Reserved		Reserved	Reserved		
86	Strap							
87	Default: 0	ICSP_MCLR	I	I	VTR	VTR	No Gate	Note 2
87	1	Reserved	Reserved		Reserved	Reserved		
87	2	Reserved	Reserved		Reserved	Reserved		
87	3	Reserved	Reserved		Reserved	Reserved		
87	Strap							
88	Default: 0	GPIO127	PIO	I-4	VTR	VTR/VCC	No Gate	
88	1	PS2_DAT1B	PIO		VTR	VTR/VCC	Low	
88	2	Reserved	Reserved		Reserved	Reserved		
88	3	Reserved	Reserved		Reserved	Reserved		
88	Strap							
89	Default: 0	GPIO130	PIO	I-4	VTR	VTR/VCC	No Gate	
89	1	SMB03_DATA	PIO		VTR	VTR	High	Note 4
89	2	SMB03_DATA18	PIO		VTR	VTR	High	Note 11
89	3	Reserved	Reserved		Reserved	Reserved		
89	Strap							
90	Default: 0	GPIO035	PIO	I-4	VTR	VTR/VCC	No Gate	
90	1	Reserved	Reserved		Reserved	Reserved		
90	2	SB-TSI_CLK	SB-TSI		SB-TSI	SB-TSI	High	
90	3	Reserved	Reserved		Reserved	Reserved		
90	Strap							
91	Default: 0	GPIO131	PIO	I-4	VTR	VTR/VCC	No Gate	
91	1	SMB03_CLK	PIO		VTR	VTR	High	Note 4
91	2	SMB03_CLK18	PIO		VTR	VTR	High	Note 11
91	3	Reserved	Reserved		Reserved	Reserved		
91	Strap							
92	Default: 0	GPIO132	PIO	I-4	VTR	VTR/VCC	No Gate	
92	1	KSO16	PIO		VTR	VTR	Reserved	Note 15
92	2	Reserved	Reserved		Reserved	Reserved		
92	3	Reserved	Reserved		Reserved	Reserved		
92	Strap							
93	Default: 0	GPIO140	PIO	I-4	VTR	VTR/VCC	No Gate	
93	1	KSO17	PIO		VTR	VTR	Reserved	Note 15
93	2	Reserved	Reserved		Reserved	Reserved		
93	3	Reserved	Reserved		Reserved	Reserved		
93	Strap							
94	Default: 0	GPIO033	PIO	I-4	VTR	VTR/VCC	No Gate	
94	1	PECI_DAT	PECI_IO		PECI_IO	PECI_IO	Low	Note 12
94	2	SB_TSI_DAT	SB-TSI		SB-TSI	SB-TSI	Low	Note 12

MEC140x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
94	3	Reserved	Reserved		Reserved	Reserved		
94	Strap							
95		VREF_CPU	VREF_CPU		VREF_CPU	VREF_CPU		Note 6, Note 18
95								
95								
95								
95	Strap							
96	Default: 0	GPIO141	PIO	I-4	VTR	VTR/VCC	No Gate	
96	1	SMB04_DATA	PIO		VTR	VTR	High	Note 4
96	2	SMB04_DATA18	PIO		VTR	VTR	High	Note 11
96	3	Reserved	Reserved		Reserved	Reserved		
96	Strap							
97	Default: 0	GPIO142	PIO	I-4	VTR	VTR/VCC	No Gate	
97	1	SMB04_CLK	PIO		VTR	VTR	High	Note 4
97	2	SMB04_CLK18	PIO		VTR	VTR	High	Note 11
97	3	Reserved	Reserved		Reserved	Reserved		
97	Strap							
98	Default: 0	GPIO143	PIO	I-4	VTR	VTR/VCC	No Gate	Note 9
98	1	KSI0	PIO		VTR	VTR	Low	Note 15
98	2	DTR#	PIO		VTR	VTR	Reserved	
98	3	Reserved	Reserved		Reserved	Reserved		
98	Strap							
99	Default: 0	GPIO144	PIO	I-4	VTR	VTR/VCC	No Gate	
99	1	KSI1	PIO		VTR	VTR	Low	Note 15
99	2	DCD#	PIO		VTR	VTR	High	
99	3	Reserved	Reserved		Reserved	Reserved		
99	Strap							
100		VSS	PWR		PWR	PWR		
100								
100								
100								
100	Strap							
101	Default: 0	GPIO145 (ICSP_CLOCK)	PIO	I-4	VTR	VTR/VCC	No Gate	Note 2
101	1	Reserved	PIO		Reserved	Reserved		
101	2	Reserved	Reserved		Reserved	Reserved		
101	3	Reserved	Reserved		Reserved	Reserved		
101	Strap							
102	Default: 0	GPIO146 (ICSP_DATA)	PIO	I-4	VTR	VTR/VCC	No Gate	Note 2
102	1	Reserved	PIO		Reserved	Reserved		
102	2	Reserved	Reserved		Reserved	Reserved		

MEC140x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
102	3	Reserved	Reserved		Reserved	Reserved		
102	Strap							
103		VTR	PWR		PWR	PWR		
103								
103								
103								
103	Strap							
104	Default: 0	GPIO147	PIO	I-4	VTR	VTR/VCC	No Gate	
104	1	KSI4	PIO		VTR	VTR	Low	Note 15
104	2	DSR#	PIO		VTR	VTR	High	
104	3	Reserved	Reserved		Reserved	Reserved		
104	Strap							
105	Default: 0	GPIO150	PIO	I-4	VTR	VTR/VCC	No Gate	
105	1	KSI5	PIO		VTR	VTR	Low	Note 15
105	2	RI#	PIO		VTR	VTR	High	
105	3	Reserved	Reserved		Reserved	Reserved		
105	Strap							
106	Default: 0	GPIO156	PIO	I-4	VTR	VTR/VCC	No Gate	
106	1	LED1	PIO		VTR	VTR	Reserved	
106	2	Reserved	Reserved		Reserved	Reserved		
106	3	Reserved	Reserved		Reserved	Reserved		
106	Strap							
107	Default: 0	GPIO151	PIO	I-4	VTR	VTR/VCC	No Gate	
107	1	KSI6	PIO		VTR	VTR	Low	Note 15
107	2	RTS#	PIO		VTR	VTR	Reserved	
107	3	Reserved	Reserved		Reserved	Reserved		
107	Strap							
108	Default: 0	GPIO152	PIO	I-4	VTR	VTR/VCC	No Gate	
108	1	KSI7	PIO		VTR	VTR	Low	Note 15
108	2	CTS#	PIO		VTR	VTR	High	
108	3	Reserved	Reserved		Reserved	Reserved		
108	Strap							
109	Default: 0	GPIO153	PIO	I-2	VTR	VTR/VCC	No Gate	
109	1	ADC4	I_AN		I_AN	I_AN	No Gate	Note 8
109	2	Reserved	Reserved		Reserved	Reserved		
109	3	Reserved	Reserved		Reserved	Reserved		
109	Strap							
110	Default: 0	GPIO154	PIO	I-2	VTR	VTR/VCC	No Gate	
110	1	ADC3	I_AN		I_AN	I_AN	No Gate	Note 8
110	2	Reserved	Reserved		Reserved	Reserved		
110	3	Reserved	Reserved		Reserved	Reserved		
110	Strap							

MEC140x								
VTQFP Pin#	Mux	Signal Name	Buffer Type	Default Buffer Operation	Signal Power Well	Emulated Power Well	Gated State	Notes
111	Default: 0	GPIO155	PIO	I-2	VTR	VTR/VCC	No Gate	
111	1	ADC2	I_AN		I_AN	I_AN	No Gate	Note 8
111	2	Reserved	Reserved		Reserved	Reserved		
111	3	Reserved	Reserved		Reserved	Reserved		
111	Strap							
112		AVSS	PWR		PWR	PWR		
112								
112								
112								
112	Strap							
113	Default: 0	GPIO122	PIO	I-2	VTR	VTR/VCC	No Gate	
113	1	ADC1	I_AN		I_AN	I_AN	No Gate	Note 8
113	2	Reserved	Reserved		Reserved	Reserved		
113	3	Reserved	Reserved		Reserved	Reserved		
113	Strap							
114	Default: 0	GPIO121	PIO	I-2	VTR	VTR/VCC	No Gate	
114	1	ADC0	I_AN		I_AN	I_AN	No Gate	Note 8
114	2	Reserved	Reserved		Reserved	Reserved		
114	3	Reserved	Reserved		Reserved	Reserved		
114	Strap							
115	0	Reserved	Reserved		Reserved	Reserved		
115	Default: 1	ADC_VREF	ADC_VREF		ADC_VREF	ADC_VREF	No Gate	Note 18
115	2	Reserved	Reserved		Reserved	Reserved		
115	3	Reserved	Reserved		Reserved	Reserved		
115	Strap							
116	Default: 0	GPIO022	PIO	I-2	VTR	VTR/VCC	No Gate	
116	1	ADC5	I_AN		I_AN	I_AN	No Gate	Note 8
116	2	Reserved	Reserved		Reserved	Reserved		
116	3	Reserved	Reserved		Reserved	Reserved		
116	Strap							
117	Default: 0	GPIO023	PIO	I-2	VTR	VTR/VCC	No Gate	
117	1	ADC6	I_AN		I_AN	I_AN	No Gate	Note 8
117	2	A20M	PIO		VTR	VCC	Reserved	
117	3	Reserved	Reserved		Reserved	Reserved		
117	Strap							
118	Default: 0	GPIO024	PIO	I-2	VTR	VTR/VCC	No Gate	
118	1	ADC7	I_AN		I_AN	I_AN	No Gate	Note 8
118	2	Reserved	Reserved		Reserved	Reserved		
118	3	Reserved	Reserved		Reserved	Reserved		
118	Strap							
119	0	GPIO004	PIO		VBAT	VTR/VCC	No Gate	
119	Default: 1	BGPO	PIO	O-4 mA	VBAT	VBAT	Reserved	

MEC140x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
119	2	Reserved	Reserved		Reserved	Reserved		
119	3	Reserved	Reserved		Reserved	Reserved		
119	Strap							
120	0	GPIO003	PIO		VBAT	VTR/VCC	No Gate	
120	Default: 1	SYSPWR_PRES	ILLK	ILLK-4	VBAT	VBAT	Low	Note 12
120	2	Reserved	Reserved		Reserved	Reserved		
120	3	Reserved	Reserved		Reserved	Reserved		
120	Strap							
121	0	GPIO036	PIO		VBAT	VTR/VCC	No Gate	
121	Default: 1	VCI_OUT	PIO	O-8 mA	VBAT	VBAT	Reserved	
121	2	Reserved	Reserved		Reserved	Reserved		
121	3	Reserved	Reserved		Reserved	Reserved		
121	Strap							
122		VBAT	PWR		PWR	PWR		
122								
122								
122								
122	Strap							
123	Default: 0	XTAL1	OCLK	OCLK	VBAT	VBAT	No Gate	
123	1	Reserved	Reserved		Reserved	Reserved		
123	2	Reserved	Reserved		Reserved	Reserved		
123	3	Reserved	Reserved		Reserved	Reserved		
123	Strap							
124		VSS_VBAT	PWR		PWR	PWR		
124								
124								
124								
124	Strap							
125	Default: 0	XTAL2	ICLK	ICLK	VBAT	VBAT	No Gate	
125	1	Reserved	Reserved		Reserved	Reserved		
125	2	Reserved	Reserved		Reserved	Reserved		
125	3	Reserved	Reserved		Reserved	Reserved		
125	Strap							
126	0	GPIO162	PIO		VBAT	VTR/VCC	No Gate	
126	Default: 1	VCI_IN1#	ILLK	ILLK-4	VBAT	VBAT	High	Note 14
126	2	Reserved	Reserved		Reserved	Reserved		
126	3	Reserved	Reserved		Reserved	Reserved		
126	Strap							
127	0	GPIO163	PIO		VBAT	VTR/VCC	No Gate	
127	Default: 1	VCI_IN0#	ILLK	ILLK-4	VBAT	VBAT	High	Note 14
127	2	Reserved	Reserved		Reserved	Reserved		
127	3	Reserved	Reserved		Reserved	Reserved		

MEC140x								
VTQFP Pin#	Mux	Signal Name	Buffer Type	Default Buffer Operation	Signal Power Well	Emulated Power Well	Gated State	Notes
127	Strap							
128	0	GPIO164	PIO		VBAT	VTR/VCC	No Gate	
128	Default: 1	VCI_OVRD_IN	ILLK	ILLK-4	VBAT	VBAT	Low	Note 14
128	2	Reserved	Reserved		Reserved	Reserved		
128	3	Reserved	Reserved		Reserved	Reserved		
128	Strap							

TABLE 2-3: MEC141X PIN MULTIPLEXING

MEC141x								
VTQFP Pin#	Mux	Signal Name	Buffer Type	Default Buffer Operation	Signal Power Well	Emulated Power Well	Gated State	Notes
1	Default: 0	GPIO157	PIO	I-4	VTR	VTR/VCC	No Gate	
1	1	LED0	PIO		VTR	VTR	Reserved	
1	2	TST_CLK_OUT	PIO		VTR	VTR	Reserved	Note 13
1	3	Reserved	Reserved		Reserved	Reserved		
1	Strap							
2	Default: 0	GPIO027	PIO	I-4	VTR	VTR/VCC	No Gate	
2	1	KSO00	PIO		VTR	VTR	Reserved	Note 15
2	2	PVT_IO1	PIO		VTR	VTR	Low	Note 10
2	3	Reserved	Reserved		Reserved	Reserved		
2	Strap							
3	Default: 0	GPIO001	PIO	I-4	VTR	VTR/VCC	No Gate	
3	1	SPI_CS#	PIO		VTR	VTR	Reserved	
3	2	32KHZ_OUT	PIO		VTR	VTR	Reserved	
3	3	Reserved	Reserved		Reserved	Reserved		
3	Strap							
4	Default: 0	GPIO002	PIO	I-4	VTR	VTR/VCC	No Gate	
4	1	PWM7	PIO		VTR	VTR	Reserved	
4	2	Reserved	Reserved		Reserved	Reserved		
4	3	Reserved	Reserved		Reserved	Reserved		
4	Strap							
5		VTR	PWR		PWR	PWR		
5								
5								
5								
5	Strap							
6	Default: 0	GPIO005	PIO	I-4	VTR	VTR/VCC	No Gate	
6	1	SMB00_DATA	PIO		VTR	VTR	High	Note 4

MEC141x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
6	2	SMB00_DATA18	PIO		VTR	VTR	High	Note 11
6	3	KSI2	PIO		VTR	VTR	Low	Note 15
6	Strap							
7	Default: 0	GPIO006	PIO	I-4	VTR	VTR/VCC	No Gate	
7	1	SMB00_CLK	PIO		VTR	VTR	High	Note 4
7	2	SMB00_CLK18	PIO		VTR	VTR	High	Note 11
7	3	KSI3	PIO		VTR	VTR	Low	Note 15
7	Strap							
8	Default: 0	GPIO007	PIO	I-4	VTR	VTR/VCC	No Gate	
8	1	SMB01_DATA	PIO		VTR	VTR	High	Note 4
8	2	SMB01_DATA18	PIO		VTR	VTR	High	Note 11
8	3	Reserved	Reserved		Reserved	Reserved		
8	Strap							
9	Default: 0	GPIO010	PIO	I-4	VTR	VTR/VCC	No Gate	
9	1	SMB01_CLK	PIO		VTR	VTR	High	Note 4
9	2	SMB01_CLK18	PIO		VTR	VTR	High	Note 11
9	3	Reserved	Reserved		Reserved	Reserved		
9	Strap							
10	Default: 0	GPIO011	PIO	I-4	VTR	VTR/VCC	No Gate	
10	1	nSMI	PIO		VTR	VTR	Reserved	
10	2	nEMI_INT	PIO		VTR	VTR	Reserved	
10	3	Reserved	Reserved		Reserved	Reserved		
10	Strap							
11	Default: 0	GPIO012	PIO	I-4	VTR	VTR/VCC	No Gate	
11	1	SMB02_DATA	PIO		VTR	VTR	High	Note 4
11	2	SMB02_DATA18	PIO		VTR	VTR	High	Note 11
11	3	Reserved	Reserved		Reserved	Reserved		
11	Strap							
12	Default: 0	GPIO013	PIO	I-4	VTR	VTR/VCC	No Gate	
12	1	SMB02_CLK	PIO		VTR	VTR	High	Note 4
12	2	SMB02_CLK18	PIO		VTR	VTR	High	Note 11
12	3	Reserved	Reserved		Reserved	Reserved		
12	Strap							
13	0	GPIO014	PIO		VTR	VTR/VCC	No Gate	
13	Default: 1	nRESET_IN	PIO	I-4	VTR	VTR	High	
13	2	Reserved	Reserved		Reserved	Reserved		
13	3	Reserved	Reserved		Reserved	Reserved		
13	Strap							
14	Default: 0	GPIO015	PIO	I-4	VTR	VTR/VCC	No Gate	
14	1	KSO01	PIO		VTR	VTR	Reserved	Note 15
14	2	PVT_CS#	PIO		VTR	VTR	Reserved	Note 10
14	3	Reserved	Reserved		Reserved	Reserved		

MEC141x								
VTQFP Pin#	Mux	Signal Name	Buffer Type	Default Buffer Operation	Signal Power Well	Emulated Power Well	Gated State	Notes
14	Strap							
15	Default: 0	GPIO016	PIO	I-4	VTR	VTR/VCC	No Gate	
15	1	KSO02	PIO		VTR	VTR	Reserved	Note 15
15	2	PVT_SCLK	PIO		VTR	VTR	Reserved	Note 10
15	3	Reserved	Reserved		Reserved	Reserved		
15	Strap							
16	Default: 0	GPIO017	PIO	I-4	VTR	VTR/VCC	No Gate	
16	1	KSO03	PIO		VTR	VTR	Reserved	Note 15
16	2	PVT_IO0	PIO		VTR	VTR	Low	Note 10
16	3	Reserved	Reserved		Reserved	Reserved		
16	Strap							
17		VSS	PWR		PWR	PWR		
17								
17								
17								
17	Strap							
18		VR_CAP	PWR		PWR	PWR		Note 3
18								
18								
18								
18	Strap							
19		VTR	PWR		PWR	PWR		
19								
19								
19								
19	Strap							
20	Default: 0	GPIO020	PIO	I-4	VTR	VTR/VCC	No Gate	
20	1	CMP_VIN0	I_AN		I_AN	I_AN	No Gate	
20	2	Reserved	Reserved		Reserved	Reserved		
20	3	Reserved	Reserved		Reserved	Reserved		
20	Strap							
21	Default: 0	GPIO021	PIO	I-4	PWR	VTR/VCC	No Gate	
21	1	CMP_VIN1	I_AN		I_AN	I_AN	No Gate	
21	2	Reserved	Reserved		Reserved	Reserved		
21	3	Reserved	Reserved		Reserved	Reserved		
21	Strap							
22	0	Reserved	Reserved		Reserved	Reserved		
22	Default: 1	DAC_VREF	DAC_VREF		DAC_VREF	DAC_VREF	No Gate	Note 18
22	2	Reserved	Reserved		Reserved	Reserved		
22	3	Reserved	Reserved		Reserved	Reserved		
22	Strap							
23	Default: 0	GPIO160	PIO	I-4	VTR	VTR/VCC	No Gate	

MEC141x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
23	1	DAC_0	O_AN		VTR	VTR	Reserved	
23	2	Reserved	Reserved		Reserved	Reserved		
23	3	Reserved	Reserved		Reserved	Reserved		
23	Strap							
24	Default: 0	GPIO161	PIO	I-4	VTR	VTR/VCC	No Gate	
24	1	DAC_1	O_AN		VTR	VTR	Reserved	
24	2	Reserved	Reserved		Reserved	Reserved		
24	3	Reserved	Reserved		Reserved	Reserved		
24	Strap							
25	Default: 0	GPIO165	PIO	I-4	VTR	VTR/VCC	No Gate	
25	1	CMP_VREF0	CMP_VREF		CMP_VREF	CMP_VREF	No Gate	
25	2	Reserved	Reserved		Reserved	Reserved		
25	3	Reserved	Reserved		Reserved	Reserved		
25	Strap							
26	Default: 0	GPIO166	PIO	I-4	VTR	VTR/VCC	No Gate	
26	1	CMP_VREF1	CMP_VREF		CMP_VREF	CMP_VREF	No Gate	
26	2	UART_CLK	PIO		VTR	VTR/VCC	Low	
26	3	Reserved	Reserved		Reserved	Reserved		
26	Strap							
27	Default: 0	GPIO123	PIO	I-4	VTR	VTR/VCC	No Gate	Note 16
27	1	SHD_CS#	PIO		VTR	VTR	Reserved	Note 10
27	2	Reserved	Reserved		Reserved	Reserved		
27	3	Reserved	Reserved		Reserved	Reserved		
27	Strap	BSS_STRAP						
28	Default: 0	GPIO133	PIO	I-4	VTR	VTR/VCC	No Gate	
28	1	SHD_IO0	PIO		VTR	VTR	Low	Note 10
28	2	Reserved	Reserved		Reserved	Reserved		
28	3	Reserved	Reserved		Reserved	Reserved		
28	Strap							
29	Default: 0	GPIO134	PIO	I-4	VTR	VTR/VCC	No Gate	
29	1	SHD_IO1	PIO		VTR	VTR	Low	Note 10
29	2	Reserved	Reserved		Reserved	Reserved		
29	3	Reserved	Reserved		Reserved	Reserved		
29	Strap							
30	Default: 0	GPIO135	PIO	I-4	VTR	VTR/VCC	No Gate	Note 17
30	1	SHD_IO2	PIO		VTR	VTR	Low	Note 10
30	2	Reserved	Reserved		Reserved	Reserved		
30	3	Reserved	Reserved		Reserved	Reserved		
30	Strap							
31	Default: 0	GPIO136	PIO	I-4	VTR	VTR/VCC	No Gate	
31	1	SHD_IO3	PIO		VTR	VTR	Low	Note 10
31	2	Reserved	Reserved		Reserved	Reserved		

MEC141x								
VTQFP Pin#	Mux	Signal Name	Buffer Type	Default Buffer Operation	Signal Power Well	Emulated Power Well	Gated State	Notes
31	3	Reserved	Reserved		Reserved	Reserved		
31	Strap							
32	Default: 0	GPIO126	PIO	I-4	VTR	VTR/VCC	No Gate	
32	1	SHD_SCLK	PIO		VTR	VTR	Reserved	Note 10
32	2	Reserved	Reserved		Reserved	Reserved		
32	3	Reserved	Reserved		Reserved	Reserved		
32	Strap							
33	Default: 0	GPIO062	PIO	I-4	VTR	VTR/VCC	No Gate	
33	1	SPI_IO3	PIO		VTR	VTR	Low	
33	2	Reserved	Reserved		Reserved	Reserved		
33	3	Reserved	Reserved		Reserved	Reserved		
33	Strap							
34	Default: 0	GPIO030	PIO	I-4	VTR	VTR/VCC	No Gate	
34	1	BCM_INT0#	PIO		VTR	VTR	High	
34	2	PWM4	PIO		VTR	VTR	Reserved	
34	3	Reserved	Reserved		Reserved	Reserved		
34	Strap							
35	Default: 0	GPIO031	PIO	I-4	VTR	VTR/VCC	No Gate	
35	1	BCM_DAT0	PIO		VTR	VTR	Low	Note 7
35	2	PWM5	PIO		VTR	VTR	Reserved	
35	3	Reserved	Reserved		Reserved	Reserved		
35	Strap							
36	Default: 0	GPIO032	PIO	I-4	VTR	VTR/VCC	No Gate	
36	1	BCM_CLK0	PIO		VTR	VTR	Reserved	
36	2	PWM6	PIO		VTR	VTR	Reserved	
36	3	Reserved	Reserved		Reserved	Reserved		
36	Strap							
37	Default: 0	GPIO045	PIO	I-4	VTR	VTR/VCC	No Gate	
37	1	BCM_INT1#	PIO		VTR	VTR	High	
37	2	KSO04	PIO		VTR	VTR	Reserved	Note 15
37	3	Reserved	Reserved		Reserved	Reserved		
37	Strap							
38	Default: 0	GPIO046	PIO	I-4	VTR	VTR/VCC	No Gate	
38	1	BCM_DAT1	PIO		VTR	VTR	Low	Note 7
38	2	KSO05	PIO		VTR	VTR	Reserved	Note 15
38	3	Reserved	Reserved		Reserved	Reserved		
38	Strap							
39	Default: 0	GPIO047	PIO	I-4	VTR	VTR/VCC	No Gate	
39	1	BCM_CLK1	PIO		VTR	VTR	Reserved	
39	2	KSO06	PIO		VTR	VTR	Reserved	Note 15
39	3	Reserved	Reserved		Reserved	Reserved		
39	Strap							

MEC141x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
40	Default: 0	GPIO050	PIO	I-4	VTR	VTR/VCC	No Gate	
40	1	TACH0	PIO		VTR	VTR	Low	
40	2	Reserved	Reserved		Reserved	Reserved		
40	3	Reserved	Reserved		Reserved	Reserved		
40	Strap							
41	Default: 0	GPIO051	PIO	I-4	VTR	VTR/VCC	No Gate	
41	1	TACH1	PIO		VTR	VTR	Low	
41	2	Reserved	Reserved		Reserved	Reserved		
41	3	Reserved	Reserved		Reserved	Reserved		
41	Strap							
42	Default: 0	GPIO052	PIO	I-4	VTR	VTR/VCC	No Gate	
42	1	SPI_IO2	PIO		VTR	VTR	Low	
42	2	Reserved	Reserved		Reserved	Reserved		
42	3	Reserved	Reserved		Reserved	Reserved		
42	Strap							
43		VTR	PWR		PWR	PWR		
43								
43								
43								
43	Strap							
44	Default: 0	GPIO053	PIO	I-4	VTR	VTR/VCC	No Gate	
44	1	PWM0	PIO		VTR	VTR	Reserved	
44	2	Reserved	Reserved		Reserved	Reserved		
44	3	Reserved	Reserved		Reserved	Reserved		
44	Strap							
45	Default: 0	GPIO054	PIO	I-4	VTR	VTR/VCC	No Gate	
45	1	PWM1	PIO		VTR	VTR	Reserved	
45	2	Reserved	Reserved		Reserved	Reserved		
45	3	Reserved	Reserved		Reserved	Reserved		
45	Strap							
46	Default: 0	GPIO055	PIO	I-4	VTR	VTR/VCC	No Gate	
46	1	PWM2	PIO		VTR	VTR	Reserved	
46	2	KSO08	PIO		VTR	VTR	Reserved	Note 15
46	3	PVT_IO3	PIO		VTR	VTR	Low	Note 10
46	Strap							
47	Default: 0	GPIO056	PIO	I-4	VTR	VTR/VCC	No Gate	
47	1	PWM3	PIO		VTR	VTR	Reserved	
47	2	Reserved	Reserved		Reserved	Reserved		
47	3	Reserved	Reserved		Reserved	Reserved		
47	Strap							
48	Default: 0	GPIO057	PIO	I-4	VTR	VTR/VCC	No Gate	
48	1	VCC_PWRGD	PIO		VTR	VTR	High	

MEC141x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
48	2	Reserved	Reserved		Reserved	Reserved		
48	3	Reserved	Reserved		Reserved	Reserved		
48	Strap							
49	Default: 0	GPIO060	PIO	I-4	VTR	VTR/VCC	No Gate	
49	1	KBRST	PIO		VTR	VCC	Reserved	
49	2	Reserved	Reserved		Reserved	Reserved		
49	3	Reserved	Reserved		Reserved	Reserved		
49	Strap							
50	Default: 0	GPIO025	PIO	I-4	VTR	VTR/VCC	No Gate	
50	1	KSO07	PIO		VTR	VTR	Reserved	Note 15
50	2	PVT_IO2	PIO		VTR	VTR	Low	Note 10
50	3	Reserved	Reserved		Reserved	Reserved		
50	Strap							
51		VSS	PWR		PWR	PWR		
51								
51								
51								
51	Strap							
52	Default: 0	GPIO026	PIO	I-4	VTR	VTR/VCC	No Gate	
52	1	PS2_CLK1B	PIO		VTR	VTR/VCC	Low	
52	2	Reserved	Reserved		Reserved	Reserved		
52	3	Reserved	Reserved		Reserved	Reserved		
52	Strap							
53	Default: 0	GPIO061	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
53	1	LPCPD#	PCI_IO		VTR	VCC	High	
53	2	ESPI_RESET#	PIO		VTR_33_18	VTR	Low	
53	3	Reserved	Reserved		Reserved	Reserved		
53	Strap							
54		VTR_33_18	PWR		PWR	VTR		
54								
54								
54								
54	Strap							
55	Default: 0	GPIO063	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
55	1	SER_IRQ	PCI_IO		VTR	VCC	High	Note 1
55	2	ESPI_ALERT#	PIO		VTR_33_18	VTR	Reserved	
55	3	Reserved	Reserved		Reserved	Reserved		
55	Strap							
56	Default: 0	GPIO064	PCI_PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
56	1	LRESET#	PCI_IO		VTR	VCC	Low	
56	2	Reserved	Reserved		Reserved	Reserved		
56	3	Reserved	Reserved		Reserved	Reserved		

MEC141x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
56	Strap							
57	Default: 0	GPIO034	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
57	1	PCI_CLK	PCI_IO		VTR	VCC	Low	
57	2	ESPI_CLK	PIO		VTR_33_18	VTR	Low	
57	3	Reserved	Reserved		Reserved	Reserved		
57	Strap							
58	Default: 0	GPIO044	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
58	1	LFRAME#	PCI_IO		VTR	VCC	High	
58	2	ESPI_CS#	PIO		VTR_33_18	VTR	High	
58	3	Reserved	Reserved		Reserved	Reserved		
58	Strap							
59	Default: 0	GPIO040	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
59	1	LAD0	PCI_IO		VTR	VCC	High	Note 1
59	2	ESPI_IO0	PIO		VTR_33_18	VTR	Low	
59	3	Reserved	Reserved		Reserved	Reserved		
59	Strap							
60	Default: 0	GPIO041	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
60	1	LAD1	PCI_IO		VTR	VCC	High	Note 1
60	2	ESPI_IO1	PIO		VTR_33_18	VTR	Low	
60	3	Reserved	Reserved		Reserved	Reserved		
60	Strap							
61	Default: 0	GPIO042	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
61	1	LAD2	PCI_IO		VTR	VCC	High	Note 1
61	2	ESPI_IO2	PIO		VTR_33_18	VTR	Low	
61	3	Reserved	Reserved		Reserved	Reserved		
61	Strap							
62	Default: 0	GPIO043	PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
62	1	LAD3	PCI_IO		VTR	VCC	High	Note 1
62	2	ESPI_IO3	PIO		VTR_33_18	VTR	Low	
62	3	Reserved	Reserved		Reserved	Reserved		
62	Strap							
63	Default: 0	GPIO067	PCI_PIO	I-4	VTR_33_18	VTR/VCC	No Gate	
63	1	CLKRUN#	PCI_IO		VTR	VCC	Low	
63	2	Reserved	Reserved		Reserved	Reserved		
63	3	Reserved	Reserved		Reserved	Reserved		
63	Strap							
64		VSS	PWR		PWR	PWR		
64								
64								
64								
64	Strap							
65		VTR	PWR		PWR	PWR		

MEC141x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
65								
65								
65								
65	Strap							
66	Default: 0	GPIO100	PIO	I-4	VTR	VTR/VCC	No Gate	
66	1	nEC_SCI	PIO		VTR	VTR	Reserved	
66	2	Reserved	Reserved		Reserved	Reserved		
66	3	Reserved	Reserved		Reserved	Reserved		
66	Strap							
67	Default: 0	GPIO101	PIO	I-4	VTR	VTR/VCC	No Gate	
67	1	SPI_CLK	PIO		VTR	VTR	Reserved	
67	2	Reserved	Reserved		Reserved	Reserved		
67	3	Reserved	Reserved		Reserved	Reserved		
67	Strap							
68	Default: 0	GPIO102	PIO	I-4	VTR	VTR/VCC	No Gate	
68	1	KSO09	PIO		VTR	VTR	Reserved	Note 15
68	2	Reserved	Reserved		Reserved	Reserved		
68	3	Reserved	Reserved		Reserved	Reserved		
68	Strap	CR_STRAP						
69	Default: 0	GPIO103	PIO	I-4	VTR	VTR/VCC	No Gate	
69	1	SPI_IO0	PIO		VTR	VTR	Low	
69	2	Reserved	Reserved		Reserved	Reserved		
69	3	Reserved	Reserved		Reserved	Reserved		
69	Strap							
70	Default: 0	GPIO104	PIO	I-4	VTR	VTR/VCC	No Gate	
70	1	LED2	PIO		VTR	VTR	Reserved	
70	2	Reserved	Reserved		Reserved	Reserved		
70	3	Reserved	Reserved		Reserved	Reserved		
70	Strap							
71	Default: 0	GPIO105	PIO	I-4	VTR	VTR/VCC	No Gate	
71	1	SPI_IO1	PIO		VTR	VTR	Low	
71	2	Reserved	Reserved		Reserved	Reserved		
71	3	Reserved	Reserved		Reserved	Reserved		
71	Strap							
72	Default: 0	GPIO106	PIO	I-4	VTR	VTR/VCC	No Gate	
72	1	KSO10	PIO		VTR	VTR	Reserved	Note 15
72	2	Reserved	Reserved		Reserved	Reserved		
72	3	Reserved	Reserved		Reserved	Reserved		
72	Strap							

MEC141x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
73	Default: 0	GPIO107	PIO	I-4	VTR	VTR/VCC	No Gate	Note 5
73	1	nRESET_OUT	PIO		VTR	VTR	Reserved	Note 5
73	2	Reserved	Reserved		Reserved	Reserved		
73	3	Reserved	Reserved		Reserved	Reserved		
73	Strap							
74	Default: 0	GPIO110	PIO	I-4	VTR	VTR/VCC	No Gate	
74	1	KSO11	PIO		VTR	VTR	Reserved	Note 15
74	2	Reserved	Reserved		Reserved	Reserved		
74	3	Reserved	Reserved		Reserved	Reserved		
74	Strap							
75	Default: 0	GPIO111	PIO	I-4	VTR	VTR/VCC	No Gate	
75	1	KSO12	PIO		VTR	VTR	Reserved	Note 15
75	2	Reserved	Reserved		Reserved	Reserved		
75	3	Reserved	Reserved		Reserved	Reserved		
75	Strap							
76	Default: 0	GPIO112	PIO	I-4	VTR	VTR/VCC	No Gate	
76	1	PS2_CLK1A	PIO		VTR	VTR/VCC	Low	
76	2	KSO13	PIO		VTR	VTR	Reserved	Note 15
76	3	Reserved	Reserved		Reserved	Reserved		
76	Strap							
77	Default: 0	GPIO113	PIO	I-4	VTR	VTR/VCC	No Gate	
77	1	PS2_DAT1A	PIO		VTR	VTR/VCC	Low	
77	2	KSO14	PIO		VTR	VTR	Reserved	Note 15
77	3	Reserved	Reserved		Reserved	Reserved		
77	Strap							
78	Default: 0	GPIO114	PIO	I-4	VTR	VTR/VCC	No Gate	
78	1	PS2_CLK0	PIO		VTR	VTR/VCC	Low	
78	2	Reserved	Reserved		Reserved	Reserved		
78	3	Reserved	Reserved		Reserved	Reserved		
78	Strap							
79	Default: 0	GPIO115	PIO	I-4	VTR	VTR/VCC	No Gate	
79	1	PS2_DAT0	PIO		VTR	VTR/VCC	Low	
79	2	Reserved	Reserved		Reserved	Reserved		
79	3	Reserved	Reserved		Reserved	Reserved		
79	Strap							
80	Default: 0	GPIO116	PIO	I-4	VTR	VTR/VCC	No Gate	
80	1	TFDP_DATA	PIO		VTR	VTR	Reserved	
80	2	UART_RX	PIO		VTR	VTR	Low	
80	3	Reserved	Reserved		Reserved	Reserved		
80	Strap							
81	Default: 0	GPIO117	PIO	I-4	VTR	VTR/VCC	No Gate	
81	1	TFDP_CLK	PIO		VTR	VTR	Reserved	

MEC141x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
81	2	UART_TX	PIO		VTR	VTR	Reserved	
81	3	Reserved	Reserved		Reserved	Reserved		
81	Strap							
82		VTR	PWR		PWR	PWR		
82								
82								
82								
82	Strap							
83	Default: 0	GPIO120	PIO	I-4	VTR	VTR/VCC	No Gate	
83	1	CMP_VOUT1	PIO		VTR	VTR	Reserved	
83	2	Reserved	Reserved		Reserved	Reserved		
83	3	Reserved	Reserved		Reserved	Reserved		
83	Strap							
84		VSS	PWR		PWR	PWR		
84								
84								
84								
84	Strap							
85	Default: 0	GPIO124	PIO	I-4	VTR	VTR/VCC	No Gate	
85	1	CMP_VOUT0	PIO		VTR	VTR	Reserved	
85	2	Reserved	Reserved		Reserved	Reserved		
85	3	Reserved	Reserved		Reserved	Reserved		
85	Strap							
86	Default: 0	GPIO125	PIO	I-4	VTR	VTR/VCC	No Gate	
86	1	KSO15	PIO		VTR	VTR	Reserved	Note 15
86	2	Reserved	Reserved		Reserved	Reserved		
86	3	Reserved	Reserved		Reserved	Reserved		
86	Strap							
87	Default: 0	ICSP_MCLR	I	I	VTR	VTR	No Gate	Note 2
87	1	Reserved	Reserved		Reserved	Reserved		
87	2	Reserved	Reserved		Reserved	Reserved		
87	3	Reserved	Reserved		Reserved	Reserved		
87	Strap							
88	Default: 0	GPIO127	PIO	I-4	VTR	VTR/VCC	No Gate	
88	1	PS2_DAT1B	PIO		VTR	VTR/VCC	Low	
88	2	Reserved	Reserved		Reserved	Reserved		
88	3	Reserved	Reserved		Reserved	Reserved		
88	Strap							
89	Default: 0	GPIO130	PIO	I-4	VTR	VTR/VCC	No Gate	
89	1	SMB03_DATA	PIO		VTR	VTR	High	Note 4
89	2	SMB03_DATA18	PIO		VTR	VTR	High	Note 11
89	3	Reserved	Reserved		Reserved	Reserved		

MEC141x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
89	Strap							
90	Default: 0	GPIO035	PIO	I-4	VTR	VTR/VCC	No Gate	
90	1	Reserved	Reserved		Reserved	Reserved		
90	2	SB-TSI_CLK	SB-TSI		SB-TSI	SB-TSI	High	
90	3	Reserved	Reserved		Reserved	Reserved		
90	Strap							
91	Default: 0	GPIO131	PIO	I-4	VTR	VTR/VCC	No Gate	
91	1	SMB03_CLK	PIO		VTR	VTR	High	Note 4
91	2	SMB03_CLK18	PIO		VTR	VTR	High	Note 11
91	3	Reserved	Reserved		Reserved	Reserved		
91	Strap							
92	Default: 0	GPIO132	PIO	I-4	VTR	VTR/VCC	No Gate	
92	1	KSO16	PIO		VTR	VTR	Reserved	Note 15
92	2	Reserved	Reserved		Reserved	Reserved		
92	3	Reserved	Reserved		Reserved	Reserved		
92	Strap							
93	Default: 0	GPIO140	PIO	I-4	VTR	VTR/VCC	No Gate	
93	1	KSO17	PIO		VTR	VTR	Reserved	Note 15
93	2	Reserved	Reserved		Reserved	Reserved		
93	3	Reserved	Reserved		Reserved	Reserved		
93	Strap							
94	Default: 0	GPIO033	PIO	I-4	VTR	VTR/VCC	No Gate	
94	1	PECI_DAT	PECI_IO		PECI_IO	PECI_IO	Low	Note 12
94	2	SB_TSI_DAT	SB-TSI		SB-TSI	SB-TSI	Low	Note 12
94	3	Reserved	Reserved		Reserved	Reserved		
94	Strap							
95		VREF_CPU	VREF_CPU		VREF_CPU	VREF_CPU		Note 6, Note 18
95								
95								
95								
95	Strap							
96	Default: 0	GPIO141	PIO	I-4	VTR	VTR/VCC	No Gate	
96	1	SMB04_DATA	PIO		VTR	VTR	High	Note 4
96	2	SMB04_DATA18	PIO		VTR	VTR	High	Note 11
96	3	Reserved	Reserved		Reserved	Reserved		
96	Strap							
97	Default: 0	GPIO142	PIO	I-4	VTR	VTR/VCC	No Gate	
97	1	SMB04_CLK	PIO		VTR	VTR	High	Note 4
97	2	SMB04_CLK18	PIO		VTR	VTR	High	Note 11
97	3	Reserved	Reserved		Reserved	Reserved		
97	Strap							
98	Default: 0	GPIO143	PIO	I-4	VTR	VTR/VCC	No Gate	Note 9

MEC141x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
98	1	KSI0	PIO		VTR	VTR	Low	Note 15
98	2	DTR#	PIO		VTR	VTR	Reserved	
98	3	Reserved	Reserved		Reserved	Reserved		
98	Strap							
99	Default: 0	GPIO144	PIO	I-4	VTR	VTR/VCC	No Gate	
99	1	KSI1	PIO		VTR	VTR	Low	Note 15
99	2	DCD#	PIO		VTR	VTR	High	
99	3	Reserved	Reserved		Reserved	Reserved		
99	Strap							
100		VSS	PWR		PWR	PWR		
100								
100								
100								
100	Strap							
101	Default: 0	GPIO145 (ICSP_CLOCK)	PIO	I-4	VTR	VTR/VCC	No Gate	Note 2
101	1	Reserved	PIO		Reserved	Reserved		
101	2	Reserved	Reserved		Reserved	Reserved		
101	3	Reserved	Reserved		Reserved	Reserved		
101	Strap							
102	Default: 0	GPIO146 (ICSP_DATA)	PIO	I-4	VTR	VTR/VCC	No Gate	Note 2
102	1	Reserved	PIO		Reserved	Reserved		
102	2	Reserved	Reserved		Reserved	Reserved		
102	3	Reserved	Reserved		Reserved	Reserved		
102	Strap							
103		VTR	PWR		PWR	PWR		
103								
103								
103								
103	Strap							
104	Default: 0	GPIO147	PIO	I-4	VTR	VTR/VCC	No Gate	
104	1	KSI4	PIO		VTR	VTR	Low	Note 15
104	2	DSR#	PIO		VTR	VTR	High	
104	3	Reserved	Reserved		Reserved	Reserved		
104	Strap							
105	Default: 0	GPIO150	PIO	I-4	VTR	VTR/VCC	No Gate	
105	1	KSI5	PIO		VTR	VTR	Low	Note 15
105	2	RI#	PIO		VTR	VTR	High	
105	3	Reserved	Reserved		Reserved	Reserved		
105	Strap							
106	Default: 0	GPIO156	PIO	I-4	VTR	VTR/VCC	No Gate	
106	1	LED1	PIO		VTR	VTR	Reserved	

MEC141x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
106	2	Reserved	Reserved		Reserved	Reserved		
106	3	Reserved	Reserved		Reserved	Reserved		
106	Strap							
107	Default: 0	GPIO151	PIO	I-4	VTR	VTR/VCC	No Gate	
107	1	KSI6	PIO		VTR	VTR	Low	Note 15
107	2	RTS#	PIO		VTR	VTR	Reserved	
107	3	Reserved	Reserved		Reserved	Reserved		
107	Strap							
108	Default: 0	GPIO152	PIO	I-4	VTR	VTR/VCC	No Gate	
108	1	KSI7	PIO		VTR	VTR	Low	Note 15
108	2	CTS#	PIO		VTR	VTR	High	
108	3	Reserved	Reserved		Reserved	Reserved		
108	Strap							
109	Default: 0	GPIO153	PIO	I-2	VTR	VTR/VCC	No Gate	
109	1	ADC4	I_AN		I_AN	I_AN	No Gate	Note 8
109	2	Reserved	Reserved		Reserved	Reserved		
109	3	Reserved	Reserved		Reserved	Reserved		
109	Strap							
110	Default: 0	GPIO154	PIO	I-2	VTR	VTR/VCC	No Gate	
110	1	ADC3	I_AN		I_AN	I_AN	No Gate	Note 8
110	2	Reserved	Reserved		Reserved	Reserved		
110	3	Reserved	Reserved		Reserved	Reserved		
110	Strap							
111	Default: 0	GPIO155	PIO	I-2	VTR	VTR/VCC	No Gate	
111	1	ADC2	I_AN		I_AN	I_AN	No Gate	Note 8
111	2	Reserved	Reserved		Reserved	Reserved		
111	3	Reserved	Reserved		Reserved	Reserved		
111	Strap							
112		AVSS	PWR		PWR	PWR		
112								
112								
112								
112	Strap							
113	Default: 0	GPIO122	PIO	I-2	VTR	VTR/VCC	No Gate	
113	1	ADC1	I_AN		I_AN	I_AN	No Gate	Note 8
113	2	Reserved	Reserved		Reserved	Reserved		
113	3	Reserved	Reserved		Reserved	Reserved		
113	Strap							
114	Default: 0	GPIO121	PIO	I-2	VTR	VTR/VCC	No Gate	
114	1	ADC0	I_AN		I_AN	I_AN	No Gate	Note 8
114	2	Reserved	Reserved		Reserved	Reserved		
114	3	Reserved	Reserved		Reserved	Reserved		

MEC141x								
VTQFP Pin#	Mux	Signal Name	Buffer Type	Default Buffer Operation	Signal Power Well	Emulated Power Well	Gated State	Notes
114	Strap							
115	0	Reserved	Reserved		Reserved	Reserved		
115	Default: 1	ADC_VREF	ADC_VREF		ADC_VREF	ADC_VREF	No Gate	Note 18
115	2	Reserved	Reserved		Reserved	Reserved		
115	3	Reserved	Reserved		Reserved	Reserved		
115	Strap							
116	Default: 0	GPIO022	PIO	I-2	VTR	VTR/VCC	No Gate	
116	1	ADC5	I_AN		I_AN	I_AN	No Gate	Note 8
116	2	Reserved	Reserved		Reserved	Reserved		
116	3	Reserved	Reserved		Reserved	Reserved		
116	Strap							
117	Default: 0	GPIO023	PIO	I-2	VTR	VTR/VCC	No Gate	
117	1	ADC6	I_AN		I_AN	I_AN	No Gate	Note 8
117	2	A20M	PIO		VTR	VCC	Reserved	
117	3	Reserved	Reserved		Reserved	Reserved		
117	Strap							
118	Default: 0	GPIO024	PIO	I-2	VTR	VTR/VCC	No Gate	
118	1	ADC7	I_AN		I_AN	I_AN	No Gate	Note 8
118	2	Reserved	Reserved		Reserved	Reserved		
118	3	Reserved	Reserved		Reserved	Reserved		
118	Strap							
119	0	GPIO004	PIO		VBAT	VTR/VCC	No Gate	
119	Default: 1	BGPO	PIO	O-4 mA	VBAT	VBAT	Reserved	
119	2	Reserved	Reserved		Reserved	Reserved		
119	3	Reserved	Reserved		Reserved	Reserved		
119	Strap							
120	0	GPIO003	PIO		VBAT	VTR/VCC	No Gate	
120	Default: 1	SYSPWR_PRES	ILLK	ILLK-4	VBAT	VBAT	Low	Note 12
120	2	Reserved	Reserved		Reserved	Reserved		
120	3	Reserved	Reserved		Reserved	Reserved		
120	Strap							
121	0	GPIO036	PIO		VBAT	VTR/VCC	No Gate	
121	Default: 1	VCI_OUT	PIO	O-8 mA	VBAT	VBAT	Reserved	
121	2	Reserved	Reserved		Reserved	Reserved		
121	3	Reserved	Reserved		Reserved	Reserved		
121	Strap							
122		VBAT	PWR		PWR	PWR		
122								
122								
122								
122	Strap							
123	Default: 0	XTAL1	OCLK	OCLK	VBAT	VBAT	No Gate	

MEC141x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
123	1	Reserved	Reserved		Reserved	Reserved		
123	2	Reserved	Reserved		Reserved	Reserved		
123	3	Reserved	Reserved		Reserved	Reserved		
123	Strap							
124		VSS_VBAT	PWR		PWR	PWR		
124								
124								
124								
124	Strap							
125	Default: 0	XTAL2	ICLK	ICLK	VBAT	VBAT	No Gate	
125	1	Reserved	Reserved		Reserved	Reserved		
125	2	Reserved	Reserved		Reserved	Reserved		
125	3	Reserved	Reserved		Reserved	Reserved		
125	Strap							
126	0	GPIO162	PIO		VBAT	VTR/VCC	No Gate	
126	Default: 1	VCI_IN1#	ILLK	ILLK-4	VBAT	VBAT	High	Note 14
126	2	Reserved	Reserved		Reserved	Reserved		
126	3	Reserved	Reserved		Reserved	Reserved		
126	Strap							
127	0	GPIO163	PIO		VBAT	VTR/VCC	No Gate	
127	Default: 1	VCI_IN0#	ILLK	ILLK-4	VBAT	VBAT	High	Note 14
127	2	Reserved	Reserved		Reserved	Reserved		
127	3	Reserved	Reserved		Reserved	Reserved		
127	Strap							
128	0	GPIO164	PIO		VBAT	VTR/VCC	No Gate	
128	Default: 1	VCI_OVRD_IN	ILLK	ILLK-4	VBAT	VBAT	Low	Note 14
128	2	Reserved	Reserved		Reserved	Reserved		
128	3	Reserved	Reserved		Reserved	Reserved		
128	Strap							

2.11 Pin States After VTR Power-On

The following tables uses ‘Z’ to indicate a tristate signal.

2.11.1 DEFAULT OPERATION AFTER A VBAT AND VTR POR

Signal	VBAT Applied	VBAT Stable	VTR Applied	nSYS_RST De-asserted	VCC_PWRGD Asserted	VCC_PWRGD De-asserted	nSYS_RST Asserted	VTR Un-powered	VBAT Un-powered	Notes
GPIOXXX	unpow-ered	unpow-ered	low	In	In	In	Z	glitch	unpow-ered	
nRESET_IN	unpow-ered	unpow-ered	low	In	In	In	Z	glitch	unpow-ered	
ICSP_MCLR	unpow-ered	unpow-ered	low	In	In	In	Z	glitch	unpow-ered	
BGPO	low	Out=0	Retain	Retain	Retain	Retain	Retain	Retain	unpow-ered	Note A
SYS-PWR_PRES	In	In	In	In	In	In	In	In	unpow-ered	
VCI_INx#	In	In	In	In	In	In	In	In	unpow-ered	
VCI_OUT	Out logic	Out logic	Out logic	Out logic	Out logic	Out logic	Out logic	Out logic	unpow-ered	Note B
VCI_OVRD_IN	In	In	In	In	In	In	In	In	unpow-ered	

Note:

- ***Note A***: Pin is programmable by the EC and retains its value through a VTR power cycle.
- ***Note B***: Pin is programmable by the EC and affected by other VBAT inputs pins.

2.11.2 DEFAULT OPERATION AFTER A VTR POR ONLY (VBAT REMAINS ON)

The following table lists the VTR POR default conditions for VBAT powered pins where the EC had selected an alternate function that was not the default function.

Signal	VBAT Applied	VBAT Stable	VTR Applied	nSYS_RST De-asserted	VCC_PWRGD Asserted	VCC_PWRGD De-asserted	nSYS_RST Asserted	VTR Un-powered	VBAT Un-powered	Notes
GPIO003	N/A	N/A	low	In	In	In	Z	In	unpowered	Note C
GPIO004	N/A	N/A	low	In	In	In	Z	In	unpowered	Note C
GPIO036	N/A	N/A	low	In	In	In	Z	In	unpowered	Note C
GPIO162	N/A	N/A	low	In	In	In	Z	In	unpowered	Note C
GPIO163	N/A	N/A	low	In	In	In	Z	In	unpowered	Note C
GPIO164	N/A	N/A	low	In	In	In	Z	In	unpowered	Note C

Note:

- ***Note C:*** The GPIO Control logic is powered by VTR and loses its configuration through a VTR POR.

2.12 Strapping Options

2.12.1 BOOT SOURCE SELECT STRAPS

The Crisis Recovery Strap option (CR_STRAP) is implemented on GPIO102/KSO09[CR_STRAP].

- If this pin is connected to ground the Boot ROM will load the SPI Flash image from the SPI Flash located on the Private SPI Interface (PVT_xxxx).
- If this pin is pulled high, which is the normal operation for the Key Scan Interface, the Boot ROM will load the SPI Flash image from the Shared Flash Interface (SHD_xxxx) or the eSPI Flash channel as selected by the Boot Source Select Strap (BSS_STRAP) on the GPIO123/SHD_CS# pin.

CR_STRAP	BSS_STRAP	Source
0	X	Use 3.3V Private SPI
1	0	Use eSPI Flash Channel
	1	Use 3.3V Shared SPI

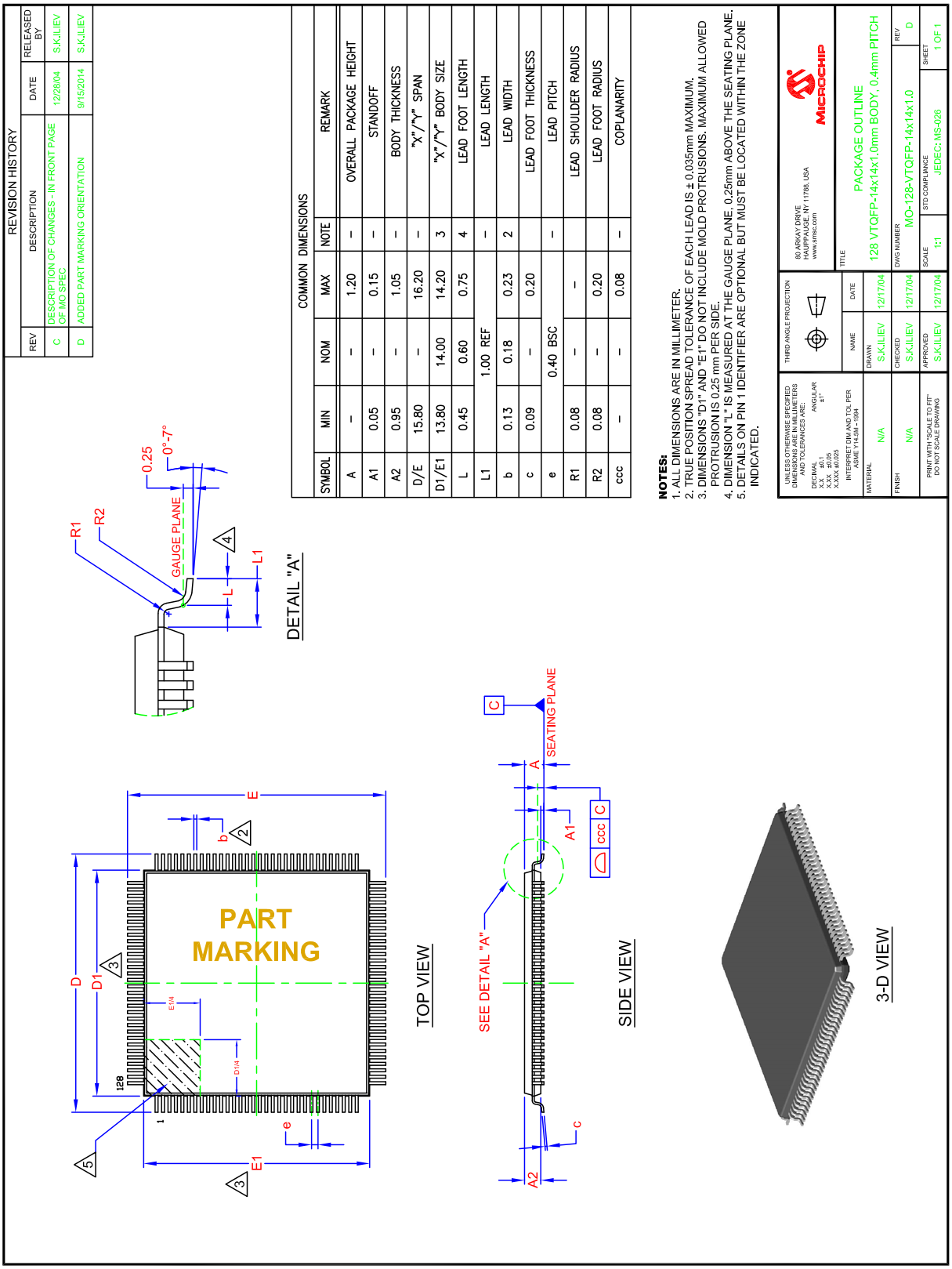
Note:

If the eSPI Flash Channel is used for booting, the GPIO123/SHD_CS# pin must be used as RSMRST#. This pin will be driven high by the boot ROM code in order to activate the eSPI flash channel.

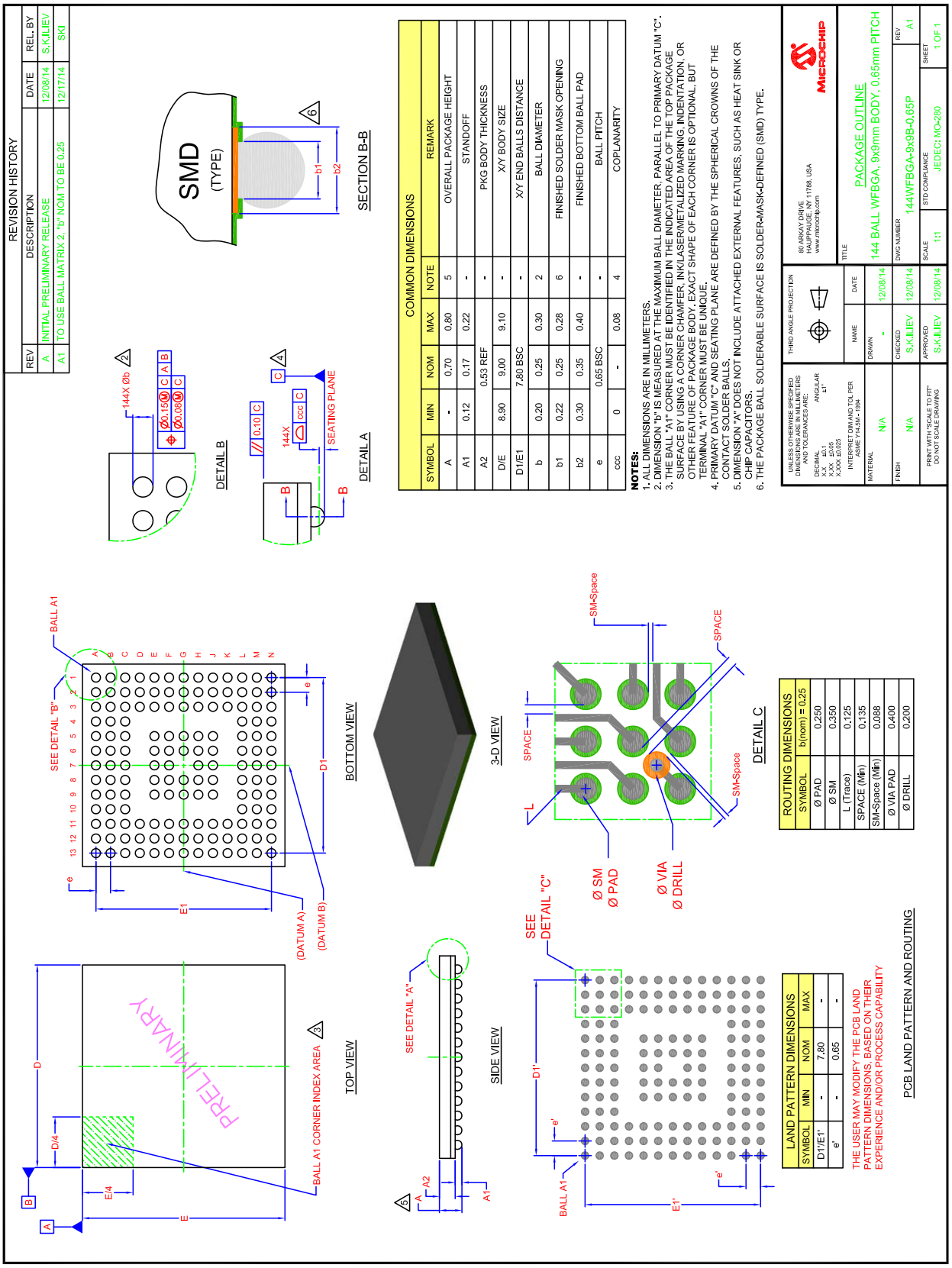
If the SHD_SPI port is used for booting, then any unused GPIO may be used for RSMRST#.

2.13 Packages

2.13.1 128-PIN VTQFP PACKAGE OUTLINE



2.13.2 144-PIN WFBGA PACKAGE OUTLINE



3.0 POWER, CLOCKS, AND RESETS

3.1 Introduction

The Power, Clocks, and Resets (PCR) chapter identifies all the power supplies, clock sources, and reset inputs to the chip and defines all the derived power, clock, and reset signals. In addition, this section identifies Power, Clock, and Reset events that may be used to generate an interrupt event, as well as, the Chip Power Management Features.

3.2 References

- eSPI Controller Specification, 2013 Microchip Technology

3.3 Interrupts

The Power, Clocks, and Resets logic generates no events

3.4 Power

3.4.1 POWER SOURCES

TABLE 3-1: lists the power supplies from which the MEC140x/1x draws current. These current values are defined in Section 42.3, "Power Consumption," on page 499.

TABLE 3-1: POWER SOURCE DEFINITIONS

Power Well	Nominal Voltage	Description	Source
VTR_33_18	3.3V or 1.8V	3.3V or 1.8V System Power Supply. This supply is used to power the host interface to this chip. It is either connected to 3.3V VTR power supply or the eSPI 1.8V power supply.	Pin Interface
VTR	3.3V	3.3V System Power Supply. This is typically connected to the “Always-on” or “Suspend” supply rails in system. This supply must be on prior to the system RSMRST# signal being deasserted. This supply is used to derive the chip’s core power and to supply the 3.3V I/O rail.	Pin Interface
VBAT (Note 3-1)	3.0V	System Battery Back-up Power Well. This is the “coin-cell” battery.	Pin Interface

Note: The minimum rise/fall time requirement on VTR is 200us. The minimum rise/fall time requirement on VTR_33_18 is 100mV/usec or 18us. VTR_33_18 must turn on at the same time or after the 3.3V VTR supply is powered.

Note: The Minimum rise time requirement on VBAT is 100us.

Note 3-1 Note on Battery Replacement: Microchip recommends removing all power sources to the device defined in Table 3-1, "Power Source Definitions" and all external voltage references defined in

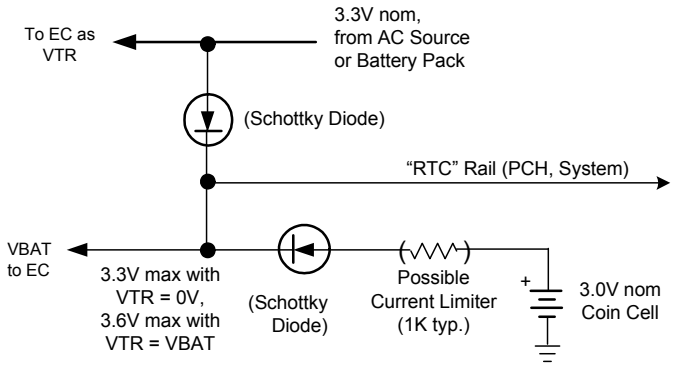
Table 3-2, "Voltage Reference Definitions" before removing and replacing the battery. In addition, upon removing the battery, ground the battery pin before replacing the battery.

APPLICATION NOTE: Battery Circuit Requirement:

- VBAT must always be present if VTR is present.

The following circuit is recommended to fulfill this requirement:

FIGURE 3-1: RECOMMENDED BATTERY CIRCUIT



3.4.2 VOLTAGE REFERENCES

TABLE 3-2: lists the External Voltage References to which the MEC140x/1x provides high impedance interfaces.

TABLE 3-2: VOLTAGE REFERENCE DEFINITIONS

Power Well	Nominal Input Voltage	Scaling Ratio	Nominal Monitored Voltage	Description	Source
VREF_CPU (Note 3-2, Note 3-3)	Variable	n/a	Variable	Processor Voltage External Voltage Reference Used to scale Processor Interface signals	Pin Interface
DAC_VREF	Variable	n/a	Variable	DAC Reference Voltage	Pin Interface
CMP_VREF0	Variable	n/a	Variable	Determines reference voltage on the negative terminal of Comparator 0	Pin Interface
CMP_VREF1	Variable	n/a	Variable	Determines reference voltage on the negative terminal of Comparator 1	Pin Interface
ADC_VREF	Variable	n/a	Variable	ADC Reference Voltage	Pin Interface

Note 3-2 For specific electrical characteristics for the voltage reference inputs see Table 42-5, "DC Electrical Characteristics," on page 491.

Note 3-3 In order to achieve the lowest leakage current when both PECI and SB TSI are not used, set the VREF_CPU Disable bit to 1. This bit is defined in Section 34.8.5, VREF_CPU DISABLE.

3.4.3 POWER GOOD SIGNALS

The power good timing and thresholds are defined in the Section 43.1, "Voltage Thresholds and Power Good Timing," on page 501.

TABLE 3-3: POWER GOOD SIGNAL DEFINITIONS

Power Good Signal	Description	Source
VTRGD	VTRGD is an internal power good signal used to indicate when the VTR rail is on and stable.	VTRGD is asserted following a delay after the VTR power well exceeds its preset voltage threshold. VTRGD is de-asserted as soon as either of these voltages drop below this threshold. **Note:** See Section 43.1.1, "VTR Threshold and VTRGD Timing," on page 501.
VCC_PWRGD	VCC_PWRGD is used to indicate when the main power rail voltage is on and stable.	VCC_PWRGD Input pin

3.4.4 SYSTEM POWER SEQUENCING

The following table defines the behavior of the Power Sources in each of the defined ACPI power states.

TABLE 3-4: TYPICAL POWER SUPPLIES VS. ACPI POWER STATES

Supply Name	ACPI Power State						Description
	S0 (FULL ON)	S1 (POS)	S3 (STR)	S4 (STD)	S5 (Soft Off)	G3 (MECH Off)	
VTR_33_18	ON	ON	ON/OFF	ON/OFF	ON/OFF	OFF	LPC/eSPI Host Interface Power Supply.
VTR	ON	ON	ON	ON	ON	OFF	“Always-on” Supply. (Note 3-4)
VBAT	ON	ON	ON	ON (Note 3-5)	ON (Note 3-5)	ON (Note 3-5)	Battery Back-up Supply

Note 3-4 VTR power supply is always on while the battery pack or ac power is applied to the system.

Note 3-5 This device requires that the VBAT power is on when the VTR power supply is on. External circuitry, a diode isolation circuit, is implemented on the motherboard to extend the battery life. This external circuitry ensures the VBAT pin will derive power from the VTR power well when it is on. Therefore, the VBAT supply will never appear to be off when the VTR rail is on. See APPLICATION NOTE: on page 65.

3.5 Clocks

The following section defines the clocks that are generated and derived.

3.5.1 RAW CLOCK SOURCES

The table defines raw clocks that are either generated externally or via an internal oscillator.

TABLE 3-5: SOURCE CLOCK DEFINITIONS

Clock Name	Frequency	Description	Source
SUSCLK	32.768 kHz	32.768 kHz Suspend Well Clock Source is a single-ended input that is an accurate 32.768 kHz clock. (Note 3-6)	Pin Interface (XTAL2)
32.768 kHz Crystal Oscillator	32.768 kHz	A 32.768 kHz parallel resonant crystal connected between the XTAL1 and XTAL2 pins.	Pin Interface (XTAL1 and XTAL2)
48 MHz Ring Oscillator	48 MHz	The 48 MHz Ring Oscillator is a high-accuracy, low power, low start-up latency 48 MHz Ring Oscillator.	Internal Oscillator powered by VTR. May be stopped by Chip Power Management Features. (Note 3-7)
32kHz_INT_OSC	32.768 kHz	32.768 kHz low power, lower accuracy Internal Oscillator powered by VBAT supply. **Note:** This clock may be used when the external 32 kHz clock is unavailable, and will allow the 48 MHz clock to reach frequency lock.	Internal Oscillator **Note:** This clock source is enabled via the Clock Enable Register on page 440.

Note 3-6 The chipset will not produce a valid 32 kHz clock until about 5 ms (PCH) or 110 ms (ICH) after the deassertion of RSMRST#. See chipset specification for the actual timing.

Note 3-7 The 48 MHz Ring Oscillator will reach frequency lock if either the external 32kHz clock source or the 32 kHz Internal Oscillator is used, as selected via the 48MHz Oscillator Reference Select bit in the Clock Enable Register on page 440. The external 32k Hz clock source provides a stable timebase for the 48 MHz Ring Oscillator as well as the clock source for the 32 kHz Clock Domain. After VBAT POR there is a 500ms max time for the 48 MHz Ring Oscillator to become accurate. See Section 43.2, "Clocking AC Timing Characteristics," on page 504.

Note 3-8 Without the external clock, the 48MHz clock will vary up to +/-4% which may affect the timing parameters of certain blocks. In particular it may not be accurate enough to ensure that the UART will work, depending on the accuracy of the clock of the external device.

3.5.2 DERIVED CLOCKS

This table defines the clocks derived from the raw clock sources.

TABLE 3-6: DERIVED CLOCK DEFINITIONS

Clock Name	Frequency	Description	Source
EC_PROC_CLK	Programmable	Derived clock for Embedded Controller/Processor	48 MHz Ring Oscillator
24MHz_Clk	24 MHz	Derived clock for UART	48 MHz Ring Oscillator
16MHz_Clk	16 MHz	Derived clock for SMBus Controller	48 MHz Ring Oscillator
2MHz_Clk	2 MHz	Derived clock for PS/2 Controller	48 MHz Ring Oscillator

TABLE 3-6: DERIVED CLOCK DEFINITIONS (CONTINUED)

Clock Name	Frequency	Description	Source
1.8432MHz_Clk	1.843 MHz	Derived clock for UART	48 MHz Ring Oscillator
1MHz_Clk	1 MHz	Derived clock for 8042 Emulated Keyboard Controller	48 MHz Ring Oscillator
100kHz_Clk	100 kHz	Derived clock for PWM and TACH blocks	48 MHz Ring Oscillator
32KHz_Clk	32.768 kHz	Internal 32kHz clock domain	Pin Interface or 48 MHz Ring Oscillator: Pins: XTAL2: 32 kHz Crystal input/ single-ended clock source input pin. XTAL1: 32 kHz Crystal output The XOSEL bit configures the source of this clock domain as either a single-ended 32.768 kHz clock input (SUSCLK) or the 32.768 kHz Crystal Oscillator. If neither of these is available, this clock domain is derived from the 32kHz_INT_OSC or the 48 MHz Ring Oscillator, as configured by bits in the Clock Enable Register. (See Note 3-9)
5Hz_Clk	5 Hz	Derived clock for Breathing LED block	48 MHz Ring Oscillator

Note 3-9 Bits[4:0] of the Clock Enable Register on page 440 determine the source of the 32KHz_Clk.

3.5.3 GENERATED CLOCK OUTPUTS

This section describes clocks generated by the MEC140x/1x that may be used by the external system.

TABLE 3-7: GENERATED CLOCK DEFINITIONS

Clock Name	Frequency	Description	Source
32KHZ_OUT	32.768 kHz	32.768 kHz output. Configured 32kHz clock source routed to pin interface.	Derived 32KHz_Clk

3.5.4 32 KHZ CLOCK SWITCHING

The 32kHz clock switching logic switches the clock source of the 32kHz clock domain to be either the single-ended 32.768 kHz clock input (SUSCLK), the 32.768 kHz Crystal Oscillator, the 32kHz_INT_OSC or 48 MHz Ring Oscillator. Following a VBAT_POR, the XOSEL, 32KHz Clock Switcher Control, INT_32K_OSC_EN and EXT_32K_OSC_EN bits in the Clock Enable Register are programmed to configure the source of this clock domain. See Table 35-3, “32kHz Clock Control,” on page 441.

Note 1: Switching delay when configuring the 32k Hz clock source will be on the order of 100 us or three 32k Hz clocks.

2: The 48 MHz Ring Oscillator will reach frequency lock if either the external 32kHz clock source or the 32 kHz Internal Oscillator is used, as selected via the 48MHz Oscillator Reference Select bit in the Clock Enable Register on page 440. The 48 MHz Ring Oscillator will remain locked when the external 32kHz clock source is removed.

3.5.5 CLOCK DOMAINS VS. ACPI POWER STATES

Table 3-8, "Typical MEC140x/1x Clocks vs. ACPI Power States" shows the relationship between ACPI power states and MEC140x/1x clock domains:

TABLE 3-8: TYPICAL MEC140X/1X CLOCKS VS. ACPI POWER STATES

Clock Name	ACPI Power State						Description
	S0 (FULL ON)	S1 (POS)	S3 (STR)	S4 (STD)	S5 (Soft Off)	G3 (MECH Off)	
SUSCLK	ON	ON	ON	ON	ON	OFF	This clock is the system suspend clock source. (Note 3-6).
32.768 kHz Crystal Oscillator	ON	ON	ON	ON	ON	ON	This clock is generated from a 32.768 kHz parallel resonant crystal connected between the XTAL1 and XTAL2 pins.
32KHz_Clk	ON	ON	ON	ON	ON	ON/ OFF	This clock domain is generated from the 32 kHz clock input (SUSCLK) when available or the crystal oscillator pins. Otherwise it is generated internally from the 32kHz_INT_OSC or the 48 MHz Ring Oscillator.
48 MHz Ring Oscillator	ON	ON	ON	ON	ON	OFF	This clock is powered by the MEC140x/1x suspend supply (VTR) but may start and stop as described in Section 3.7, "Chip Power Management Features," on page 71 (see also Note 3-4).
32kHz_INT_OSC	ON	ON	ON	ON	ON	OFF	This clock is powered by the MEC140x/1x VBAT power supply. This clock may be used when 48 MHz Ring Oscillator is not available.

3.6 Resets

FIGURE 3-2: RESETS DIAGRAM (MEC140X/1X)

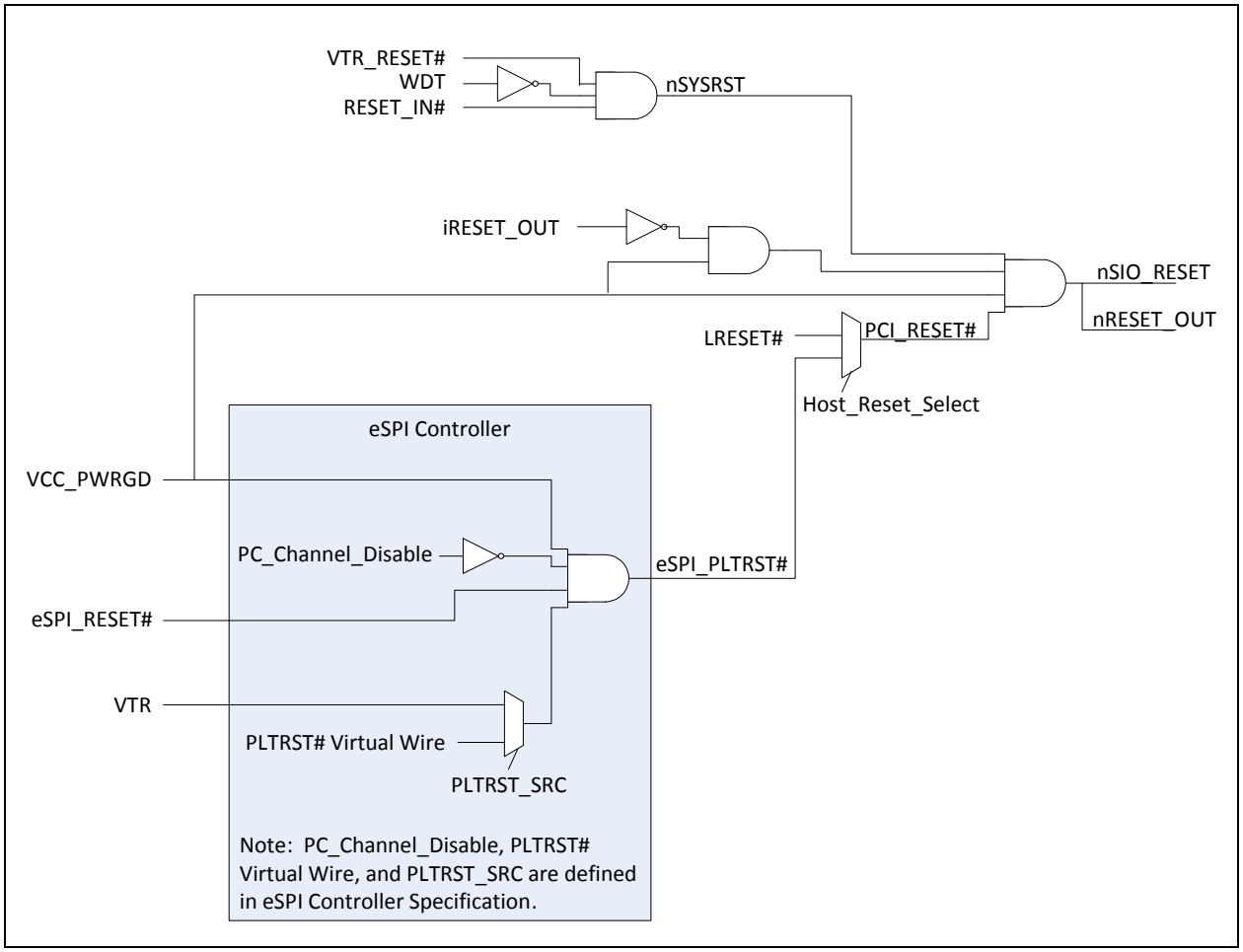


TABLE 3-9: DEFINITION OF RESET SIGNALS

Reset	Description	Source
VBAT_POR	Internal VBAT Reset signal. This signal is used to reset VBAT powered registers.	VBAT_POR is a pulse that is asserted at the rising edge of VTRGD if the VBAT voltage is below a nominal 1.25V. VBAT_POR is also asserted as a level if, while VTRGD is not asserted (‘0’), the coin cell is replaced with a new cell that delivers at least a nominal 1.25V. In this latter case VBAT_POR is de-asserted when VTRGD is asserted. No action is taken if the coin cell is replaced, or if the VBAT voltage falls below 1.25 V nominal, while VTRGD is asserted.
VTR_RESET#	VTR_RESET# is a Power-On-Reset.	VTR_RESET# is deasserted at the rising edge of VTRGD and is asserted only when VTRGD is low.

TABLE 3-9: DEFINITION OF RESET SIGNALS (CONTINUED)

Reset	Description	Source
nRESET_IN	External Pin that can generate the equivalent of a VTR POR event. Asserting this signal will cause the nSYSRST to be asserted, which resets the majority of the chip.	Pin Interface
nRESET_OUT	External Pin that can generate the equivalent of a VCC POR or main reset event to other external devices.	This signal is asserted low when the nSIO_RESET is asserted low.
nSYSRST	Internal VTR Reset signal. This signal is used to reset VTR powered registers.	nSYSRST is asserted when VTRGD is low, when a WDT_RESET event occurs, when the nRESET_IN pin is asserted low, or when the EJTAG.PrRST bit is asserted. It is only deasserted when VTRGD is high, nRESET_IN is high, the EJTAG.PrRST bit is deasserted,.and their is no WDT_RESET event active. The EJTAG.PrRST bit is defined in the MIPS® EJTAG Specification, DN: MD00047, Rev 5.06, March 05, 2011.
LRESET#	System reset signal connected to the LPC LRESET# pin (also referred to as PCI Reset).	Pin Interface
eSPI_RESET#	System reset signal connected to the eSPI eSPI_RESET# pin	Pin Interface
eSPI_PLTRST#	Platform Reset.	Generated by the eSPI Block when VCC_PWRGD is low, when eSPI_RESET# is low, by a Virtual Wire, or by PC_Channel_Disable.
PCI_RESET#	System reset signal	Generated by either the LPC LRESET# pin (also referred to as PCI Reset) or the eSPI_PLTRST# depending on the configuration of the Host_Reset_Select bit.
nSIO_RESET	Performs a reset when VCC is turned off or when the system host resets the LPC or eSPI Host Interfaces.	nSIO_RESET is a signal that is asserted if nSYSRST is low, VCC_PWRGD is low, or PCI_RESET# is asserted low and may be deasserted when these three signals are all high. The iRESET_OUT bit controls the deassertion of nSIO_RESET. A WDT_RESET event will also cause an nSIO_RESET assertion.
WDT_RESET	Internal WDT Reset signal. This signal resets VTR powered registers with the exception of the WDT Event Count register. Note that the glitch protect circuits do not activate on a WDT reset. WDT_RESETdoes not reset VBAT registers or logic.	A WDT_RESET is asserted by a WDT Event. This event is indicated by the WDT bit in the Power-Fail and Reset Status Register
EC_PROC_RESET#	Internal reset signal to reset the processor in the EC Subsystem.	An EC_PROC_RESET# is a stretched version of the nSYSRST. This reset asserts at the same time that nSYSRST asserts and is held asserted for 1ms after the nSYSRST deasserts.

3.7 Chip Power Management Features

This device is designed to always operate in its lowest power state during normal operation. In addition, this device offers additional programmable options to put individual logical blocks to sleep as defined in Section 3.7.1, "Block Low Power Modes," on page 72 and to gate off or power down the internal oscillator as described in Section 3.7.2, "Configuring the Chip's Sleep States," on page 72.

3.7.1 BLOCK LOW POWER MODES

All power related control signals are generated and monitored centrally in the chip's Power, Clocks, and Resets (PCR) block. The power manager of the PCR block uses a sleep interface to communicate with all the blocks. The sleep interface consists of three signals:

- **sleep_en (request to sleep the block)** is generated by the PCR block. A group of sleep_en signals are generated for every clock segment. Each group consists of a sleep_en signal for every block in that clock segment.
- **clk_req (request clock on)** is generated by every block. They are grouped by blocks on the same clock segment. The PCR monitors these signals to see when it can gate off clocks.
- **reset_en (reset on sleep)** bits determine if the block (including registers) will be reset when it enters sleep mode.

A block can always drive clk_req low synchronously, but it MUST drive it high asynchronously since its internal clocks are gated and it has to assume that the clock input itself is gated. Therefore the block can only drive clk_req high as a result of a register access or some other input signal.

The following table defines a block's power management protocol:

Power State	sleep_en	clk_req	Description
Normal operation	Low	Low	Block is idle and NOT requesting clocks. The block gates its own internal clock.
Normal operation	Low	High	Block is NOT idle and requests clocks.
Request sleep	Rising Edge	Low	Block is IDLE and enters sleep mode immediately. The block gates its own internal clock. The block cannot request clocks again until sleep_en goes low.
Request sleep	Rising Edge	High then Low	Block is not IDLE and will stop requesting clocks and enter sleep when it finishes what it is doing. This delay is block specific, but should be less than 1 ms. The block gates its own internal clock. After driving clk_req low, the block cannot request clocks again until sleep_en goes low.
Register Access	X	High	Register access to a block is always available regardless of sleep_en. Therefore the block ungates its internal clock and drives clk_req high during the access. The block will regate its internal clock and drive clk_req low when the access is done.

A wake event clears all sleep enable bits momentarily, and then returns the sleep enable bits back to their original state. The block that needs to respond to the wake event will do so. See Section 10.11.3, "Wake-Capable Interrupt Events," on page 162.

The Sleep Enable, Clock Required and Reset Enable registers are defined in Section 3.8, "EC-Only Registers," on page 74.

3.7.2 CONFIGURING THE CHIP'S SLEEP STATES

The chip supports four sleep states: SYSTEM HEAVY SLEEP 1, SYSTEM HEAVY SLEEP 2, SYSTEM HEAVY SLEEP 3, SYSTEM DEEPEST SLEEP. The chip will enter one of these four sleep states only when all the blocks have been commanded to sleep and none of them require the 48 MHz Ring Oscillator (i.e., all clock required status bits = 0), and the processor has executed its sleep instruction. These sleep states must be selected by firmware via the System Sleep Control bits implemented in the System Sleep Control Register (SYS_SLP_CNTRL) on page 81 prior to issuing the sleep instruction. Table 3-12, "System Sleep Control Bit Encoding," on page 82 defines each of these sleep states.

There are two ways to command the chip blocks to enter sleep.

1. Assert the Sleep All bit located in the System Sleep Control Register (SYS_SLP_CNTRL) on page 81
2. Assert all the individual block sleep enable bits

Blocks will only enter sleep after their sleep signal is asserted and they no longer require the 48 MHz Ring Oscillator source. Each block has a corresponding clock required status bit indicating when the block has entered sleep. The general operation is that a block will keep the 48 MHz Ring Oscillator on until it completes its current transaction. Once the block has completed its work, it deasserts its clock required signal. Blocks like timers, PWMs, etc. will deassert their clock required signals immediately. See the individual block Low Power Mode sections to determine how each individual block enters sleep.

3.7.3 DETERMINING WHEN THE CHIP IS SLEEPING

There are two methods to verify the chip's clock has stopped, which indicates the device is in one of these three sleep states: SYSTEM HEAVY SLEEP 2, SYSTEM HEAVY SLEEP 3, SYSTEM DEEPEST SLEEP. Note that the 48 MHz Ring Oscillator continues to run in the SYSTEM HEAVY SLEEP 1 state to minimize wake latency.

Option 1: TST_CLK_OUT pin

The TST_CLK_OUT, which is located on the GPIO157/LED0/TST_CLK_OUT pin, is used to route the internal 48 MHz Ring Oscillator to a pin. If the clock is toggling the chip is in the full on running state. if the clock is not toggling the chip has entered the programmed sleep state.

Option 2: MTAP Test Bit

Bit [1] SLEEPING has been implemented in the MTAP registers (MCHP_CMD <0x07>) to allow the firmware developer to determine if the chip is sleeping via the ICSP debug port. This MTAP command does not require the 48 MHz Ring Oscillator to be clocking and therefore will not change the chip's sleep state. Note that all of the ICSP debugger commands that access the processor JTAG port will bring the device out of sleep.

3.7.4 WAKING THE CHIP FROM SLEEPING STATE

The chip will remain in the configured sleep state until it detects either a wake event, an ICSP access, or a full VTR POR. All the wake-capable interrupt events are defined in the Section 10.0, "Jump Table Vectored Interrupt Controller (JTVIC)". They are identified as Wake Events in Table 10-2, “Interrupt Source, Enable Set, Enable Clear, and Result Bit Assignments,” on page 164.

3.7.4.1 Wake-Only Events

Two GIRQ registers have been reserved for special wake events. GIRQ16 is used for wake-events that do not require software processing. These events are used to turn the clock on so the peripherals can start processing the data. There is no information for the firmware to process. When GIRQ16 is active the firmware can simply clear the source and return to the sleep state. GIRQ22 is a duplicate of GIRQ16 with one major difference. GIRQ22 does not generate a processor interrupt. It only wakes the 48 MHz Ring Oscillator so the peripherals can start processing the data.

Example: LPC I/O Traffic targeting EMI block.

The LPC Interface detects traffic on the bus and requires the clock to be on to process the incoming data. If GIRQ22 is enabled, the LPC block will be able to autonomously receive data for the programmed I/O ranges without processor intervention. Once the data is loaded into the HOST-to-EC Mailbox Register the Host-to-EC IRQ will trigger an interrupt to the embedded controller to service this command.

An alternate solution would be to enable the GIRQ16 LPC interrupt. The process is similar, except the embedded controller will receive an interrupt for the LPC activity, as well as the Host-to-EC IRQ, and will need to clear this event also.

3.7.4.2 ICSP Debugger Wake Events

The ICSP Debugger will cause the chip to wake and run debug code. Auto Clear Sleep and Sleep Debug bits have been implemented to allow firmware to re-enter sleep following a debug access. It is recommended to set these bits to ‘1’ as described in the following Application Note.

APPLICATION NOTE: Methods for putting the device back to sleep after a debug access.

Option 1: Automatically Re-enter Sleep after Debug Wake Event (preferred)

To automatically re-enter sleep after a debug wake event the firmware should follow this recommended usage model

1. FW has decided to go to sleep.
2. Set sleep_all bit to command all blocks to sleep.
3. Set sleep_debug bit.
4. Set auto_clr_sleep to make sure sleep_all and sleep_debug will clear automatically when the processor vectors to an interrupt.

Note:

- Steps 2-4 can be done in one write to System Sleep Control Register (SYS_SLP_CNTRL) register.
- The sleep_all and the sleep_debug bits MUST not be set in an interrupt handler.

5. Issue processor sleep instuction. Note that you must use a do...while around the sleep instruction (WAIT) . Stay in loop while sleep_debug bit is still set.

6. processor goes to sleep.

Option 2: Debug_Done Interrupt Event

Firmware can enable the Debug_Done interrupt event before issuing the processor sleep instruction. This bit is asserted when the debugger accesses the device. However, the user code will not see this event until the debugger has completed its debug task. Once the user code sees this event the chip may be put back into a sleep state. Note that the sleep control bits may have been modified by the debug activity, so some additional reprogramming may be necessary.

3.8 EC-Only Registers

TABLE 3-10: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
PCR	0	EC	32-bit internal address space	0008_0100h

TABLE 3-11: POWER, CLOCKS AND RESET VTR-POWERED REGISTERS SUMMARY

Offset	Register Name
00h	Test Register
04h	Test Register
08h	EC Sleep Enable Register (EC_SLP_EN)
0Ch	EC Clock Required Status Registers (EC_-CLK_REQ_STS)
10h	Host Sleep Enable Register (HOST_SLP_EN)
14h	Host Clock Required Status Registers (HOST_-CLK_REQ)
18h	System Sleep Control Register (SYS_SLP_CNTRL)
20h	Processor Clock Control Register (PROC_CLK_CN-TRL)
24h	EC Sleep Enable 2 Register (EC_SLP_EN2)
28h	EC Clock Required 2 Status Register (EC_-CLK_REQ2_STS)
2Ch	Slow Clock Control Register (SLOW_CLK_CNTRL)
30h	Oscillator ID Register (CHIP_OSC_ID)
34h	PCR chip sub-system power reset status (CHIP_P-WR_RST_STS)
38h	Test Register
3Ch	Host Reset Enable Register (HOST_RST_EN)
40h	EC Reset Enable Register (EC_RST_EN)
44h	EC Reset Enable 2 Register (EC_RST_EN2)
48h	Power Reset Control (PWR_RST_CTRL) Register

Note: All register addresses are naturally aligned on 32-bit boundaries. Offsets for registers that are smaller than 32 bits are reserved and must not be used for any other purpose.

3.9 Sleep Enable and Clock Required Registers

The following are the Sleep Enable and Clock Required registers for the MEC140x/1x.

3.9.1 EC SLEEP ENABLE REGISTER (EC_SLP_EN)

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	TIMER16_1 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment. See Note: on page 76.	R/W	0h	nSYSRST
30	TIMER16_0 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment. See Note: on page 76.	R/W	0h	nSYSRST
29	EC_REG_BANK Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
28:27	RESERVED	RES		
26	PWM7 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
25	PWM6 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
24	PWM5 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
23	PWM4 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
22	PWM3 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
21	PWM2 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
20	PWM1 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
19:12	RESERVED	RES		

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
11	TACH1 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
10	SMB0 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
9	WDT Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
8	PROCESSOR Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
7	TFDP Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
6	DMA Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
5	PMC Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
4	PWM0 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
3	RESERVED	RES		
2	TACH0 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
1	PECI Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
0	INT Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST

Note: The basic timers in this device have an auto-reload mode. When this mode is selected, the block's clk_req equation is always asserted, which will prevent the device from gating its clock tree and going to sleep. When the firmware intends to put the device to sleep, none of the timers should be in auto-reload mode. Alternatively, use the timer's HALT function inside the control register to stop the timer in auto-reload mode so it can go to sleep.

3.9.2 EC CLOCK REQUIRED STATUS REGISTERS (EC_CLK_REQ_STS)

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	TIMER16_1 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
30	TIMER16_0 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
29	EC_REG_BANK Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
28:27	RESERVED	RES		
26	PWM7 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
25	PWM6 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
24	PWM5 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
23	PWM4 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
22	PWM3 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
21	PWM2 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
20	PWM1 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
19:12	RESERVED	RES		
11	TACH1 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
10	SMB0 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
9	WDT Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
8	PROCESSOR Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	1h	nSYSRST
7	TFDP Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
6	DMA Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
5	PMC Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
4	PWM0 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
3	RESERVED	RES		
2	TACH0 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
1	PECI Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
0	INT Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST

3.9.3 HOST SLEEP ENABLE REGISTER (HOST_SLP_EN)

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:21	RESERVED	RES		
20	Reserved - Should be set to ‘1’	R/W	0h	nSYSRST
19	eSPI Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
18	RESERVED	RES		
17	Mailbox Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
16	8042EM Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
15	ACPI PM1 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
14	ACPI EC 1 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
13	ACPI EC 0 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
12	GLBL_CFG 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
11	ACPI_EC3 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
10	ACPI_EC2 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
9:4	RESERVED	RES		
3	BIOS1 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
2	BIOS0 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
1	UART 0 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
0	LPC Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST

3.9.4 HOST CLOCK REQUIRED STATUS REGISTERS (HOST_CLK_REQ)

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:21	RESERVED	RES		
20	Reserved	R	0h	nSYSRST
19	eSPI Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
18	RESERVED	RES		
17	Mailbox Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
16	8042EM Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
15	ACPI PM1 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
14	ACPI EC 1 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
13	ACPI EC 0 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
12	GLBL_CFG Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	-	nSYSRST
11	ACPI EC 3 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
10	ACPI EC 2 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
9:4	RESERVED	RES		
3	BIOS1 Clock Required 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R	-	nSYSRST
2	BIOS0 Clock Required 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R	-	nSYSRST

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	UART 0 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	-	nSYSRST
0	LPC Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	-	nSYSRST

3.9.5 SYSTEM SLEEP CONTROL REGISTER (SYS_SLP_CNTRL)

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	RESERVED	RES		
6	Auto Clear Sleep 0: Sleep All and Sleep Debug are not cleared by HW when processor vectors to an interrupt, 1: Sleep All and Sleep Debug will be cleared by HW when the processor vectors to an interrupt.	R/W	0h	nSYSRST
5	Sleep Debug 0: don't keep processor asleep after debug wake, 1: keep processor asleep after a debug wake. If the Auto Clear Sleep bit is set, HW clears this bit when the processor vectors to an interrupt. (same as Sleep All bit). Firmware must play a role in keeping the processor asleep after a debug wake. Firmware needs to implement a Do-While loop around the processors sleep instruction. While this bit is 1, the sleep instruction must be re-executed. **Note:** See Application Note below this table.	R/W	0h	nSYSRST
4	Sleep All 0: blocks are not commanded to sleep, 1: all blocks are commanded to sleep. **Note:** If the Auto Clear Sleep bit is set, HW clears this bit when the processor vectors to an interrupt.	R/W	0h	nSYSRST
3	RESERVED	RES		
2	Core regulator standby 0: keep regulator fully operational when sleeping. 1: standby the regulator when sleeping. Allows enough power for chip static operation for memory retention. **Note:** See Table 3-12, "System Sleep Control Bit Encoding"	R/W	0h	nSYSRST

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	Ring oscillator output gate 0: keep ROSC ungated when sleeping. 1: gate the ROSC output when sleeping. **Note:** See Table 3-12, "System Sleep Control Bit Encoding"	R/W	0h	nSYSRST
0	Ring oscillator power down 0: keep ROSC operating when sleeping. 1: disable ROSC when sleeping. Clocks will start on wakeup, but there is a clock lock latency penalty. **Note:** See Table 3-12, "System Sleep Control Bit Encoding"	R/W	0h	nSYSRST

APPLICATION NOTE: Sample code for Sleep Debug = 1

```
do {
  wait; //processor sleep instruction
} while (sleep_debug == 1);
```

Note: The System Sleep States shown in TABLE 3-12: are determined by bits 2:0 in this register. The device only enters these sleep states after all the blocks have been commanded to sleep and they no longer require the 48 MHz Ring Oscillator; that is, if the sleep enable bits are set for all blocks or the Sleep All bit is set and no clocks are required.

TABLE 3-12: SYSTEM SLEEP CONTROL BIT ENCODING

D2	D1	D0	Wake Latency	Description
0	0	0	0	The Core regulator and the Ring Oscillator remain powered and running during sleep cycles (SYSTEM HEAVY SLEEP 1) (DEFAULT)
0	1	0	0	The Core regulator remains powered and the Ring oscillator is running but gated during sleep cycles (SYSTEM HEAVY SLEEP 2)
0	X	1	(Note 3-10)	The Core regulator remains powered and the Ring oscillator is powered down during sleep cycles (SYSTEM HEAVY SLEEP 3)
1	X	1	(Note 3-10)	The Core regulator is put into standby state and the Ring oscillator is powered down during sleep cycles. (SYSTEM DEEPEST SLEEP)

Note 3-10 The latency following a wake event for the SMBus and UART is 600us (typ). It is less than 10us for LPC, eSPI and PS2.

3.9.6 PROCESSOR CLOCK CONTROL REGISTER (PROC_CLK_CNTRL)

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	RESERVED	RES		
7:0	Processor Clock Divide Value 1: divide 48 MHz Ring Oscillator by 1 (i.e., 48 MHz). 4: divide 48 MHz Ring Oscillator by 4 (i.e., 12 MHz). 16: divide 48 MHz Ring Oscillator by 16 (i.e., 3 MHz). 48: divide 48 MHz Ring Oscillator by 48 (i.e., 1 MHz). No other values are supported.	R/W	4h	nSYSRST

3.9.7 EC SLEEP ENABLE 2 REGISTER (EC_SLP_EN2)

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	RESERVED	RES		
23	Reserved - Should be set to ‘1’	R/W	0h	nSYSRST
22	TIMER16_3 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment. See Note: on page 76.	R/W	0h	nSYSRST
21	TIMER16_2_Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment. See Note: on page 76.	R/W	0h	nSYSRST
20	BC-Link1 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
19	BC-Link0 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
18	LED2 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
17	LED1 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
16	LED0 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15	RESERVED	RES		
14	SMB2 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
13	SMB1 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
12	RTOS Timer Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
11	KEYSCAN Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
10	HTIMER Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
9	Quad SPI Master Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
8	RESERVED	RES		
7	RESERVED	RES		
6	PS2_1 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment. See Note 3-11.	R/W	0h	nSYSRST
5	PS2_0 Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment. See Note 3-11.	R/W	0h	nSYSRST
4	RESERVED	RES		
3	ADC Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
2	DAC1 Sleep Enable (DAC0_SLP_EN) 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment. **Note:** The effect of setting this bit is determined by DAC_VREF SLEEP_CONTROL programmed in the DAC Registers.	R/W	0h	nSYSRST

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	DAC0 Sleep Enable (DAC0_SLP_EN) 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment. **Note:** The effect of setting this bit is determined by DAC_VREF SLEEP_CONTROL programmed in the DAC Registers.	R/W	0h	nSYSRST
0	Reserved	R		

Note 3-11 The PS2 block will only sleep while the PS2 is disabled or in Rx mode with no traffic on the bus.

3.9.8 EC CLOCK REQUIRED 2 STATUS REGISTER (EC_CLK_REQ2_STS)

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:23	Reserved	R	0h	nSYSRST
22	TIMER16_3 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
21	TIMER16_2_Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
20	BC-Link 1Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
19	BC-Link 0 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
18	LED2 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
17	LED1 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
16	LED0 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
15	RESERVED	RES		

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
14	SMB2 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
13	SMB1 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
12	RTOS Timer Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
11	KEYSCAN Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
10	HTIMER Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
9	Quad SPI Master Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
8	RESERVED	RES		
7	RESERVED	RES		
6	PS2_1 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
5	PS2_0 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
4	RESERVED	RES		
3	ADC Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
2	DAC1 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
1	DAC0 Clock Required 0: block does NOT need clocks. 1: block requires clocks.	R	0h	nSYSRST
0	RESERVED	RES		

3.9.9 SLOW CLOCK CONTROL REGISTER (SLOW_CLK_CNTRL)

Offset	2Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:10	RESERVED	RES		
9:0	Slow Clock (100 kHz) Divide Value Configures the 100kHz_Clk. 0: Clock off n: divide by n. The Slow Clock value = 48 MHz Ring Oscillator / n. **Note:** The default setting is for 100 kHz.	R/W	1E0h	nSYSRST

3.9.10 OSCILLATOR ID REGISTER (CHIP_OSC_ID)

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	RESERVED	RES		
8	OSC_LOCK Oscillator Lock Status	R	0h	nSYSRST
7:0	Reserved	R	N/A	nSYSRST

3.9.11 PCR CHIP SUB-SYSTEM POWER RESET STATUS (CHIP_PWR_RST_STS)

Offset	34h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:12	RESERVED	RES		
11	PCICLK_ACTIVE This bit monitors the state of the PCI clock input. This status bit detects edges on the clock input but does not validate the frequency. 0: The 33MHz PCI clock input is not present. 1: The 33MHz PCI clock is present.	R	-	nSYSRST

Offset	34h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	32K_ACTIVE This bit monitors the state of the external 32K clock input. This status bit detects edges on the clock input but does not validate the frequency. 0: The external 32K clock input is not present. 1: The external 32K clock input is present.	R	-	nSYSRST
9	VBAT_LOW This bit is set if VBAT is below 2V when VTRGD is asserted. It is also set on the rising edge of VTRGD if a new coin was inserted while VTR was off.	R	-	nSYSRST
8:7	RESERVED	RES		
6	VTR reset status Indicates the status of nSYSRST. 0 = No reset occurred since the last time this bit was cleared. 1 = A reset occurred.	R/WC	1h	nSYSRST
5	VBAT reset status Indicates the status of VBAT_POR. 0 = No reset occurred while VTR was off or since the last time this bit was cleared. 1 = A reset occurred. **Note:** The bit will not clear if a write 1 is attempted at the same time that a VBAT_RST_N occurs. This ensures a reset event is never missed.	R/WC	-	nSYSRST
4	RESERVED	RES		
3	SIO_Reset Status Indicates the status of nSIO_RESET. 0 = reset active. 1 = reset not active.	R	xh	Note 3-12
2	VCC_PWRGD Status Indicates the status of VCC_PWRGD pin. 0 = VCC_PWRGD not asserted (Low). 1 = VCC_PWRGD asserted (High).	R	xh	Note 3-12
1:0	RESERVED	RES		

Note 3-12 This read-only status bit always reflects the current status of the event and is not affected by any Reset events.

3.9.12 HOST RESET ENABLE REGISTER (HOST_RST_EN)

Offset	3Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:17	RESERVED	RES		
16	8042EM Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
15:12	RESERVED	RES		
12	GLBL_CFG Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
11:2	RESERVED	RES		
1	UART 0 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
0	RESERVED	RES		

Note: If a block is configured such that it is to be reset when it goes to sleep, then registers within the block may not be writable when the block is asleep.

3.9.13 EC RESET ENABLE REGISTER (EC_RST_EN)

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	TIMER16_1 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
30	TIMER16_0 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
29	EC_REG_BANK Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
28:27	RESERVED	RES		
26	PWM7 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
25	PWM6 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
24	PWM5 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
23	PWM4 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
22	PWM3 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
21	PWM2 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
20	PWM1 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
19:12	RESERVED	RES		
11	TACH1 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
10	SMB0 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
9	WDT Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
8	RESERVED	RES		
7	TFDP Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
6	DMA Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
5	PMC Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
4	PWM0 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	RESERVED	RES		
2	TACH0 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
1	PECI Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
0	INT Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST

Note: If a block is configured such that it is to be reset when it goes to sleep, then registers within the block may not be writable when the block is asleep.

3.9.14 EC RESET ENABLE 2 REGISTER (EC_RST_EN2)

Offset	44h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:23	RESERVED	RES		
22	TIMER16_3 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
21	TIMER16_2_Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
20	BC-Link 1 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
19	BC-Link 0 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
18	LED2 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
17	LED1 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST

Offset	44h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
16	LED0 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
15	SMB3 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
14	SMB2 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
13	SMB1 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
12	RESERVED	RES		
11	KEYSCAN Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
10	HTIMER Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
9	Quad SPI Master Sleep Enable 0: block is free to use clocks as necessary. 1: block is commanded to sleep at next available moment.	R/W	0h	nSYSRST
8:7	RESERVED	RES		
6	PS2_1 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
5	PS2_0 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
4	RESERVED	RES		
3	ADC Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	nSYSRST
2:0	RESERVED	RES		

Note: If a block is configured such that it is to be reset when it goes to sleep, then registers within the block may not be writable when the block is asleep.

3.9.15 POWER RESET CONTROL (PWR_RST_CTRL) REGISTER

Offset	48h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	RESERVED	RES		
1	Host_Reset_Select This bit determines the platform reset signal that will be used to assert nSIO_RESET. See FIGURE 3-2: Resets Diagram (MEC140x/1X) on page 70. 0 = LRESET# pin generates internal Platform Reset 1 = eSPI Platform Reset (eSPI_PLTRST#)	R/W	0h	nSYSRST
0	iRESET_OUT The iRESET_OUT bit is used by firmware to control the internal nSIO_RESET signal function and the external nRESET_OUT pin. The external pin nRESET_OUT is always driven by nSIO_RESET. Firmware can program the state of iRESET_OUT except when the VCC_PWRGD pin is not asserted (‘0’), in which case iRESET_OUT is ‘don’t care’ and nSIO_RESET is asserted (‘0’) (TABLE 3-13:). The internal nSIO_RESET signal is asserted when iRESET_OUT is asserted even if the nRESET_OUT pin is configured as an alternate function. The iRESET_OUT bit must be cleared to take the Host accessible peripherals out of reset. See Section 1.5, "MEC140x Internal Address Spaces," on page 10 for host accessible peripherals.	R/W	1h	nSYSRST

TABLE 3-13: iRESET_OUT BIT BEHAVIOR

nSYSRST	VCC_PWRGD	PCI_RESET#	iRESET_OUT	nSIO_RESET & nRESET_OUT
0	X	X	X	0 (ASSERTED)
1	0	X	X	0 (ASSERTED)
	1	0	X	0 (ASSERTED)
		1	1	0 (ASSERTED)
			0	1 (NOT ASSERTED)

4.0 LPC INTERFACE

4.1 Introduction

The Intel® Low Pin Count (LPC) Interface is the LPC Interface used by the system host to configure the chip and communicate with the logical devices implemented in the design through a series of read/write registers. Register access is accomplished through the LPC transfer cycles defined in Table 4-5, "LPC Cycle Types Supported".

The Logical Devices implemented in the design are identified in Table 4-15, “I/O Base Address Registers,” on page 117. The Base Address Registers allow any logical device’s runtime registers to be relocated in LPC I/O space. All chip configuration registers for the device are accessed indirectly through the LPC I/O Configuration Port (see Section 4.8.3, "Configuration Port," on page 105).

4.2 References

- Intel® Low Pin Count (LPC) Interface Specification, v1.1
- PCI Local Bus Specification, Rev. 2.2
- Serial IRQ Specification for PCI Systems Version 6.0.
- PCI Mobile Design Guide Rev 1.0

4.3 Terminology

This table defines specialized terms localized to this feature.

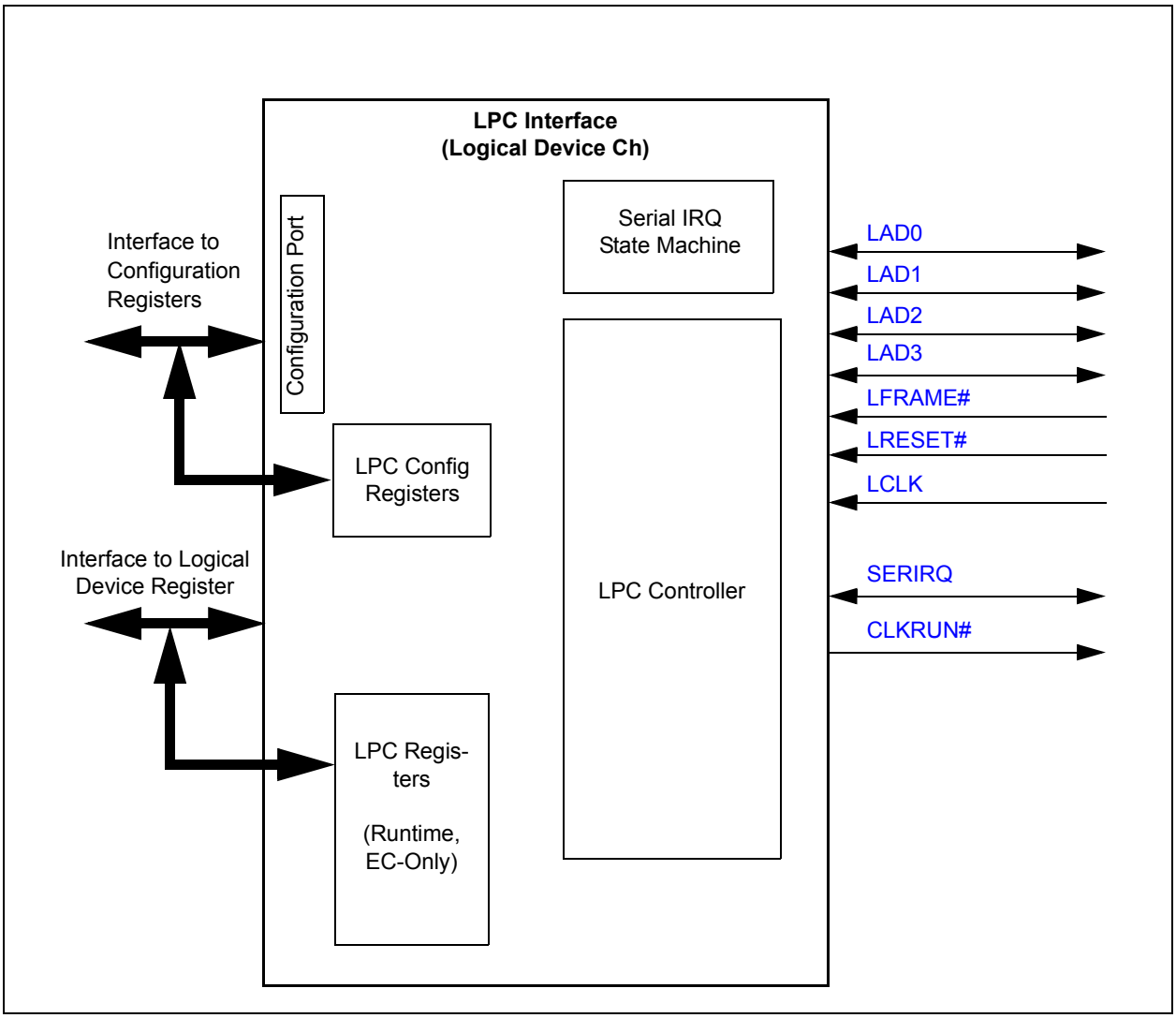
TABLE 4-1: TERMINOLOGY

Term	Definition
System Host	Refers to the external CPU that communicates with this device via the LPC Interface.
Logical Devices	Logical Devices are LPC accessible features that are allocated a Base Address and range in LPC I/O address space
Runtime Register	Runtime Registers are register that are directly I/O accessible by the System Host via the LPC interface. These registers are defined in Section 4.10, "Runtime Registers," on page 120.
Configuration Registers	Registers that are only accessible in CONFIG_MODE. These registers are defined in Section 4.9, "LPC Configuration Registers," on page 112.
EC_Only Registers	Registers that are not accessible by the System Host. They are only accessible by an internal embedded controller. These registers are defined in Section 4.11, "EC-Only Registers," on page 121.

4.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 4-1: BLOCK DIAGRAM OF LPC INTERFACE CONTROLLER WITH CLKRUN# SUPPORT



4.4.1 SIGNAL DESCRIPTION

TABLE 4-2: SIGNAL DESCRIPTION

Name	Direction	Description
LAD0	Input/Output	Bit[0] of the LPC multiplexed command, address, and data bus.
LAD1	Input/Output	Bit[1] of the LPC multiplexed command, address, and data bus.

TABLE 4-2: SIGNAL DESCRIPTION (CONTINUED)

Name	Direction	Description
LAD2	Input/Output	Bit[2] of the LPC multiplexed command, address, and data bus.
LAD3	Input/Output	Bit[3] of the LPC multiplexed command, address, and data bus.
LFRAME#	Input	Active low signal indicates start of new cycle and termination of broken cycle.
LRESET#	Input	Active low signal used as LPC Interface Reset. Same as PCI Reset on host. **Note:** LRESET# is typically connected to the host PCI RESET (PCIRST#) signal.
LCLK	Input	PCI clock input (PCI_CLK)
SERIRQ	Input/Output	Serial IRQ pin used with the LCLK signal to transfer interrupts to the host.
CLKRUN#	Open-Drain Output	Clock Control for LCLK
LPCPD#	Input	Power Down: Indicates that the device should prepare for power to be removed from the LPC I/F.

4.4.2 REGISTER INTERFACES

The registers defined for the LPC Interface block are accessible by the various hosts as indicated in Section 4.9, "LPC Configuration Registers", Section 4.11, "EC-Only Registers"and Section 4.10, "Runtime Registers".

4.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

4.5.1 POWER DOMAINS

Name	Description
VTR	The LPC Interface block and registers are powered by VTR.

4.5.2 CLOCK INPUTS

Name	Description
LCLK	This LPC Interface has a single clock input, called LCLK.

Note: The PCI_CLK input to LCLK can run at 24MHz or 33MHz. When the PCI_CLK input is 24MHz the Handshake bit in the EC Clock Control Register must be set to a '1' to capture LPC transactions properly. See Section 4.11.4, "EC Clock Control Register," on page 123.

4.5.3 RESETS

Name	Description
nSYSRST	Power on Reset to the block. This signal resets all the register and logic in this block to its default state.
nSIO_RESET	This signal is used to indicate when the main power rail in the system is reset. The LPC interface is operational when main power is present. This signal is used to reset selected registers as defined in the Register Interfaces descriptions.
LRESET#	The LRESET# signal comes from the LPC pin interface. This signal is defined in the Intel® Low Pin Count (LPC) Interface Specification, v1.1.

The following table defines the effective reset state that result from the combination of these three reset signals.

TABLE 4-3: LPC Interface BLOCK RESET STATES

nSYSRST (Note 4-2)	LRESET# (Note 4-1, Note 4-4)	nSIO_RESET (Note 4-3)	Reset State
Asserted	X	X	Resets all registers and logic
Deasserted	Asserted	X	Resets selected registers and logic
	Deasserted	Asserted	Resets selected registers
		Deasserted	Nothing is in Reset

Note 4-1 The EC can determine the state of the LRESET# input using registers in LPC Bus Monitor Register on page 122.

Note 4-2 nSYSRST is asserted when VTR is turned off and is released after VTR is turned on. The nSYSRST will be released before the System Host is expected to attempt communication over the LPC Interface.

Note 4-3 See the individual register descriptions to determine which registers are effected by nSIO_RESET.

Note 4-4 The LPC Interface will be ready to receive a new transaction when LRESET# is deasserted. See the individual register descriptions to determine which registers are effected by this reset.

In system, the LPC Interface is required to be operational in ACPI Sleep States S0 - S2. When the system enters Sleep States S3 - S5 the LPC interface must tristate its outputs. The following table shows the behavior of LPC output and input/output signals under reset conditions.

TABLE 4-4: LPC INTERFACE SIGNALS BEHAVIOR ON RESET

Pins	nSYSRST	nSIO_RESET	LPCPD#	LRESET# Asserted
LAD[3:0]	Tri-state	Tri-state	Tri-state	Tri-State
SERIRQ	Tri-state	Tri-state	Tri-state	Tri-State
CLKRUN#	Tri-state	Tri-state	Tri-state	Tri-State

4.6 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
LPC_WAKE	This signal is asserted when the LPC interface detects LPC traffic. If enabled, it may be used to wake the 48 MHz Ring Oscillator when the chip is in a sleep state. **Note:** This interrupt is grouped with other Wake-Only events in GIRQQ16 and GIRQ22.
LPC_INTERNAL_ERR	The LPC_INTERNAL_ERR event is sourced by bit D0 of the Host Bus Error Register.

4.7 Low Power Modes

The LPC Controller may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

The LPC Controller will deassert its clock required signal when it is commanded to sleep and it is not processing an LPC transaction, thereby allowing the 48 MHz Ring Oscillator to be turned off. If a subsequent transaction is detected on the LPC interface, the LPC Controller will assert the LPC_WAKE signal to the JTVIC Controller. If enabled, this event will wake the 48 MHz Ring Oscillator.

4.8 Description

This LPC Controller is compliant with the Intel® Low Pin Count (LPC) Interface Specification, v1.1. Section 4.8.1, "LPC Controller Description" further clarifies which LPC Interface features have been implemented and qualifies any system specific requirements.

The LPC Controller claims only LPC transactions targeted for one of its peripherals. Section 4.8.2, on page 101, describes the mechanism for Claiming and Forwarding Transactions for Supported LPC Cycles. LPC transactions may be used to configure the chip and to access registers during operation. The mechanism to configure the chip is described in Section 4.8.3, "Configuration Port," on page 105.

The LPC memory cycles may also be used to access the Base Address Registers of certain devices as well as internal SRAM.

Once configured, the LPC peripherals implemented as logical devices on chip may use the SERIRQ to notify the host of an event. See Section 4.8.4, "Serial IRQs," on page 107.

4.8.1 LPC CONTROLLER DESCRIPTION

The following sections qualify the LPC features implemented according to the LPC Specification.

4.8.1.1 Cycle Types Supported

The following cycle types are supported by the LPC Interface Controller. All other cycles that it does not support are ignored.

TABLE 4-5: LPC CYCLE TYPES SUPPORTED

Cycle Type	Transfer Size
I/O Read	1 byte
I/O Write	1 byte
Memory Read	1 byte
Memory Write	1 byte

When the LPC Controller detects a transaction targetted for this device it will claims and forward that transaction as defined in Section 4.8.2, "Claiming and Forwarding Transactions for Supported LPC Cycles," on page 101.

LPC I/O Cycles

The system host may use LPC I/O cycles to read/write the I/O mapped configuration and runtime registers implemented in this device. See the Intel® Low Pin Count (LPC) Interface Specification, v1.1, Section 5.2 for definition of LPC I/O Cycles.

LPC Memory Cycles

The system host may use LPC memory cycles to access memory mapped registers and internal RAMs implemented in this device. See the Intel® Low Pin Count (LPC) Interface Specification, v1.1, Section 5.1 for definition of LPC Memory Cycles.

4.8.1.2 LAD[3:0] Fields

The LAD[3:0] signals support multiple fields for each protocol as defined in section 4.2.1 LAD[3:0] of the Intel® Low Pin Count (LPC) Interface Specification, v1.1. The following sections further qualify the fields supported.

Wait SYNCs on LPC

LPC transactions that access registers located on the device will require a minimum of two wait SYNCs on the LPC bus. The number of SYNCs may be larger if the internal bus is in use by the embedded controller, of if the data referenced by the host is not present in a register. The device always uses Long Wait SYNCs, rather than Short Wait SYNCs, when responding to an LPC bus request.

Note:	All LPC transactions are synchronized to the LCLK and will complete with a maximum of 8 wait states, unless otherwise noted.

ERROR SYNCs on LPC

The device does not issue ERROR SYNC cycles.

4.8.1.3 LPC Clock Run and LPC Power Down Behavior

Using LPCPD#

The device tolerates the LPCPD# signal going active and then inactive again without LRESET# going active. This is a requirement for notebook power management functions.

The Intel® Low Pin Count (LPC) Interface Specification, v1.1, Section 8.2 states that “After LPCPD# is de-asserted, the LPC interface may be reset dependent upon the characteristics of system reset signal connected to LRESET#.” This text must be qualified for mobile systems where it is possible that when exiting a "light" sleep state (ACPI S1, APM POS), LPCPD# may be asserted but the LPC Bus power may not be removed, in which case LRESET# will not occur. When exiting a "deeper" sleep state (ACPI S3-S5, APM STR, STD, soft-off), LRESET# will occur.

The LPCPD# pin is implemented as a “local” powergood for the LPC bus in the device. It is not to be used as a global powergood for the chip. It is used to minimize the LPC power dissipation.

Prior to going to a low-power state, the system asserts the LPCPD# signal. LPCPD# goes active at least 30 microseconds prior to the LCLK signal stopping low and power being shut to the other LPC interface signals. Upon recognizing LPCPD# active, there are no further transactions on the LPC interface.

Using CLKRUN#

CLKRUN# is used to indicate the status of LCLK as well as to request that a stopped clock be started. See FIGURE 4-2: CLKRUN# System Implementation Example on page 100, an example of a typical system implementation using CLKRUN#.

LCLK Run Support can be enabled and disabled via SIRQ_MODE as shown in Table 4-6, "LPC Controller CLKRUN# Function". When the SIRQ_MODE is ‘0,’ Serial IRQs are disabled, the CLKRUN# pin is disabled, and the affects of Interrupt requests on CLKRUN# are ignored. When the SIRQ_MODE is ‘1,’ Serial IRQs are enabled, the CLKRUN# pin is enabled, and the CLKRUN# support related to Interrupts requests as described in the section below is enabled.

The CLKRUN# pin is an open drain output and input. Refer to the PCI Mobile Design Guide Rev 1.0 for a description of the CLKRUN# function. If CLKRUN# is sampled “high”, LCLK is stopped or stopping. If CLKRUN# is sampled “low”, LCLK is starting or started (running).

CLKRUN# Support for Serial IRQ Cycle

If a logical device asserts or de-asserts an interrupt and CLKRUN# is sampled “high”, the LPC Controller can request the restoration of the clock by asserting the CLKRUN# signal asynchronously (TABLE 4-6:). The LPC Controller holds CLKRUN# low until it detects two rising edges of the clock. After the second clock edge, the controller must disable the open drain driver (FIGURE 4-3:).

The LPC Controller must not assert CLKRUN# if it is already driven low by the central resource; i.e., the PCI CLOCK GENERATOR in FIGURE 4-2:. The controller will not assert CLKRUN# under any conditions if the Serial IRQs are disabled.

The LPC Controller must not assert CLKRUN# unless the line has been de-asserted for two successive clocks; i.e., before the clock was stopped (FIGURE 4-3:).

The LPC Controller will not assert CLKRUN# if it is already driven low by the central resource; i.e., the PCI CLOCK GENERATOR. The LPC Controller also will not assert CLKRUN# unless the signal has been de-asserted for two successive clocks; i.e., before the clock was stopped.

TABLE 4-6: LPC CONTROLLER CLKRUN# FUNCTION

SIRQ_MODE	Internal Interrupt Or DMA Request	CLKRUN#	Action
0	X	X	None
1	NO CHANGE	X	None
	CHANGE(Note 4-6)	0	None
		1	Assert CLKRUN#

Note 4-5 SIRQ_MODE is defined in Section 4.8.4.1, "Enabling SERIRQ Function," on page 107.

Note 4-6 “Change” means either-edge change on any or all parallel IRQs routed to the Serial IRQ block. The “change” detection logic must run asynchronously to LCLK and regardless of the Serial IRQ mode; i.e., “continuous” or “quiet”.

FIGURE 4-2: CLKRUN# SYSTEM IMPLEMENTATION EXAMPLE

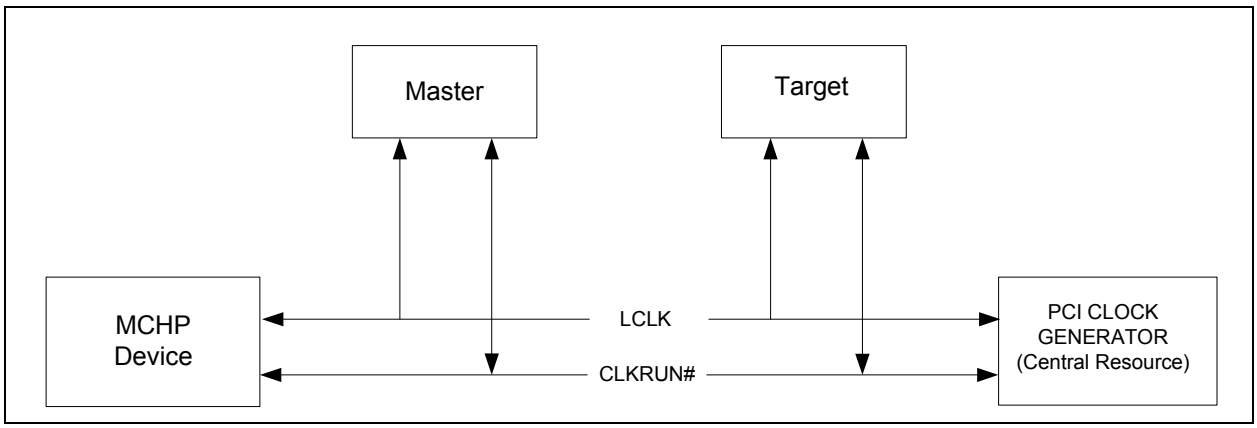
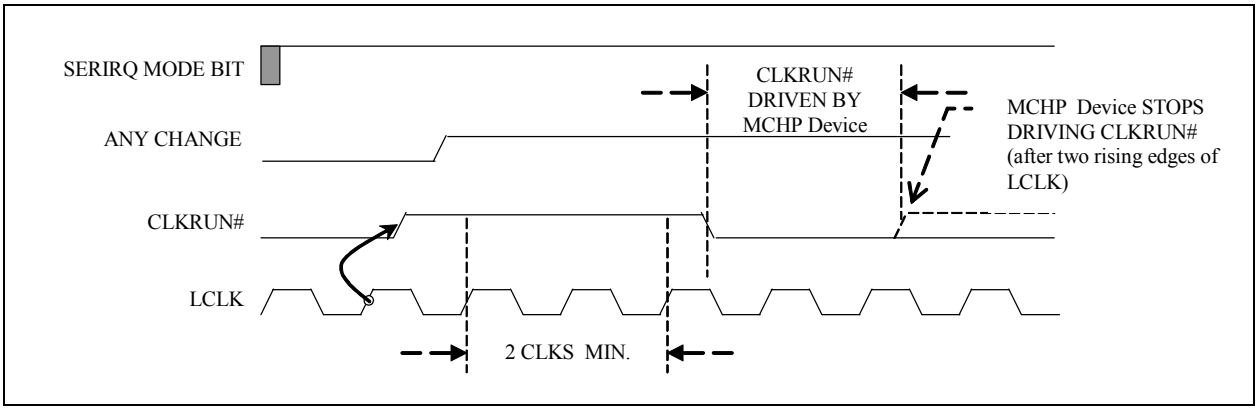


FIGURE 4-3: CLOCK START ILLUSTRATION



Note 1: The signal “ANY CHANGE” is the same as “CHANGE/ASSERTION” in TABLE 4-6:.

2: The LPC Controller must continually monitor the state of CLKRUN# to maintain LCLK until an active “any IRQ change” condition has been transferred to the host in a Serial IRQ cycle or “any DRQ assertion” condition has been transferred to the host in a DMA cycle. For example, if “any IRQ change or DRQ assertion” is asserted before CLKRUN# is de-asserted (not shown in FIGURE 4-3:), the controller must assert CLKRUN# as needed until the Serial IRQ cycle or DMA cycle has completed.

4.8.2 CLAIMING AND FORWARDING TRANSACTIONS FOR SUPPORTED LPC CYCLES

The following sections define how the LPC Controller determines if a cycle is targetted for one of the chip’s logical devices and how that transaction is then forwarded to that logical device. The following sections include:

- Section 4.8.2.1, "I/O Transactions," on page 101
- Section 4.8.2.2, "Device Memory Transactions," on page 104

4.8.2.1 I/O Transactions

The system host will generate I/O commands to communicate with I/O peripherals, such as Keyboard Controller, UART, etc. The LPC Controller will claim only I/O transactions targeted to it and it will ignore all others. The following sections describe how I/O transactions are claimed and forwarded to access the Runtime and Configuration registers.

CLAIMING LPC I/O TRANSACTIONS

Each host I/O accessible block (i.e., logical device) has an associated I/O Base Address register. The format of this register is defined in Section 4.9.3, "I/O Base Address Registers (IO_BARs)," on page 115. If the VALID bit is set in the logical device’s BAR register the LPC interface will claim all I/O addresses that match the unmasked portion of the programmed LPC Host Address using the following equation.

(LPC Address & ~BAR.MASK) == (BAR.LPC_Address & ~BAR.MASK) && (BAR.Valid == 1)

Masked bits are treated as don’t care in the address matching decoder.

Note:	The LPC Controller’s Base Address register is used to define the Base I/O Address of the Configuration Port.

FORWARDING I/O TRANSACTIONS

Once an LPC Address is claimed for a specific logical device, the 8 LSbs of the I/O Address are used as the offset from the hard-coded logical device’s Runtime Registers Base Address located in the EC/Host Address space (i.e., F_0000h to F_FFFFh). This allows each Host I/O Accessible Block the ability to map up to 256 contiguous bytes into I/O space.

EC/Host Address = Logical Device Runtime Register Base Address[31:0] + (LPC I/O Address[6:0] & BAR.MASK)

Note:	The Runtime Registers are always located on even 1k byte boundaries in the internal EC/Host Address space.

TABLE 4-7: LPC RUNTIME (CONFIGURATION PORT) ADDRESSING

Configuration Port LPC I/O Address	EC/Host Address	Register
002Eh	F_3000h	INDEX
002Fh	F_3001h	DATA

Note: The Logical Device number for the matching device is located in the Frame field of the BAR. The Frame field is mapped to bits [15:10] of the EC/Host Address space. In this example bits[15:10] = 00_1100 = Ch.

The system host will use I/O transactions to access the Configuration and Runtime registers. To access the Runtime registers, the host must configure the I/O Base Address Registers (IO_BARs), which are accessible via the Configuration Port first. The Configuration Port, Logical Device Ch, is located at the Base I/O Address programmed in the BAR Configuration register located at offset 60h.

For illustration purposes only, lets examine two types of logical devices (these may or may not reside in this design).

EXAMPLE 4-1: KEYBOARD CONTROLLER

The Keyboard Controller (8042 Interface) Base Address Register has 60h in the LPC Address field, the Frame field is 01h, and the MASK field is 04h. Because of the single '1b' bit in MASK, the BAR will match LPC I/O patterns in the form '0000_0000_0110_0x00b', so both 60h and 64h will be matched and claimed by the LPC Controller.

EXAMPLE 4-2: 16550 UART

If a standard 16550 UART was located at LPC I/O address 238h, then the UART Receive buffer would appear at address 238h and the Line Status register at 23Dh. If the BAR for the UART was set to 0238_8047h, then the UART will be matched at I/O address 238h.

The following table illustrates the I/O Address Mapping for each logical device implemented in the MEC140x/1x.

TABLE 4-8: LPC I/O REGISTER MAP

Logical Device	BAR LPC Host Address	Example BAR LPC Host Address	LPC Address Mask	Offsets Claimed	Register Name
LPC Interface (Configuration Port)	2 Byte Boundary	002Eh	1	BAR+0	INDEX
				+1	DATA
EMI 0	16 Byte Boundary	0060h	F	BAR+0	Host-to-EC Mailbox
				+1	EC-to-Host Mailbox
				+2	EC Address LSB
				+3	EC Address MSB
				+4	EC Data Byte 0
				+5	EC Data Byte 1
				+6	EC Data Byte 2
				+7	EC Data Byte 3
				+8	Interrupt Source LSB
				+9	Interrupt Source MSB
				+A	Interrupt Mask LSB
				+B	Interrupt Mask MSB
				+C	Application ID

TABLE 4-8: LPC I/O REGISTER MAP (CONTINUED)

Logical Device	BAR LPC Host Address	Example BAR LPC Host Address	LPC Address Mask	Offsets Claimed	Register Name
8042 Emulated Keyboard Controller	Byte Boundary	0060h	4	BAR+0	Write: WRITE_DATA Read: READ_DATA
				+4	Write: WRITE_CMD Read: STATUS
ACPI EC0	Byte Boundary	0062h	4	BAR+0	ACPI_OS_DATA_BYTE_0
				+4	Write: ACPI_OS_COMMAND Read: OS STATUS OS
ACPI EC1	8 Byte Boundary	0070h	7	BAR+0	ACPI_OS_DATA_BYTE_0
				+1	ACPI_OS_DATA_BYTE_1
				+2	ACPI_OS_DATA_BYTE_2
				+3	ACPI_OS_DATA_BYTE_3
				+4	Write: ACPI_OS_COMMAND Read: OS STATUS OS
				+5	OS Byte Control
				+6	Reserved
				+7	Reserved
ACPI PM1	8 Byte Boundary	0078h	7	BAR+0	Power Management 1 Status 1
				+1	Power Management 1 Status 2
				+2	Power Management 1 Enable 1
				+3	Power Management 1 Enable 2
				+4	Power Management 1 Control 1
				+5	Power Management 1 Control 2
				+6	Power Management 2 Control 1
				+7	Power Management 2 Control 2
Legacy Port92/GateA20	Any I/O Byte Address	0092h	0	BAR+0	Port 92
UART 0	8 Byte Boundary	03F0h	7	BAR+0	Write (DLAB=0): Transmit Buffer Read (DLAB=0): Receive Buffer R/W (DLAB=1): Programmable BAUD Rate Generator LSB
				+1	R/W (DLAB=0): Interrupt Enable Register R/W (DLAB=1): Programmable BAUD Rate Generator MSB
				+2	Write: FIFO Control Read: Interrupt Identification
				+3	Line Control
				+4	Modem Control
				+5	Line Status
				+6	Modem Status
				+7	Scratchpad Register
Mailbox Interface	2 Byte Boundary	0100h	1	BAR+0	MBX_INDEX
				+1	MBX_DATA

TABLE 4-8: LPC I/O REGISTER MAP (CONTINUED)

Logical Device	BAR LPC Host Address	Example BAR LPC Host Address	LPC Address Mask	Offsets Claimed	Register Name
ACPI EC2	8 Byte Boundary	0030h	7	BAR+0	ACPI_OS_DATA_BYTE_0
				+1	ACPI_OS_DATA_BYTE_1
				+2	ACPI_OS_DATA_BYTE_2
				+3	ACPI_OS_DATA_BYTE_3
				+4	Write: ACPI_OS_COMMAND Read: OS STATUS OS
				+5	OS Byte Control
				+6	Reserved
				+7	Reserved
ACPI EC3	8 Byte Boundary	0038h	7	BAR+0	ACPI_OS_DATA_BYTE_0
				+1	ACPI_OS_DATA_BYTE_1
				+2	ACPI_OS_DATA_BYTE_2
				+3	ACPI_OS_DATA_BYTE_3
				+4	Write: ACPI_OS_COMMAND Read: OS STATUS OS
				+5	OS Byte Control
				+6	Reserved
				+7	Reserved
Port 80 BIOS Debug Port 0	Any I/O Byte Address	0080h	0	BAR+0	Host Data
Port 80 BIOS Debug Port 1	Any I/O Byte Address	0081h	0	BAR+0	Host Data

4.8.2.2 Device Memory Transactions

LPC Memory cycles are single byte read or writes that occur in a 32-bit address range. The LPC block will claim a memory transaction that is targeted for one of these logical devices. A Device Memory Base Address Register has been implemented for the logical devices listed in Table 4-16, “Device Memory Base Address Registers,” on page 119

On every LPC bus Memory access all Base Address Registers are checked in parallel and if any matches the LPC memory address the LPC Interface claims the bus cycle. The memory address is claimed as described in I/O Transactions on page 101 except that the LPC memory cycle address is 32 bits instead of the 16 bit I/O cycle address.

Software should insure that no two BARs map the same LPC memory address. If two BARs do map to the same address, the BAR_CONFLICT bit in the Host Bus Error Register is set when an LPC access targeting the BAR Conflict address. An EC interrupt can be generated.

Each Device Memory BAR is 48 bits wide. The format of each Device Memory BAR is summarized in Device Memory Base Address Register Format. An LPC memory request is translated by the Device Memory BAR into an 8-bit read or write transaction on the AHB bus. The 32-bit LPC memory address is translated into a 32-bit AHB address.

The Base Address Register Table is itself part of the AHB address space. It resides in the Configuration quadrant of Logical Device Ch, the LPC Interface.

4.8.2.3 SRAM Memory Transactions

In addition to mapping LPC Memory transactions into Logical Devices, Memory transactions can be mapped into internal address space, as configured by the SRAM Memory BARs. LPC Memory cycles are single byte read or writes that occur in a 32-bit address range. The LPC block will claim LPC memory cycles that match the programmed SRAM Memory BAR Register if the VALID in the SRAM Memory BAR Configuration is set to 1. No memory cycles will be claimed if this bit is cleared.

The LPC interface can claim up to a 4 kB block of memory addresses and map them to the internal address space. The location of the block of memory in the 32-bit internal space, as well as access to it, is controlled by the EC, using the SRAM Memory Host Configuration Register.

The block of memory in the internal 32-bit address space must start on any size-byte address boundary. For example, if the memory is 4k bytes than it is only relocatable on 4k byte boundaries.

CLAIMING LPC MEMORY TRANSACTIONS

A Base Address Register will match an LPC Memory address, and thus the device will claim the LPC bus cycle, if the following relation holds:

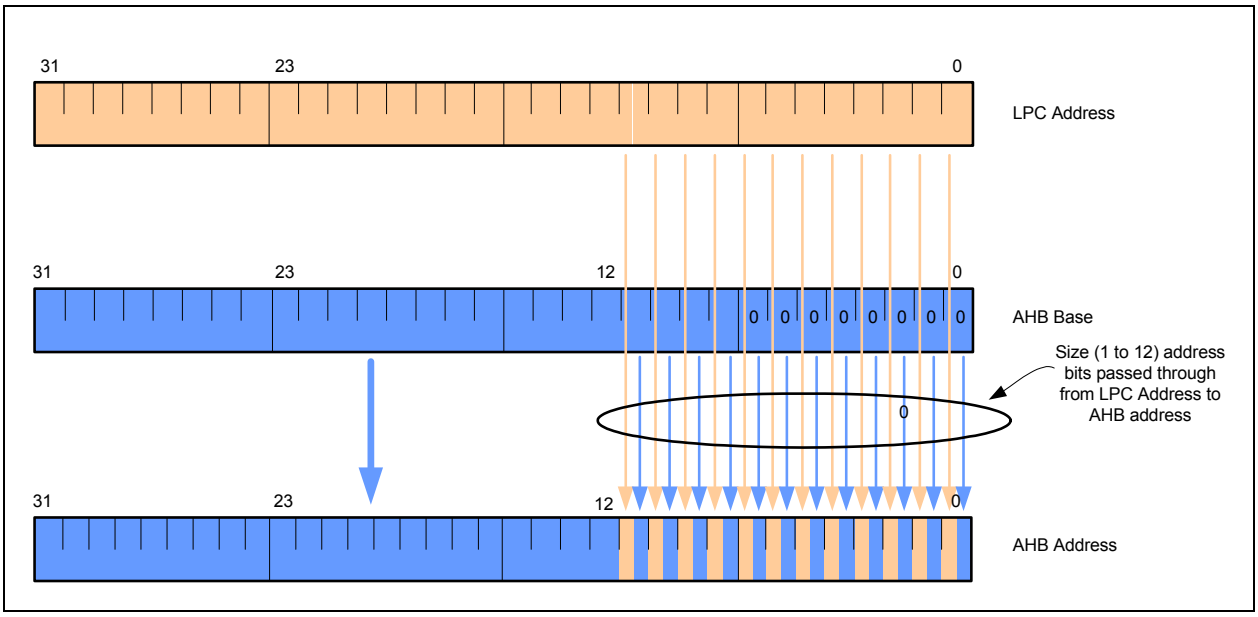
bit (LPC Address & ~(BAR.2^{SIZE}-1) == (BAR.Host_Address & ~(BAR.2^{SIZE}-1)) && (BAR.Valid == 1)

If the BAR matches, the LPC cycle will be claimed by the device. The LPC request will be translated to an AHB address according to the following formula:

AHB Address = (BAR.AHB_Base & ~(BAR.2^{SIZE}-1)) | (LPC_Address & (BAR.2^{SIZE}-1))

The mapping is also illustrated in FIGURE 4-4:

FIGURE 4-4: AHB ADDRESS BIT MAPPING



FORWARDING SRAM MEMORY TRANSACTIONS

The LPC interface can claim up to a 4 kB block of memory addresses and map them to the internal address space.

The firmware programs the base address of the internal memory space in SRAM Memory Host Configuration Register, which is mapped to the LPC memory address programmed by the host in the SRAM Memory BAR register. The firmware also programs the size of the memory to be accessed. The LPC block uses the size field to determine which memory addresses to claim (see Section , "Claiming LPC Memory Transactions," on page 105), as well as to prevent reading/writing an unmapped internal memory location.

4.8.3 CONFIGURATION PORT

The LPC Host can access the Chip's Configuration Registers through the Configuration Port when CONFIG MODE is enabled. The device defaults to CONFIG MODE being disabled.

Note:	The data read from the Configuration Port Data register is undefined when CONFIG MODE is not enabled.

The Configuration Port is composed of an INDEX and DATA Register. The INDEX register is used as an address pointer to an 8-bit configuration register and the DATA register is used to read or write the data value from the indexed configuration register. Once CONFIG MODE is enabled, reading the Configuration Port Data register will return the data value that is in the indexed Configuration Register.

If no value was written to the INDEX register, reading the Data Register in the Configuration Port will return the value in Configuration Address location 00h (default).

TABLE 4-9: CONFIGURATION PORT

Default I/O Address (Note 4-7)	Type	Register Name	Relative Address	Default Value	Notes
002Eh	Read / Write	INDEX	Configuration Port's Base Address + 0	00h	Note 4-7
002Fh	Read / Write	DATA	Configuration Port's Base Address + 1	00h	

Note 4-7 The default Base I/O Address of the Configuration Port can be relocated by programming the BAR register for Logical Device Ch (LPC/Configuration Port) at offset 60h. The Relative Address shows the general case for determining the I/O address for each register.

4.8.3.1 Enable CONFIG MODE

The INDEX and DATA registers are effective only when the chip is in CONFIG MODE. CONFIG MODE is enabled when the Config Entry Key is successfully written to the I/O address of the INDEX register of the CONFIG PORT while the CONFIG MODE is disabled (see Section 4.8.3.2, "Disable CONFIG MODE").

Config Entry Key = < 55h>

4.8.3.2 Disable CONFIG MODE

CONFIG MODE defaults to disabled on a nSYSRST, nSIO_RESET, and when LRESET# is asserted. CONFIG MODE is also disabled when the following Config Exit Key is successfully written to the I/O address of the INDEX PORT of the CONFIG PORT while CONFIG MODE is enabled.

Config Exit Key = < AAh>

4.8.3.3 Configuration Sequence Example

To program the configuration registers, the following sequence must be followed:

1. Enable Configuration State
2. Program the Configuration Registers
3. Disable Configuration State.

The following is an example of a configuration program in Intel 8086 assembly language.

```
;----------------------------.
; ENABLE CONFIGURATION STATE
;----------------------------'
MOV     DX,CONFIG_PORT_BASE_ADDRESS
MOV     AX,055H; Config Entry Key
OUT     DX,AL
;----------------------------.
; CONFIGURE BASE ADDRESS,    |
; LOGICAL DEVICE 8           |
;----------------------------'
MOV     DX,CONFIG_PORT_BASE_ADDRESS
MOV     AL,07H
OUT     DX,AL; Point to LD# Config Reg
MOV     DX,CONFIG_PORT_BASE_ADDRESS+1
MOV     AL, 08H
OUT DX,AL; Point to Logical Device 8
```

```
;
MOV     DX,CONFIG_PORT_BASE_ADDRESS
MOV     AL,60H
OUT     DX,AL   ; Point to BASE ADDRESS REGISTER
MOV     DX,CONFIG_PORT_BASE_ADDRESS+1
MOV     AL,02H
OUT     DX,AL   ; Update BASE ADDRESS REGISTER
;----------------------------.
; DISABLE CONFIGURATION STATE
;----------------------------'
MOV     DX,CONFIG_PORT_BASE_ADDRESS
MOV     AX,0AAH; Config Exit Key
OUT     DX,AL.
```

4.8.4 SERIAL IRQS

The device supports the serial interrupt scheme, which is adopted by several companies, to transmit interrupt information to the system. The serial interrupt scheme adheres to the Serial IRQ Specification for PCI Systems Version 6.0..

4.8.4.1 Enabling SERIRQ Function

Each Serial IRQ channel defaults to disabled. To enable a Serial IRQ channel the host must program the Serial IRQ Configuration Registers on page 113.

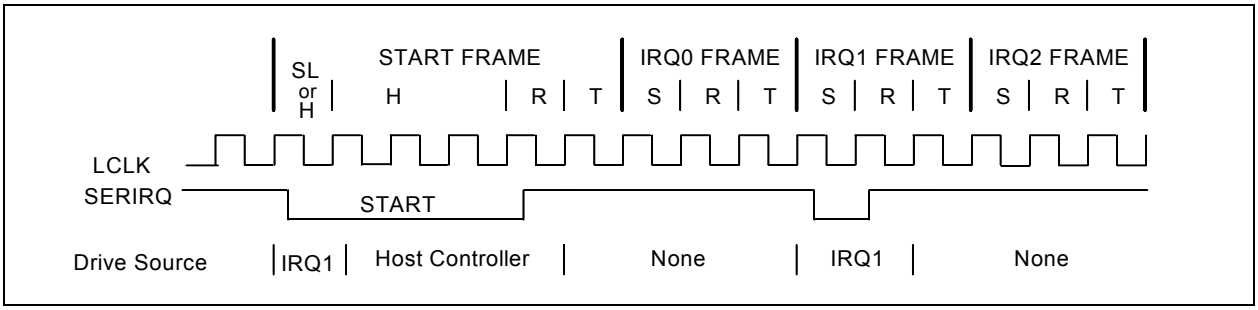
4.8.4.2 TIMING DIAGRAMS for SERIRQ CYCLE

LCLK = LCLK pin

SERIRQ = Serial IRQ pin

Start Frame timing with source sampled a low pulse on IRQ1

FIGURE 4-5: SERIAL INTERRUPTS WAVEFORM “START FRAME”

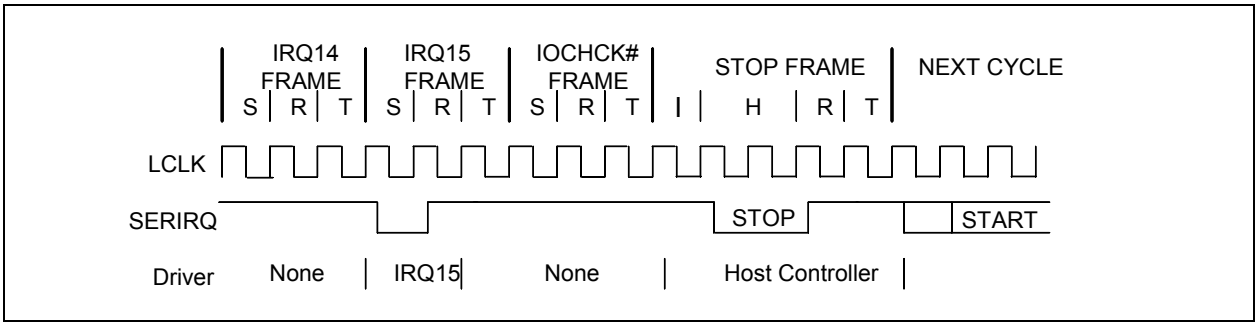


H=Host Control SL=Slave Control R=Recovery T=Turn-around S=Sample

Start Frame pulse can be 4-8 clocks wide.

Stop Frame Timing with Host using 17 SERIRQ sampling period

FIGURE 4-6: SERIAL INTERRUPT WAVEFORM "STOP FRAME"



H=Host Control R=Recovery T=Turn-around S=Sample I= Idle

Stop pulse is two clocks wide for Quiet mode, three clocks wide for Continuous mode.

There may be none, one, or more Idle states during the Stop Frame.

The next SERIRQ cycle's Start Frame pulse may or may not start immediately after the turn-around clock of the Stop Frame.

4.8.4.3 SERIRQ Cycle Control

SERIRQ START FRAME

There are two modes of operation for the SERIRQ Start Frame.

Quiet (Active) Mode

Any device may initiate a Start Frame by driving the SERIRQ low for one clock, while the SERIRQ is Idle. After driving low for one clock, the SERIRQ must immediately be tri-stated without at any time driving high. A Start Frame may not be initiated while the SERIRQ is active. The SERIRQ is Idle between Stop and Start Frames. The SERIRQ is active between Start and Stop Frames. This mode of operation allows the SERIRQ to be Idle when there are no IRQ/Data transitions which should be most of the time.

Once a Start Frame has been initiated, the host controller will take over driving the SERIRQ low in the next clock and will continue driving the SERIRQ low for a programmable period of three to seven clocks. This makes a total low pulse width of four to eight clocks. Finally, the host controller will drive the SERIRQ back high for one clock then tri-state.

Any SERIRQ Device which detects any transition on an IRQ/Data line for which it is responsible must initiate a Start Frame in order to update the host controller unless the SERIRQ is already in an SERIRQ Cycle and the IRQ/Data transition can be delivered in that SERIRQ Cycle.

Continuous (Idle) Mode

Only the Host controller can initiate a Start Frame to update IRQ/Data line information. All other SERIRQ agents become passive and may not initiate a Start Frame. SERIRQ will be driven low for four to eight clocks by host controller. This mode has two functions. It can be used to stop or idle the SERIRQ or the host controller can operate SERIRQ in a continuous mode by initiating a Start Frame at the end of every Stop Frame.

An SERIRQ mode transition can only occur during the Stop Frame. Upon reset, SERIRQ bus is defaulted to continuous mode, therefore only the host controller can initiate the first Start Frame. Slaves must continuously sample the Stop Frames pulse width to determine the next SERIRQ Cycle's mode.

SERIRQ DATA FRAME

Once a Start Frame has been initiated, the LPC Controller will watch for the rising edge of the Start Pulse and start counting IRQ/Data Frames from there. Each IRQ/Data Frame is three clocks: Sample phase, Recovery phase, and Turn-around phase. During the sample phase, the LPC Controller must drive the SERIRQ (SIRQ pin) low, if and only if, its last detected IRQ/Data value was low. If its detected IRQ/Data value is high, SERIRQ must be left tri-stated. During the recovery phase, the LPC Controller must drive the SERIRQ high, if and only if, it had driven the SERIRQ low during the

previous sample phase. During the turn-around phase, the controller must tri-state the SERIRQ. The device drives the SERIRQ line low at the appropriate sample point if its associated IRQ/Data line is low, regardless of which device initiated the start frame.

The Sample phase for each IRQ/Data follows the low to high transition of the Start Frame pulse by a number of clocks equal to the IRQ/Data Frame times three, minus one e.g. The IRQ5 Sample clock is the sixth IRQ/Data Frame, then the sample phase is {(6 x 3) - 1 = 17} the seventeenth clock after the rising edge of the Start Pulse.

TABLE 4-10: SERIRQ SAMPLING PERIODS

SERIRQ Period	Signal Sampled	# of Clocks Past Start
1	Not Used	2
2	IRQ1	5
3	IRQ2	8
4	IRQ3	11
5	IRQ4	14
6	IRQ5	17
7	IRQ6	20
8	IRQ7	23
9	IRQ8	26
10	IRQ9	29
11	IRQ10	32
12	IRQ11	35
13	IRQ12	38
14	IRQ13	41
15	IRQ14	44
16	IRQ15	47

The SIRQ data frame will now support IRQ2 from a logical device; previously SERIRQ Period 3 was reserved for use by the System Management Interrupt (LSMI#). When using Period 3 for IRQ2, the user should mask off the SMI via the ESMI Mask Register. Likewise, when using Period 3 for LSMI#, the user should not configure any logical devices as using IRQ2.

SERIRQ Period 14 is used to transfer IRQ13. Each Logical devices will have IRQ13 as a choice for their primary interrupt.

STOP CYCLE CONTROL

Once all IRQ/Data Frames have completed, the host controller will terminate SERIRQ activity by initiating a Stop Frame. Only the host controller can initiate the Stop Frame. A Stop Frame is indicated when the SERIRQ is low for two or three clocks. If the Stop Frame's low time is two clocks, then the next SERIRQ cycle's sampled mode is the Quiet mode; and any SERIRQ device may initiate a Start Frame in the second clock or more after the rising edge of the Stop Frame's pulse. If the Stop Frame's low time is three clocks, then the next SERIRQ cycle's sampled mode is the continuous mode, and only the host controller may initiate a Start Frame in the second clock or more after the rising edge of the Stop Frame's pulse.

4.8.4.4 Latency

Latency for IRQ/Data updates over the SERIRQ bus in bridge-less systems with the minimum IRQ/Data Frames of 17 will range up to 96 clocks (3.84μS with a 25 MHz LCLK or 2.88μs with a 33 MHz LCLK).

Note:	If one or more PCI to PCI Bridge is added to a system, the latency for IRQ/Data updates from the secondary or tertiary buses will be a few clocks longer for synchronous buses, and approximately double for asynchronous buses.

4.8.4.5 EOI/ISR Read Latency

Any serialized IRQ scheme has a potential implementation issue related to IRQ latency. IRQ latency could cause an EOI or ISR Read to precede an IRQ transition that it should have followed. This could cause a system fault. The host interrupt controller is responsible for ensuring that these latency issues are mitigated. The recommended solution is to delay EOIs and ISR Reads to the interrupt controller by the same amount as the SERIRQ Cycle latency in order to ensure that these events do not occur out of order.

4.8.4.6 AC/DC Specification Issue

All Serial IRQ agents must drive/sample SERIRQ synchronously related to the rising edge of LCLK. The SERIRQ pin uses the electrical specification of the PCI bus. Electrical parameters will follow the PCI Local Bus Specification, Rev. 2.2 definition of “sustained tri-state.”

4.8.4.7 Reset and Initialization

The SERIRQ bus uses LRESET# as its reset signal and follows the PCI bus reset mechanism. The SERIRQ pin is tri-stated by all agents while LRESET# is active. With reset, SERIRQ slaves and bridges are put into the (continuous) Idle mode. The host controller is responsible for starting the initial SERIRQ cycle to collect system’s IRQ/Data default values. The system then follows with the Continuous/Quiet mode protocol (Stop Frame pulse width) for subsequent SERIRQ cycles. It is the host controller’s responsibility to provide the default values to the 8259’s and other system logic before the first SERIRQ cycle is performed. For SERIRQ system suspend, insertion, or removal application, the host controller should be programmed into Continuous (IDLE) mode first. This is to ensure the SERIRQ bus is in Idle state before the system configuration changes.

4.8.4.8 SERIRQ Interrupts

The LPC Controller routes Logical Device interrupts onto SIRQ stream frames IRQ[0:15]. Routing is controlled by the SIRQ Interrupt Configuration Registers. There is one SIRQ Interrupt Configuration Register for each accessible SIRQ Frame (IRQ); all 16 registers are listed in Table 4-14, "SIRQ Interrupt Configuration Register Map".

The format for each SIRQ Interrupt Configuration Register is described in Section 4.9.2.1, "SIRQ Configuration Register Format," on page 114. Each Logical Device can have up to two LPC SERIRQ interrupts. When the device is polled by the host, each SIRQ frame routes the level of the Logical Device interrupt (selected by the corresponding SIRQ Interrupt Configuration Register) to the SIRQ stream.

4.8.4.9 SERIRQ Routing

Each SIRQ Interrupt Configuration Register controls a series of multiplexers which route to a single Logical Device interrupt as illustrated in FIGURE 4-7: SIRQ Routing Internal Logical Devices on page 112. The following table defines the Serial IRQ routing for each logical device implemented in the chip.

TABLE 4-11: LOGICAL DEVICE SIRQ ROUTING

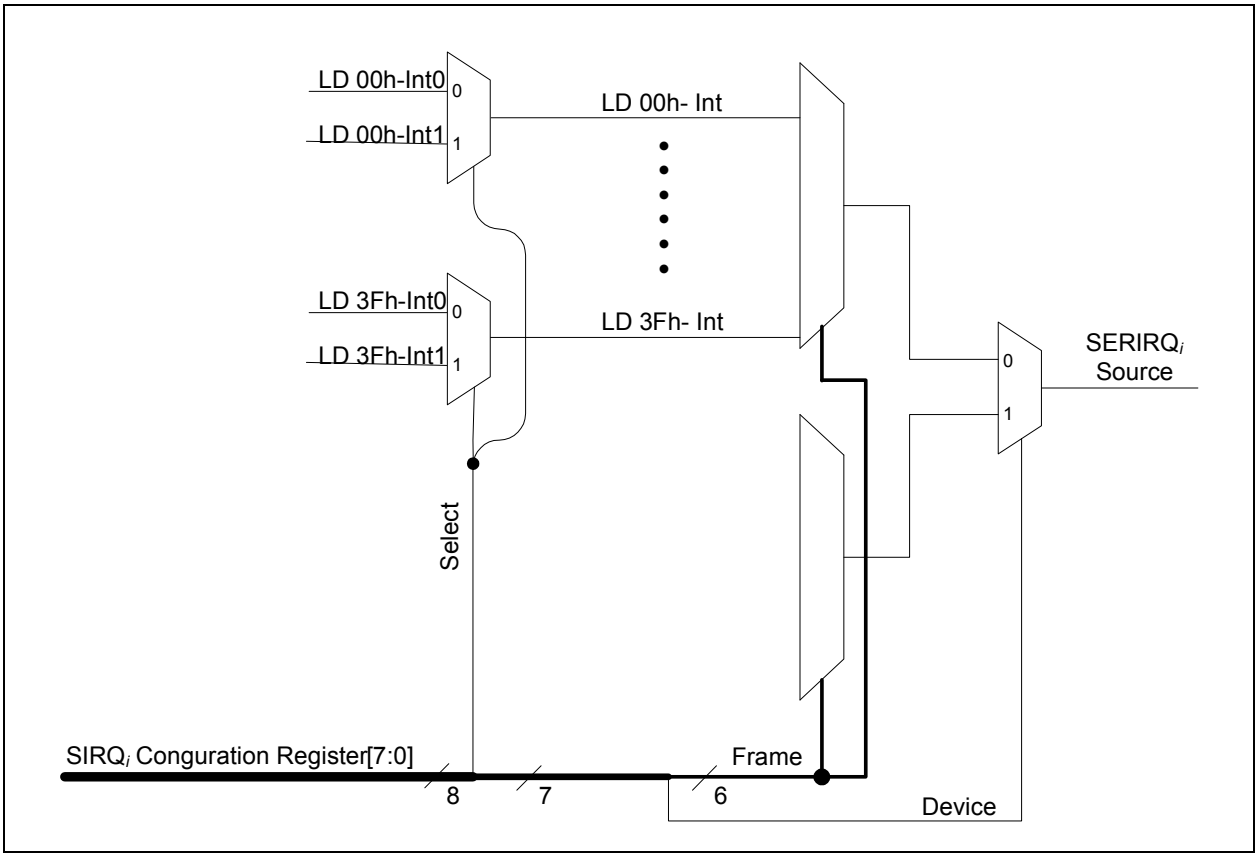
SIRQ Interrupt Configuration Register			**Logical Device Interrupt Source**	
Select	**Device**	**Frame**	**Logical Device (Block Instance - Note 26.2)**	**Interrupt Source**
0	0	C	LPC Interface (Configuration Port)	EC_IRQ
0	0	9	Mailbox Interface	MBX_Host_SIRQ

TABLE 4-11: LOGICAL DEVICE SIRQ ROUTING

SIRQ Interrupt Configuration Register			Logical Device Interrupt Source	
Select	**Device**	**Frame**	**Logical Device (Block Instance - Note 26.2)**	**Interrupt Source**
1	0	9	Mailbox Interface	MBX_Host_SMI
0	0	1	8042 Emulated Keyboard Controller	KIRQ
1	0	1	8042 Emulated Keyboard Controller	MIRQ
0	0	3	ACPI EC0	EC_OBF
0	0	4	ACPI EC1	EC_OBF
0	0	A	ACPI EC2	EC_OBF
0	0	B	ACPI EC3	EC_OBF
0	0	7	UART 0	UART
0	0	0	EMI 0	EC-to-Host
1	0	0	EMI 0	Host_SWI_Event

Note 4-8 The Block Instance number is only included if there are multiple instantiations of a block. Otherwise, single block instances do not require this differentiation.

FIGURE 4-7: SIRQ ROUTING INTERNAL LOGICAL DEVICES



Note: Two Logical Devices cannot share a Serial IRQ.

4.9 LPC Configuration Registers

The configuration registers listed in Table 4-13, "Configuration Register Summary" are for a single instance of the LPC Interface. The addresses of each register listed in TABLE 4-13: are defined as a relative offset to the host “Begin Address” defined in TABLE 4-12:.

TABLE 4-12: CONFIGURATION REGISTER ADDRESS RANGE

Instance NAME	Instance Number	Host	Address Space	Begin Address
LPC Interface	0	LPC	Configuration Port	INDEX = 00h
	0	EC	32-bit internal address space	000F_3300h

Note 4-9 The Begin Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 4-13: CONFIGURATION REGISTER SUMMARY

Register Name	Offset	Size	Notes
LPC Activate Register	30h	8	
SIRQ Configuration Register Format	40h - 4Fh	8	
I/O Base Address Registers (IO_BARs)	60h - 9Fh See TABLE 4-15:	32	
SRAM Memory BAR	A0h	32	
SRAM Memory BAR Configuration	A4h	32	
Device Memory Base Address Registers (DEV_MEM_BARs)	C0h - FFh See TABLE 4-16:	48	

4.9.1 LPC ACTIVATE REGISTER

The LPC Activate Register controls the LPC device itself. The Host can shut down the LPC Logical Device by clearing the Activate bit, but it cannot restart the LPC interface, since once the LPC interface is inactive the Host has no access to any registers on the device. The Embedded Controller can set or clear the Activate bit at any time.

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	RESERVED	RES	-	-
0	ACTIVATE 1= Activate When this bit is 1, the LPC Logical Device is powered and functional. 0= Deactivate When this bit is 0, the logical device is powered down and inactive. Except for the LPC Activate Register itself, clocks to the block are gated and the LPC Logical Device will permit the ring oscillator to be shut down (see Section 4.11.4, "EC Clock Control Register," on page 123). LPC bus output pads will be tri-stated.	R/W	0b	nSYSRST

APPLICATION NOTE: The bit in the LPC Activate Register should not be written '0' to by the Host over LPC.

4.9.2 SERIAL IRQ CONFIGURATION REGISTERS

The LPC Controller implements 16 IRQ channels that may be configured to be asserted by any logical device.

- For a description of the SIRQ Configuration Register format see Table 4-14, "SIRQ Interrupt Configuration Register Map," on page 114.
- For a summary of the SIRQ IRQ Configuration registers implemented see Table 4-15, "I/O Base Address Registers," on page 117.
- For a list of the SIRQ sources see Table 4-11, "Logical Device SIRQ Routing," on page 110.

4.9.2.1 SIRQ Configuration Register Format

Offset	See Table 4-14, “SIRQ Interrupt Configuration Register Map,” on page 114.			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	SELECT If this bit is 0, the first interrupt signal from the Logical Device is selected for the SERIRQ vector. If this bit is 1, the second interrupt signal from the Logical Device is selected. **Note:** The Keyboard Controller is an example of a Logical Devices that requires a second interrupt signal. Most Logical Devices require only a single interrupt and ignore this field as result.	R/W	Note 4-10	nSIO_ RESET
6	DEVICE This field should always be set to 0 in order to enable a SERIRQ.	R/W	Note 4-10	nSIO_ RESET
5:0	FRAME These six bits select the Logical Device for on-chip devices as the source for the interrupt. **Note:** The LPC Logical Device (Logical Device Number 0Ch) can be used by the Embedded Controller to generate a Serial Interrupt Request to the Host under software control.	R/W	Note 4-10	nSIO_ RESET

Note 4-10 See Table 4-14, “SIRQ Interrupt Configuration Register Map,” on page 114.

4.9.2.2 SIRQ Configuration Registers

TABLE 4-14: SIRQ INTERRUPT CONFIGURATION REGISTER MAP

Offset	Type	Reset	Configuration Register Name
40h	R/W	FFh	IRQ0
41h	R/W	FFh	IRQ1
42h	R/W	FFh	IRQ2
43h	R/W	FFh	IRQ3
44h	R/W	FFh	IRQ4
45h	R/W	FFh	IRQ5
46h	R/W	FFh	IRQ6
47h	R/W	FFh	IRQ7
48h	R/W	FFh	IRQ8
49h	R/W	FFh	IRQ9
4Ah	R/W	FFh	IRQ10
4Bh	R/W	FFh	IRQ11

TABLE 4-14: SIRQ INTERRUPT CONFIGURATION REGISTER MAP (CONTINUED)

Offset	Type	Reset	Configuration Register Name
4Ch	R/W	FFh	IRQ12
4Dh	R/W	FFh	IRQ13
4Eh	R/W	FFh	IRQ14
4Fh	R/W	FFh	IRQ15

Note: A SERIRQ interrupt is deactivated by setting an entry in the SIRQ Interrupt Configuration Register Map to FFh, which is the default reset value.

4.9.3 I/O BASE ADDRESS REGISTERS (IO_BARS)

The LPC Controller has implemented an I/O Base Address Register (BAR) for each Logical Device in the LPC Configuration space.

- For a description of the I/O Base Address Register format see Section 4.9.3.1, "I/O Base Address Register Format," on page 115.
- For a description of the I/O BARs per Logical Device see Table 4-15, “I/O Base Address Registers,” on page 117.

On every LPC bus I/O access the unmasked portion of the programmed LPC Host Address in each of the Base Address Registers are checked in parallel and if any matches the LPC I/O address the LPC Controller claims the bus cycle.

Note: Software should that insure that no two I/O BARs map the same LPC I/O address. If two I/O BARs do map to the same address, the LPC_INTERNAL_ERR and BAR_CONFLICT status bits are set when an LPC access is targeting the address with the BAR conflict.

The format of each BAR is summarized in Section 4.9.3.1, "I/O Base Address Register Format," on page 115.

4.9.3.1 I/O Base Address Register Format

Each LPC accessible logical device has a programmable I/O Base Address Register. The following table defines the generic format used for all of these registers. See Table 4-15, "I/O Base Address Registers" for a list of all the Logical Device Base Address registers implemented.

Offset	See Table 4-15, “I/O Base Address Registers,” on page 117			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	LPC Host Address These 16 bits are used to match LPC I/O addresses	R/W (Note 4-12)	See TABLE 4-15:	Note 4-11
15	VALID If this bit is 1, the BAR is valid and will participate in LPC matches. If it is 0 this BAR is ignored	R/W	See TABLE 4-15:	Note 4-11
14	DEVICE (device) This bit combined with FRAME constitute the Logical Device Number. DEVICE identifies the physical location of the logical device. This bit should always be set to 0.	R	See TABLE 4-15:	Note 4-11

Offset	See Table 4-15, “I/O Base Address Registers,” on page 117			
Bits	**Description**	**Type**	**Default**	**Reset Event**
13:8	FRAME These 6 bits are used to specify a logical device frame number within a bus. This field is multiplied by 400h to provide the frame address within the peripheral bus address. Frame values for frames corresponding to logical devices that are not present on the device are invalid.	R	See TABLE 4-15:	Note 4-11
7:0	MASK These 8 bits are used to mask off address bits in the address match between an LPC I/O address and the Host Address field of the BARs, as described in Section 4.8.2.1, "I/O Transactions". A block of up to 256 8-bit registers can be assigned to one base address.	R (See TABLE 4-15:)	See TABLE 4-15:	Note 4-11

Note 4-11 Offset 60h is the LPC Base Address register. The LPC Base Address register is only reset on nSYSRST. All other Base Address Registers are reset on nSIO_RESET.

Note 4-12 Bits[31:16] LPC Host Address bit field in the LPC Base Address register at offset 60h must be written LSB then MSB. This particular register has a shadow that lets the Host come in and write to the lower byte of the 16-bit address, and the resulting 16-bit LPC Host address field does not update. Writing to the upper byte triggers a full 16-bit field update.

4.9.3.2 Logical Device IO_BAR Description

The following table defines the IO_BAR of each logical device implemented in the design.

Note: After the VCC_PWRGD signal is asserted, the iRESET_OUT bit of the Power Reset Control (PWR_RST_CTRL) Register must be cleared by firmware in order to write the BAR registers listed.

TABLE 4-15: I/O BASE ADDRESS REGISTERS

				Base Address Register Bit Field Descriptions				
				Bits [D31:D16]	Bit [D15]	Bit [D14]	Bits [D13:D8]	Bits [D6:D0]
Offset	Logical Device Number (hex)	Logical Devices	Reset Default	Default LPC I/O Host Address	VALID	DEVICE	FRAME	MASK (Note 3)
60h	C	LPC Interface (Configuration Port)	002E_0C01 (Note 1)	002E	0	0	C	1
64h	0	EMI 0	0000_000F	0000	0	0	0	F
68h	1	8042 Emulated Keyboard Controller	0060_0104	0060	0	0	1	4
6Ch	3	ACPI EC0	0062_0304	0062	0	0	3	4
70h	4	ACPI EC1	0066_0407	0066	0	0	4	7
74h	5	ACPI PM1	0000_0507	0000	0	0	5	7
78h	6	Legacy Port92/GateA20	0092_0600	0092	0	0	6	0
7Ch	7	UART 0	0000_0707	0000	0	0	7	7
80h	9	Mailbox Interface	0000_0901	0000	0	0	9	1
84h	A	ACPI EC2	0000_0A07	0000	0	0	A	7
88h	B	ACPI EC3	0000_0B07	0000	0	0	B	7
8Ch	15	Port 80 BIOS Debug Port 0	0000_1500	0000	0	0	15	0
90h	16	Port 80 BIOS Debug Port 1	0000_1600	0000	0	0	16	0

Note 1: The default Base I/O Address of the Configuration Port can be relocated by programming the BAR register for Logical Device Ch (LPC/Configuration Port) at offset 60h.

Note 2: The FRAME and MASK fields for these Legacy devices are not used to determine which LPC I/O addresses to claim. The address range match is maintained within the blocks themselves.

Note 3: The ACPI-ECx Mask bit field is a read/write bit field. All other MASK bit fields are read-only as defined in the register description.

4.9.4 SRAM MEMORY BAR

Offset	A0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	LPC Host Address[31:24] These 32 bits are used to match LPC Memory addresses	R/W	0h	nSIO_RESET

4.9.5 SRAM MEMORY BAR CONFIGURATION

Offset	A4h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	RESERVED	RES	-	-
7	VALID If this bit is 1, the SRAM Memory BAR is valid and will participate in LPC matches. If it is 0 this SRAM Memory BAR is ignored.	R/W	0h	nSIO_RESET
6:1	RESERVED	RES	-	-

4.9.6 DEVICE MEMORY BASE ADDRESS REGISTERS (DEV_MEM_BARS)

Some Logical Devices have a Memory Base Address Register. These Device Memory BARs are located in blocks of

Configuration Registers in Logical Device 0Ch, in the AHB address range 000F_33C0h through 000F_33FFh. The following table defines the generic format used for all of these registers.

Each DEV_MEM_BAR is 48 bits wide. The format of each Device Memory BAR is summarized in Section 4.9.6.1, "Device Memory Base Address Register Format". An LPC memory request is translated by the Device Memory BAR into an 8-bit read or write transaction on the AHB bus. The 32-bit LPC memory address is translated into a 32-bit AHB address.

4.9.6.1 Device Memory Base Address Register Format

Offset	See Table 4-16, “Device Memory Base Address Registers,” on page 119			
Bits	**Description**	**Type**	**Default**	**Reset Event**
47:16	LPC Host Address These 16 bits are used to match LPC I/O addresses	R/W	See TABLE 4-16:	nSIO_RESET
15	VALID If this bit is 1, the BAR is valid and will participate in LPC matches. If it is 0 this BAR is ignored	R/W	See TABLE 4-16:	nSIO_RESET
14	DEVICE (device) This bit combined with FRAME constitute the Logical Device Number. DEVICE identifies the physical location of the logical device. This bit should always be set to 0.	R	See TABLE 4-16:	nSIO_RESET

Offset	See Table 4-16, “Device Memory Base Address Registers,” on page 119			
Bits	**Description**	**Type**	**Default**	**Reset Event**
13:8	FRAME These 6 bits are used to specify a logical device frame number within a bus. This field is multiplied by 400h to provide the frame address within the peripheral bus address. Frame values for frames corresponding to logical devices that are not present on the device are invalid.	R or R/W (see Note 3)	See TABLE 4-16:	nSIO_RESET
7:0	MASK These 8 bits are used to mask off address bits in the address match between an LPC I/O address and the Host Address field of the BARs, as described in Section 4.8.2.2, "Device Memory Transactions". A block of up to 256 8-bit registers can be assigned to one base address.	R or R/W (see Note 3)	See TABLE 4-16:	nSIO_RESET

3: The Mask and Frame fields of all logical devices are read-only except for 3h (ACPI EC Channel 0).

4.9.6.2 Device Memory Base Address Register Table

The table below lists the Base Address Registers for logical devices which have LPC memory access in this device. LPC Memory cycle access is controlled by LPC Memory Base Address Registers. LPC Memory BAR registers are located in LDN Ch (LPC Interface) at AHB base address 000F_3300h starting at the offset shown in the Device Memory Base Address Registers table.

TABLE 4-16: DEVICE MEMORY BASE ADDRESS REGISTERS

				Base Address Register Bit Field Descriptions				
				Bits [D47:D16]	**Bit [D15]**	**Bit [D14]**	**Bits [D13:D8]**	**Bits [D6:D0]**
Offset	**Logical Device Number (hex)**	**Logical Devices**	**Reset Default**	**Default LPC Mem Host Address**	**VALID**	**DEVICE**	**FRAME**	**MASK (Note 2)**
C0h	0	EMI 0	0000_0000_000F	0000_0000	0	0	0	F
C6h	3	ACPI EC0	0000_0062_0304	0000_0062	0	0	3	4
CCh	4	ACPI EC1	0000_0066_0407	0000_0066	0	0	4	7
D2h	9	Mailbox Interface	0000_0000_0901	0000_0000	0	0	9	1
D8h	A	ACPI EC2	0000_0000_0A07	0000_0000	0	0	A	7
DEh	B	ACPI EC3	0000_0000_0B07	0000_0000	0	0	B	7

Note 1: The FRAME and MASK fields for these Legacy devices are not used to determine which LPC Memory addresses to claim. The address range match is maintained within the blocks themselves.
Note 2: The ACPI-ECx Mask bit field is a read/write bit field. All other MASK bit fields are read-only as defined in the register

4.10 Runtime Registers

The runtime registers listed in Table 4-18, "Runtime Register Summary" are for a single instance of the LPC Interface. The addresses of each register listed in TABLE 4-18: are defined as a relative offset to the host “Begin Address” define in TABLE 4-2:.

TABLE 4-17: RUNTIME REGISTER ADDRESS RANGE TABLE

Instance Name	Instance Number	Host	Address Space	Begin Address
LPC Interface	0	LPC	I/O	Programmed BAR
		EC	32-bit internal address space	000F_3000h

Note 1: The Begin Address indicates where the first register can be accessed in a particular address space for a block instance.

2: The LPC Runtime registers are only accessible from the LPC interface and are used to implement the LPC Configuration Port. They are not accessible by any other Host.

TABLE 4-18: RUNTIME REGISTER SUMMARY

Offset	Register Name
00h	INDEX Register
01h	DATA Register

Note: The LPC Runtime Register space has been used to implement the INDEX and DATA registers in the Configuration Port. In CONFIG_MODE, the Configuration Port is used to access the Configuration Registers.

4.10.1 INDEX REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	INDEX The INDEX register, which is part of the Configuration Port, is used as a pointer to a Configuration Register Address. **Note:** For a description of accessing the Configuration Port see Section 4.8.3, "Configuration Port," on page 105.	R/W	0h	nSYSRST

4.10.2 DATA REGISTER

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	DATA The DATA register, which is part of the Configuration Port, is used to read or write data to the register currently being selected by the INDEX Register. **Note:** For a description of accessing the Configuration Port see Section 4.8.3, "Configuration Port," on page 105	R/W	0h	nSYSRST

4.11 EC-Only Registers

Note: EC-Only registers are not accessible by the LPC interface.

The registers listed in Table 4-20, "EC-Only Register Summary" are for a single instance of the LPC Interface. Their addresses are defined as a relative offset to the host base address defined in TABLE 4-19:.

The following table defines the fixed host base address for each LPC Interface instance.

TABLE 4-19: EC-ONLY REGISTER ADDRESS RANGE TABLE

INSTANCE NAME	INSTANCE NUMBER	HOST	ADDRESS SPACE	BEGIN ADDRESS
LPC Interface	0	EC	32-bit internal address space	000F_3100h

Note: The Begin Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 4-20: EC-ONLY REGISTER SUMMARY

Offset	Register Name
04h	LPC Bus Monitor Register
08h	Host Bus Error Register
0Ch	EC SERIRQ Register
10h	EC Clock Control Register
14h	Test Register
18h	Test Register
20h	I/O BAR Inhibit Register
24h	Reserved
28h	Reserved
2Ch	Reserved
30h	LPC BAR Init Register
40h	Device Memory BAR Inhibit Register
FCh	SRAM Memory Host Configuration Register

Note 4-13 Some Test registers are read/write registers. Modifying these registers may have unwanted results.

4.11.1 LPC BUS MONITOR REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	RESERVED	RES	-	-
1	TEST	R	0h	nSYSRST
0	LPCPD_STATUS This bit reflects the state of the LPCPD# input pin. The LPCPD_STATUS bit is the inverse of the LPCPD# pin (see Section 4.8.1.3, "LPC Clock Run and LPC Power Down Behavior," on page 99). When the LPCPD_STATUS bit is '0b', the LPCPD# input pin is deasserted (that is, the pin has the value '1b'). When the LPCPD_STATUS bit is '1b', the LPCPD# input pin is asserted (that is, the pin has the value '0b').	R	0h	nSYSRST

4.11.2 HOST BUS ERROR REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	ErrorAddress[23:16] This 24-bit field captures the 24-bit internal address of every LPC transaction whenever the bit LPC_INTERNAL_ERR in this register is 0. When LPC_INTERNAL_ERR is 1 this register is not updated but retains its previous value. When bus errors occur this field saves the address of the first address that caused an error.	R	0h	nSYSRST
5	DMA_ERR This bit is set to 1 whenever EN_INTERNAL_ERR is 1 and an LPC DMA access causes an internal bus error. Once set, it remains set until cleared by being written with a 1.	R/WC	0h	nSYSRST
4	CONFIG_ERR This bit is set to 1 whenever EN_INTERNAL_ERR is 1 and an LPC Configuration access causes an internal bus error. Once set, it remains set until cleared by being written with a 1.	R/WC	0h	nSYSRST
3	RUNTIME_ERR This bit is set to 1 whenever EN_INTERNAL_ERR is 1 and an LPC I/O access causes an internal bus error. This error will only occur if a BAR is misconfigured. Once set, it remains set until cleared by being written with a 1.	R/WC	0h	nSYSRST

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	BAR_CONFLICT This bit is set to 1 whenever a BAR conflict occurs on an LPC address. A Bar conflict occurs when more than one BAR matches the address during of an LPC cycle access. Once this bit is set, it remains set until cleared by being written with a 1.	R/WC	0h	nSYSRST
1	EN_INTERNAL_ERR When this bit is 0, only a BAR conflict, which occurs when two BARs match the same LPC I/O address, will cause LPC_INTERNAL_ERR to be set. When this bit is 1, internal bus errors will also cause LPC_INTERNAL_ERR to be set.	R/W	0h	nSYSRST
0	LPC_INTERNAL_ERR This bit is set whenever a BAR conflict or an internal bus error occurs as a result of an LPC access. Once set, it remains set until cleared by being written with a 1. This signal may be used to generate interrupts. See Section 4.6, "Interrupts," on page 98.	R/WC	0h	nSYSRST

4.11.3 EC SERIRQ REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	RESERVED	RES	-	-
0	EC_IRQ If the LPC Logical Device is selected as the source for a Serial Interrupt Request by an Interrupt Configuration register (see Section 4.8.4.8, "SERIRQ Interrupts," on page 110), this bit is used as the interrupt source.	R/W	0h	nSYSRST

4.11.4 EC CLOCK CONTROL REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	RESERVED	RES	-	-
2	Handshake This bit controls throughput of LPC transactions. When this bit is a ‘0’ the part supports a 33MHz PCI Clock. When this bit is a ‘1’, the part supports a PCI Clock from 24MHz to 33MHz.	RES	1h	nSYSRST

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1:0	Clock_Control This field controls when the host interface will permit the internal ring oscillator to be shut down. The choices are as follows: **0h:** The host interface will permit the internal clocks to be shut down if the LPCPD# signal is asserted (sampled low) **1h:** The host interface will permit the internal clocks to be shut down if the CLKRUN# signals “CLOCK STOP” and there are no pending serial interrupt request or DMA requests from devices associated with the device. The CLKRUN# signals “CLOCK STOP” by CLKRUN# being high for 5 LPCCLK’s after the raising edge of CLKRUN# **2h:** The host interface will permit the ring oscillator to be shut down after the completion of every LPC transaction. This mode may cause an increase in the time to respond to LPC transactions if the ring oscillator is off when the LPC transaction is detected. **3h:** The ring oscillator is not permitted to shut down as long as the host interface is active The bit in the LPC Activate Register should not be written ‘0’ to by the Host over LPC. When the bit in the LPC Activate Register is 0, the Host Interface will permit the ring oscillator to be shut down and the Clock_Control Field is ignored. The Clock_Control Field only effects the Host Interface when The bit in the LPC Activate Register should not be written ‘0’ to by the Host over LPC. bit in the LPC Activate Register is 1. Although the Host Interface can permit the internal oscillator to shut down, it cannot turn the oscillator on in response to an LPC transaction that occurs while the oscillator is off. In order to restart the oscillator in order to complete an LPC transaction, EC firmware must enable the LPC_WAKE interrupt. See the Application Note in Section 10.11.3.1, "GIRQ16 and GIRQ22 Wake-Only Events" for details.	R/W	0h	nSYSRST

4.11.5 I/O BAR INHIBIT REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
63:0	BAR_Inhibit[63:0] When bit *Di* of BAR_Inhibit is 1, the BAR for Logical Device *i* is disabled and its addresses will not be claimed on the LPC bus, independent of the value of the Valid bit in the BAR.The association between bits in BAR_Inhibit and Logical Devices is illustrated in Table 4-21, "BAR Inhibit Device Map".	R/W	0h	nSYSRST

TABLE 4-21: BAR INHIBIT DEVICE MAP

Bar Inhibit Bit	Logical Device Number
0	0h
1	1h
. . .	. . .
31	31h

4.11.6 LPC BAR INIT REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	BAR_Init This field is loaded into the LPC BAR at offset 60h on nSIO_RESET.	R/W	002Eh	nSIO_RESET

4.11.7 DEVICE MEMORY BAR INHIBIT REGISTER

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
63:0	Device Mem BAR_Inhibit[63:0] When bit i of the Device Mem BAR_Inhibit[63:0] field is asserted (‘1’), where i is the logical device number of one of the Device Memory Base Address Registers, the BAR for the associated device is disabled and its LPC Memory addresses will not be claimed on the LPC bus, independent of the value of the Valid bit in the BAR. When bit i is not asserted (default), BAR activity for the Logical Device is based on the Valid bit in the BAR. All of the Device Mem BAR_Inhibit[63:0] bits are R/W and have no affect on reserved logical device numbers.	R/W	0h	nSYSRST

4.11.8 SRAM MEMORY HOST CONFIGURATION REGISTER

Offset	FCh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	AHB Base These 24 bits define the base of a region in AHB address space that will be mapped to the LPC Memory space. Valid AHB addresses are integer multiples ot the memory size. For example, if the memory is 4k bytes than the AHB Base address must be located on a 4k byte boundary. **Note:** The 24 bits in this field are left-shited by 8 bits to form a 32-bit AHB address, so all memory blocks begin on a 256-byte boundary.	R/W	0h	nSYSRST
7	Inhibit Host access to the memory block is inhibited when this bit is 1. The Host can access the memory region mapped by the fields AHB Base and Size when this bit is 0.	R/W	0h	nSYSRST
6:4	RESERVED	RES	-	-
3:0	Size The number of address bits to pass unchanged when translating an LPC address to an AHB address. These 4 bits in effect define the size of the block to be claimed by the LPC bridge, defined as a power of 2. A value of 0 defines a 2^0 or a 1-byte region starting at LPC Host Address. A value of 12 defines a 2^{12} or a 4K-byte region. Values larger than 12 are undefined..	R/W	0h	nSYSRST

5.0 ENHANCED SERIAL PERIPHERAL INTERFACE (ESPI)

5.1 Introduction

The Intel® Enhanced Serial Peripheral Interface (eSPI) is used by the system host to configure the chip and communicate with the logical devices implemented in the design through a series of read/write registers. It is Intel's successor to the Low Pin Count (LPC) bus, used in previous devices to provide System Host access to devices internal to the Embedded Controller.

5.2 References

1. Intel, *Enhanced Serial Peripheral Interface (eSPI): Interface Specification (for Client Platforms)*
2. Microchip "eSPI Controller" Specification
3. MEC140x/1x eSPI Addendum

6.0 QUAD SPI MASTER CONTROLLER

6.1 Overview

The Quad SPI Master Controller may be used to communicate with various peripheral devices that use a Serial Peripheral Interface, such as EEPROMS, DACs and ADCs. The controller can be configured to support advanced SPI Flash devices with multi-phase access protocols. Data can be transfered in Half Duplex, Single Data Rate, Dual Data Rate and Quad Data Rate modes. In all modes and all SPI clock speeds, the controller supports back-to-back reads and writes without clock stretching if internal bandwidth permits.

6.2 References

No references have been cited for this feature.

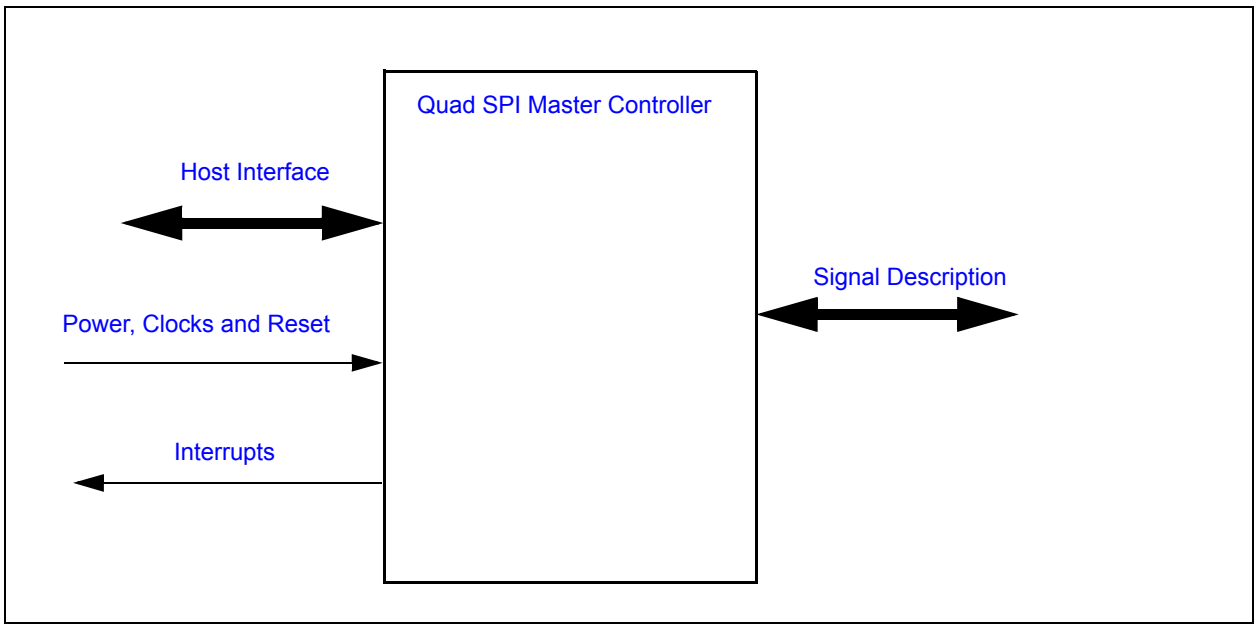
6.3 Terminology

No terminology for this block.

6.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 6-1: I/O DIAGRAM OF BLOCK



6.5 Signal Description

TABLE 6-1: SIGNAL DESCRIPTION

Name	Direction	Description
SPI_CLK	Output	SPI Clock output used to drive the SPCLK pin.
SPI_CS#	Output	SPI chip select

TABLE 6-1: SIGNAL DESCRIPTION (CONTINUED)

Name	Direction	Description
SPI_IO0	Input/Output	SPI Data pin 0. Also used as SPI_MOSI, Master-Out/Slave-In when the interface is used in Single wire mode
SPI_IO1	Input/Output	SPI Data pin 1. Also used as SPI_MISO, Master-In/Slave-Out when the interface is used in Single wire mode
SPI_IO2	Input/Output	SPI Data pin 2 when the SPI interface is used in Quad Mode. Also can be used by firmware as WP.
SPI_IO3	Input/Output	SPI Data pin 3 when the SPI interface is used in Quad Mode. Also can be used by firmware as HOLD.

6.6 Host Interface

The registers defined for the General Purpose Serial Peripheral Interface are accessible by the various hosts as indicated in Section 6.11, "EC-Only Registers".

6.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

6.7.1 POWER

Name	Description
VTR	The logic and registers implemented in this block are powered by this power well.

6.7.2 CLOCKS

Name	Description
48 MHz Ring Oscillator	This is a clock source for the SPI clock generator.

6.7.3 RESETS

Name	Description
nSYSRST	This signal resets all the registers and logic in this block to their default state.QMSPI Status Register
RESET	This reset is generated if either the nSYSRST is asserted or the SOFT_RESET is asserted.

6.8 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
QMSPI_INT	Interrupt generated by the Quad SPI Master Controller. Events that may cause the interrupt to be asserted are stored in the QMSPI Status Register.

6.9 Low Power Modes

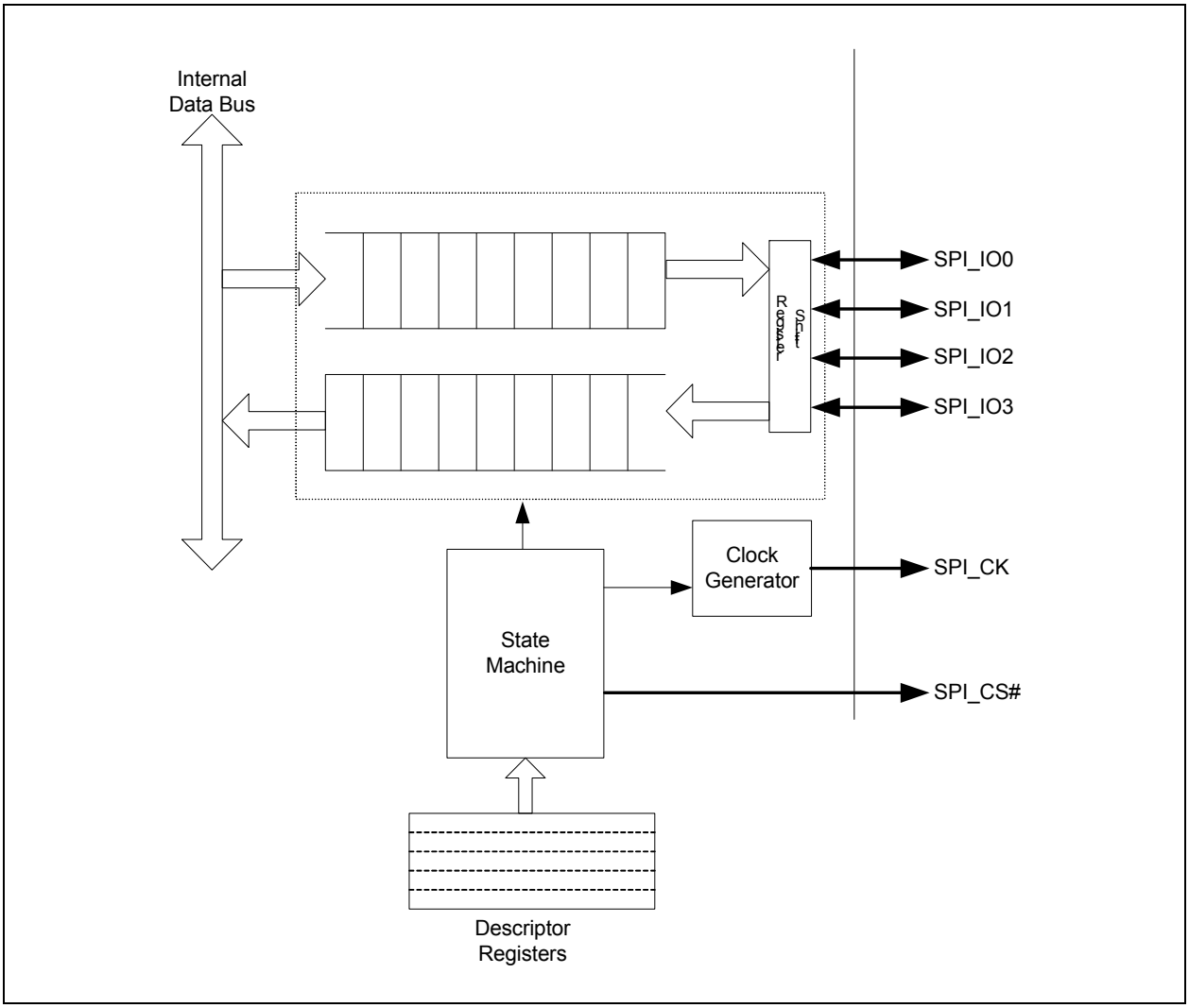
The Quad SPI Master Controller is always in its lowest power state unless a transaction is in process. A transaction is in process between the time the START bit is written with a ‘1’ and the TRANSFER_DONE bit is set by hardware to ‘1’.

If the QMSPI SLEEP_ENABLE input is asserted, writes to the START bit are ignored and the Quad SPI Master Controller will remain in its lowest power state.

6.10 Description

- Support for multiple SPI pin configurations
  - Single wire half duplex
  - Two wire full duplex
  - Two wire double data rate
  - Four wire quad data rate
- Separate FIFO buffers for Receive and Transmit
  - 8 byte FIFO depth in each FIFO
  - Each FIFO can be 1 byte, 2 bytes or 4 bytes wide
- Support for all four SPI clock formats
- Programmable SPI Clock generator, with clock polarity and phase controls
- Separate DMA support for Receive and Transmit data transfers
- Configurable interrupts, for errors, individual bytes, or entire transactions
- Descriptor Mode, in which a set of five descriptor registers can configure the controller to autonomously perform multi-phase SPI data transfers
- Capable of wire speed transfers in all SPI modes and all configurable SPI clock rates (internal bus contention may cause clock stretching)

FIGURE 6-2: QUAD MASTER SPI BLOCK DIAGRAM



6.10.1 SPI CONFIGURATIONS MODES

- Half Duplex. All SPI data transfers take place on a single wire, SPI_IO0
- Full Duplex. This is the legacy SPI configuration, where all SPI data is transfered one bit at a time and data from the SPI Master to the SPI Slave takes place on SPI_MOSI (SPI_IO0) and at the same time data from the SPI Slave to the SPI Master takes place on SPI_MISO (SPI_IO1)
- Dual Data Rate. Data transfers between the SPI Master and the SPI Slave take place two bits at a time, using SPI_IO0 and SPI_IO1
- Quad Data Rate. Data transfers between the SPI Master and the SPI Slave take place four bits at a time, using all four SPI data wires, SPI_IO0, SPI_IO1, SPI_IO2 and SPI_IO3

6.10.2 SPI CONTROLLER MODES

- Manual. In this mode, firmware control all SPI data transfers byte at a time
- DMA. Firmware configures the SPI Master controller for characteristics like data width but the transfer of data between the FIFO buffers in the SPI controller and memory is controlled by the DMA controller. DMA transfers can take place from the Slave to the Master, from the Master to the Slave, or in both directions simultaneously
- Descriptor. Descriptor Mode extends the SPI Controller so that firmware can configure a multi-phase SPI transfer, in which each phase may have a different SPI bus width, a different direction, and a different length. For example, firmware can configure the controller so that a read from an advanced SPI Flash, which consists of a command

phase, an address phase, a dummy cycle phase and the read phase, can take place as a single operation, with a single interrupt to firmware when the entire transfer is completed

6.10.3 SPI CLOCK

The SPI output clock is derived from the 48 MHz Ring Oscillator, divided by a value programmed in the CLOCK_DIVIDE field of the QMSPI Mode Register. Sample frequencies are shown in the following table:

TABLE 6-2: EXAMPLE SPI FREQUENCIES

CLOCK_DIVIDE	SPI Clock Frequency
0	187.5 KHz
1	48 MHz
2	24 MHz
3	16 MHz
6	8 MHz
48	1 MHz
128	375 KHz
255	188.25 KHz

6.10.4 ERROR CONDITIONS

The Quad SPI Master Controller can detect some illegal configurations. When these errors are detected, an error is signaled via the PROGRAMMING_ERROR status bit. This bit is asserted when any of the following errors are detected:

- Both Receive and the Transmit transfers are enabled when the SPI Master Controller is configured for Dual Data Rate or Quad Data Rate
- Both Pull-up and Pull-down resistors are enabled on either the Receive data pins or the Transmit data pins
- The transfer length is programmed in bit mode, but the total number of bits is not a multiple of 2 (when the controller is configured for Dual Data Rate) or 4 (when the controller is configured for Quad Data Rate)
- Both the STOP bit and the START bits in the QMSPI Execute Register are set to ‘1’ simultaneously

6.11 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the General Purpose Serial Peripheral Interface. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 6-3: EC-ONLY REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
Quad Mode Serial Peripheral Interface	0	EC	32-bit internal address space	0000_5400h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 6-4: EC-ONLY REGISTER SUMMARY

Offset	Register Name
0h	QMSPI Mode Register
4h	QMSPI Control Register
8h	QMSPI Execute Register
Ch	QMSPI Interface Control Register
10h	QMSPI Status Register
14h	QMSPI Buffer Count Status Register
18h	QMSPI Interrupt Enable Register
1Ch	QMSPI Buffer Count Trigger Register
20h	QMSPI Transmit Buffer Register
24h	QMSPI Receive Buffer Register
30h	QMSPI Description Buffer 0 Register
34h	QMSPI Description Buffer 1 Register
38h	QMSPI Description Buffer 2 Register
3Ch	QMSPI Description Buffer 3 Register
40h	QMSPI Description Buffer 4 Register

6.11.1 QMSPI MODE REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	R	-	-
24:16	CLOCK_DIVIDE The SPI clock divide in number of system clocks. A value of 1 divides the master clock by 1, a value of 255 divides the master clock by 255. A value of 0 divides the master clock by 256. See Table 6-2, "Example SPI Frequencies" for examples.	R/W	0h	RESET
15:11	Reserved	R	-	-

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	CHPA_MISO Clock phase of the Master data in. In normal SPI modes, this field must be programmed with the same value as CHPA_MOSI in this register. If CPOL=0: 1=Data are captured on the rising edge of the SPI clock 0=Data are captured on the falling edge of the SPI clock If CPOL=0: 1=Data are captured on the falling edge of the SPI clock 0=Data are captured on the rising edge of the SPI clock	R/W	0h	RESET
9	CHPA_MOSI Clock phase of the Master data out. In normal SPI modes, this field must be programmed with the same value as CHPA_MISO in this register. If CPOL=0: 1=Data changes on the falling edge of the SPI clock 0=Data changes on the rising edge of the SPI clock If CPOL=0: 1=Data changes on the rising edge of the SPI clock 0=Data changes on the falling edge of the SPI clock	R/W	0h	RESET
8	CPOL Polarity of the SPI clock line when there are no transactions in process. 1=SPI Clock starts High 0=SPI Clock starts Low	R/W	0h	RESET
7:2	Reserved	R	-	-
1	SOFT_RESET Writing this bit with a ‘1’ will reset the Quad SPI block. It is self-clearing.	W	0h	nSYSRST
0	ACTIVATE 1=Enabled. The block is fully operational 0=Disabled. Clocks are gated to conserve power and the output signals are set to their inactive state	R/W	0h	RESET

6.11.2 QMSPI CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	TRANSFER_LENGTH The length of the SPI transfer. The count is in bytes or bits, depending on the value of TRANSFER_LENGTH_BITS. A value of '0' means an infinite length transfer.	R/W	0h	RESET
15:12	DESCRIPTION_BUFFER_POINTER This field selects the first buffer used if Description Buffers are enabled.	R/W	0h	RESET
11	DESCRIPTION_BUFFER_ENABLE This enables the Description Buffers to be used. 1=Description Buffers in use. The first buffer is defined in DESCRIPTION_BUFFER_POINTER 0=Description Buffers disabled	R/W	0h	RESET
10	TRANSFER_LENGTH_BITS 1=TRANSFER_LENGTH defined in bits 0=TRANSFER_LENGTH defined in bytes	R/W	0h	RESET
9	CLOSE_TRANSFER_ENABLE This selects what action is taken at the end of a transfer. When the transaction closes, the Chip Select de-asserts, the SPI interface returns to IDLE and the DMA interface terminates When Description Buffers are in use this bit must be set only on the Last Buffer. 1=The transaction is terminated 0=The transaction is not terminated	R/W	1h	RESET
8:7	RX_DMA_ENABLE This bit enables DMA support for Receive Transfer. If enabled, DMA will be requested to empty the FIFO until either the interface reaches TRANSFER_LENGTH or the DMA sends a termination request. The size defined here must match DMA programmed access size. 1=DMA is enabled.and set to 1 Byte 2=DMA is enabled and set to 2 Bytes 3=DMA is enabled and set to 4 Bytes 0=DMA is disabled. All data in the Receive Buffer must be emptied by firmware	R/W	0h	RESET
6	RX_TRANSFER_ENABLE This bit enables the receive function of the SPI interface. 1=Receive is enabled. Data received from the SPI Slave is stored in the Receive Buffer 0=Receive is disabled	R/W	0h	RESET

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
5:4	TX_DMA_ENABLE This bit enables DMA support for Transmit Transfer. If enabled, DMA will be requested to fill the FIFO until either the interface reaches TRANSFER_LENGTH or the DMA sends a termination request. The size defined here must match DMA programmed access size. 1=DMA is enabled.and set to 1 Byte 2=DMA is enabled and set to 2 Bytes 3=DMA is enabled and set to 4 Bytes 0=DMA is disabled. All data in the Transmit Buffer must be emptied by firmware	R/W	0h	RESET
3:2	TX_TRANSFER_ENABLE This field bit selects the transmit function of the SPI interface. 3=Transmit Enabled in 1 Mode. The MOSI or IO Bus will send out only 1's. The Transmit Buffer will not be used 2=Transmit Enabled in 0 Mode. The MOSI or IO Bus will send out only 0's. The Transmit Buffer will not be used. 1=Transmit Enabled. Data will be fetched from the Transmit Buffer and sent out on the MOSI or IO Bus. 0=Transmit is Disabled. Not data is sent. This will cause the MOSI be to be undriven, or the IO bus to be undriven if Receive is also disabled.	R/W	0h	RESET
1:0	INTERFACE_MODE This field sets the transmission mode. If this field is set for Dual Mode or Quad Mode then either TX_TRANSFER_ENABLE or RX_TRANSFER_ENABLE **must** be 0. 3=Reserved 2=Quad Mode 1=Dual Mode 0=Single/Duplex Mode	R/W	0h	RESET

6.11.3 QMSPI EXECUTE REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	R	-	-
2	CLEAR_DATA_BUFFER Writing a ‘1’ to this bit will clear out the Transmit and Receive FIFOs. Any data stored in the FIFOs is discarded and all count fields are reset. Writing a ‘0’ to this bit has no effect. This bit is self-clearing.	W	0h	RESET

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	STOP Writing a ‘1’ to this bit will stop any transfer in progress at the next byte boundary. Writing a ‘0’ to this bit has no effect. This bit is self-clearing. This bit must not be set to ‘1’ if the field START in this register is set to ‘1’.	W	0h	RESET
0	START Writing a ‘1’ to this bit will start the SPI transfer. Writing a ‘0’ to this bit has no effect. This bit is self-clearing. This bit must not be set to ‘1’ if the field STOP in this register is set to ‘1’.	W	1h	RESET

6.11.4 QMSPI INTERFACE CONTROL REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7	PULLUP_ON_NOT_DRIVEN 1=Enable pull-up resistors on Transmit pins while the pins are not driven 0=No pull-up resistors enabled ion Transmit pins	R/W	0h	RESET
6	PULLDOWN_ON_NOT_DRIVEN 1=Enable pull-down resistors on Transmit pins while the pins are not driven 0=No pull-down resistors enabled ion Transmit pins	R/W	0h	RESET
5	PULLUP_ON_NOT_SELECTED 1=Enable pull-up resistors on Receive pins while the SPI Chip Select signal is not asserted 0=No pull-up resistors enabled on Receive pins	R/W	1h	RESET
4	PULLDOWN_ON_NOT_SELECTED 1=Enable pull-down resistors on Receive pins while the SPI Chip Select signal is not asserted 0=No pull-down resistors enabled on Receive pins	R/W	0h	RESET

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	HOLD_OUT_ENABLE 1=HOLD SPI Output Port is driven 0=HOLD SPI Output Port is not driven	R/W	0h	RESET
2	HOLD_OUT_VALUE This bit sets the value on the HOLD SPI Output Port if it is driven. 1=HOLD is driven to 1 0=HOLD is driven to 0	R/W	1h	RESET
1	WRITE_PROTECT_OUT_ENABLE 1=WRITE PROTECT SPI Output Port is driven 0=WRITE PROTECT SPI Output Port is not driven	R/W	0h	RESET
0	WRITE_PROTECT_OUT_VALUE This bit sets the value on the WRITE PROTECT SPI Output Port if it is driven. 1=WRITE PROTECT is driven to 1 0=WRITE PROTECT is driven to 0	R/W	1h	RESET

6.11.5 QMSPI STATUS REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	Reserved	R	-	-
27:24	CURRENT_DESCRIPTION_BUFFER This field shows the Description Buffer currently active. This field has no meaning if Description Buffers are not enabled.	R	0h	RESET
23:17	Reserved	R	-	-
16	TRANSFER_ACTIVE 1=A transfer is currently executing 0=No transfer currently in progress	R	0h	RESET
15	RECEIVE_BUFFER_STALL 1=The SPI interface had been stalled due to a flow issue (an attempt by the interface to write to a full Receive Buffer) 0=No stalls occurred	R/WC	0h	RESET

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
14	RECEIVE_BUFFER_REQUEST This status is asserted if the Receive Buffer reaches a high water mark established by the RECEIVE_BUFFER_TRIGGER field. 1=RECEIVE_BUFFER_COUNT is greater than or equal to RECEIVE_BUFFER_TRIGGER 0=RECEIVE_BUFFER_COUNT is less than RECEIVE_BUFFER_TRIGGER	R/WC	0h	RESET
13	RECEIVE_BUFFER_EMPTY 1=The Receive Buffer is empty 0=The Receive Buffer is not empty	R	1h	RESET
12	RECEIVE_BUFFER_FULL 1=The Receive Buffer is full 0=The Receive Buffer is not full	R	0h	RESET
11	TRANSMIT_BUFFER_STALL 1=The SPI interface had been stalled due to a flow issue (an attempt by the interface to read from an empty Transmit Buffer) 0=No stalls occurred	R/WC	0h	RESET
10	TRANSMIT_BUFFER_REQUEST This status is asserted if the Transmit Buffer reaches a high water mark established by the TRANSMIT_BUFFER_TRIGGER field. 1=TRANSMIT_BUFFER_COUNT is less than or equal to TRANSMIT_BUFFER_TRIGGER 0=TRANSMIT_BUFFER_COUNT is greater than TRANSMIT_BUFFER_TRIGGER	R/WC	0h	RESET
9	TRANSMIT_BUFFER_EMPTY 1=The Transmit Buffer is empty 0=The Transmit Buffer is not empty	R	0h	RESET
8	TRANSMIT_BUFFER_FULL 1=The Transmit Buffer is full 0=The Transmit Buffer is not full	R	0h	RESET
7:5	Reserved	R	-	-
4	PROGRAMMING_ERROR This bit if a programming error is detected. Programming errors are listed in Section 6.10.4, "Error Conditions". 1=Programming Error detected 0=No programming error detected	R/WC	0h	RESET

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	RECEIVE_BUFFER_ERROR 1=Underflow error occurred (attempt to read from an empty Receive Buffer) 0=No underflow occurred	R/WC	0h	RESET
2	TRANSMIT_BUFFER_ERROR 1=Overflow error occurred (attempt to write to a full Transmit Buffer) 0=No overflow occurred	R/WC	0h	RESET
1	DMA_COMPLETE This field has no meaning if DMA is not enabled. This bit will be set to '1' when the DMA controller asserts the DONE signal to the SPI controller. This occurs either when the SPI controller has closed the DMA transfer, or the DMA channel has completed its count. If both Transmit and Receive DMA transfers are active, then this bit will only assert after both have completed. If CLOSE_TRANSFER_ENABLE is enabled, DMA_COMPLETE and TRANSFER_COMPLETE will be asserted simultaneously. This status is not inhibited by the description buffers, so it can fire on all valid description buffers while operating in that mode. 1=DMA completed 0=DMA not completed	R/WC	0h	RESET
0	TRANSFER_COMPLETE In Manual Mode (neither DMA nor Description Buffers are enabled), this bit will be set to '1' when the transfer matches TRANSFER_LENGTH. If DMA Mode is enabled, this bit will be set to '1' when DMA_COMPLETE is set to '1'. In Description Buffer Mode, this bit will be set to '1' only when the Last Buffer completes its transfer. In all cases, this bit will be set to '1' if the STOP bit is set to '1' and the controller has completed the current 8 bits being copied. 1=Transfer completed 0=Transfer not complete	R/WC	0h	RESET

6.11.6 QMSPI BUFFER COUNT STATUS REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	RECEIVE_BUFFER_COUNT This is a count of the number of bytes currently valid in the Receive Buffer.	R	0h	RESET
15:0	TRANSMIT_BUFFER_COUNT This is a count of the number of bytes currently valid in the Transmit Buffer.	R	0h	RESET

6.11.7 QMSPI INTERRUPT ENABLE REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	R	-	-
14	RECEIVE_BUFFER_REQUEST_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_REQUEST is asserted 0=Disable the interrupt	R/W	0h	RESET
13	RECEIVE_BUFFER_EMPTY_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_EMPTY is asserted 0=Disable the interrupt	R/W	1h	RESET
12	RECEIVE_BUFFER_FULL_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_FULL is asserted 0=Disable the interrupt	R/W	0h	RESET
11	Reserved	R	-	-
10	TRANSMIT_BUFFER_REQUEST_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_REQUEST is asserted 0=Disable the interrupt	R/W	0h	RESET
9	TRANSMIT_BUFFER_EMPTY_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_EMPTY is asserted 0=Disable the interrupt	R/W	0h	RESET

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
8	TRANSMIT_BUFFER_FULL_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_FULL is asserted 0=Disable the interrupt	R/W	0h	RESET
7:5	Reserved	R	-	-
4	PROGRAMMING_ERROR_ENABLE 1=Enable an interrupt if PROGRAMMING_ERROR is asserted 0=Disable the interrupt	R/W	0h	RESET
3	RECEIVE_BUFFER_ERROR_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_ERROR is asserted 0=Disable the interrupt	R/W	0h	RESET
2	TRANSMIT_BUFFER_ERROR_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_ERROR is asserted 0=Disable the interrupt	R/W	0h	RESET
1	DMA_COMPLETE_ENABLE 1=Enable an interrupt if DMA_COMPLETE is asserted 0=Disable the interrupt	R/W	0h	RESET
0	TRANSFER_COMPLETE_ENABLE 1=Enable an interrupt if TRANSFER_COMPLETE is asserted 0=Disable the interrupt	R/W	0h	RESET

6.11.8 QMSPI BUFFER COUNT TRIGGER REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	RECEIVE_BUFFER_TRIGGER An interrupt is triggered if the RECEIVE_BUFFER_COUNT field is greater than or equal to this value. A value of ‘0’ disables the interrupt.	R/W	0h	RESET
15:0	TRANSMIT_BUFFER_TRIGGER An interrupt is triggered if the TRANSMIT_BUFFER_COUNT field is less than or equal to this value. A value of ‘0’ disables the interrupt.	R/W	0h	RESET

6.11.9 QMSPI TRANSMIT BUFFER REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	TRANSMIT_BUFFER Writes to this register store data to be transmitted from the SPI Master to the external SPI Slave. Writes to this block will be written to the Transmit FIFO. A 1 Byte write fills 1 byte of the FIFO. A Word write fills 2 Bytes and a Doubleword write fills 4 bytes. The data must always be aligned to the bottom most byte (so 1 byte write is on bits [7:0] and Word write is on [15:0]). An overflow condition,TRANSMIT_BUFFER_ERROR, if a write to a full FIFO occurs. Write accesses to this register increment the TRANSMIT_BUFFER_COUNT field.	W	0h	RESET

6.11.10 QMSPI RECEIVE BUFFER REGISTER

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	RECEIVE_BUFFER Buffer that stores data from the external SPI Slave device to the SPI Master (this block), which is received over MISO or IO. Reads from this register will empty the Rx FIFO. A 1 Byte read will have valid data on bits [7:0] and a Word read will have data on bits [15:0]. It is possible to request more data than the FIFO has (underflow condition), but this will cause an error (Rx Buffer Error). Read accesses to this register decrement the RECEIVE_BUFFER_COUNT field.	R	0h	RESET

6.11.11 QMSPI DESCRIPTION BUFFER 0 REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	TRANSFER_LENGTH The length of the SPI transfer. The count is in bytes or bits, depending on the value of TRANSFER_LENGTH_BITS. A value of '0' means an infinite length transfer.	R/W	0h	RESET
15:12	DESCRIPTION_BUFFER_NEXT_POINTER This defines the next buffer to be used if Description Buffers are enabled and this is not the last buffer. This can point to the current buffer, creating an infinite loop.	R/W	0h	RESET
11	DESCRIPTION_BUFFER_LAST If this bit is '1' then this is the last Description Buffer in the chain. When the transfer described by this buffer completes the TRANSFER_COMPLETE status will be set to '1'. If this bit is '0', then this is not the last buffer in use. When the transfer completes the next buffer will be activated, and no additional status will be asserted.	R/W	0h	RESET
10	TRANSFER_LENGTH_BITS 1=TRANSFER_LENGTH defined in bits 0=TRANSFER_LENGTH defined in bytes	R/W	0h	RESET
9	CLOSE_TRANFSER_ENABLE This selects what action is taken at the end of a transfer. This bit must be set only on the Last Buffer. 1=The transfer is terminated. The Chip Select de-asserts, the SPI interface returns to IDLE and the DMA interface completes the transfer. 0=The transfer is not closed. Chip Select remains asserted and the DMA interface and the SPI interface remain active	R/W	1h	RESET
8:7	RX_DMA_ENABLE This bit enables DMA support for Receive Transfer. If enabled, DMA will be requested to empty the FIFO until either the interface reaches TRANSFER_LENGTH or the DMA sends a termination request. The size defined here must match DMA programmed access size. 1=DMA is enabled.and set to 1 Byte 2=DMA is enabled and set to 2 Bytes 3=DMA is enabled and set to 4 Bytes 0=DMA is disabled. All data in the Receive Buffer must be emptied by firmware	R/W	0h	RESET

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
6	RX_TRANSFER_ENABLE This bit enables the receive function of the SPI interface. 1=Receive is enabled. Data received from the SPI Slave is stored in the Receive Buffer 0=Receive is disabled	R/W	0h	RESET
5:4	TX_DMA_ENABLE This bit enables DMA support for Transmit Transfer. If enabled, DMA will be requested to fill the FIFO until either the interface reaches TRANSFER_LENGTH or the DMA sends a termination request. The size defined here must match DMA programmed access size. 1=DMA is enabled.and set to 1 Byte 2=DMA is enabled and set to 2 Bytes 3=DMA is enabled and set to 4 Bytes 0=DMA is disabled. All data in the Transmit Buffer must be emptied by firmware	R/W	0h	RESET
3:2	TX_TRANSFER_ENABLE This field bit selects the transmit function of the SPI interface. 3=Transmit Enabled in 1 Mode. The MOSI or IO Bus will send out only 1's. The Transmit Buffer will not be used 2=Transmit Enabled in 0 Mode. The MOSI or IO Bus will send out only 0's. The Transmit Buffer will not be used. 1=Transmit Enabled. Data will be fetched from the Transmit Buffer and sent out on the MOSI or IO Bus. 0=Transmit is Disabled. No data is sent. This will cause the MOSI be to be undriven, or the IO bus to be undriven if Receive is also disabled.	R/W	0h	RESET
1:0	INTERFACE_MODE This field sets the transmission mode. If this field is set for Dual Mode or Quad Mode then either TX_TRANSFER_ENABLE or RX_TRANSFER_ENABLE **must** be 0. 3=Reserved 2=Quad Mode 1=Dual Mode 0=Single/Duplex Mode	R/W	0h	RESET

6.11.12 QMSPI DESCRIPTION BUFFER 1 REGISTER

The format for this register is the same as the format o the QMSPI Description Buffer 0 Register.

6.11.13 QMSPI DESCRIPTION BUFFER 2 REGISTER

The format for this register is the same as the format o the QMSPI Description Buffer 0 Register.

6.11.14 QMSPI DESCRIPTION BUFFER 3 REGISTER

The format for this register is the same as the format o the QMSPI Description Buffer 0 Register.

6.11.15 QMSPI DESCRIPTION BUFFER 4 REGISTER

The format for this register is the same as the format o the QMSPI Description Buffer 0 Register.

7.0 CHIP CONFIGURATION

7.1 Introduction

This chapter defines the mechanism to configure the device.

7.2 Terminology

This section documents terms used locally in this chapter. Common terminology that is used in the chip specification is captured in the Chip-Level Terminology section.

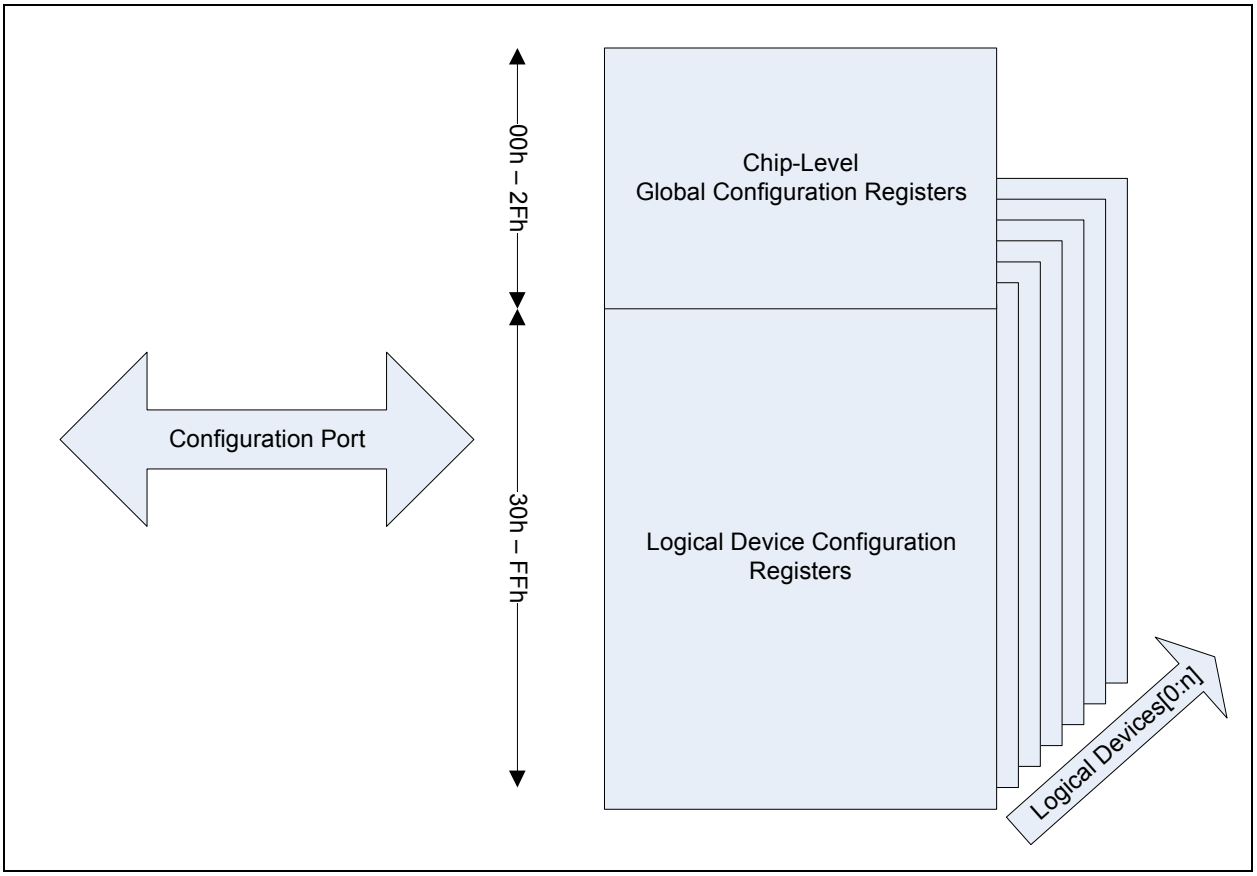
TABLE 7-1: TERMINOLOGY

Term	Definition
Global Configuration Registers	Registers used to configure the chip that are always accessible via the Configuration Port
Logical Device Configuration Registers	Registers used to configure a logical device in the chip. These registers are only accessible via the Configuration Port when enabled via the Global Configuration registers.

7.3 Interface

This block is designed to be accessed via the Host accessible Configuration Port.

FIGURE 7-1: BLOCK DIAGRAM OF CONFIGURATION PORT



Note:	Each logical device has a bank of Configuration registers that are accessible at offsets 30h to FFh via the Configuration Port. The Logical Device number programmed in offset 07h determines which bank of configuration registers is currently accessible.

7.3.1 HOST INTERFACE

The registers defined for the Chip Configuration are accessible by the Configuration Port when the device is in CONFIG MODE. For a description of the Configuration Port and CONFIG MODE see the description of the LPC Interface.

7.4 Power, Clocks and Reset

This section defines the Power, Clock, and Reset input parameters to this block.

7.4.1 POWER DOMAINS

Name	Description
VTR	The logic and registers implemented in this block reside on this single power well.

7.4.2 CLOCK INPUTS

This block does not require any special clock inputs.

7.4.3 RESETS

Name	Description
nSYSRST	Power on Reset to the block. This signal resets all the register and logic in this block to its default state.

7.5 Interrupts

This block does not generate any interrupts.

7.6 Low Power Modes

This block always automatically adjusts to operate in the lowest power mode.

7.7 Description

The Chip Configuration Registers are divided into two groups: Global Configuration Registers and Logical Device Configuration registers. The following descriptions assume that the LPC interface has already been configured to operate in CONFIG MODE.

- Global Configuration Registers are always accessible via the LPC Configuration Port.
- The Logical Device Configuration registers are only accessible via the LPC Configuration Port when the corresponding Logical Device Number is loaded in the Logical Device Number register. The Logical Device Number register is a Global Configuration Register.

There are 48 8-bit Global Configuration Registers (at offsets 00h through 2Fh), plus up to 208 8-bit registers associated with each Logical Device. The Logical Device is selected with the Logical Device Number Register (Global Configuration Register 07h).

Sequence to Access Logical Device Configuration Register:

a) Write the number of the Logical Device being accessed in the Logical Device Number Configuration Register by writing 07h into the INDEX PORT and the Logical Device Number into the DATA PORT.

b) Write the address of the desired logical device configuration register to the INDEX PORT and then write or read the value of the configuration register through the DATA PORT.

Note 1: If accessing the Global Configuration Registers, step (a) is not required.

2: Any write to an undefined or reserved Configuration register is terminated normally on the LPC bus without any modification of state in the MEC140x/1x. Any read to an undefined or reserved Configuration register returns FFh.

The following sections define the Global Configuration registers and the Logical Configuration registers.

7.7.1 GLOBAL CONTROL/CONFIGURATION REGISTERS

As with all Configuration Registers, the INDEX PORT is used to select a Global Configuration Register in the chip. The DATA PORT is then used to access the selected register. The INDEX and DATA PORTs are defined in the LPC Interface description.

TABLE 7-2: CHIP-LEVEL (GLOBAL) CONTROL/CONFIGURATION REGISTERS

Register	Offset	Description
Chip (Global) Control Registers		
Reserved	00h - 06h	Reserved - Writes are ignored, reads return 0.
Logical Device Number	07h	A write to this register selects the current logical device. This allows access to the control and configuration registers for each logical device. **Note:** The Activate command operates only on the selected logical device.
Reserved	08h - 1Bh	Reserved - Writes are ignored, reads return 0.
Device Revision	1Ch	A read-only register which provides device revision information. Bits[7:0] = current revision when read
Device Sub ID	1Dh	Device Sub ID[7:0] Read-Only register which provides the device sub-identification. The value of this register is product dependent. See Table 7-3, “Device Identification per Product,” on page 150.
Device ID[7:0]	1Eh	Device ID[7:0] Read-Only register which provides Device ID LSB. The value of this register is product dependent. See Table 7-3, “Device Identification per Product,” on page 150.
Device ID[15:8]	1Fh	Device ID[15:8] Read-Only register which provides Device ID MSB. The value of this register is product dependent. See Table 7-3, “Device Identification per Product,” on page 150.
Legacy Identification	20h	Legacy Identification A read-only register which provides device identification to legacy and test software. This field is hard-coded to FEh, indicating this is a MIPs product with 16-bit Device ID at offsets 1Eh & 1Fh.
Reserved	21h - 23h	Reserved.

TABLE 7-2: CHIP-LEVEL (GLOBAL) CONTROL/CONFIGURATION REGISTERS (CONTINUED)

Register	Offset	Description
Device Mode	24h	Bit [1:0] Reserved – writes ignored, reads return “0”. Bit[2] SerIRQ Mode) = 0: Serial IRQ Disabled. = 1: Serial IRQ Enabled (Default). Bit [7:3] Reserved – writes ignored, reads return “0”.
Test	25h - 2Fh	Test This register locations are reserved for Microchip use. Modifying these locations may cause unwanted results.

7.7.2 DEVICE IDENTIFICATION

TABLE 7-3: DEVICE IDENTIFICATION PER PRODUCT

Product	Device ID [15:0]	Device Sub ID [7:0]
MEC1404	0002h	10h
MEC1406	0004h	10h
MEC1408	0006h	10h
MEC1414	0008h	10h
MEC1416	000Ah	10h
MEC1418	000Ch	10h

7.7.3 LOGICAL DEVICE CONFIGURATION REGISTERS

The Logical Device Configuration registers support motherboard designs in which the resources required by their components are known and assigned by the BIOS at POST.

Each logical device may have a set of directly I/O addressable Runtime Registers, Configuration Registers accessible via the Configuration Port, or DMA registers. The following table lists the register types for each LPC Host-accessible Logical Device implemented in the design. The Embedded Controller (EC) can access all Configuration Registers and all Runtime Registers directly.

TABLE 7-4: HOST LOGICAL DEVICES ON MEC140X/1X

Logical Device Number (hex)	Logical Devices	LPC I/O Runtime Access	LPC I/O Configuration Access	eSPI I/O Runtime Access	eSPI I/O Configuration Access
0	EMI 0	yes	no	yes	no
1	8042 Emulated Keyboard Controller	no	yes	no	yes
3	ACPI EC0	yes	no	yes	no
4	ACPI EC1	yes	no	yes	no
5	ACPI PM1	yes	no	yes	no
6	Legacy Port92/GateA20	yes	yes	yes	yes
7	UART 0	yes	yes	yes	yes
9	Mailbox Interface	yes	no	yes	no
A	ACPI EC2	yes	no	yes	no
B	ACPI EC3	yes	no	yes	no
C	LPC Interface (Configuration Port)	yes	yes	no	no
10	eSPI I/O Component (Configuration Port)	no	no	yes	yes
12	eSPI Memory Component	no	no	yes	yes
15	Port 80 BIOS Debug Port 0	yes	yes	yes	yes
16	Port 80 BIOS Debug Port 1	yes	yes	yes	yes

8.0 MIPS32 M14K EMBEDDED CONTROLLER

8.1 Features

- A Modified 5-stage pipelined Harvard Architecture with a Closely-Coupled Data Memory and Instruction Memory interfaces
- Single Cycle 32-bit instruction set
- microMIPS-Compatible Instruction Set (default)
  - microMIPS supports all MIPS32 instructions (except branch-likely instructions)
  - Stack pointer implicit in instruction
  - MIPS32 assembly and ABI (Application Binary Interface) compatible.
- External Interrupt Controller (EIC) mode.
  - Programmable exception vector base
  - Atomic interrupt enable/disable
  - Bit field manipulation instructions
- Simple Fixed Mapping Translation (FMT) mechanism
- Multiply/Divide Unit (high-performance configuration)
  - Maximum issue rate of one 32x16 multiply per clock via on-chip 32x16 hardware multiplier array.
  - Maximum issue rate of one 32x32 multiply every other clock
  - Early-in iterative divide. Minimum 11 and maximum 34 clock latency (dividend (*rs*) sign extension-dependent)
- Power Control
  - Programmable Clock Rates: 48 MHz, 24 MHz, 3 MHz, and 1 MHz
  - Sleep mode: Minimum frequency: 0 MHz
  - Power-down mode (triggered by WAIT instruction)
  - Clocks are gated in Low Power Modes
- EJTAG Debug Mechanism
  - CPU control with start, stop, and single-stepping
  - Virtual instruction and data address/value breakpoints
  - Hardware breakpoint supports both address match and address range triggering.
  - Simple hardware breakpoints on virtual addresses: 4I/2D breakpoints
  - PC/Address Sampling function
  - Support EJTAG (IEEE 1149.1)
  - Supported by MPLAB REAL ICE tools

8.2 References

1. MIPS32® M14K™ Processor Core Software User's Manual, Document Number: MD00668, Revision 02.03, April 30, 2012
2. MIPS32® M14K™ Processor Core Data Sheet, Document Number MD00666, Revision 2.03, April 30, 2012
3. MIPS32® M14K™ Architecture for Programmers Volume I-B: Introduction to the microMIPS32™ Architecture, Document Number MD00741, Revision 3.02, March 21, 2011
4. MIPS32® M14K™ Architecture for Programmers Volume II-B: The microMIPS32™ Instruction Set, Document Number MD00582, Revision 3.05, April 04, 2011
5. MIPS® EJTAG Specification, Document Number MD00047, Revision 5.06, March 05, 2011

Note: Resources for the MIPS32® M4K™ Processor Core are available at: www.imgtec.com.

8.3 Terminology

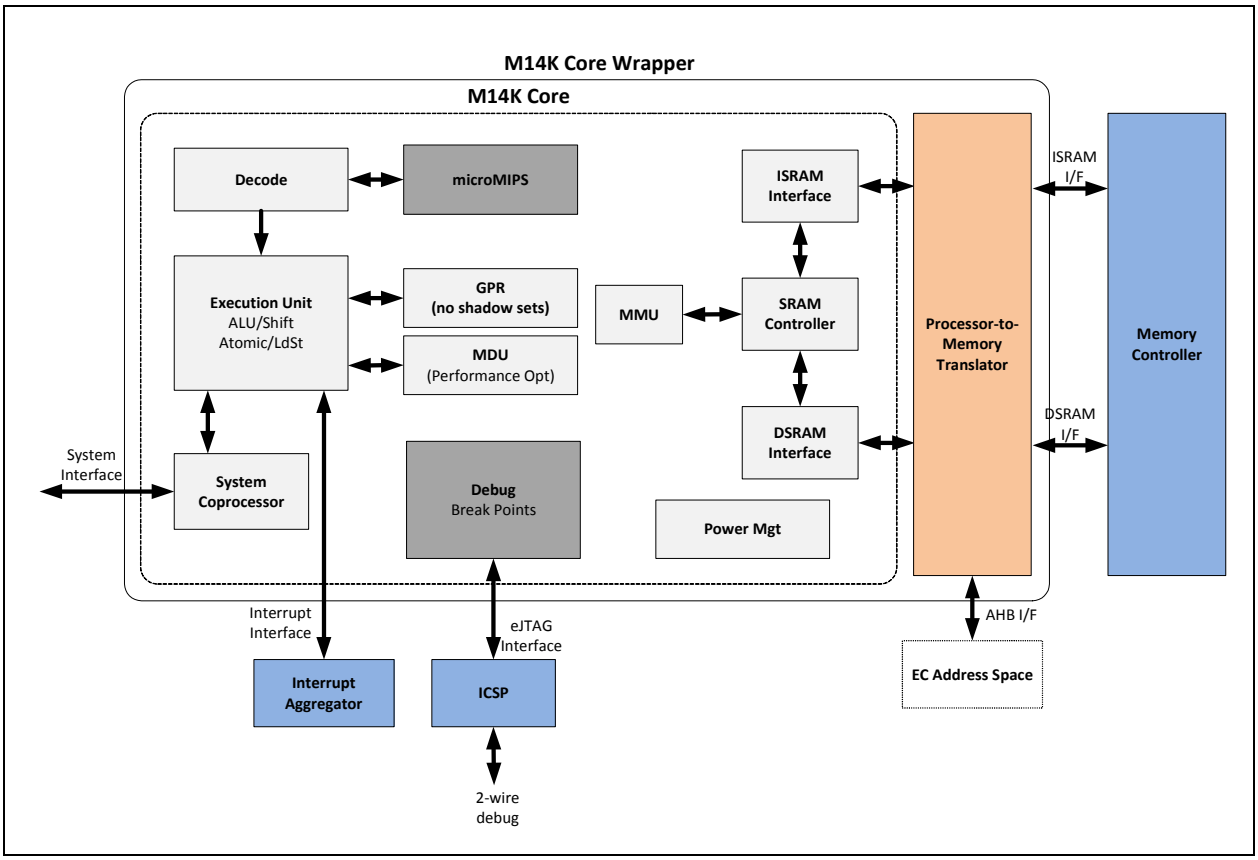
There is no terminology defined for this chapter.

8.4 Interfaces

The Embedded Controller (EC) has five interfaces: ISRAM Interface, DSRAM Interface, Debug (EJTAG) Interface, AHB System Interface, and an Interrupt Interface.

The EC executes instruction out of instruction memory (e.g., ROM) or data memory (e.g., RAM) via the ISRAM Interface; memory accesses are handled via the DSRAM Interface; and EC accesses the peripherals residing in the internal address space via the AHB interface. The host can probe the EC and all EC addressable memory via the eJTAG debug interface.

FIGURE 8-1: MIPS32 M14K EMBEDDED CONTROLLER I/O BLOCK DIAGRAM



Note:	Blocks in the diagram that are external to the M14K Core and are highlighted in blue are defined in their respective chapters.

8.4.1 EJTAG INTERFACE

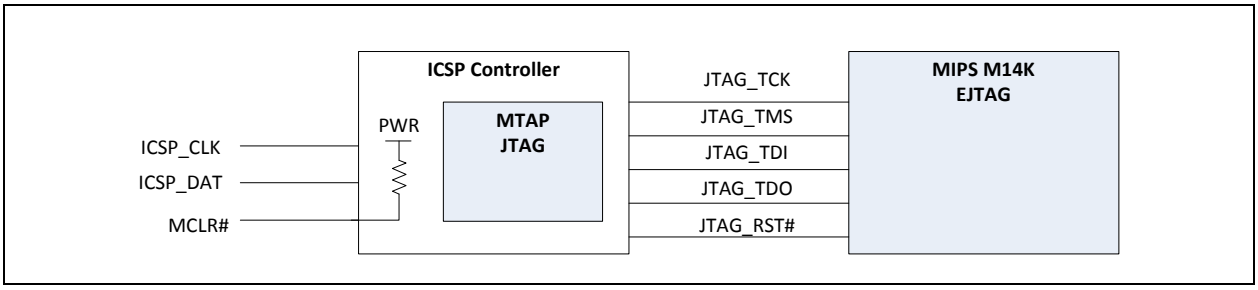
TABLE 8-1: EJTAG SIGNAL DESCRIPTION

Signal Name	Pin Name	Direction	Description
TCK	JTAG_CLK	Input	Test Clock
TMS	JTAG_TMS	Input	Test Mode Select
TDI	JTAG_TDI	Input	Test Data In
TDO	JTAG_TDO	Output	Test Data Out
TRST#	JTAG_RST#	Input	Test Reset, low active

8.4.1.1 Mapping ICSP to EJTAG Interface

The JTAG debug interface signals are connected internally to the ICSP block. The ICSP block converts the 2-wire ICSP interface into standard EJTAG signaling. port that is connected to the external pin interface.

FIGURE 8-2: ICSP-TO-EJTAG



Note: The MCLR# is pulled up internally and requires no external logic.

For a description of the ICSP Controller see Section 41.4, "ICSP Controller," on page 475.

8.4.2 AHB INTERFACE

A Processor-to-Memory Translator has been appended to the ISRAM and DSRAM interfaces. This translator will pass traffic to either the ISRAM, DSRAM, or AHB interface based on the address of the access. The AHB Interface is the embedded controller's interface to the EC Address Space (i.e., 32-bit internal address space) that is not used as EC Code or Data space (e.g., Peripheral Registers).

The MIPS32 M14K core can have at most one access pending on the AHB at one time. It can perform 8-bit, 16-bit and 32-bit loads and stores on the AHB.

Possible AHB bus errors are described in Section 8.4.2.1, "AHB & Code/Data Bus Errors," on page 154. The processor responds to a bus error with Memory Error exception, except where noted.

8.4.2.1 AHB & Code/Data Bus Errors

AHB bus requests can be terminated with an AHB bus error. The handling of bus errors by the EC is described in Chapter 4, Exceptions and Interrupts in the M14K Core, of the MIPS32® M14K™ Processor Core Software User's Manual, Document Number: MD00668, Revision 02.03, April 30, 2012.

Bus errors may be caused by:

- Code accesses to a memory location outside of the Code/ROM memory range will generate a processor exception
- Data accesses to out-of-bounds memory location in data region (0xBFD18000 - 0xBFFF_FFFF) returns garbage (no processor exception).
- EC I/O requests to undefined EC Address memory locations via the System AHB Interface.will generate a processor exception

8.4.3 SYSTEM INTERFACE

TABLE 8-2: SYSTEM INTERFACE SIGNAL DESCRIPTION

Signal Name	Direction	Description	Connected at Chip-Level
SI_RP	Output	The SI_RP signal represents the state of the RP bit (27) in the CP0 Status register. This signal may be used at the chip-level to decide whether to enter a lower power state.	No
SI-EXL	Output	The SI_EXL signal represents the state of the EXL bit (1) in the CP0 Status register. This signal may be used for throttling the clock after a wake event.	No
SI_ERL	Output	The SI_ERL signal represents the state of the ERL bit (2) in the CP0 Status register. This signal indicates an error has occurred.	No
EJ_DebugM	Output	The EJ_DebugM signal indicates that the processor has entered debug mode.	Yes

8.4.4 ISRAM INTERFACE

The ISRAM interface is the embedded controller's instruction fetch interface. Code Instructions may be executed from the Instruction Memory or the Data Memory.

8.4.5 DSRAM INTERFACE

The DSRAM Interface is the embedded controller's data interface, which can access both the Data Memory and the Instruction Memory (literals).

8.4.6 INTERRUPT INTERFACE

The MIPS32® M14K™ Embedded Controller is configured for External Interrupt Controller (EIC) mode.

The interrupts implemented on this chip are defined in Section 10.0, "Jump Table Vectored Interrupt Controller (JTVIC)," on page 159. The interrupt unit generates interrupt requests (IRQs) to the CPU and has the ability to bring the CPU out of sleep mode when a valid wake-capable interrupt request is present.

All interrupts can either be pulse or level triggered as well as having individual mask bits and priority levels.

8.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

8.5.1 POWER DOMAINS

Name	DESCRIPTION
VTR	The embedded controller is powered by VTR.

8.5.2 CLOCK INPUTS

Name	DESCRIPTION
EC_PROC_CLK	The EC clock is the clock source to the embedded controller. **Note:** The EC clock can be throttled up or down externally by the chip's Power, Clocks, and Reset (PCR) circuitry.

8.5.3 RESETS

Name	DESCRIPTION
EC_PROC_RESET#	The embedded controller is reset by EC_PROC_RESET#.

8.6 Interrupts

The embedded controller does not generate any interrupts.

Note: The embedded controller is equipped with an Interrupt Interface to respond to interrupts. See Section 8.4.6, "Interrupt Interface," on page 155.

8.7 Exceptions

Exceptions are synchronous to instructions, are not maskable, and have higher priority than interrupts.

Name	Description
Reset_Exception	The Reset_Exception is asserted when either an SI_RESET (i.e., Soft Reset) or a SI_ColdReset (i.e., POR) is asserted. Events that can cause a SI_RESET are a Soft Reset initiated by firmware or a WDT Event.
Debug_Exception	The Debug_Exception is asserted for an EJTAG command.
NMI	None - There are no NMI's implemented in this device.

8.8 Low Power Modes

The embedded controller may put itself and the chip into lower power states by configuring the chip's Sleep logic implemented in the chip's Power, Clocks, and Reset (PCR) circuitry and then executing the WAIT instruction.

The core provides two mechanisms for system-level, low-power support: Register-controlled power management and Instruction-controlled power management

8.8.1 REGISTER-CONTROLLED POWER MANAGEMENT

Register-Controlled Power Management is not supported.

8.8.2 INSTRUCTION-CONTROLLED POWER MANAGEMENT

In instruction-controlled power-down mode execution of the WAIT instruction is used to invoke low-power mode and put the chip into sleep mode. It stays in sleep mode until an interrupt or restart occurs. Power consumption is reduced during sleep mode since the pipeline ceases to change state, and the RAMs are disabled. More power reduction is achieved when clock gating option is used, whereby all non-essential clocks are switched off. The chip's Power, Clocks, and Reset (PCR) circuitry may be enabled to gate the clocks externally to the core when the embedded controller enters the sleep state.

8.9 Description

The block diagram shown in FIGURE 8-1: MIPS32 M14K Embedded Controller I/O Block Diagram on page 153 illustrates the IP configuration selected. This EC design includes the Fixed/Required M14K features, such as the Decode, Execution Unit, etc that are shaded light gray. The EC design has also opted to include the microMIPs instruction set and Debug capabilities. All other optional features have not been implemented.

The following sections define the optional features and configuration options selected. This chapter is intended to be used in combination with the MIPS documentation, such as the MIPS32® M14K™ Processor Core Software User's Manual, listed in the Section 8.2, "References," on page 152.

8.9.1 POWER ON RESET

Following a power on reset event the EC_PROC_RESET# signal is de-asserted and the embedded controller starts executing code from the first physical address of the Boot ROM.

8.9.2 INSTRUCTION SET

The M14K core defaults to the microMIPS instruction set and is runtime configurable as either microMIPS Instruction set.

This device does not support the following atomic instructions. A critical section should be used instead of these instructions. NOTE: A critical section will not protect a memory location from DMA access.

LL – Load Linked Word. LL and SC must be used together to implement an atomic transaction.

SC – Store Conditional Word

ACLR – Atomically Clear Bit within Byte

ASET - Atomically Set Bit within Byte

The device does not support the following interrupt return instruction. This instruction requires additional shadow register set. Use ERET instead.

IRET – Interrupt Return with automated interrupt epilog handling.

8.9.3 EJTAG HARDWARE DEBUG BREAK POINTS

This M14K core is configured for two data and four instruction breakpoints, without complex breakpoints

8.9.4 GENERAL PURPOSED REGISTER (GPR) SHADOW REGISTERS

The M14K core contains thirty-two 32-bit general-purpose registers used for integer operations and address calculation. No optional register sets were implemented.

8.9.5 MULTIPLY/DIVIDE UNIT (MDU)

This device is configured for the higher performance 32x16 array option.

8.9.6 SYSTEM CONTROL COPROCESSOR (CP0)

8.9.6.1 System Interface

The System Interface signals are defined in the Interfaces section. See Section 8.4.3, "System Interface," on page 154.

8.9.6.2 Interrupt Handling

This device is configured for External Interrupt Controller (EIC) mode.

8.9.7 MEMORY MANAGEMENT UNIT (MMU)

The M14K core implements a simple Fixed Mapping (FM) memory management unit.

9.0 MEMORY ORGANIZATION

The MEC140x/1x implements two address spaces: Virtual and Physical. All hardware resources such as program memory, data memory and peripherals are located at their respective physical addresses. Virtual addresses are exclusively used by the CPU to fetch and execute instructions as well as access peripherals. Physical addresses are used by peripherals such as the Internal DMA Controller that access memory independently of CPU.

The following table lists all the defined memory regions in the 4 GB EC Address Space. Accessing undefined memory regions may cause unwanted results, such as a memory exception.

TABLE 9-1: EC ADDRESS SPACE

Location	Space	Virtual Start Address	Physical Start Address	Physical End Address	Size
CC-MMCR (Note 1)	KSEG1	0xBFFF_C000	0x1FFF_C000	0x1FFF_FFFF	16 kB
Data RAM	KSEG1	0xBFD1_8000	0x1FD1_8000	0x1FD1_FFFF	32 kB (Note 4)
Code RAM (Note 3)	KSEG1	0xBFD0_0000 0xBFCF_8000 0xBFCF_0000	0x1FD0_0000 0x1FCF_8000 0x1FCF_0000	0x1FD1_7FFF	96 kB 128 kB 160 kB (Note 5)
Boot ROM	KSEG1	0xBFC0_0000	0x1FC0_0000	0x1FC0_FFFF	64 kB
MMCR (Note 2)	KSEG1	0xA000_0000	0x0000_0000	0x001F_FFFF	2 MB

Note 1: CC-MMCR = closely-coupled memory-mapped control registers, i.e. interrupt registers (JTVIC).

2: MMCR = memory-mapped control registers, i.e. all the peripheral registers.

3: The IRQ EBASE must be programmed at BFD0_0000h in order to be on a 256k byte boundary. All IRQ routine entry points must be located above this address.

4: 32kB is the default Data RAM size; however, other sizes of Data RAM can be used (for example 8kB or 16kB) with the remainder used as Code RAM. See the MEC14xx Programmers Reference Guide for configuring the different settings.

5: The size of the code RAM is part dependent.

The embedded controller executes code out of the EC Instruction Memory via the closely-coupled ISRAM Interface. The Code RAM, Boot ROM and Debug RAM are all accessible as EC Instruction Memory. Data references can come from either the EC Data Memory via the closely-coupled DSRAM Interface (i.e., Data RAM access) or from any address located in the EC Address Space via the System Interface.

The Code and Data SRAM is optimized to the memory allocation shown in the table. This allows code and data accesses to happen simultaneously. However, software may use Code RAM for data and Data RAM for code. The only penalty will be access time. When the ISRAM and DSRAM interfaces both attempt to access the same memory region the accesses become serialized.

Example:

The 128KBytes SRAM (Code or Data) memory is allocated as follows:

- 96 kB Optimized for Code
- 32 kB Optimized for Data.

A user may choose to organize their code and data space as follows:

STACK	8 kB
DATA	20 kB
CODE	100 kB

Notice that although the Code Space is optimized for 96 kB the user can choose to allocate part of the data memory for code. The only difference will be the access time for the code implemented in the data space since code and data accesses will become serialized in that range.

10.0 JUMP TABLE VECTORED INTERRUPT CONTROLLER (JTVIC)

10.1 Overview

The Jump Table Vectored Interrupt Controller (JTVIC) works in conjunction with the MIPS32® M14K™ Processor Interrupt Interface. The interrupt events are synchronous events that may be serviced in either Aggregated Mode or Disaggregated mode. The JTVIC block presents the Vector for the highest priority interrupt pending. The priority-level is firmware selectable.

A subset of the interrupts are classified as wake events that can be recognized without a running clock, e.g., while the MEC140x/1x is in sleep state. These asynchronous events are routed to the chip's clock generation logic and are used to resume the clock's operation from a sleep state and wake the processor.

10.2 References

- MIPS32® M14K™ Processor Core Data Sheet, April 30, 2012.
- MIPS32® M14K™ Software Users Manual, Document Number: MD00668, Revision 02.03, April 30, 2012.
- MIPS32® M14K™ Integrator's Guide, Document Number: MD00667, Revision 02.03, April 30, 2012.

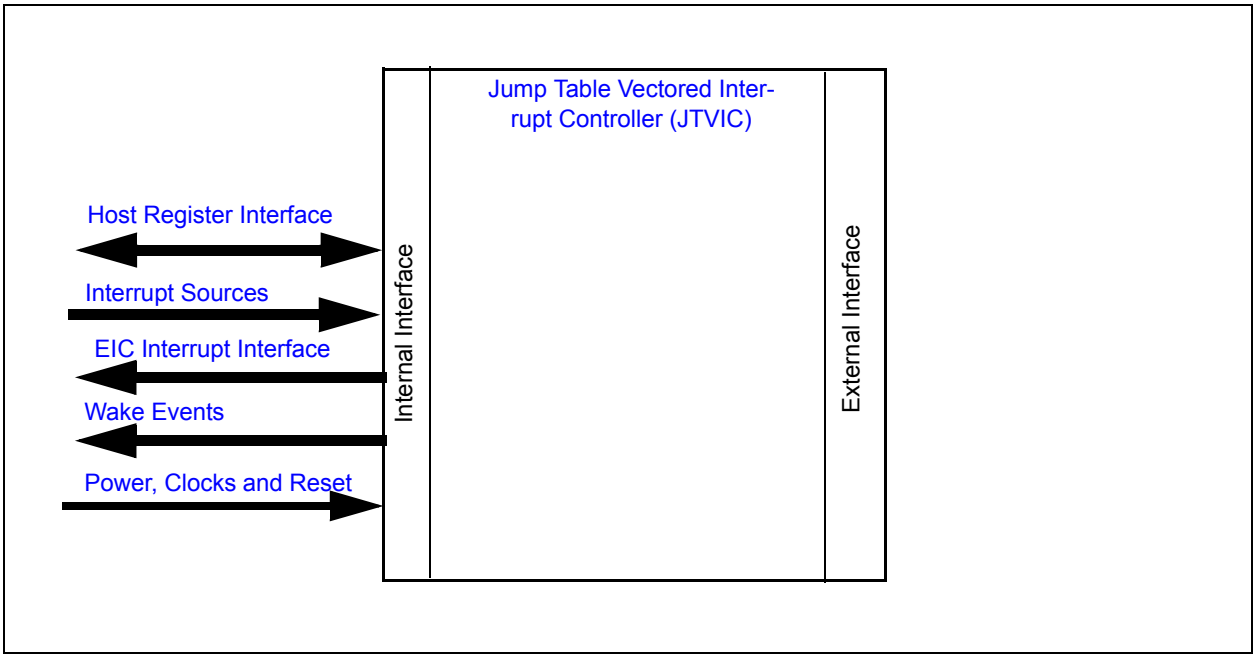
10.3 Terminology

Term	Definition
IPL	Interrupt Priority Level
PIPL	Pending Interrupt Priority Level

10.4 Interface

This block is designed to be accessed internally via a registered host interface.

FIGURE 10-1: I/O DIAGRAM OF BLOCK



10.5 Host Register Interface

The registers defined for the Jump Table Vectored Interrupt Controller (JTVIC) Interface are accessible by the various hosts as indicated in Section 10.12, "JTVIC Registers".

10.6 Interrupt Sources

All the chip's interrupt sources are routed to the Jump Table Vectored Interrupt Controller (JTVIC) GIRQx Source Registers. The list of interrupt sources is defined in Table 10-2, "Interrupt Source, Enable Set, Enable Clear, and Result Bit Assignments," on page 164.

10.7 EIC Interrupt Interface

The Jump Table Vectored Interrupt Controller (JTVIC) is designed to generate interrupts to the Embedded Controller's External Interrupt Controller (EIC) interface. This IP block aggregates all the chip's interrupt Sources (defined in Table 10-2, "Interrupt Source, Enable Set, Enable Clear, and Result Bit Assignments," on page 164), determines the highest priority interrupt that is active, and generates the Offset Vector used to jump to the respective IRQ Handler.

10.7.1 EIC INTERRUPT SIGNALS

Name	Direction	Description
Interrupt Request	Output	Signal to the processor that an interrupt request is pending
Vector_Address	Output	Offset appended to processor EBASE address to create pointer to IRQ handler. **Note:** The processor EBASE must be programmed on a 256k Byte boundary.
RIPL	Output	Requested Interrupt Priority Level.

10.8 Wake Events

All interrupt sources that indicate they are wake-capable generate an asynchronous wake event to the chip's sleep control logic to restore the oscillator to the fully operational state. Wake-capable signals do not require the internal oscillator to be running.

10.9 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

10.9.1 POWER

Name	Description
VTR	The logic and registers implemented in this block are powered by this power well.

10.9.2 CLOCKS

Name	Description
48 MHz Ring Oscillator	Clock used for register read/write access.

10.9.3 RESETS

Name	Description
nSYSRST	This signal resets all the registers and logic in this block to their default state.

10.10 Low Power Modes

The JTVIC always operates in the lowest power state; gating its own clock when it is not required. The only time this block requires the 48 MHz Ring Oscillator is for register reads/writes and for propagating interrupt events to the embedded controller.

If the 48 MHz Ring Oscillator is off, the wake-capable interrupts may be used to resume operation thereby allowing the interrupt events to propagate to the embedded controller.

10.11 Description

10.11.1 FEATURES

- Supports up to 1024 Interrupt Sources
- Aggregated and Disaggregated Modes of Operation
  - Aggregated Mode offers a programmable Vector Offset per GIRQ
  - Disaggregated Mode offers a programmable Vector Offset per Source Bit
- 4 levels of configurable priority

10.11.2 OVERVIEW

This module is a highly-configurable and expandable vectored interrupt controller which is designed to work with an MIPS M14k processor's EIC (External Interrupt Controller) mode of interrupt operation, with direct vector addressing (i.e. direct address driven into the processor instead of an "interrupt vector number"). The controller supports four levels of priority on a per-interrupt-source basis.

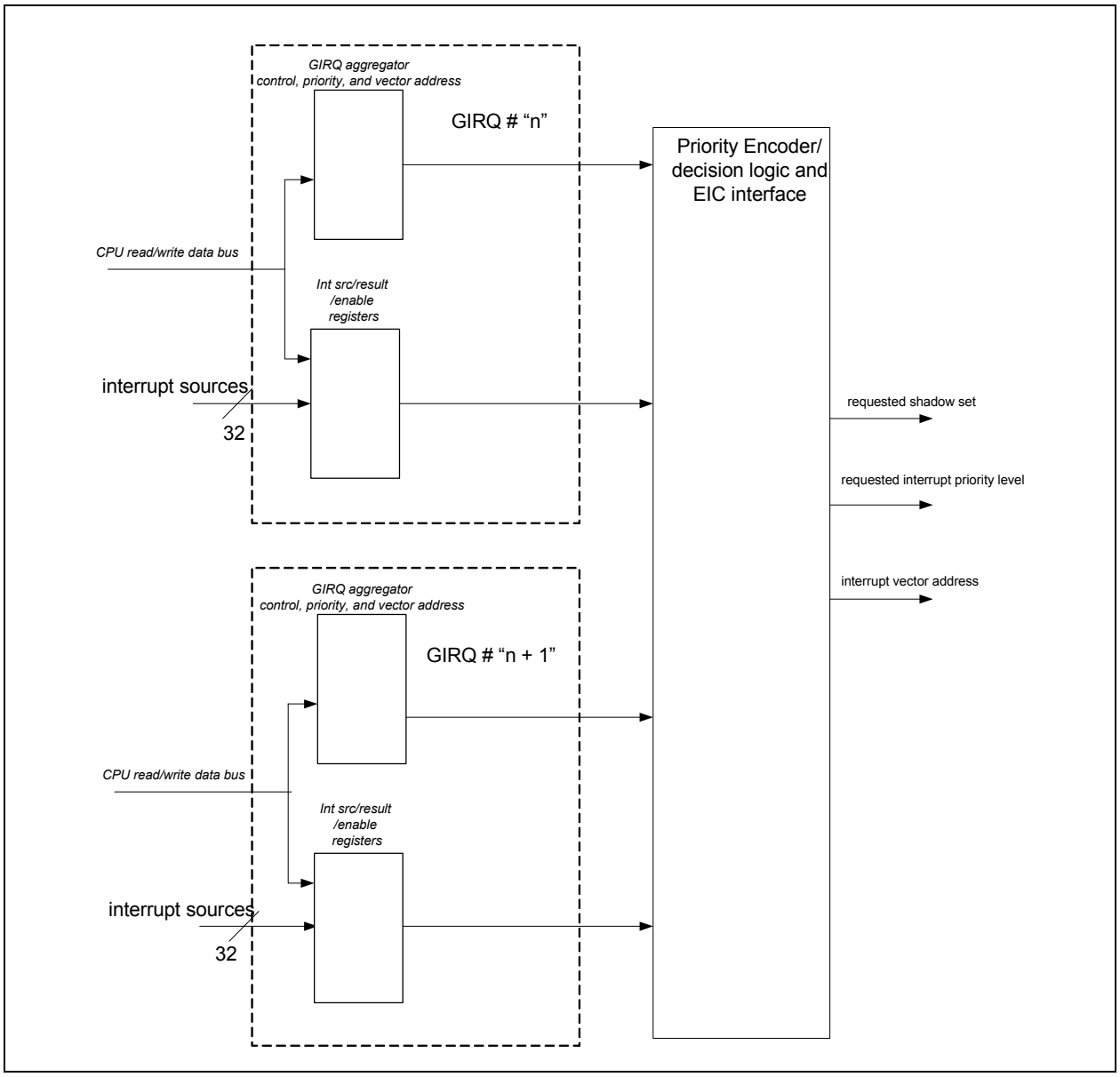
The controller operates in two different modes, aggregated and dis-aggregated (or mini-jump-table), on a grouped-IRQ (GIRQ) basis. NOTE: a GIRQ is a grouping of up to 32 interrupt sources.

Thus this controller can be configured as fully aggregated all the way to fully dis-aggregated, and everything in between.

In aggregated mode the controller stores ISR vector addresses in local registers, thus saving firmware from having to build ISR jump tables in local SRAM. One vector address per GIRQ.

In dis-aggregated/jump-table mode, the controller can selectively break apart individual GIRQ interrupt sources into separate vector addresses.

FIGURE 10-2: TOP-LEVEL BLOCK DIAGRAM OF INTERRUPT GENERATION LOGIC



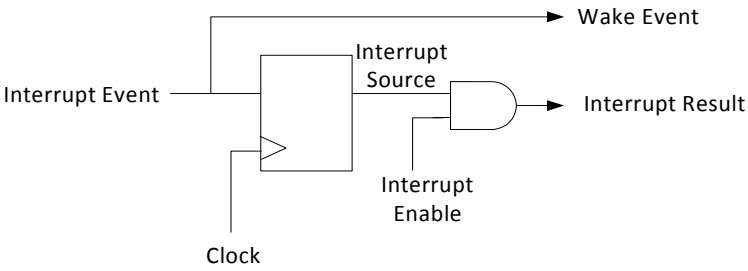
10.11.3 WAKE-CAPABLE INTERRUPT EVENTS

Wake-capable interrupts are listed in Section 10.11.4, "List of Interrupt Events," on page 164 with a designation of ‘Yes’ in the Wake Event column

All interrupts, except GIRQ22, generate an EC Interrupt event. They are routed to source bits that are synchronized to the 48 MHz Ring Oscillator. If enabled, the Interrupt Result is fed into the Priority Encoder/Decision Logic, which generates the interrupt vector to the EIC Interrupt Interface.

Some Interrupts, which are labeled Wake-Capable, are also routed as Wake Events to the Chip’s Wake Logic. These are asynchronous events that are used to resume the 48 MHz Ring Oscillator operation from a sleep state and wake the processor.

FIGURE 10-3: INTERRUPT SOURCE, ENABLE, AND RESULT LOGIC



10.11.3.1 GIRQ16 and GIRQ22 Wake-Only Events

GIRQ16 and GIRQ22 are reserved for Wake-Only events that do not require functional software service.

TABLE 10-1: WAKE-ONLY EVENTS

Wake Event	Description
LPC_WAKE	This bit is set when the LPC interface detects activity on the interface. It's sole purpose is to restart the 48 MHz Ring Oscillator.
SMB_WAKE	This bit is set when an i2c/SMBus interface detects a START event on the interface. It's sole purpose is to restart the 48 MHz Ring Oscillator.
PS2_DATx_WAKE	This bit is set when the PS/2 interface detects activity on it's interface. It's sole purpose is to restart the 48 MHz Ring Oscillator.
KSC_INT_WAKE	This bit is set when the Keyboard Matrix Scan Controller detects activity on it's interface. It's sole purpose is to restart the 48 MHz Ring Oscillator.
DEBUG_DONE	This bit is set when the ICSP debugger interface detects activity on the interface. It's sole purpose is to notify the EC firmware that the 48 MHz Ring Oscillator was taken out of sleep state by the debug interface.
ESPI_WAKE	This bit is set when the eSPI interface detects activity on the interface. It's sole purpose is to restart the 48 MHz Ring Oscillator.

GIRQ16 will generate both a wake event and an interrupt vector to the EIC Interrupt Interface. This will require the embedded firmware to clear the interrupt status event and re-execute the sleep instruction. GIRQ16 is a legacy interrupt used to ensure the 48 MHz Ring Oscillator remained on for the minimum time. This interrupt may be deprecated in future designs

GIRQ22 does not generate an interrupt vector to the EIC Interrupt Interface. GIRQ22 only generates a wake event to restart the 48 MHz Ring Oscillator running. Hardware automatically wakes the oscillator to process the wake event, clears the event, and resumes sleeping without firmware intervention.

Note: The sleeping state of the chip is determined by bits[2:0] of the System Sleep Control Register (SYS_SLP_CNTRL) on page 81.

APPLICATION NOTE: Configuring Wake-Only Events

Wake-Only interrupt event should be enabled just before executing the EC sleep instruction. Firmware should execute the following sequence of events:

1. Set bits[2:0] in the System Sleep Control Register (SYS_SLP_CNTRL)
2. Enable Wake Events in either GIRQ16 or GIRQ22
3. Execute Sleep Instruction (_wait;)

For example, in order to enable LPC transactions to MEC140x/1x Logical Devices while the MEC140x/1x is in a Sleep mode in which the main oscillator is shut off, just before entering sleep EC firmware must enable one of the LPC_WAKE interrupts. The firmware designer may choose either the LPC_WAKE located in GIRQ16 or in GIRQ22. When responding to the GIRQ16 interrupt EC firmware should disable the LPC_WAKE interrupt until firmware determines that it is again appropriate to enter a Deep Sleep mode. GIRQ22 handles this automatically in hardware.

10.11.4 LIST OF INTERRUPT EVENTS

The following table lists all the Interrupt Source, Enable, and Result bits and indicates if they are wake-capable.

TABLE 10-2: INTERRUPT SOURCE, ENABLE SET, ENABLE CLEAR, AND RESULT BIT ASSIGNMENTS

Aggregator IRQ	Aggregator Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description
GIRQ8	0	GPIO140	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	1	GPIO141	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	2	GPIO142	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	3	GPIO143	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	4	GPIO144	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	5	GPIO145	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	6	GPIO146	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	7	GPIO147	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	8	GPIO150	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	9	GPIO151	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	10	GPIO152	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	11	GPIO153	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	12	GPIO154	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	13	GPIO155	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	14	GPIO156	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	15	GPIO157	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	16	GPIO160	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	17	GPIO161	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	18	GPIO162	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	19	GPIO163	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	20	GPIO164	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	21	GPIO165	GPIO Event	Yes	GPIO Interrupt Event
GIRQ8	22	GPIO166	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	0	GPIO100	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	1	GPIO101	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	2	GPIO102	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	3	GPIO103	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	4	GPIO104	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	5	GPIO105	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	6	GPIO106	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	7	GPIO107	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	8	GPIO110	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	9	GPIO111	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	10	GPIO112	GPIO Event	Yes	GPIO Interrupt Event

TABLE 10-2: INTERRUPT SOURCE, ENABLE SET, ENABLE CLEAR, AND RESULT BIT ASSIGNMENTS (CONTINUED)

Aggregator IRQ	Aggregator Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description
GIRQ9	11	GPIO113	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	12	GPIO114	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	13	GPIO115	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	14	GPIO116	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	15	GPIO117	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	16	GPIO120	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	17	GPIO121	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	18	GPIO122	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	19	GPIO123	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	20	GPIO124	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	21	GPIO125	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	22	GPIO126	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	23	GPIO127	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	24	GPIO130	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	25	GPIO131	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	26	GPIO132	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	27	GPIO133	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	28	GPIO134	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	29	GPIO135	GPIO Event	Yes	GPIO Interrupt Event
GIRQ9	30	GPIO136	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	0	GPIO040	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	1	GPIO041	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	2	GPIO042	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	3	GPIO043	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	4	GPIO044	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	5	GPIO045	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	6	GPIO046	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	7	GPIO047	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	8	GPIO050	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	9	GPIO051	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	10	GPIO052	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	11	GPIO053	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	12	GPIO054	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	13	GPIO055	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	14	GPIO056	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	15	GPIO057	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	16	GPIO060	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	17	GPIO061	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	18	GPIO062	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	19	GPIO063	GPIO Event	Yes	GPIO Interrupt Event
GIRQ10	20	GPIO064	GPIO Event	Yes	GPIO Interrupt Event

TABLE 10-2: INTERRUPT SOURCE, ENABLE SET, ENABLE CLEAR, AND RESULT BIT ASSIGNMENTS (CONTINUED)

Aggregator IRQ	Aggregator Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description
GIRQ10	21-22	Test	Test	-	-
GIRQ10	23	GPIO067	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	1	GPIO001	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	2	GPIO002	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	3	GPIO003	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	4	GPIO004	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	5	GPIO005	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	6	GPIO006	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	7	GPIO007	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	8	GPIO010	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	9	GPIO011	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	10	GPIO012	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	11	GPIO013	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	12	GPIO014	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	13	GPIO015	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	14	GPIO016	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	15	GPIO017	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	16	GPIO020	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	17	GPIO021	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	18	GPIO022	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	19	GPIO023	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	20	GPIO024	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	21	GPIO025	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	22	GPIO026	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	23	GPIO027	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	24	GPIO030	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	25	GPIO031	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	26	GPIO032	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	27	GPIO033	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	28	GPIO034	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	29	GPIO035	GPIO Event	Yes	GPIO Interrupt Event
GIRQ11	30	GPIO036	GPIO Event	Yes	GPIO Interrupt Event
GIRQ12	0	SMBus Controller 0	SMB	No	SMBus Controller 0 Interrupt Event
GIRQ12	1	SMBus Controller 1	SMB	No	SMBus Controller 1 Interrupt Event
GIRQ12	2	SMBus Controller 2	SMB	No	SMBus Controller 2 Interrupt Event
GIRQ13	0	DMA Controller	DMA0	No	DMA Controller - Channel 0 Interrupt Event
GIRQ13	1	DMA Controller	DMA1	No	DMA Controller - Channel 1 Interrupt Event

TABLE 10-2: INTERRUPT SOURCE, ENABLE SET, ENABLE CLEAR, AND RESULT BIT ASSIGNMENTS (CONTINUED)

Aggregator IRQ	Aggregator Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description
GIRQ13	2	DMA Controller	DMA2	No	DMA Controller - Channel 2 Interrupt Event
GIRQ13	3	DMA Controller	DMA3	No	DMA Controller - Channel 3 Interrupt Event
GIRQ13	4	DMA Controller	DMA4	No	DMA Controller - Channel 4 Interrupt Event
GIRQ13	5	DMA Controller	DMA5	No	DMA Controller - Channel 5 Interrupt Event
GIRQ13	6	DMA Controller	DMA6	No	DMA Controller - Channel 5 Interrupt Event
GIRQ14	0	LPC Interface	LPC_INTERNAL_ERR	No	LPC Internal Error Event
GIRQ14	1	Power, Clocks, and Resets	PFR_Status	No	Power-Fail and Reset Status Register Events
GIRQ14	2	Blinking/Breathing LED 0	PWM_WDT	No	Blinking/Breathing LED 0 - Watchdog Event
GIRQ14	3	Blinking/Breathing LED 1	PWM_WDT	No	Blinking/Breathing LED 1 - Watchdog Event
GIRQ14	4	Blinking/Breathing LED 2	PWM_WDT	No	Blinking/Breathing LED 2 - Watchdog Event
GIRQ14	5	Internal 32KHz	INT_32K_RDY	No	Internal 32 KHz oscillator ready flag
GIRQ15	0	Mailbox Register Interface	MBX Host-to-EC	No	Mailbox Register Interface - Host-to-EC Interrupt Event
GIRQ15	1	Reserved	Reserved	-	-
GIRQ15	2	EMI 0	Host-to-EC	No	Embedded Memory Interface 0 - Host-to-EC Interrupt Event
GIRQ15	3	Reserved	Reserved	-	-
GIRQ15	4	8042 Emulated Keyboard Controller	OBF	No	8042 Emulated Keyboard Controller - Output Buffer Full Event
GIRQ15	5	8042 Emulated Keyboard Controller	IBF	No	8042 Emulated Keyboard Controller - Input Buffer Full Event
GIRQ15	6	Port 80 BIOS Debug Port 0	BDP_INT	No	Port 80h BIOS Debug Port Event
GIRQ15	7	Port 80 BIOS Debug Port 1	BDP_INT	No	Port 80h BIOS Debug Port Event
GIRQ15	8	ACPI_PM1 Interface	PM1_CTL	No	ACPI_PM1 Interface - PM1_CTL2 Interrupt Event
GIRQ15	9	ACPI_PM1 Interface	PM1_EN	No	ACPI_PM1 Interface - PM1_EN2 Interrupt Event
GIRQ15	10	ACPI_PM1 Interface	PM1_STS	No	ACPI_PM1 Interface - PM1_STS2 Interrupt Event
GIRQ15	11	ACPI_EC Interface 0	IBF	No	ACPI EC Interface 0 - Input Buffer Full Event
GIRQ15	12	ACPI_EC Interface 0	OBF	No	ACPI EC Interface 0 - Output Buffer Full Event

TABLE 10-2: INTERRUPT SOURCE, ENABLE SET, ENABLE CLEAR, AND RESULT BIT ASSIGNMENTS (CONTINUED)

Aggregator IRQ	Aggregator Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description
GIRQ15	13	ACPI_EC Interface 1	IBF	No	ACPI EC Interface 1 - Input Buffer Full Event
GIRQ15	14	ACPI_EC Interface 1	OBF	No	ACPI EC Interface 1 - Output Buffer Full Event
GIRQ15	15	ACPI_EC Interface 2	IBF	No	ACPI EC Interface 2 - Input Buffer Full Event
GIRQ15	16	ACPI_EC Interface 2	OBF	No	ACPI EC Interface 2 - Output Buffer Full Event
GIRQ15	17	ACPI_EC Interface 3	IBF	No	ACPI EC Interface 3 - Input Buffer Full Event
GIRQ15	18	ACPI_EC Interface 3	OBF	No	ACPI EC Interface 3 - Output Buffer Full Event
GIRQ16	0	LPC Interface	LPC_WAKE	Yes	Wake-Only Interrupt Event - LPC Traffic Detected
GIRQ16	1	SMBus Controller 0	SMB_WAKE	Yes	Wake-Only Interrupt Event - SMBus.0 START Detected
GIRQ16	2	SMBus Controller 1	SMB_WAKE	Yes	Wake-Only Interrupt Event - SMBus.1 START Detected
GIRQ16	3	SMBus Controller 2	SMB_WAKE	Yes	Wake-Only Interrupt Event - SMBus.2 START Detected
GIRQ16	4	PS2 Device Interface 0	PS2_DAT0_WAKE	Yes	Wake-Only Interrupt Event - PS/2.0 Start Bit Detected
GIRQ16	5	PS2 Device Interface 1A	PS2_DAT1A_WAKE	Yes	Wake-Only Interrupt Event - PS/2.1A Start Bit Detected
GIRQ16	6	PS2 Device Interface 1B	PS2_DAT1B_WAKE	Yes	Wake-Only Interrupt Event - PS/2.1B Start Bit Detected
GIRQ16	7	Keyboard Matrix Scan Interface	KSC_INT_WAKE	Yes	Wake-Only Interrupt Event - Keyboard Scan Interface Active
GIRQ16	8	ICSP Debugger	DEBUG_DONE	Yes	Wake-Only Interrupt Event - Processor may use this bit to put the chip back to sleep after Debug Access.
GIRQ16	9	ESPI Interface	ESPI_WAKE	Yes	Wake-Only Interrupt Event - ESPI Traffic Detected
GIRQ17	0	ADC Controller	ADC_Single_Int	No	ADC Controller - Single-Sample ADC Conversion Event
GIRQ17	1	ADC Controller	ADC_Repeat_Int	No	ADC Controller - Repeat-Sample ADC Conversion Event
GIRQ17	2	Reserved	Reserved	-	-
GIRQ17	3	Reserved	Reserved	-	-
GIRQ17	4	PS2 Device Interface 0	PS2_ACT	No	PS/2 Device Interface 0 - Activity Interrupt Event
GIRQ17	5	PS2 Device Interface 1	PS2_ACT	No	PS/2 Device Interface 1 - Activity Interrupt Event
GIRQ17	6	Keyboard Scan Interface	KSC_INT	No	Keyboard Scan Interface - Runtime Interrupt
GIRQ17	7	UART	UART	No	UART Interrupt Event

TABLE 10-2: INTERRUPT SOURCE, ENABLE SET, ENABLE CLEAR, AND RESULT BIT ASSIGNMENTS (CONTINUED)

Aggregator IRQ	Aggregator Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description
GIRQ17	8	PECI Interface	PECIHOST	No	PECI Host Event
GIRQ17	9	TACH 0	TACH	No	Tachometer 0 Interrupt Event
GIRQ17	10	TACH 1	TACH	No	Tachometer 1 Interrupt Event
GIRQ18	0	Quad Master SPI Controller	QMSPI_INT	No	Master SPI Controller Requires Servicing
GIRQ19	0	eSPI_Slave	INTR_PC	No	Peripheral Channel Interrupt
GIRQ19	1	eSPI_Slave	INTR_BM1	No	Bus Mastering Channel 1 Interrupt
GIRQ19	2	eSPI_Slave	INTR_BM2	No	Bus Mastering Channel 2 Interrupt
GIRQ19	3	eSPI_Slave	INTR_LTR	No	Peripheral Message (LTR) Interrupt
GIRQ19	4	eSPI_Slave	INTR_OOB_UP	No	Out of Band Channel Up Interrupt
GIRQ19	5	eSPI_Slave	INTR_OOB_DOWN	No	Out of Band Channel Down Interrupt
GIRQ19	6	eSPI_Slave	INTR_FLASH	No	Flash Channel Interrupt
GIRQ19	7	eSPI_Slave	eSPI_RESET	No	eSPI_RESET
GIRQ19	8	MCHP Reserved	MCHP Reserved	-	-
GIRQ20	0	BC-Link 0 Master	BCM_BUSY_CLR	No	BC-Link Busy Clear Flag
GIRQ20	1	BC-Link 0 Master	BCM_ERR	No	BC-Link Error Flag Interrupt
GIRQ20	2	BC-Link 0 Master	BCM_INT	Yes	BC-Link Companion Interrupt Event
GIRQ20	3	BC-Link 1 Master	BCM_BUSY_CLR	No	BC-Link Busy Clear Flag
GIRQ20	4	BC-Link 1 Master	BCM_ERR	No	BC-Link Error Flag Interrupt
GIRQ20	5	BC-Link 1 Master	BCM_INT	Yes	BC-Link Companion Interrupt Event
GIRQ21	0-2	Test	Test	-	-
GIRQ22	0	LPC Interface	LPC_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - LPC Traffic Detected
GIRQ22	1	SMBus Controller 0	SMB_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMBus.0 START Detected
GIRQ22	2	SMBus Controller 1	SMB_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMBus.1 START Detected
GIRQ22	3	SMBus Controller 2	SMB_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMBus.2 START Detected
GIRQ22	4	PS2 Device Interface 0	PS2_DAT0_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - PS/2.0 Start Bit Detected
GIRQ22	5	PS2 Device Interface 1A	PS2_DAT1A_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - PS/2.1A Start Bit Detected

TABLE 10-2: INTERRUPT SOURCE, ENABLE SET, ENABLE CLEAR, AND RESULT BIT ASSIGNMENTS (CONTINUED)

Aggregator IRQ	Aggregator Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description
GIRQ22	6	PS2 Device Interface 1B	PS2_-DAT1B_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - PS/2.1B Start Bit Detected
GIRQ22	7	Keyboard Matrix Scan Interface	KSC_INT_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - Keyboard Scan Interface Active
GIRQ22	8	ICSP Debugger	DEBUG_-DONE_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - Processor may use this bit to put the chip back to sleep after Debug Access.
GIRQ22	9	ESPI Interface	ESPI_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - ESPI Traffic Detected
GIRQ23	0	16-Bit - Basic Timer 0	Timer_Event	No	Basic Timer Event
GIRQ23	1	16-Bit - Basic Timer 1	Timer_Event	No	Basic Timer Event
GIRQ23	2	16-Bit - Basic Timer 2	Timer_Event	No	Basic Timer Event
GIRQ23	3	16-Bit - Basic Timer 3	Timer_Event	No	Basic Timer Event
GIRQ23	4	RTOS Timer	RTOS_TIMER	Yes	32-bit RTOS Timer Event
GIRQ23	5	Hibernation Timer	HTIMER	Yes	Hibernation Timer Event
GIRQ23	6	Week Alarm	WEEK_ALARM_INT	Yes	Week Alarm Interrupt.
GIRQ23	7	Week Alarm	SUB-_WEEK_ALARM_INT	Yes	Sub-Week Alarm Interrupt
GIRQ23	8	Week Alarm	ONE_SECOND	Yes	Week Alarm - One Second Interrupt
GIRQ23	9	Week Alarm	SUB_SECOND	Yes	Week Alarm - Sub-second Interrupt
GIRQ23	10	Week Alarm	SYSPWR_PRES	Yes	System Power Present Pin Interrupt
GIRQ23	11	VBAT-Powered Control Interface	VCI_OVRD_IN	Yes	VCI_OVRD_IN Active-high Input Pin Interrupt
GIRQ23	12	VBAT-Powered Control Interface	VCI_IN0	Yes	VCI_IN0 Active-low Input Pin Interrupt
GIRQ23	13	VBAT-Powered Control Interface	VCI_IN1	Yes	VCI_IN1 Active-low Input Pin Interrupt
GIRQ23	14	Reserved	Reserved	-	-
GIRQ23	15	Reserved	Reserved	-	-
GIRQ24	0	MIPS M14K	Core Timer Interrupt	No	Core Timer Interrupt
GIRQ24	1	MIPS M14K	Software Interrupt 0	No	Software Interrupt 0
GIRQ24	2	MIPS M14K	Software Interrupt 1	No	Software Interrupt 1
GIRQ25	0	eSPI_Slave	MSVW00_SRC0	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	1	eSPI_Slave	MSVW00_SRC1	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	2	eSPI_Slave	MSVW00_SRC2	Yes	Master-to-Slave Virtual Wire Interrupt Event

TABLE 10-2: INTERRUPT SOURCE, ENABLE SET, ENABLE CLEAR, AND RESULT BIT ASSIGNMENTS (CONTINUED)

Aggregator IRQ	Aggregator Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description
GIRQ25	3	eSPI_Slave	MSVW00_SRC3	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	4	eSPI_Slave	MSVW01_SRC0	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	5	eSPI_Slave	MSVW01_SRC1	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	6	eSPI_Slave	MSVW01_SRC2	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	7	eSPI_Slave	MSVW01_SRC3	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	8	eSPI_Slave	MSVW02_SRC0	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	9	eSPI_Slave	MSVW02_SRC1	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	10	eSPI_Slave	MSVW02_SRC2	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	11	eSPI_Slave	MSVW02_SRC3	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	12	eSPI_Slave	MSVW03_SRC0	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	13	eSPI_Slave	MSVW03_SRC1	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	14	eSPI_Slave	MSVW03_SRC2	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	15	eSPI_Slave	MSVW03_SRC3	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	16	eSPI_Slave	MSVW04_SRC0	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	17	eSPI_Slave	MSVW04_SRC1	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	18	eSPI_Slave	MSVW04_SRC2	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	19	eSPI_Slave	MSVW04_SRC3	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	20	eSPI_Slave	MSVW05_SRC0	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	21	eSPI_Slave	MSVW05_SRC1	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	22	eSPI_Slave	MSVW05_SRC2	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	23	eSPI_Slave	MSVW05_SRC3	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	24	eSPI_Slave	MSVW06_SRC0	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	25	eSPI_Slave	MSVW06_SRC1	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ25	26	eSPI_Slave	MSVW06_SRC2	Yes	Master-to-Slave Virtual Wire Interrupt Event

TABLE 10-2: INTERRUPT SOURCE, ENABLE SET, ENABLE CLEAR, AND RESULT BIT ASSIGNMENTS (CONTINUED)

Aggregator IRQ	Aggregator Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description
GIRQ25	27	eSPI_Slave	MSVW06_SRC3	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ26	0	eSPI_Slave	MSVW07_SRC0	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ26	1	eSPI_Slave	MSVW07_SRC1	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ26	2	eSPI_Slave	MSVW07_SRC2	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ26	3	eSPI_Slave	MSVW07_SRC3	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ26	4	eSPI_Slave	MSVW08_SRC0	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ26	5	eSPI_Slave	MSVW08_SRC1	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ26	6	eSPI_Slave	MSVW08_SRC2	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ26	7	eSPI_Slave	MSVW08_SRC3	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ26	8	eSPI_Slave	MSVW09_SRC0	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ26	9	eSPI_Slave	MSVW09_SRC1	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ26	10	eSPI_Slave	MSVW09_SRC2	Yes	Master-to-Slave Virtual Wire Interrupt Event
GIRQ26	11	eSPI_Slave	MSVW09_SRC3	Yes	Master-to-Slave Virtual Wire Interrupt Event

10.11.5 PRIORITY ENCODER AND DECODER

Every GIRQ Result bit has an associated Interrupt Priority Level (IPL) that is configurable by firmware (see Interrupt Priority Control Registers on page 186). The Priority Encoder and Decoder logic always presents the interrupt event that results in the highest Requested Interrupt Priority Level (RIPL) for a given mode of operation. The processor compares the RIPL to the current IPL being serviced to determine if it should preempt the current IRQ handler or allow the current IRQ handler to complete execution.

There are two modes of operation that effect how the hardware determines the RIPL: Aggregated Mode and Disaggregated mode. Firmware can select the mode of operation per GIRQ by programming the JTEnable (Jump-Table Enable) bit located in the Aggregator Control Registers. This allows the firmware to implement a fully aggregated solution, a fully disaggregated solution, or a hybrid solution.

10.11.5.1 Fully Aggregated Mode

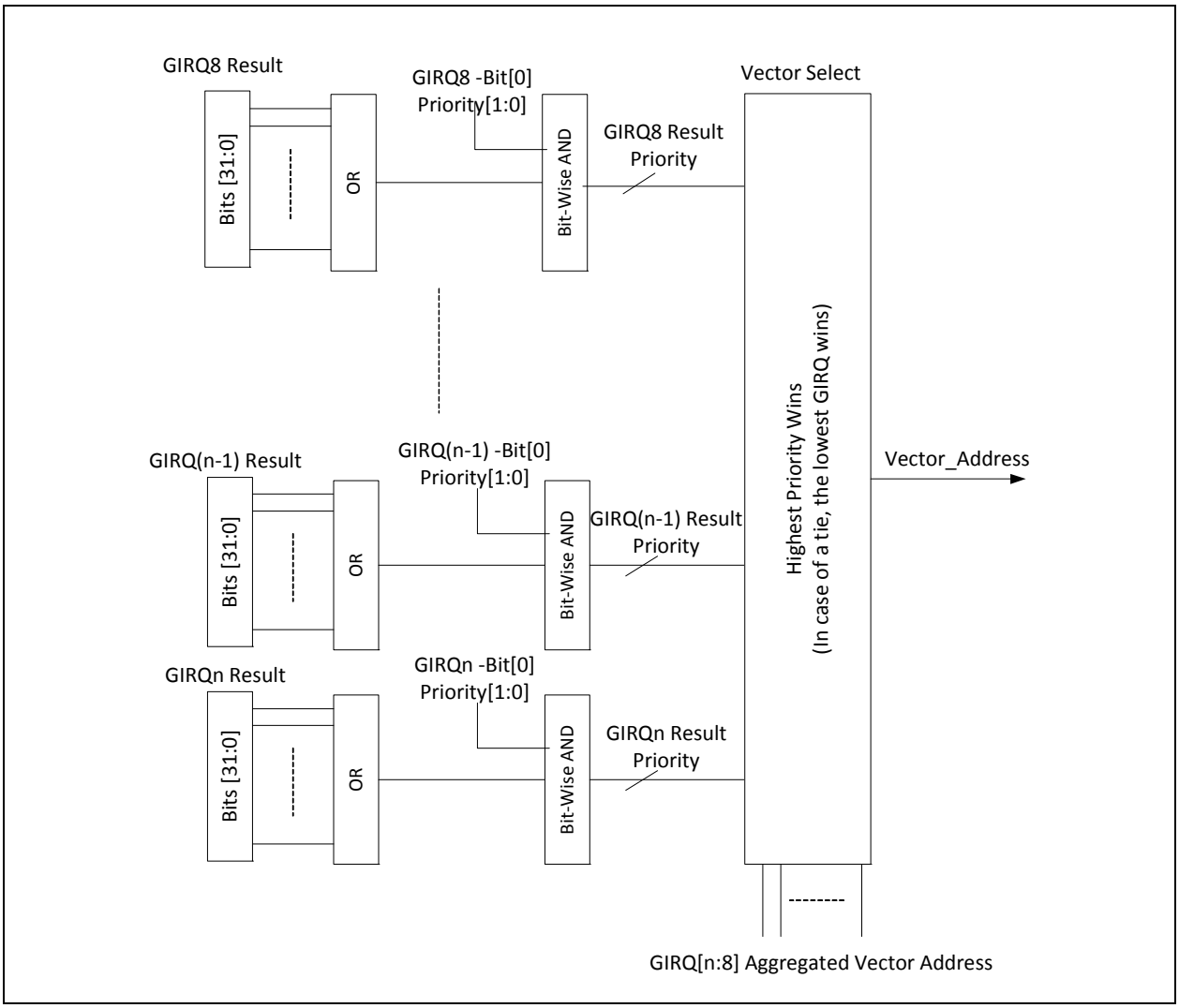
DETERMINING PRIORITY IN AGGREGATED MODE

In the fully aggregated mode, each GIRQ group is assigned the priority-level that is programmed for Result Bit 0 of that group. Priority Control bits for GIRQ Result Bits [31:1] have no function in this mode.

The Priority Encoder and Decision Logic generates the Vector for the active GIRQ interrupt with the highest priority. A GIRQ interrupt will be active if one or more of the bits within the GIRQ Result register are asserted. If two or more GIRQ events are active with the same priority-level the lowest numbered GIRQ wins.

The following diagram illustrates this selection process.

FIGURE 10-4: FULLY AGGREGATED PRIORITY ENCODER AND VECTOR ADDRESS



STEPS TO SET UP A PARTICULAR GIRQ GROUPING OF INTERRUPTS TO VECTOR TO AN ISR IN AGGREGATED MODE.

1. Determine location in code space of the ISR to handle GIRQ “n”. Program this 17-bit offset into the GIRQ aggregator control/vector address register. Of course, have the processor’s EBASE register programmed to the correct location as well.
2. (optional) Clear all source bits for the interrupts within GIRQ “n”.
3. Enable the individual interrupts within GIRQ “n” that you wish the ISR to handle.
4. Enable global interrupts in the processor.

ILLUSTRATIVE SCENARIO.

GIRQ #8 has 31 GPIOs from pins configured to generate interrupts that will be handled by an ISR labeled “GIRQ08_handler”. The 31 GPIOs are named (from GIRQ #8’s bit 0 through bit 30): GPIO001, GPIO002,….,GPIO030.

EBASE is at 0xbfd0_0000. The linker placed the handler at 0xbfd0_0500.

The firmware programs GIRQ #8's aggregator control to 0x0000_0500, sets each interrupt source priority to, say, 0x0 (2 bits of priority), which corresponds to priority level 1 to the processor. Then enables all interrupt lines by writing 0xffff_ffff to GIRQ #8's interrupt "enable set" register address.

If GPIO029 later fires an interrupt to the controller, the controller will send an EIC vector of 0x500 with a requested interrupt priority level 1 to the processor. The same goes for any of the other GPIOs firing an interrupt via GIRQ #8.

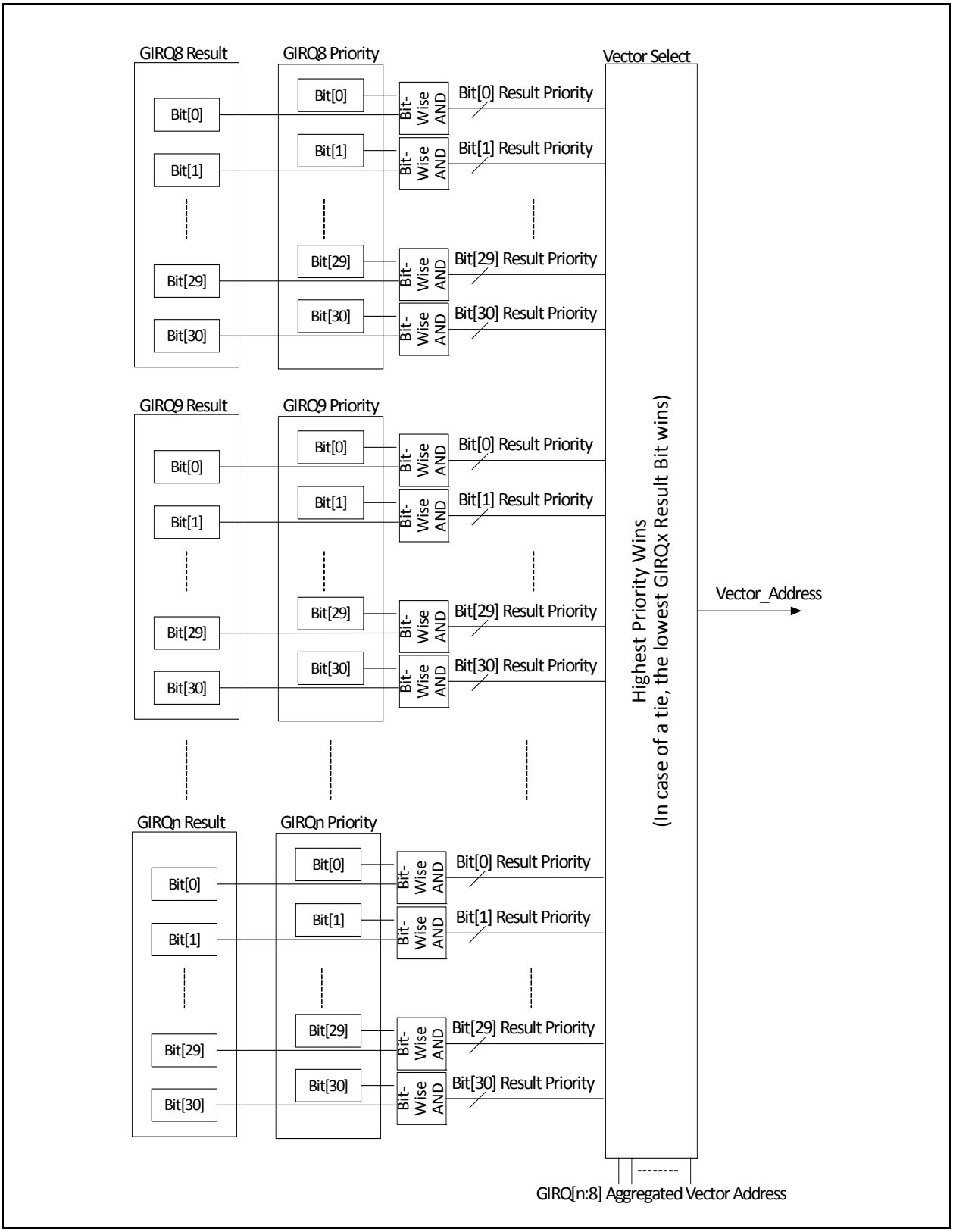
10.11.5.2 Fully Disaggregated Mode

DETERMINING PRIORITY IN DISAGGREGATED MODE

In the fully disaggregated mode, each GIRQx[n] Result Bit is assigned the priority-level that is programmed in the corresponding GIRQx[n] Priority bit. The Priority Encoder and Decision Logic generates the Vector for the active Result bit with the highest priority. If two or more Result bits are active with the same priority-level the lowest Result Bit wins.

The following diagram illustrates this selection process.

FIGURE 10-5: FULLY DISAGGREGATED PRIORITY ENCODER AND VECTOR ADDRESS



STEPS TO SET UP A PARTICULAR GIRQ GROUPING OF INTERRUPTS TO VECTOR TO AN ISR IN DISAGGREGATED/JT MODE.

1. Determine a location in code space to contain a mini-jump table, of size 31 entries or less, depending on how populated a particular GIRQ is (i.e. 15 populated sources = 15 jump table entries in SRAM).
2. Build up to 31 ISRs, one for each interrupt source in this GIRQ. The jump table gets populated with jump instructions the locations of these ISRs.
3. Program the 17-bit offset for the entry location of the mini-jump table into the GIRQ aggregator control/vector address register. EBASE must be programmed at 0xbfd0_0000.
4. (optional) Clear all source bits for the interrupts within GIRQ “n”.
5. Enable the individual interrupts within GIRQ “n” that you wish to be interrupt the processor.
6. Enable global interrupts in the processor.

ILLUSTRATIVE SCENARIO:

GIRQ #8 has 31 GPIOs from pins configured to generate interrupts that will be handled by an 31 ISRs labeled “GIRQ08_GPIO001_handler”, “GIRQ08_GPIO002_handler”, etc.

The 31 GPIOs are named (from GIRQ #8’s bit 0 through bit 30): GPIO001, GPIO002,….,GPIO030.

EBASE is at 0xbfd0_0000. Firmware places the jump table at address 0xbfd0_0500. The jump table gets populated with jump instructions to the 31 ISRs.

The firmware programs GIRQ #8’s aggregator control to 0x0000_0501 (bits 17:1 are the vector address, bit 0 is the GIRQ control to aggregate/dis-aggregate).

Firmware then sets each interrupt source priority to, say, 0x0 (2 bits of priority), which corresponds to priority level 1 to the processor. Then enables all interrupt lines by writing 0xffff_ffff to GIRQ #8’s interrupt “enable set” register address.

If GPIO029 later fires an interrupt to the controller, the controller will send an EIC vector of 0x5e8 with a requested interrupt priority level 1 to the processor. This causes the processor to vector to the 30th entry in the mini-jump table, which then jumps to the “GIRQ08_GPIO029_handler” code.

This address: 0x5e8 = vector base + 29*(vector spacing) which is by default 8 bytes.

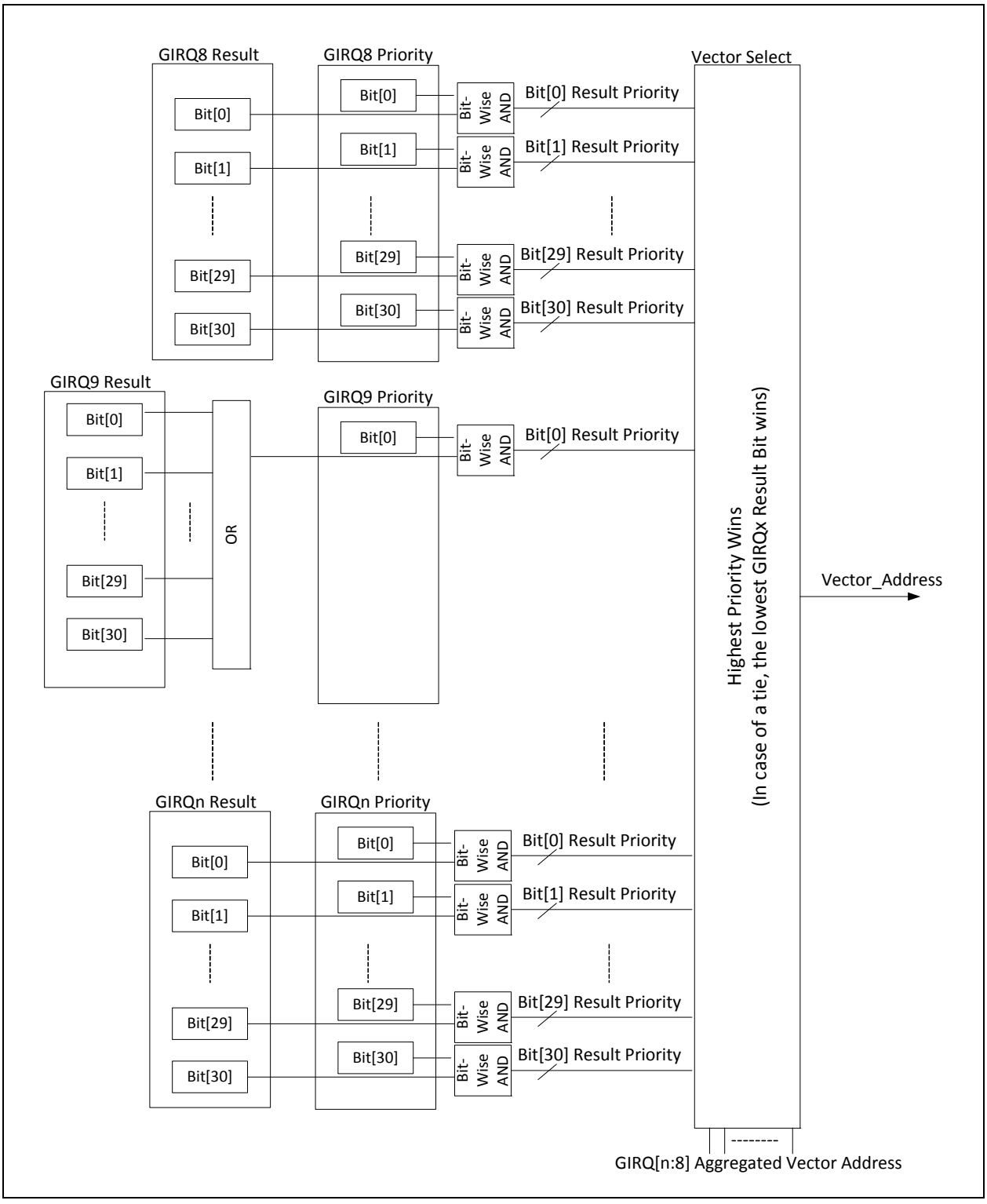
Later, GPIO002 fires an interrupt to the controller, which causes the controller to send an EIC vector of 0x510 with a requested interrupt priority level 1 to the processor. This causes the processor to vector to the 3rd entry in the mini-jump table, which then jumps to the “GIRQ08_GPIO002_handler” code.

10.11.6 HYBRID MODE

The Hybrid is a combination of the aggregated and disaggregated modes.

Each GIRQ group has the option of operating in either aggregated mode or disaggregated mode. This mode is similar to the disaggregated mode, except the grouped GIRQs will OR their result through bit 0 of that GIRQ. Each GIRQx[n] Result Bit is assigned the priority-level that is programmed in the corresponding GIRQx[n] Priority bit. The Priority Encoder and Decision Logic generates the Vector for the active Result bit with the highest priority. If two or more Result bits are active with the same priority-level the lowest Result Bit wins.

The following diagram illustrates this selection process.


GIRQ8 Result
GIRQ8 Priority
Vector Select
Bit[0]
Bit[1]
Bit[29]
Bit[30]
Bit-Wise AND
Bit[0] Result Priority
Bit[1] Result Priority
Bit[29] Result Priority
Bit[30] Result Priority
GIRQ9 Result
GIRQ9 Priority
OR
GIRQn Result
GIRQn Priority
Highest Priority Wins
(In case of a tie, the lowest GIRQx Result Bit wins)
Vector_Address
GIRQ[n:8] Aggregated Vector Address

10.12 JTVIC Registers

The registers listed in the JTVIC Register Summary table are for a single instance of the Jump Table Vectored Interrupt Controller (JTVIC). The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 10-3: EC-ONLY REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
Interrupt Controller	0	EC	32-bit internal address space	1FFF_C000h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 10-4: JTVIC REGISTER SUMMARY

Offset	Register Name
Interrupt Source, Enable Set, Enable Clear, and Result Registers	
00h	GIRQ8 Source Register
04h	GIRQ8 Enable Set Register
08h	GIRQ8 Enable Clear Register
0Ch	GIRQ8 Result Register
10h	GIRQ9 Source Register
14h	GIRQ9 Enable Set Register
18h	GIRQ9 Enable Clear Register
1Ch	GIRQ9 Result Register
20h	GIRQ10 Source Register
24h	GIRQ10 Enable Set Register
28h	GIRQ10 Enable Clear Register
2Ch	GIRQ10 Result Register
30h	GIRQ11 Source Register
34h	GIRQ11 Enable Set Register
38h	GIRQ11 Enable Clear Register
3Ch	GIRQ11 Result Register
40h	GIRQ12 Source Register
44h	GIRQ12 Enable Set Register
48h	GIRQ12 Enable Clear Register
4Ch	GIRQ12 Result Register

TABLE 10-4: JTVIC REGISTER SUMMARY (CONTINUED)

Offset	Register Name
50h	GIRQ13 Source Register
54h	GIRQ13 Enable Set Register
58h	GIRQ13 Enable Clear Register
5Ch	GIRQ13 Result Register
60h	GIRQ14 Source Register
64h	GIRQ14 Enable Set Register
68h	GIRQ14 Enable Clear Register
6Ch	GIRQ14 Result Register
70h	GIRQ15 Source Register
74h	GIRQ15 Enable Set Register
78h	GIRQ15 Enable Clear Register
7Ch	GIRQ15 Result Register
80h	GIRQ16 Source Register
84h	GIRQ16 Enable Set Register
88h	GIRQ16 Enable Clear Register
8Ch	GIRQ16 Result Register
90h	GIRQ17 Source Register
94h	GIRQ17 Enable Set Register
98h	GIRQ17 Enable Clear Register
9Ch	GIRQ17 Result Register
A0h	GIRQ18 Source Register
A4h	GIRQ18 Enable Set Register
A8h	GIRQ18 Enable Clear Register
ACh	GIRQ18 Result Register
B0h	GIRQ19 Source Register
B4h	GIRQ19 Enable Set Register
B8h	GIRQ19 Enable Clear Register
BCh	GIRQ19 Result Register
C0h	GIRQ20 Source Register
C4h	GIRQ20 Enable Set Register
C8h	GIRQ20 Enable Clear Register
CCh	GIRQ20 Result Register

TABLE 10-4: JTVIC REGISTER SUMMARY (CONTINUED)

Offset	Register Name
D0h	GIRQ21 Source Register
D4h	GIRQ21 Enable Set Register
D8h	GIRQ21 Enable Clear Register
DCh	GIRQ21 Result Register
E0h	GIRQ22 Source Register
E4h	GIRQ22 Enable Set Register
E8h	GIRQ22 Enable Clear Register
ECh	GIRQ22 Result Register
F0h	GIRQ23 Source Register
F4h	GIRQ23 Enable Set Register
F8h	GIRQ23 Enable Clear Register
FCh	GIRQ23 Result Register
100h	GIRQ24 Source Register
104h	GIRQ24 Enable Set Register
108h	GIRQ24 Enable Clear Register
10Ch	GIRQ24 Result Register
110h	GIRQ25 Source Register
114h	GIRQ25 Enable Set Register
118h	GIRQ25 Enable Clear Register
11Ch	GIRQ25 Result Register
120h	GIRQ26 Source Register
124h	GIRQ26 Enable Set Register
128h	GIRQ26 Enable Clear Register
12Ch	GIRQ26 Result Register
Aggregator Control Registers	
200h	GIRQ8 Aggregator Control Register
204h	GIRQ9 Aggregator Control Register
208h	GIRQ10 Aggregator Control Register
20Ch	GIRQ11 Aggregator Control Register
210h	GIRQ12 Aggregator Control Register
214h	GIRQ13Aggregator Control Register
218h	GIRQ14 Aggregator Control Register

TABLE 10-4: JTVIC REGISTER SUMMARY (CONTINUED)

Offset	Register Name
21Ch	GIRQ15 Aggregator Control Register
220h	GIRQ16 Aggregator Control Register
224h	GIRQ17Aggregator Control Register
228h	GIRQ18 Aggregator Control Register
22Ch	GIRQ19 Aggregator Control Register
230h	GIRQ20 Aggregator Control Register
234h	GIRQ21 Aggregator Control Register
238h	GIRQ22 Aggregator Control Register
23Ch	GIRQ23 Aggregator Control Register
240h	GIRQ24 Aggregator Control Register
244h	GIRQ25 Aggregator Control Register
248h	GIRQ26 Aggregator Control Register
Interrupt Priority Control Registers	
300h	GIRQ8 [7:0] Interrupt Priority Register
304h	GIRQ8 [15:8] Interrupt Priority Register
308h	GIRQ8 [23:16] Interrupt Priority Register
30Ch	GIRQ8 [31:24] Interrupt Priority Register
310h	GIRQ9 [7:0] Interrupt Priority Register
314h	GIRQ9 [15:8] Interrupt Priority Register
318h	GIRQ9 [23:16] Interrupt Priority Register
31Ch	GIRQ9 [31:24] Interrupt Priority Register
320h	GIRQ10 [7:0] Interrupt Priority Register
324h	GIRQ10 [15:8] Interrupt Priority Register
328h	GIRQ10 [23:16] Interrupt Priority Register
32Ch	GIRQ10 [31:24] Interrupt Priority Register
330h	GIRQ11 [7:0] Interrupt Priority Register
334h	GIRQ11 [15:8] Interrupt Priority Register
338h	GIRQ11 [23:16] Interrupt Priority Register
33Ch	GIRQ11 [31:24] Interrupt Priority Register
340h	GIRQ12 [7:0] Interrupt Priority Register
344h	GIRQ12 [15:8] Interrupt Priority Register
348h	GIRQ12 [23:16] Interrupt Priority Register

TABLE 10-4: JTVIC REGISTER SUMMARY (CONTINUED)

Offset	Register Name
34Ch	GIRQ12 [31:24] Interrupt Priority Register
350h	GIRQ13 [7:0] Interrupt Priority Register
354h	GIRQ13 [15:8] Interrupt Priority Register
358h	GIRQ13 [23:16] Interrupt Priority Register
35Ch	GIRQ13 [31:24] Interrupt Priority Register
360h	GIRQ14 [7:0] Interrupt Priority Register
364h	GIRQ14 [15:8] Interrupt Priority Register
368h	GIRQ14 [23:16] Interrupt Priority Register
36Ch	GIRQ14 [31:24] Interrupt Priority Register
370h	GIRQ15 [7:0] Interrupt Priority Register
374h	GIRQ15 [15:8] Interrupt Priority Register
378h	GIRQ15 [23:16] Interrupt Priority Register
37Ch	GIRQ15 [31:24] Interrupt Priority Register
380h	GIRQ16 [7:0] Interrupt Priority Register
384h	GIRQ16 [15:8] Interrupt Priority Register
388h	GIRQ16 [23:16] Interrupt Priority Register
38Ch	GIRQ16 [31:24] Interrupt Priority Register
390h	GIRQ17 [7:0] Interrupt Priority Register
394h	GIRQ17 [15:8] Interrupt Priority Register
398h	GIRQ17 [23:16] Interrupt Priority Register
39Ch	GIRQ17 [31:24] Interrupt Priority Register
3A0h	GIRQ18 [7:0] Interrupt Priority Register
3A4h	GIRQ18 [15:8] Interrupt Priority Register
3A8h	GIRQ18 [23:16] Interrupt Priority Register
3ACh	GIRQ18 [31:24] Interrupt Priority Register
3B0h	GIRQ19 [7:0] Interrupt Priority Register
3B4h	GIRQ19 [15:8] Interrupt Priority Register
3B8h	GIRQ19 [23:16] Interrupt Priority Register
3BCh	GIRQ19 [31:24] Interrupt Priority Register
3C0h	GIRQ20 [7:0] Interrupt Priority Register
3C4h	GIRQ20 [15:8] Interrupt Priority Register
3C8h	GIRQ20 [23:16] Interrupt Priority Register

TABLE 10-4: JTVIC REGISTER SUMMARY (CONTINUED)

Offset	Register Name
3CCh	GIRQ20 [31:24] Interrupt Priority Register
3D0h	GIRQ21 [7:0] Interrupt Priority Register
3D4h	GIRQ21 [15:8] Interrupt Priority Register
3D8h	GIRQ21 [23:16] Interrupt Priority Register
3DCh	GIRQ21 [31:24] Interrupt Priority Register
3E0h	GIRQ22 [7:0] Interrupt Priority Register
3E4h	GIRQ22 [15:8] Interrupt Priority Register
3E8h	GIRQ22 [23:16] Interrupt Priority Register
3ECh	GIRQ22 [31:24] Interrupt Priority Register
3F0h	GIRQ23 [7:0] Interrupt Priority Register
3F4h	GIRQ23 [15:8] Interrupt Priority Register
3F8h	GIRQ23 [23:16] Interrupt Priority Register
3FCh	GIRQ23 [31:24] Interrupt Priority Register
400h	GIRQ24 [7:0] Interrupt Priority Register
404h	GIRQ24 [15:8] Interrupt Priority Register
408h	GIRQ24 [23:16] Interrupt Priority Register
40Ch	GIRQ24 [31:24] Interrupt Priority Register
410h	GIRQ25 [7:0] Interrupt Priority Register
414h	GIRQ25 [15:8] Interrupt Priority Register
418h	GIRQ25 [23:16] Interrupt Priority Register
41Ch	GIRQ25 [31:24] Interrupt Priority Register
420h	GIRQ26 [7:0] Interrupt Priority Register
424h	GIRQ26 [15:8] Interrupt Priority Register
428h	GIRQ26 [23:16] Interrupt Priority Register
42Ch	GIRQ26 [31:24] Interrupt Priority Register
JTVIC Control Registers	
500h	JTVIC Control Register
504h	Interrupt Pending Register
508h	Aggregated Group Enable Set Register
50Ch	Aggregated Group Enabled Clear Register
510h	GIRQ Active Register

10.12.1 INTERRUPT SOURCE, ENABLE SET, ENABLE CLEAR, AND RESULT REGISTERS

All of the GIRQx Source, Enable, and Result registers have the same format. The following tables define the generic format for each of these registers. The bit definitions are defined in Table 10-2, “Interrupt Source, Enable Set, Enable Clear, and Result Bit Assignments,” on page 164.

TABLE 10-5: GIRQX SOURCE REGISTER FORMAT

Offset	See Table 10-4, "JTVIC Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	R	-	-
30:0	GIRQx Source Bit [30:0] The GIRQx Source bits are R/WC sticky status bits indicating the state of interrupt before the interrupt enable bit. For GIRQx Bit Assignments see Table 10-2, “Interrupt Source, Enable Set, Enable Clear, and Result Bit Assignments,” on page 164. Unassigned bits are Reserved; Reads return 0.	R/WC	0h	nSYSRST

TABLE 10-6: GIRQX ENABLE SET REGISTER FORMAT

Offset	See Table 10-4, "JTVIC Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	R	-	-
30:0	GIRQx Enable Set [31:0] Each GIRQx bit can be individually enabled to assert an interrupt event. 0= Writing a zero has no effect. 1= Writing a one will enable respective GIRQx. Reading always returns the current value of the GIRQx ENABLE bit. The state of the GIRQx ENABLE bit is determined by the corresponding GIRQx Enable Set bit and the GIRQx Enable Clear bit. (0=disabled, 1-enabled) **Note:** For GIRQx Bit Assignments see Table 10-2, “Interrupt Source, Enable Set, Enable Clear, and Result Bit Assignments,” on page 164. Unassigned bits are Reserved; Reads return 0.	R/WS	0h	nSYSRST

TABLE 10-7: GIRQX ENABLE CLEAR REGISTER FORMAT

Offset	See Table 10-4, "JTVIC Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	R	-	-
30:0	GIRQx Enable Clear[31:0] Each GIRQx bit can be individually disabled to assert an interrupt event. 0= Writing a zero has no effect. 1= Writing a one will disable respective GIRQx. Reading always returns the current value of the GIRQx ENABLE bit. The state of the GIRQx ENABLE bit is determined by the corresponding GIRQx Enable Set bit and the GIRQx Enable Clear bit. (0=disabled, 1-enabled) **Note:** For GIRQx Bit Assignments see Table 10-2, "Interrupt Source, Enable Set, Enable Clear, and Result Bit Assignments," on page 164. Unassigned bits are Reserved; Reads return 0.	R/WC	0h	nSYSRST

TABLE 10-8: GIRQX RESULT REGISTER FORMAT

Offset	See Table 10-4, "JTVIC Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Bit D31 is hard-coded to '1'.	R	1h	-
30:0	GIRQx Interrupt Result The GIRQx Result bits are Read-Only status bits indicating the state of interrupt after the interrupt enable bit. **Note:** For GIRQx Bit Assignments see Table 10-2, "Interrupt Source, Enable Set, Enable Clear, and Result Bit Assignments," on page 164. Unassigned bits are Reserved; Reads return 0.	R	0h	nSYSRST

10.12.2 AGGREGATOR CONTROL REGISTERS

TABLE 10-9: GIRQX AGGREGATOR CONTROL REGISTER FORMAT

Offset	-			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:18	Reserved	R	-	-
17:1	Aggregator Vector Address • In Aggregated Mode the Aggregated Vector Address is added to the processor EBASE to determine the physical jump address. • In Disaggregated Mode this is used as part of the calculation to determine the Jump Table Vector physical address. See JTEnable (Jump-Table Enable) bit description.	R/W	00h	nSYSRST

TABLE 10-9: GIRQX AGGREGATOR CONTROL REGISTER FORMAT

Offset	-			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	JTEnable (Jump-Table Enable) 0 = aggregated : present only the vector address from bits 17:1 1 = disaggregated/jump-table: present vector address from bits 17:1 + (vector_spacing)*(winning interrupt source bit position)	R/W	0h	nSYSRST

10.12.3 INTERRUPT PRIORITY CONTROL REGISTERS

TABLE 10-10: GIRQX [N+7:N] INTERRUPT PRIORITY REGISTER FORMAT

Offset	-			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:30	Reserved	R	-	-
29:28	GIRQX [N+7] Priority 00 = Priority Level 1 01 = Priority Level 3 10 = Priority Level 5 11 = Priority Level 7	R/W	0h	nSYSRST
27:26	Reserved	R	-	-
25:24	GIRQX [N+6] Priority 00 = Priority Level 1 01 = Priority Level 3 10 = Priority Level 5 11 = Priority Level 7	R/W	0h	nSYSRST
23:22	Reserved	R	-	-
21:20	GIRQX [N+5] Priority 00 = Priority Level 1 01 = Priority Level 3 10 = Priority Level 5 11 = Priority Level 7	R/W	0h	nSYSRST
19:18	Reserved	R	-	-
17:16	GIRQX [N+4] Priority 00 = Priority Level 1 01 = Priority Level 3 10 = Priority Level 5 11 = Priority Level 7	R/W	0h	nSYSRST
15:14	Reserved	R	-	-
13:12	GIRQX [N+3] Priority 00 = Priority Level 1 01 = Priority Level 3 10 = Priority Level 5 11 = Priority Level 7	R/W	0h	nSYSRST
11:10	Reserved	R	-	-

TABLE 10-10: GIRQX [N+7:N] INTERRUPT PRIORITY REGISTER FORMAT (CONTINUED)

Offset	-			
Bits	**Description**	**Type**	**Default**	**Reset Event**
9:8	GIRQX [N+2] Priority 00 = Priority Level 1 01 = Priority Level 3 10 = Priority Level 5 11 = Priority Level 7	R/W	0h	nSYSRST
7:6	Reserved	R	-	-
5:4	GIRQX [N+1] Priority 00 = Priority Level 1 01 = Priority Level 3 10 = Priority Level 5 11 = Priority Level 7	R/W	0h	nSYSRST
3:2	Reserved	R	-	-
1:0	GIRQX [N] Priority 00 = Priority Level 1 01 = Priority Level 3 10 = Priority Level 5 11 = Priority Level 7	R/W	0h	nSYSRST

10.12.4 JTVIC CONTROL REGISTERS

TABLE 10-11: JTVIC CONTROL REGISTER

Offset	500h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	R	-	-
8	Vector Spacing 0 = 8 Bytes 1 = 512 Bytes	R/W	00h	nSYSRST
7:1	Reserved	R	-	-
0	Soft Reset Soft Reset resets all flops in the JTVIC block **except** the interrupt source bits and the soft reset bit itself. 0 = Not Reset - Normal Operation 1 = Reset	R/W	0h	nSYSRST

TABLE 10-12: INTERRUPT PENDING REGISTER

Offset	504h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:19	Reserved	R	-	-
18:0	GIRQ[26:8] Aggregated Group Interrupt Source Pending This register shows the GIRQx pending interrupt sources. Each bit is the OR'd result of the corresponding GIRQx Interrupt Source register.	R	0h	nSYSRST

TABLE 10-13: AGGREGATED GROUP ENABLE SET REGISTER

Offset	508h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:19	Reserved	R	-	-
18:0	GIRQ[26:8] Aggregated Group Enable Set Each IRQ Vector can be individually enabled to assert an interrupt event to the EC. 0= Writing a zero has no effect. 1= Writing a one will enable respective IRQ*i*. Reading always returns the current value of the IRQ i VECTOR ENABLE bit. The state of the IRQ i VECTOR ENABLE bit is determined by the corresponding IRQ i Vector Enable Set bit and the IRQ i Vector Enable Clear bit. (0=disabled, 1-enabled)	R/W	0h	nSYSRST

TABLE 10-14: AGGREGATED GROUP ENABLE CLEAR REGISTER

Offset	50Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:19	Reserved	R	-	-
18:0	GIRQ[26:8] Aggregated Group Enable Clear Each IRQ Vector can be individually disabled to assert an interrupt event to the EC. 0= Writing a zero has no effect. 1= Writing a one will disable respective IRQ*i* vector. Reading always returns the current value of the IRQ i VECTOR ENABLE bit. The state of the IRQ i VECTOR ENABLE bit is determined by the corresponding IRQ i Vector Enable Set bit and the IRQ i Vector Enable Clear bit. (0=disabled, 1-enabled)	R/W	0h	nSYSRST

TABLE 10-15: GIRQX ACTIVE REGISTER

Offset	510h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:19	Reserved	R	-	-
18:0	GIRQ[26:8] Aggregated Group Active Each read only bit reflects the current state of the IRQ *i* vector to the EC. Each bit is the OR'd result of the corresponding GIRQx Interrupt Result register. If the IRQ *i* vector is disabled via the GIRQ[26:8] Aggregated Group Enable Clear register the corresponding IRQ *i* vector to the EC is forced to 0. If the IRQ *i* vector is enabled, the corresponding IRQ *i* vector to the EC represents the current status of the IRQ event.	R	0h	nSYSRST

11.0 WATCHDOG TIMER (WDT)

11.1 Introduction

The function of the Watchdog Timer is to provide a mechanism to detect if the internal embedded controller has failed.

When enabled, the Watchdog Timer (WDT) circuit will generate a WDT Event if the user program fails to reload the WDT within a specified length of time known as the WDT Interval.

11.2 References

No references have been cited for this chapter.

11.3 Terminology

There is no terminology defined for this chapter.

11.4 Interface

This block is designed to be accessed internally via a registered host interface or externally via the signal interface.

11.4.1 SIGNAL DESCRIPTION

TABLE 11-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
WDT_Stall[2:0]	Input	External 3-bit wide bus used to stall the WDT. Each of these signals may prevent the WDT from generating false WDT Events.

TABLE 11-2: MEC140X/1X WDT_STALL CONNECTIONS

Signal Name	Control Signals	Description
WDT_Stall[0]	Hibernation Timer	If enabled via the WDT_STALL_EN[0], the WDT will be stalled when the Hibernation Timer is counting.
WDT_Stall[1]	Week Timer Active	If enabled via the WDT_STALL_EN[1], the WDT will be stalled if the Week Timer is counting.
WDT_Stall[2]	ICSP Active	If enabled via the WDT_STALL_EN[2], the WDT will be stalled if there is activity on the ICSP ports. This allows the ICSP to be enabled, via the ICSP_MCLR pin, but not stall the WDT if there is no activity on the interface. The WDT_Stall[2] is also asserted when the WDT Enable bit in the ICDCON test register is 0.

11.5 Host Interface

The registers defined for the Watchdog Timer (WDT) are accessible by the embedded controller as indicated in Section 11.8, "EC-Only Registers". All registers accesses are synchronized to the host clock and complete immediately. Register reads/writes are not delayed by the 5Hz_Clk.

11.6 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

11.6.1 POWER DOMAINS

Name	Description
VTR	The logic and registers implemented in this block reside on this single power well.

11.6.2 CLOCK INPUTS

Name	Description
5Hz_Clk	The 5Hz_Clk clock input is the clock source to the Watchdog Timer functional logic, including the counter.

11.6.3 RESETS

Name	Description
nSYSRST	Power on Reset to the block. This signal resets all the register and logic in this block to its default state.

Source	Description
WDT Event	Pulse generated when WDT expires. This signal is used to reset the embedded controller and its subsystem. The event is cleared after an nSYSRST.

11.7 Description

11.7.1 WDT OPERATION

11.7.1.1 WDT Activation Mechanism

The WDT is activated by the following sequence of operations during normal operation:

1. Load the WDT Load Register with the count value.
2. Set the WDT Enable bit in the WDT Control Register.

The WDT Activation Mechanism starts the WDT decrementing counter.

11.7.1.2 WDT Deactivation Mechanism

The WDT is deactivated by the clearing the WDT Enable bit in the WDT Control Register. The WDT Deactivation Mechanism places the WDT in a low power state in which clock are gated and the counter stops decrementing.

11.7.1.3 WDT Reload Mechanism

The WDT must be reloaded within periods that are shorter than the programmed watchdog interval; otherwise, the WDT will underflow and a WDT Event will be generated and the WDT Status bit will be set in the WDT Control Register. It is the responsibility of the user program to continually execute code which reloads the watchdog timer, causing the counter to be reloaded

There are three methods of reloading the WDT: a write to the WDT Load Register, a write to the WDT Kick Register, or WDT event.

11.7.1.4 WDT Interval

The WDT Interval is the time it takes for the WDT to decrements from the WDT Load Register value to 0000h. The WDT Count Register value takes 33/5Hz_Clk seconds (ex. 33/32.768 KHz = 1.007ms) to decrement by 1 count.

11.8 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Watchdog Timer (WDT). The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 11-3: EC-ONLY REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
WDT	0	EC	32-bit internal address space	0000_0400h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 11-4: EC-ONLY REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	WDT Load Register
04h	WDT Control Register
08h	WDT Kick Register
0Ch	WDT Count Register

11.8.1 WDT LOAD REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	WDT Load Writing this field reloads the Watch Dog Timer counter.	R/W	Fh	nSYSRST

11.8.2 WDT CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:5	RESERVED	R	-	-
4	WDT_STALL_EN[2] This bit is used to enable Bit[2] of the WDT_Stall[2:0] input bus. For a description of the stall feature see EC-Only Registers on page 191. 0= EC-Only Registers not enabled on WDT_Stall[2] 1= EC-Only Registers enabled on WDT_Stall[2]	R/W	0b	nSYSRST
3	WDT_STALL_EN[1] This bit is used to enable Bit[1] of the WDT_Stall[2:0] input bus. For a description of the stall feature see EC-Only Registers on page 191. 0= EC-Only Registers not enabled on WDT_Stall[1] 1= EC-Only Registers enabled on WDT_Stall[1]	R/W	0b	nSYSRST
2	WDT_STALL_EN[0] This bit is used to enable Bit[0] of the WDT_Stall[2:0] input bus. For a description of the stall feature see EC-Only Registers on page 191. 0= EC-Only Registers not enabled on WDT_Stall[0] 1= EC-Only Registers enabled on WDT_Stall[0]	R/W	0b	nSYSRST
1	WDT Status WDT_RST is set by hardware if the last reset of MEC140x/1x was caused by an underflow of the WDT. See Section 11.7.1.3, "WDT Reload Mechanism," on page 190 for more information. This bit must be cleared by the EC firmware writing a ‘1’ to this bit. Writing a ‘0’ to this bit has no effect.	R/WC	0b	nSYSRST
0	WDT Enable In WDT Operation, the WDT is activated by the sequence of operations defined in Section 11.7.1.1, "WDT Activation Mechanism" and deactivated by the sequence of operations defined in Section 11.7.1.2, "WDT Deactivation Mechanism". 0 = block disabled 1 = block enabled **Note:** The default of the WDT is inactive.	R/W	0b	nSYSRST

11.8.3 WDT KICK REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Kick The WDT Kick Register is a strobe. Reads of the WDT Kick Register return 0. Writes to the WDT Kick Register cause the WDT to reload the WDT Load Register value and start decrementing when the WDT Enable bit in the WDT Control Register is set to '1'. When the WDT Enable bit in the WDT Control Register is cleared to '0', writes to the WDT Kick Register have no effect.	W	n/a	nSYSRST

11.8.4 WDT COUNT REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	WDT Count This read-only register provide the current WDT count.	R	Fh	nSYSRST

12.0 EMBEDDED MEMORY INTERFACE (EMI)

12.1 Introduction

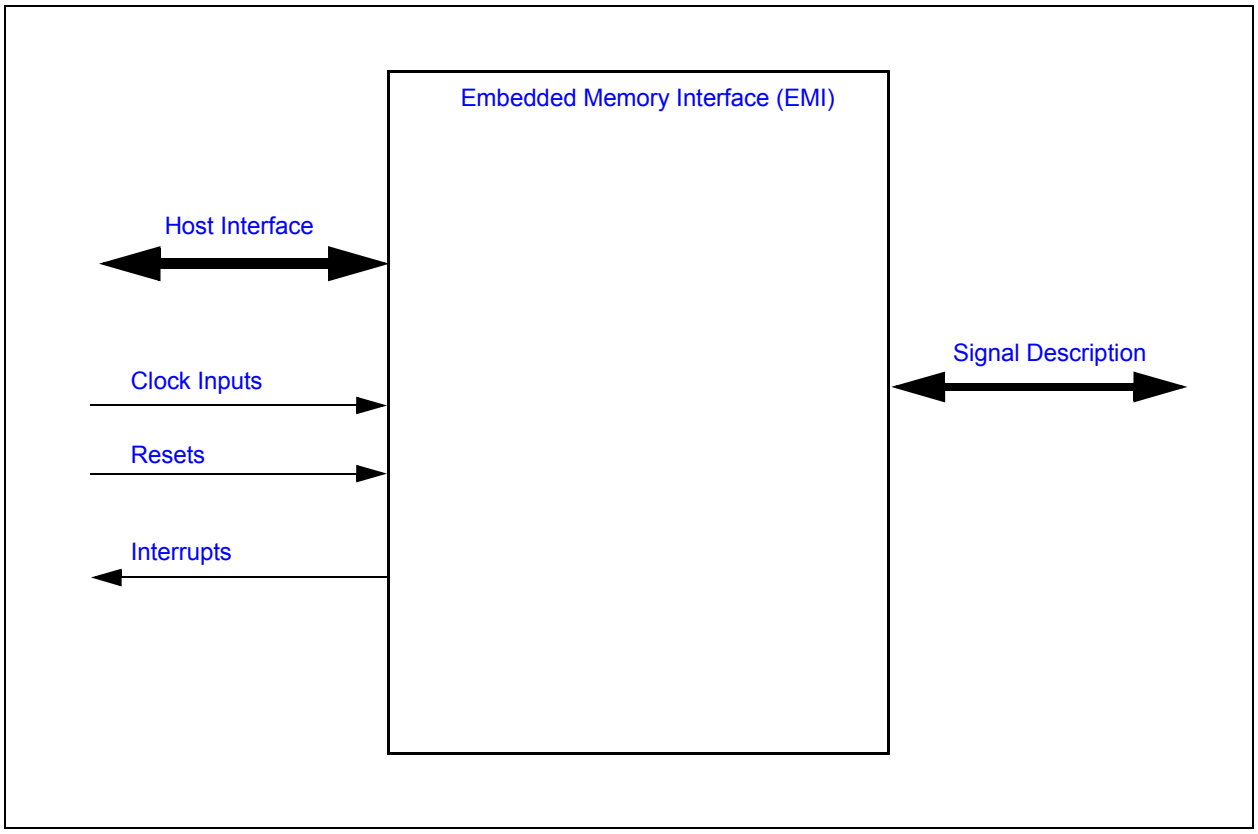
The Embedded Memory Interface (EMI) provides a standard run-time mechanism for the system host to communicate with the Embedded Controller (EC) and other logical components. The Embedded Memory Interface includes 13 byte-addressable registers in the Host's address space, as well as 22 bytes of registers that are accessible only by the EC. The Embedded Memory Interface can be used by the Host to access bytes of memory designated by the EC without requiring any assistance from the EC. The EC may configure these regions of memory as read-only, write-only, or read/write capable.

Note:	The Embedded Memory Interface (EMI) is supported for the LPC interface, however, it is not supported for eSPI.

12.2 Interface

This block is designed to be accessed externally and internally via a register interface.

FIGURE 12-1: I/O DIAGRAM OF BLOCK



12.3 Signal Description

TABLE 12-1: SIGNAL DESCRIPTION

Name	Direction	Description
nEMI_INT	OUTPUT	Active-low signal asserted when either the EC-to-Host or the Host_SWI_Event is asserted. This signal can be routed to nSMI and nPME inputs in the system as required.

12.4 Host Interface

The registers defined for the Embedded Memory Interface (EMI) are accessible by the System Host and the Embedded Controller as indicated in Section 12.10, "EC-Only Registers" and Section 12.9, "Runtime Registers".

12.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

12.5.1 POWER DOMAINS

Name	Description
VTR	The logic and registers implemented in this block reside on this single power well.

12.5.2 CLOCK INPUTS

This block has no special clocking requirements. Host register accesses are synchronized to the host bus clock and EC register accesses are synchronized to the EC bus clock, thereby allowing the transactions to complete in one bus clock.

12.5.3 RESETS

Name	Description
nSYSRST	This reset signal resets all the logic and register in this block.

12.6 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
EC-to-Host	This interrupt source for the SIRQ logic is generated when the EC_WR bit is ‘1’ and enabled by the EC_WR_EN bit.
Host_SWI_Event	This interrupt source for the SIRQ logic is generated when any of the EC_SWI bits are asserted and the corresponding EC_SWI_EN bit are asserted as well. This event is also asserted if the EC_WR/EC_WR_EN event occurs as well.

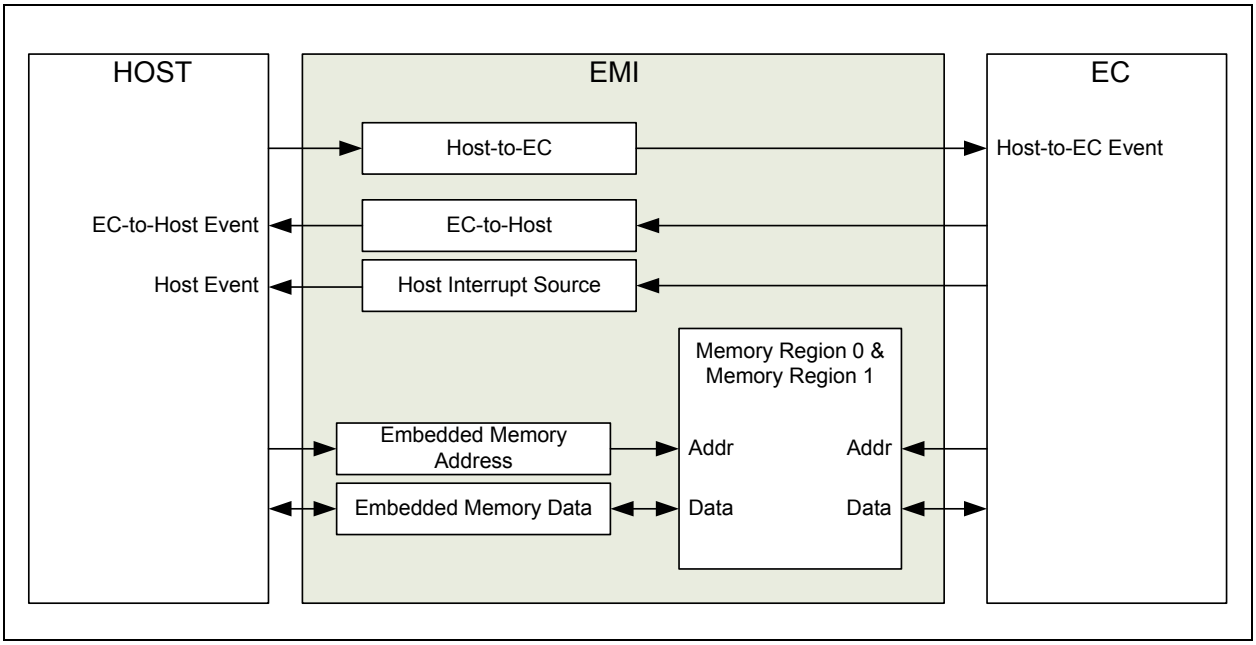
Source	Description
HOST-TO-EC	Interrupt source for the Interrupt Aggregator, generated by the host writing the HOST-to-EC Mailbox Register.

12.7 Low Power Modes

The Embedded Memory Interface (EMI) automatically enters low power mode when no transaction target it.

12.8 Description

FIGURE 12-2: EMBEDDED MEMORY INTERFACE BLOCK DIAGRAM



The Embedded Memory Interface (EMI) is composed of a mailbox, a direct memory interface, and an Application ID register.

The mailbox contains two registers, the HOST-to-EC Mailbox Register and the EC-to-HOST Mailbox Register, that act as a communication portal between the system host and the embedded controller. When the HOST-to-EC Mailbox Register is written an interrupt is generated to the embedded controller. Similarly, when the EC-to-HOST Mailbox Register is written an interrupt is generated to the system host. The source of the system host interrupt may be read in the Interrupt Source Register. These interrupt events may be individually prevented from generating a Host_SWI_Event via the Interrupt Mask Register.

The direct memory interface, which is composed of a byte addressable 16-bit EC Address Register and a 32-bit EC Data Register, permits the Host to read or write a portion of the EC's internal address space. The embedded controller may enable up to two regions of the EC's internal address space to be exposed to the system host. The system host may access these memory locations without intervention or assistance from the EC.

The Embedded Memory Interface can be configured so that data transfers between the Embedded Memory Interface data bytes and the 32- bit internal address space may be multiple bytes, while Host I/O is always executed a byte at a time.

When the Host reads one of the four bytes in the Embedded Memory Interface data register, data from the internal 32-bit address space, at the address defined by the Embedded Memory Interface address register, is returned to the Host. This read access will load 1, 2, or 4 bytes into the Data register depending on the configuration of the ACCESS_TYPE bits. Similarly, writing one of the four bytes in the data register will write the corresponding byte(s) from the data register

into the internal 32-bit address space as indicated by the ACCESS_TYPE bits. This configuration option is done to ensure that data the EC treats as 16-bit or 32-bit will be consistent in the Host, even though one byte of the data may change between two or more 8-bit accesses by the Host.

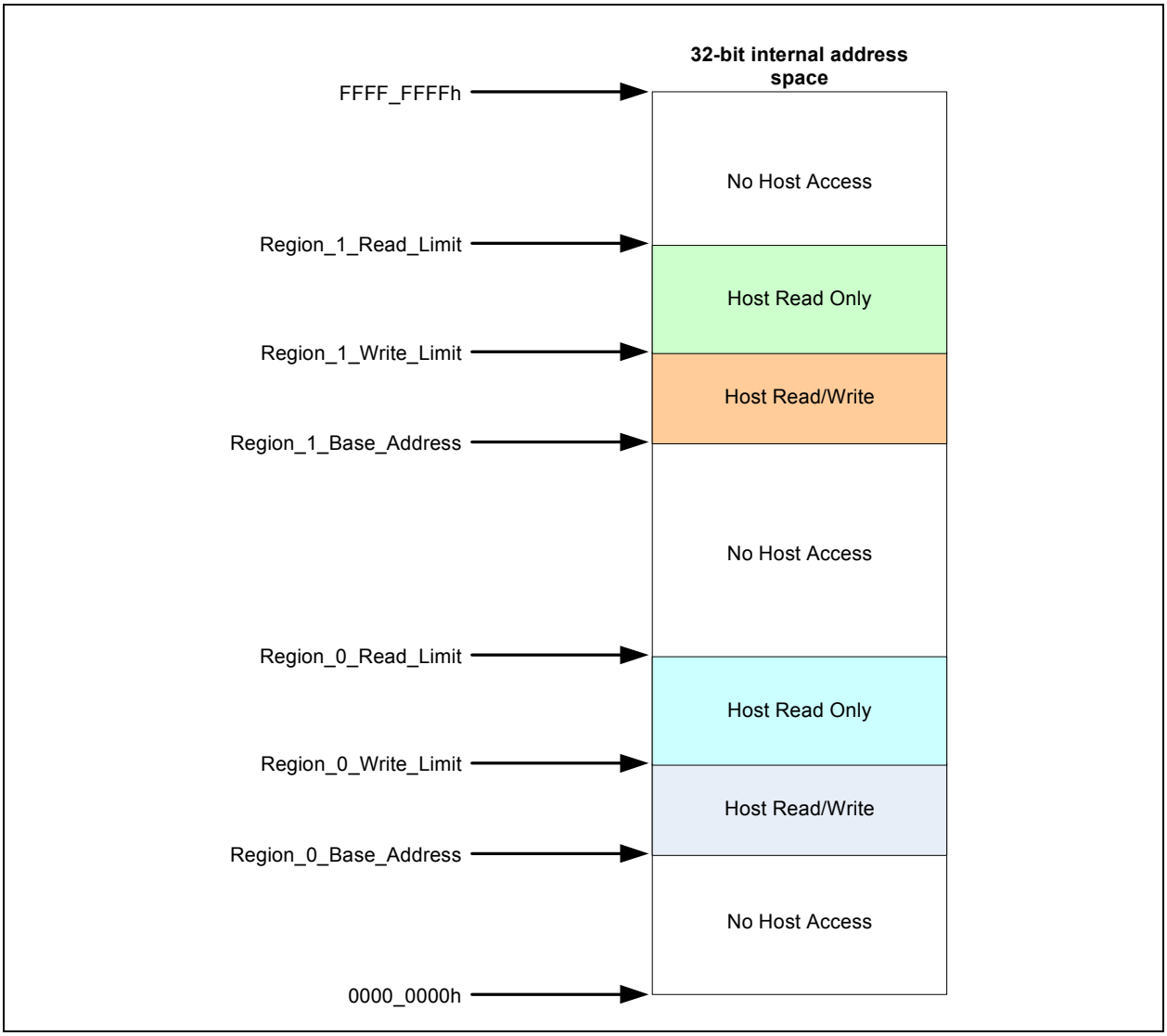
In addition, there is an auto-increment function for the Embedded Memory Interface address register. When enabled, the Host can read or write blocks of memory in the 32- bit internal address space by repeatedly accessing the Embedded Memory Interface data register, without requiring Host updates to the Embedded Memory Interface address register.

Finally, the Application ID Register may be used by the host to provide an arbitration mechanism if more than one software thread requires access through the EMI interface. See Section 12.8.4, "Embedded Memory Interface Usage," on page 199 for more details.

12.8.1 EMBEDDED MEMORY MAP

Each Embedded Memory interface provides direct access for the Host into two windows in the EC 32-bit internal address space. This mapping is shown in Figure 12-3, "Embedded Memory Addressing":

FIGURE 12-3: EMBEDDED MEMORY ADDRESSING



The Base addresses, the Read limits and the Write limits are defined by registers that are in the EC address space and cannot be accessed by the Host. In each region, the Read limit need not be greater than the Write limit. The regions can be contiguous or overlapping. For example, if the Region 0 Read limit is set to 0 and the Write limit is set to a positive number, then the Embedded Memory interface defines a region in the EC memory that the EC can read and write but is write-only for the host. This might be useful for storage of security data, which the Host might wish to send to the EC but should not be readable in the event a virus invades the Host.

Each window into the EC memory can be as large as 32k bytes in the 32-bit internal address space. Table 9-1, “EC Address Space,” on page 158 shows the host accessible regions.

12.8.2 EC DATA REGISTER

The 4 1-byte EC Data Byte registers function as a 32-bit register, which creates a 4 byte window into the Memory REGION being accessed. The 4-byte window is always aligned on a 4-byte boundary. Depending on the read/write configuration of the memory region being accessed, the bytes may be extracted from or loaded into memory as a byte, word, or a DWord. The ACCESS_TYPE determines the size of the memory access. The address accessed is determined by the two EC_Address byte registers, which together function as a 15-bit EC Address Register.

- A write to the EC Data Register when the EC Address is in a read-only or a no-access region, as defined by the Memory Base and Limit registers, will update the EC Data Register but memory will not be modified.
- A read to the EC Data Register when the EC Address is in a no-access region, as defined by the Memory Base and Limit registers, will not trigger a memory read and will not modify the EC Data Register. In auto-increment mode (ACCESS_TYPE=11b), reads of Byte 3 of the EC Data Register will still trigger increments of the EC Address Register when the address is out of bounds, while writes of Byte 3 will not.

12.8.3 ACCESS TYPES

The access type field (ACCESS_TYPE in the EC Address LSB Register) defines the type of host access that occurs when the EC Data Register is read or written.

11:Auto-increment 32-bit access. This defines a 32-bit access, as in the 10 case. In addition, any read or write of Byte 3 in the EC Data Register causes the EC Data Register to be incremented by 1. That is, the EC_Address field will point to the next 32-bit double word in the 32- bit internal address space.

10:32-bit access. A read of Byte 0 in the EC Data Register causes the 32 bits in the 32- bit internal address space at an offset of EC_Address to be loaded into the entire EC Data Register. The read then returns the contents of Byte 0. A read of Byte 1, Byte 2 or Byte 3 in the EC Data Register returns the contents of the register, without any update from the 32-bit internal address space.

A write of Byte 3 in the EC Data Register causes the EC Data Register to be written into the 32 bits in the 32- bit internal address space at an offset of EC_Address. A write of Byte 0, Byte 1 or Byte 2 in the EC Data Register updates the contents of the register, without any change to the 32- bit internal address space.

01:16-bit access. A read of Byte 0 in the EC Data Register causes the 16 bits in the 32- bit internal address space at an offset of EC_Address to be loaded into Byte 0 and Byte 1 of the EC Data Register. The read then returns the contents of Byte 0. A read of Byte 2 in the EC Data Register causes the 16 bits in the 32- bit internal address space at an offset of EC_Address+2 to be loaded into Byte 2 and Byte 3 of the EC Data Register. The read then returns the contents of Byte 2. A read of Byte 1 or Byte 3 in the EC Data Register return the contents of the register, without any update from the 32- bit internal address space.

A write of Byte 1 in the EC Data Register causes Bytes 1 and 0 of the EC Data Register to be written into the 16 bits in the 32- bit internal address space at an offset of EC_Address. A write of Byte 3 in the EC Data Register causes Bytes 3 and 2 of the EC Data Register to be written into the 16 bits in the 32- bit internal address space at an offset of EC_Address+2. A write of Byte 0 or Byte 2 in the EC Data Register updates the contents of the register, without any change to the 32- bit internal address space.

00:8-bit access. Any byte read of Byte 0 through Byte 3 in the EC Data Register causes the corresponding byte within the 32-bit double word addressed by EC_Address to be loaded into the byte of EC Data Register and returned by the read. Any byte write to Byte 0 through Byte 3 in the EC Data Register writes the corresponding byte within the 32-bit double word addressed by EC_Address, as well as the byte of the EC Data Register.

12.8.4 EMBEDDED MEMORY INTERFACE USAGE

The Embedded Memory Interface provides a generic facility for communication between the Host and the EC and can be used for many functions. Some examples are:

- Virtual registers. A block of memory in the 32-bit internal address space can be used to implement a set of virtual registers. The Host is given direct read-only access to this address space, referred to as peek mode. The EC may read or write this memory as needed.
- Program downloading. Because the Instruction Closely Coupled Memory is implemented in the same 32-bit internal address space, the Embedded Memory Interface can be used by the Host to download new program segments for the EC in the upper 32KB SRAM. The Read/Write window would be configured by the Host to point to the beginning of the loadable program region, which could then be loaded by the Host.
- Data exchange. The Read/Write portion of the memory window can be used to contain a communication packet. The Host, by default, “owns” the packet, and can write it at any time. When the Host wishes to communicate with the EC, it sends the EC a command, through the Host-to-EC message facility, to read the packet and perform some operations as a result. When it is completed processing the packet, the EC can inform the Host, either through a message in the EC-to-Host channel or by triggering an event such as an SMI directly. If return results are required, the EC can write the results into the Read/Write region, which the Host can read directly when it is informed that the EC has completed processing. Depending on the command, the operations could entail update of virtual registers in the 32-bit internal address space, reads of any register in the EC address space, or writes of any register in the EC address space. Because there are two regions that are defined by the base registers, the memory used for the communication packet does not have to be contiguous with a set of virtual registers.

Because there are two Embedded Memory Interface memory regions, the Embedded Memory Interface cannot be used for more than two of these functions at a time. The Host can request that the EC switch from one function to another through the use of the Host-to-EC mailbox register.

The Application ID Register is provided to help software applications track ownership of an Embedded Memory Interface. An application can write the register with its Application ID, then immediately read it back. If the read value is not the same as the value written, then another application has ownership of the interface.

Note:	The protocol used to pass commands back and forth through the Embedded Memory Interface Registers Interface is left to the System designer. Microchip can provide an application example of working code in which the host uses the Embedded Memory Interface registers to gain access to all of the EC registers.

12.9 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the **EMI**. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the Runtime Register Base Address Table.

TABLE 12-2: RUNTIME REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
EMI	0	EC	32-bit internal address space	000F_0000h
		LPC	I/O	Programmed BAR

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 12-3: RUNTIME REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	HOST-to-EC Mailbox Register
01h	EC-to-HOST Mailbox Register
02h	EC Address LSB Register
03h	EC Address MSB Register
04h	EC Data Byte 0 Register
05h	EC Data Byte 1 Register
06h	EC Data Byte 2 Register
07h	EC Data Byte 3 Register
08h	Interrupt Source LSB Register
09h	Interrupt Source MSB Register
0Ah	Interrupt Mask LSB Register
0Bh	Interrupt Mask MSB Register
0Ch	Application ID Register

12.9.1 HOST-TO-EC MAILBOX REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	HOST_EC_MBOX 8-bit mailbox used communicate information from the system host to the embedded controller. Writing this register generates an event to notify the embedded controller. The embedded controller has the option of clearing some or all of the bits in this register. This is dependent on the protocol layer implemented using the EMI Mailbox. The host must know this protocol to determine the meaning of the value that will be reported on a read. This bit field is aliased to the HOST_EC_MBOX bit field in the HOST-to-EC Mailbox Register	R/W	0h	nSYSRST

12.9.2 EC-TO-HOST MAILBOX REGISTER

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_HOST_MBOX 8-bit mailbox used communicate information from the embedded controller to the system host. Writing this register generates an event to notify the system host. The system host has the option of clearing some or all of the bits in this register. This is dependent on the protocol layer implemented using the EMI Mailbox. The embedded controller must know this protocol to determine the meaning of the value that will be reported on a read. This bit field is aliased to the EC_HOST_MBOX bit field in the EC-to-HOST Mailbox Register	R/WC	0h	nSYSRST

12.9.3 EC ADDRESS LSB REGISTER

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:2	EC_ADDRESS_LSB This field defines bits[7:2] of EC_Address [15:0]. Bits[1:0] of the EC_Address are always forced to 00b. The EC_Address is aligned on a DWord boundary. It is the address of the memory being accessed by EC Data Byte 0 Register, which is an offset from the programmed base address of the selected REGION.	R/W	0h	nSYSRST
1:0	ACCESS_TYPE This field defines the type of access that occurs when the EC Data Register is read or written. 11b=Auto-increment 32-bit access. 10b=32-bit access. 01b=16-bit access. 00b=8-bit access. Each of these access types are defined in detail in Section 12.8.3, "Access Types".	R/W	0h	nSYSRST

12.9.4 EC ADDRESS MSB REGISTER

Offset	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	REGION The field specifies which of two segments in the 32-bit internal address space is to be accessed by the EC_Address[14:2] to generate accesses to the memory. 1= The address defined by EC_Address[14:2] is relative to the base address specified by the Memory Base Address 1 Register. 0= The address defined by EC_Address[14:2] is relative to the base address specified by the Memory Base Address 0 Register.	R/W	0h	nSYSRST
6:0	EC_ADDRESS_MSB This field defines bits[14:8] of EC_Address. Bits[1:0] of the EC_Address are always forced to 00b. The EC_Address is aligned on a DWord boundary. It is the address of the memory being accessed by EC Data Byte 0 Register, which is an offset from the programmed base address of the selected REGION.	R/W	0h	nSYSRST

12.9.5 EC DATA BYTE 0 REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_DATA_BYTE_0 This is byte 0 (Least Significant Byte) of the 32-bit EC Data Register. Use of the Data Byte registers to access EC memory is defined in detail in Section 12.8.2, "EC Data Register".	R/W	0h	nSYSRST

12.9.6 EC DATA BYTE 1 REGISTER

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_DATA_BYTE_1 This is byte 1 of the 32-bit EC Data Register. Use of the Data Byte registers to access EC memory is defined in detail in Section 12.8.2, "EC Data Register".	R/W	0h	nSYSRST

12.9.7 EC DATA BYTE 2 REGISTER

Offset	06h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_DATA_BYTE_2 This is byte 2 of the 32-bit EC Data Register. Use of the Data Byte registers to access EC memory is defined in detail in Section 12.8.2, "EC Data Register".	R/W	0h	nSYSRST

12.9.8 EC DATA BYTE 3 REGISTER

Offset	07h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_DATA_BYTE_3 This is byte 3 (Most Significant Byte) of the 32-bit EC Data Register. Use of the Data Byte registers to access EC memory is defined in detail in Section 12.8.2, "EC Data Register".	R/W	0h	nSYSRST

12.9.9 INTERRUPT SOURCE LSB REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	EC_SWI_LSB EC Software Interrupt Least Significant Bits. These bits are software interrupt bits that may be set by the EC to notify the host of an event. The meaning of these bits is dependent on the firmware implementation. Each bit in this field is cleared when written with a '1b'. The ability to clear the bit can be disabled by the EC if the corresponding bit in the Host Clear Enable Register is set to '0b'. This may be used by firmware for events that cannot be cleared while the event is still active.	R/WC	0h	nSYSRST
0	EC_WR EC Mailbox Write. This bit is set when the EC-to-HOST Mailbox Register has been written by the EC at offset 01h of the EC-Only registers.	R	0h	nSYSRST

12.9.10 INTERRUPT SOURCE MSB REGISTER

Offset	09h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_SWI_MSB EC Software Interrupt Most Significant Bits. These bits are software interrupt bits that may be set by the EC to notify the host of an event. The meaning of these bits is dependent on the firmware implementation. Each bit in this field is cleared when written with a '1b'. The ability to clear the bit can be disabled by the EC. if the corresponding bit in the Host Clear Enable Register is set to '0b'. This may be used by firmware for events that cannot be cleared while the event is still active.	R/WC	0h	nSYSRST

12.9.11 INTERRUPT MASK LSB REGISTER

Offset	0Ah			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	EC_SWI_EN_LSB EC Software Interrupt Enable Least Significant Bits. Each bit that is set to '1b' in this field enables the generation of a Host_SWI_Event interrupt by the corresponding bit in the EC_SWI field in the Interrupt Source LSB Register.	R/W	0h	nSYSRST
0	EC_WR_EN EC Mailbox Write.Interrupt Enable. If this bit is '1b', the interrupt generated by bit EC_WR in the Interrupt Source LSB Register is enabled to generate a EC-to-Host interrupt event.	R/W	0h	nSYSRST

12.9.12 INTERRUPT MASK MSB REGISTER

Offset	0Bh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_SWI_EN_MSB EC Software Interrupt Enable Most Significant Bits. Each bit that is set to '1b' in this field enables the generation of a Host_SWI_Event interrupt by the corresponding bit in the EC_SWI field in the Interrupt Source MSB Register.	R/W	0h	nSYSRST

12.9.13 APPLICATION ID REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	APPLICATION_ID When this field is 00h it can be written with any value. When set to a non-zero value, writing that value will clear this register to 00h. When set to a non-zero value, writing any value other than the current contents will have no effect.	R/W	0h	nSYSRST

12.10 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Embedded Memory Interface (EMI). The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 12-4: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
EMI	0	EC	32-bit internal address space	000F_0100h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 12-5: EC-ONLY REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	HOST-to-EC Mailbox Register
01h	EC-to-HOST Mailbox Register
04h	Memory Base Address 0 Register
08h	Memory Read Limit 0 Register
0Ah	Memory Write Limit 0 Register
0Ch	Memory Base Address 1 Register
10h	Memory Read Limit 1 Register
12h	Memory Write Limit 1 Register
14h	Interrupt Set Register
16h	Host Clear Enable Register

12.10.1 HOST-TO-EC MAILBOX REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	HOST_EC_MBOX 8-bit mailbox used communicate information from the system host to the embedded controller. Writing this register generates an event to notify the embedded controller. The embedded controller has the option of clearing some or all of the bits in this register. This is dependent on the protocol layer implemented using the EMI Mailbox. The host must know this protocol to determine the meaning of the value that will be reported on a read. This bit field is aliased to the HOST_EC_MBOX bit field in the HOST-to-EC Mailbox Register.	R/WC	0h	nSYSRST

12.10.2 EC-TO-HOST MAILBOX REGISTER

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_HOST_MBOX 8-bit mailbox used communicate information from the embedded controller to the system host. Writing this register generates an event to notify the system host. The system host has the option of clearing some or all of the bits in this register. This is dependent on the protocol layer implemented using the EMI Mailbox. The embedded controller must know this protocol to determine the meaning of the value that will be reported on a read. This bit field is aliased to EC_HOST_MBOX bit field in EC-to-HOST Mailbox Register (EC_TO_HOST) on page 129 defined in Section 10.10, "Runtime Registers".	R/W	0h	nSYSRST

12.10.3 MEMORY BASE ADDRESS 0 REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	MEMORY_BASE_ADDRESS_0 This memory base address defines the beginning of region 0 in the Embedded Controller's 32-bit internal address space. Memory allocated to region 0 is intended to be shared between the Host and the EC. The region defined by this base register is used when bit 15 of the EC Address Register is 0. The access will be to a memory location at an offset defined by the EC_Address relative to the beginning of the region defined by this register. Therefore, a read or write to the memory that is triggered by the EC Data Register will occur at Memory_Base_Address_0 + EC_Address.	R/W	0h	nSYSRST
1:0	Reserved	R	-	-

12.10.4 MEMORY READ LIMIT 0 REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15	Reserved	R	-	-
14:2	MEMORY_READ_LIMIT_0 Whenever a read of any byte in the EC Data Register is attempted, and bit 15 of EC_Address is 0, the field EC_Address[14:2] in the EC_Address_Register is compared to this field. As long as EC_Address[14:2] is less than this field the EC_Data_Register will be loaded from the 32-bit internal address space.	R/W	0h	nSYSRST
1:0	Reserved	R	-	-

12.10.5 MEMORY WRITE LIMIT 0 REGISTER

Offset	0Ah			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15	Reserved	R	-	-
14:2	MEMORY_WRITE_LIMIT_0 Whenever a write of any byte in EC DATA Register is attempted and bit 15 of EC_Address is 0, the field EC_ADDRESS_MSB in the EC_Address Register is compared to this field. As long as EC_Address[14:2] is less than Memory_Write_Limit_0[14:2] the addressed bytes in the EC DATA Register will be written into the internal 32-bit address space. If EC_Address[14:2] is greater than or equal to the Memory_Write_Limit_0[14:2] no writes will take place.	R/W	0h	nSYSRST
1:0	Reserved	R	-	-

12.10.6 MEMORY BASE ADDRESS 1 REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	MEMORY_BASE_ADDRESS_1 This memory base address defines the beginning of region 1 in the Embedded Controller's 32-bit internal address space. Memory allocated to region 1 is intended to be shared between the Host and the EC. The region defined by this base register is used when bit 15 of the EC Address Register is 1. The access will be to a memory location at an offset defined by the EC_Address relative to the beginning of the region defined by this register. Therefore, a read or write to the memory that is triggered by the EC Data Register will occur at Memory_Base_Address_1 + EC_Address.	R/W	0h	nSYSRST
1:0	Reserved	R	-	-

12.10.7 MEMORY READ LIMIT 1 REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15	Reserved	R	-	-
14:2	MEMORY_READ_LIMIT_1 Whenever a read of any byte in the EC Data Register is attempted, and bit 15 of EC_ADDRESS is 1, the field EC_ADDRESS in the EC_Address_Register is compared to this field. As long as EC_ADDRESS is less than this value, the EC_Data_Register will be loaded from the 32-bit internal address space.	R/W	0h	nSYSRST
1:0	Reserved	R	-	-

12.10.8 MEMORY WRITE LIMIT 1 REGISTER

Offset	12h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15	Reserved	R	-	-
14:2	MEMORY_WRITE_LIMIT_1 Whenever a write of any byte in EC DATA Register is attempted and bit 15 of EC_Address is 1, the field EC_Address[14:2] in the EC_Address Register is compared to this field. As long as EC_Address[14:2] is less than Memory_Write_Limit_1[14:2] the addressed bytes in the EC DATA Register will be written into the internal 32-bit address space. If EC_Address[14:2] is greater than or equal to the Memory_Write_Limit_1[14:2] no writes will take place.	R/W	0h	nSYSRST
1:0	Reserved	R	-	-

12.10.9 INTERRUPT SET REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:1	EC_SWI_SET EC Software Interrupt Set. This register provides the EC with a means of updating the Interrupt Source Registers. Writing a bit in this field with a ‘1b’ sets the corresponding bit in the Interrupt Source Register to ‘1b’. Writing a bit in this field with a ‘0b’ has no effect. Reading this field returns the current contents of the Interrupt Source Register.	R/WS	0h	nSYSRST
0	Reserved	R	-	-

12.10.10 HOST CLEAR ENABLE REGISTER

Offset	16h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:1	HOST_CLEAR_ENABLE When a bit in this field is ‘0b’, the corresponding bit in the Interrupt Source Register cannot be cleared by writes to the Interrupt Source Register. When a bit in this field is ‘1b’, the corresponding bit in the Interrupt Source Register can be cleared when that register bit is written with a ‘1b’. These bits allow the EC to control whether the status bits in the Interrupt Source Register are based on an edge or level event.	R/W	0h	nSYSRST
0	Reserved	R	-	-

13.0 MAILBOX INTERFACE

13.1 Overview

The Mailbox provides a standard run-time mechanism for the host to communicate with the Embedded Controller (EC)

13.2 References

No references have been cited for this feature.

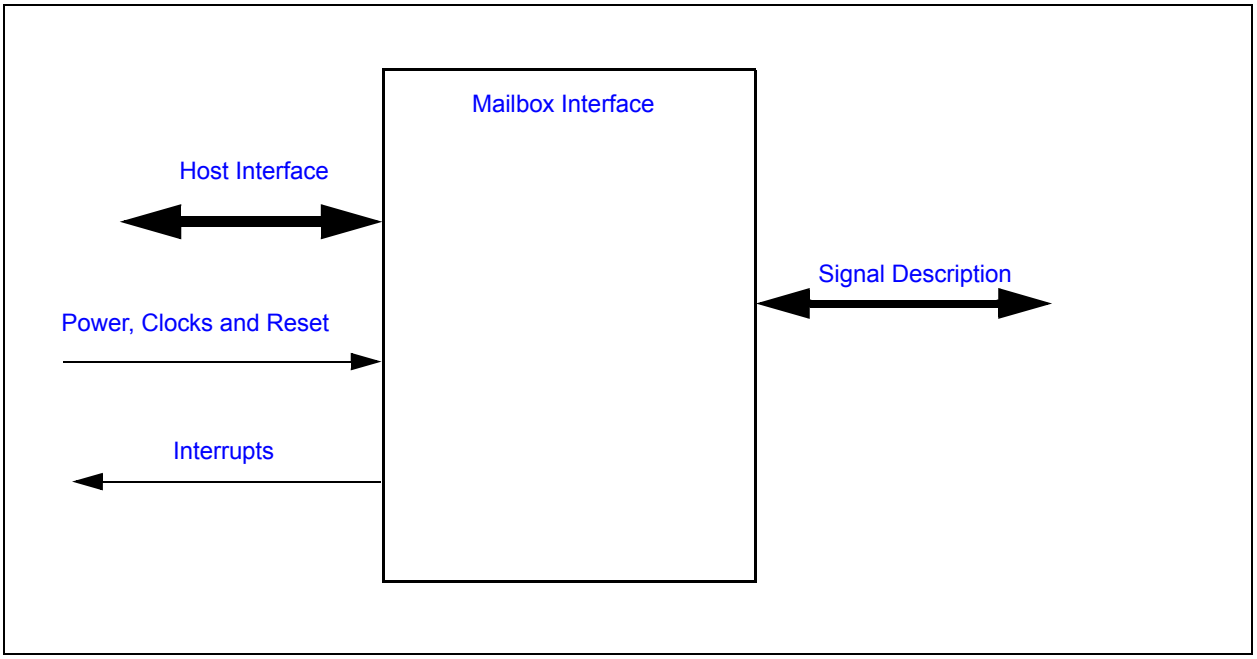
13.3 Terminology

There is no terminology defined for this section.

13.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 13-1: I/O DIAGRAM OF BLOCK



13.5 Signal Description

TABLE 13-1: SIGNAL DESCRIPTION

Name	Direction	Description
nSMI	OUTPUT	SMI alert signal to the Host.

13.6 Host Interface

The Mailbox interface is accessed by host software via a registered interface, as defined in Section 13.11, "Runtime Registers" and Section 13.12, "EC-Only Registers".

13.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

13.7.1 POWER DOMAINS

Name	Description
VTR	The logic and registers implemented in this block are powered by this power well.

13.7.2 CLOCK INPUTS

Name	Description
48 MHz Ring Oscillator	This is the clock source for Mailbox logic.

13.7.3 RESETS

Name	Description
nSYSRST	This signal resets all the registers and logic in this block to their default state.
VCC_PWRGD	This signal is asserted when the main power rail is asserted. The Host Access Port is reset when this signal is de-asserted.

13.8 Interrupts

Source	Description
MBX_Host_SIRQ	This interrupt source for the SIRQ logic is generated when the EC_WR bit is ‘1’ and enabled by the EC_WR_EN bit.
MBX_Host_SMI	This interrupt source for the SIRQ logic is generated when any of the EC_SWI bits are asserted and the corresponding EC_SWI_EN bit are asserted as well. This event is also asserted if the EC_WR/EC_WR_EN event occurs as well. This bit is also routed to the nSMI pin.

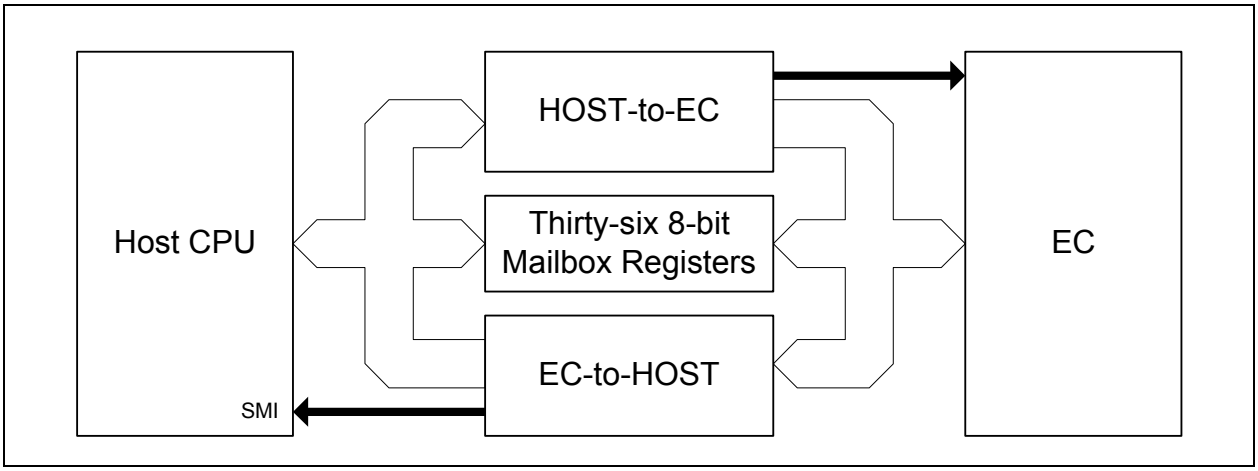
Source	Description
MBX	Interrupt generated by the host writing the HOST-to-EC Mailbox register.
MBX_DATA	Interrupt generated by the host writing the MBX_DATA register.

13.9 Low Power Modes

The Mailbox automatically enters a low power mode whenever it is not actively.

13.10 Description

FIGURE 13-2: MAILBOX BLOCK DIAGRAM



13.10.1 HOST ACCESS PORT

The Mailbox includes a total of 36 index-addressable 8-bit Mailbox registers and a two byte Mailbox Registers Host Access Port. Thirty-two of the 36 index-addressable 8-bit registers are EC Mailbox registers, which can be read and written by both the EC and the Host. The remaining four registers are used for signaling between the Host and the EC. The Host Access Port consists of two 8-bit run-time registers that occupy two addresses in the HOST I/O space, MBX_INDEX Register and MBX_DATA Register. The Host Access Port is used by the host to access the 36 index-addressable 8-bit registers.

To access a Mailbox register once the Mailbox Registers Interface Base Address has been initialized, the Mailbox register index address is first written to the MBX Index port. After the Index port has been written, the Mailbox data byte can be read or written via the MBX data port.

The Host Access Port is intended to be accessed by the Host only, however it may be accessed by the EC at the Offset shown from its EC base address in Table 13-2, "Runtime Register Base Address Table".

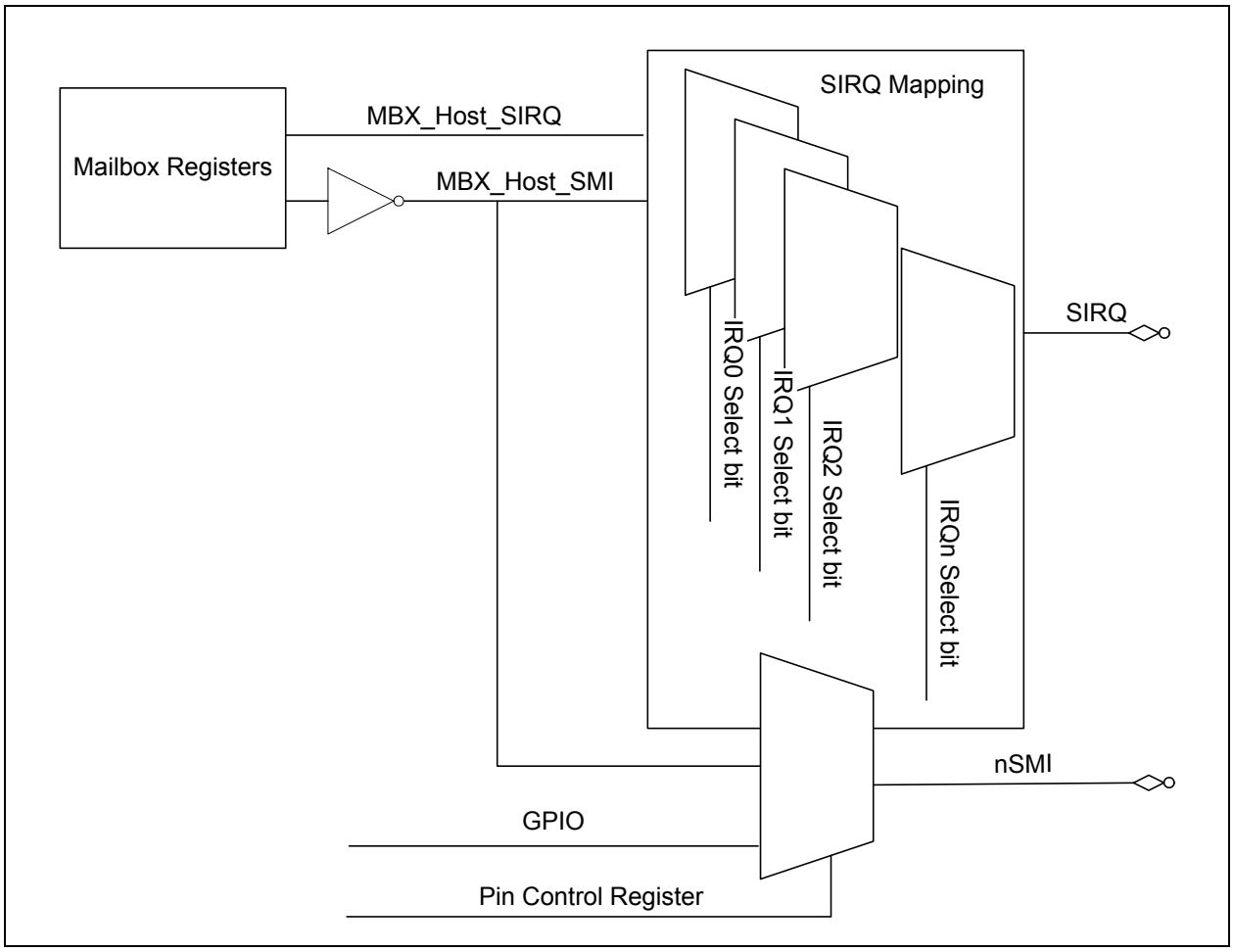
13.10.2 HOST INTERRUPT GENERATION

The Mailbox can generate a SIRQ event for EC-to-HOST EC events, using the EC-to-Host Mailbox Register. This interrupt is routed to the SIRQ block.

The Mailbox can also generate an SMI event, using SMI Interrupt Source Register. The SMI event can be routed to any frame in the SIRQ stream as well as to the nSMI pin. The SMI event can be routed to nSMI pin by selecting the nSMI signal function in the associated GPIO Pin Control Register. The SMI event produces a standard active low frame on the serial IRQ stream and active low level on the open drain nSMI pin.

Routing for both the SIRQ logic and the nSMI pin is shown in FIGURE 13-3:

FIGURE 13-3: MAILBOX SIRQ AND SMI ROUTING



13.10.3 EC MAILBOX CONTROL

The HOST-to-EC Mailbox Register and EC-to-Host Mailbox Register are designed to pass commands between the host and the EC. If enabled, these registers can generate interrupts to both the Host and the EC.

The two registers are not dual-ported, so the HOST BIOS and Keyboard BIOS must be designed to properly share these registers. When the host performs a write of the HOST-to-EC Mailbox Register, an interrupt will be generated and seen by the EC if unmasked. When the EC writes FFh to the Mailbox Register, the register resets to 00h, providing a simple means for the EC to inform the host that an operation has been completed.

When the EC writes the EC-to-Host Mailbox Register, an SMI may be generated and seen by the host if unmasked. When the Host CPU writes FFh to the register, the register resets to 00h, providing a simple means for the host to inform that EC that an operation has been completed.

Note:	The protocol used to pass commands back and forth through the Mailbox Registers Interface is left to the System designer. Microchip can provide an application example of working code in which the host uses the Mailbox registers to gain access to all of the EC registers.

13.11 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the Mailbox. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the Runtime Register Base Address Table.

TABLE 13-2: RUNTIME REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
Mailbox Interface	0	LPC	I/O	Programmed BAR
		EC	32-bit address space	000F_2400h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 13-3: RUNTIME REGISTER SUMMARY

Offset	Register Name (Mnemonic)
0h	MBX_INDEX Register
4h	MBX_DATA Register

13.11.1 MBX_INDEX REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	INDEX The index into the mailbox registers listed in Table 13-5, "EC-Only Register Summary".	R/W	0h	nSYSRST and VCC_PWRGD= 0

13.11.2 MBX_DATA REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	DATA Data port used to access the registers listed in Table 13-5, "EC-Only Register Summary".	R/W	0h	nSYSRST and VCC_PWRGD= 0

13.12 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Mailbox. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 13-4: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
Mailbox Interface	0	EC	32-bit address space	000F_2500h

The EC-Only registers can be accessed by the EC at the EC Offset from the Base Address. In addition, the registers can be accessed through the Host Access Port, at the indexes listed in the following tables as “MBX_INDEX”.

TABLE 13-5: EC-ONLY REGISTER SUMMARY

EC Offset	Host I/O Index (MBX_INDEX)	Register Name (Mnemonic)
00h	00h	HOST-to-EC Mailbox Register
04h	01h	EC-to-Host Mailbox Register
08h	02h	SMI Interrupt Source Register
0Ch	03h	SMI Interrupt Mask Register
10h	10h	Mailbox register [0]
	11h	Mailbox register [1]
	12h	Mailbox register [2]
	13h	Mailbox register [3]
14h	14h	Mailbox register [4]
	15h	Mailbox register [5]
	16h	Mailbox register [6]
	17h	Mailbox register [7]
18h	18h	Mailbox register [8]
	19h	Mailbox register [9]
	1Ah	Mailbox register [A]
	1Bh	Mailbox register [B]
1Ch	1Ch	Mailbox register [C]
	1Dh	Mailbox register [D]
	1Eh	Mailbox register [E]
	1Fh	Mailbox register [F]

TABLE 13-5: EC-ONLY REGISTER SUMMARY (CONTINUED)

EC Offset	Host I/O Index (MBX_INDEX)	Register Name (Mnemonic)
20h	20h	Mailbox register [10]
	21h	Mailbox register [11]
	22h	Mailbox register [12]
	23h	Mailbox register [13]
24h	24h	Mailbox register [14]
	25h	Mailbox register [15]
	26h	Mailbox register [16]
	27h	Mailbox register [17]
28h	28h	Mailbox register [18]
	29h	Mailbox register [19]
	2Ah	Mailbox register [1A]
	2Bh	Mailbox register [1B]
2Ch	2Ch	Mailbox register [1C]
	2Dh	Mailbox register [1D]
	2Eh	Mailbox register [1E]
	2Fh	Mailbox register [1F]

13.12.1 HOST-TO-EC MAILBOX REGISTER

Offset	0h			
MBX_ INDEX	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	HOST_EC_MBOX If enabled, an interrupt to the EC marked by the MBX_DATA bit in the Interrupt Aggregator will be generated whenever the Host writes this register. This register is cleared when written with FFh.	R/W	0h	nSYSRST

13.12.2 EC-TO-HOST MAILBOX REGISTER

Offset	4h			
MBX_ INDEX	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_HOST_MBOX An EC write to this register will set bit EC_WR in the SMI Interrupt Source Register to ‘1b’. If enabled, this will’ generate a Host SMI. This register is cleared when written with FFh.	R	0h	nSYSRST

13.12.3 SMI INTERRUPT SOURCE REGISTER

Offset	8h			
MBX_ INDEX	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	EC_SWI EC Software Interrupt. An SIRQ to the Host is generated when any bit in this register when this bit is set to ‘1b’ and the corresponding bit in the SMI Interrupt Mask Register register is ‘1b’. This field is Read/Write when accessed by the EC at the EC offset. When written through the Host Access Port, each bit in this field is cleared when written with a ‘1b’. Writes of ‘0b’ have no effect.	Host Access Port: R/WC EC: R/W	0h	nSYSRST
0	EC_WR EC Mailbox Write. This bit is set automatically when the EC-to-Host Mailbox Register has been written. An SMI or SIRQ to the Host is generated when n this bit is ‘1b’ and the corresponding bit in the SMI Interrupt Mask Register register is ‘1b’. This bit is automatically cleared by a read of the EC-to-Host Mail-box Register through the Host Access Port. This bit is read-only when read through the Host Access Port. It is neither readable nor writable directly by the EC when accessed at the EC offset.	Host Access Port: R EC: -	0h	nSYSRST

13.12.4 SMI INTERRUPT MASK REGISTER

Offset	Ch			
MBX_ INDEX	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	EC_SWI_EN EC Software Interrupt Enable. If this bit is ‘1b’, the bit EC_WR in the SMI Interrupt Source Register is enabled for the generation of SIRQ or nSMI events.	R/W	0h	nSYSRST
0	EC_WR_EN EC Mailbox Write.Interrupt Enable. Each bit in this field that is ‘1b’ enables the generation of SIRQ interrupts when the corresponding bit in the EC_SWI field in the SMI Interrupt Source Register is ‘1b’.	R	0h	nSYSRST

14.0 ACPI EMBEDDED CONTROLLER INTERFACE (ACPI-ECI)

14.1 Introduction

The ACPI Embedded Controller Interface (ACPI-ECI) is a Host/EC Message Interface. The ACPI specification defines the standard hardware and software communications interface between the OS and an embedded controller. This interface allows the OS to support a standard driver that can directly communicate with the embedded controller, allowing other drivers within the system to communicate with and use the EC resources; for example, Smart Battery and AML code.

The ACPI Embedded Controller Interface (ACPI-ECI) provides a four byte full duplex data interface which is a superset of the standard ACPI Embedded Controller Interface (ACPI-ECI) one byte data interface. The ACPI Embedded Controller Interface (ACPI-ECI) defaults to the standard one byte interface.

The MEC140x/1x has two instances of the ACPI Embedded Controller Interface.

1. The EC host in TABLE 14-4: and TABLE 14-6: corresponds to the EC in the ACPI specification. This interface is referred to elsewhere in this chapter as ACPI_EC.
2. The LPC host in TABLE 14-4: and TABLE 14-6: corresponds to the “System Host Interface to OS” in the ACPI specification. This interface is referred to elsewhere in this chapter as ACPI_OS.

14.2 References

- Advanced Configuration and Power Interface Specification, Revision 4.0 June 16, 2009, Hewlett-Packard Corporation Intel Corporation Microsoft Corporation Phoenix Technologies Ltd. Toshiba Corporation

14.3 Terminology

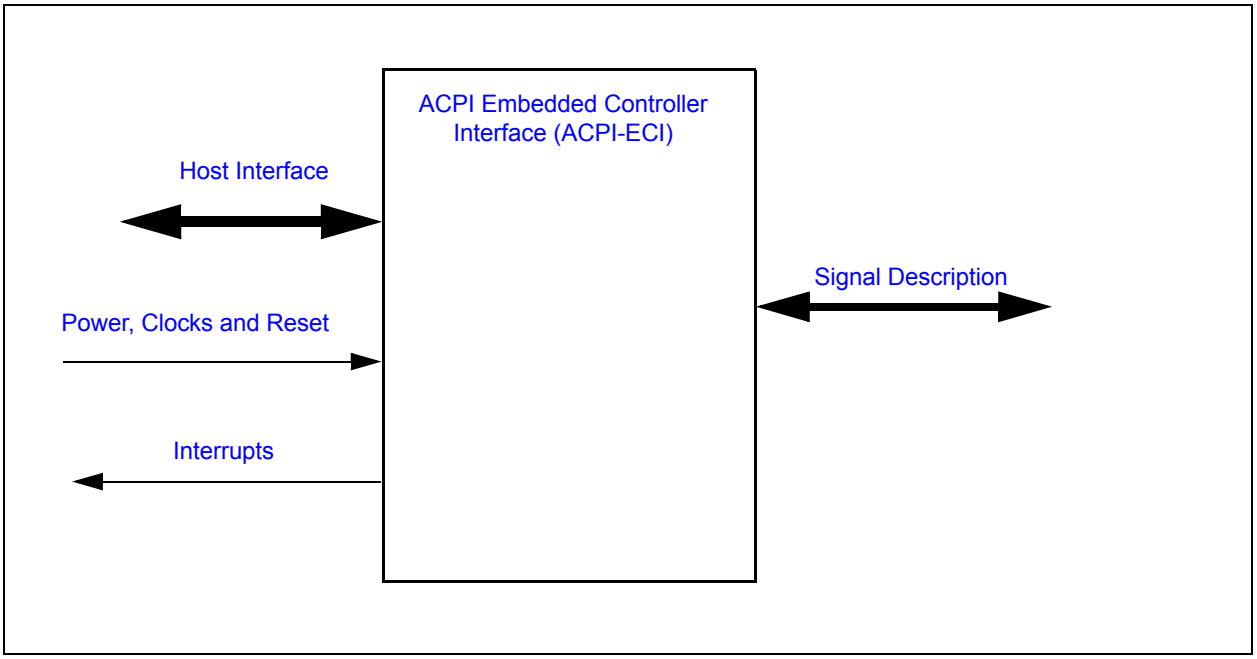
TABLE 14-1: TERMINOLOGY

Term	Definition
ACPI_EC	The EC host corresponding to the ACPI specification interface to the EC.
ACPI_OS	The LPC host corresponding to the ACPI specification interface to the “System Host Interface to OS”. ACPI_OS terminology is not meant to distinguish the ACPI System Management from Operating System but merely the hardware path upstream towards the CPU.

14.4 Interface

This block is designed to be accessed externally and internally via a register interface.

FIGURE 14-1: I/O DIAGRAM OF BLOCK



14.5 Signal Description

There are no external signals.

14.6 Host Interface

The registers defined for the ACPI Embedded Controller Interface (ACPI-ECI) are accessible by the System Host and the Embedded Controller as indicated in Section 14.12, "Runtime Registers" and Section 14.13, "EC-Only Registers".

14.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

14.7.1 POWER DOMAINS

Name	Description
VTR	The logic and registers implemented in this block reside on this single power well.

14.7.2 CLOCK INPUTS

This block only requires the Host interface clocks to synchronize registers access.

14.7.3 RESETS

Name	Description
nSYSRST	nSYSRST resets all the logic and registers in ACPI Embedded Controller Interface (ACPI-ECI).

14.8 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
OBF	OBF interrupt is asserted when the OBF in the EC STATUS Register is cleared to ‘0’.
IBF	IBF interrupt is asserted when the IBF in the EC STATUS Register is set to ‘1’.

Note: The usage model from the ACPI specification requires both SMI’s and SCI’s. The ACPI_OS SMI & SCI interrupts are not implemented in the ACPI Embedded Controller Interface (ACPI-ECI). The SMI_EVT and SCI_EVT bits in the OS STATUS OS Register are software flags and this block do not initiate SMI or SCI events.

14.9 Low Power Modes

The ACPI Embedded Controller Interface (ACPI-ECI) automatically enters low power mode when no transaction targets it.

14.10 Description

The ACPI Embedded Controller Interface (ACPI-ECI) provides an APCI-EC interface that adheres to the ACPI specification. The ACPI Embedded Controller Interface (ACPI-ECI) includes two modes of operation: Legacy Mode and Four-byte Mode.

The ACPI Embedded Controller Interface (ACPI-ECI) defaults to Legacy Mode which provides single byte Full Duplex operation. Legacy Mode corresponds to the ACPI specification functionality as illustrated in FIGURE 14-2: on page 223. The EC interrupts in FIGURE 14-2: on page 223 are implemented as OBF & IBF. See Section 14.8, "Interrupts," on page 222.

In Four-byte Mode, the ACPI Embedded Controller Interface (ACPI-ECI) provides four byte Full Duplex operation. Four-byte Mode is a superset of the ACPI specification functionality as illustrated in FIGURE 14-2: on page 223.

Both Legacy Mode & Four-byte Mode provide Full Duplex Communications which allows data/command transfers in one direction while maintaining data from the other direction; communications can flow both ways simultaneously.

In Legacy Mode, ACPI Embedded Controller Interface (ACPI-ECI) contains three registers: ACPI OS COMMAND Register, OS STATUS OS Register, and OS2EC Data EC Byte 0 Register. The standard ACPI Embedded Controller Interface (ACPI-ECI) registers occupy two addresses in the ACPI_OS space (TABLE 14-5:).

The OS2EC Data EC Byte 0 Register and ACPI OS COMMAND Register registers appear as a single 8-bit data register in the ACPI_EC. The CMD bit in the OS STATUS OS Register is used by the ACPI_EC to discriminate commands from data written by the ACPI_OS to the ACPI_EC. CMD bit is controlled by hardware: ACPI_OS writes to the OS2EC Data EC Byte 0 Register register clear the CMD bit; ACPI_OS writes to the ACPI OS COMMAND Register set the CMD bit.

FIGURE 14-2: BLOCK DIAGRAM CORRESPONDING TO THE ACPI SPECIFICATION

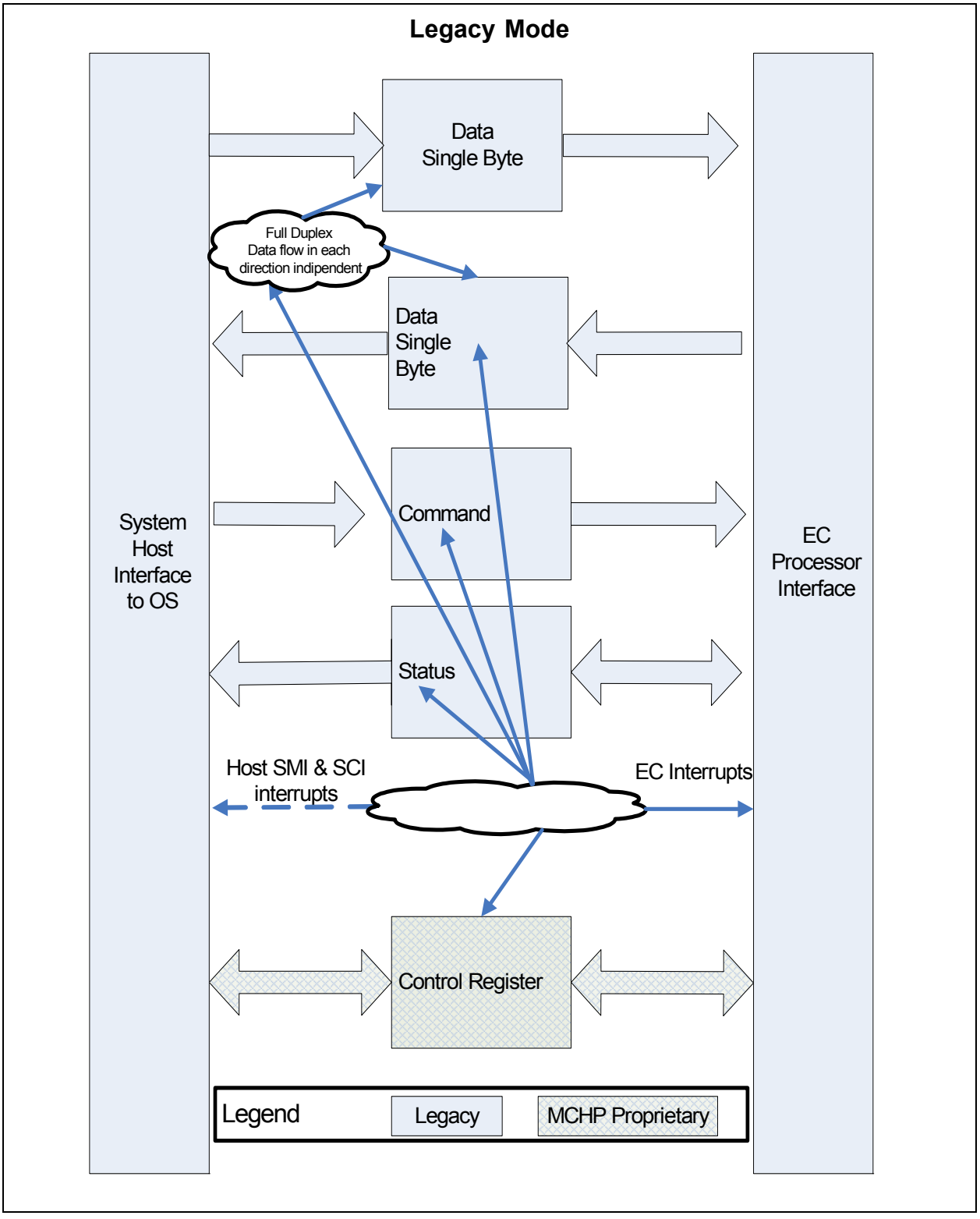
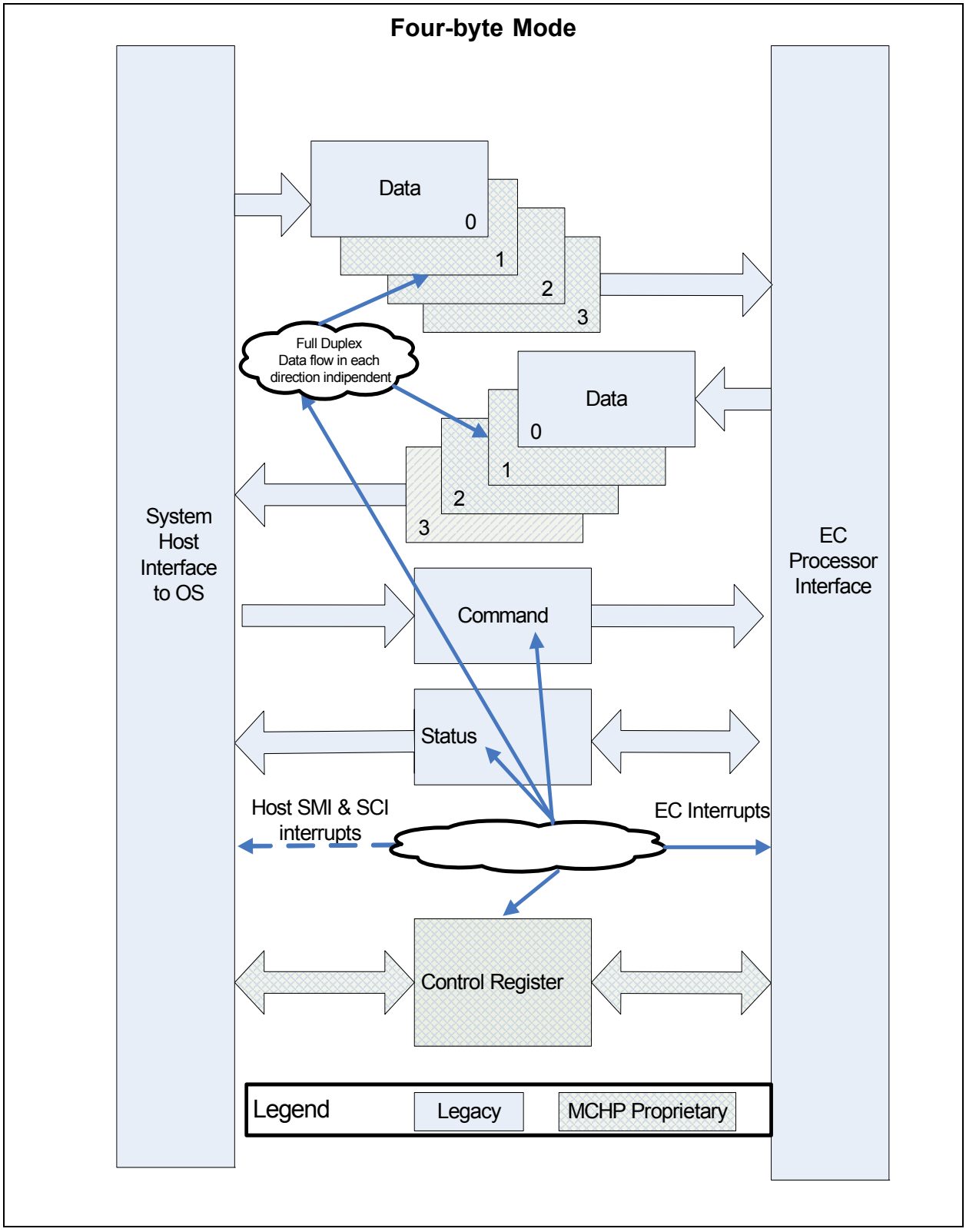


FIGURE 14-2: BLOCK DIAGRAM CORRESPONDING TO THE ACPI SPECIFICATION



14.11 Register Aliasing between Runtime and EC-Only Registers

Table 14-2, "Runtime Register Aliasing into EC-Only Registers" indicates the aliasing from Runtime registers to EC-Only registers. The “Host/EC Access” column distinguishes the aliasing based on access type. See individual register descriptions for more details.

TABLE 14-2: RUNTIME REGISTER ALIASING INTO EC-ONLY REGISTERS

Host Offset	Runtime Register Register Name (Mnemonic)	Host Access	EC Offset	Aliased EC-Only Register Register Name (Mnemonic)	EC Access
00h	ACPI OS Data Register Byte 0 Register	W	108h	OS2EC Data EC Byte 0 Register	R
00h	ACPI OS Data Register Byte 0 Register	R	100h	EC2OS Data EC Byte 0 Register	W
01h	ACPI OS Data Register Byte 1 Register	W	109h	OS2EC Data EC Byte 1 Register	R
01h	ACPI OS Data Register Byte 1 Register	R	101h	EC2OS Data EC Byte 1 Register	W
02h	ACPI OS Data Register Byte 2 Register	W	10Ah	OS2EC Data EC Byte 2 Register	R
02h	ACPI OS Data Register Byte 2 Register	R	102h	EC2OS Data EC Byte 2 Register	W
03h	ACPI OS Data Register Byte 3 Register	W	10Bh	OS2EC Data EC Byte 3 Register	R
03h	ACPI OS Data Register Byte 3 Register	R	103h	EC2OS Data EC Byte 3 Register	W
04h	ACPI OS COMMAND Register	W	108h	OS2EC Data EC Byte 0 Register	R
04h	OS STATUS OS Register	R	104h	EC STATUS Register	W
05h	OS Byte Control Register	R	105h	EC Byte Control Register	R/W
06h	Reserved		106h	Reserved	
07h	Reserved		107h	Reserved	

Table 14-3, "EC-Only Registers Summary" indicates the aliasing from EC-Only to Runtime registers. The “Host/EC Access” column distinguishes the aliasing based on access type. See individual register descriptions for more details.

TABLE 14-3: EC-ONLY REGISTERS SUMMARY

EC Offset	EC-Only Registers Register Name (Mnemonic)	EC Access	Host Offset	Aliased Runtime Register Register Name (Mnemonic)	Host Access
108h	OS2EC Data EC Byte 0 Register	R	00h	ACPI OS Data Register Byte 0 Register	W
108h	OS2EC Data EC Byte 0 Register	R	04h	ACPI OS COMMAND Register	W

TABLE 14-3: EC-ONLY REGISTERS SUMMARY

EC Offset	EC-Only Registers Register Name (Mnemonic)	EC Access	Host Offset	Aliased Runtime Register Register Name (Mnemonic)	Host Access
109h	OS2EC Data EC Byte 1 Register	R	01h	ACPI OS Data Register Byte 1 Register	W
10Ah	OS2EC Data EC Byte 2 Register	R	02h	ACPI OS Data Register Byte 2 Register	W
10Bh	OS2EC Data EC Byte 3 Register	R	03h	ACPI OS Data Register Byte 3 Register	W
104h	EC STATUS Register	W	04h	OS STATUS OS Register	W
105h	EC Byte Control Register	R/W	05h	OS Byte Control Register	R
106h	Reserved	R		Reserved	R
107h	Reserved	R		Reserved	R
100h	EC2OS Data EC Byte 0 Register	W	00h	ACPI OS Data Register Byte 0 Register	R
101h	EC2OS Data EC Byte 1 Register	W	01h	ACPI OS Data Register Byte 1 Register	R
102h	EC2OS Data EC Byte 2 Register	W	02h	ACPI OS Data Register Byte 2 Register	R
103h	EC2OS Data EC Byte 3 Register	W	03h	ACPI OS Data Register Byte 3 Register	R

14.12 Runtime Registers

The registers listed in the Runtime Register Summary table are for four instances of the ACPI Embedded Controller Interface (ACPI-ECI). The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the Runtime Register Base Address Table.

Note: The Runtime registers may be accessed by the EC but typically the Host will access the Runtime Registers and the EC will access just the EC-Only registers.

TABLE 14-4: RUNTIME REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
ACPI-EC	0	LPC	I/O	Programmed BAR
		EC	32-bit internal address space	000F_0C00h
ACPI-EC	1	LPC	I/O	Programmed BAR
		EC	32-bit internal address space	000F_1000h
ACPI-EC	2	LPC	I/O	Programmed BAR
		EC	32-bit internal address space	000F_2800h

TABLE 14-4: RUNTIME REGISTER BASE ADDRESS (CONTINUED)

Block Instance	Instance Number	Host	Address Space	Base Address
ACPI-EC	3	LPC	I/O	Programmed BAR
		EC	32-bit internal address space	000F_2C00h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 14-5: RUNTIME REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	ACPI OS Data Register Byte 0 Register
01h	ACPI OS Data Register Byte 1 Register
02h	ACPI OS Data Register Byte 2 Register
03h	ACPI OS Data Register Byte 3 Register
04h	ACPI OS COMMAND Register
04h	OS STATUS OS Register
05h	OS Byte Control Register
06h	Reserved
07h	Reserved

14.12.1 ACPI OS DATA REGISTER BYTE 0 REGISTER

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 227, OS2EC DATA BYTES[3:0] on page 234, and EC2OS DATA BYTES[3:0] on page 236 for detailed description of access rules.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI_OS_DATA_BYTE_0 This is byte 0 of the 32-bit ACPI-OS DATA BYTES[3:0].	R/W	0h	nSYSRST

ACPI-OS DATA BYTES[3:0]

Writes by the ACPI_OS to the ACPI-OS DATA BYTES[3:0] are aliased to the OS2EC DATA BYTES[3:0]. Reads by the ACPI_OS from the ACPI-OS DATA BYTES[3:0] are aliased to the EC2OS DATA BYTES[3:0].

All access to the ACPI-OS DATA BYTES[3:0] registers should be orderly: Least Significant Byte to Most Significant Byte when byte access is used.

Writes to any of the four ACPI-OS DATA BYTES[3:0] registers clears the CMD bit in the OS STATUS OS Register (the state of the FOUR_BYTE_ACCESS (see Note) bit in the OS Byte Control Register has no impact.)

When the FOUR_BYTE_ACCESS (see Note) bit in the OS Byte Control Register is cleared to '0', the following access rules apply:

1. Writes to the ACPI OS Data Register Byte 0 Register sets the IBF bit in the OS STATUS OS Register.
2. Reads from the ACPI OS Data Register Byte 0 Register clears the OBF bit in the OS STATUS OS Register.
3. All writes to ACPI-OS DATA BYTES[3:1] complete without error but the data are not registered.
4. All reads from ACPI-OS DATA BYTES[3:1] return 00h without error.
5. Access to ACPI-OS DATA BYTES[3:1] has no effect on the IBF & OBF bits in the OS STATUS OS Register.

When the Four Byte Access bit in the OS Byte Control Register is set to '1', the following access rules apply (see **Note**):

1. Writes to the ACPI OS Data Register Byte 3 Register sets the IBF bit in the OS STATUS OS Register.
2. Reads from the ACPI OS Data Register Byte 3 Register clears the OBF bit in the OS STATUS OS Register.

Note:	In eSPI mode, instance 0 of the ACPI Embedded Controller Interface (ACPI-EC0) only operates in Legacy Mode which provides single byte Full Duplex operation. Four-byte Mode is not supported for ACPI-EC0 in eSPI mode.

14.12.2 ACPI OS DATA REGISTER BYTE 1 REGISTER

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 227, OS2EC DATA BYTES[3:0] on page 234, and EC2OS DATA BYTES[3:0] on page 236 for detailed description of access rules.

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI_OS_DATA_BYTE_1 This is byte 1 of the 32-bit ACPI-OS DATA BYTES[3:0].	R/W	0h	nSYSRST

14.12.3 ACPI OS DATA REGISTER BYTE 2 REGISTER

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 227, OS2EC DATA BYTES[3:0] on page 234, and EC2OS DATA BYTES[3:0] on page 236 for detailed description of access rules.

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI_OS_DATA_BYTE_2 This is byte 2 of the 32-bit ACPI-OS DATA BYTES[3:0].	R/W	0h	nSYSRST

14.12.4 ACPI OS DATA REGISTER BYTE 3 REGISTER

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 227, OS2EC DATA BYTES[3:0] on page 234, and EC2OS DATA BYTES[3:0] on page 236 for detailed description of access rules.

Offset	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI_OS_DATA_BYTE_3 This is byte 3 of the 32-bit ACPI-OS DATA BYTES[3:0].	R/W	0h	nSYSRST

14.12.5 ACPI OS COMMAND REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI_OSS_COMMAND Writes to the this register are aliased in the OS2EC Data EC Byte 0 Register. Writes to the this register also set the CMD and IBF bits in the OS STATUS OS Register	W	0h	nSYSRST

14.12.6 OS STATUS OS REGISTER

This read-only register is aliased to the EC STATUS Register on page 237. the EC STATUS Register on page 237 has read write access.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	UD0B User Defined	R	0b	nSYSRST
6	SMI_EVT This bit is set when an SMI event is pending; i.e., the ACPI_EC is requesting an SMI query; This bit is cleared when no SMI events are pending. This bit is an ACPI_EC-maintained software flag that is set when the ACPI_EC has detected an internal event that requires system management interrupt handler attention. The ACPI_EC sets this bit before generating an SMI. **Note:** The usage model from the ACPI specification requires both SMI's and SCI's. The ACPI_OS SMI & SCI interrupts are not implemented in the ACPI Embedded Controller Interface (ACPI-ECI). The SMI_EVT and SCI_EVT bits in the OS STATUS OS Register are software flags and this block do not initiate SMI or SCI events.	R	0b	nSYSRST

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
5	SCI_EVT This bit is set by software when an SCI event is pending; i.e., the ACPI_EC is requesting an SCI query; SCI Event flag is clear when no SCI events are pending. This bit is an ACPI_EC-maintained software flag that is set when the embedded controller has detected an internal event that requires operating system attention. The ACPI_EC sets this bit before generating an SCI to the OS. **Note:** The usage model from the ACPI specification requires both SMI's and SCI's. The ACPI_OS SMI & SCI interrupts are not implemented in the ACPI Embedded Controller Interface (ACPI-ECI). The SMI_EVT and SCI_EVT bits in the OS STATUS OS Register are software flags and this block do not initiate SMI or SCI events.	R	0b	nSYSRST
4	BURST The BURST bit is set when the ACPI_EC is in Burst Mode for polled command processing; the BURST bit is cleared when the ACPI_EC is in Normal mode for interrupt-driven command processing. The BURST bit is an ACPI_EC-maintained software flag that indicates the embedded controller has received the Burst Enable command from the host, has halted normal processing, and is waiting for a series of commands to be sent from the host. Burst Mode allows the OS or system management handler to quickly read and write several bytes of data at a time without the overhead of SCIs between commands. The BURST bit is maintained by ACPI_EC software, only.	R	0b	nSYSRST
3	CMD This bit is set when the OS2EC Data EC Byte 0 Register contains a command byte written into ACPI OS COMMAND Register; this bit is cleared when the OS2EC DATA BYTES[3:0] contains a data byte written into the ACPI-OS DATA BYTES[3:0]. This bit is hardware controlled: • ACPI_OS writes to any of the four ACPI-OS DATA BYTES[3:0] bytes clears this bit • ACPI_OS writes to the ACPI OS COMMAND Register sets this bit. **Note:** This bit allows the embedded controller to differentiate the start of a command sequence from a data byte write operation.	R	0b	nSYSRST
2	UD1B User Defined	R	0b	nSYSRST

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	IBF The Input Buffer Full bit is set to indicate that a the ACPI_OS has written a command or data to the ACPI_EC and that data is ready. This bit is automatically cleared when data has been read by the ACPI_EC. **Note:** The setting and clearing of this IBF varies depending on the setting of the following bits: CMD bit in this register and FOUR_BYTE_ACCESS (see Note) bit in the OS Byte Control Register. Three scenarios follow: 1. The IBF is set when the ACPI_OS writes to the ACPI OS COMMAND Register. This same write autonomously sets the CMD bit in this register. The IBF is cleared if the CMD bit in this register is set and the ACPI_EC reads from the OS2EC Data EC Byte 0 Register. **Note:** When CMD bit in this register is set the FOUR_BYTE_ACCESS (see Note) bit in the OS Byte Control Register has no impact on the IBF bit behavior. 2. A write by the to the ACPI_OS to the ACPI OS Data Register Byte 0 Register sets the IBF bit if the FOUR_BYTE_ACCESS (see Note) bit in the OS Byte Control Register is in the cleared to '0' state prior to this write. This same write autonomously clears the CMD bit in this register. A read of the OS2EC Data EC Byte 0 Register clears the IBF bit if the FOUR_BYTE_ACCESS (see Note) bit in the OS Byte Control Register is in the cleared to '0' state prior to this read. 3. A write by the to the ACPI_OS to the ACPI OS Data Register Byte 3 Register sets the IBF bit if the FOUR_BYTE_ACCESS (see Note) bit in the OS Byte Control Register is in the set to '1' state prior to this write. This same write autonomously clears the CMD bit in this register. A read of the OS2EC Data EC Byte 3 Register clears the IBF bit if the FOUR_BYTE_ACCESS (see Note) bit in the OS Byte Control Register is in the set to '1' state prior to this read. An IBF interrupt signals the ACPI_EC that there is data available. The ACPI Specification usage model is as follows: 1. The ACPI_EC reads the EC STATUS Register and sees the IBF flag set, 2. The ACPI_EC reads all the data available in the OS2EC DATA BYTES[3:0]. This causes the IBF bit to be automatically cleared by hardware. 3. The ACPI_EC must then generate a software interrupt (See Note: on page 222) to alert the ACPI_OS that the data has been read and that the host is free to write more data to the ACPI_EC as needed.	R	0h	nSYSRST

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	OBF The Output Buffer Full bit is set to indicate that a the ACPI_EC has written a data to the ACPI_OS and that data is ready. This bit is automatically cleared when all the data has been read by the ACPI_OS. **Note:** The setting and clearing of this OBF varies depending on the setting FOUR_BYTE_ACCESS (see Note) bit in the OS Byte Control Register. Two scenarios follow: 1. The OBF bit is set if the Four Byte Access bit in the OS Byte Control Register is '0' when the ACPI_EC writes to the EC2OS Data EC Byte 0 Register. The OBF is cleared if the Four Byte Access bit in the OS Byte Control Register is cleared to '0' when the ACPI_OS reads from the ACPI OS Data Register Byte 0 Register. 2. The OBF is set if the Four Byte Access bit in the OS Byte Control Register is set to '1' when the ACPI_EC writes to the EC2OS Data EC Byte 3 Register. The OBF is cleared if the Four Byte Access bit in the OS Byte Control Register is set to '1' when the ACPI_OS reads from the ACPI OS Data Register Byte 3 Register. The ACPI Specification usage model is as follows: 1. The ACPI_EC must generate a software interrupt (See Note: on page 222) to alert the ACPI_OS that the data is available. 2. The ACPI_OS reads the OS STATUS OS Register and sees the OBF flag set, the ACPI_OS reads all the data available in the ACPI-OS DATA BYTES[3:0]. 3. The ACPI_OS reads all the data available in the ACPI-OS DATA BYTES[3:0]. This causes the OBF bit to be automatically cleared by hardware and the associated OBF interrupt to be asserted.	R	0h	nSYSRST

14.12.7 OS BYTE CONTROL REGISTER

This register is aliased to the EC Byte Control Register on page 238. No behavioral differences occur due to address aliasing.

Offset	05			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	R	-	-
0	FOUR_BYTE_ACCESS (see **Note**) When this bit is set to '1', the ACPI Embedded Controller Interface (ACPI-ECI) accesses four bytes through the ACPI-OS DATA BYTES[3:0]. When this bit is cleared to '0', the ACPI Embedded Controller Interface (ACPI-ECI) accesses one byte through the ACPI OS Data Register Byte 0 Register. The corresponds to Legacy Mode described in Section 14.10, "Description," on page 222. **Note 1:** This bit effects the behavior of the IBF & OBF bits in the OS STATUS OS Register. **2:** See ACPI-OS DATA BYTES[3:0] on page 227, OS2EC DATA BYTES[3:0] on page 234, and EC2OS DATA BYTES[3:0] on page 236 for detailed description of access rules.	R	0b	nSYSRST

Note:	The ACPI_OS access Base Address Register (BAR) should be configured to match the access width selected by the Four Byte Access bit in the OS Byte Control Register. This BAR in not described in this chapter.

14.13 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for four instances of the ACPI Embedded Controller Interface (ACPI-ECI). The addresses of each register listed in this table are defined as a relative offset to the host "Base Address" defined in the EC-Only Register Base Address Table.

TABLE 14-6: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
ACPI-EC	0	EC	32-bit internal address space	000F_0C00h
ACPI-EC	1	EC	32-bit internal address space	000F_1000h
ACPI-EC	2	EC	32-bit internal address space	000F_2800h
ACPI-EC	3	EC	32-bit internal address space	000F_2C00h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 14-7: EC-ONLY REGISTER SUMMARY

Offset	Register Name (Mnemonic)
100h	EC2OS Data EC Byte 0 Register
101h	EC2OS Data EC Byte 1 Register
102h	EC2OS Data EC Byte 2 Register
103h	EC2OS Data EC Byte 3 Register
104h	EC STATUS Register
105h	EC Byte Control Register
106h	Reserved
107h	Reserved
108h	OS2EC Data EC Byte 0 Register
109h	OS2EC Data EC Byte 1 Register
10Ah	OS2EC Data EC Byte 2 Register
10Bh	OS2EC Data EC Byte 3 Register

14.13.1 OS2EC DATA EC BYTE 0 REGISTER

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 227, OS2EC DATA BYTES[3:0] on page 234, and EC2OS DATA BYTES[3:0] on page 236 for detailed description of access rules.

Offset	108h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS_TO_EC_DATA_BYTE_0 This is byte 0 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	nSYSRST

OS2EC DATA BYTES[3:0]

When the CMD bit in the OS STATUS OS Register is cleared to ‘0’, reads by the ACPI_EC from the OS2EC DATA BYTES[3:0] are aliased to the ACPI-OS DATA BYTES[3:0].

All access to the OS2EC DATA BYTES[3:0] registers should be orderly: Least Significant Byte to Most Significant Byte when byte access is used.

When the FOUR_BYTE_ACCESS (see Note) bit in the OS Byte Control Register is cleared to ‘0’, the following access rules apply:

1. Writes to the OS2EC DATA BYTES[3:0] have no effect on the OBF bit in the OS STATUS OS Register.
2. Reads from the OS2EC Data EC Byte 0 Register clears the IBF bit in the OS STATUS OS Register.
3. All reads from OS2EC DATA BYTES[3:1] return 00h without error.
4. Access to OS2EC DATA BYTES[3:1 has no effect on the IBF & OBF bits in the OS STATUS OS Register.

When the FOUR_BYTE_ACCESS (see Note) bit in the OS Byte Control Register is set to ‘1’, the following access rules apply:

1. Writes to the OS2EC DATA BYTES[3:0] have no effect on the OBF bit in the OS STATUS OS Register.
2. Reads from the OS2EC Data EC Byte 3 Register clears the IBF bit in the OS STATUS OS Register.

14.13.2 OS2EC DATA EC BYTE 1 REGISTER

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 227, OS2EC DATA BYTES[3:0] on page 234, and EC2OS DATA BYTES[3:0] on page 236 for detailed description of access rules.

Offset	109h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS2EC_DATA_ BYTE_1 This is byte 1 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	nSYSRST

14.13.3 OS2EC DATA EC BYTE 2 REGISTER

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 227, OS2EC DATA BYTES[3:0] on page 234, and EC2OS DATA BYTES[3:0] on page 236 for detailed description of access rules.

Offset	10Ah			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS2EC_DATA_BYTE_2 This is byte 2 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	nSYSRST

14.13.4 OS2EC DATA EC BYTE 3 REGISTER

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 227, OS2EC DATA BYTES[3:0] on page 234, and EC2OS DATA BYTES[3:0] on page 236 for detailed description of access rules.

Offset	10Bh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS2EC_DATA_BYTE_3 This is byte 3 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	nSYSRST

14.13.5 EC2OS DATA EC BYTE 0 REGISTER

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 227, OS2EC DATA BYTES[3:0] on page 234, and EC2OS DATA BYTES[3:0] on page 236 for detailed description of access rules.

Offset	100h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC2OS_DATA_BYTE_0 This is byte 0 of the 32-bit EC2OS DATA BYTES[3:0].	R/W	0h	nSYSRST

EC2OS DATA BYTES[3:0]

Writes by the ACPI_EC to the EC2OS DATA BYTES[3:0] are aliased to the ACPI-OS DATA BYTES[3:0]

All access to the EC2OS DATA BYTES[3:0] registers should be orderly: Least Significant Byte to Most Significant Byte when byte access is used.

When the FOUR_BYTE_ACCESS (see Note) bit in the OS Byte Control Register is cleared to '0', the following access rules apply:

1. Writes to the EC2OS Data EC Byte 0 Register set the OBF bit in the OS STATUS OS Register.
2. Reads from the EC2OS DATA BYTES[3:0] have no effect on the IBF bit in the OS STATUS OS Register.
3. All reads from EC2OS DATA BYTES[3:1] return 00h without error.
4. All writes to EC2OS DATA BYTES[3:1] complete without error but the data are not registered.
5. Access to EC2OS DATA BYTES[3:1] have no effect on the IBF & OBF bits in the OS STATUS OS Register.

When the FOUR_BYTE_ACCESS (see Note) bit in the OS Byte Control Register is set to '1', the following access rules apply:

1. Writes to the EC2OS Data EC Byte 3 Register set the OBF bit in the OS STATUS OS Register.
2. Reads from the EC2OS DATA BYTES[3:0] have no effect on the IBF bit in the OS STATUS OS Register.

14.13.6 EC2OS DATA EC BYTE 1 REGISTER

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 227, OS2EC DATA BYTES[3:0] on page 234, and EC2OS DATA BYTES[3:0] on page 236 for detailed description of access rules.

Offset	101h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC2OS_DATA_BYTE_1 This is byte 1 of the 32-bit EC2OS DATA BYTES[3:0].	R/W	0h	nSYSRST

14.13.7 EC2OS DATA EC BYTE 2 REGISTER

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 227, OS2EC DATA BYTES[3:0] on page 234, and EC2OS DATA BYTES[3:0] on page 236 for detailed description of access rules.

Offset	102h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC2OS_DATA_BYTE_2 This is byte 2 of the 32-bit EC2OS DATA BYTES[3:0].	R/W	0h	nSYSRST

14.13.8 EC2OS DATA EC BYTE 3 REGISTER

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 227, OS2EC DATA BYTES[3:0] on page 234, and EC2OS DATA BYTES[3:0] on page 236 for detailed description of access rules.

Offset	103h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC2OS_DATA_BYTE_3 This is byte 3 of the 32-bit EC2OS DATA BYTES[3:0].	R/W	0h	nSYSRST

14.13.9 EC STATUS REGISTER

This register is aliased to the OS STATUS OS Register on page 229. The OS STATUS OS Register is a read only version of this register.

Offset	104h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	UD0A User Defined	R/W	0b	nSYSRST
6	SMI_EVT See SMI_EVT bit in OS STATUS OS Register on page 229 for bit description.	R/W	0b	nSYSRST
5	SCI_EVT See SMI_EVT bit in OS STATUS OS Register on page 229 for bit description.	R/W	0b	nSYSRST
4	BURST See BURST bit in OS STATUS OS Register on page 229 for bit description.	R/W	0b	nSYSRST
3	CMD See CMD bit in OS STATUS OS Register on page 229 for bit description.	R	0b	nSYSRST
2	UD1A User Defined	R/W	0b	nSYSRST
1	IBF See IBF bit in OS STATUS OS Register on page 229 for bit description.	R	0h	nSYSRST
0	OBF See OBF bit in OS STATUS OS Register on page 229 for bit description.	R	0h	nSYSRST

Note: The IBF and OBF bits are not de-asserted by hardware when the host is powered off, or the LPC interface powers down; for example, following system state changes S3->S0, S5->S0, G3-> S0. For further information on how these bits are cleared, refer to IBF and OBF bit descriptions in the STATUS OS-Register definition.

14.13.10 EC BYTE CONTROL REGISTER

This register is aliased to the OS Byte Control Register on page 233. The OS Byte Control Register is a read only version of this register.

Offset	105h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	R	-	-
0	FOUR_BYTE_ACCESS See FOUR_BYTE_ACCESS (see Note) bit in OS Byte Control Register on page 233 for bit description.	R/W	0b	nSYSRST

15.0 ACPI PM1 BLOCK INTERFACE

15.1 Introduction

The MEC140x/1x supports ACPI as described in this section. These features comply with the ACPI Specification through a combination of hardware and EC software.

15.2 References

ACPI Specification, Revision 1.0

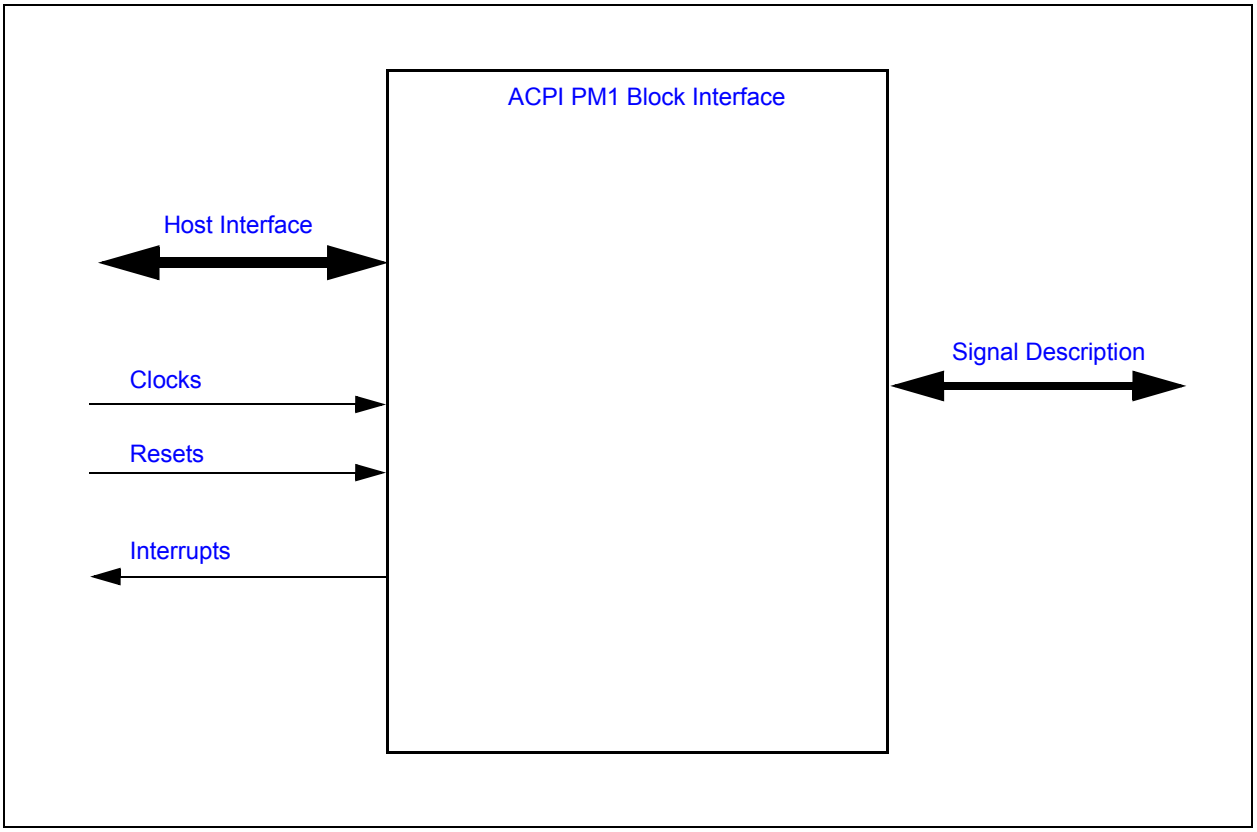
15.3 Terminology

None

15.4 Interface

This block is an IP block designed to be incorporated into a chip. It is designed to be accessed externally via the pin interface and internally via a registered host interface. The following diagram illustrates the various interfaces to the block.

FIGURE 15-1: I/O DIAGRAM OF BLOCK



15.5 Signal Description

Table 15-1, "ACPI PM1 Signal Description Table" lists the signals that are typically routed to the pin interface.

TABLE 15-1: ACPI PM1 SIGNAL DESCRIPTION TABLE

Name	Direction	Description
EC_SCI#	Output	Any or all of the PWRBTN_STS, SLPBTN_STS, and RTC_STS bits in the Power Management 1 Status 2 Register can assert the EC_SCI# pin if enabled by the associated bits in the Power Management 1 Enable 2 Register register. The EC_SCI_STS bit in the EC_PM_STS Register register can also be used to generate an SCI on the EC_SCI# pin.

15.6 Host Interface

The registers defined for the ACPI PM1 Block Interface are accessible by the various hosts as indicated in Section 15.11, "Runtime Registers".

15.7 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

15.7.1 POWER DOMAINS

Name	Description
VTR	This power well sources all of the registers and logic in this block, except where noted.

15.7.2 CLOCKS

This section describes all the clocks in the block, including those that are derived from the I/O Interface as well as the ones that are derived or generated internally.

Name	Description
48 MHz Ring Oscillator	This clock signal drives selected logic (e.g., counters).

15.7.3 RESETS

Name	Description
nSYSRST	This reset signal resets all of the registers and logic in this block.

15.8 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
PM1_CTL	This Interrupt is generated to the EC by the Host writing to the Power Management 1 Control 2 Register register
PM1_EN	This Interrupt is generated to the EC by the Host writing to the Power Management 1 Enable 2 Register register
PM1_STS	This Interrupt is generated to the EC by the Host writing to the Power Management 1 Status 2 Register register

15.9 Low Power Modes

The ACPI PM1 Block Interface may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

15.10 Description

This section describes the functions of the ACPI PM1 Block Interface in more detail.

The MEC140x/1x implements the ACPI fixed registers but includes only those bits that apply to the power button sleep button and RTC alarm events. The ACPI WAK_STS, SLP_TYP, and SLP_EN bits are also supported.

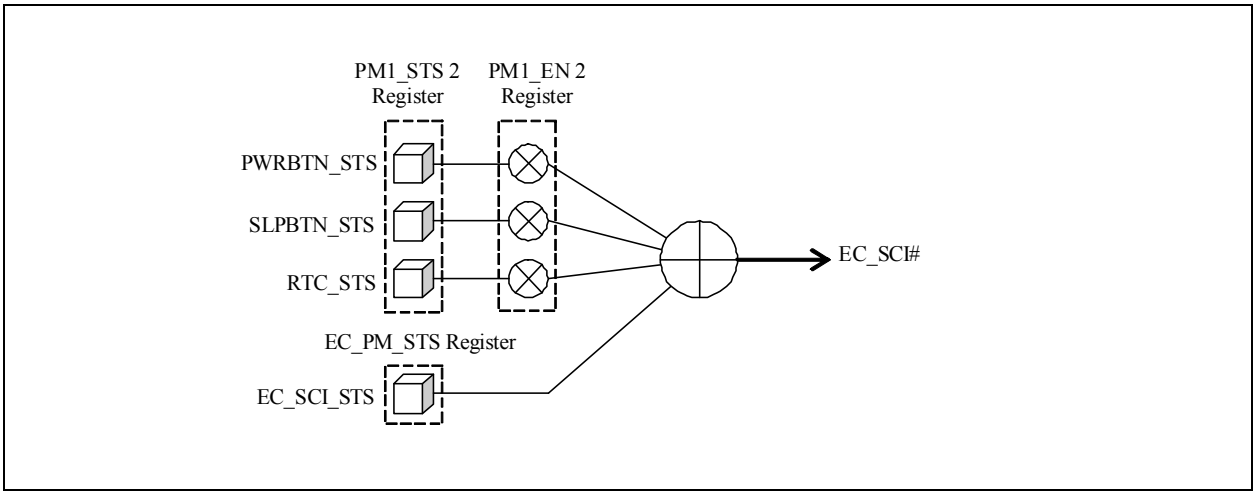
The MEC140x/1x can generate SCI Interrupts to the Host. The functions described in the following sub-sections can generate a SCI event on the EC_SCI# pin. In the MEC140x/1x, an SCI event is considered the same as an ACPI wakeup or runtime event.

15.10.1 SCI EVENT-GENERATING FUNCTIONS

Event	Event Bit	Definition
Power Button with Override	PWRBTN_STS	The power button has a status and an enable bit in the PM1_BLK of registers to provide an SCI upon the button press. The status bit is software Read/Writable by the EC; the enable bit is Read-only by the EC. It also has a status and enable bit in the PM1_BLK of registers to indicate and control the power button override (fail-safe) event. These bits are not required by ACPI. The PWRBTN_STS bit is set by the Host to enable the generation of an SCI due to the power button event. The status bit is set by the EC when it generates a power button event and is cleared by the Host writing a '1' to this bit (writing a '0' has no effect); it can also be cleared by the EC. If the enable bit is set, the EC generates an SCI power management event.
	PWRBTNOR_STS	The power button has a status and an enable bit in the PM1_BLK of registers to provide an SCI upon the power button override.The power button override event status bit is software Read/Writable by the EC; the enable bit is software read-only by the EC.The enable bit for the override event is located at bit 1 in the Power Management 1 Control Register 2 (PM1_CNTRL 2).The power button bit has a status and enable bit in the Runtime Registers to provide an SCI power management event on a button press The PWRBTNOR_STS bit is set by the Host to enable the generation of an SCI due to the power button override event. The status bit is set by the EC when it generates a power button event and is cleared by the Host writing a '1' to this bit (writing a '0' has no effect); it can also be cleared by the EC. If the enable bit is set, the EC generates an SCI power management event.
Sleep Button	SLPBTN_STS	The sleep button that has a status and an enable bit in the Runtime Registers to provide an SCI power management event on a button press. The status bit is software Read/Writable by the EC; the enable bit is Read-only by the EC. The SLPBTN_STS bit is set by the Host to enable the generation of an SCI due to the sleep button event. The status bit is set by the EC when it generates a sleep button event and is cleared by the Host writing a '1' to this bit (writing a '0' has no effect); it can also be cleared by the EC. If the enable bit is set, the EC will generate an SCI power management event.
RTC Alarm	RTC_STS	The ACPI specification requires that the RTC alarm generate a hardware wake-up event from the sleeping state. The RTC alarm can be enabled as an SCI event and its status can be determined through bits in the Runtime Registers. The status bit is software Read/Writable by the EC; the enable bit is Read-only by the EC. The RTC_STS bit is set by the Host to enable the generation of an SCI due to the RTC alarm event. The status bit is set by the EC when the RTC generates an alarm event and is cleared by the Host writing a '1' to this bit (writing a '0' has no effect); it can also be cleared by the EC. If the enable bit is set, the EC will generate an SCI power management event.

FIGURE 15-2: describes the relationship of PM1 Status and Enable bits to the EC_SCI# pin.

FIGURE 15-2: EC_SCI# INTERFACE



15.11 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the ACPI PM1 interface. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the Runtime Register Base Address Table.

TABLE 15-2: RUNTIME REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
ACPI PM1 Interface	0	LPC	I/O	Programmed BAR
	0	EC	32-bit internal address space	000F_1400h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

All the registers in Table 15-3, "Runtime Registers Summary" may be accessed by the Host and EC with the exception of the EC_PM_STS Register register which is EC-accessed only.

TABLE 15-3: RUNTIME REGISTERS SUMMARY

Offset	Register Name
00h	Power Management 1 Status 1 Register
01h	Power Management 1 Status 2 Register
02h	Power Management 1 Enable 1 Register
03h	Power Management 1 Enable 2 Register
04h	Power Management 1 Control 1 Register
05h	Power Management 1 Control 2 Register
06h	Power Management 2 Control 1 Register

TABLE 15-3: RUNTIME REGISTERS SUMMARY (CONTINUED)

Offset	Register Name
07h	Power Management 2 Control 2 Register
10h	EC_PM_STS Register

15.11.1 POWER MANAGEMENT 1 STATUS 1 REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Reserved	R	-	-

15.11.2 POWER MANAGEMENT 1 STATUS 2 REGISTER

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	WAK_STS This bit can be set or cleared by the EC. The Host writing a one to this bit can also clear this bit.	R/WC (See Note:)	00h	nSYSRST
6:4	Reserved	R	-	-
3	PWRBTNOR_STS This bit can be set or cleared by the EC to simulate a Power button override event status if the power is controlled by the EC. The Host writing a one to this bit can also clear this bit. The EC must generate the associated hardware event under software control.	R/WC (See Note:)	00h	nSYSRST
2	RTC_STS This bit can be set or cleared by the EC to simulate a RTC status. The Host writing a one to this bit can also clear this bit. The EC must generate the associated SCI interrupt under software control.	R/WC (See Note:)	00h	nSYSRST
1	SLPBTN_STS This bit can be set or cleared by the EC to simulate a Sleep button status if the sleep state is controlled by the EC. The Host writing a one to this bit can also clear this bit. The EC must generate the associated SCI interrupt under software control.	R/WC (See Note:)	00h	nSYSRST
0	PWRBTN_STS This bit can be set or cleared by the EC to simulate a Power button status if the power is controlled by the EC. The Host writing a one to this bit can also clear this bit. The EC must generate the associated SCI interrupt under software control.	R/WC (See Note:)	00h	nSYSRST

Note:	These bits are set/cleared by the EC directly i.e., writing ‘1’ sets the bit and writing ‘0’ clears it. These bits can also be cleared by the Host software writing a one to this bit position and by nSYSRST. Writing a 0 by the Host has no effect.

15.11.3 POWER MANAGEMENT 1 ENABLE 1 REGISTER

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Reserved	R	-	-

15.11.4 POWER MANAGEMENT 1 ENABLE 2 REGISTER

Offset	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:3	Reserved	R	-	-
2	RTC_EN This bit can be read or written by the Host. It can be read by the EC.	R/W (See Note:)	00h	nSYSRST
1	SLPBTN_EN This bit can be read or written by the Host. It can be read by the EC.	R/W (See Note:)	00h	nSYSRST
0	PWRBTN_EN This bit can be read or written by the Host. It can be read by the EC.	R/W (See Note:)	00h	nSYSRST

Note:	These bits are read-only by the EC.

15.11.5 POWER MANAGEMENT 1 CONTROL 1 REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Reserved	R	0h	nSYSRST

15.11.6 POWER MANAGEMENT 1 CONTROL 2 REGISTER

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	Reserved	R	-	-
5	SLP_EN See TABLE 15-4:.	See TABLE 15-4:.	00h	nSYSRST
4:2	SLP_TYP These bits can be set or cleared by the Host, read by the EC.	R/W (See Note:)	00h	nSYSRST
1	PWRBTNOR_EN This bit can be set or cleared by the Host, read by the EC.	R/W (See Note:)	00h	nSYSRST
0	Reserved	R	-	-

Note: These bits are read-only by the EC.

TABLE 15-4: SLP_EN DEFINITION

Host / EC	R/W	Description
Host	Read	Always reads 0
	Write	Writing a 0 has no effect, Writing a 1 sets this bit
EC	Read	Reads the value of the bit
	Write	Writing a 0 has no effect, Writing a 1 clears this bit

15.11.7 POWER MANAGEMENT 2 CONTROL 1 REGISTER

Offset	06h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Reserved	R	-	-

15.11.8 POWER MANAGEMENT 2 CONTROL 2 REGISTER

Offset	07h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Reserved	R	-	-

15.11.9 EC_PM_STS REGISTER

Offset				
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	UD	R/W	00h	nSYSRST
0	EC_SCI_STS If the EC_SCI_STS bit is “1”, an interrupt is generated on the EC_SCI# pin.	R/W	00h	nSYSRST

Note: These bits are only accessed by the EC. There is no host access to this register.

15.12 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the ACPI PM1 interface. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 15-5: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
ACPI PM1 Interface	0	EC	32-bit address space	000F_1500h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 15-6: EC-ONLY REGISTERS SUMMARY

Offset	Register Name
00h	Power Management 1 Status 1 Register
01h	Power Management 1 Status 2 Register
02h	Power Management 1 Enable 1 Register
03h	Power Management 1 Enable 2 Register
04h	Power Management 1 Control 1 Register
05h	Power Management 1 Control 2 Register
06h	Power Management 2 Control 1 Register
07h	Power Management 2 Control 2 Register
10h	EC_PM_STS Register

Note: The Power Management Status, Enable and Control registers in Table 15-6, "EC-Only Registers Summary" are described in Section 15.11, "Runtime Registers," on page 243.

15.12.1 EC_PM_STS REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	UD	R/W	00h	nSYSRST
0	EC_SCI_STS If the EC_SCI_STS bit is “1”, an interrupt is generated on the EC_SCI# pin.	R/W	00h	nSYSRST

Note: This register is only accessed by the EC. There is no host access to this register.

16.0 8042 EMULATED KEYBOARD CONTROLLER

16.1 Introduction

The MEC140x/1x keyboard controller uses the EC to produce a superset of the features provided by the industry-standard 8042 keyboard controller. The 8042 Emulated Keyboard Controller is a Host/EC Message Interface with hardware assists to emulate 8042 behavior and provide Legacy GATEA20 support.

Note:	There is no VCC emulation in hardware for this interface.

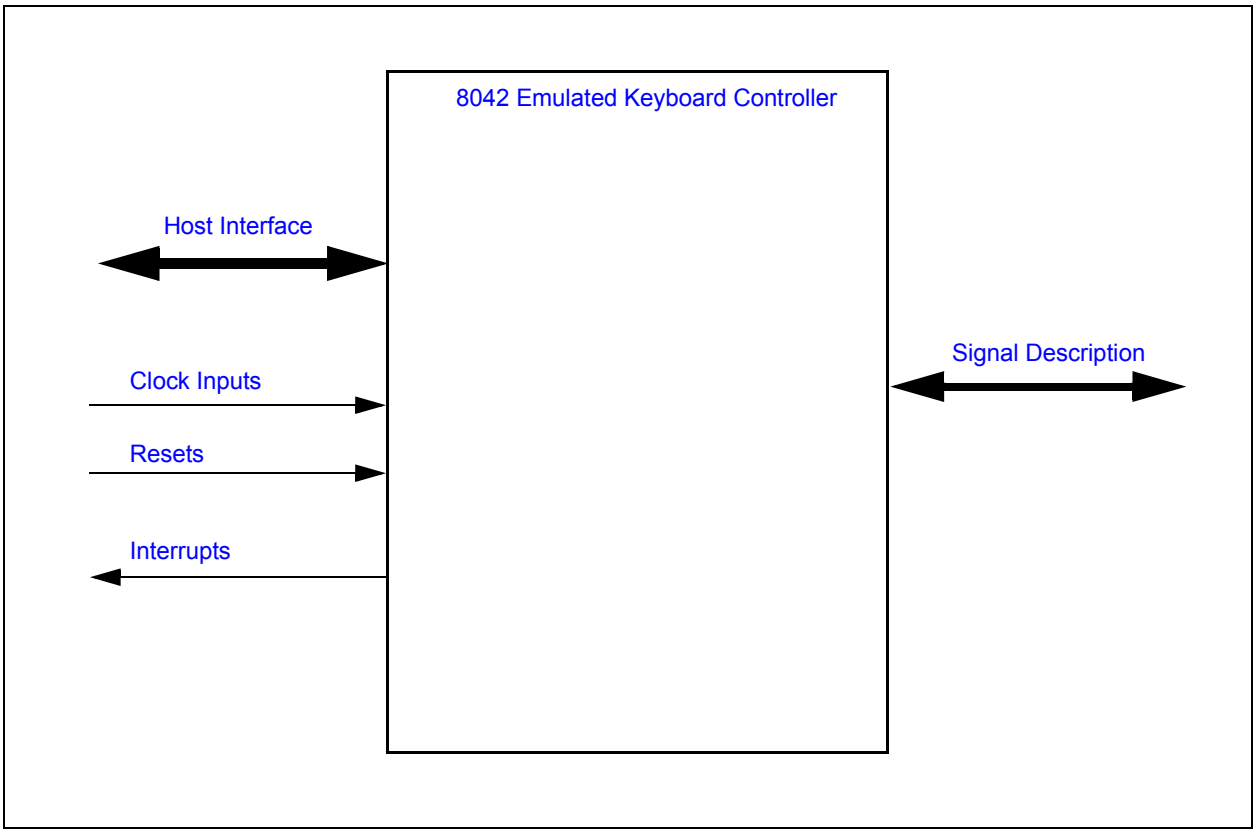
16.2 References

There are no references for this block.

16.3 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 16-1: I/O DIAGRAM OF BLOCK



16.4 Signal Description

TABLE 16-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
KBRST	Output	Keyboard Reset, routed to pin

16.5 Host Interface

The 8042 interface is accessed by host software via a registered interface, as defined in Section 16.13, "Configuration Registers" and Section 16.14, "Runtime Registers".

16.6 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

16.6.1 POWER DOMAINS

Name	Description
VTR	This Power Well is used to power the registers and logic in this block.

16.6.2 CLOCK INPUTS

Name	Description
1MHz	Clock used for the counter in the CPU_RESET circuitry.

16.6.3 RESETS

Name	Description
nSYSRST	This reset is asserted when VTR is applied.
VCC_PWRGD	This signal is asserted when the main power rail is asserted.
PCI_RESET#	This signal is asserted when LRESET# is asserted.
nSIO_RESET	This signal is asserted when VTR is low, PWRGD is low, or LRESET# is asserted.

16.7 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
KIRQ	This interrupt source for the SIRQ logic, representing a Keyboard interrupt, is generated when the PCOBF status bit is ‘1’.
MIRQ	This interrupt source for the SIRQ logic, representing a Mouse interrupt, is generated when the AUXOBF status bit is ‘1’.

Source	Description
IBF	Interrupt generated by the host writing either data or command to the data register
OBF	Interrupt generated by the host reading either data or aux data from the data register

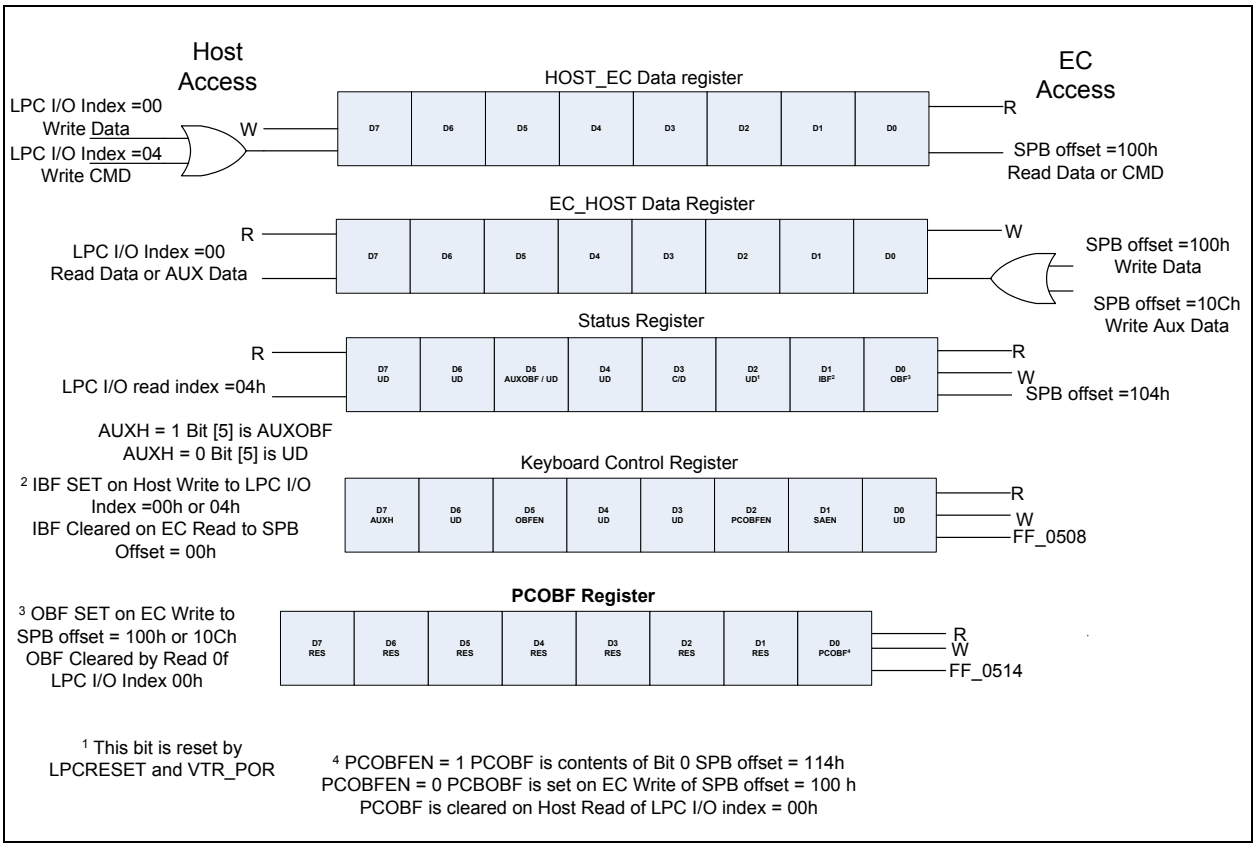
16.8 Low Power Modes

The 8042 Interface may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

16.9 Description

16.9.1 BLOCK DIAGRAM

FIGURE 16-2: BLOCK DIAGRAM OF 8042 Emulated Keyboard Controller



16.10 EC-to-Host Keyboard Communication

The EC can write to the EC_HOST Data / AUX Data Register by writing to the HOST2EC Data Register at EC-Only offset 0h or the EC AUX Data Register at EC-Only offset Ch. A write to either of these addresses automatically sets bit 0 (OBF) in the Status register. A write to the HOST2EC Data Register may also set PCOBF. A write to the EC AUX Data Register may also set AUXOBF.

16.10.1 PCOBF DESCRIPTION

If enabled by the bit OBFEN, the bit PCOBF is gated onto KIRQ. The KIRQ signal is a system interrupt which signifies that the EC has written to the EC2Host Data Register (EC-Only offset 0h). On power-up, PCOBF is reset to 0. PCOBF will normally reflect the status of writes to EC2Host Data Register, if PCOBFEN is "0". PCOBF is cleared by hardware on a HOST read of the EC_HOST Data / AUX Data Register.

KIRQ is normally selected as IRQ1 for keyboard support.

Additional flexibility has been added which allows firmware to directly control the PCOBF output signal, independent of data transfers to the host-interface data output register. This feature allows the MEC140x/1x to be operated via the host "polled" mode. Firmware control is active when PCOBFEN is '1'. Firmware sets PCOBF high by writing a "1" to the PCOBF field of the PCOBF Register. Firmware must also clear PCOBF by writing a "0" to the PCOBF field.

The PCOBF register is also readable; the value read back on bit 0 of the register always reflects the present value of the PCOBF output. If PCOBFEN = 1, then this value reflects the output of the firmware latch in the PCOBF Register. If PCOBFEN = 0, then the value read back reflects the in-process status of write cycles to the HOST2EC Data Register (i.e., if the value read back is high, the host interface output data register has just been written to). If OBFEN=0, then KIRQ is driven inactive (low).

16.10.2 AUXOBF DESCRIPTION

If enabled by the bit OBFEN, the bit AUXOBF is multiplexed onto MIRQ. The AUXOBF/MIRQ signal is a system interrupt which signifies that the EC has written to the EC_HOST Data / AUX Data Register. On power-up, after nSYSRST, AUXOBF is reset to 0. AUXOBF will normally reflects the status of writes to EC EC AUX Data Register (EC-Only offset Ch). AUXOBF is cleared by hardware on a read of the Host Data Register. If OBFEN=0, then MIRQ is driven inactive (low).

MIRQ is normally selected as IRQ12 for mouse support.

Firmware can also directly control the AUXOBF output signal, similar to the mechanism it can use to control PCOBF. Firmware control is active when AUXH is '0'. Firmware sets AUXOBF high by writing a "1" to the AUXOBF field of the EC Keyboard Status Register. Firmware must also clear AUXOBF by writing a "0" to the AUXOBF field.

TABLE 16-2: OBFEN AND PCOBFEN EFFECTS ON KIRQ

OBFEN	PCOBFEN	
0	X	KIRQ is inactive and driven low
1	0	KIRQ = PCOBF (status of writes to HOST2EC Data Register)
1	1	KIRQ = PCOBF (status of writes to PCOBF Register)

TABLE 16-3: OBFEN AND AUXH EFFECTS ON MIRQ

OBFEN	AUXH	
0	X	MIRQ is inactive and driven low
1	0	MIRQ = AUXOBF (status of writes to EC AUX Data Register)
1	1	MIRQ = AUXOBF (status of writes to AUXOBF in EC Keyboard Status Register)

16.11 Legacy Port92/GATEA20 Support

The MEC140x/1x supports LPC I/O writes to port HOST I/O address 92h as a quick alternate mechanism for generating a CPU_RESET pulse or controlling the state of GATEA20. The Port92/GateA20 logic has a separate Logical Device Number and Base Address register (see Section 16.16, "Legacy Port92/GATEA20 Configuration Registers" and Section 16.17, "Legacy Port92/GATEA20 Runtime Registers". The Base Address Register for the Port92/GateA20 Logical Device has only one writable bit, the Valid Bit, since the only I/O accessible Register has a fixed address.

The Port 92 Register resides at HOST I/O address 92h and is used to support the alternate reset (ALT_RST#) and alternate GATEA20 (ALT_A20) functions. This register defaults to 00h on assertion of nSIO_RESET.

Setting the Port92 Enable bit (Port 92 Enable Register) enables the Port92h Register. When Port92 is disabled, by clearing the Port92 Enable bit, then access to this register is completely disabled (I/O writes to host 92h are ignored and I/O reads float the system data bus SD[7:0]).

16.11.1 GATE A20 SPEEDUP

The MEC140x/1x contains on-chip logic support for the GATEA20 hardware speed-up feature. GATEA20 is part of the control required to mask address line A20 to emulate 8086 addressing.

In addition to the ability for the host to control the GATEA20 output signal directly, a configuration bit called SAEN in the Keyboard Control Register is provided; when set, SAEN allows firmware to control the GATEA20 output. When SAEN is set, a 1 bit register (GATEA20 Control Register) controls the GATEA20 output.

Host control and firmware control of GATEA20 affect two separate register elements. Read back of GATEA20 through the use of EC OFFSET 100h reflects the present state of the GATEA20 output signal: if SAEN is set, the value read back corresponds to the last firmware-initiated control of GATEA20; if SAEN is reset, the value read back corresponds to the last host-initiated control of GATEA20.

Host control of the GATEA20 output is provided by the hardware interpretation of the “GATEA20 sequence” (see Table 16-4, "GATEA20 Command/Data Sequence Examples"). The foregoing description assumes that the SAEN configuration bit is reset.

When the MEC140x/1x receives a “D1” command followed by data (via the host interface), the on-chip hardware copies the value of data bit 1 in the received data field to the GATEA20 host latch. At no time during this host-interface transaction will PCOBF or the IBF flag (bit 1) in the EC Keyboard Status Register be activated; for example, this host control of GATEA20 is transparent to firmware, with no consequent degradation of overall system performance. TABLE 16-4: details the possible GATEA20 sequences and the MEC140x/1x responses.

An additional level of control flexibility is offered via a memory-mapped synchronous set and reset capability. Any data written to the SETGA20L Register causes the GATEA20 host latch to be set; any data written to the RSTGA20L Register causes it to be reset. This control mechanism should be used with caution. It was added to augment the “normal” control flow as described above, not to replace it. Since the host and the firmware have asynchronous control capability of the host latch via this mechanism, a potential conflict could arise. Therefore, after using the SETGA20L and RSTGA20L registers, firmware should read back the GATEA20 status via the GATEA20 Control Register (with SAEN = 0) to confirm the actual GATEA20 response.

TABLE 16-4: GATEA20 COMMAND/DATA SEQUENCE EXAMPLES

Command(C) / Data (D)	R/W	D[7:0]	IBF Flag	GATEA20	Comments
C D C	W W W	D1 DF FF	0 0 0	Q 1 1	GATEA20 Turn-on Sequence
C D C	W W W	D1 DD FF	0 0 0	Q 0 0	GATEA20 Turn-off Sequence
C C D C	W W W W	D1 D1 DF FF	0 0 0 0	Q Q 1 1	GATEA20 Turn-on Sequence(*)
C C D C	W W W W	D1 D1 DD FF	0 0 0 0	Q Q 0 0	GATEA20 Turn-off Sequence(*)
C C C	W W W	D1 XX** FF	0 1 1	Q Q Q	Invalid Sequence

Note: The following notes apply:

- All examples assume that the SAEN configuration bit is 0.
- “Q” indicates the bit remains set at the previous state.
- *Not a standard sequence.
- **XX = Anything except D1.
- If multiple data bytes, set IBF and wait at state 0. Let the software know something unusual happened.
- For data bytes, only D[1] is used; all other bits are don't care.
- Host Commands (FF, FE, & D1) do not cause IBF. The method of blocking IBF in FIGURE 16-4: is the nIOW not being asserted when FF, FE, & D1 Host commands are written”.

The hardware GATEA20 state machine returns to state S1 from state S2 when CMD = D1, as shown in the following figures:.

FIGURE 16-3: GATEA20 STATE MACHINE

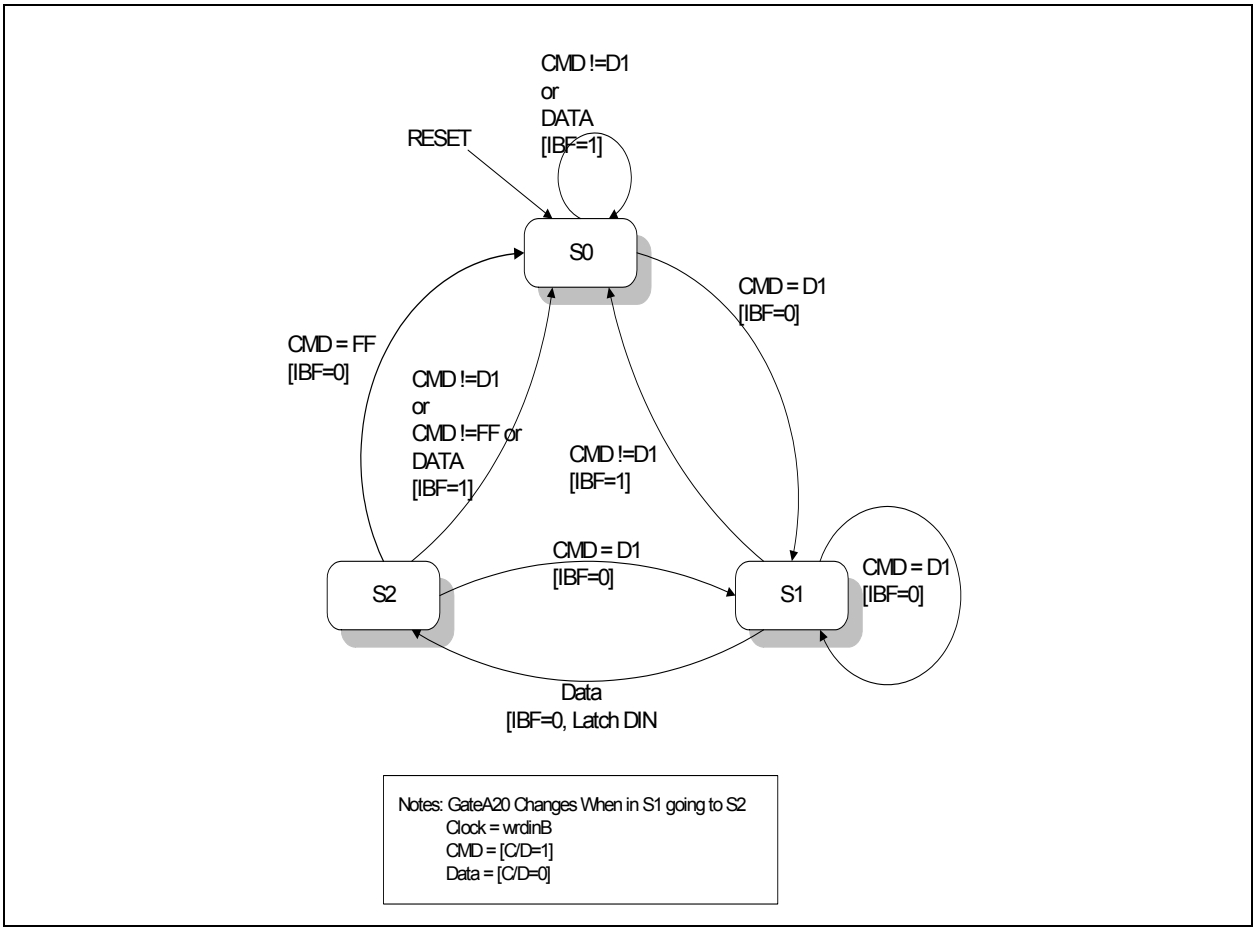
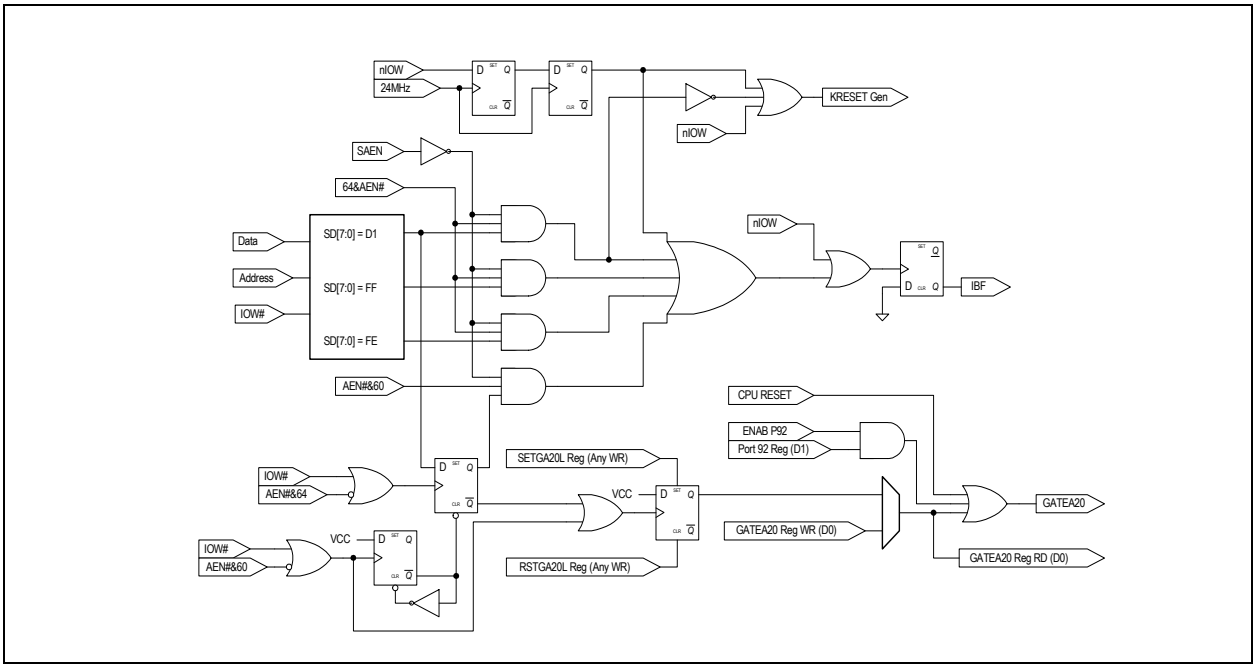


FIGURE 16-4: GATEA20 IMPLEMENTATION DIAGRAM



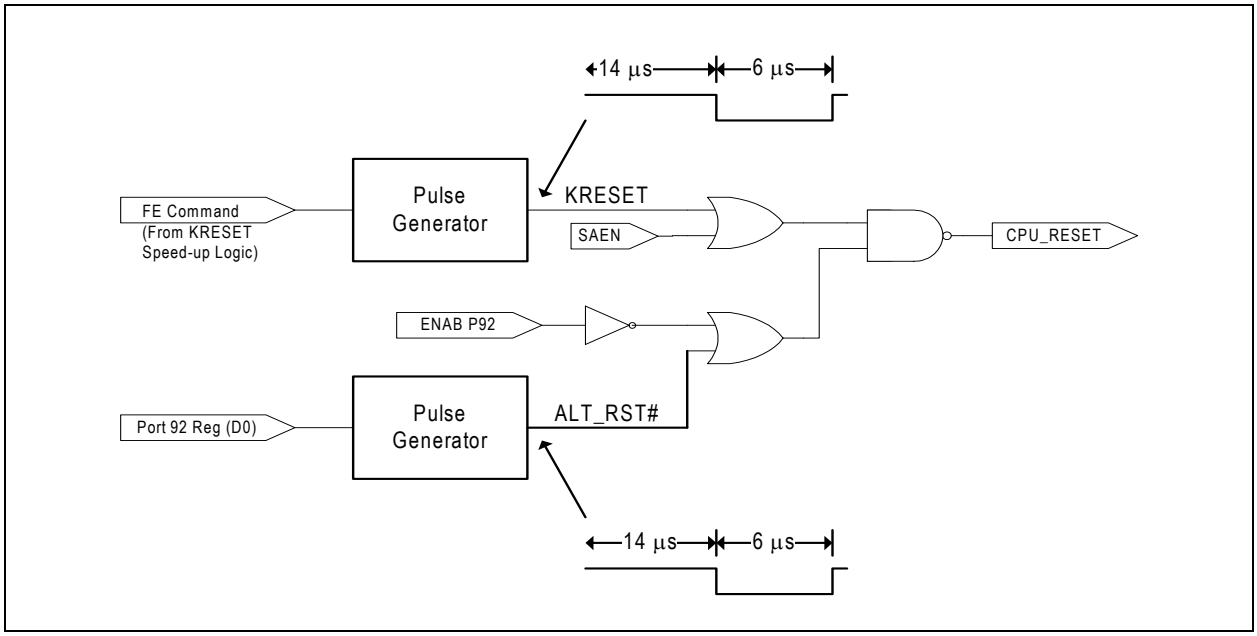
16.11.2 CPU_RESET HARDWARE SPEED-UP

The ALT_CPU_RESET bit generates, under program control, the ALT_RST# signal, which provides an alternate, means to drive the MEC140x/1x CPU_RESET pin which in turn is used to reset the Host CPU. The ALT_RST# signal is internally NANDed together with the KBDRESET# pulse from the KRESET Speed up logic to provide an alternate software means of resetting the host CPU.

Before another ALT_RST# pulse can be generated, ALT_CPU_RESET must be cleared to '0' either by an nSIO_RESET or by a write to the Port 92 Register with bit 0 = '0'. An ALT_RST# pulse is not generated in the event that the ALT_CPU_RESET bit is cleared and set before the prior ALT_RESET# pulse has completed.

If the 8042EM Sleep Enable is asserted, or the 8042 EM ACTIVATE bit is de-asserted, the 1MHz clocks source is disabled.

FIGURE 16-5: CPU_RESET IMPLEMENTATION DIAGRAM



16.12 Instance Description

There are two blocks defined in this chapter: Emulated 8042 Interface and the Legacy Port92/GATEA20 Support. The MEC140x/1x has one instance of each block.

16.13 Configuration Registers

The registers listed in the Configuration Register Summary table are for a single instance of the Emulated 8042 Interface. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the Configuration Register Base Address Table.

TABLE 16-5: CONFIGURATION REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Logical Device Number	Host	Address Space	Base Address
Emulated 8042 Interface	0	1	LPC	Configuration Port	INDEX = 00h
			EC	32-bit internal address space	000F_0700h

Each Configuration register access through the Host Access Port is via its LDN and its Host Access Port Index. EC access is a relative offset to the EC Base Address.

TABLE 16-6: CONFIGURATION REGISTER SUMMARY

Offset	Register Name (Mnemonic)
30h	Activate Register

16.13.1 ACTIVATE REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	R	-	-
0	ACTIVATE 1=The 8042 Interface is powered and functional. 0=The 8042 Interface is powered down and inactive.	R/W	0b	VCC_PWRGD and nSYSRST

16.14 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the Emulated 8042 Interface. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the Runtime Register Base Address Table.

TABLE 16-7: RUNTIME REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
Emulated 8042 Interface	0	LPC	I/O	Programmed BAR
		EC	32-bit address space	000F_0400h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 16-8: RUNTIME REGISTER SUMMARY

Offset	Register Name (Mnemonic)
0h/04h	HOST_EC Data / CMD Register
0h	EC_HOST Data / AUX Data Register
4h	Keyboard Status Read Register

16.14.1 HOST_EC DATA / CMD REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	WRITE_DATA This 8-bit register is write-only. When written, the C/D bit in the Keyboard Status Read Register is cleared to ‘0’, signifying data, and the IBF in the same register is set to ‘1’. When the Runtime Register at offset 0h is read by the Host, it functions as the EC_HOST Data / AUX Data Register.	W	0h	nSYSRST

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	WRITE_CMD This 8-bit register is write-only and is an alias of the register at offset 0h. When written, the C/D bit in the Keyboard Status Read Register is set to '1', signifying a command, and the IBF in the same register is set to '1'. When the Runtime Register at offset 4h is read by the Host, it functions as the Keyboard Status Read Register.	W	0h	nSYSRST

16.14.2 EC_HOST DATA / AUX DATA REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	READ_DATA This 8-bit register is read-only. When read by the Host, the PCOBF and/or AUXOBF interrupts are cleared and the OBF flag in the status register is cleared.	R	0h	nSYSRST

16.14.3 KEYBOARD STATUS READ REGISTER

This register is a read-only alias of the EC Keyboard Status Register.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	UD2 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R	0h	nSYSRST
5	AUXOBF Auxiliary Output Buffer Full. This bit is set to "1" whenever the EC writes the EC AUX Data Register. This flag is reset to "0" whenever the EC writes the EC2Host Data Register.	R	0h	nSYSRST
4	UD1 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R	0h	nSYSRST

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	C/D Command Data. This bit specifies whether the input data register contains data or a command (“0” = data, “1” = command). During a Host command write operation (when the Host writes the HOST_EC Data / CMD Register at offset 04h), this bit is set to “1”. During a Host data write operation (when the Host writes the HOST_EC Data / CMD Register at offset 0h), this bit is set to “0”.	R	0h	nSYSRST
2	UD0 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias. **Note:** This bit is reset to ‘0’ when the LRESET# pin signal is asserted.	R	0h	nSYSRST and PCI_RESET#
1	IBF Input Buffer Full. This bit is set to “1” whenever the Host writes data or a command into the HOST_EC Data / CMD Register. When this bit is set, the EC's IBF interrupt is asserted, if enabled. When the EC reads the HOST2EC Data Register, this bit is automatically reset and the interrupt is cleared. **Note:** This bit is not reset when VCC_PWRGD is asserted or when the LPC interface powers down. To clear this bit, firmware must read the HOST2EC Data Register in the EC-Only address space.	R	0h	nSYSRST
0	OBF Output Buffer Full. This bit is set when the EC writes a byte of Data or AUX Data into the EC_HOST Data / AUX Data Register. When the Host reads the HOST_EC Data / CMD Register, this bit is automatically cleared by hardware and a OBF interrupt is generated. **Note:** This bit is not reset when VCC_PWRGD is asserted or when the LPC interface powers down. To clear this bit, firmware must read the HOST_EC Data / CMD Register in the Runtime address space.	R	0h	nSYSRST

16.15 Emulated 8042 Interface EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Emulated 8042 Interface. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 16-9: EC-ONLY REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
Emulated 8042 Interface	0	EC	32-bit address space	000F_0500h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 16-10: EC-ONLY REGISTER SUMMARY

Offset	Register Name (Mnemonic)
0h	HOST2EC Data Register
0h	EC2Host Data Register
4h	EC Keyboard Status Register
8h	Keyboard Control Register
Ch	EC AUX Data Register
14h	PCOBF Register

16.15.1 HOST2EC DATA REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	HOST2EC_DATA This register is an alias of the HOST_EC Data / CMD Register. When read at the EC-Only offset of 0h, it returns the data written by the Host to either Runtime Register offset 0h or Runtime Register offset 04h.	R	0h	nSYSRST

16.15.2 EC2HOST DATA REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC2HOST_DATA This register is an alias of the EC_HOST Data / AUX Data Register. Writing this register sets the OBF status bit.	W	0h	nSYSRST

16.15.3 EC KEYBOARD STATUS REGISTER

This register is an alias of the Keyboard Status Read Register. The fields C/D, IBF, and OBF remain read-only.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	UD2 User-defined data. Readable and writable by the EC.	R/W	0h	nSYSRST
5	AUXOBF Auxiliary Output Buffer Full. This bit is set to ‘1’ whenever the EC writes the EC AUX Data Register. This flag is reset to ‘0’ whenever the EC writes the EC2Host Data Register.	R/W	0h	nSYSRST

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
4	UD1 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R/W	0h	nSYSRST
3	C/D Command Data. This bit specifies whether the input data register contains data or a command. During a Host command write operation (when the Host writes the HOST_EC Data / CMD Register at offset 04h), this bit is set to '1'. During a Host data write operation (when the Host writes the HOST_EC Data / CMD Register at offset 0h), this bit is set to '0'. 1=Command 0=Data	R	0h	nSYSRST
2	UD0 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias. This bit is reset to '0' when the LRESET# pin signal is asserted.	R/W	0h	nSYSRST and PCI_RESET#
1	IBF Input Buffer Full. This bit is set to "1" whenever the Host writes data or a command into the HOST_EC Data / CMD Register. When this bit is set, the EC's IBF interrupt is asserted, if enabled. When the EC reads the HOST2EC Data Register this bit is automatically reset and the interrupt is cleared. This bit is not reset when VCC_PWRGD is asserted or when the LPC interface powers down. To clear this bit, firmware must read the HOST2EC Data Register in the EC-Only address space.	R	0h	nSYSRST
0	OBF Output Buffer Full. This bit is set when the EC writes a byte of Data or AUX Data into the EC_HOST Data / AUX Data Register. When the Host reads the HOST_EC Data / CMD Register, this bit is automatically cleared by hardware and a OBF interrupt is generated. This bit is not reset when VCC_PWRGD is asserted or when the LPC interface powers down. To clear this bit, firmware must read the HOST_EC Data / CMD Register in the Runtime address space.	R	0h	nSYSRST

16.15.4 KEYBOARD CONTROL REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	AUXH AUX in Hardware. 1=AUXOBF of the Keyboard Status Read Register is set in hardware by a write to the EC AUX Data Register 0=AUXOBF is not modified in hardware, but can be read and written by the EC using the EC-Only alias of the EC Keyboard Status Register	R/W	0h	nSYSRST
6	UD5 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R/W	0h	nSYSRST
5	OBFEN When this bit is '1', the system interrupt signal KIRQ is driven by the bit PCOBF and MIRQ is driven by AUXOBF. When this bit is '0', KIRQ and MIRQ are driven low. This bit must not be changed when OBF of the status register is equal to '1'.	R/W	0h	nSYSRST
4:3	UD4 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R/W	0h	nSYSRST
2	PCOBFEN 1= reflects the value written to the PCOBF Register 0=PCOBF reflects the status of writes to the EC2Host Data Register	R/W	0h	nSYSRST
1	SAEN Software-assist enable. 1=This bit allows control of the GATEA20 signal via firmware 0=GATEA20 corresponds to either the last Host-initiated control of GATEA20 or the firmware write to the Keyboard Control Register or the EC AUX Data Register.	R/W	0h	nSYSRST
0	UD3 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R/W	0h	nSYSRST

16.15.5 EC AUX DATA REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_AUX_DATA This 8-bit register is write-only. When written, the C/D in the Keyboard Status Read Register is cleared to '0', signifying data, and the IBF in the same register is set to '1'. When the Runtime Register at offset 0h is read by the Host, it functions as the EC_HOST Data / AUX Data Register.	W	0h	nSYSRST

16.15.6 PCOBF REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	R	-	-
0	PCOBF For a description of this bit, see Section 16.10.1, "PCOBF Description".	R/W	0h	nSYSRST

16.16 Legacy Port92/GATEA20 Configuration Registers

The registers listed in the Configuration Register Summary table are for a single instance of the Legacy Port92/GATEA20 logic. The addresses of each register listed in this table are defined as a relative offset to the host "Base Address" defined in the Configuration Register Base Address Table.

TABLE 16-11: CONFIGURATION BASE ADDRESS

Block Instance	Instance Number	Logical Device Number	Host	Address Space	Base Address
Port92-Legacy	0	1	LPC	Configuration Port	INDEX = 00h
			EC	32-bit internal address space	000F_1800h

Each Configuration register access through the Host Access Port is via its LDN and its Host Access Port Index. EC access is a relative offset to the EC Base Address.

TABLE 16-12: CONFIGURATION REGISTER SUMMARY

Offset	Register Name (Mnemonic)
30h	Port 92 Enable Register

16.16.1 PORT 92 ENABLE REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	R	-	-
0	P92_EN When this bit is '1', the Port92h Register is enabled. When this bit is '0', the Port92h Register is disabled, and Host writes to LPC address 92h are ignored.	R/W	0h	VCC_PWRGD and nSYSRST

16.17 Legacy Port92/GATEA20 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the Legacy Port92/GATEA20 logic. The addresses of each register listed in this table are defined as a relative offset to the host "Base Address" defined in the Runtime Register Base Address Table.

TABLE 16-13: RUNTIME REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
Port92-Legacy	0	LPC	I/O	0092h
		EC	32-bit address space	000F_1800h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 16-14: RUNTIME REGISTER SUMMARY

Offset	Register Name (Mnemonic)
0h	Port 92 Register

16.17.1 PORT 92 REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:2	Reserved	R	-	-
1	ALT_GATE_A20 This bit provides an alternate means for system control of the GATEA20 pin. ALT_A20 low drives GATEA20 low, if A20 from the keyboard controller is also low. When Port 92 is enabled, writing a 1 to this bit forces ALT_A20 high. ALT_A20 high drives GATEA20 high regardless of the state of A20 from the keyboard controller. 0=ALT_A20 is driven low 1=ALT_A20 is driven high	R/W	0h	nSIO_ RESET
0	ALT_CPU_RESET This bit provides an alternate means to generate a CPU_RESET pulse. The CPU_RESET output provides a means to reset the system CPU to effect a mode switch from Protected Virtual Address Mode to the Real Address Mode. This provides a faster means of reset than is provided through the EC keyboard controller. Writing a “1” to this bit will cause the ALT_RST# internal signal to pulse (active low) for a minimum of 6µs after a delay of 14µs. Before another ALT_RST# pulse can be generated, this bit must be written back to “0”.	R/W	0h	nSIO_ RESET

16.18 Emulated 8042 Interface EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Legacy Port92/GATEA20 logic. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 16-15: EC-ONLY REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
Port92-Legacy	0	EC	32-bit address space	000F_1900h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 16-16: EC-ONLY REGISTER SUMMARY

Offset	Register Name (Mnemonic)
0h	GATEA20 Control Register
8h	SETGA20L Register
Ch	RSTGA20L Register

16.18.1 GATEA20 CONTROL REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	R	-	-
0	GATEA20 0=The GATEA20 output is driven low 1=The GATEA20 output is driven high	R/W	0h	nSYSRST

16.18.2 SETGA20L REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	SETGA20L See Section 16.11.1, "GATE A20 Speedup" for information on this register. A write to this register sets GATEA20 in the GATEA20 Control Register.	W	-	-

16.18.3 RSTGA20L REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	RSTGA20L See Section 16.11.1, "GATE A20 Speedup" for information on this register. A write to this register sets GATEA20 in the GATEA20 Control Register.	W	-	-

17.0 UART

17.1 Introduction

The 16550 UART (Universal Asynchronous Receiver/Transmitter) is a full-function Two Pin Serial Port that supports the standard RS-232 Interface.

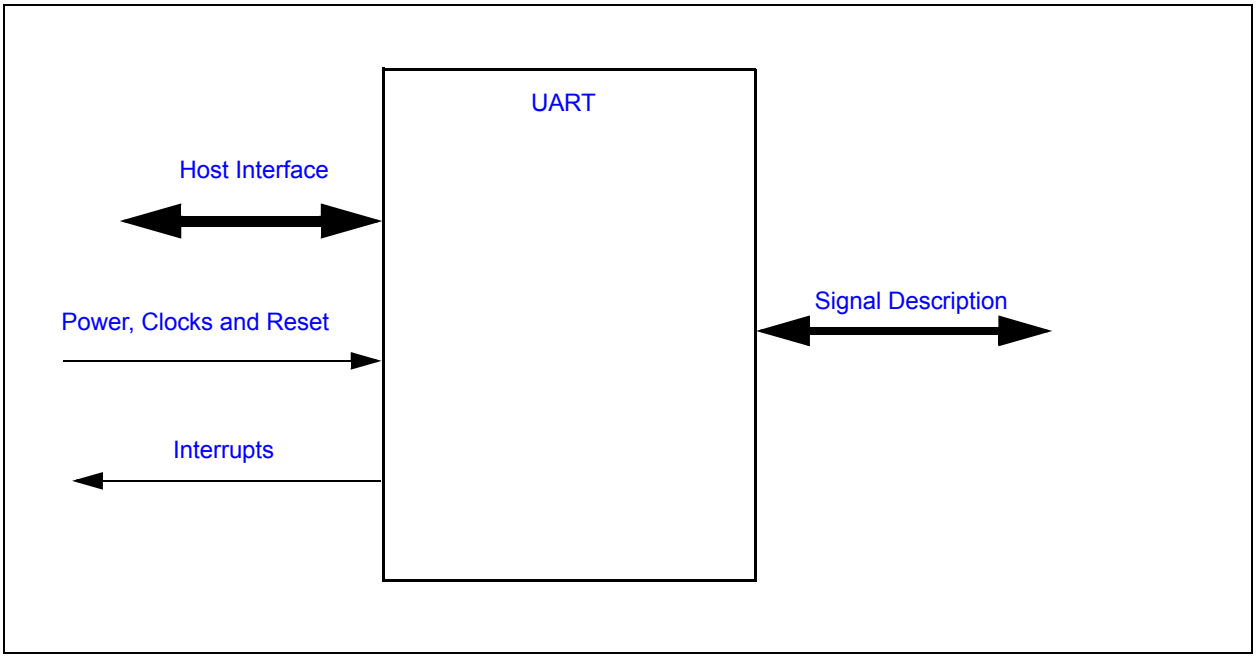
17.2 References

- EIA Standard RS-232-C specification

17.3 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 17-1: I/O DIAGRAM OF BLOCK



17.4 Signal Description

TABLE 17-1: SIGNAL DESCRIPTION

Name	Direction	Description
DTR#	Output	Active low Data Terminal ready output for the Serial Port. Handshake output signal notifies modem that the UART is ready to transmit data. This signal can be programmed by writing to bit 1 of the Modem Control Register (MCR). **Note:** Defaults to tri-state on V3_DUAL power on.

TABLE 17-1: SIGNAL DESCRIPTION (CONTINUED)

Name	Direction	Description
DCD#	Output	Active low Data Carrier Detect input for the serial port. Handshake signal which notifies the UART that carrier signal is detected by the modem. The CPU can monitor the status of DCD# signal by reading bit 7 of Modem Status Register (MSR). A DCD# signal state change from low to high after the last MSR read will set MSR bit 3 to a 1. If bit 3 of Interrupt Enable Register is set, the interrupt is generated when DCD # changes state.
DSR#	Input	Active low Data Set Ready input for the serial port. Handshake signal which notifies the UART that the modem is ready to establish the communication link. The CPU can monitor the status of DSR# signal by reading bit 5 of Modem Status Register (MSR). A DSR# signal state change from low to high after the last MSR read will set MSR bit 1 to a 1. If bit 3 of Interrupt Enable Register is set, the interrupt is generated when DSR# changes state.
RI#	Input	Active low Ring Indicator input for the serial port. Handshake signal which notifies the UART that the telephone ring signal is detected by the modem. The CPU can monitor the status of RI# signal by reading bit 6 of Modem Status Register (MSR). A RI# signal state change from low to high after the last MSR read will set MSR bit 2 to a 1. If bit 3 of Interrupt Enable Register is set, the interrupt is generated when nRI changes state.
RTS#	Output	Active low Request to Send output for the Serial Port. Handshake output signal notifies modem that the UART is ready to transmit data. This signal can be programmed by writing to bit 1 of the Modem Control Register (MCR). The hardware reset will reset the RTS# signal to inactive mode (high). RTS# is forced inactive during loop mode operation. Defaults to tri-state on V3_DUAL power on.
CTS#	Input	Active low Clear to Send input for the serial port. Handshake signal which notifies the UART that the modem is ready to receive data. The CPU can monitor the status of CTS# signal by reading bit 4 of Modem Status Register (MSR). A CTS# signal state change from low to high after the last MSR read will set MSR bit 0 to a 1. If bit 3 of the Interrupt Enable Register is set, the interrupt is generated when CTS# changes state. The CTS# signal has no effect on the transmitter.
TXD	Output	Transmit serial data output.
RXD	Input	Receiver serial data input.
UART_CLK	Input	External Baud Clock Generator input. The source of the baud clock is controlled by CLK_SRC on page 272.

17.5 Host Interface

The UART is accessed by host software via a registered interface, as defined in Section 17.10, "Configuration Registers"and Section 17.11, "Runtime Registers".

17.6 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

17.6.1 POWER DOMAINS

Name	Description
VTR	This Power Well is used to power the registers and logic in this block.

17.6.2 CLOCK INPUTS

Name	Description
1.8432MHz_Clk	The UART requires a 1.8432 MHz ± 2% clock input for baud rate generation.
24MHz_Clk	24 MHz ± 2% clock input. This clock may be enabled to generate the baud rate, which requires a 1.8432 MHz ± 2% clock input.

17.6.3 RESETS

Name	Description
nSYSRST	This reset is asserted when VTR is applied.
nSIO_RESET	This is an alternate reset condition, typically asserted when the main power rail is asserted.
RESET	This reset is determined by the POWER bit signal. When the power bit signal is 1, this signal is equal to nSIO_RESET. When the power bit signal is 0, this signal is equal to nSYSRST.

17.7 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
UART	The UART interrupt event output indicates if an interrupt is pending. See Table 17-8, “Interrupt Control,” on page 278.

Source	Description
UART	The UART interrupt event output indicates if an interrupt is pending. See Table 17-8, “Interrupt Control,” on page 278.

17.8 Low Power Modes

The UART may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

17.9 Description

The UART is compatible with the 16450, the 16450 ACE registers and the 16C550A. The UART performs serial-to-parallel conversions on received characters and parallel-to-serial conversions on transmit characters. Two sets of baud rates are provided. When the 1.8432 MHz source clock is selected, standard baud rates from 50 to 115.2K are available. When the source clock is 32.26 MHz, baud rates from 126K to 2,016K are available. The character options are programmable for 1 start; 1, 1.5 or 2 stop bits; even, odd, sticky or no parity; and prioritized interrupts. The UART contains a programmable baud rate generator that is capable of dividing the input clock signal by 1 to 65535. The UART is also capable of supporting the MIDI data rate. Refer to the Configuration Registers for information on disabling, powering down and changing the base address of the UART. The UART interrupt is enabled by programming OUT2 of the UART to logic "1." Because OUT2 is logic "0," it disables the UART's interrupt. The UART is accessible by both the Host and the EC.

17.9.1 PROGRAMMABLE BAUD RATE

The Serial Port contains a programmable Baud Rate Generator that is capable of dividing the internal clock source by any divisor from 1 to 65535. The clock source is either the 1.8432MHz_Clk clock source or the 24MHz_Clk clock source. The output frequency of the Baud Rate Generator is 16x the Baud rate. Two eight bit latches store the divisor in 16 bit binary format. These Divisor Latches must be loaded during initialization in order to ensure desired operation of the Baud Rate Generator. Upon loading either of the Divisor Latches, a 16 bit Baud counter is immediately loaded. This prevents long counts on initial load. If a 0 is loaded into the BRG registers, the output divides the clock by the number 3. If a 1 is loaded, the output is the inverse of the input oscillator. If a two is loaded, the output is a divide by 2 signal with a 50% duty cycle. If a 3 or greater is loaded, the output is low for 2 bits and high for the remainder of the count.

The following tables show possible baud rates.

TABLE 17-2: UART BAUD RATES USING CLOCK SOURCE 1.8432MHz_Clk

Desired Baud Rate	BAUD_CLOCK_SEL	Divisor Used to Generate 16X Clock
50	0	2304
75	0	1536
110	0	1047
134.5	0	857
150	0	768
300	0	384
600	0	192
1200	0	96
1800	0	64
2000	0	58
2400	0	48
3600	0	32
4800	0	24
7200	0	16
9600	0	12

TABLE 17-2: UART BAUD RATES USING CLOCK SOURCE 1.8432MHz_Clk (CONTINUED)

Desired Baud Rate	BAUD_CLOCK_SEL	Divisor Used to Generate 16X Clock
19200	0	6
38400	0	3
57600	0	2
115200	0	1

TABLE 17-3: UART BAUD RATES USING CLOCK SOURCE 24MHz_Clk

Desired Baud Rate	BAUD_CLOCK_SEL	Divisor Used to Generate 16X Clock
125000	1	12
136400	1	11
150000	1	10
166700	1	9
187500	1	8
214300	1	7
250000	1	6
300000	1	5
375000	1	4
500000	1	3
750000	1	2
1500000	1	1

17.10 Configuration Registers

The registers listed in the Configuration Register Summary table are for a single instance of the UART. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the Configuration Register Base Address Table.

FIGURE 17-2: CONFIGURATION REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
UART	0	LPC	Configuration Port	INDEX = 00h
UART	0	EC	32-bit internal address space	000F_1F00h

Each Configuration register access through the Host Access Port is via its LDN and its Host Access Port Index. EC access is a relative offset to the EC Base Address.

TABLE 17-4: CONFIGURATION REGISTER SUMMARY

Offset	Register Name (Mnemonic)
30h	Activate Register
F0h	Configuration Select Register

17.10.1 ACTIVATE REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	R	-	-
0	ACTIVATE When this bit is 1, the UART logical device is powered and functional. When this bit is 0, the UART logical device is powered down and inactive.	R/W	0b	RESET

17.10.2 CONFIGURATION SELECT REGISTER

Offset	F0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:3	Reserved	R	-	-
2	POLARITY 1=The UART_TX and UART_RX pins functions are inverted 0=The UART_TX and UART_RX pins functions are not inverted	R/W	0b	RESET
1	POWER 1=The RESET reset signal is derived from nSIO_RESET 0=The RESET reset signal is derived from nSYSRST	R/W	1b	RESET
0	CLK_SRC 1=The UART Baud Clock is derived from an external clock source 0=The UART Baud Clock is derived from one of the two internal clock sources	R/W	0b	RESET

17.11 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the UART. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in Runtime Register Base Address Table.

TABLE 17-5: RUNTIME REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
UART	0	LPC	I/O	Programmed BAR
		EC	32-bit internal address space	000F_1C00h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 17-6: RUNTIME REGISTER SUMMARY

DLAB Note 1:	Offset	Register Name (Mnemonic)
0	0h	Receive Buffer Register
0	0h	Transmit Buffer Register
1	0h	Programmable Baud Rate Generator LSB Register
1	1h	Programmable Baud Rate Generator MSB Register
0	1h	Interrupt Enable Register
x	02h	FIFO Control Register
x	02h	Interrupt Identification Register
x	03h	Line Control Register
x	04h	Modem Control Register
x	05h	Line Status Register
x	06h	Modem Status Register
x	07h	Scratchpad Register

Note 1: DLAB is bit 7 of the Line Control Register.

17.11.1 RECEIVE BUFFER REGISTER

Offset	0h (DLAB=0)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	RECEIVED_DATA This register holds the received incoming data byte. Bit 0 is the least significant bit, which is transmitted and received first. Received data is double buffered; this uses an additional shift register to receive the serial data stream and convert it to a parallel 8 bit word which is transferred to the Receive Buffer register. The shift register is not accessible.	R	0h	RESET

17.11.2 TRANSMIT BUFFER REGISTER

Offset	0h (DLAB=0)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	TRANSMIT_DATA This register contains the data byte to be transmitted. The transmit buffer is double buffered, utilizing an additional shift register (not accessible) to convert the 8 bit data word to a serial format. This shift register is loaded from the Transmit Buffer when the transmission of the previous byte is complete.	W	0h	RESET

17.11.3 PROGRAMMABLE BAUD RATE GENERATOR LSB REGISTER

Offset	00h (DLAB=1)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	BAUD_RATE_DIVISOR_LSB See Section 17.9.1, "Programmable Baud Rate".	R/W	0h	RESET

17.11.4 PROGRAMMABLE BAUD RATE GENERATOR MSB REGISTER

Offset	01h (DLAB=1)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	BAUD_CLK_SEL 0=If CLK_SRC is '0', the baud clock is derived from the 1.8432MHz_Clk. If CLK_SRC is '1', this bit has no effect 1=If CLK_SRC is '0', the baud clock is derived from the 24MHz_Clk. If CLK_SRC is '1', this bit has no effect	R/W	0h	RESET
6:0	BAUD_RATE_DIVISOR_MSB See Section 17.9.1, "Programmable Baud Rate".	R/W	0h	RESET

17.11.5 INTERRUPT ENABLE REGISTER

The lower four bits of this register control the enables of the five interrupt sources of the Serial Port interrupt. It is possible to totally disable the interrupt system by resetting bits 0 through 3 of this register. Similarly, setting the appropriate bits of this register to a high, selected interrupts can be enabled. Disabling the interrupt system inhibits the Interrupt Identification Register and disables any Serial Port interrupt out of the MEC140x/1x. All other system functions operate in their normal manner, including the Line Status and MODEM Status Registers. The contents of the Interrupt Enable Register are described below.

Offset	01h (DLAB=0)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:4	Reserved	R	-	-
3	EMSI This bit enables the MODEM Status Interrupt when set to logic "1". This is caused when one of the Modem Status Register bits changes state.	R/W	0h	RESET
2	ELSI This bit enables the Received Line Status Interrupt when set to logic "1". The error sources causing the interrupt are Overrun, Parity, Framing and Break. The Line Status Register must be read to determine the source.	R/W	0h	RESET
1	ETHREI This bit enables the Transmitter Holding Register Empty Interrupt when set to logic "1".	R/W	0h	RESET
0	ERDAI This bit enables the Received Data Available Interrupt (and timeout interrupts in the FIFO mode) when set to logic "1".	R/W	0h	RESET

17.11.6 FIFO CONTROL REGISTER

This is a write only register at the same location as the Interrupt Identification Register.

Note: DMA is not supported.

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	RECV_FIFO_TRIGGER_LEVEL These bits are used to set the trigger level for the RCVR FIFO interrupt.	W	0h	RESET
5:4	Reserved	R	-	-
3	DMA_MODE_SELECT Writing to this bit has no effect on the operation of the UART. The RXRDY and TXRDY pins are not available on this chip.	W	0h	RESET
2	CLEAR_XMIT_FIFO Setting this bit to a logic “1” clears all bytes in the XMIT FIFO and resets its counter logic to “0”. The shift register is not cleared. This bit is self-clearing.	W	0h	RESET
1	CLEAR_RECv_FIFO Setting this bit to a logic “1” clears all bytes in the RCVR FIFO and resets its counter logic to “0”. The shift register is not cleared. This bit is self-clearing.	W	0h	RESET
0	EXRF Enable XMIT and RECV FIFO. Setting this bit to a logic “1” enables both the XMIT and RCVR FIFOs. Clearing this bit to a logic “0” disables both the XMIT and RCVR FIFOs and clears all bytes from both FIFOs. When changing from FIFO Mode to non-FIFO (16450) mode, data is automatically cleared from the FIFOs. This bit must be a 1 when other bits in this register are written to or they will not be properly programmed.	W	0h	RESET

TABLE 17-7: RECV FIFO TRIGGER LEVELS

Bit 7	Bit 6	RECV FIFO Trigger Level (BYTES)
0	0	1
	1	4
1	0	8
	1	14

17.11.7 INTERRUPT IDENTIFICATION REGISTER

By accessing this register, the host CPU can determine the highest priority interrupt and its source. Four levels of priority interrupt exist. They are in descending order of priority:

1. Receiver Line Status (highest priority)
2. Received Data Ready
3. Transmitter Holding Register Empty
4. MODEM Status (lowest priority)

Information indicating that a prioritized interrupt is pending and the source of that interrupt is stored in the Interrupt Identification Register (refer to TABLE 17-8:). When the CPU accesses the IIR, the Serial Port freezes all interrupts and indicates the highest priority pending interrupt to the CPU. During this CPU access, even if the Serial Port records new interrupts, the current indication does not change until access is completed. The contents of the IIR are described below.

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	FIFO_EN These two bits are set when the FIFO CONTROL Register bit 0 equals 1.	R	0h	RESET
5:4	Reserved	R	-	-
3:1	INTID These bits identify the highest priority interrupt pending as indicated by Table 17-8, "Interrupt Control". In non-FIFO mode, Bit[3] is a logic “0”. In FIFO mode Bit[3] is set along with Bit[2] when a timeout interrupt is pending.	R	0h	RESET
0	IPEND This bit can be used in either a hardwired prioritized or polled environment to indicate whether an interrupt is pending. When bit 0 is a logic ‘0’ an interrupt is pending and the contents of the IIR may be used as a pointer to the appropriate internal service routine. When bit 0 is a logic ‘1’ no interrupt is pending.	R	1h	RESET

TABLE 17-8: INTERRUPT CONTROL

FIFO Mode Only	Interrupt Identification Register			Interrupt SET and RESET Functions			
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**Priority Level**	**Interrupt Type**	**Interrupt Source**	**Interrupt Reset Control**
0	0	0	1	-	None	None	-
	1	1	0	Highest	Receiver Line Status	Overrun Error, Parity Error, Framing Error or Break Interrupt	Reading the Line Status Register
		0		Second	Received Data Available	Receiver Data Available	Read Receiver Buffer or the FIFO drops below the trigger level.
1					Character Time-out Indication	No Characters Have Been Removed From or Input to the RCVR FIFO during the last 4 Char times and there is at least 1 char in it during this time	Reading the Receiver Buffer Register
0	0	1		Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR Register (if Source of Interrupt) or Writing the Transmitter Holding Register
	0	0		Fourth	MODEM Status	Clear to Send or Data Set Ready or Ring Indicator or Data Carrier Detect	Reading the MODEM Status Register

17.11.8 LINE CONTROL REGISTER

Offset	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	DLAB Divisor Latch Access Bit (DLAB). It must be set high (logic “1”) to access the Divisor Latches of the Baud Rate Generator during read or write operations. It must be set low (logic “0”) to access the Receiver Buffer Register, the Transmitter Holding Register, or the Interrupt Enable Register.	R/W	0h	RESET
6	BREAK_CONTROL Set Break Control bit. When bit 6 is a logic “1”, the transmit data output (TXD) is forced to the Spacing or logic “0” state and remains there (until reset by a low level bit 6) regardless of other transmitter activity. This feature enables the Serial Port to alert a terminal in a communications system.	R/W	0h	RESET
5	STICK_PARITY Stick Parity bit. When parity is enabled it is used in conjunction with bit 4 to select Mark or Space Parity. When LCR bits 3, 4 and 5 are 1 the Parity bit is transmitted and checked as a 0 (Space Parity). If bits 3 and 5 are 1 and bit 4 is a 0, then the Parity bit is transmitted and checked as 1 (Mark Parity). If bit 5 is 0 Stick Parity is disabled. Bit 3 is a logic “1” and bit 5 is a logic “1”, the parity bit is transmitted and then detected by the receiver in the opposite state indicated by bit 4.	R/W	0h	RESET
4	PARITY_SELECT Even Parity Select bit. When bit 3 is a logic “1” and bit 4 is a logic “0”, an odd number of logic “1”'s is transmitted or checked in the data word bits and the parity bit. When bit 3 is a logic “1” and bit 4 is a logic “1” an even number of bits is transmitted and checked.	R/W	0h	RESET
3	ENABLE_PARITY Parity Enable bit. When bit 3 is a logic “1”, a parity bit is generated (transmit data) or checked (receive data) between the last data word bit and the first stop bit of the serial data. (The parity bit is used to generate an even or odd number of 1s when the data word bits and the parity bit are summed).	R/W	0h	RESET
2	STOP_BITS This bit specifies the number of stop bits in each transmitted or received serial character. TABLE 17-9: summarizes the information.	R/W	0h	RESET
1:0	WORD_LENGTH These two bits specify the number of bits in each transmitted or received serial character. The encoding of bits 0 and 1 is as follows:	R/W	0h	RESET

TABLE 17-9: STOP BITS

Bit 2	Word Length	Number of Stop Bits
0	--	1
1	5 bits	1.5
	6 bits	2
	7 bits	
	8 bits	

Note 17-1 The receiver will ignore all stop bits beyond the first, regardless of the number used in transmitting.

TABLE 17-10: SERIAL CHARACTER

Bit 1	Bit 0	Word Length
0	0	5 Bits
0	1	6 Bits
1	0	7 Bits
1	1	8 Bits

The Start, Stop and Parity bits are not included in the word length.

17.11.9 MODEM CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:5	Reserved	R	-	-
4	LOOPBACK This bit provides the loopback feature for diagnostic testing of the Serial Port. When bit 4 is set to logic “1”, the following occur: 1. The TXD is set to the Marking State (logic “1”). 2. The receiver Serial Input (RXD) is disconnected. 3. The output of the Transmitter Shift Register is “looped back” into the Receiver Shift Register input. 4. All MODEM Control inputs (nCTS, nDSR, nRI and nDCD) are disconnected. 5. The four MODEM Control outputs (nDTR, nRTS, OUT1 and OUT2) are internally connected to the four MODEM Control inputs (nDSR, nCTS, RI, DCD). 6. The Modem Control output pins are forced inactive high. 7. Data that is transmitted is immediately received. This feature allows the processor to verify the transmit and receive data paths of the Serial Port. In the diagnostic mode, the receiver and the transmitter interrupts are fully operational. The MODEM Control Interrupts are also operational but the interrupts' sources are now the lower four bits of the MODEM Control Register instead of the MODEM Control inputs. The interrupts are still controlled by the Interrupt Enable Register.	R/W	0h	RESET
3	OUT2 Output 2 (OUT2). This bit is used to enable an UART interrupt. When OUT2 is a logic “0”, the serial port interrupt output is forced to a high impedance state - disabled. When OUT2 is a logic “1”, the serial port interrupt outputs are enabled.	R/W	0h	RESET
2	OUT1 This bit controls the Output 1 (OUT1) bit. This bit does not have an output pin and can only be read or written by the CPU.	R/W	0h	RESET
1	RTS This bit controls the Request To Send (nRTS) output. Bit 1 affects the nRTS output in a manner identical to that described above for bit 0.	R/W	0h	RESET
0	DTR This bit controls the Data Terminal Ready (nDTR) output. When bit 0 is set to a logic “1”, the nDTR output is forced to a logic “0”. When bit 0 is a logic “0”, the nDTR output is forced to a logic “1”.	R/W	0h	RESET

17.11.10 LINE STATUS REGISTER

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	FIFO_ERROR This bit is permanently set to logic “0” in the 450 mode. In the FIFO mode, this bit is set to a logic “1” when there is at least one parity error, framing error or break indication in the FIFO. This bit is cleared when the LSR is read if there are no subsequent errors in the FIFO.	R	0h	RESET
6	TRANSMIT_ERROR Transmitter Empty. Bit 6 is set to a logic “1” whenever the Transmitter Holding Register (THR) and Transmitter Shift Register (TSR) are both empty. It is reset to logic “0” whenever either the THR or TSR contains a data character. Bit 6 is a read only bit. In the FIFO mode this bit is set whenever the THR and TSR are both empty,	R	0h	RESET
5	TRANSMIT_EMPTY Transmitter Holding Register Empty Bit 5 indicates that the Serial Port is ready to accept a new character for transmission. In addition, this bit causes the Serial Port to issue an interrupt when the Transmitter Holding Register interrupt enable is set high. The THRE bit is set to a logic “1” when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. The bit is reset to logic “0” whenever the CPU loads the Transmitter Holding Register. In the FIFO mode this bit is set when the XMIT FIFO is empty, it is cleared when at least 1 byte is written to the XMIT FIFO. Bit 5 is a read only bit.	R	0h	RESET
4	BREAK_INTERRUPT Break Interrupt. Bit 4 is set to a logic “1” whenever the received data input is held in the Spacing state (logic “0”) for longer than a full word transmission time (that is, the total time of the start bit + data bits + parity bits + stop bits). The BI is reset after the CPU reads the contents of the Line Status Register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO. When break occurs only one zero character is loaded into the FIFO. Restarting after a break is received, requires the serial data (RXD) to be logic “1” for at least 1/2 bit time. Bits 1 through 4 are the error conditions that produce a Receiver Line Status Interrupt BIT 3 whenever any of the corresponding conditions are detected and the interrupt is enabled	R	0h	RESET

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	FRAME_ERROR Framing Error. Bit 3 indicates that the received character did not have a valid stop bit. Bit 3 is set to a logic “1” whenever the stop bit following the last data bit or parity bit is detected as a zero bit (Spacing level). This bit is reset to a logic “0” whenever the Line Status Register is read. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO. The Serial Port will try to resynchronize after a framing error. To do this, it assumes that the framing error was due to the next start bit, so it samples this 'start' bit twice and then takes in the 'data'.	R	0h	RESET
2	PARITY ERROR Parity Error. Bit 2 indicates that the received data character does not have the correct even or odd parity, as selected by the even parity select bit. This bit is set to a logic “1” upon detection of a parity error and is reset to a logic “0” whenever the Line Status Register is read. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO.	R	0h	RESET
1	OVERRUN_ERROR Overrun Error. Bit 1 indicates that data in the Receiver Buffer Register was not read before the next character was transferred into the register, thereby destroying the previous character. In FIFO mode, an overrun error will occur only when the FIFO is full and the next character has been completely received in the shift register, the character in the shift register is overwritten but not transferred to the FIFO. This bit is set to a logic “1” immediately upon detection of an overrun condition, and reset whenever the Line Status Register is read.	R	0h	RESET
0	DATA_READY Data Ready. It is set to a logic ‘1’ whenever a complete incoming character has been received and transferred into the Receiver Buffer Register or the FIFO. Bit 0 is reset to a logic ‘0’ by reading all of the data in the Receive Buffer Register or the FIFO.	R	0h	RESET

17.11.11 MODEM STATUS REGISTER

Offset	06h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	DCD This bit is the complement of the Data Carrier Detect (nDCD) input. If bit 4 of the MCR is set to logic ‘1’, this bit is equivalent to OUT2 in the MCR.	R	0h	RESET
6	RI# This bit is the complement of the Ring Indicator (nRI) input. If bit 4 of the MCR is set to logic ‘1’, this bit is equivalent to OUT1 in the MCR.	R	0h	RESET
5	DSR This bit is the complement of the Data Set Ready (nDSR) input. If bit 4 of the MCR is set to logic ‘1’, this bit is equivalent to DTR in the MCR.	R	0h	RESET
4	CTS This bit is the complement of the Clear To Send (nCTS) input. If bit 4 of the MCR is set to logic ‘1’, this bit is equivalent to nRTS in the MCR.	R	0h	RESET
3	DCD Delta Data Carrier Detect (DDCD). Bit 3 indicates that the nDCD input to the chip has changed state. NOTE: Whenever bit 0, 1, 2, or 3 is set to a logic ‘1’, a MODEM Status Interrupt is generated.	R	0h	RESET
2	RI Trailing Edge of Ring Indicator (TERI). Bit 2 indicates that the nRI input has changed from logic ‘0’ to logic ‘1’.	R	0h	RESET
1	DSR Delta Data Set Ready (DDSR). Bit 1 indicates that the nDSR input has changed state since the last time the MSR was read.	R	0h	RESET
0	CTS Delta Clear To Send (DCTS). Bit 0 indicates that the nCTS input to the chip has changed state since the last time the MSR was read.	R	0h	RESET

Note: The Modem Status Register (MSR) only provides the current state of the UART MODEM control lines in Loopback Mode. The MEC140x/1x does not support external connections for the MODEM Control inputs (nCTS, nDSR, nRI and nDCD) or for the four MODEM Control outputs (nDTR, nRTS, OUT1 and OUT2).

17.11.12 SCRATCHPAD REGISTER

Offset	07h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	SCRATCH This 8 bit read/write register has no effect on the operation of the Serial Port. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.	R/W	0h	RESET

18.0 BASIC TIMER

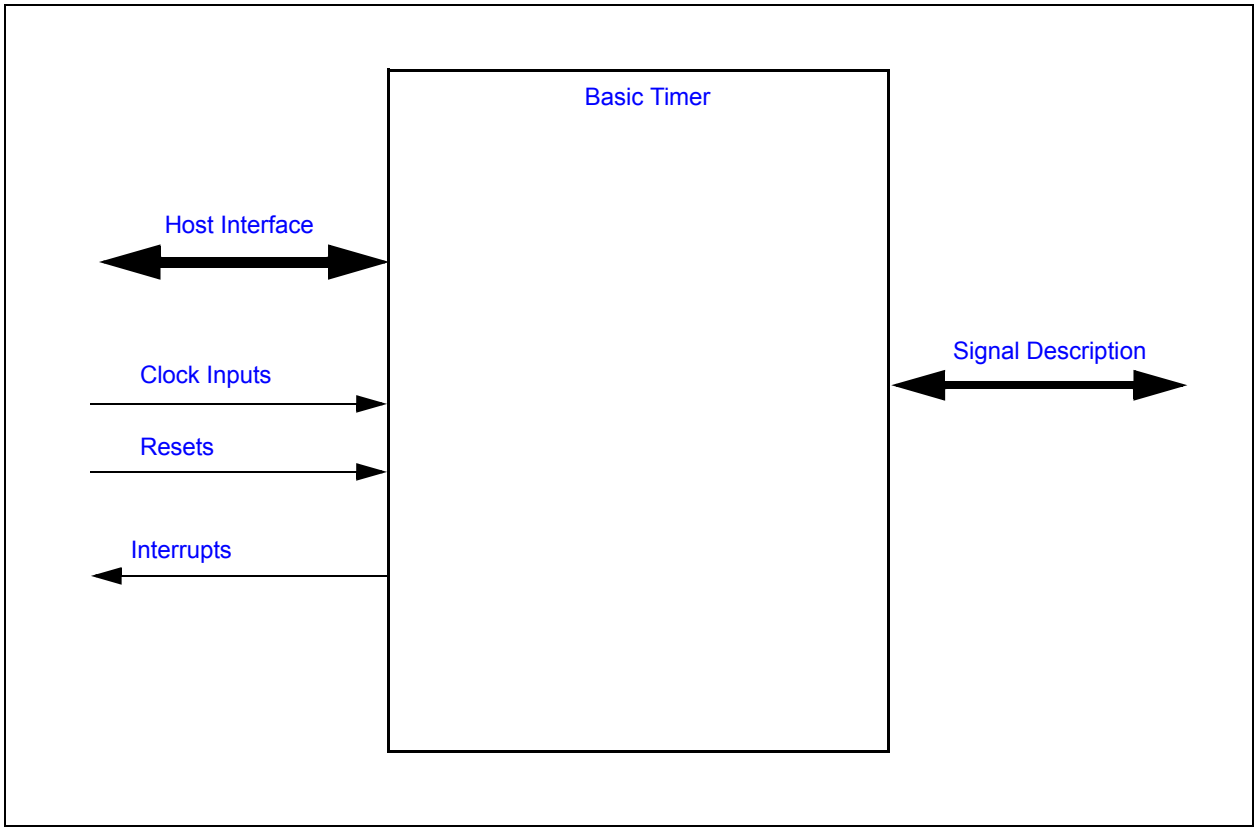
18.1 Introduction

This timer block offers a simple mechanism for firmware to maintain a time base. This timer may be instantiated as 16 bits or 32 bits. The name of the timer instance indicates the size of the timer.

18.2 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 18-1: I/O DIAGRAM OF BLOCK



18.3 Signal Description

There are no external signals for this block.

18.4 Host Interface

The embedded controller may access this block via the registers defined in Section 18.9, "EC-Only Registers," on page 288.

18.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

18.5.1 POWER DOMAINS

Name	Description
VTR	The timer control logic and registers are all implemented on this single power domain.

18.5.2 CLOCK INPUTS

Name	Description
48 MHz Ring Oscillator	This is the clock source to the timer logic. The Pre-scaler may be used to adjust the minimum resolution per bit of the counter.

18.5.3 RESETS

Name	Description
nSYSRST	This reset signal, which is an input to this block, resets all the logic and registers to their initial default state.
Soft Reset	This reset signal, which is created by this block, resets all the logic and registers to their initial default state. This reset is generated by the block when the **SOFT_RESET** bit is set in the Timer Control Register register.
Timer_Reset	This reset signal, which is created by this block, is asserted when either the nSYSRST or the Soft Reset signal is asserted. The nSYSRST and Soft Reset signals are OR'd together to create this signal.

18.6 Interrupts

Source	Description
Timer_Event	This interrupt event fires when a 16-bit timer *x* reaches its limit. This event is sourced by the EVENT_INTERRUPT status bit if enabled.

18.7 Low Power Modes

The Basic Timer may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. This block is only be permitted to enter low power modes when the block is not active.

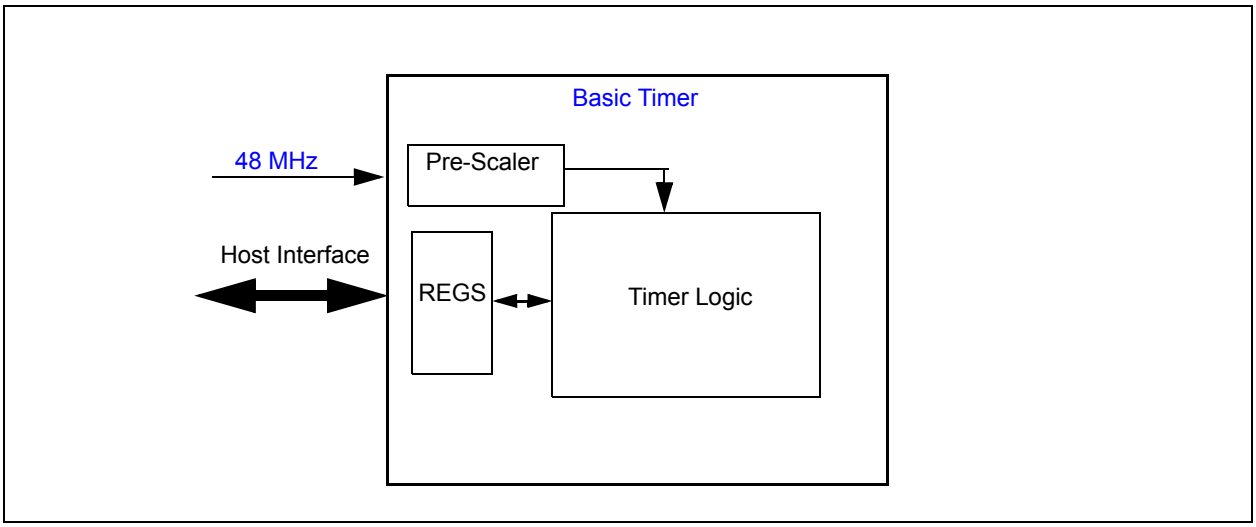
The sleep state of this timer is as follows:

- Asleep while the block is not Enabled
- Asleep while the block is not running (start inactive).
- Asleep while the block is halted (even if running).

The block is active while start is active.

18.8 Description

FIGURE 18-2: BLOCK DIAGRAM



This timer block offers a simple mechanism for firmware to maintain a time base in the design. The timer may be enabled to execute the following features:

- Programmable resolution per LSB of the counter via the Pre-scale bits in the Timer Control Register
- Programmable as either an up or down counter
- One-shot or Continuous Modes
- In one-shot mode the Auto Restart feature stops the counter when it reaches its limit and generates a level event.
- In Continuous Mode the Auto Restart feature restarts that counter from the programmed preload value and generates a pulse event.
- Counter may be reloaded, halted, or started via the Timer Control register
- Block may be reset by either a Power On Reset (POR) or via a Soft Reset.

18.9 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Basic Timer. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 18-1: EC-ONLY REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
TIMER16 (16-bit Timer)	0	EC	32-bit internal address space	0000_0C00h
TIMER16 (16-bit Timer)	1	EC	32-bit internal address space	0000_0C20h
TIMER16 (16-bit Timer)	2	EC	32-bit internal address space	0000_0C40h
TIMER16 (16-bit Timer)	3	EC	32-bit internal address space	0000_0C60h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 18-2: RUNTIME REGISTER SUMMARY

Offset	Register Name
00h	Timer Count Register
04h	Timer Preload Register
08h	Timer Status Register
0Ch	Timer Int Enable Register
10h	Timer Control Register

18.9.1 TIMER COUNT REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	COUNTER This is the value of the Timer counter. This is updated by Hardware but may be set by Firmware. If it is set while the Hardware Timer is operating, functionality can not be ensured. When read, it is buffered so single byte reads will be able to catch the full 4 byte register without it changing. The size of the Counter is indicated by the instance name. Bits 0 to (size-1) are r/w counter bits. Bits 31 down to size are reserved. Reads return 0 and writes have no effect.	R/W	0h	Timer_Reset

18.9.2 TIMER PRELOAD REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	PRE_LOAD This is the value of the Timer pre-load for the counter. This is used by H/W when the counter is to be restarted automatically; this will become the new value of the counter upon restart. The size of the Pre-Load value is the same as the size of the counter. The size of the Counter is indicated by the instance name. Bits 0 to (size-1) are r/w pre-load bits. Bits 31 down to size are reserved. Reads return 0 and writes have no effect.	R/W	0h	Timer_Reset

18.9.3 TIMER STATUS REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Reserved	R	-	-
0	EVENT_INTERRUPT This is the interrupt status that fires when the timer reaches its limit. This may be level or a self clearing signal cycle pulse, based on the AUTO_RESTART bit in the Timer Control Register. If the timer is set to automatically restart, it will provide a pulse, otherwise a level is provided.	R/WC	0h	Timer_Reset

18.9.4 TIMER INT ENABLE REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Reserved	R	-	-
0	EVENT_INTERRUPT_ENABLE This is the interrupt enable for the status EVENT_INTERRUPT bit in the Timer Status Register	R/W	0h	Timer_Reset

18.9.5 TIMER CONTROL REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	PRE_SCALE This is used to divide down the system clock through clock enables to lower the power consumption of the block and allow slow timers. Updating this value during operation may result in erroneous clock enable pulses until the clock divider restarts. The number of clocks per clock enable pulse is (Value + 1); a setting of 0 runs at the full clock speed, while a setting of 1 runs at half speed.	R/W	0h	Timer_Reset
15:8	Reserved	R	-	-
7	HALT This is a halt bit. This will halt the timer as long as it is active. Once the halt is inactive, the timer will start from where it left off. 1=Timer is halted. It stops counting. The clock divider will also be reset. 0=Timer runs normally	R/W	0h	Timer_Reset

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
6	RELOAD This bit reloads the counter without interrupting it operation. This will not function if the timer has already completed (when the START bit in this register is '0'). This is used to periodically prevent the timer from firing when an event occurs. Usage while the timer is off may result in erroneous behavior.	R/W	0h	Timer_Reset
5	START This bit triggers the timer counter. The counter will operate until it hits its terminating condition. This will clear this bit. It should be noted that when operating in restart mode, there is no terminating condition for the counter, so this bit will never clear. Clearing this bit will halt the timer counter. Setting this bit will: • Reset the clock divider counter. • Enable the clock divider counter. • Start the timer counter. • Clear all interrupts. Clearing this bit will: • Disable the clock divider counter. • Stop the timer counter.	R/W	0h	Timer_Reset
4	SOFT_RESET This is a soft reset. This is self clearing 1 cycle after it is written.	WO	0h	Timer_Reset
3	AUTO_RESTART This will select the action taken upon completing a count. 1=The counter will automatically restart the count, using the contents of the Timer Preload Register to load the Timer Count Register The interrupt will be set in edge mode 0=The counter will simply enter a done state and wait for further control inputs. The interrupt will be set in level mode.	R/W	0h	Timer_Reset
2	COUNT_UP This selects the counter direction. When the counter in incrementing the counter will saturate and trigger the event when it reaches all F's. When the counter is decrementing the counter will saturate when it reaches 0h. 1=The counter will increment 0=The counter will decrement	R/W	0h	Timer_Reset
1	Reserved	R	-	-
0	ENABLE This enables the block for operation. 1=This block will function normally 0=This block will gate its clock and go into its lowest power state	R/W	0h	Timer_Reset

19.0 RTOS TIMER

19.1 Introduction

The RTOS Timer is a low-power, 32-bit timer designed to operate on the 32kHz oscillator which is available during all chip sleep states. This allows firmware the option to sleep the processor, enter heavy or deep chip sleep states, and wake after a programmed amount of time. The timer may be used as a one-shot timer or a continuous timer. When the timer transitions to 0 it is capable of generating a wake-capable interrupt to the embedded controller. This timer may be halted during debug by hardware or via a software control bit.

19.2 References

No references have been cited for this chapter

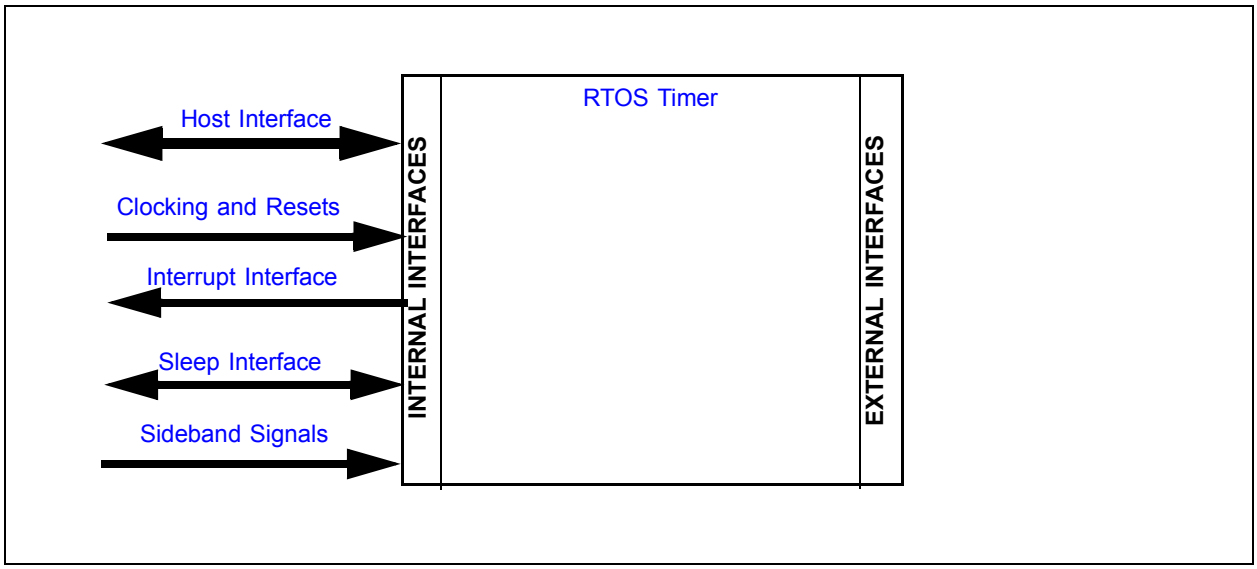
19.3 Terminology

No terms have been cited for this chapter.

19.4 Interface

This block is an IP block designed to be incorporated into a chip. It is designed to be accessed internally via a registered host interface. The following diagram illustrates the various interfaces to the block.

FIGURE 19-1: RTOS TIMER INTERFACE DIAGRAM



19.4.1 HOST INTERFACE

The registers defined in Section 19.9, "RTOS Timer Registers," on page 296 are accessible by the Host Interfaces defined in Table 19-6, “RTOS Timer Registers Base Address,” on page 296.

19.4.2 CLOCKING AND RESETS

This IP block has the following clocks and reset ports. For a complete list of all the clocks and resets associated with this block see Section 19.5, "Power, Clocks and Resets," on page 294.

TABLE 19-1: CLOCKING AND RESETS SIGNAL DESCRIPTION

Name	Direction	Description
nSYSRST	Input	Reset asserted when power is applied to this block
48 MHz Ring Oscillator	Input	System Clock
32KHz_Clk	Input	Timer Clock

19.4.3 INTERRUPT INTERFACE

This section defines the interrupt Interface signals routed to the chip interrupt aggregator.

TABLE 19-2: INTERRUPT INTERFACE SIGNAL DESCRIPTION

Name	Direction	Description
RTOS_TIMER	Output	RTOS Timer Interrupt Event

19.4.4 SLEEP INTERFACE

TABLE 19-3: SIDEBAND SIGNALS SIGNAL DESCRIPTION

Name	Direction	Description
Sleep Enable	Input	Firmware Sleep Request to turn off 48 MHz Ring Oscillator to this block. **Note:** This input is controlled by the RTOS Timer Sleep Enable bit located in the chip's EC Sleep Enable 2 Register (EC_SLP_EN2) on page 83.
Clock Required	Output	Signal indicating this block requires the 48 MHz Ring Oscillator for operation. **Note:** Firmware may read the value of the RTOS Timer Clock Required signal in the chip's EC Clock Required 2 Status Register (EC_CLK_REQ2_STS) on page 85.

19.4.5 SIDEBAND SIGNALS

TABLE 19-4: INTERRUPT INTERFACE SIGNAL DESCRIPTION

Name	Direction	Description
Halt	Input	RTOS Timer Halt signal. **Note:** This signal is connected to the same signal that halts the embedded controller during debug (e.g., JTAG Debugger is active, break points, etc.).

19.5 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

19.5.1 POWER DOMAINS

Name	Description
VTR	This power well sources all of the registers and logic in this block.

19.5.2 CLOCKS

This section describes all the clocks in the block, including those that are derived from the I/O Interface as well as the ones that are derived or generated internally.

Name	Description
32KHz_Clk	Timer Clock Source
48 MHz Ring Oscillator	System Clock used by Host Interface for register access

19.5.3 RESETS

Name	Description
nSYSRST	This power on reset (POR) signal resets all of the registers and logic in this block.

19.6 Interrupt Generation

This section defines the Interrupt Sources generated from this block.

Source	Description
RTOS_TIMER	**Note:** The RTOS Timer block generates a pulse anytime the RTOS Timer transitions from 1 to 0. This pulse is used to generate a wake-capable interrupt event that is latched by the Jump Table Vectored Interrupt Controller (JTVIC).

19.7 Low Power Modes

The RTOS Timer may be put into a low power state by the chip Power, Clocks, and Reset (PCR) circuitry.

The timer operates off of the 32KHz_Clk, and therefore will operate normally when 48 MHz Ring Oscillator is stopped. The sleep enable input has no effect on the RTOS Timer and the clock required output is only asserted during register read/write cycles for as long as necessary to propagate updates to the block core.

19.7.1 SLEEP INTERFACE - SYSTEM CLOCK

The RTOS Timer is designed to always operate in its lowest functional power consumption state. In addition, it can be commanded to enter a lower power state via the Sleep Enable signal. The block notifies the chip's power management circuitry when it is in its low power state by driving the Clock Required signal low. The following table defines all the blocks Power States associated with the System Clock.

Note:	The logic clocked by the system clock is considered to be in the idle state when the host is not accessing the register interface.

TABLE 19-5: RTOS Timer - SYSTEM CLOCK POWER STATES

Power State	Block Enable Bit	Sleep Enable	Clock Required	Description
Idle	x	x	0	Block is idle and operating in its lowest power consumption state. The 48 MHz Ring Oscillator is not used in this state. The block automatically enters this state anytime it is not performing a function requiring this clock source (e.g., Register accesses).
Operating	x	x	1	Block is not idle. This block will assert Clock Required signal only during register access and when it needs to generate interrupt. The sleep_en signal has no effect on this clock requirement.

Note: The RTOS Timer Registers are readable and writable in all defined Power States.

19.7.2 WAKING FROM LOW POWER STATES

The chip Power, Clocks, and Resets logic is responsible for monitoring wake events that turn on 48 MHz Ring Oscillator. The RTOS_TIMER interrupt event is a wake-capable event that may be used to turn on 48 MHz Ring Oscillator.

19.8 Description

The RTOS Timer is a very basic timer with simple down counter functionality with auto-reload and halt features. The timer counts with Timer Clock when the timer is programmed with pre-load value.

The counter can be configured as one-shot timer by not setting the Auto Reload bit. The timer will load the value of the pre-load register and start to count down when the Timer Start bit is asserted by the firmware. The timer will generate interrupt when the counter transitions from count = 1 to count = 0 as defined in the Interrupt Generation section.

If the timer is needed again with same pre-load value, firmware has to only set the Timer Start bit. This will restart the timer again.

The counter can also be programmed as continuous running mode by enabling the Auto Reload bit. In this mode counter reloads itself every time timer equals 0. The timer also generates interrupt as defined in the interrupt section.

If the RTOS Timer Pre-Load register is written when the counter is counting, the new preload value will take effect only when the counter reaches 0 if the auto-reload bit has been set.

If the RTOS Timer Pre-Load register is programmed with 32'h0 while the Timer is counting, the Timer will continue to count until it counts to 0. Then the Timer Start bit will be cleared. If the Timer Start bit is written when the RTOS Timer Pre-Load register is 0, the Timer Start bit will be self-cleared.

19.8.1 EXTERNAL HARDWARE HALT

The Halt signal is an input signal to the block. This signal when asserted (high) and enabled in the Timer Control Register will halt the counter. When this signal is de-asserted (low), the timer will continue to count.

19.8.2 FIRMWARE HALT

The Timer can also be halted by setting Firmware Timer Halt bit in the Timer Control Register.

19.9 RTOS Timer Registers

The registers listed in the Table 19-7, "RTOS Timer Registers Summary" are for a single instance of the RTOS Timer block. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in Table 19-6, "RTOS Timer Registers Base Address".

TABLE 19-6: RTOS TIMER REGISTERS BASE ADDRESS

Instance Name	Instance Number	Host	Address Space	Base Address
RTOS Timer	0	EC	32-bit internal address space	0000_7400h

Note: The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 19-7: RTOS TIMER REGISTERS SUMMARY

Offset	Register Name
00h	RTOS Timer Count Value
04h	RTOS Timer Pre-Load
08h	Timer Control

19.9.1 RTOS TIMER COUNT VALUE

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Timer Count Value This is the value of the RTOS Timer counter. This is the actual Timer counter value. **Note:** This register should be read as DWORD. There is no latching mechanism of the upper bytes implemented, if the register is accessed as byte/word. Reading the register as byte/word may not give you true counter value.	R	0b	nSYSRST

19.9.2 RTOS TIMER PRE-LOAD

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Timer Pre-Load Count Value This is the pre load value for the counter. This value is loaded in the timer counter after setting the Timer Start bit or when the counter reloads if the Auto Reload bit is set. **Note:** This register must be programmed with new Pre-Load count value before Timer Start bit is enabled. If this sequence is not followed, the new Pre-Load count value will only take effect when the counter expires if the Auto Reload bit is set. **Note:** Programming this register with 0's will disable the counter and clear the "start" bit if set.	R/W	0h	nSYSRST

19.9.3 TIMER CONTROL

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:5	RESERVED	RES	-	-
4	Firmware Timer Halt This bit gives the firmware the ability to halt the counter without the use of the hardware Halt signal. 0: Do not halt the counter 1: Halt the counter	R/W	0h	nSYSRST
3	Ext Hardware Halt Enable 0: Do not allow hardware Halt signal to stop the counter. 1: Allow hardware Halt signal to stop the counter.	R/W	0h	nSYSRST

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	Timer Start This bit triggers the timer counter. The counter will operate until it hits its terminating condition. This will clear this bit. It should be noted that while operating in Auto Reload mode; there is no terminating condition for the counter, so this bit will never clear. Clearing this bit by firmware will reset the timer counter. Setting this bit will: Load the Pre-Load value into the counter. Start the timer counter. Clearing this bit will: Reset the counter to 0. Will not generate the interrupt. The hardware will clear this bit for following conditions: When One-Shot mode expires. When TimerPreLoad = TimerCountValue=0	R/W	0h	nSYSRST
1	Auto Reload This will select the action taken upon completing a count. ***0:*** The counter will simply enter a done state and wait for further control inputs. One-Shot mode. ***1:*** The counter will automatically restart the count using the RTOS Timer Pre-load value. Continuous mode.	R/W	0h	nSYSRST
0	Block Enable This bit enables the block for operation. ***0:*** This bit will gate Timer clock and go into its lowest power state. Falling edge of this bit will clear all the timer logic and register bits to default state. ***1:*** This block will function normally. **Note:** Registers are always accessible regardless of the state of this bit.	R/W	0h	nSYSRST

20.0 HIBERNATION TIMER

20.1 Introduction

The Hibernation Timer can generate a wake event to the Embedded Controller (EC) when it is in a hibernation mode. This block supports wake events up to 2 hours in duration. The timer is a 16-bit binary count-down timer that can be programmed in 30.5µs and 0.125 second increments for period ranges of 30.5µs to 2s or 0.125s to 136.5 minutes, respectively. Writing a non-zero value to this register starts the counter from that value. A wake-up interrupt is generated when the count reaches zero.

20.2 References

No references have been cited for this chapter

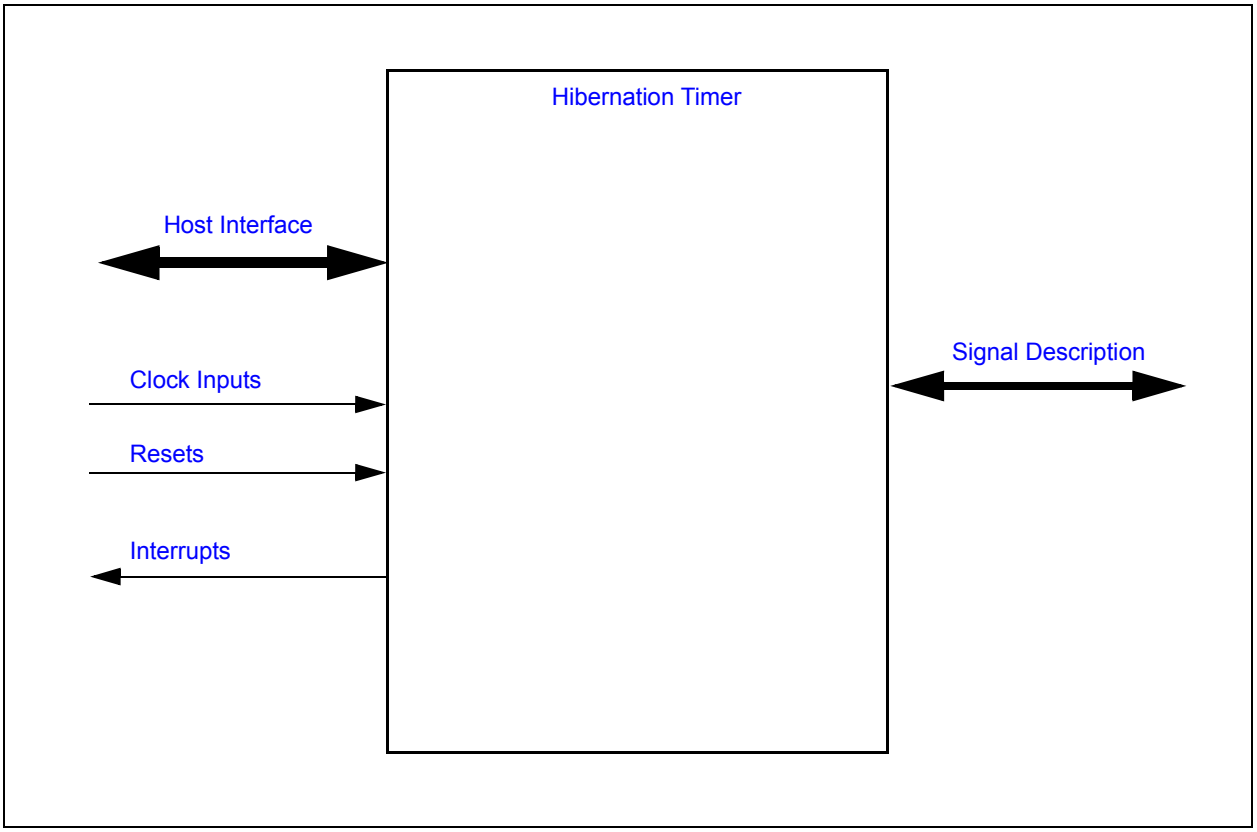
20.3 Terminology

No terms have been cited for this chapter.

20.4 Interface

This block is an IP block designed to be incorporated into a chip. It is designed to be accessed externally via the pin interface and internally via a registered host interface. The following diagram illustrates the various interfaces to the block.

FIGURE 20-1: HIBERNATION TIMER INTERFACE DIAGRAM



20.5 Signal Description

There are no external signals for this block.

20.6 Host Interface

The registers defined for the Hibernation Timer are accessible by the various hosts as indicated in Section 20.10, "EC-Only Registers".

20.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

20.7.1 POWER DOMAINS

Name	Description
VTR	The timer control logic and registers are all implemented on this single power domain.

20.7.2 CLOCK INPUTS

Name	Description
5Hz_Clk	This is the clock source to the timer logic. The Pre-scaler may be used to adjust the minimum resolution per bit of the counter. if the main oscillator is stopped then an external 32.768kHz clock source must be active for the Hibernation Timer to continue to operate.

20.7.3 RESETS

Name	Description
nSYSRST	This reset signal, which is an input to this block, resets all the logic and registers to their initial default state.

20.8 Interrupts

This section defines the interrupt Interface signals routed to the chip interrupt aggregator.

Each instance of the Hibernation Timer in the MEC140x/1x can be used to generate interrupts and wake-up events when the timer decrements to zero.

TABLE 20-1: INTERRUPT INTERFACE SIGNAL DESCRIPTION

Name	Direction	Description
HTIMER	Output	Signal indicating that the timer is enabled and decrements to 0. This signal is used to generate an Hibernation Timer interrupt event.

20.9 Low Power Modes

The Hibernation Timer may be put into a low power state by the chip Power, Clocks, and Reset (PCR) circuitry.

The timer operates off of the 5Hz_Clk, and therefore will operate normally when 48 MHz Ring Oscillator is stopped.

The sleep enable inputs have no effect on the Hibernation Timer and the clock required outputs are only asserted during register read/write cycles for as long as necessary to propagate updates to the block core.

20.10 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Hibernation Timer. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 20-2: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
Hibernation Timer	0	EC	32-bit internal address space	0000_9800h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 20-3: HIBERNATION TIMER SUMMARY

Offset	Register Name
00h	HTimer Preload Register
04h	HTimer Control Register
08h	HTimer Count Register

20.10.1 HTIMER PRELOAD REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	HT_PRELOAD This register is used to set the Hibernation Timer Preload value. Writing this register to a non-zero value resets the down counter to start counting down from this programmed value. Writing this register to 0000h disables the hibernation counter. The resolution of this timer is determined by the CTRL bit in the HTimer Control Register. Writes to the HTimer Control Register are completed with an EC bus cycle.	R/W	000h	nSYSRST

20.10.2 HTIMER CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:1	Reserved	R	-	-
0	CTRL 1= The Hibernation Timer has a resolution of 0.125s per LSB, which yields a maximum time in excess of 2 hours. 0= The Hibernation Timer has a resolution of 30.5µs per LSB, which yields a maximum time of ~2seconds.	R	0000h	nSYSRST

20.10.3 HTIMER COUNT REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	COUNT The current state of the Hibernation Timer.	R	0000h	nSYSRST

21.0 RTC/WEEK TIMER

21.1 Introduction

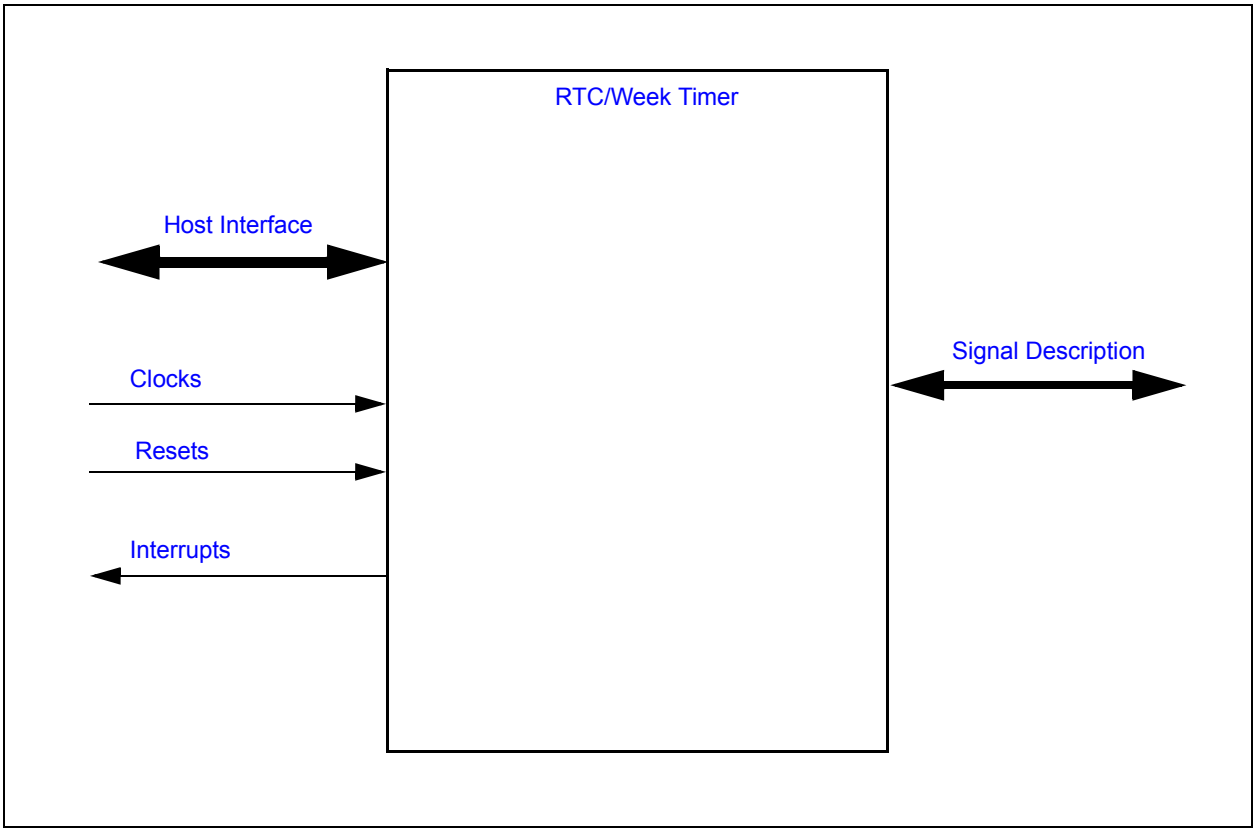
The RTC/Week Alarm Interface provides two timekeeping functions: a Week Timer and a Sub-Week Timer. Both the Week Timer and the Sub-Week Timer assert the Power-Up Event Output which automatically powers-up the system from the G3 state. Features include:

- EC interrupts based on matching a counter value
- Repeating interrupts at 1 second and sub-1 second intervals
- System Wake capability on interrupts, including Wake from Deep Sleep.

21.2 Interface

This block's connections are entirely internal to the chip.

FIGURE 21-1: I/O DIAGRAM OF BLOCK



21.3 Signal Description

TABLE 21-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
BGPO	OUTPUT	Battery-powered general purpose output

TABLE 21-2: INTERNAL SIGNAL DESCRIPTION TABLE

Name	Direction	Description
POWER_UP_EVENT	OUTPUT	Signal to the VBAT-Powered Control Interface. When this signal is asserted, the VCI output signal asserts. See Section 21.8, "Power-Up Events".

21.4 Host Interface

The registers defined for the RTC/Week Timer are accessible only by the EC.

21.5 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

21.5.1 POWER DOMAINS

TABLE 21-3: POWER SOURCES

Name	Description
VBAT	This power well sources all of the internal registers and logic in this block.
VTR	This power well sources only bus communication. The block continues to operate internally while this rail is down.

21.5.2 CLOCKS

TABLE 21-4: CLOCKS

Name	Description
32KHz_Clk	This 32KHz clock input drives all internal logic, and will be present at all times that the VBAT well is powered.

21.5.3 RESETS

TABLE 21-5: RESET SIGNALS

Name	Description
VBAT_POR	This reset signal is used reset all of the registers and logic in this block.
VTR_RESET#	This reset signal is used to inhibit the bus communication logic, and isolates this block from VTR powered circuitry on-chip. Otherwise it has no effect on the internal state.

21.6 Interrupts

TABLE 21-6: EC INTERRUPTS

Source	Description
WEEK_ALARM_INT	This interrupt is signaled to the Interrupt Aggregator when the Week Alarm Counter Register is greater than or equal to the Week Timer Compare Register. The interrupt signal is always generated by the RTC/Week Timer if the block is enabled; the interrupt is enabled or disabled in the Interrupt Aggregator.
SUB_WEEK_ALARM_INT	This interrupt is signaled to the Interrupt Aggregator when the Sub-Week Alarm Counter Register decrements from '1' to '0'. The interrupt signal is always generated by the RTC/Week Timer if the block is enabled; the interrupt is enabled or disabled in the Interrupt Aggregator.
ONE_SECOND	This interrupt is signaled to the Interrupt Aggregator at an isochronous rate of once per second. The interrupt signal is always generated by the RTC/Week Timer if the block is enabled; the interrupt is enabled or disabled in the Interrupt Aggregator.
SUB_SECOND	This interrupt is signaled to the Interrupt Aggregator at an isochronous rate programmable between 0.5Hz and 32.768KHz. The rate interrupts are signaled is determined by the SPISR field in the Sub-Second Programmable Interrupt Select Register. See Table 21-10, "SPISR Encoding". The interrupt signal is always generated by the RTC/Week Timer if the block is enabled; the interrupt is enabled or disabled in the Interrupt Aggregator.
SYSPWR_PRES	This wake interrupt is signaled to the Interrupt Aggregator when an Alarm event occurs. The associated GPIO pin Control Register must be programmed in order to configure the interrupt condition.

21.7 Low Power Modes

The RTC/Week Alarm has no low-power modes. It runs continuously while the VBAT well is powered.

21.8 Power-Up Events

The RTC/Week Timer POWER_UP_EVENT can be used to power up the system after a timed interval. The POWER_UP_EVENT is routed to the VBAT-Powered Control Interface. The VCI_OUT pin that is part of the VCI is asserted if the POWER_UP_EVENT is asserted.

The POWER_UP_EVENT can be asserted under the following two conditions:

1. The Week Alarm Counter Register is greater than or equal to the Week Timer Compare Register
2. The Sub-Week Alarm Counter Register decrements from '1' to '0'

The assertion of the POWER_UP_EVENT is inhibited by the following two conditions:

1. The POWERUP_EN field in the Control Register is '0'
2. The SYSPWR_PRES_ENABLE field in the Sub-Week Control Register is '1' and the SYSPWR_PRES input pin is '0'. This option permits inhibiting a timeout causing a system wake during a deep sleep and draining the battery if AC Power is not present.

Once a POWER_UP_EVENT is asserted the POWERUP_EN bit must be cleared to reset the output. Clearing POWERUP_EN is necessary to avoid unintended power-up cycles.

21.9 Description

The RTC/Week Alarm block provides battery-powered timekeeping functions, derived from a low-power 32KHz clock, that operate even when the device's main power is off. The block contains a set of counters that can be used to generate one-shot and periodic interrupts to the EC for periods ranging from about 30 microseconds to over 8 years. The RTC/Week Alarm can be used in conjunction with the VBAT-Powered Control Interface to power up a sleeping system after a configurable period.

In addition to basic timekeeping, the RTC/Week Alarm block can be used to control the battery-powered general purpose BGPO outputs.

21.9.1 INTERNAL COUNTERS

The RTC/Week Timer includes 3 counters:

21.9.1.1 28-bit Week Alarm Counter

This counter is 28 bits wide. The clock for this counter is the overflow of the Clock Divider, and as long as the RTC/Week Timer is enabled, it is incremented at a 1 Hz rate.

Both an interrupt and a power-up event can be generated when the contents of this counter matches the contents of the Week Timer Compare Register.

21.9.1.2 9-bit Sub-Week Alarm Counter

This counter is 9 bits wide. It is decremented by 1 at each tick of its selected clock. It can be configured either as a one-shot or repeating event generator.

Both an interrupt and a power-up event can be generated when this counter decrements from 1 to 0.

The Sub-Week Alarm Counter can be configured with a number of different clock sources for its time base, derived from either the Week Alarm Counter or the Clock Divider, by setting the SUBWEEK_TICK field of the Sub-Week Control Register.

TABLE 21-7: SUB-WEEK ALARM COUNTER CLOCK

SUBWEEK_ TICK	Source	SPISR	Frequency	Minimum Duration	Maximum Duration
0	Counter Disabled				
1	Sub-Second	0	Counter Disabled		
		1	2 Hz	500 ms	255.5 sec
		2	4 Hz	250 ms	127.8 sec
		3	8 Hz	125 ms	63.9 sec
		4	16 Hz	62.5	31.9 sec
		5	32 Hz	31.25 ms	16.0 sec
		6	64 Hz	15.6 ms	8 sec
		7	128 Hz	7.8 ms	4 sec
		8	256 Hz	3.9 ms	2 sec
		9	512 Hz	1.95 ms	1 sec
		10	1024 Hz	977 µS	499 ms
		11	2048 Hz	488 µS	249.5 ms
		12	4096 Hz	244 µS	124.8 ms
		13	8192 Hz	122 µS	62.4 ms
		14	16.384 Khz	61.1 µS	31.2 ms
		15	32.768 KHz	30.5 µS	15.6 ms
2	Second	n/a	1 Hz	1 sec	511 sec
3	Reserved				
4	Week Counter bit 3	n/a	125 Hz	8 sec	68.1 min

TABLE 21-7: SUB-WEEK ALARM COUNTER CLOCK (CONTINUED)

SUBWEEK_ TICK	Source	SPISR	Frequency	Minimum Duration	Maximum Duration
5	Week Counter bit 5	n/a	31.25 Hz	32 sec	272.5 min
6	Week Counter bit 7	n/a	7.8125 Hz	128 sec	18.17 hour
7	Week Counter bit 9	n/a	1.95 Hz	512 sec	72.68 hour

Note 1: The Week Alarm Counter **must not** be modified by firmware if Sub-Week Alarm Counter is using the Week Alarm Counter as its clock source (i.e., the SUBWEEK_TICK field is set to any of the values 4, 5, 6 or 7). The Sub-Week Alarm Counter must be disabled before changing the Week Alarm Counter. For example, the following sequence may be used:

1. Write 0h to the Sub-Week Alarm Counter Register (disabling the Sub-Week Counter)
2. Write the Week Alarm Counter Register
3. Write a new value to the Sub-Week Alarm Counter Register, restarting the Sub-Week Counter

21.9.1.3 15-bit Clock Divider

This counter is 15 bits wide. The clock for this counter is 32KHz_Clk, and as long as the RTC/Week Timer is enabled, it is incremented at 32.768KHz rate. The Clock Divider automatically The Clock Divider generates a clock out of 1 Hz when the counter wraps from 7FFFh to 0h.

By selecting one of the 15 bits of the counter, using the Sub-Second Programmable Interrupt Select Register, the Clock Divider can be used either to generate a time base for the Sub-Week Alarm Counter or as an isochronous interrupt to the EC, the SUB_SECOND interrupt.. See Table 21-10, "SPISR Encoding" for a list of available frequencies.

21.9.2 TIMER VALID STATUS

If power on reset occurs on the VBAT power rail while the main device power is off, the counters in the RTC/Week Alarm are invalid. If firmware detects a POR on the VBAT power rail after a system boot, by checking the status bits in the Power, Clocks and Resets registers, the RTC/Week Alarm block must be reinitialized.

21.9.3 APPLICATION NOTE: REGISTER TIMING

Register writes in the RTC/Week Alarm complete within two cycles of the 32KHz_Clk clock.The write completes even if the main system clock is stopped before the two cycles of the 32K clock complete. Register reads complete in one cycle of the internal bus clock.

All RTC/Week Alarm interrupts that are asserted within the same cycle of the 32KHz_Clk clock are synchronously asserted to the EC.

21.9.4 APPLICATION NOTE: USE OF THE WEEK TIMER AS A 43-BIT COUNTER

The Week Timer cannot be directly used as a 42-bit counter that is incremented directly by the 32.768KHz clock domain. The upper 28 bits (28-bit Week Alarm Counter) are incremented at a 1Hz rate and the lower 16 bits (15-bit Clock Divider) are incremented at a 32.768KHz rate, but the increments are not performed in parallel. In particular, the upper 28 bits are incremented when the lower 15 bits increment from 0 to 1, so as long as the Clock Divider Register is 0 the two registers together, treated as a single value, have a smaller value then before the lower register rolled over from 7FFFh to 0h.

The following code can be used to treat the two registers as a single large counter. This example extracts a 32-bit value from the middle of the 43-bit counter:

```
dword TIME_STAMP(void)
{
    AHB_dword wct_value;
    AHB_dword cd_value1;
    AHB_dword cd_value2;
    dword irqEnableSave;
```

```
    //Disable interrupts
    irqEnableSave = IRQ_ENABLE;
    IRQ_ENABLE = 0;

    //Read 15-bit clk divider reading register, save result in A
    cd_value1 = WTIMER->CLOCK_DIVIDER;
    //Read 28 bit up-counter timer register, save result in B
    wct_value = WTIMER->WEEK_COUNTER_TIMER;
    //Read 15-bit clk divider reading register, save result in C
    cd_value2 = WTIMER->CLOCK_DIVIDER;

    if (0 == cd_value2)
    {
        wct_value = wct_value + 1;
    }
    else if ( (cd_value2 < cd_value1) || (0 == cd_value1))
    {
        wct_value = WTIMER->WEEK_COUNTER_TIMER;
    }

    //Enable interrupts
    IRQ_ENABLE = irqEnableSave;

    return (WTIMER_BASE + ((wct_value << 10) | (cd_value2>>5)));
}
```

21.10 Runtime Registers

The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in Runtime Register Base Address Table.

TABLE 21-8: RUNTIME REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
Week Alarm	0	EC	32-bit internal Address Space	0000_CC80h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance. Add the register’s Offset to this value to obtain the direct address of the register.

TABLE 21-9: RUNTIME REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	Control Register
04h	Week Alarm Counter Register
08h	Week Timer Compare Register
0Ch	Clock Divider Register
10h	Sub-Second Programmable Interrupt Select Register
14h	Sub-Week Control Register
18h	Sub-Week Alarm Counter Register

21.10.1 CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	R	-	-
6	POWERUP_EN This bit controls the state of the Power-Up Event Output and enables Week POWER-UP Event decoding in the VBAT-Powered Control Interface on page 462 . See Section 2.5.8, "Power-Up Event Output," on page 307 for a functional description of the POWER-UP_EN bit. 1=Power-Up Event Output Enabled 0=Power-Up Event Output Disabled and Reset	R/W	00h	VBAT_POR
5	BGPO VBAT-powered General Purpose Output Control that is used as part of the VBAT-Powered Control Interface. 1=Output high 0=Output low	R/W	00h	VBAT_POR
4:1	Reserved	R	-	-
0	WT_ENABILE The WT_ENABLE bit is used to start and stop the Week Alarm Counter Register and the Clock Divider Register. The value in the Counter Register is held when the WT_ENABLE bit is not asserted (‘0’) and the count is resumed from the last value when the bit is asserted (‘1’). The 15-Bit Clock Divider is reset to 00h and the RTC/Week Alarm Interface is in its lowest power consumption state when the WT_ENABLE bit is not asserted.	R/W	00h	VBAT_POR

21.10.2 WEEK ALARM COUNTER REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	Reserved	R	-	-
27:0	WEEK_COUNTER While the WT_ENABLE bit is ‘1’, this register is incremented at a 1 Hz rate. Writes of this register may require one second to take effect. Reads return the current state of the register. Reads and writes complete independently of the state of WT_ENABLE.	R/W	00h	VBAT_POR

21.10.3 WEEK TIMER COMPARE REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	Reserved	R	-	-
27:0	WEEK_COMPARE A Week Alarm Interrupt and a Week Alarm Power-Up Event are asserted when the Week Alarm Counter Register is greater than or equal to the contents of this register. Reads and writes complete independently of the state of WT_ENABLE.	R/W	FFFFFFFh	VBAT_POR

21.10.4 CLOCK DIVIDER REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	R	-	-
14:0	CLOCK_DIVIDER Reads of this register return the current state of the Week Timer 15-bit clock divider.	R	-	VBAT_POR

21.10.5 SUB-SECOND PROGRAMMABLE INTERRUPT SELECT REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	R	-	-
3:0	SPISR This field determines the rate at which Sub-Second interrupt events are generated. Table 21-10, "SPISR Encoding" shows the relation between the SPISR encoding and Sub-Second interrupt rate.	R/W	00h	VBAT_POR

TABLE 21-10: SPISR ENCODING

SPISR Value	Sub-Second Interrupt Rate, Hz	Interrupt Period
0	Interrupts disabled	
1	2	500 ms
2	4	250 ms
3	8	125 ms
4	16	62.5 ms
5	32	31.25 ms
6	64	15.63 ms
7	128	7.813 ms
8	256	3.906 ms
9	512	1.953 ms
10	1024	977 µS
11	2048	488 µS
12	4096	244 µS
13	8192	122 µS
14	16384	61 µS
15	32768	30.5 µS

21.10.6 SUB-WEEK CONTROL REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:10	Reserved	R	-	-
9:7	SUBWEEK_TICK This field selects the clock source for the Sub-Week Counter. See Table 21-7, "Sub-Week Alarm Counter Clock" for the description of the options for this field. See also Note 1.	R/W	0	VBAT_POR
6	AUTO_RELOAD 1= No reload occurs when the Sub-Week Counter expires 0= Reloads the SUBWEEK_COUNTER_LOAD field into the Sub-Week Counter when the counter expires.	R/W	0	VBAT_POR
5	SYSPWR_PRES_ENABLE Enables SYSPWR_PRES Pin to disable Week the Week timer and Sub-Week Timer Power-Up Events from driving VCI_OUT high 1=The SYSPWR_PRES Pin input low disables both the Week timer and Sub-Week Timer Power-Up Events from driving VCI_OUT high 0=The SYSPWR_PRES Pin input has no effect on the Week timer and Sub-Week Timer Power-Up Events driving VCI_OUT high	R/W	0	VBAT_POR

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
4	SYSPWR_PRES_STATUS Current status of the SYSPWR_PRES pin.	R	-	VBAT_POR
53:2	Reserved	R	-	-
1	WEEK_TIMER_POWERUP_EVENT_STATUS This bit is set to '1' when the Week Alarm Counter Register is greater than or equal the contents of the Week Timer Compare Register and the POWERUP_EN is '1'. Writes of '1' clear this bit. Writes of '0' have no effect. **Note:** This bit does not have to be cleared to remove a Week Timer Power-Up Event.	R/WC	0	VBAT_POR
0	SUBWEEK_TIMER_POWERUP_EVENT_STATUS This bit is set to '1' when the Sub-Week Alarm Counter Register decrements from '1' to '0' and the POWERUP_EN is '1'. Writes of '1' clear this bit. Writes of '0' have no effect. **Note:** This bit MUST be cleared to remove a Sub-Week Timer Power-Up Event.	R/WC	0	VBAT_POR

21.10.7 SUB-WEEK ALARM COUNTER REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:25	Reserved	R	-	-
24:16	SUBWEEK_COUNTER_STATUS Reads of this register return the current state of the 9-bit Sub-Week Alarm counter.	R	00h	VBAT_POR
15:9	Reserved	R	-	-
8:0	SUBWEEK_COUNTER_LOAD Writes with a non-zero value to this field reload the 9-bit Sub-Week Alarm counter. Writes of 0 disable the counter. If the Sub-Week Alarm counter decrements to 0 and the AUTO_RELOAD bit is set, the value in this field is automatically loaded into the Sub-Week Alarm counter.	R/W	00h	VBAT_POR

22.0 GPIO INTERFACE

22.1 General Description

The MEC140x/1x GPIO Interface provides general purpose input monitoring and output control, as well as managing many aspects of pin functionality; including, multi-function Pin Multiplexing Control, GPIO Direction control, PU/PD (PU_PD) resistors, asynchronous wakeup and synchronous Interrupt Detection (int_det), GPIO Direction, and Polarity control, as well as control of pin drive strength and slew rate.

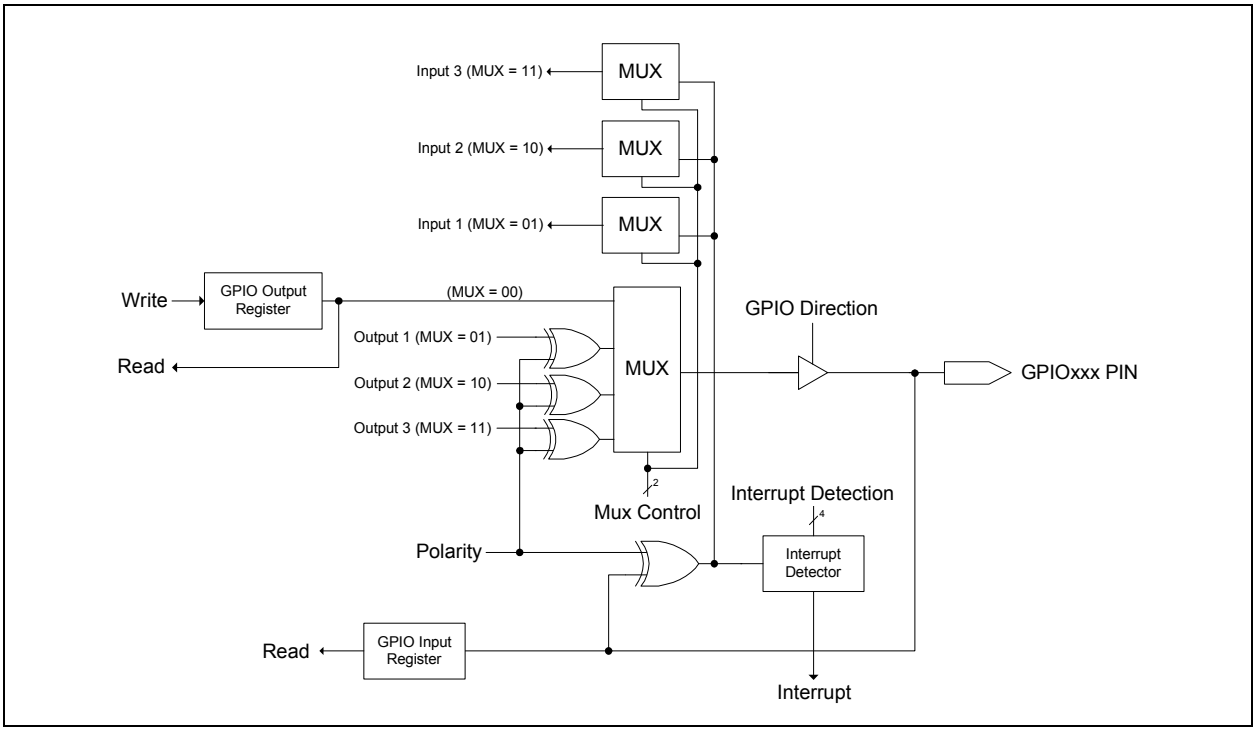
Features of the GPIO Interface include:

- Inputs:
  - Asynchronous rising and falling edge wakeup detection
  - Interrupt High or Low Level
- On Output:
  - Push Pull or Open Drain output
- Pull up or pull down resistor control
- Interrupt and wake capability available for all GPIOs
- Programmable pin drive strength and slew rate limiting
- Group- or individual control of GPIO data.
- Multiplexing of all multi-function pins are controlled by the GPIO interface

22.2 Block Diagram

The GPIO Interface Block Diagram shown in FIGURE 22-1: illustrates the functionality of a single MEC140x/1x GPIO Interface pin. The source for the Pin Multiplexing Control, Interrupt Detection (int_det), GPIO Direction, and Polarity controls in FIGURE 22-1: is a Pin Control Register that is associated with each pin (see Section 22.6.1.1, "Pin Control Register," on page 329).

FIGURE 22-1: GPIO INTERFACE BLOCK DIAGRAM



22.3 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

22.3.1 POWER DOMAINS

Name	Description
VTR	The registers and logic in this block are powered by VTR.

22.3.2 CLOCK INPUTS

Name	Description
48 MHz Ring Oscillator	The 48 MHz Ring Oscillator is used for synchronizing the GPIO inputs.

22.3.3 RESETS

Name	Description
nSYSRST	This reset is asserted when VTR is applied.
nSIO_RESET	This is an alternate reset condition, typically asserted when the main power rail is asserted. This reset is used for VCC Power Well Emulation.

22.4 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
GPIO_Event	Each pin in the GPIO Interface has the ability to generate an interrupt event. This event may be used as a wake event. The GPIO Interface can generate an interrupt source event on a high level, low level, rising edge and falling edge, as configured by the Interrupt Detection (int_det) bits in the Pin Control Register associated with the GPIO signal function. **Note:** The minimum pulse width required to generate an interrupt/wakeup event is 5ns.

22.5 Description

The GPIO Interface refers to all the GPIOxxx pins implemented in the design. GPIO stands for General Purpose I/O.

The GPIO signals may be used by firmware to both monitor and control a pin in “bit-banged” mode. The GPIOs may be individually controlled via their Pin Control Register or group controlled via the Output and Input GPIO registers. The GPIO Output Control Select

The GPIO Pin control registers are used to select the alternate functions on GPIO pins (unless otherwise specified), to control the buffer direction, strength, and polarity, to control the internal pull-ups and pull-downs, for VCC emulation, and for selecting the event type that causes a GPIO interrupt.

The GPIO input is always live, even when an alternate function is selected. Firmware may read the GPIO input anytime to see the value on the pin. In addition, the GPIO interrupt is always functional, and may be used for either the GPIO itself or to support the alternate functions on the pin. See FIGURE 22-1: GPIO Interface Block Diagram on page 313.

22.5.1 ACCESSING GPIOS

There are two ways to access GPIO output data. Bit [10] is used to determine which GPIO output data bit affects the GPIO output pin.

- Grouped Output GPIO Data
  - Outputs to individual GPIO ports are grouped into 32-bit GPIO Output Registers.
- Individual GPIO output data
  - Alternatively, each GPIO output port is individually accessible via Bit [16] in the port's Pin Control Register. On reads, Bit [16] returns the programmed value, not the value on the pin.

There are two ways to access GPIO input data.

- Input GPIO Data
  - Inputs from individual GPIO ports are grouped into 32-bit GPIO Input Registers and always reflect the current state of the GPIO input from the pad.
- GPIO input from pad
  - Alternatively, each GPIO input port is individually accessible via Bit [24] in the port's Pin Control Register. Bit [24] always reflects the current state of GPIO input from the pad.

22.5.2 GPIO INDEXING

Each GPIO signal function name consists of a 4-character prefix ("GPIO") followed by a 3-digit octal-encoded index number. In the MEC140x/1x GPIO indexing is done sequentially starting from 'GPIO000.'

22.5.3 GPIO MULTIPLEXING

The GPIO Mux Control bits located in the Pin Control Register are used to support up to three alternate functions on any GPIO pin. The following tables define all the GPIO Multiplexing Options implemented for each of the MEC140x/1x products.

22.5.3.1 MEC140x GPIO Multiplexing Options

MEC140x				
GPIO Name (Octal)	**Mux Control = 00**	**Mux Control = 01**	**Mux Control = 10**	**Mux Control = 11**
GPIO001	GPIO001	SPI_CS#	32KHZ_OUT	Reserved
GPIO002	GPIO002	PWM7	Reserved	Reserved
GPIO003	GPIO003	SYSPWR_PRES	Reserved	Reserved
GPIO004	GPIO004	BGPO	Reserved	Reserved
GPIO005	GPIO005	SMB00_DATA	SMB00_DATA18	KSI2
GPIO006	GPIO006	SMB00_CLK	SMB00_CLK18	KSI3
GPIO007	GPIO007	SMB01_DATA	SMB01_DATA18	Reserved
GPIO010	GPIO010	SMB01_CLK	SMB01_CLK18	Reserved
GPIO011	GPIO011	nSMI	nEMI_INT	Reserved
GPIO012	GPIO012	SMB02_DATA	SMB02_DATA18	Reserved
GPIO013	GPIO013	SMB02_CLK	SMB02_CLK18	Reserved
GPIO014	GPIO014	nRESET_IN	Reserved	Reserved
GPIO015	GPIO015	KSO01	PVT_CS#	Reserved
GPIO016	GPIO016	KSO02	PVT_SCLK	Reserved
GPIO017	GPIO017	KSO03	PVT_IO0	Reserved
GPIO020	GPIO020	CMP_VIN0	Reserved	Reserved
GPIO021	GPIO021	CMP_VIN1	Reserved	Reserved
GPIO022	GPIO022	ADC5	Reserved	Reserved
GPIO023	GPIO023	ADC6	A20M	Reserved
GPIO024	GPIO024	ADC7	Reserved	Reserved

MEC140x				
GPIO Name (Octal)	**Mux Control = 00**	**Mux Control = 01**	**Mux Control = 10**	**Mux Control = 11**
GPIO025	GPIO025	KSO07	PVT_IO2	Reserved
GPIO026	GPIO026	PS2_CLK1B	Reserved	Reserved
GPIO027	GPIO027	KSO00	PVT_IO1	Reserved
GPIO030	GPIO030	BCM_INT0#	PWM4	Reserved
GPIO031	GPIO031	BCM_DAT0	PWM5	Reserved
GPIO032	GPIO032	BCM_CLK0	PWM6	Reserved
GPIO033	GPIO033	PECI_DAT	SB_TSI_DAT	Reserved
GPIO034	GPIO034	PCI_CLK	Reserved	Reserved
GPIO035	GPIO035	Reserved	SB-TSI_CLK	Reserved
GPIO036	GPIO036	VCI_OUT	Reserved	Reserved
GPIO040	GPIO040	LAD0	Reserved	Reserved
GPIO041	GPIO041	LAD1	Reserved	Reserved
GPIO042	GPIO042	LAD2	Reserved	Reserved
GPIO043	GPIO043	LAD3	Reserved	Reserved
GPIO044	GPIO044	LFRAME#	Reserved	Reserved
GPIO045	GPIO045	BCM_INT1#	KSO04	Reserved
GPIO046	GPIO046	BCM_DAT1	KSO05	Reserved
GPIO047	GPIO047	BCM_CLK1	KSO06	Reserved
GPIO050	GPIO050	TACH0	Reserved	Reserved
GPIO051	GPIO051	TACH1	Reserved	Reserved
GPIO052	GPIO052	SPI_IO2	Reserved	Reserved
GPIO053	GPIO053	PWM0	Reserved	Reserved
GPIO054	GPIO054	PWM1	Reserved	Reserved
GPIO055	GPIO055	PWM2	KSO08	PVT_IO3
GPIO056	GPIO056	PWM3	Reserved	Reserved
GPIO057	GPIO057	VCC_PWRGD	Reserved	Reserved
GPIO060	GPIO060	KBRST	Reserved	Reserved
GPIO061	GPIO061	LPCPD#	Reserved	Reserved
GPIO062	GPIO062	SPI_IO3	Reserved	Reserved
GPIO063	GPIO063	SER_IRQ	Reserved	Reserved
GPIO064	GPIO064	LRESET#	Reserved	Reserved
(GPIO065)	Reserved	ADC_VREF	Reserved	Reserved
(GPIO066)	Reserved	DAC_VREF	Reserved	Reserved
GPIO067	GPIO067	CLKRUN#	Reserved	Reserved
GPIO100	GPIO100	nEC_SCI	Reserved	Reserved
GPIO101	GPIO101	SPI_CLK	Reserved	Reserved
GPIO102	GPIO102	KSO09	Reserved	Reserved
GPIO103	GPIO103	SPI_IO0	Reserved	Reserved
GPIO104	GPIO104	LED2	Reserved	Reserved
GPIO105	GPIO105	SPI_IO1	Reserved	Reserved
GPIO106	GPIO106	KSO10	Reserved	Reserved
GPIO107	GPIO107	nRESET_OUT	Reserved	Reserved
GPIO110	GPIO110	KSO11	Reserved	Reserved
GPIO111	GPIO111	KSO12	Reserved	Reserved

MEC140x				
GPIO Name (Octal)	**Mux Control = 00**	**Mux Control = 01**	**Mux Control = 10**	**Mux Control = 11**
GPIO112	GPIO112	PS2_CLK1A	KSO13	Reserved
GPIO113	GPIO113	PS2_DAT1A	KSO14	Reserved
GPIO114	GPIO114	PS2_CLK0	Reserved	Reserved
GPIO115	GPIO115	PS2_DAT0	Reserved	Reserved
GPIO116	GPIO116	TFDP_DATA	UART_RX	Reserved
GPIO117	GPIO117	TFDP_CLK	UART_TX	Reserved
GPIO120	GPIO120	CMP_VOUT1	Reserved	Reserved
GPIO121	GPIO121	ADC0	Reserved	Reserved
GPIO122	GPIO122	ADC1	Reserved	Reserved
GPIO123	GPIO123	SHD_CS#	Reserved	Reserved
GPIO124	GPIO124	CMP_VOUT0	Reserved	Reserved
GPIO125	GPIO125	KSO15	Reserved	Reserved
GPIO126	GPIO126	SHD_SCLK	Reserved	Reserved
GPIO127	GPIO127	PS2_DAT1B	Reserved	Reserved
GPIO130	GPIO130	SMB03_DATA	SMB03_DATA18	Reserved
GPIO131	GPIO131	SMB03_CLK	SMB03_CLK18	Reserved
GPIO132	GPIO132	KSO16	Reserved	Reserved
GPIO133	GPIO133	SHD_IO0	Reserved	Reserved
GPIO134	GPIO134	SHD_IO1	Reserved	Reserved
GPIO135	GPIO135	SHD_IO2	Reserved	Reserved
GPIO136	GPIO136	SHD_IO3	Reserved	Reserved
GPIO140	GPIO140	KSO17	Reserved	Reserved
GPIO141	GPIO141	SMB04_DATA	SMB04_DATA18	Reserved
GPIO142	GPIO142	SMB04_CLK	SMB04_CLK18	Reserved
GPIO143	GPIO143	KSI0	DTR#	Reserved
GPIO144	GPIO144	KSI1	DCD#	Reserved
GPIO145	GPIO145	Reserved	Reserved	Reserved
GPIO146	GPIO146	Reserved	Reserved	Reserved
GPIO147	GPIO147	KSI4	DSR#	Reserved
GPIO150	GPIO150	KSI5	RI#	Reserved
GPIO151	GPIO151	KSI6	RTS#	Reserved
GPIO152	GPIO152	KSI7	CTS#	Reserved
GPIO153	GPIO153	ADC4	Reserved	Reserved
GPIO154	GPIO154	ADC3	Reserved	Reserved
GPIO155	GPIO155	ADC2	Reserved	Reserved
GPIO156	GPIO156	LED1	Reserved	Reserved
GPIO157	GPIO157	LED0	TST_CLK_OUT	Reserved
GPIO160	GPIO160	DAC_0	Reserved	Reserved
GPIO161	GPIO161	DAC_1	Reserved	Reserved
GPIO162	GPIO162	VCI_IN1#	Reserved	Reserved
GPIO163	GPIO163	VCI_IN0#	Reserved	Reserved
GPIO164	GPIO164	VCI_OVRD_IN	Reserved	Reserved
GPIO165	GPIO165	CMP_VREF0	Reserved	Reserved
GPIO166	GPIO166	CMP_VREF1	UART_CLK	Reserved

22.5.3.2 MEC141x GPIO Multiplexing Options

MEC141x				
GPIO Name (Octal)	**Mux Control = 00**	**Mux Control = 01**	**Mux Control = 10**	**Mux Control = 11**
GPIO001	GPIO001	SPI_CS#	32KHZ_OUT	Reserved
GPIO002	GPIO002	PWM7	Reserved	Reserved
GPIO003	GPIO003	SYSPWR_PRES	Reserved	Reserved
GPIO004	GPIO004	BGPO	Reserved	Reserved
GPIO005	GPIO005	SMB00_DATA	SMB00_DATA18	KSI2
GPIO006	GPIO006	SMB00_CLK	SMB00_CLK18	KSI3
GPIO007	GPIO007	SMB01_DATA	SMB01_DATA18	Reserved
GPIO010	GPIO010	SMB01_CLK	SMB01_CLK18	Reserved
GPIO011	GPIO011	nSMI	nEMI_INT	Reserved
GPIO012	GPIO012	SMB02_DATA	SMB02_DATA18	Reserved
GPIO013	GPIO013	SMB02_CLK	SMB02_CLK18	Reserved
GPIO014	GPIO014	nRESET_IN	Reserved	Reserved
GPIO015	GPIO015	KSO01	PVT_CS#	Reserved
GPIO016	GPIO016	KSO02	PVT_SCLK	Reserved
GPIO017	GPIO017	KSO03	PVT_IO0	Reserved
GPIO020	GPIO020	CMP_VIN0	Reserved	Reserved
GPIO021	GPIO021	CMP_VIN1	Reserved	Reserved
GPIO022	GPIO022	ADC5	Reserved	Reserved
GPIO023	GPIO023	ADC6	A20M	Reserved
GPIO024	GPIO024	ADC7	Reserved	Reserved
GPIO025	GPIO025	KSO07	PVT_IO2	Reserved
GPIO026	GPIO026	PS2_CLK1B	Reserved	Reserved
GPIO027	GPIO027	KSO00	PVT_IO1	Reserved
GPIO030	GPIO030	BCM_INT0#	PWM4	Reserved
GPIO031	GPIO031	BCM_DAT0	PWM5	Reserved
GPIO032	GPIO032	BCM_CLK0	PWM6	Reserved
GPIO033	GPIO033	PECI_DAT	SB_TSI_DAT	Reserved
GPIO034	GPIO034	PCI_CLK	ESPI_CLK	Reserved
GPIO035	GPIO035	Reserved	SB-TSI_CLK	Reserved
GPIO036	GPIO036	VCI_OUT	Reserved	Reserved
GPIO040	GPIO040	LAD0	ESPI_IO0	Reserved
GPIO041	GPIO041	LAD1	ESPI_IO1	Reserved
GPIO042	GPIO042	LAD2	ESPI_IO2	Reserved
GPIO043	GPIO043	LAD3	ESPI_IO3	Reserved
GPIO044	GPIO044	LFRAME#	ESPI_CS#	Reserved
GPIO045	GPIO045	BCM_INT1#	KSO04	Reserved
GPIO046	GPIO046	BCM_DAT1	KSO05	Reserved
GPIO047	GPIO047	BCM_CLK1	KSO06	Reserved
GPIO050	GPIO050	TACH0	Reserved	Reserved
GPIO051	GPIO051	TACH1	Reserved	Reserved

MEC141x				
GPIO Name (Octal)	**Mux Control = 00**	**Mux Control = 01**	**Mux Control = 10**	**Mux Control = 11**
GPIO052	GPIO052	SPI_IO2	Reserved	Reserved
GPIO053	GPIO053	PWM0	Reserved	Reserved
GPIO054	GPIO054	PWM1	Reserved	Reserved
GPIO055	GPIO055	PWM2	KSO08	PVT_IO3
GPIO056	GPIO056	PWM3	Reserved	Reserved
GPIO057	GPIO057	VCC_PWRGD	Reserved	Reserved
GPIO060	GPIO060	KBRST	Reserved	Reserved
GPIO061	GPIO061	LPCPD#	ESPI_RESET#	Reserved
GPIO062	GPIO062	SPI_IO3	Reserved	Reserved
GPIO063	GPIO063	SER_IRQ	ESPI_ALERT#	Reserved
GPIO064	GPIO064	LRESET#	Reserved	Reserved
(GPIO065)	Reserved	ADC_VREF	Reserved	Reserved
(GPIO066)	Reserved	DAC_VREF	Reserved	Reserved
GPIO067	GPIO067	CLKRUN#	Reserved	Reserved
GPIO100	GPIO100	nEC_SCI	Reserved	Reserved
GPIO101	GPIO101	SPI_CLK	Reserved	Reserved
GPIO102	GPIO102	KSO09	Reserved	Reserved
GPIO103	GPIO103	SPI_IO0	Reserved	Reserved
GPIO104	GPIO104	LED2	Reserved	Reserved
GPIO105	GPIO105	SPI_IO1	Reserved	Reserved
GPIO106	GPIO106	KSO10	Reserved	Reserved
GPIO107	GPIO107	nRESET_OUT	Reserved	Reserved
GPIO110	GPIO110	KSO11	Reserved	Reserved
GPIO111	GPIO111	KSO12	Reserved	Reserved
GPIO112	GPIO112	PS2_CLK1A	KSO13	Reserved
GPIO113	GPIO113	PS2_DAT1A	KSO14	Reserved
GPIO114	GPIO114	PS2_CLK0	Reserved	Reserved
GPIO115	GPIO115	PS2_DAT0	Reserved	Reserved
GPIO116	GPIO116	TFDP_DATA	UART_RX	Reserved
GPIO117	GPIO117	TFDP_CLK	UART_TX	Reserved
GPIO120	GPIO120	CMP_VOUT1	Reserved	Reserved
GPIO121	GPIO121	ADC0	Reserved	Reserved
GPIO122	GPIO122	ADC1	Reserved	Reserved
GPIO123	GPIO123	SHD_CS#	Reserved	Reserved
GPIO124	GPIO124	CMP_VOUT0	Reserved	Reserved
GPIO125	GPIO125	KSO15	Reserved	Reserved
GPIO126	GPIO126	SHD_SCLK	Reserved	Reserved
GPIO127	GPIO127	PS2_DAT1B	Reserved	Reserved
GPIO130	GPIO130	SMB03_DATA	SMB03_DATA18	Reserved
GPIO131	GPIO131	SMB03_CLK	SMB03_CLK18	Reserved
GPIO132	GPIO132	KSO16	Reserved	Reserved
GPIO133	GPIO133	SHD_IO0	Reserved	Reserved
GPIO134	GPIO134	SHD_IO1	Reserved	Reserved
GPIO135	GPIO135	SHD_IO2	Reserved	Reserved

MEC141x				
GPIO Name (Octal)	**Mux Control = 00**	**Mux Control = 01**	**Mux Control = 10**	**Mux Control = 11**
GPIO136	GPIO136	SHD_IO3	Reserved	Reserved
GPIO140	GPIO140	KSO17	Reserved	Reserved
GPIO141	GPIO141	SMB04_DATA	SMB04_DATA18	Reserved
GPIO142	GPIO142	SMB04_CLK	SMB04_CLK18	Reserved
GPIO143	GPIO143	KSI0	DTR#	Reserved
GPIO144	GPIO144	KSI1	DCD#	Reserved
GPIO145	GPIO145	Reserved	Reserved	Reserved
GPIO146	GPIO146	Reserved	Reserved	Reserved
GPIO147	GPIO147	KSI4	DSR#	Reserved
GPIO150	GPIO150	KSI5	RI#	Reserved
GPIO151	GPIO151	KSI6	RTS#	Reserved
GPIO152	GPIO152	KSI7	CTS#	Reserved
GPIO153	GPIO153	ADC4	Reserved	Reserved
GPIO154	GPIO154	ADC3	Reserved	Reserved
GPIO155	GPIO155	ADC2	Reserved	Reserved
GPIO156	GPIO156	LED1	Reserved	Reserved
GPIO157	GPIO157	LED0	TST_CLK_OUT	Reserved
GPIO160	GPIO160	DAC_0	Reserved	Reserved
GPIO161	GPIO161	DAC_1	Reserved	Reserved
GPIO162	GPIO162	VCI_IN1#	Reserved	Reserved
GPIO163	GPIO163	VCI_IN0#	Reserved	Reserved
GPIO164	GPIO164	VCI_OVRD_IN	Reserved	Reserved
GPIO165	GPIO165	CMP_VREF0	Reserved	Reserved
GPIO166	GPIO166	CMP_VREF1	UART_CLK	Reserved

22.5.4 PIN CONTROL REGISTERS

Each GPIO has two Pin Control registers. The Pin Control Register, which is the primary register, is used to read the value of the input data and set the output either high or low. It is used to select the alternate function via the Mux Control bits, set the Polarity of the input, configure and enable the output buffer, configure the GPIO interrupt event source, enable internal pull-up/pull-down resistors, and to enable VCC Emulation via the Power Gating Signals control bits. The Pin Control Register 2 is used to configure the output buffer drive strength and slew rate.

The following tables define the default settings for the two Pin Control registers for each GPIO in each product group.

22.5.4.1 MEC140x Pin Control Registers Defaults

MEC140x						
GPIO Name (Octal)	**Pin Control Register Offset (Hex)**	**Pin Control Register Default (Hex)**	**Default Function**	**Pin Control Register 2 Offset (Hex)**	**Pin Control Register 2 Default (Hex)**	**Default Drive Strength (mA)**
GPIO001	0004	00000000	GPIO001	504	00000010	4
GPIO002	0008	00000000	GPIO002	508	00000010	4
GPIO003	000C	00001000	SYS-PWR_PRES	50C	00000010	4

MEC140x						
GPIO Name (Octal)	**Pin Control Register Offset (Hex)**	**Pin Control Register Default (Hex)**	**Default Function**	**Pin Control Register 2 Offset (Hex)**	**Pin Control Register 2 Default (Hex)**	**Default Drive Strength (mA)**
GPIO004	0010	00001000	BGPO	510	00000010	4
GPIO005	0014	00000000	GPIO005	514	00000010	4
GPIO006	0018	00000000	GPIO006	518	00000010	4
GPIO007	001C	00000000	GPIO007	51C	00000010	4
GPIO010	0020	00000000	GPIO010	520	00000010	4
GPIO011	0024	00000000	GPIO011	524	00000010	4
GPIO012	0028	00000000	GPIO012	528	00000010	4
GPIO013	002C	00000000	GPIO013	52C	00000010	4
GPIO014	0030	00001000	nRESET_IN	530	00000010	4
GPIO015	0034	00000000	GPIO015	534	00000010	4
GPIO016	0038	00000000	GPIO016	538	00000010	4
GPIO017	003C	00000000	GPIO017	53C	00000010	4
GPIO020	0040	00000000	GPIO020	540	00000010	4
GPIO021	0044	00000000	GPIO021	544	00000010	4
GPIO022	0048	00000000	GPIO022	548	00000000	2
GPIO023	004C	00000000	GPIO023	54C	00000000	2
GPIO024	0050	00000000	GPIO024	550	00000000	2
GPIO025	0054	00000000	GPIO025	554	00000010	4
GPIO026	0058	00000000	GPIO026	558	00000010	4
GPIO027	005C	00000000	GPIO027	55C	00000010	4
GPIO030	0060	00000000	GPIO030	560	00000010	4
GPIO031	0064	00000000	GPIO031	564	00000010	4
GPIO032	0068	00000000	GPIO032	568	00000010	4
GPIO033	006C	00000000	GPIO033	56C	00000010	4
GPIO034	0070	00000000	GPIO034	570	00000010	4
GPIO035	0074	00000000	GPIO035	574	00000010	4
GPIO036	0078	00001000	VCI_OUT	578	00000020	8
GPIO040	0080	00000000	GPIO040	580	00000010	4
GPIO041	0084	00000000	GPIO041	584	00000010	4
GPIO042	0088	00000000	GPIO042	588	00000010	4
GPIO043	008C	00000000	GPIO043	58C	00000010	4
GPIO044	0090	00000000	GPIO044	590	00000010	4
GPIO045	0094	00000000	GPIO045	594	00000010	4
GPIO046	0098	00000000	GPIO046	598	00000010	4
GPIO047	009C	00000000	GPIO047	59C	00000010	4
GPIO050	00A0	00000000	GPIO050	5A0	00000010	4
GPIO051	00A4	00000000	GPIO051	5A4	00000010	4
GPIO052	00A8	00000000	GPIO052	5A8	00000010	4
GPIO053	00AC	00000000	GPIO053	5AC	00000010	4
GPIO054	00B0	00000000	GPIO054	5B0	00000010	4
GPIO055	00B4	00000000	GPIO055	5B4	00000010	4
GPIO056	00B8	00000000	GPIO056	5B8	00000010	4

MEC140x						
GPIO Name (Octal)	**Pin Control Register Offset (Hex)**	**Pin Control Register Default (Hex)**	**Default Function**	**Pin Control Register 2 Offset (Hex)**	**Pin Control Register 2 Default (Hex)**	**Default Drive Strength (mA)**
GPIO057	00BC	00000000	GPIO057	5BC	00000010	4
GPIO060	00C0	00000000	GPIO060	5C0	00000010	4
GPIO061	00C4	00000000	GPIO061	5C4	00000010	4
GPIO062	00C8	00000000	GPIO062	5C8	00000010	4
GPIO063	00CC	00000000	GPIO063	5CC	00000010	4
GPIO064	00D0	00000000	GPIO064	5D0	00000010	4
(GPIO065)	00D4	00001000	ADC_VREF	5D4	00000000	Reserved
(GPIO066)	00D8	00001000	DAC_VREF	5D8	00000010	Reserved
GPIO067	00DC	00000000	GPIO067	5DC	00000010	4
GPIO100	0100	00000000	GPIO100	5E0	00000010	4
GPIO101	0104	00000000	GPIO101	5E4	00000010	4
GPIO102	0108	00000000	GPIO102	5E8	00000010	4
GPIO103	010C	00000000	GPIO103	5EC	00000010	4
GPIO104	0110	00000000	GPIO104	5F0	00000010	4
GPIO105	0114	00000000	GPIO105	5F4	00000010	4
GPIO106	0118	00000000	GPIO106	5F8	00000010	4
GPIO107	011C	00000000	GPIO107	5FC	00000010	4
GPIO110	0120	00000000	GPIO110	600	00000010	4
GPIO111	0124	00000000	GPIO111	604	00000010	4
GPIO112	0128	00000000	GPIO112	608	00000010	4
GPIO113	012C	00000000	GPIO113	60C	00000010	4
GPIO114	0130	00000000	GPIO114	610	00000010	4
GPIO115	0134	00000000	GPIO115	614	00000010	4
GPIO116	0138	00000000	GPIO116	618	00000010	4
GPIO117	013C	00000000	GPIO117	61C	00000010	4
GPIO120	0140	00000000	GPIO120	620	00000010	4
GPIO121	0144	00000000	GPIO121	624	00000000	2
GPIO122	0148	00000000	GPIO122	628	00000000	2
GPIO123	014C	00000000	GPIO123	62C	00000010	4
GPIO124	0150	00000000	GPIO124	630	00000010	4
GPIO125	0154	00000000	GPIO125	634	00000010	4
GPIO126	0158	00000000	GPIO126	638	00000010	4
GPIO127	015C	00000000	GPIO127	63C	00000010	4
GPIO130	0160	00000000	GPIO130	640	00000010	4
GPIO131	0164	00000000	GPIO131	644	00000010	4
GPIO132	0168	00000000	GPIO132	648	00000010	4
GPIO133	016C	00000000	GPIO133	64C	00000010	4
GPIO134	0170	00000000	GPIO134	650	00000010	4
GPIO135	0174	00000000	GPIO135	654	00000010	4
GPIO136	0178	00000000	GPIO136	658	00000010	4
GPIO140	0180	00000000	GPIO140	660	00000010	4
GPIO141	0184	00000000	GPIO141	664	00000010	4

MEC140x						
GPIO Name (Octal)	**Pin Control Register Offset (Hex)**	**Pin Control Register Default (Hex)**	**Default Function**	**Pin Control Register 2 Offset (Hex)**	**Pin Control Register 2 Default (Hex)**	**Default Drive Strength (mA)**
GPIO142	0188	00000000	GPIO142	668	00000010	4
GPIO143	018C	00000000	GPIO143	66C	00000010	4
GPIO144	0190	00000000	GPIO144	670	00000010	4
GPIO145	0194	00000000	GPIO145	674	00000010	4
GPIO146	0198	00000000	GPIO146	678	00000010	4
GPIO147	019C	00000000	GPIO147	67C	00000010	4
GPIO150	01A0	00000000	GPIO150	680	00000010	4
GPIO151	01A4	00000000	GPIO151	684	00000010	4
GPIO152	01A8	00000000	GPIO152	688	00000010	4
GPIO153	01AC	00000000	GPIO153	68C	00000000	2
GPIO154	01B0	00000000	GPIO154	690	00000000	2
GPIO155	01B4	00000000	GPIO155	694	00000000	2
GPIO156	01B8	00000000	GPIO156	698	00000010	4
GPIO157	01BC	00000000	GPIO157	69C	00000010	4
GPIO160	01C0	00000000	GPIO160	6A0	00000010	4
GPIO161	01C4	00000000	GPIO161	6A4	00000010	4
GPIO162	01C8	00001000	VCI_IN1#	6A8	00000010	4
GPIO163	01CC	00001000	VCI_IN0#	6AC	00000010	4
GPIO164	01D0	00001000	VCI_OVRD_IN	6B0	00000010	4
GPIO165	01D4	00000000	GPIO165	6B4	00000010	4
GPIO166	01D8	00000000	GPIO166	6B8	00000010	4

22.5.4.2 MEC141x Pin Control Registers Defaults

MEC141x						
GPIO Name (Octal)	**Pin Control Register Offset (Hex)**	**Pin Control Register Default (Hex)**	**Default Function**	**Pin Control Register 2 Offset (Hex)**	**Pin Control Register 2 Default (Hex)**	**Default Drive Strength (mA)**
GPIO001	0004	00000000	GPIO001	504	00000010	4
GPIO002	0008	00000000	GPIO002	508	00000010	4
GPIO003	000C	00001000	SYS-PWR_PRES	50C	00000010	4
GPIO004	0010	00001000	BGPO	510	00000020	8
GPIO005	0014	00000000	GPIO005	514	00000010	4
GPIO006	0018	00000000	GPIO006	518	00000010	4
GPIO007	001C	00000000	GPIO007	51C	00000010	4
GPIO010	0020	00000000	GPIO010	520	00000010	4
GPIO011	0024	00000000	GPIO011	524	00000010	4
GPIO012	0028	00000000	GPIO012	528	00000010	4
GPIO013	002C	00000000	GPIO013	52C	00000010	4

MEC141x						
GPIO Name (Octal)	**Pin Control Register Offset (Hex)**	**Pin Control Register Default (Hex)**	**Default Function**	**Pin Control Register 2 Offset (Hex)**	**Pin Control Register 2 Default (Hex)**	**Default Drive Strength (mA)**
GPIO014	0030	00001000	nRESET_IN	530	00000010	4
GPIO015	0034	00000000	GPIO015	534	00000010	4
GPIO016	0038	00000000	GPIO016	538	00000010	4
GPIO017	003C	00000000	GPIO017	53C	00000010	4
GPIO020	0040	00000000	GPIO020	540	00000010	4
GPIO021	0044	00000000	GPIO021	544	00000010	4
GPIO022	0048	00000000	GPIO022	548	00000000	2
GPIO023	004C	00000000	GPIO023	54C	00000000	2
GPIO024	0050	00000000	GPIO024	550	00000000	2
GPIO025	0054	00000000	GPIO025	554	00000010	4
GPIO026	0058	00000000	GPIO026	558	00000010	4
GPIO027	005C	00000000	GPIO027	55C	00000010	4
GPIO030	0060	00000000	GPIO030	560	00000010	4
GPIO031	0064	00000000	GPIO031	564	00000010	4
GPIO032	0068	00000000	GPIO032	568	00000010	4
GPIO033	006C	00000000	GPIO033	56C	00000010	4
GPIO034	0070	00000000	GPIO034	570	00000010	4
GPIO035	0074	00000000	GPIO035	574	00000010	4
GPIO036	0078	00001000	VCI_OUT	578	00000020	8
GPIO040	0080	00000000	GPIO040	580	00000010	4
GPIO041	0084	00000000	GPIO041	584	00000010	4
GPIO042	0088	00000000	GPIO042	588	00000010	4
GPIO043	008C	00000000	GPIO043	58C	00000010	4
GPIO044	0090	00000000	GPIO044	590	00000010	4
GPIO045	0094	00000000	GPIO045	594	00000010	4
GPIO046	0098	00000000	GPIO046	598	00000010	4
GPIO047	009C	00000000	GPIO047	59C	00000010	4
GPIO050	00A0	00000000	GPIO050	5A0	00000010	4
GPIO051	00A4	00000000	GPIO051	5A4	00000010	4
GPIO052	00A8	00000000	GPIO052	5A8	00000010	4
GPIO053	00AC	00000000	GPIO053	5AC	00000010	4
GPIO054	00B0	00000000	GPIO054	5B0	00000010	4
GPIO055	00B4	00000000	GPIO055	5B4	00000010	4
GPIO056	00B8	00000000	GPIO056	5B8	00000010	4
GPIO057	00BC	00000000	GPIO057	5BC	00000010	4
GPIO060	00C0	00000000	GPIO060	5C0	00000010	4
GPIO061	00C4	00000000	GPIO061	5C4	00000010	4
GPIO062	00C8	00000000	GPIO062	5C8	00000010	4
GPIO063	00CC	00000000	GPIO063	5CC	00000010	4
GPIO064	00D0	00000000	GPIO064	5D0	00000010	4
(GPIO065)	00D4	00001000	ADC_VREF	5D4	00000000	Reserved
(GPIO066)	00D8	00001000	DAC_VREF	5D8	00000010	Reserved

MEC141x						
GPIO Name (Octal)	**Pin Control Register Offset (Hex)**	**Pin Control Register Default (Hex)**	**Default Function**	**Pin Control Register 2 Offset (Hex)**	**Pin Control Register 2 Default (Hex)**	**Default Drive Strength (mA)**
GPIO067	00DC	00000000	GPIO067	5DC	00000010	4
GPIO100	0100	00000000	GPIO100	5E0	00000010	4
GPIO101	0104	00000000	GPIO101	5E4	00000010	4
GPIO102	0108	00000000	GPIO102	5E8	00000010	4
GPIO103	010C	00000000	GPIO103	5EC	00000010	4
GPIO104	0110	00000000	GPIO104	5F0	00000010	4
GPIO105	0114	00000000	GPIO105	5F4	00000010	4
GPIO106	0118	00000000	GPIO106	5F8	00000010	4
GPIO107	011C	00000000	GPIO107	5FC	00000010	4
GPIO110	0120	00000000	GPIO110	600	00000010	4
GPIO111	0124	00000000	GPIO111	604	00000010	4
GPIO112	0128	00000000	GPIO112	608	00000010	4
GPIO113	012C	00000000	GPIO113	60C	00000010	4
GPIO114	0130	00000000	GPIO114	610	00000010	4
GPIO115	0134	00000000	GPIO115	614	00000010	4
GPIO116	0138	00000000	GPIO116	618	00000010	4
GPIO117	013C	00000000	GPIO117	61C	00000010	4
GPIO120	0140	00000000	GPIO120	620	00000010	4
GPIO121	0144	00000000	GPIO121	624	00000000	2
GPIO122	0148	00000000	GPIO122	628	00000000	2
GPIO123	014C	00000000	GPIO123	62C	00000010	4
GPIO124	0150	00000000	GPIO124	630	00000010	4
GPIO125	0154	00000000	GPIO125	634	00000010	4
GPIO126	0158	00000000	GPIO126	638	00000010	4
GPIO127	015C	00000000	GPIO127	63C	00000010	4
GPIO130	0160	00000000	GPIO130	640	00000010	4
GPIO131	0164	00000000	GPIO131	644	00000010	4
GPIO132	0168	00000000	GPIO132	648	00000010	4
GPIO133	016C	00000000	GPIO133	64C	00000010	4
GPIO134	0170	00000000	GPIO134	650	00000010	4
GPIO135	0174	00000000	GPIO135	654	00000010	4
GPIO136	0178	00000000	GPIO136	658	00000010	4
GPIO140	0180	00000000	GPIO140	660	00000010	4
GPIO141	0184	00000000	GPIO141	664	00000010	4
GPIO142	0188	00000000	GPIO142	668	00000010	4
GPIO143	018C	00000000	GPIO143	66C	00000010	4
GPIO144	0190	00000000	GPIO144	670	00000010	4
GPIO145	0194	00000000	GPIO145	674	00000010	4
GPIO146	0198	00000000	GPIO146	678	00000010	4
GPIO147	019C	00000000	GPIO147	67C	00000010	4
GPIO150	01A0	00000000	GPIO150	680	00000010	4
GPIO151	01A4	00000000	GPIO151	684	00000010	4

MEC141x						
GPIO Name (Octal)	Pin Control Register Offset (Hex)	Pin Control Register Default (Hex)	Default Function	Pin Control Register 2 Offset (Hex)	Pin Control Register 2 Default (Hex)	Default Drive Strength (mA)
GPIO152	01A8	00000000	GPIO152	688	00000010	4
GPIO153	01AC	00000000	GPIO153	68C	00000000	2
GPIO154	01B0	00000000	GPIO154	690	00000000	2
GPIO155	01B4	00000000	GPIO155	694	00000000	2
GPIO156	01B8	00000000	GPIO156	698	00000010	4
GPIO157	01BC	00000000	GPIO157	69C	00000010	4
GPIO160	01C0	00000000	GPIO160	6A0	00000010	4
GPIO161	01C4	00000000	GPIO161	6A4	00000010	4
GPIO162	01C8	00001000	VCI_IN1#	6A8	00000010	4
GPIO163	01CC	00001000	VCI_IN0#	6AC	00000010	4
GPIO164	01D0	00001000	VCI_OVRD_IN	6B0	00000010	4
GPIO165	01D4	00000000	GPIO165	6B4	00000010	4
GPIO166	01D8	00000000	GPIO166	6B8	00000010	4

22.6 GPIO Registers

The registers listed in the Register Summary table are for a single instance of the MEC140x/1x. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the Register Base Address Table.

TABLE 22-1: REGISTER BASE ADDRESS TABLE

Instance Name	Instance Number	Host	Address Space	Base Address
GPIO	0	LPC	I/O	Note 22-2
	0	EC	32-bit internal address space	0008_1000h

Note 22-1 The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

Note 22-2 The GPIO registers may be accessed by the LPC Host via the EMI block via GPIO commands or by direct access if enabled by firmware. See the firmware documentation for a description of this access method.

TABLE 22-2: REGISTER SUMMARY

Offset	Register Name
000h	Reserved (GPIO000 not implemented)
004h - 01Ch	GPIO001-GPIO007 Pin Control Register
020h - 03Ch	GPIO010-GPIO017 Pin Control Register

TABLE 22-2: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
040h - 05Ch	GPIO020-GPIO027 Pin Control Register
060h - 078h	GPIO030-GPIO036 Pin Control Register
080h - 09Ch	GPIO040-GPIO047 Pin Control Register
0A0h - 0BCh	GPIO050-GPIO057 Pin Control Register
0C0h - 0CCh	GPIO060-GPIO063 Pin Control Register
0D0h	Reserved (GPIO064 not implemented - see Note 22-4)
0D4h - 0D8h	GPIO065-GPIO066 Pin Control Register
0DCh	Reserved (GPIO067 not implemented - see Note 22-4)
0E0h - 0F8h	Reserved (GPIO070-GPIO076 not implemented)
100h - 11Ch	GPIO100-GPIO107 Pin Control Register
120h - 13Ch	GPIO110-GPIO117 Pin Control Register
140h - 15Ch	GPIO120-GPIO127 Pin Control Register
160h - 178h	GPIO130-GPIO136 Pin Control Register
180h - 19Ch	GPIO140-GPIO147 Pin Control Register
1A0h - 1BCh	GPIO150-GPIO157 Pin Control Register
1C0h - 1D8h	GPIO160-GPIO166 Pin Control Register
280h (Note 22-3)	Output GPIO[000:036]
284h (Note 22-3)	Output GPIO[040:076]
288h (Note 22-3)	Output GPIO[100:127]
28Ch (Note 22-3)	Output GPIO[140:176]
300h (Note 22-3)	Input GPIO[000:036]
304h (Note 22-3)	Input GPIO[040:076]
308h (Note 22-3)	Input GPIO[100:127]
30Ch (Note 22-3)	Input GPIO[140:176]
3F0h	GPIO Lock 3
3F4h	GPIO Lock 2

TABLE 22-2: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
3F8h	GPIO Lock 1
3FCh	GPIO Lock 0
500h	Reserved
504h - 51Ch	GPIO001-GPIO007 Pin Control Register 2
520h - 53Ch	GPIO010-GPIO017 Pin Control Register 2
540h - 55Ch	GPIO020-GPIO027 Pin Control Register 2
560h - 578h	GPIO030-GPIO036 Pin Control Register 2 (see Note 22-5 for limitations)
580h - 59Ch	GPIO040-GPIO047 Pin Control Register 2 (see Note 22-5 for limitations)
5A0h - 5BCh	GPIO050-GPIO057 Pin Control Register 2
5C0h - 5CCh	GPIO060-GPIO063 Pin Control Register 2 (see Note 22-5 for limitations)
5D0h	Reserved (GPIO064 not implemented - see Note 22-4)
5D4h - 5D8h	GPIO065-GPIO066 Pin Control Register 2
5DCh	Reserved (GPIO067 not implemented - see Note 22-4)
5E0h - 5F8h	Reserved (GPIO070-GPIO076 not implemented)
5E0h - 5FCh	GPIO100-GPIO107 Pin Control Register 2
600h - 61Ch	GPIO110-GPIO117 Pin Control Register 2
620h - 63Ch	GPIO120-GPIO127 Pin Control Register 2
640h - 658h	GPIO130-GPIO136 Pin Control Register 2
660h - 67Ch	GPIO140-GPIO147 Pin Control Register 2
680h - 69Ch	GPIO150-GPIO157 Pin Control Register 2
6A0h - 6B8h	GPIO160-GPIO166 Pin Control Register 2

Note 22-3 The GPIO input and output registers are LPC I/O accessible via Region 0 of the EMI block. This access is defined in the EMI Protocols chapter of the firmware specification.

Note 22-4 There is no Pin Control Register 2 for GPIO064 and GPIO067, which are PCI_PIO buffer type pins. The drive strength and slew rate are not configurable on these pins.

Note 22-5 The drive strength and slew rate are not configurable for the LPC functions on GPIO034, GPIO061, GPIO063, and GPIO40 - GPIO044 since they are controlled by the PCI_PIO type buffers.

22.6.1 PIN CONTROL REGISTERS

Two Pin Control Registers are implemented for each GPIO. The Pin Control Register format is described in Section 22.6.1.1, "Pin Control Register," on page 329. The Pin Control Register 2 format is described in Section 22.6.1.2, "Pin Control Register 2," on page 332. Pin Control Register address offsets and defaults for each product are defined in Section 22.5.4.1, "MEC140x Pin Control Registers Defaults," on page 320, and Section 22.5.4.2, "MEC141x Pin Control Registers Defaults," on page 323.

22.6.1.1 Pin Control Register

Offset	See Table 22-2, "Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:25	RESERVED	RES	-	-
24	GPIO input from pad On reads, Bit [24] reflects the state of GPIO input from the pad regardless of setting of Bit [10]. **Note:** This bit is forced high when the selected power well is off as selected by the Power Gating Signal bits. See bits[3:2].	R	Note 22-6	nSYSRST
23:17	RESERVED	RES	-	-
16	GPIO output data If enabled by the GPIO Output Control Select bit, the GPIO output data bit determines the level on the GPIO pin when the pin is configured for the GPIO output function. On writes: If enabled via the GPIO Output Control Select 0: GPIO[x] out = '0' 1: GPIO[x] out = '1' **Note:** If disabled via the GPIO Output Control Select then the GPIO[x] out pin is unaffected by writing this bit. On reads: Bit [16] returns the last programmed value, not the value on the pin.	R/W (GPIO Output Control Select = 0) R (GPIO Output Control Select=1)	Note 22-6	nSYSRST
15:14	RESERVED	RES	-	-
13:12	Mux Control The Mux Control field determines the active signal function for a pin. 00 = GPIO Function Selected 01 = Signal Function 1 Selected 10 = Signal Function 2 Selected 11 = Signal Function 3 Selected	R/W	Note 22-6	nSYSRST
11	Polarity 0 = Non-inverted 1 = Inverted When the Polarity bit is set to '1' and the Mux Control bits are greater than '00,' the selected signal function outputs are inverted and Interrupt Detection (int_det) sense defined in Table 22-3, "Edge Enable and Interrupt Detection Bits Definition" is inverted. When the Mux Control field selects the GPIO signal function (Mux = '00'), the Polarity bit does not effect the output. Regardless of the state of the Mux Control field and the Polarity bit, the state of the pin is always reported without inversion in the GPIO input register.	R/W	Note 22-6	nSYSRST

Offset	See Table 22-2, "Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	GPIO Output Control Select Every GPIO has two mechanisms to set a GPIO data output: Output GPIO Bit located in the grouped GPIO Output Registers and the single GPIO output data bit located in bit 16 of this register. This control bit determines the source of the GPIO output. 0 = Pin Control Bit[16] GPIO output data bit enabled When this bit is zero the single GPIO output data bit is enabled. (GPIO output data is R/W capable and the Grouped Output GPIO is disabled (i.e., Read-Only). 1 = Grouped Output GPIO enable When this bit is one the GPIO output data write is disabled (i.e., Read-Only) and the Grouped Output GPIO is enabled (i.e., R/W). **Note:** See description in Section 22.5.1, "Accessing GPIOs".	R/W	Note 22-6	nSYSRST
9	GPIO Direction 0 = Input 1 = Output The GPIO Direction bit controls the buffer direction only when the Mux Control field is '00' selecting the pin signal function to be GPIO. When the Mux Control field is greater than '00' (i.e., a non-GPIO signal function is selected) the GPIO Direction bit has no affect and the selected signal function logic directly controls the pin direction.	R/W	Note 22-6	nSYSRST
8	Output Buffer Type 0 = Push-Pull 1 = Open Drain **Note:** Unless explicitly stated otherwise, pins with (I/O/OD) or (O/OD) in their buffer type column in the tables in are compliant with the following Programmable OD/PP Multiplexing Design Rule: Each compliant pin has a programmable open drain/push-pull buffer controlled by the Output Buffer Type bit in the associated Pin Control Register. The state of this bit controls the mode of the interface buffer for all selected functions, including the GPIO function.	R/W	Note 22-6	nSYSRST
7	Edge Enable (edge_en) 0 = Edge detection disabled 1 = Edge detection enabled **Note:** See Table 22-3, "Edge Enable and Interrupt Detection Bits Definition".	R/W	Note 22-6	nSYSRST

Offset	See Table 22-2, "Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
6:4	Interrupt Detection (int_det) The interrupt detection bits determine the event that generates a GPIO_Event. **Note:** See Table 22-3, "Edge Enable and Interrupt Detection Bits Definition". **Note:** Since the GPIO input is always available, even when the GPIO is not selected as the alternate function, the GPIO interrupts may be used for detecting pin activity on alternate functions. The only exception to this is the analog functions (e.g., ADC, DAC, Comparator inputs)	R/W	Note 22-6	nSYSRST
3:2	Power Gating Signals The Power Gating Signals provide the chip Power Emulation options. The pin will be tristated when the selected power well is off (i.e., gated) as indicated. The Emulated Power Well column defined in Pin Multiplexing tables indicates the emulation options supported for each signal. The Signal Power Well column defines the buffer power supply per function. **Note:** Note that all GPIOs support Power Gating unless otherwise noted. 00 = VTR The output buffer is tristated when VTRGD = 0. 01 = VCC The output buffer is tristate when VCC_PWRGD = 0. 10 = Reserved 11 = Reserved	R/W	Note 22-6	nSYSRST
1:0	PU/PD (PU_PD) These bits are used to enable an internal pull-up or pull-down resistor device on the pin. 00 = None. Pin tristates when no active driver is present on the pin. 01 = Pull Up Enabled 10 = Pull Down Enabled (Note 22-7) 11 = Repeater mode. Pin is kept at previous voltage level when no active driver is present on the pin.	R/W	Note 22-6	nSYSRST

Note 22-6 See Section 22.5.4, "Pin Control Registers," on page 320 for the offset and default values for each GPIO Pin Control Register.

Note 22-7 The internal pull-down control should not be selected when configured for an LPC function, which uses the PCI_PIO buffer. Signals with PCI_PIO buffer type do not have an internal pull-down. This configuration option has no effect on the pin.

TABLE 22-3: EDGE ENABLE AND INTERRUPT DETECTION BITS DEFINITION

Edge Enable	Interrupt Detection Bits			Selected Function
D7	D6	D5	D4	
0	0	0	0	Low Level Sensitive
0	0	0	1	High Level Sensitive
0	0	1	0	Reserved
0	0	1	1	Reserved
0	1	0	0	Interrupt events are disabled
0	1	0	1	Reserved
0	1	1	0	Reserved
0	1	1	1	Reserved
1	1	0	1	Rising Edge Triggered
1	1	1	0	Falling Edge Triggered
1	1	1	1	Either edge triggered

Note: Only edge triggered interrupts can wake up the main ring oscillator. The GPIO must be enabled for edge-triggered interrupts and the GPIO interrupt must be enabled in the interrupt aggregator in order to wake up the ring when the ring is shut down.

APPLICATION NOTE: All GPIO interrupt detection configurations default to '0000', which is low level interrupt. Having interrupt detection enabled will un-gated the clock to the GPIO module whenever the interrupt is active, which increases power consumption. Interrupt detection should be disabled when not required to save power; this is especially true for pin interfaces (i.e., LPC).

22.6.1.2 Pin Control Register 2

Offset	See Note 22-6			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	RESERVED	RES	-	-
5:4	Drive Strength These bits are used to select the drive strength on the pin. 00 = 2mA 01 = 4mA 10 = 8mA 11 = 12mA	R/W	00	nSYSRST

Offset	See Note 22-6			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3:1	RESERVED	RES	-	-
0	Slew Rate This bit is used to select the slew rate on the pin. 0 = slow (half frequency) 1 = fast	R/W	0	nSYSRST

22.6.2 GPIO OUTPUT REGISTERS

If enabled by the GPIO Output Control Select bit, the grouped GPIO Output bits determine the level on the GPIO pin when the pin is configured for the GPIO output function.

On writes:

If enabled via the GPIO Output Control Select

0: GPIO[x] out = '0'

1: GPIO[x] out = '1'

If disabled via the GPIO Output Control Select then the GPIO[x] out pin is unaffected by writing the corresponding GPIO bit in the grouped Output GPIO[xxx:yyy] register.

On reads:

The GPIO output bit in the grouped Output GPIO[xxx:yyy] register returns the last programmed value, not the value on the pin.

Note: Bits associated with GPIOs not implemented are Reserved.

22.6.2.1 Output GPIO[000:036]

Offset	280h (Note 22-3)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	RESERVED	RES	-	-
30:24	GPIO[036:030] Output	R/W	00h	nSYSRST
23:16	GPIO[027:020] Output	R/W	00h	nSYSRST
15:8	GPIO[017:010] Output	R/W	00h	nSYSRST
7:0	GPIO[007:000] Output	R/W	00h	nSYSRST

22.6.2.2 Output GPIO[040:076]

Offset	284h (Note 22-3)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	RESERVED	RES	-	-
23:16	GPIO[067:060] Output **Note:** GPIO064 and GPIO067 are not implemented. Firmware should always write 0 to these locations.	R/W	00h	nSYSRST
15:8	GPIO[057:050] Output	R/W	00h	nSYSRST
7:0	GPIO[047:040] Output	R/W	00h	nSYSRST

22.6.2.3 Output GPIO[100:127]

Offset	288h (Note 22-3)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	RESERVED	RES	-	-
30:24	GPIO[136:130] Output	R/W	00h	nSYSRST
23:16	GPIO[127:120] Output	R/W	00h	nSYSRST
15:8	GPIO[117:110] Output	R/W	00h	nSYSRST
7:0	GPIO[107:100] Output	R/W	00h	nSYSRST

22.6.2.4 Output GPIO[140:176]

Offset	28Ch (Note 22-3)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:23	RESERVED	RES	-	-
22:16	GPIO[166:160] Output	R/W	00h	nSYSRST

Offset	28Ch (Note 22-3)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:8	GPIO[157:150] Output	R/W	00h	nSYSRST
7:0	GPIO[147:140] Output	R/W	00h	nSYSRST

22.6.3 GPIO INPUT REGISTERS

The GPIO Input Registers can always be used to read the state of a pin, even when the pin is in an output mode and/or when a signal function other than the GPIO signal function is selected; i.e., the Pin Control Register Mux Control bits are not equal to '00.'

The MSbit of the Input GPIO registers have been implemented as a read/write scratch pad bit to support processor specific instructions.

Note: Bits associated with GPIOs that are not implemented are shown as Reserved.

22.6.3.1 Input GPIO[000:036]

Offset	300h (Note 22-3)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	nSYSRST
30:24	GPIO[036:030] Input	R	00h	nSYSRST
23:16	GPIO[027:020] Input	R	00h	nSYSRST
15:8	GPIO[017:010] Input	R	00h	nSYSRST
7:0	GPIO[007:000] Input	R	00h	nSYSRST

22.6.3.2 Input GPIO[040:076]

Offset	304h (Note 22-3)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	nSYSRST
30:24	RESERVED	R	-	-
23:16	GPIO[067:060] Input **Note:** GPIO064 and GPIO067 are not implemented.	R	00h	nSYSRST
15:8	GPIO[057:050] Input	R	00h	nSYSRST
7:0	GPIO[047:040] Input	R	00h	nSYSRST

22.6.3.3 Input GPIO[100:127]

Offset	308h (Note 22-3)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	nSYSRST
30:24	GPIO[136:130] Input	R	00h	nSYSRST
23:16	GPIO[127:120] Input	R	00h	nSYSRST
15:8	GPIO[117:110] Input	R	00h	nSYSRST
7:0	GPIO[107:100] Input	R	00h	nSYSRST

22.6.3.4 Input GPIO[140:176]

Offset	30Ch(Note 22-3)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	nSYSRST
32:16	GPIO[166:160] Input	R	00h	nSYSRST
15:8	GPIO[157:150] Input	R	00h	nSYSRST
7:0	GPIO[147:140] Input	R	00h	nSYSRST

23.0 SMBUS INTERFACE

23.1 Introduction

The MEC140x/1x SMBus Interface includes one instance of the SMBus controller core. This chapter describes aspects of the SMBus Interface that are unique to the MEC140x/1x instantiations of this core; including, Power Domain, Resets, Clocks, Interrupts, Registers and the Physical Interface. For a *General Description, Features, Block Diagram, Functional Description, Registers Interface and other core-specific details, see* Ref [1] (note: in this chapter, *italicized text* typically refers to SMBus controller core interface elements as described in Ref [1]).

23.2 References

1. SMBus Controller Core with Network Layer Support (SMB2) - 16MHz I2C Baud Clock“, Revision 3.52, Core-Level Architecture Specification, MCHP, 10/25/13

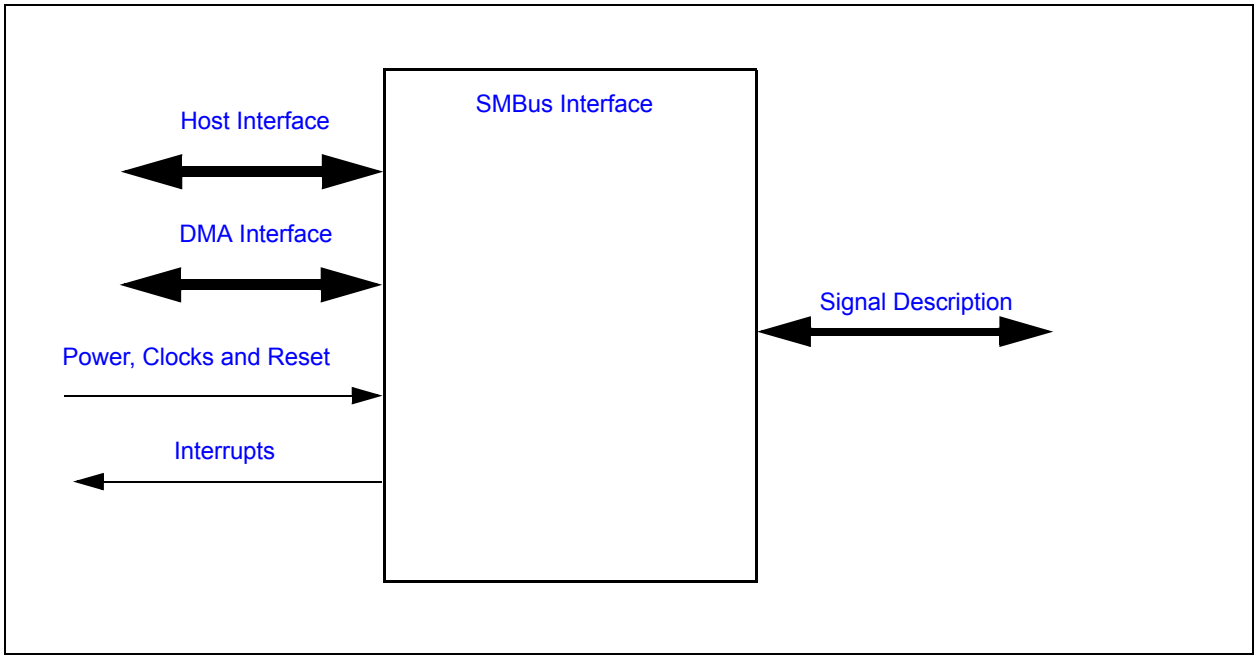
23.3 Terminology

There is no terminology defined for this chapter.

23.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface. In addition, this block is equipped with

FIGURE 23-1: I/O DIAGRAM OF BLOCK



23.5 Signal Description

The pin signals are defined in Section 2.0, "Pin Configuration," on page 12.

23.6 Host Interface

The registers defined for the SMBus Interface are accessible as indicated in Section 23.12, "SMBus Registers".

23.7 DMA Interface

This block is designed to communicate with the Internal DMA Controller. This feature is defined in the SMBus Controller Core Interface specification (*See* Ref [1]).

Note: For a description of the Internal DMA Controller implemented in this design see Chapter 24.0, "Internal DMA Controller".

23.8 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

23.8.1 POWER DOMAINS

Name	Description
VTR	This power well sources all of the registers and logic in this block, except where noted.

23.8.2 CLOCK INPUTS

Name	Description
48 MHz Ring Oscillator	This is the clock signal drives the SMBus controller core. The core also uses this clock to generate the SMB_CLK on the pin interface.
16MHz_Clk	This is the clock signal is used for baud rate generation.

23.8.3 RESETS

Name	Description
nSYSRST	This reset signal resets all of the registers and logic in the SMBus controller core.

23.9 Interrupts

Source	Description
SMB_WAKE	The SMBus_Wake event is generated when a valid SMBus START sequence is detected on the SMBus pin interface.
SMB	SMBus Activity Interrupt Event

23.10 Low Power Modes

The SMBus Interface may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. If an SMBus START is detected while the SMBus block is in a low power state the block will generate the SMB_WAKE event. In enabled in the Jump Table Vectored Interrupt Controller (JTVIC) on page 159, this event may be used to wake the chip from a low power sleep state.

23.11 Description

23.11.1 SMBUS CONTROLLER CORE

The MEC140x/1x SMBus Interface behavior is defined in the SMBus Controller Core Interface specification (*See* Ref [1]).

23.11.2 PHYSICAL INTERFACE

23.11.2.1 Overview

The Physical Interface for the SMB controller core is configurable for up to 15 ports as defined below in SMBus Port Selection.

Each of the 3 SMBus controllers can be connected to any of the ports defined in the table. The *PORT SEL [3:0]* bits in each controller will appear the same (TABLE 23-1:). The default for each field is Fh, Reserved, which means that the SMB controller is not connected to a port.

An SMB port should be connected to a single controller. An attempt to configure the *PORT SEL [3:0]* bits in one controller to a value already assigned to another controller may result in unexpected results.

The port signal-function names and pin numbers are defined in the Pin Configuration chapter. The SMBus port selection is made using the *PORT SEL [3:0]* bits in the *Configuration Register* as described in Ref [1] and in the subsections that follow.

For SMBus port signal functions that are alternate functions of GPIO pins, the buffer type for these pins must be configured as open-drain outputs when the port is selected as defined in SMBus Port Selection. For more information regarding the SMBus controller core see *Section 2.2,* "*Physical Interface*" in Ref[1].

23.11.2.2 SMBus Port Selection

TABLE 23-1: SMBUS PORT SELECTION

PORT SEL [3:0]				PORT (SEE PIN CONFIGURATION CHAPTER FOR A DESCRIPTION OF THE SMBUS PIN CONFIGURATION.)
3	**2**	**1**	**0**	
0	0	0	0	SMB00 (3.3V or 1.8V)
0	0	0	1	SMB01 (3.3V or 1.8V)
0	0	1	0	SMB02 (3.3V or 1.8V)
0	0	1	1	SMB03 (3.3V or 1.8V)
0	1	0	0	SMB04 (3.3V or 1.8V)
0	1	0	1	SB-TSI
0110b - 1111b				Reserved

Note 1: see Pin Configuration chapter for a description of the SMBus pin configuration.

2: The SMB00 to SMB04 Ports have the option to be configured for either 3.3V or 1.8V signaling. This selection is determined by the GPIO alternate function mux. SMBxx_DATA/SMBxx_CLK are 3.3V I/O signaling. SMBxx_DATA18/SMBxx_CLK18 are 1.8V I/O signaling.

23.12 SMBus Registers

The registers listed in the *SMBus Core Register Summary* table in the SMBus Controller Core Interface specification (Ref [1]) are for a single instance of the SMBus Controller Core. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the following table:

TABLE 23-2: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
SMBus Controller	0	EC	32-bit internal address space	0000_1800h
SMBus Controller	1	EC	32-bit internal address space	0000_AC00h
SMBus Controller	2	EC	32-bit internal address space	0000_B000h

Note: The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

24.0 INTERNAL DMA CONTROLLER

24.1 Features

- Supports Memory-to-Memory BYTE, WORD, and DWORD transfers
- Used to Perform DMA transactions for DMA capable hardware IP blocks
- Supports 7 DMA Channels that may be configured for any Hardware Device or Memory transfer
- Channel 0 Supports CRC-32 generation

24.2 Introduction

The Internal DMA Controller transfers data to/from the source from/to the destination. The firmware is responsible for setting up each channel. Afterwards either the firmware or the hardware may perform the flow control. The hardware flow control exists entirely inside the source device. Each transfer may be 1, 2, or 4 bytes in size, so long as the device supports a transfer of that size. Every device must be on the internal 32-bit address space.

24.3 References

No references have been cited for this chapter

24.4 Terminology

TABLE 24-1: TERMINOLOGY

Term	Definition
DMA Transfer	This is a complete **DMA Transfer** which is done after the **Master Device** terminates the transfer, the Firmware Aborts the transfer or the DMA reaches its transfer limit. A DMA Transfer may consist of one or more data packets.
Data Packet	Each data packet may be composed of 1, 2, or 4 bytes. The size of the data packet is limited by the max size supported by both the source and the destination. Both source and destination will transfer the same number of bytes per packet.
Channel	The Channel is responsible for end-to-end (source-to-destination) Data Packet delivery.
Device	A Device may refer to a Master or Slave connected to the DMA Channel. Each DMA Channel may be assigned one or more devices.
Master Device	This is the master of the DMA, which determines when it is active. The Firmware is the master while operating in Firmware Flow Control. The Hardware is the master while operating in Hardware Flow Control. The Master Device in Hardware Mode is selected by **DMA Channel Control:Hardware Flow Control Device**. It is the index of the **Flow Control** Port.
Slave Device	The Slave Device is defined as the device associated with the targeted Memory Address.

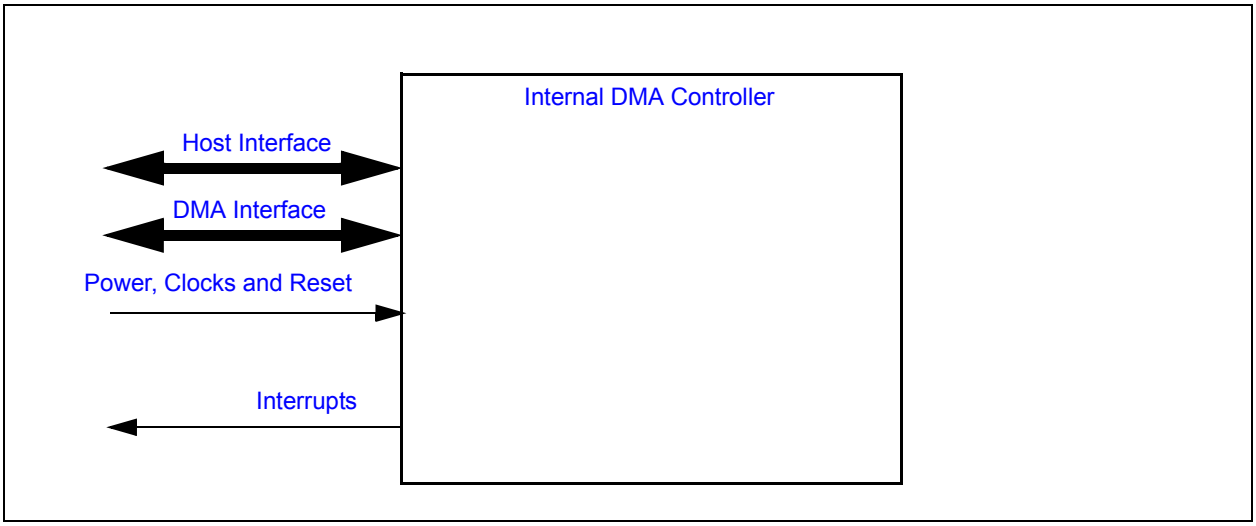
TABLE 24-1: TERMINOLOGY (CONTINUED)

Term	Definition
Source	The DMA Controller moves data from the Source to the Destination. The Source provides the data. The Source may be either the Master or Slave Controller.
Destination	The DMA Controller moves data from the Source to the Destination. The Destination receives the data. The Destination may be either the Master or Slave Controller.

24.5 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 24-1: INTERNAL DMA CONTROLLER I/O DIAGRAM



24.5.1 SIGNAL DESCRIPTION

This block doesn't have any external signals that may be routed to the pin interface. This DMA Controller is intended to be used internally to transfer large amounts of data without the embedded controller being actively involved in the transfer.

24.5.2 HOST INTERFACE

The registers defined for the Internal DMA Controller are accessible by the various hosts as indicated in Section 24.10, "DMA Main Registers".

24.5.3 DMA INTERFACE

Each DMA Master Device that may engage in a DMA transfer must have a compliant DMA interface. The following table lists the DMA Devices in the MEC140x/1x.

TABLE 24-2: DMA CONTROLLER DEVICE SELECTION

Device Name	Device Number (Note 1)	Controller Source
SMBus 0 Controller	0	Slave
	1	Master

TABLE 24-2: DMA CONTROLLER DEVICE SELECTION (CONTINUED)

Device Name	Device Number (Note 1)	Controller Source
SMBus 1 Controller	2	Slave
	3	Master
SMBus 2 Controller	4	Slave
	5	Master
QUAD SPI Master Controller	6	Transmit
	7	Receive

Note 1: The Device Number is programmed into field HARDWARE_FLOW_CONTROL_DEVICE of the DMA Channel N Control Register register.

24.6 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

24.6.1 POWER DOMAINS

Name	Description
VTR	This power well sources all of the registers and logic in this block, except where noted.

24.6.2 CLOCK INPUTS

Name	Description
48 MHz Ring Oscillator	This clock signal drives selected logic (e.g., counters).

24.6.3 RESETS

Name	Description
nSYSRST	This reset signal resets all of the registers and logic in this block.
DMA_RESET	This reset is generated if either the nSYSRST is asserted or the SOFT_RESET is asserted.

24.7 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
DMA0	Direct Memory Access Channel 0 This signal is generated by the STATUS_DONE bit.
DMA1	Direct Memory Access Channel 1 This signal is generated by the STATUS_DONE bit.
DMA2	Direct Memory Access Channel 2 This signal is generated by the STATUS_DONE bit.
DMA3	Direct Memory Access Channel 3 This signal is generated by the STATUS_DONE bit.
DMA4	Direct Memory Access Channel 4 This signal is generated by the STATUS_DONE bit.
DMA5	Direct Memory Access Channel 5 This signal is generated by the STATUS_DONE bit.
DMA6	Direct Memory Access Channel 6 This signal is generated by the STATUS_DONE bit.

24.8 Low Power Modes

The Internal DMA Controller may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

When the block is commanded to go to sleep it will place the DMA block into sleep mode only after all transactions on the DMA have been completed. For Firmware Flow Controlled transactions, the DMA will wait until it hits its terminal count and clears the Go control bit. For Hardware Flow Control, the DMA will go to sleep after either the terminal count is hit, or the Master device flags the terminate signal.

24.9 Description

The MEC140x/1x features a multi-channel DMA controller. The DMA controller can autonomously move data from/to any DMA capable master device to/from any populated memory location. This mechanism allows hardware IP blocks to transfer large amounts of data into or out of memory without EC intervention.

The DMA has the following characteristics:

- Data is only moved 1 Data Packet at a time
- Data only moves between devices on the accessible via the internal 32-bit address space
- Each DMA Channel may be configured to communicate with any DMA capable device on the 32-bit internal address space. Each device has been assigned a device number. See Section 24.5.3, "DMA Interface," on page 343.

The controller will accesses SRAM buffers only with incrementing addresses (that is, it cannot start at the top of a buffer, nor does it handle circular buffers automatically). The controller does not handle chaining (that is, automatically starting a new DMA transfer when one finishes).

A DMA Channel can optionally generate a CRC-32 on the data transfered by the Channel.

24.9.1 CONFIGURATION

The DMA Controller is enabled via the ACTIVATE bit in DMA Main Control Register register.

Each DMA Channel must also be individually enabled via the CHANNEL_ACTIVATE bit in the DMA Channel N Activate Register to be operational.

Before starting a DMA transaction on a DMA Channel the host must assign a DMA Master to the channel via bits[15:9] HARDWARE_FLOW_CONTROL_DEVICE. The host must not configure two different channels to the same DMA Master at the same time.

Data will be transfered between the DMA Master, starting at the programmed DEVICE_ADDRESS, and the targeted memory location, starting at the MEMORY_START_ADDRESS. The address for either the DMA Master or the targeted memory location may remain static or it may increment. To enable the DMA Master to increment its address set the INCREMENT_DEVICE_ADDRESS bit. To enable the targeted memory location to increment its addresses set the INCREMENT_MEMORY_ADDRESS. The DMA transfer will continue as long as the target memory address being accessed is less than the MEMORY_END_ADDRESS. If the DMA Controller detects that the memory location it is attempting to access on the Target is equal to the MEMORY_END_ADDRESS it will notify the DMA Master that the transaction is done. Otherwise the Data will be transferred in packets. The size of the packet is determined by the TRANSFER_SIZE.

24.9.2 OPERATION

The DMA Controller is designed to move data from one memory location to another.

24.9.2.1 Establishing a Connection

A DMA Master will initiate a DMA Transaction by requesting access to a channel. The DMA arbiter, which evaluates each channel request using a basic round robin algorithm, will grant access to the DMA master. Once granted, the channel will hold the grant until it decides to release it, by notifying the DMA Controller that it is done.

Note:	If Firmware wants to prevent any other channels from being granted while it is active it can set the LOCK_CHANNEL bit.

24.9.2.2 Initiating a Transfer

Once a connection is established the DMA Master will issue a DMA request to start a DMA transfer.

Firmware can initiate a transaction by setting the TRANSFER_GO bit. The DMA transfer will remain active until either the Master issues a Terminate or the DMA Controller signals that the transfer is DONE. Firmware may terminate a transaction by setting the TRANSFER_ABORT bit.

Note:	Before initiating a DMA transaction via firmware the hardware flow control must be disabled via the DISABLE_HARDWARE_FLOW_CONTROL bit.

Data may be moved from the DMA Master to the targeted Memory address or from the targeted Memory Address to the DMA Master. The direction of the transfer is determined by the TRANSFER_DIRECTION bit.

Once a transaction has been initiated firmware can use the STATUS_DONE bit to determine when the transaction is completed. This status bit is routed to the interrupt interface. In the same register there are additional status bits that indicate if the transaction completed successfully or with errors. This bits are OR'd together with the STATUS_DONE bit to generate the interrupt event. Each status be may be individually enabled/disabled from generating this event.

24.9.2.3 CRC Generation

A CRC generator can be attached to a DMA channel in order to generate a CRC on the data as it is transfered from the source to the destination. The CRC used is the CRC-32 algorithm used in IEEE 802.3 and many other protocols, using the polynomial $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$. The CRC generation takes place in parallel with the data transfer; enabling CRC will not increase the time to complete a DMA transaction. The CRC generator has the optional ability to automatically transfer the generated CRC to the destination after the data transfer has completed.

CRC generation is subject to a number of restrictions:

- The CRC is only generated on channels that have the CRC hardware. See Table 24-6, "DMA Channel N Register Summary" for a definition of which channels have the ability to generate a CRC
- The DMA transfer must be 32-bits
- If CRC is enabled, DMA interrupts are inhibited until the CRC is completed, including the optional post-transfer copy of it is enabled
- The CRC must be initialized by firmware. The value FFFFFFFFh must be written to the Data Register in order to initialize the generator for the standard CRC-32-IEEE algorithm

24.9.3 DMA REGISTERS

The DMA Controller consists of a single Main Block of registers that applies to all channels and channel specific registers. Table 24-4, "DMA Main Register Summary" lists the registers in the Main Block and Table 24-6, "DMA Channel N Register Summary" lists the registers in each channel.

24.10 DMA Main Registers

The addresses of each register listed in these tables are defined as a relative offset to the “Base Address” defined in the DMA Main Register Base Address. The Base Address indicates where the first register can be accessed in a particular bank of registers.

TABLE 24-3: DMA MAIN REGISTER BASE ADDRESS

Instance Name	Channel Number	Host	Address Space	Base Address
DMA Controller	Main Block	EC	32-bit internal address space	0000_2400h

TABLE 24-4: DMA MAIN REGISTER SUMMARY

Offset	REGISTER NAME (Mnemonic)
00h	DMA Main Control Register
04h	DMA Data Packet Register

24.10.1 DMA MAIN CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:2	Reserved	R	-	-
1	SOFT_RESET Soft reset the entire module. This bit is self-clearing.	W	0b	-
0	ACTIVATE Enable the blocks operation. 1=Enable block. Each individual channel must be enabled separately. 0=Disable all channels.	R/WS	0b	DMA_RESET

24.10.2 DMA DATA PACKET REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	DATA_PACKET Debug register that has the data that is stored in the Data Packet. This data is read data from the currently active transfer source.	R	0000h	-

24.11 DMA Channel Registers

The addresses of each register listed in these tables are defined as a relative offset to the “Base Address” defined in the DMA Channel N Register Base Address. The Base Address indicates where the first register can be accessed in a particular bank of registers.

TABLE 24-5: DMA CHANNEL N REGISTER BASE ADDRESS

Instance Name	Channel Number (N)	Host	Address Space	Base Address
DMA Controller	0 (Note 1)	EC	32-bit internal address space	0000_2440h
DMA Controller	1	EC	32-bit internal address space	0000_2480h
DMA Controller	2	EC	32-bit internal address space	0000_24C0h
DMA Controller	3	EC	32-bit internal address space	0000_2500h
DMA Controller	4	EC	32-bit internal address space	0000_2540h
DMA Controller	5	EC	32-bit internal address space	0000_2580h
DMA Controller	6	EC	32-bit internal address space	0000_25C0h

Note 1: Only DMA Channel 0 has CRC-32 generation support, which can be used with the Quad SPI Master Controller or for Memory-to-Memory DMA transfers.

TABLE 24-6: DMA CHANNEL N REGISTER SUMMARY

Offset	Register Name (Mnemonic) (Note 2)
00h	DMA Channel N Activate Register
04h	DMA Channel N Memory Start Address Register
08h	DMA Channel N Memory End Address Register
0Ch	DMA Channel N Device Address Register
10h	DMA Channel N Control Register
14h	DMA Channel N Interrupt Status Register
18h	DMA Channel N Interrupt Enable Register
1Ch	Test
20h (Note 3)	DMA Channel N CRC Enable Register

TABLE 24-6: DMA CHANNEL N REGISTER SUMMARY (CONTINUED)

Offset	Register Name (Mnemonic) (Note 2)
24h (Note 3)	DMA Channel N CRC Data Register
28h (Note 3)	DMA Channel N CRC Post Status Register
2Ch (Note 3)	DMA Channel N CRC Test Register

2: The letter 'N' following DMA Channel indicates the Channel Number. Each Channel implemented will have these registers to determine that channel's operation.

3: These registers are only present in DMA Channel 0. Offsets 20h to 2Ch are reserved in all the other channels.

24.11.1 DMA CHANNEL N ACTIVATE REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	R	-	-
0	CHANNEL_ACTIVATE Enable this channel for operation. The DMA Main Control:Activate must also be enabled for this channel to be operational.	R/W	0h	DMA_RESET

24.11.2 DMA CHANNEL N MEMORY START ADDRESS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	MEMORY_START_ADDRESS This is the starting address for the Memory device. This field is updated by Hardware after every packet transfer by the size of the transfer, as defined by DMA Channel Control:Channel Transfer Size while the DMA Channel Control:Increment Memory Address is Enabled. The Memory device is defined as the device that is the slave device in the transfer. ex. With Hardware Flow Control, the Memory device is the device that is not connected to the Hardware Flow Controlling device. **Note:** This field is only as large as the maximum allowed AHB Address Size in the system. If the HADDR size is 24 Bits, then Bits [31:24] will be RESERVED.	R/W	0000h	DMA_ RESET

24.11.3 DMA CHANNEL N MEMORY END ADDRESS REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	MEMORY_END_ADDRESS This is the ending address for the Memory device. This will define the limit of the transfer, so long as DMA Channel Control:Increment Memory Address is Enabled. When the Memory Start Address is equal to this value, the DMA will terminate the transfer and flag the status DMA Channel Interrupt:Status Done. **Note:** This field is only as large as the maximum allowed AHB Address Size in the system. If the HADDR size is 24 Bits, then Bits [31:24] will be RESERVED.	R/W	0000h	DMA_ RESET

24.11.4 DMA CHANNEL N DEVICE ADDRESS REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	DEVICE_ADDRESS This is the Master Device address. This is used as the address that will access the Device on the DMA. The Device is defined as the Master of the DMA transfer; as in the device that is controlling the Hardware Flow Control. **APPLICATION NOTE:** Only Channel 0 has CRC function which may be utilized only by the Quad SPI Master Controller and for Memory-to-Memory transfers. It is recommended to use Channels 1-6 for the SMBus Controllers. This field is updated by Hardware after every Data Packet transfer by the size of the transfer, as defined by DMA Channel Control:Transfer Size while the DMA Channel Control:Increment Device Address is Enabled. **Note:** This field is only as large as the maximum allowed AHB Address Size in the system. If the HADDR size is 24 Bits, then Bits [31:24] will be RESERVED.	R/W	0000h	DMA_RESET

24.11.5 DMA CHANNEL N CONTROL REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:26	Reserved	R	-	-
25	TRANSFER_ABORT This is used to abort the current transfer on this DMA Channel. The aborted transfer will be forced to terminate immediately.	R/W	0h	DMA_RESET
24	TRANSFER_GO This is used for the **Firmware Flow Control** DMA transfer. This is used to start a transfer under the **Firmware Flow Control**. Do not use this in conjunction with the **Hardware Flow Control**; **DMA Channel Control:Disable Hardware Flow Control** must be set in order for this field to function correctly.	R/W	0h	DMA_RESET
23	Reserved	R	-	-
22:20	TRANSFER_SIZE This is the transfer size in Bytes of each Data Packet transfer. **Note:** The transfer size must be a legal transfer size. Valid sizes are 1, 2 and 4 Bytes.	R/W	0h	DMA_RESET

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
19	DISABLE_HARDWARE_FLOW_CONTROL This will Disable the **Hardware Flow Control**. When disabled, any DMA Master device attempting to communicate to the DMA over the **DMA Flow Control Interface (Ports: dma_req, dma_term, and dma_done)** will be ignored. This should be set before using the DMA channel in **Firmware Flow Control** mode.	RW	0h	DMA_RESET
18	LOCK_CHANNEL This is used to lock the arbitration of the Channel Arbiter on this channel once this channel is granted. Once this is locked, it will remain on the arbiter until it has completed it transfer (either the Transfer Aborted, Transfer Done or Transfer Terminated conditions). **Note:** This setting may starve other channels if the locked channel takes an excessive period of time to complete.	RW	0h	DMA_RESET
17	INCREMENT_DEVICE_ADDRESS This will enable an auto-increment to the **DMA Channel Device Address.** **1:** Increment the **DMA Channel Device Address** by **DMA Channel Control:Transfer Size** after every **Data Packet transfer** **0:** Do nothing	RW	0h	DMA_RESET
16	INCREMENT_MEMORY_ADDRESS This will enable an auto-increment to the **DMA Channel Memory Address.** 1=Increment the DMA Channel Memory Address by DMA Channel Control:Transfer Size after every Data Packet transfer 0=Do nothing **Note:** *If this is not set, the DMA will never terminate the transfer on its own. It will have to be terminated through the* Hardware Flow Control *or through a* DMA Channel Control:Transfer Abort.	RW	0h	DMA_RESET
15:9	HARDWARE_FLOW_CONTROL_DEVICE This is the device that is connected to this channel as its **Hardware Flow Control** master. The **Flow Control Interface** is a bus with each master concatenated onto it. This selects which bus index of the concatenated **Flow Control Interface** bus is targeted towards this channel. The **Flow Control Interface** Port list is **dma_req, dma_term,** and **dma_done**.	RW	0h	DMA_RESET
8	TRANSFER_DIRECTION This determines the direction of the DMA Transfer. 1=Data Packet Read from **Memory Start Address** followed by Data Packet Write to **Device Address** 0=Data Packet Read from **Device Address** followed by Data Packet Write to **Memory Start Address**	RW	0h	DMA_RESET
7:6	Reserved	R	-	-

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
5	BUSY This is a status signal. 1=The DMA Channel is busy (FSM is not IDLE) 0=The DMA Channel is not busy (FSM is IDLE)	RO	0h	DMA_RESET
4:3	STATUS This is a status signal. The status decode is listed in priority order with the highest priority first. 3: Error detected by the DMA 2: The DMA Channel is externally done, in that the Device has terminated the transfer over the **Hardware Flow Control** through the Port **dma_term** 1: The DMA Channel is locally done, in that **Memory Start Address** equals **Memory End Address** 0: DMA Channel Control:Run is Disabled (0x0) **Note:** This functionality has been replaced by the Interrupt field, and as such should never be used. The field will not flag back appropriately timed status, and if used may cause the firmware to become out-of-sync with the hardware. This field has multiple non-exclusive statuses, but may only display a single status. As such, multiple statuses may be TRUE, but this will appear as though only a single status has been triggered.	R	0h	DMA_RESET
2	DONE This is a status signal. It is only valid while **DMA Channel Control:Run** is Enabled. This is the inverse of the **DMA Channel Control:Busy** field, except this is qualified with the **DMA Channel Control:Run** field. 1=Channel is done 0=Channel is not done or it is OFF	RO	0h	DMA_RESET
1	REQUEST This is a status field. 1: There is a transfer request from the Master Device 0: There is no transfer request from the Master Device	RO	0h	DMA_RESET
0	RUN This is a control field. **Note:** This bit only applies to **Hardware Flow Control** mode. Do not use this bit in conjunction with the **Firmware Flow Control.** 1: This channel is enabled and will service transfer requests 0=This channel is disabled. All transfer requests are ignored	RW	0h	DMA_RESET

24.11.6 DMA CHANNEL N INTERRUPT STATUS REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:3	Reserved	R	-	-
2	STATUS_DONE This is an interrupt source register. This flags when the DMA Channel has completed a transfer successfully on its side. A completed transfer is defined as when the DMA Channel reaches its limit; **Memory Start Address** equals **Memory End Address**. A completion due to a **Hardware Flow Control Terminate** will not flag this interrupt. 1=**Memory Start Address** equals **Memory End Address** 1=**Memory Start Address** does not equal **Memory End Address**	R/WC	0h	DMA_ RESET
1	STATUS_FLOW_CONTROL This is an interrupt source register. This flags when the DMA Channel has encountered a **Hardware Flow Control Request** after the DMA Channel has completed the transfer. This means the Master Device is attempting to overflow the DMA. 1=Hardware Flow Control is requesting after the transfer has completed 0=No Hardware Flow Control event		0h	DMA_ RESET
0	STATUS_BUS_ERROR This is an interrupt source register. This flags when there is an Error detected over the internal 32-bit Bus. **1=** Error detected.	R/WC	0h	DMA_ RESET

24.11.7 DMA CHANNEL N INTERRUPT ENABLE REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:3	Reserved	R	-	-
2	STATUS_ENABLE_DONE This is an interrupt enable for **DMA Channel Interrupt:Status Done.** 1=Enable Interrupt 0=Disable Interrupt	R/W	0h	DMA_ RESET

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	STATUS_ENABLE_FLOW_CONTROL_ERROR This is an interrupt enable for **DMA Channel Interrupt:Status Flow Control Error.** 1=Enable Interrupt 0=Disable Interrupt	R/W	0h	DMA_ RESET
0	STATUS_ENABLE_BUS_ERROR This is an interrupt enable for **DMA Channel Interrupt:Status Bus Error.** 1=Enable Interrupt 0=Disable Interrupt	R/W	0h	DMA_ RESET

24.11.8 DMA CHANNEL N CRC ENABLE REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	R	-	-
1	CRC_POST_TRANSFER_ENABLE The bit enables the transfer of the calculated CRC-32 after the completion of the DMA transaction. If the DMA transaction is aborted by either firmware or an internal bus error, the transfer will not occur. If the target of the DMA transfer is a device and the device signaled the termination of the DMA transaction, the CRC post transfer will not occur. 1=Enable the transfer of CRC-32 for DMA Channel N after the DMA transaction completes 0=Disable the automatic transfer of the CRC	R/W	0h	DMA_ RESET
0	CRC_ENABLE 1=Enable the calculation of CRC-32 for DMA Channel N 0=Disable the calculation of CRC-32 for DMA Channel N	R/W	0h	DMA_ RESET

24.11.9 DMA CHANNEL N CRC DATA REGISTER

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CRC Writes to this register initialize the CRC generator. Reads from this register return the output of the CRC that is calculated from the data transfered by DMA Channel N. The output of the CRC generator is bit-reversed and inverted on reads, as required by the CRC-32-IEEE definition. A CRC can be accumulated across multiple DMA transactions on Channel N. If it is necessary to save the intermediate CRC value, the result of the read of this register must be bit-reversed and inverted before being written back to this register.	R/W	0h	DMA_ RESET

24.11.10 DMA CHANNEL N CRC POST STATUS REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	R	-	-
3	CRC_POST_TEST2 This is a test bit. Read back data is unpredictable.	R	0h	DMA_ RESET
2	CRC_POST_TRANSFER This bit is cleared to ‘0’ when a DMA transaction starts. If Post Transfer is enabled, and the CRC is successfully transferred following the completion of the DMA transaction, this bit is set to ‘1’. If the post transfer of the CRC is inhibited, because either firmware or the device terminated the transaction, this bit remains ‘0’.	R	0h	DMA_ RESET
1	CRC_POST_TEST1 This is a test bit. Read back data is unpredictable.	R	0h	DMA_ RESET
0	CRC_POST_TEST0 This is a test bit. Read back data is unpredictable.	R	0h	DMA_ RESET

24.11.11 DMA CHANNEL N CRC TEST REGISTER

Offset	2Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Reserved	R	-	-

25.0 PECI INTERFACE

25.1 Overview

The MEC140x/1x includes a PECI Interface to allow the EC to retrieve temperature readings from PECI-compliant devices. The PECI Interface implements the PHY and Link Layer of a PECI host controller as defined in References[1] and includes hardware support for the PECI 2.0 command set.

This chapter focuses on MEC140x/1x specific PECI Interface configuration information such as Power Domains, Clock Inputs, Resets, Interrupts, and other chip specific information. For a functional description of the MEC140x/1x PECI Interface refer to References [1].

25.2 References

1. PECI Interface Core, Rev. 1.31, Core-Level Architecture Specification, SMSC Confidential, 4/15/11

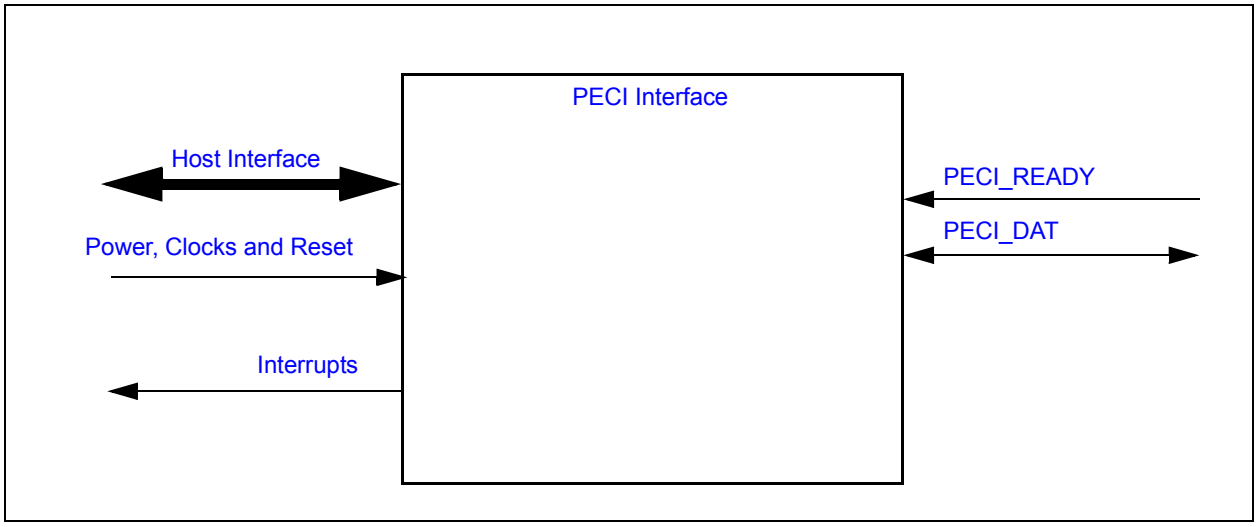
25.3 Terminology

No terminology has been defined for this chapter.

25.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 25-1: PECI INTERFACE I/O DIAGRAM



25.5 Signal Description

The Signal Description Table lists the signals that are typically routed to the pin interface.

TABLE 25-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
PECI_READY	Input	PECI Ready input pin **Note:** This signal is optional. If this signal is not on the pin interface it is pulled high internally.
PECI_DAT	Input/Output	PECI Data signal pin

Note:	Routing guidelines for the PECI_DAT pin is provided in Intel Platform design guides. Refer to the appropriate Intel document for current information. See TABLE 25-2:.

TABLE 25-2: PECI ROUTING GUIDELINES

Trace Impedance	50 Ohms +/- 15%
Spacing	10 mils
Routing Layer	Microstrip
Trace Width	Calculate to match impedance
Length	1” - 15”

25.6 Host Interface

The registers defined for the PECI Interface are accessible by the various hosts as indicated in Section 25.11, "PECI Interface Registers".

25.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

25.7.1 POWER DOMAINS

Name	Description
VTR	The PECI Interface logic and registers are powered by VTR.

25.7.2 CLOCK INPUTS

Name	Description
48 MHz Ring Oscillator	PECI Module Input Clock

25.7.3 RESETS

Name	Description
nSYSRST	PECI Core Reset Input

25.8 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
PECIHOST	PECI Host

25.9 Low Power Modes

The PECI Interface may be put into a low power state by the chip’s Power, Clocks, and Reset (PCR) circuitry.

25.10 Instance Description

There is one instance of the PECI Core implemented in the PECI Interface in the MEC140x/1x. See PECI Interface Core, Rev. 1.31, Core-Level Architecture Specification, SMSC Confidential, 4/15/11 for a description of the PECI Core.

25.11 PECI Interface Registers

The registers listed in the PECI Interface Register Summary table are for a single instance of the PECI Interface. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the PECI Interface Register Base Address Table.

TABLE 25-3: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
PECI Interface	0	EC	32-bit Internal Address Space	0000_6400h

Note: The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 25-4: PECI INTERFACE REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	Write Data Register
04h	Read Data Register
08h	Control Register
0Ch	Status Register 1
10h	Status Register 2
14h	Error Register
18h	Interrupt Enable 1 Register
1Ch	Interrupt Enable 2 Register
20h	Optimal Bit Time Register (Low Byte)
24h	Optimal Bit Time Register (High Byte)
28h	Test
2Ch	Test
30h-3Ch	Reserved
40h	Block ID Register
44h	Revision Register
48h - 7Ch	Test

Note: Test registers are reserved for Microchip use only. Reading and writing Test registers may cause undesirable results

For register details see References [1].

26.0 TACHOMETER

26.1 Introduction

This block monitors tachometer output signals (or locked rotor signals) from various types of fans, and determines their speed.

26.2 References

No references have been cited for this feature.

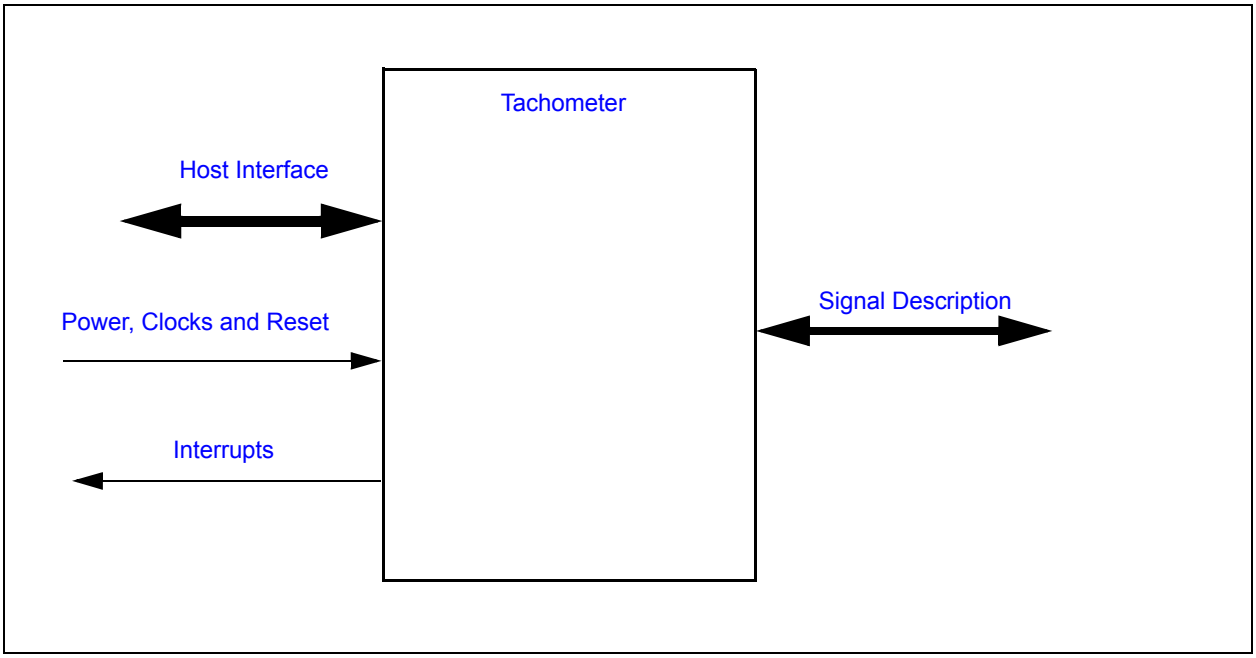
26.3 Terminology

There is no terminology defined for this section.

26.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 26-1: I/O DIAGRAM OF BLOCK



26.5 Signal Description

TABLE 26-1: SIGNAL DESCRIPTION

Name	Direction	Description
TACHx	Input	Tachometer input signal.

Note: 'x' represents the instance number (i.e., TACH0, TACH1, etc.). If there is only one tachometer input this may be omitted from the pin signal name.

26.6 Host Interface

The registers defined for the Tachometer are accessible by the various hosts as indicated in Section 26.11, "EC-Only Registers".

26.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

26.7.1 POWER DOMAINS

Name	Description
VTR	The logic and registers implemented in this block are powered by this power well.

26.7.2 CLOCK INPUTS

Name	Description
100kHz_Clk	This is the clock input to the tachometer monitor logic. In Mode 1, the TACHx input is measured in the number of these clocks.

26.7.3 RESETS

Name	Description
nSYSRST	This signal resets all the registers and logic in this block to their default state.

26.8 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
TACH	This internal signal is generated from the OR'd result of the status events, as defined in the TACHx Status Register.

26.9 Low Power Modes

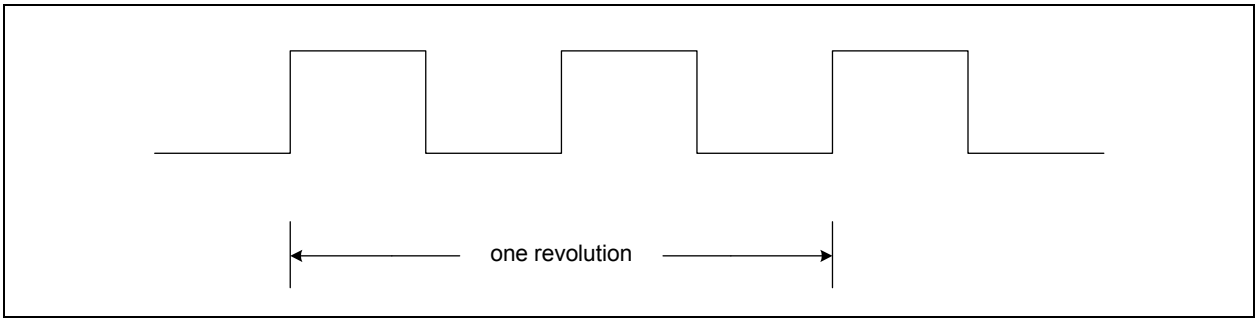
The Tachometer may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

26.10 Description

The Tachometer block monitors tachometer output signals (also referred to as TACH signals) or locked rotor signals generated by various types of fans. These signals can be used to determine the speed of the attached fan. This block is designed to monitor fans at fan speeds from 100 RPMs to 30,000 RPMs.

Typically, these are DC brushless fans that generate (with each revolution) a 50% duty cycle, two-period square wave, as shown in FIGURE 26-2: below.

FIGURE 26-2: FAN GENERATED 50%DUTY CYCLE WAVEFORM



In typical systems, the fans are powered by the main power supply. Firmware may disable this block when it detects that the main power rail has been turned off by either clearing the <enable> TACH_ENABLE bit or putting the block to sleep via the supported Low Power Mode interface (see Low Power Modes).

26.10.1 MODES OF OPERATION

The Tachometer block supports two modes of operation. The mode of operation is selected via the TACH_READING_-MODE_SELECT bit.

26.10.1.1 Free Running Counter

In Mode 0, the Tachometer block uses the TACHx input as the clock source for the internal TACH pulse counter (see TACHX_COUNTER). The counter is incremented when it detects a rising edge on the TACH input. In this mode, the firmware may periodically poll the TACHX_COUNTER field to determine the average speed over a period of time. The firmware must store the previous reading and the current reading to compute the number of pulses detected over a period of time. In this mode, the counter continuously increments until it reaches FFFFh. It then wraps back to 0000h and continues counting. The firmware must ensure that the sample rate is greater than the time it takes for the counter to wrap back to the starting point.

Note:	Tach interrupts should be disabled in Mode 0.

26.10.1.2 Mode 1 -- Number of Clock Pulses per Revolution

In Mode 1, the Tachometer block uses its 100kHz_Clk clock input to measure the programmable number of TACHx pulses. In this mode, the internal TACH pulse counter (TACHX_COUNTER) returns the value in number of 100kHz_Clk pulses per programmed number of TACH_EDGES. For fans that generate two square waves per revolution, these bits should be configured to five edges.

When the number of edges is detected, the counter is latched and the COUNT_READY_STATUS bit is asserted. If the COUNT_READY_INT_EN bit is set a TACH interrupt event will be generated.

26.10.2 OUT-OF-LIMIT EVENTS

The Tachometer Block has a pair of limit registers that may be configured to generate an event if the Tachometer indicates that the fan is operating too slow or too fast. If the TACHX_COUNTER exceeds one of the programmed limits, the TACHx High Limit Register and the TACHx Low Limit Register, the bit TACH_OUT_OF_LIMIT_STATUS will be set. If the TACH_OUT_OF_LIMIT_STATUS bit is set, the Tachometer block will generate an interrupt event.

26.11 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Tachometer. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 26-2: EC-ONLY REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
TACH	0	EC	32-bit internal address space	0000_6000h
TACH	1	EC	32-bit internal address space	0000_6010h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 26-3: TACHOMETER REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	TACHx Control Register
04h	TACHx Status Register
08h	TACHx High Limit Register
0Ch	TACHx Low Limit Register

26.11.1 TACHX CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	TACHX_COUNTER This 16-bit field contains the latched value of the internal Tach pulse counter, which may be configured by the Tach Reading Mode Select field to operate as a free-running counter or to be gated by the TACHx input signal. If the counter is free-running (Mode 0), the internal Tach counter increments (if enabled) on transitions of the raw TACHx input signal and is latched into this field every time it is incremented. The act of reading this field will not reset the counter, which rolls over to 0000h after FFFFh. The firmware will compute the delta between the current count reading and the previous count reading, to determine the number of pulses detected over a programmed period. If the counter is gated by the TACHx input and clocked by 100kHz_Clk (Mode 1), the internal counter will be latched into the reading register when the programmed number of edges is detected or when the counter reaches FFFFh. The internal counter is reset to zero after it is copied into this register. **Note:** In Mode 1, a counter value of FFFFh means that the Tachometer logic did not detect the programmed number of edges in 655ms. A stuck fan can be detected by setting the TACHx High Limit Register to a number less than FFFFh. If the internal counter then reaches FFFFh, the reading register will be set to FFFFh and an out-of-limit interrupt can be sent to the EC.	R	00h	nSYSRST
15	TACH_INPUT_INT_EN 1=Enable TACHx Input toggle interrupt from Tachometer block 0=Disable TACHx Input toggle interrupt from Tachometer block	R/W	0b	nSYSRST
14	COUNT_READY_INT_EN 1=Enable Count Ready interrupt from Tachometer block 0=Disable Count Ready interrupt from Tachometer block	R/W	0b	nSYSRST
13	Reserved	R	-	-
12:11	TACH_EDGES A tachometer signal is a square wave with a 50% duty cycle. Typically, two tachometer periods represents one revolution of the fan. A tachometer period consists of three edges. This programmed value represents the number of tachometer edges that will be used to determine the interval for which the number of 100kHz_Clk pulses will be counted 11b=9 Tach edges (4 Tach periods) 10b=5 Tach edges (2 Tach periods) 01b=3 Tach edges (1 Tach period) 00b=2 Tach edges (1/2 Tach period)	R/W	00b	nSYSRST

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	TACH_READING_MODE_SELECT 1=Counter is incremented on the rising edge of the 100kHz_Clk input. The counter is latched into the TACHX_COUNTER field and reset when the programmed number of edges is detected. 0=Counter is incremented when TACHx Input transitions from low-to-high state (default)	R/W	0b	nSYSRST
9	Reserved	R	-	-
8	FILTER_ENABLE This filter is used to remove high frequency glitches from TACHx Input. When this filter is enabled, TACHx input pulses less than two 100kHz_Clk periods wide get filtered. 1= Filter enabled 0= Filter disabled (default) It is recommended that the TACHx input filter always be enabled.	R/W	0b	nSYSRST
7:2	Reserved	R	-	-
1	TACH_ENABLE This bit gates the clocks into the block. When clocks are gated, the TACHx pin is tristated. When re-enabled, the internal counters will continue from the last known state and stale status events may still be pending. Firmware should discard any status or reading values until the reading value has been updated at least one time after the enable bit is set. 1= TACH Monitoring enabled, clocks enabled. 0= TACH Idle, clocks gated	R/W	0b	nSYSRST
0	TACH_OUT_OF_LIMIT_ENABLE This bit is used to enable the TACH_OUT_OF_LIMIT_STATUS bit in the TACHx Status Register to generate an interrupt event. 1=Enable interrupt output from Tachometer block 0=Disable interrupt output from Tachometer block (default)	R/W	0b	nSYSRST

26.11.2 TACHX STATUS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	R	-	-
3	COUNT_READY_STATUS This status bit is asserted when the TACHx input changes state and when the counter value is latched. This bit remains cleared to '0' when the TACH_READING_MODE_SELECT bit in the TACHx Control Register is '0'. When the TACH_READING_MODE_SELECT bit in the TACHx Control Register is set to '1', this bit is set to '1' when the counter value is latched by the hardware. It is cleared when written with a '1'. If COUNT_READY_INT_EN in the TACHx Control Register is set to 1, this status bit will assert the Tach Interrupt signal. 1=Reading ready 0=Reading not ready	R/WC	0b	nSYSRST
2	TOGGLE_STATUS This bit is set when TACHx Input changes state. It is cleared when written with a '1'. If TACH_INPUT_INT_EN in the TACHx Control Register is set to '1', this status bit will assert the Tach Interrupt signal. 1=TACHx Input changed state (this bit is set on a low-to-high or high-to-low transition) 0=TACHx stable	R/WC	0b	nSYSRST
1	TACH_PIN_STATUS This bit reflects the state of TACHx Input. This bit is a read only bit that may be polled by the embedded controller. 1= TACHx Input is high 0= TACHx Input is low	R	0b	nSYSRST
0	TACH_OUT_OF_LIMIT_STATUS This bit is set when the Tach Count value is greater than the high limit or less than the low limit. It is cleared when written with a '1'. To disable this status event set the limits to their extreme values. If TACH_OUT_OF_LIMIT_ENABLE in the TACHx Control Register is set to 1', this status bit will assert the Tach Interrupt signal. 1=Tach is outside of limits 0=Tach is within limits	R/WC	0b	nSYSRST

Note:

- Some fans offer a Locked Rotor output pin that generates a level event if a locked rotor is detected. This bit may be used in combination with the Tach pin status bit to detect a locked rotor signal event from a fan.
- Tach Input may come up as active for Locked Rotor events. This would not cause an interrupt event because the pin would not toggle. Firmware must read the status events as part of the initialization process, if polling is not implemented.

26.11.3 TACHX HIGH LIMIT REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	-	-	-
15:0	TACH_HIGH_LIMIT This value is compared with the value in the TACHX_COUNTER field. If the value in the counter is greater than the value programmed in this register, the TACH_OUT_OF_LIMIT_STATUS bit will be set. The TACH_OUT_OF_LIMIT_STATUS status event may be enabled to generate an interrupt to the embedded controller via the TACH_OUT_OF_LIMIT_ENABLE bit in the TACHx Control Register.	R/W	FFFFh	nSYSRST

26.11.4 TACHX LOW LIMIT REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	R	-	-
15:0	TACHX_LOW_LIMIT This value is compared with the value in the TACHX_COUNTER field of the TACHx Control Register. If the value in the counter is less than the value programmed in this register, the TACH_OUT_OF_LIMIT_STATUS bit will be set. The TACH_OUT_OF_LIMIT_STATUS status event may be enabled to generate an interrupt to the embedded controller via the TACH_OUT_OF_LIMIT_ENABLE bit in the TACHx Control Register To disable the TACH_OUT_OF_LIMIT_STATUS low event, program 0000h into this register.	R/W	0000h	nSYSRST

27.0 PWM

27.1 Introduction

This block generates a PWM output that can be used to control 4-wire fans, blinking LEDs, and other similar devices. Each PWM can generate an arbitrary duty cycle output at frequencies from less than 0.1 Hz to 24 MHz. The PWM controller can also used to generate the PROCHOT output and Speaker output.

The PWMx Counter ON Time registers and PWMx Counter OFF Time registers determine the operation of the PWM_OUTPUT signals. See Section 27.11.1, "PWMx Counter ON Time Register," on page 373 and Section 27.11.2, "PWMx Counter OFF Time Register," on page 373 for a description of the PWM_OUTPUT signals.

27.2 References

There are no standards referenced in this chapter.

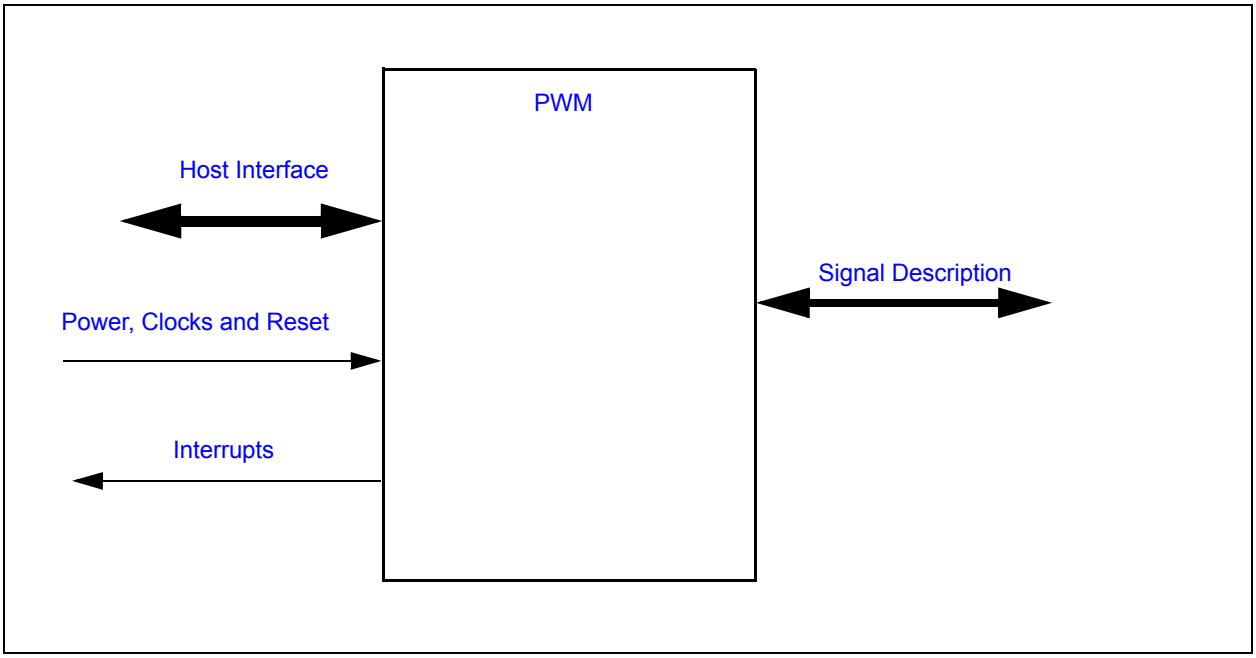
27.3 Terminology

There is no terminology defined for this section.

27.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 27-1: I/O DIAGRAM OF BLOCK



There are no external signals for this block.

27.5 Signal Description

TABLE 27-1: SIGNAL DESCRIPTION

Name	Direction	Description
PWM_OUTPUT	OUTPUT	Pulse Width Modulated signal to PWMx pin.

27.6 Host Interface

The registers defined for the PWM Interface are accessible by the various hosts as indicated in Section 27.11, "EC-Only Registers".

27.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

27.7.1 POWER DOMAINS

Name	Description
VTR	The PWM logic and registers are powered by this single power source.

27.7.2 CLOCK INPUTS

Name	Description
100kHz_Clk	100kHz_Clk clock input for generating low PWM frequencies, such as 10 Hz to 100 Hz.
48 MHz Ring Oscillator	48 MHz Ring Oscillator clock input for generating high PWM frequencies, such as 15 kHz to 30 kHz.

27.7.3 RESETS

Name	Description
nSYSRST	This reset signal resets all the logic in this block to its initial state including the registers, which are set to their defined default state.

27.8 Interrupts

The PWM block does not generate any interrupt events.

27.9 Low Power Modes

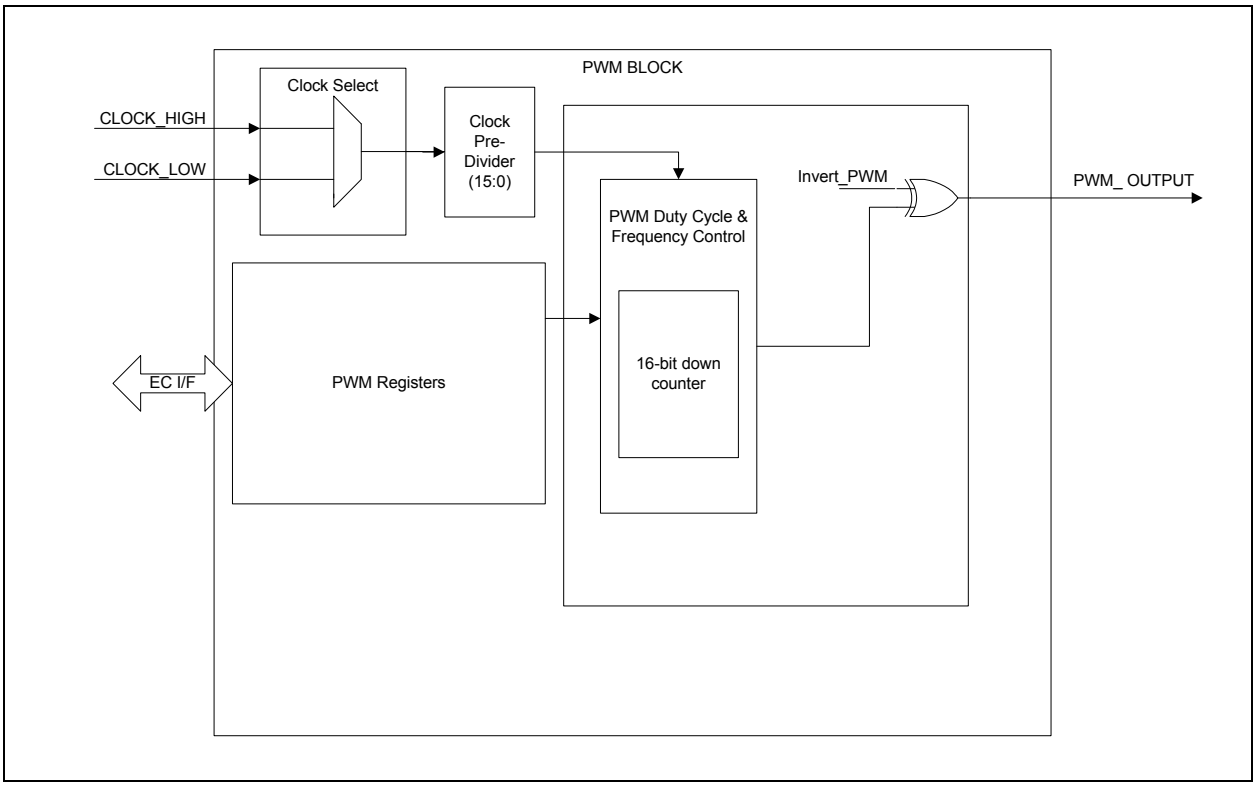
The PWM may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. When the PWM is in the sleep state, the internal counters reset to 0 and the internal state of the PWM and the PWM_OUTPUT signal set to the OFF state.

27.10 Description

The PWM_OUTPUT signal is used to generate a duty cycle of specified frequency. This block can be programmed so that the PWM signal toggles the PWM_OUTPUT, holds it high, or holds it low. When the PWM is configured to toggle, the PWM_OUTPUT alternates from high to low at the rate specified in the PWMx Counter ON Time Register and PWMx Counter OFF Time Register.

The following diagram illustrates how the clock inputs and registers are routed to the PWM Duty Cycle & Frequency Control logic to generate the PWM output.

FIGURE 27-2: BLOCK DIAGRAM OF PWM CONTROLLER



Note: In FIGURE 27-2:, the 48 MHz Ring Oscillator is represented as CLOCK_HIGH and 100kHz_Clk is represented as CLOCK_LOW.

The PWM clock source to the PWM Down Counter, used to generate a duty cycle and frequency on the PWM, is determined through the Clock select[1] and Clock Pre-Divider[6:3] bits in the PWMx Configuration Register register.

The PWMx Counter ON/OFF Time registers determine both the frequency and duty cycle of the signal generated on PWM_OUTPUT as described below.

The PWM frequency is determined by the selected clock source and the total on and off time programmed in the PWMx Counter ON Time Register and PWMx Counter OFF Time Register registers. The frequency is the time it takes (at that clock rate) to count down to 0 from the total on and off time.

The PWM duty cycle is determined by the relative values programmed in the PWMx Counter ON Time Register and PWMx Counter OFF Time Register registers.

The PWM Frequency Equation and PWM Duty Cycle Equation are shown below.

Note: Setting the PWMX_COUNTER_ON_TIME field in the PWMX COUNTER ON TIME REGISTER to a value of n will cause the On time of the PWM to be n+1 cycles of the PWM Clock Source. Setting the PWMX-_COUNTER_OFF_- TIME field in the PWMX COUNTER OFF TIME REGISTER to a value of n will cause the Off time of the PWM to be n+1 cycles of the PWM Clock Source.

FIGURE 27-3: PWM FREQUENCY EQUATION

$$\text{PWM Frequency} = \frac{1}{(PreDivisor + 1)} \times \frac{ClockSourceFrequency}{(PWMCounterOnTime + 1) + (PWMCounterOffTime + 1)}$$

In FIGURE 27-3:, the ClockSourceFrequency variable is the frequency of the clock source selected by the Clock Select bit in the PWMx Configuration Register, and PreDivisor is a field in the PWMx Configuration Register. The PWMCounterOnTime, PWMCounterOffTime are registers that are defined in Section 27.11, "EC-Only Registers".

FIGURE 27-4: PWM DUTY CYCLE EQUATION

$$\text{PWM Duty Cycle} = \frac{PWMCounterOnTime + 1}{(PWMCounterOnTime + 1) + (PWMCounterOffTime + 1)}$$

The PWMx Counter ON Time Register and PWMx Counter OFF Time Register registers should be accessed as 16-bit values.

27.11 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the PWM. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 27-2: EC-ONLY REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
PWM	0	EC	32-bit internal address space	0000_5800h
PWM	1	EC	32-bit internal address space	0000_5810h
PWM	2	EC	32-bit internal address space	0000_5820h
PWM	3	EC	32-bit internal address space	0000_5830h
PWM	4	EC	32-bit internal address space	0000_5840h
PWM	5	EC	32-bit internal address space	0000_5850h
PWM	6	EC	32-bit internal address space	0000_5860h
PWM	7	EC	32-bit internal address space	0000_5870h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 27-3: EC-ONLY REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	PWMx Counter ON Time Register
04h	PWMx Counter OFF Time Register
08h	PWMx Configuration Register

27.11.1 PWMX COUNTER ON TIME REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	R	-	-
15:0	PWMX_COUNTER_ON_TIME This field determine both the frequency and duty cycle of the PWM signal. When this field is set to zero and the PWMX_COUNTER_OFF_-TIME is not set to zero, the PWM_OUTPUT is held low (Full Off).	R/W	0000h	nSYSRST

27.11.2 PWMX COUNTER OFF TIME REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	R	-	-
15:0	PWMX_COUNTER_OFF_TIME This field determine both the frequency and duty cycle of the PWM signal. When this field is set to zero, the PWM_OUTPUT is held high (Full On).	R/W	FFFFh	nSYSRST

27.11.3 PWMX CONFIGURATION REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	R	-	-
6:3	CLOCK_PRE_DIVIDER The Clock source for the 16-bit down counter (see PWMx Counter ON Time Register and PWMx Counter OFF Time Register) is determined by bit D1 of this register. The Clock source is then divided by the value of Pre-Divider+1 and the resulting signal determines the rate at which the down counter will be decremented. For example, a Pre-Divider value of 1 divides the input clock by 2 and a value of 2 divides the input clock by 3. A Pre-Divider of 0 will disable the Pre-Divider option.	R/W	0000b	nSYSRST

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	INVERT 1= PWM_OUTPUT ON State is active low 0=PWM_OUTPUT ON State is active high	R/W	0b	nSYSRST
1	CLOCK_SELECT This bit determines the clock source used by the PWM duty cycle and frequency control logic. 1=CLOCK_LOW 0=CLOCK_HIGH	R/W	0b	nSYSRST
0	PWM_ENABLE 1=Enabled (default) 0=Disabled (gates clocks to save power) **Note:** When the PWM enable bit is set to 0 the internal counters are reset and the internal state machine is set to the OFF state. In addition, the PWM_OUTPUT signal is set to the inactive state as determined by the Invert bit. The PWMx Counter ON Time Register and PWMx Counter OFF Time Register are not affected by the PWM enable bit and may be read and written while the PWM enable bit is 0.	R/W	0b	nSYSRST

28.0 BLINKING/BREATHING PWM

28.1 Introduction

LEDs are used in computer applications to communicate internal state information to a user through a minimal interface. Typical applications will cause an LED to blink at different rates to convey different state information. For example, an LED could be full on, full off, blinking at a rate of once a second, or blinking at a rate of once every four seconds, in order to communicate four different states.

As an alternative to blinking, an LED can “breathe”, that is, oscillate between a bright state and a dim state in a continuous, or apparently continuous manner. The rate of breathing, or the level of brightness at the extremes of the oscillation period, can be used to convey state information to the user that may be more informative, or at least more novel, than traditional blinking.

The blinking/breathing hardware is implemented using a PWM. The PWM can be driven either by the 48 MHz clock or by a 32.768 KHz clock input. When driven by the 48 MHz clock, the PWM can be used as a standard 8-bit PWM in order to control a fan. When used to drive blinking or breathing LEDs, the 32.768 KHz clock source is used.

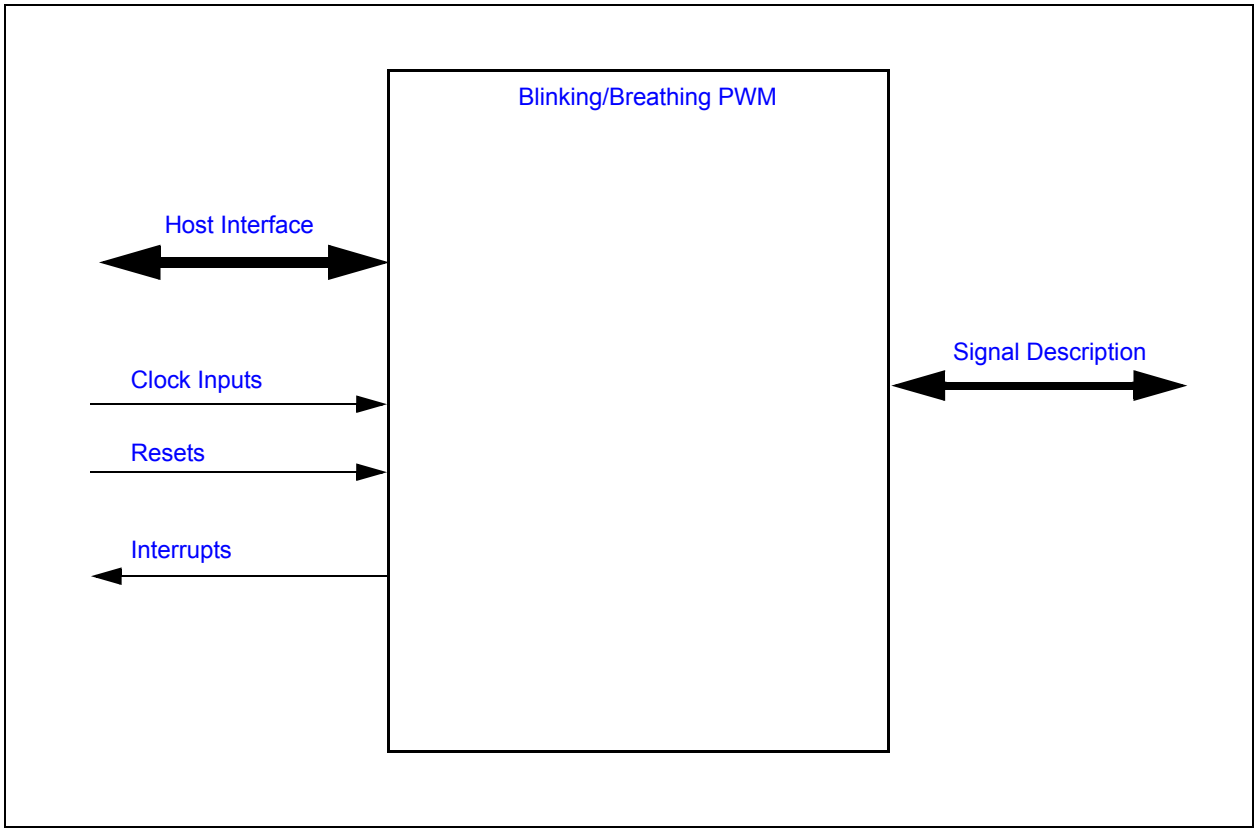
Features:

- Each PWM independently configurable
- Each PWM configurable for LED blinking and breathing output
- Highly configurable breathing rate from 60ms to 1min
- Non-linear brightness curves approximated with 8 piece wise-linear segments
- All LED PWMs can be synchronized
- Each PWM configurable for 8-bit PWM support
- Multiple clock rates
- Configurable Watchdog Timer

28.2 Interface

This block is designed to drive a pin on the pin interface and to be accessed internally via a registered host interface.

FIGURE 28-1: I/O DIAGRAM OF BLOCK



28.3 Signal Description

TABLE 28-1: SIGNAL DESCRIPTION

Name	Direction	Description
PWM Output (a.k.a. LEDx, where x represents the instantiation)	Output	Output of PWM By default, the PWM pin is configured to be active high: when the PWM is configured to be fully on, the pin is driving high. When the PWM is configured to be fully off, the pin is low. If firmware requires the Blinking/Breathing PWM to be active low, the Polarity bit in the GPIO Pin Control Register associated with the PWM can be set to 1, which inverts the output polarity.

28.4 Host Interface

The blinking/breathing PWM block is accessed by a controller over the standard register interface.

28.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

28.5.1 POWER DOMAINS

Name	Description
VTR	Main power. The source of main power for the device is system dependent.

28.5.2 CLOCK INPUTS

Name	Description
5Hz_Clk	32.768 KHz clock
48 MHz Ring Oscillator	48 MHz clock

28.5.3 RESETS

Name	Description
nSYSRST	Block reset

28.6 Interrupts

Each PWM can generate an interrupt. The interrupt is asserted for one 48 MHz clock period whenever the PWM WDT times out. The PWM WDT is described in Section 28.8.3.1, "PWM WDT," on page 382.

Source	Description
PWM_WDT	PWM watchdog time out

28.7 Low Power Mode

The Blinking/Breathing PWM may be put into a low power mode by the chip-level power, clocks, and reset (PCR) circuitry. The low power mode is only applicable when the Blinking/Breathing PWM is operating in the General Purpose PWM mode. When the low speed clock mode is selected, the blinking/breathing function continues to operate, even when the 48 MHz clock is stopped. Low power mode behavior is summarized in the following table:

TABLE 28-2: LOW POWER MODE BEHAVIOR

CLOCK_ SOURCE	CONTROL	Mode	Low Power Mode	Description
X	'00'b	PWM 'OFF'	Yes	32.768 KHz clock is required.
X	'01'b	Breathing	Yes	
1	'10'b	General Purpose PWM	No	48 MHz clock is required, even when a sleep command to the block is asserted.
0	'10'b	Blinking	Yes	32.768 KHz clock is required.
X	'11'b	PWM 'ON'	Yes	

Note: In order for the MEC140x/1x to enter its heavy and deep sleep states, the SLEEP_ENABLE input for all Blinking/Breathing PWM instances must be asserted, even if the PWMs are configured to use the low speed clock.

28.8 Description

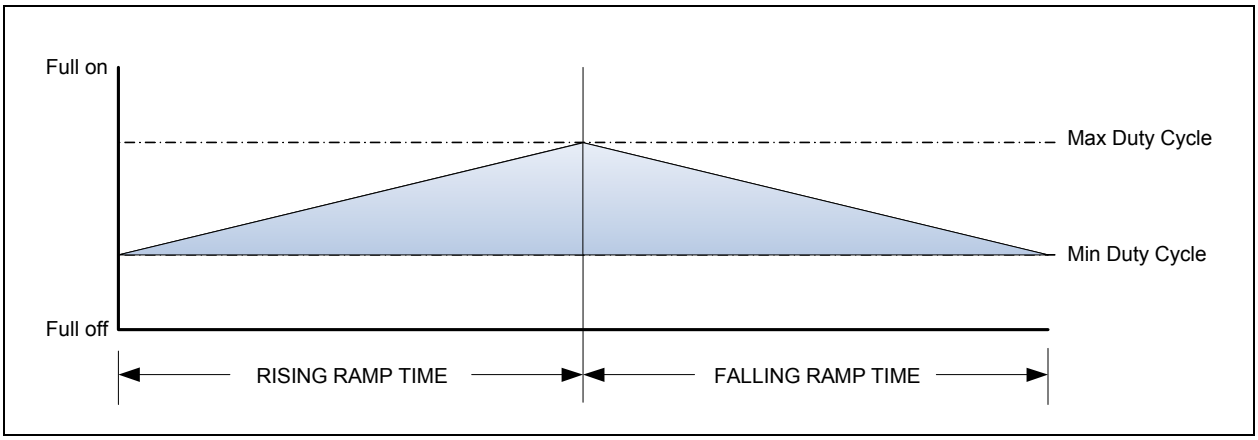
28.8.1 BREATHING

If an LED blinks rapidly enough, the eye will interpret the light as reduced brightness, rather than a blinking pattern. Therefore, if the blinking period is short enough, modifying the duty cycle will set the apparent brightness, rather than a blinking rate. At a blinking rate of 128Hz or greater, almost all people will perceive a continuous light source rather than an intermittent pattern.

Because making an LED appear to breathe is an aesthetic effect, the breathing mechanism must be adjustable or customers may find the breathing effect unattractive. There are several variables that can affect breathing appearance, as described below.

The following figure illustrates some of the variables in breathing:

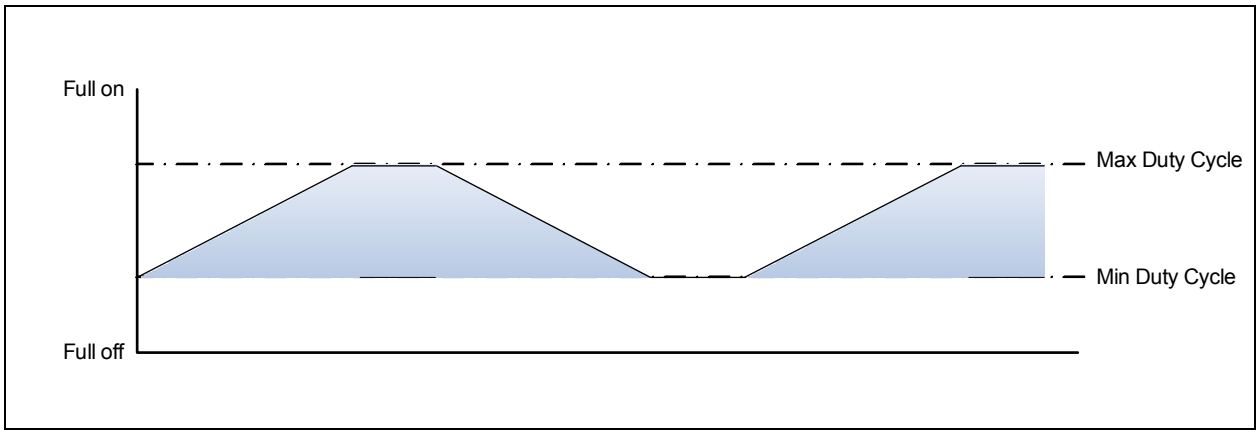
FIGURE 28-2: BREATHING LED EXAMPLE



The breathing range of and LED can range between full on and full off, or in a range that falls within the full-on/full-off range, as shown in this figure. The ramp time can be different in different applications. For example, if the ramp time was 1 second, the LED would appear to breathe quickly. A time of 2 seconds would make the LED appear to breathe more leisurely.

The breathing pattern can be clipped, as shown in the following figure, so that the breathing effect appears to pause at its maximum and minimum brightnesses:

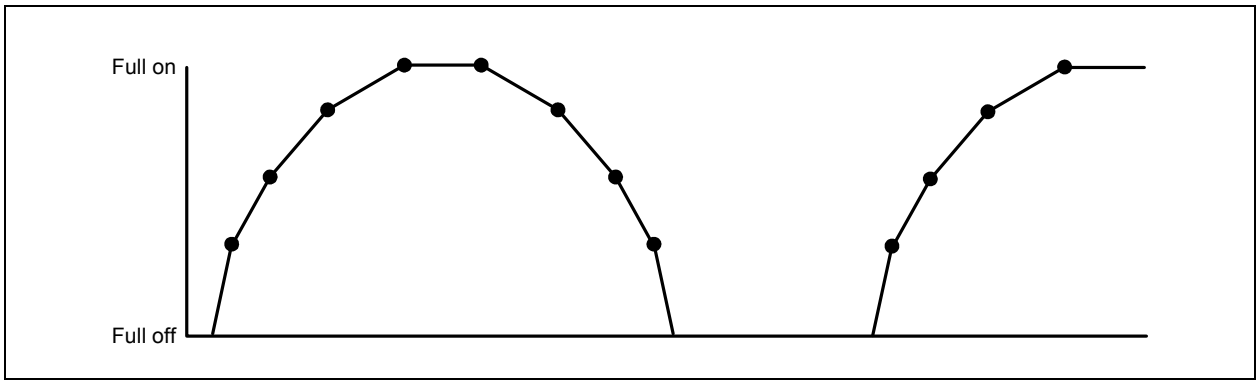
FIGURE 28-3: CLIPPING EXAMPLE



The clipping periods at the two extremes can be adjusted independently, so that for example an LED can appear to breathe (with a short delay at maximum brightness) followed by a longer “resting” period (with a long delay at minimum brightness).

The brightness can also be changed in a non-linear fashion, as shown in the following figure:

FIGURE 28-4: EXAMPLE OF A SEGMENTED CURVE



In this figure, the rise and fall curves are implemented in 4 linear segments and are the rise and fall periods are symmetric.

The breathing mode uses the 32.768 KHz clock for its time base.

28.8.2 BLINKING

When configured for blinking, a subset of the hardware used in breathing is used to implement the blinking function. The PWM (an 8-bit accumulator plus an 8-bit duty cycle register) drives the LED directly. The Duty Cycle register is programmed directly by the user, and not modified further. The PWM accumulator is configured as a simple 8-bit up counter. The counter uses the 32.768 KHz clock, and is pre-scaled by the Delay counter, to slow the PWM down from the 128Hz provided by directly running the PWM on the 32.768 KHz clock.

With the pre-scaler, the blink rate of the LED could be as fast as 128Hz (which, because it is blinking faster than the eye can distinguish, would appear as a continuous level) to 0.03125Hz (that is, with a period of 7.8ms to 32 seconds). Any duty cycle from 0% (0h) to 100% (FFh) can be configured, with an 8-bit precision. An LED with a duty cycle value of 0h will be fully off, while an LED with a duty cycle value of FFh will be fully on.

In Blinking mode the PWM counter is always in 8-bit mode.

Table 28-3, "LED Blink Configuration Examples" shows some example blinking configurations:

TABLE 28-3: LED BLINK CONFIGURATION EXAMPLES

Prescale	Duty Cycle	Blink Frequency	Blink
000h	00h	128Hz	full off
000h	FFh	128Hz	full on
001h	40h	64Hz	3.9ms on, 11.6ms off
003h	80h	32Hz	15.5ms on, 15.5ms off
07Fh	20h	1Hz	125ms on, 0.875s off
0BFh	16h	0.66Hz	125ms on, 1.375s off
0FFh	10h	0.5Hz	125ms on, 1.875s off
180h	0Bh	0.33Hz	125ms on, 2.875s off
1FFh	40h	0.25Hz	1s on, 3s off

The Blinking and General Purpose PWM modes share the hardware used in the breathing mode. The Prescale value is derived from the LD field of the LED_DELAY register and the Duty Cycle is derived from the MIN field of the LED_LIMITS register.

TABLE 28-4: BLINKING MODE CALCULATIONS

Parameter	Unit	Equation
Frequency	Hz	(5Hz_Clk frequency) /(PRESCALE + 1)/255
'H' Width	Seconds	(1/PERIOD) x (DutyCycle/255)
'L' Width	Seconds	(1/PERIOD) x (255 - DutyCycle)

28.8.3 GENERAL PURPOSE PWM

When used in the Blinking configuration with the 48 MHz Ring Oscillator, the LED module can be used as a general-purpose programmable Pulse-Width Modulator with an 8-bit programmable pulse width. It can be used for fan speed control, sound volume, etc. With the 48 MHz Ring Oscillator source, the PWM frequency can be configured in the range shown in TABLE 28-5:.

TABLE 28-5: PWM CONFIGURATION EXAMPLES

Prescale	PWM Frequency
000h	187.5 KHz
001h	93.75 KHz
003h	46.875 KHz
006h	26.8 KHz
00Bh	15.625 KHz
07Fh	1.46 KHz

TABLE 28-5: PWM CONFIGURATION EXAMPLES (CONTINUED)

Prescale	PWM Frequency
1FFh	366 Hz
FFFh	46 Hz

TABLE 28-6: GENERAL PURPOSE PWM MODE CALCULATIONS

Parameter	Unit	Equation
Frequency	Hz	(48 MHz Ring Oscillator frequency) / (PRESCALE + 1) / 255
'H' Width	Seconds	(1/PERIOD) x (DutyCycle/255)
'L' Width	Seconds	(1/PERIOD) x (255 - DutyCycle)

28.8.3.1 PWM WDT

When the PWM is configured as a general-purpose PWM (in the Blinking configuration with the 48 MHz clock), the PWM includes a Watch Dog Timer (WDT). The WDT consists of an internal 8-bit counter and an 8-bit reload value (the field WDTLD in LED Configuration Register register). The internal counter is loaded with the reset value of WDTLD (14h, or 4 seconds) on system nSYSRST and loaded with the contents of WDTLD whenever either the LED Configuration Register register is written or the MIN byte in the LED Limits Register register is written (the MIN byte controls the duty cycle of the PWM).

Whenever the internal counter is non-zero, it is decremented by 1 for every tick of the 5 Hz clock. If the counter decrements from 1 to 0, a WDT Terminal Count causes an interrupt to be generated and reset sets the CONTROL bit in the LED Configuration Register to 3h, which forces the PWM to be full on. No other PWM registers or fields are affected.

If the 5 Hz clock halts, the watchdog timer stops decrementing but retains its value, provided the device continues to be powered. When the 5 Hz clock restarts, the watchdog counter will continue decrementing where it left off.

Setting the WDTLD bits to 0 disables the PWM WDT. Other sample values for WDTLD are:

01h = 200 ms

02h = 400 ms

03h = 600 ms

04h = 800 ms

. . .

14h = 4seconds

FFh = 51 seconds

28.9 Implementation

In addition to the registers described in Section 28.10, "EC-Only Registers", the PWM is implemented using a number of components that are interconnected differently when configured for breathing operation and when configured for blinking/PWM operation.

28.9.1 BREATHING CONFIGURATION

The **PSIZE** parameter can configure the PWM to one of three modes: 8-bit, 7-bit and 6-bit. The **PERIOD CTR** counts ticks of its input clock. In 8-bit mode, it counts from 0 to 255 (that is, 256 steps), then repeats continuously. In this mode, a full cycle takes 7.8ms (128Hz). In 7-bit mode it counts from 0 to 127 (128 steps), and a full cycle takes 3.9ms (256Hz). In 6-bit mode it counts from 0 to 63 (64 steps) and a full cycle takes 1.95ms (512Hz).

The output of the LED circuit is asserted whenever the **PERIOD CTR** is less than the contents of the **DUTY CYCLE** register. The appearance of breathing is created by modifying the contents of the **DUTY CYCLE** register in a continuous manner. When the LED control is off the internal counters and registers are all reset to 0 (i.e. after a write setting the RESET bit in the LED Configuration Register Register.) Once enabled, the **DUTY CYCLE** register is increased by an

amount determined by the LED_STEP register and at a rate determined by the **DELAY** counter. Once the duty cycle reaches its maximum value (determined by the field MAX), the duty cycle is held constant for a period determined by the field HD. Once the hold time is complete, the **DUTY CYCLE** register is decreased, again by an amount determined by the LED_STEP register and at a rate determined by the **DELAY** counter. When the duty cycle then falls at or below the minimum value (determined by the field MIN), the duty cycle is held constant for a period determined by the field HD. Once the hold time is complete, the cycle repeats, with the duty cycle oscillating between MIN and MAX.

The rising and falling ramp times as shown in on page 379 can be either symmetric or asymmetric depending on the setting of the SYMMETRY bit in the LED Configuration Register Register. In Symmetric mode the rising and falling ramp rates have mirror symmetry; both rising and falling ramp rates use the same (all) 8 segments fields in each of the following registers (see TABLE 28-7:): the LED Update Stepsize Register register and the LED Update Interval Register register. In Asymmetric mode the rising ramp rate uses 4 of the 8 segments fields and the falling ramp rate uses the remaining 4 of the 8 segments fields (see TABLE 28-7:).

The parameters MIN, MAX, HD, LD and the 8 fields in LED_STEP and LED_INT determine the brightness range of the LED and the rate at which its brightness changes. See the descriptions of the fields in Section 28.10, "EC-Only Registers", as well as the examples in Section 28.9.3, "Breathing Examples" for information on how to set these fields.

TABLE 28-7: SYMMETRIC BREATHING MODE REGISTER USAGE

Rising/ Falling Ramp Times in Figure 28-3, "Clipping Example"	Duty Cycle	Segment Index	Symmetric Mode Register Fields Utilized	
X	000xxxxxb	000b	STEP[0]/INT[0]	Bits[3:0]
X	001xxxxxb	001b	STEP[1]/INT[1]	Bits[7:4]
X	010xxxxxb	010b	STEP[2]/INT[2]	Bits[11:8]
X	011xxxxxb	011b	STEP[3]/INT[3]	Bits[15:12]
X	100xxxxxb	100b	STEP[4]/INT[4]	Bits[19:16]
X	101xxxxxb	101b	STEP[5]/INT[5]	Bits[23:20]
X	110xxxxxb	110b	STEP[6]/INT[6]	Bits[27:24]
X	111xxxxxb	111b	STEP[7]/INT[7]	Bits[31:28]
Note: In Symmetric Mode the Segment_Index[2:0] = Duty Cycle Bits[7:5]				

TABLE 28-8: ASYMMETRIC BREATHING MODE REGISTER USAGE

Rising/ Falling Ramp Times in Figure 28-3, "Clipping Example"	Duty Cycle	Segment Index	Asymmetric Mode Register Fields Utilized	
Rising	00xxxxxxb	000b	STEP[0]/INT[0]	Bits[3:0]
Rising	01xxxxxxb	001b	STEP[1]/INT[1]	Bits[7:4]
Rising	10xxxxxxb	010b	STEP[2]/INT[2]	Bits[11:8]
Rising	11xxxxxxb	011b	STEP[3]/INT[3]	Bits[15:12]
falling	00xxxxxxb	100b	STEP[4]/INT[4]	Bits[19:16]
falling	01xxxxxxb	101b	STEP[5]/INT[5]	Bits[23:20]

TABLE 28-8: ASYMMETRIC BREATHING MODE REGISTER USAGE (CONTINUED)

Rising/ Falling Ramp Times in Figure 28-3, "Clipping Example"	Duty Cycle	Segment Index	Asymmetric Mode Register Fields Utilized	
falling	10xxxxxxb	110b	STEP[6]/INT[6]	Bits[27:24]
falling	11xxxxxxb	111b	STEP[7]/INT[7]	Bits[31:28]
Note: In Asymmetric Mode the Segment_Index[2:0] is the bit concatenation of following: Segment_Index[2] = (FALLING RAMP TIME in Figure 28-3, "Clipping Example") and Segment_Index[1:0] = Duty Cycle Bits[7:6].				

28.9.2 BLINKING CONFIGURATION

The Delay counter and the PWM counter are the same as in the breathing configuration, except in this configuration they are connected differently. The Delay counter is clocked on either the 32.768 KHz clock or the 48 MHz clock, rather than the output of the PWM. The PWM counter is clocked by the zero output of the Delay counter, which functions as a prescalar for the input clocks to the PWM. The Delay counter is reloaded from the LD field of the LED_DELAY register. When the LD field is 0 the input clock is passed directly to the PWM counter without prescaling. In Blinking/PWM mode the PWM counter is always 8-bit, and the PSIZE parameter has no effect.

The frequency of the PWM pulse waveform is determined by the formula:

$$f_{PWM} = \frac{f_{clock}}{(256 \times (LD + 1))}$$

where f_{PWM} is the frequency of the PWM, f_{clock} is the frequency of the input clock (32.768 KHz clock or 48 MHz clock) and LD is the contents of the LD field.

Note: At a duty cycle value of 00h (in the MIN register), the LED output is fully off. At a duty cycle value of 255h, the LED output is fully on. Alternatively, In order to force the LED to be fully on, firmware can set the CONTROL field of the Configuration register to 3 (always on).

The other registers in the block do not affect the PWM or the LED output in Blinking/PWM mode.

28.9.3 BREATHING EXAMPLES

28.9.3.1 Linear LED brightness change

In this example, the brightness of the LED increases and diminishes in a linear fashion. The entire cycle takes 5 seconds. The rise time and fall time are 1.6 seconds, with a hold time at maximum brightness of 200ms and a hold time at minimum brightness of 1.6 seconds. The LED brightness varies between full off and full on. The PWM size is set to 8-bit, so the time unit for adjusting the PWM is approximately 8ms. The registers are configured as follows:

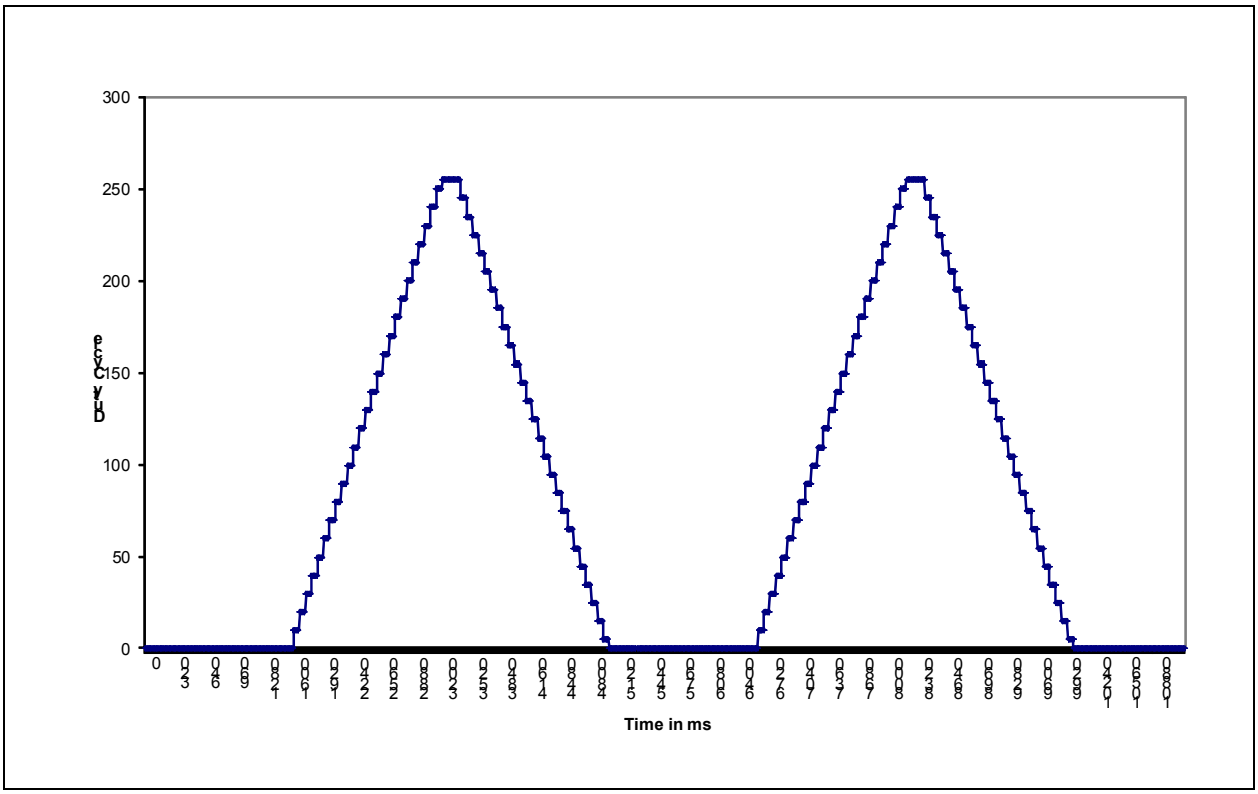
TABLE 28-9: LINEAR EXAMPLE CONFIGURATION

Field	Value
PSIZE	8-bit
MAX	255
MIN	0
HD	25 ticks (200ms)
LD	200 ticks (1.6s)

TABLE 28-9: LINEAR EXAMPLE CONFIGURATION (CONTINUED)

Field	Value							
Duty cycle most significant bits	000b	001b	010b	011b	100b	101b	110b	1110
LED_INT	8	8	8	8	8	8	8	8
LED_STEP	10	10	10	10	10	10	10	10

FIGURE 28-5: LINEAR BRIGHTNESS CURVE EXAMPLE



28.9.3.2 Non-linear LED brightness change

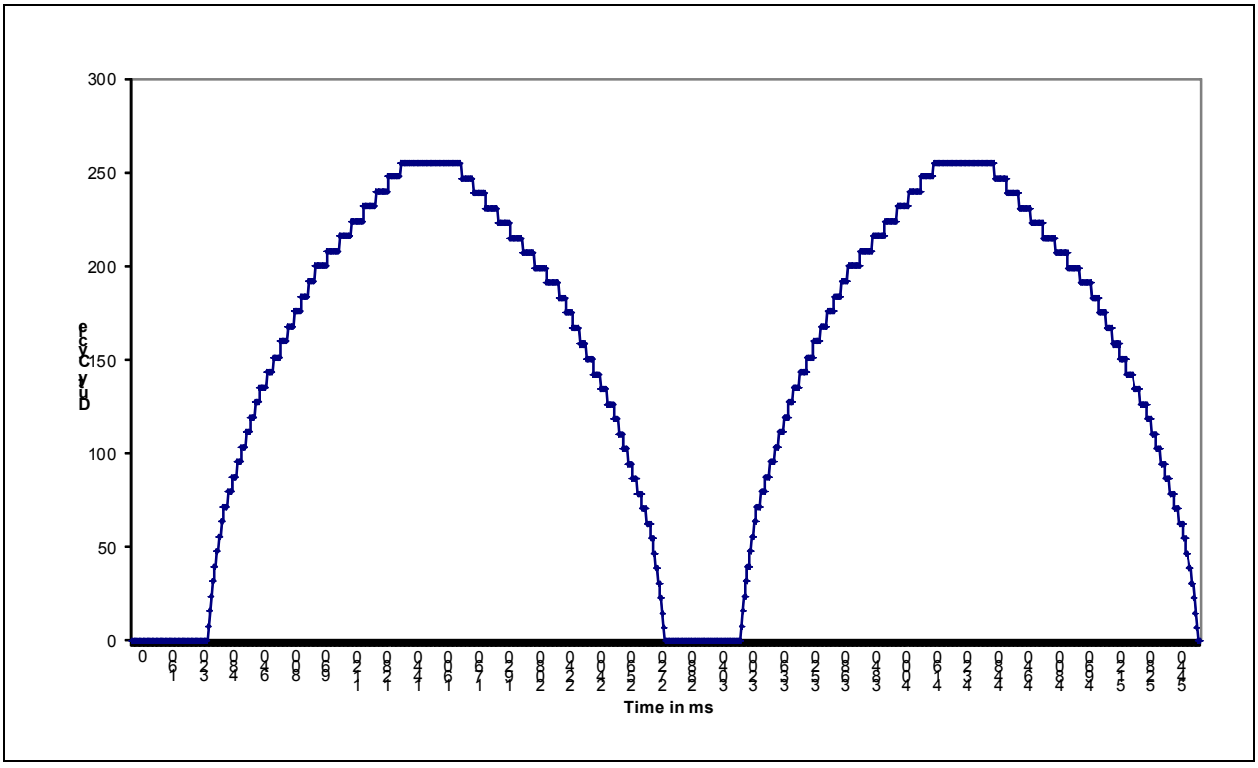
In this example, the brightness of the LED increases and diminishes in a non-linear fashion. The brightness forms a curve that is approximated by four piece wise-linear line segments. The entire cycle takes about 2.8 seconds. The rise time and fall time are about 1 second, with a hold time at maximum brightness of 320ms and a hold time at minimum brightness of 400ms. The LED brightness varies between full off and full on. The PWM size is set to 7-bit, so the time unit for adjusting the PWM is approximately 4ms. The registers are configured as follows:

TABLE 28-10: NON-LINEAR EXAMPLE CONFIGURATION

Field	Value							
PSIZE	7-bit							
MAX	255 (effectively 127)							
MIN	0							
HD	80 ticks (320ms)							
LD	100 ticks (400ms)							
Duty cycle most significant bits	000b	001b	010b	011b	100b	101b	110b	1110
LED_INT	2	3	6	6	9	9	16	16
LED_STEP	4	4	4	4	4	4	4	4

The resulting curve is shown in the following figure:

FIGURE 28-6: NON-LINEAR BRIGHTNESS CURVE EXAMPLE



28.10 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Blinking/Breathing PWM. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 28-11: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
Blinking/Breathing PWM	0	EC	32-bit internal address space	0000_B800h
Blinking/Breathing PWM	1	EC	32-bit internal address space	0000_B900h
Blinking/Breathing PWM	2	EC	32-bit internal address space	0000_BA00h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 28-12: EC-ONLY REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	LED Configuration Register
04h	LED Limits Register
08h	LED Delay Register
0Ch	LED Update Stepsize Register
10h	LED Update Interval Register

In the following register definitions, a “PWM period” is defined by time the PWM counter goes from 000h to its maximum value (FFh in 8-bit mode, FEh in 7-bit mode and FCh in 6-bit mode, as defined by the PSCALE field in register LED_CFG). The end of a PWM period occurs when the PWM counter wraps from its maximum value to 0.

The registers in this block can be written 32-bits, 16-bits or 8-bits at a time. Writes to LED Configuration Register take effect immediately. Writes to LED Limits Register are held in a holding register and only take effect only at the end of a PWM period. The update takes place at the end of every period, even if only one byte of the register was updated. This means that in blink/PWM mode, software can change the duty cycle with a single 8-bit write to the MIN field in the LED_LIMIT register. Writes to LED Delay Register, LED Update Stepsize Register and LED Update Interval Register also go initially into a holding register. The holding registers are copied to the operating registers at the end of a PWM period only if the Enable Update bit in the LED Configuration Register is set to 1. If LED_CFG is 0, data in the holding registers is retained but not copied to the operating registers when the PWM period expires. To change an LED breathing configuration, software should write these three registers with the desired values and then set LED_CFG to 1. This mechanism ensures that all parameters affecting LED breathing will be updated consistently, even if the registers are only written 8 bits at a time.

28.10.1 LED CONFIGURATION REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	R	-	-
16	SYMMETRY 1=The rising and falling ramp times are in Asymmetric mode. Table 28-8, "Asymmetric Breathing Mode Register Usage" shows the application of the Stepsize and Interval registers to the four segments of rising duty cycles and the four segments of falling duty cycles. 0=The rising and falling ramp times (as shown in Figure 28-2, "Breathing LED Example") are in Symmetric mode. Table 28-7, "Symmetric Breathing Mode Register Usage" shows the application of the Stepsize and Interval registers to the 8 segments of both rising and falling duty cycles.	R/W	0b	nSYSRST
15:8	WDT_RELOAD The PWM Watchdog Timer counter reload value. On system reset, it defaults to 14h, which corresponds to a 4 second Watchdog timeout value.	R/W	14h	nSYSRST
7	RESET Writes of'1' to this bit resets the PWM registers to their default values. This bit is self clearing. Writes of '0' to this bit have no effect.	W	0b	nSYSRST
6	ENABLE_UPDATE This bit is set to 1 when written with a '1'. Writes of '0' have no effect. Hardware clears this bit to 0 when the breathing configuration registers are updated at the end of a PWM period. The current state of the bit is readable any time. This bit is used to enable consistent configuration of LED_DELAY, LED_STEP and LED_INT. As long as this bit is 0, data written to those three registers is retained in a holding register. When this bit is 1, data in the holding register are copied to the operating registers at the end of a PWM period. When the copy completes, hardware clears this bit to 0.	R/WS	0b	nSYSRST
5:4	PWM_SIZE This bit controls the behavior of PWM: 3=Reserved 2=PWM is configured as a 6-bit PWM 1=PWM is configured as a 7-bit PWM 0=PWM is configured as an 8-bit PWM	R/W	0b	nSYSRST

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	SYNCHRONIZE When this bit is '1', all counters for all LEDs are reset to their initial values. When this bit is '0' in the LED Configuration Register for all LEDs, then all counters for LEDs that are configured to blink or breathe will increment or decrement, as required. To synchronize blinking or breathing, the SYNCHRONIZE bit should be set for at least one LED, the control registers for each LED should be set to their required values, then the SYNCHRONIZE bits should all be cleared. If the all LEDs are set for the same blink period, they will all be synchronized.	R/W	0b	nSYSRST
2	CLOCK_SOURCE This bit controls the base clock for the PWM. It is only valid when CNTRL is set to blink (2). 1=Clock source is the 48 MHz clock 0=Clock source is the 32.768 KHz clock	R/W	0b	nSYSRST
1:0	CONTROL This bit controls the behavior of PWM: 3=PWM is always on 2=LED blinking (standard PWM) 1=LED breathing configuration 0=PWM is always off. All internal registers and counters are reset to 0. Clocks are gated	R/W	00b	nSYSRST
			11b	WDT TC

28.10.2 LED LIMITS REGISTER

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period. The two byte fields may be written independently. Reads of this register return the current contents and not the value of the holding register.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	R	-	-
15:8	MAXIMUM In breathing mode, when the current duty cycle is greater than or equal to this value the breathing apparatus holds the current duty cycle for the period specified by the field HD in register LED_DELAY, then starts decrementing the current duty cycle	R/W	0h	nSYSRST
7:0	MINIMUM In breathing mode, when the current duty cycle is less than or equal to this value the breathing apparatus holds the current duty cycle for the period specified by the field LD in register LED_DELAY, then starts incrementing the current duty cycle In blinking mode, this field defines the duty cycle of the blink function.	R/W	0h	nSYSRST

28.10.3 LED DELAY REGISTER

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period if the Enable Update bit in the LED Configuration register is set to 1. Reads of this register return the current contents and not the value of the holding register.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	R	-	-
23:12	HIGH_DELAY In breathing mode, the number of PWM periods to wait before updating the current duty cycle when the current duty cycle is greater than or equal to the value MAX in register LED_LIMIT. 4095=The current duty cycle is decremented after 4096 PWM periods … 1=The delay counter is bypassed and the current duty cycle is decremented after two PWM period 0=The delay counter is bypassed and the current duty cycle is decremented after one PWM period	R/W	000h	nSYSRST
11:0	LOW_DELAY The number of PWM periods to wait before updating the current duty cycle when the current duty cycle is greater than or equal to the value MIN in register LED_LIMIT. 4095=The current duty cycle is incremented after 4096 PWM periods … 0=The delay counter is bypassed and the current duty cycle is incremented after one PWM period In blinking mode, this field defines the prescalar for the PWM clock	R/W	000h	nSYSRST

28.10.4 LED UPDATE STEPSIZE REGISTER

This register has eight segment fields which provide the amount the current duty cycle is adjusted at the end of every PWM period. Segment field selection is decoded based on the segment index. The segment index equation utilized depends on the SYMMETRY bit in the LED Configuration Register Register).

- In Symmetric Mode the Segment_Index[2:0] = Duty Cycle Bits[7:5]
- In Asymmetric Mode the Segment_Index[2:0] is the bit concatenation of following: Segment_Index[2] = (FALLING RAMP TIME in Figure 28-3, "Clipping Example") and Segment_Index[1:0] = Duty Cycle Bits[7:6].

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period if the Enable Update bit in the LED Configuration register is set to 1. Reads of this register return the current contents and not the value of the holding register.

In 8-bit mode, each 4-bit STEPSIZE field represents 16 possible duty cycle modifications, from 1 to 16 as the duty cycle is modified between 0 and 255:

15: Modify the duty cycle by 16

...

1: Modify the duty cycle by 2

0: Modify the duty cycle by 1

In 7-bit mode, the least significant bit of the 4-bit field is ignored, so each field represents 8 possible duty cycle modifications, from 1 to 8, as the duty cycle is modified between 0 and 127:

14, 15: Modify the duty cycle by 8

...

2, 3: Modify the duty cycle by 2

0, 1: Modify the duty cycle by 1

In 6-bit mode, the two least significant bits of the 4-bit field is ignored, so each field represents 4 possible duty cycle modifications, from 1 to 4 as the duty cycle is modified between 0 and 63:

12, 13, 14, 15: Modify the duty cycle by 4

8, 9, 10, 11: Modify the duty cycle by 3

4, 5, 6, 7: Modify the duty cycle by 2

0, 1, 2, 3: Modify the duty cycle by 1

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	UPDATE_STEP7 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 111.	R/W	0h	nSYSRST
27:24	UPDATE_STEP6 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 110.	R/W	0h	nSYSRST
23:20	UPDATE_STEP5 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 101	R/W	0h	nSYSRST
19:16	UPDATE_STEP4 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 100.	R/W	0h	nSYSRST
15:12	UPDATE_STEP3 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 011.	R/W	0h	nSYSRST
11:8	UPDATE_STEP2 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 010.	R/W	0h	nSYSRST
7:4	UPDATE_STEP1 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 001.	R/W	0h	nSYSRST
3:0	UPDATE_STEP0 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 000.	R/W	0h	nSYSRST

28.10.5 LED UPDATE INTERVAL REGISTER

This register has eight segment fields which provide the number of PWM periods between updates to current duty cycle. Segment field selection is decoded based on the segment index. The segment index equation utilized depends on the SYMMETRY bit in the LED Configuration Register Register)

- In Symmetric Mode the Segment_Index[2:0] = Duty Cycle Bits[7:5]
- In Asymmetric Mode the Segment_Index[2:0] is the bit concatenation of following: Segment_Index[2] = (FALLING RAMP TIME in Figure 28-3, "Clipping Example") and Segment_Index[1:0] = Duty Cycle Bits[7:6].

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period if the Enable Update bit in the LED Configuration register is set to 1. Reads of this register return the current contents and not the value of the holding register.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	UPDATE_INTERVAL7 The number of PWM periods between updates to current duty cycle when the segment index is equal to 111b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	nSYSRST
27:24	UPDATE_INTERVAL6 The number of PWM periods between updates to current duty cycle when the segment index is equal to 110b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	nSYSRST
23:20	UPDATE_INTERVAL5 The number of PWM periods between updates to current duty cycle when the segment index is equal to 101b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	nSYSRST
19:16	UPDATE_INTERVAL4 The number of PWM periods between updates to current duty cycle when the segment index is equal to 100b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	nSYSRST
15:12	UPDATE_INTERVAL3 The number of PWM periods between updates to current duty cycle when the segment index is equal to 011b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	nSYSRST
11:8	UPDATE_INTERVAL2 The number of PWM periods between updates to current duty cycle when the segment index is equal to 010b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	nSYSRST

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:4	UPDATE_INTERVAL1 The number of PWM periods between updates to current duty cycle when the segment index is equal to 001b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	nSYSRST
3:0	UPDATE_INTERVAL0 The number of PWM periods between updates to current duty cycle when the segment index is equal to 000b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	nSYSRST

29.0 PS/2 INTERFACE

29.1 Introduction

The PS/2 Interface may be used to communicate with a PS/2 keyboard or a PS/2 mouse. The physical interface provides the clock and data signaling for PS/2 data transfers. The PS/2 Controllers are directly controlled by the EC. The hardware implementation eliminates the need to bit bang I/O ports to generate PS/2 traffic, however bit banging is available via the associated GPIO pins.

29.2 References

No references have been cited for this feature.

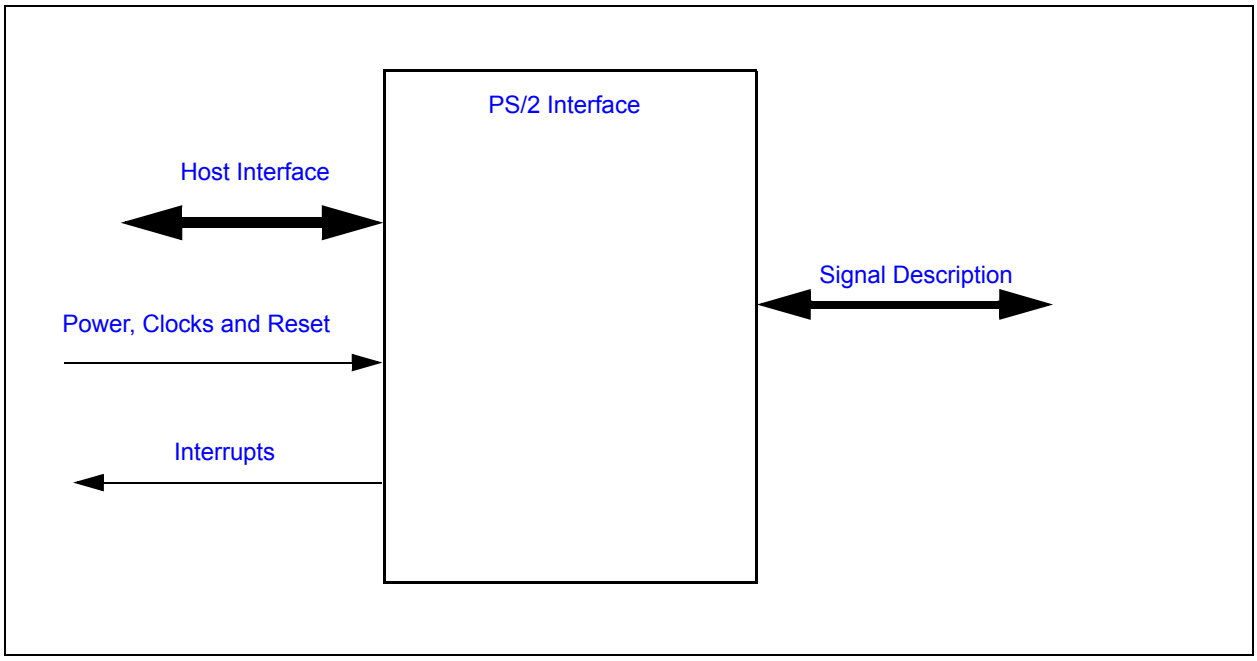
29.3 Terminology

There is no terminology defined for this section.

29.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 29-1: I/O DIAGRAM OF BLOCK



29.5 Signal Description

TABLE 29-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
PS2_DAT	INPUT/ OUTPUT	Data from the PS/2 device
PS2_CLK	INPUT/ OUTPUT	Clock from the PS/2 device

29.6 Host Interface

The registers defined for the Keyboard Scan Interface are accessible by the various hosts as indicated in Section 29.15, "EC-Only Registers".

29.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

29.7.1 POWER DOMAINS

Name	Description
VTR	The logic and registers implemented in this block are powered by this power well.

29.7.2 CLOCK INPUTS

Name	Description
48 MHz Ring Oscillator	This is the clock source for PS/2 Interface logic.
2 MHz Clock	The PS/2 state machine is clocked using the 2 MHz clock.

29.7.3 RESETS

Name	Description
nSYSRST	This signal resets all the registers and logic in this block to their default state.

29.8 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
PS2_ACT	Interrupt request to the Interrupt Aggregator for PS2 controller instance *x*, based on PS2 controller activity. Section 29.15.4, "PS2 Status Register" defines the sources for the interrupt request.
PS2_DAT*x*_WAKE	Wake-up request to the Interrupt Aggregator's wake-up interface for PS2 port *x*. In order to enable PS2 wakeup interrupts, the pin control registers for the PS2_DAT pin must be programmed to Input, Falling Edge Triggered, non-inverted polarity detection.

29.9 Low Power Modes

The PS/2 Interface may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

The PS2 interface will only sleep while the PS2 is disabled or in Rx mode with no traffic on the bus.

29.10 Description

Each EC PS/2 serial channels use a synchronous serial protocol to communicate with the auxiliary device. Each PS/2 channel has Clock and Data signal lines. The signal lines are bi-directional and employ open drain outputs capable of sinking 12m, as required by the PS/2 specification. A pull-up resistor, typically 10K, is connected to both lines. This allows either the EC PS/2 logic or the auxiliary device to drive the lines. Regardless of the drive source, the auxiliary device always provides the clock for transmit and receive operations. The serial packet is made up of eleven bits, listed in the order they appear on the data line: start bit, eight data bits (least significant bit first), odd parity, and stop bit. Each bit cell is from 60µS to 100µS long.

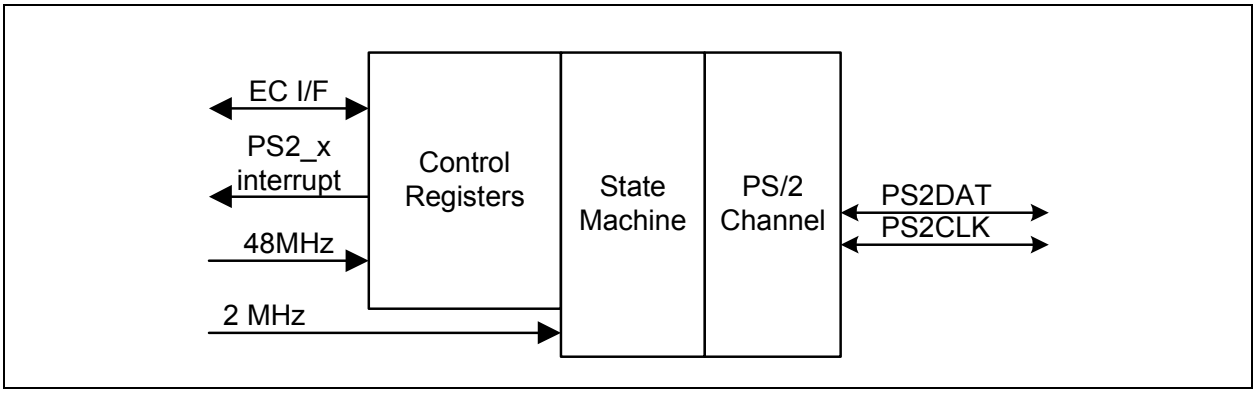
All PS/2 Serial Channel signals (PS2_CLK and PS2_DAT) are driven by open drain drivers which can be pulled to VTR or the main power rail (+3.3V nominal) through 10K-ohm resistors.

The PS/2 controller supports a PS/2 Wake Interface that can wake the EC from the IDLE or SLEEP states. The Wake Interface can generate wake interrupts without a clock. The PS/2 Wake Interface is only active when the peripheral device and external pull-up resisters are powered by the VTR supply.

There are no special precautions to be taken to prevent back drive of a PS/2 peripheral powered by the main power well when the power well is off, as long as the external 10K pull-up resistor is tied to the same power source as the peripheral.

29.11 Block Diagram

FIGURE 29-2: PORT PS/2 BLOCK DIAGRAM



29.12 PS/2 Port Physical Layer Byte Transmission Protocol

The PS/2 physical layer transfers a byte of data via an eleven bit serial stream as shown in TABLE 29-2:. A logic 1 is sent at an active high level. Data sent from a Keyboard or mouse device to the host is read on the falling edge of the clock signal. The Keyboard or mouse device always generates the clock signal. The Host may inhibit communication by pulling the Clock line low. The Clock line must be continuously high for at least 50 microseconds before the Keyboard or mouse device can begin to transmit its data. See Table 29-3, "PS/2 Port Physical Layer Bus States".

TABLE 29-2: PS/2 PORT PHYSICAL LAYER BYTE TRANSMISSION PROTOCOL

Bit	Function
1	Start bit (always 0)
2	Data bit 0 (least significant bit)
3	Data bit 1
4	Data bit 2
5	Data bit 3
6	Data bit 4
7	Data bit 5
8	Data bit 6
9	Data bit 7 (most significant bit)
10	Parity bit (odd parity)
11	Stop Bit (always 1)

FIGURE 29-3: PS/2 PORT PHYSICAL LAYER BYTE TRANSMISSION PROTOCOL

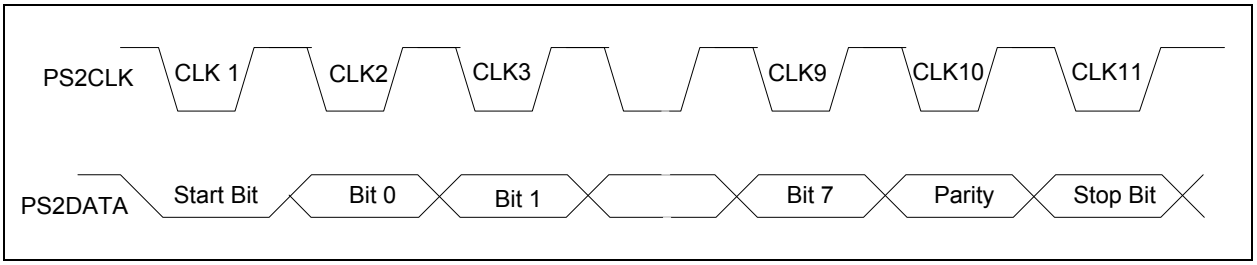


TABLE 29-3: PS/2 PORT PHYSICAL LAYER BUS STATES

Data	Clock	State
high	high	Idle
high	low	Communication Inhibited
low	low	Request to Send

29.13 Controlling PS/2 Transactions

PS/2 transfers are controlled by fields in the PS2 Control Register.

The interface is enabled by the PS2_EN bit. Transfers are enabled when PS2_EN is ‘1’ and disabled when PS2_EN is ‘0’. If the PS2_EN bit is cleared to ‘0’ while a transfer is in progress but prior to the leading edge (falling edge) of the 10th (parity bit) clock edge, the receive data is discarded (RDATA_RDY remains low). If the PS2_EN bit is cleared following the leading edge of the 10th clock signal, then the receive data is saved in the Receive Register (RDATA_RDY goes high) assuming no parity error.

The direction of a PS/2 transfer is controlled by the PS2_T/R bit.

29.13.1 RECEIVE

If PS2_T/R is ‘0’ while the PS2 Interface is enabled, the interface is configured to receive data. If while PS2_T/R is ‘0’ RDATA_RDY is ‘0’, the channel’s PS2_CLK and PS2_DAT will float waiting for the external PS/2 device to signal the start of a transmission. If RDATA_RDY is ‘1’, the channel’s PS2_DAT line will float but its PS2_CLK line will be held low, holding off the peripheral, until the Receive Register is read.

The peripheral initiates a reception by sending a start bit followed by the data bits). After a successful reception, data are placed in the PS2 Receive Buffer Register, the RDATA_RDY bit in the PS2 Status Register is set and the PS2_CLK line is forced low. Further receive transfers are inhibited until the EC reads the data in the PS2 Receive Buffer Register. RDATA_RDY is cleared and the PS2_CLK line is tri-stated following a read of the PS2 Receive Buffer Register.

The Receive Buffer Register is initialized to FFh after a read or after a Time-out has occurred.

29.13.2 TRANSMIT

If PS2_T/R is ‘1’ while the PS2 Interface is enabled, the interface is configured to transmit data. When the PS2_T/R bit is written to ‘1’ while the state machine is idle, the channel prepares for a transmission: the interface will drive the PS2_CLK line low and then float the PS2_DAT line, holding this state until a write occurs to the Transmit Register or until the PS2_T/R bit is cleared. A transmission is started by writing the PS2 Transmit Buffer Register. Writes to the Transmit Buffer Register are blocked when PS2_EN is ‘0’, PS2_T/R is ‘0’ or when the transmit state machine is active (the XMIT_IDLE bit in the PS/2 Status Register is ‘0’). The transmission of data will not start if there is valid data in the Receive Data Register (when the status bit RDATA_RDY is ‘1’). When a transmission is started, the transmission state machine becomes active (the XMIT_IDLE bit is set to ‘1’ by hardware), the PS2_DAT line is driven low and within 80ns the PS2_CLK line floats (externally pulled high by the pull-up resistor).

The transmission terminates either on the 11th clock edge of the transmission or if a Transmit Time-Out error condition occurs. When the transmission terminates, the PS2_T/R bit is cleared to ‘0’and the state machine becomes idle, setting XMIT_IDLE to ‘1’.

The PS2_T/R bit must be written to a ‘1’ before initiating another transmission to the remote device. If the PS2_T/R bit is set to ‘1’ while the channel is actively receiving data (that is, while the status bit RDATA_RDYis ‘1’) prior to the leading edge of the 10th (parity bit) clock edge, the receive data is discarded. If the bit is set after the 10th edge, the receive data is saved in the Receive Register.

29.14 Instance Description

29.15 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the PS/2 Interface. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 29-4: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
PS/2 Interface	0	EC	32-bit internal address space	0000_9000h
	1	EC	32-bit internal address space	0000_9040h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 29-5: EC-ONLY REGISTER SUMMARY

Offset	Register Name
0h	PS2 Transmit Buffer Register
0h	PS2 Receive Buffer Register
4h	PS2 Control Register
8h	PS2 Status Register

29.15.1 PS2 TRANSMIT BUFFER REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7:0	TRANSMIT_DATA Writes to this register start a transmission of the data in this register to the peripheral.	W	0h	nSYSRST

29.15.2 PS2 RECEIVE BUFFER REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7:0	RECEIVE_DATA Data received from a peripheral are recorded in this register. A transmission initiated by writing the PS2 Transmit Buffer Register will not start until valid data in this register have been read and RDATA_RDY has been cleared by hardware. The Receive Buffer Register is initialized to FFh after a read or after a Time-out has occurred.	R	FFh	nSYSRST

29.15.3 PS2 CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	R	-	-
5:4	STOP These bits are used to set the level of the stop bit expected by the PS/2 channel state machine. These bits are therefore only valid when PS2_EN is set. 00b=Receiver expects an active high stop bit. 01b=Receiver expects an active low stop bit. 10b=Receiver ignores the level of the Stop bit (11th bit is not interpreted as a stop bit). 11b=Reserved.	R/W	0h	nSYSRST
3:2	PARITY These bits are used to set the parity expected by the PS/2 channel state machine. These bits are therefore only valid when PS2_EN is set. 00b=Receiver expects Odd Parity (default). 01b=Receiver expects Even Parity. 10b=Receiver ignores level of the parity bit (10th bit is not interpreted as a parity bit). 11b=Reserved	R/W	0h	nSYSRST

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	PS2_EN PS/2 Enable. 0=The PS/2 state machine is disabled. The CLK pin is driven low and the DATA pin is tri-stated. 1=The PS/2 state machine is enabled, allowing the channel to perform automatic reception or transmission, depending on the state of PS2_T/R.	R/W	0h	nSYSRST
0	PS2_T/R PS/2 Transmit/Receive 0=The P2/2 channel is enabled to receive data. 1=The PS2 channel is enabled to transmit data.	R/W	0h	nSYSRST

Changing values in the PS2 CONTROL REGISTER at a rate faster than 2 MHz, may result in unpredictable behavior.

29.15.4 PS2 STATUS REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7	XMIT_START_TIMEOUT Transmit Start Timeout. 0=No transmit start timeout detected 1=A start bit was not received within 25 ms following the transmit start event. The transmit start bit time-out condition is also indicated by the XMIT_TIMEOUT bit.	R/WC	0h	nSYSRST
6	RX_BUSY Receive Channel Busy. 0=The channel is actively receiving PS/2 data 1=The channel is idle	R	0h	nSYSRST
5	XMIT_TIME_OUT Transmitter Idle. When the XMIT_TIMEOUT bit is set, the PS2_T/R bit is held clear, the PS/2 channel's CLK line is pulled low for a minimum of 300µs until the PS/2 Status register is read. The XMIT_TIMEOUT bit is set on one of three transmit conditions: when the transmitter bit time (the time between falling edges) exceeds 300µs, when the transmitter start bit is not received within 25ms from signaling a transmit start event or if the time from the first bit (start) to the 10th bit (parity) exceeds 2ms	R/WC	0h	nSYSRST
4	XMIT_IDLE Transmitter Idle. 0=The channel is actively transmitting PS/2 data. Writing the PS2 Transmit Buffer Register will cause the XMIT_IDLE bit to clear 1=The channel is not transmitting. This bit transitions from '0' to '1' in the following cases: • The falling edge of the 11th CLK • XMIT_TIMEOUT is set • The PS2_T/R bit is cleared • The PS2_EN bit is cleared. A low to high transition on this bit generates a PS2 Activity interrupt.	R	0h	nSYSRST
3	FE Framing Error When receiving data, the stop bit is clocked in on the falling edge of the 11th CLK edge. If the channel is configured to expect either a high or low stop bit and the 11th bit is contrary to the expected stop polarity, then the FE and REC_TIMEOUT bits are set following the falling edge of the 11th CLK edge and an interrupt is generated.	R/WC	0h	nSYSRST

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	PE Parity Error When receiving data, the parity bit is clocked in on the falling edge of the 10th CLK edge. If the channel is configured to expect either even or odd parity and the 10th bit is contrary to the expected parity, then the PE and REC_TIMEOUT bits are set following the falling edge of the 10th CLK edge and an interrupt is generated.	R/WC	0h	nSYSRST
1	REC_TIMEOUT Receive Timeout Following assertion of the REC_TIMEOUT bit, the channel's CLK line is automatically pulled low for a minimum of 300us until the PS/2 status register is read. Under PS2 automatic operation, PS2_EN is set, this bit is set on one of three receive error conditions: • When the receiver bit time (the time between falling edges) exceeds 300µs. • If the time from the first bit (start) to the 10th bit (parity) exceeds 2ms. • On a receive parity error along with the Parity Error (PE) bit. • On a receive framing error due to an incorrect STOP bit along with the framing error (FE) bit. A low to high transition on this bit generates a PS2 Activity interrupt.	R/WC	0h	nSYSRST
0	RDATA_RDY Receive Data Ready Under normal operating conditions, this bit is set following the falling edge of the 11th clock given successful reception of a data byte from the PS/2 peripheral (i.e., no parity, framing, or receive timeout errors) and indicates that the received data byte is available to be read from the Receive Register. This bit may also be set in the event that the PS2_EN bit is cleared following the 10th CLK edge. Reading the Receive Register clears this bit. A low to high transition on this bit generates a PS2 Activity interrupt.	R	0h	nSYSRST

30.0 KEYBOARD MATRIX SCAN INTERFACE

30.1 Overview

The Keyboard Matrix Scan Interface block provides a register interface to the EC to directly scan an external keyboard matrix of size up to 18x8.

The maximum configuration of the Keyboard Matrix Scan Interface is 18 outputs by 8 inputs. For a smaller matrix size, firmware should configure unused KSO pins as GPIOs or another alternate function, and it should mask out unused KSIs and associated interrupts.

30.2 References

No references have been cited for this feature.

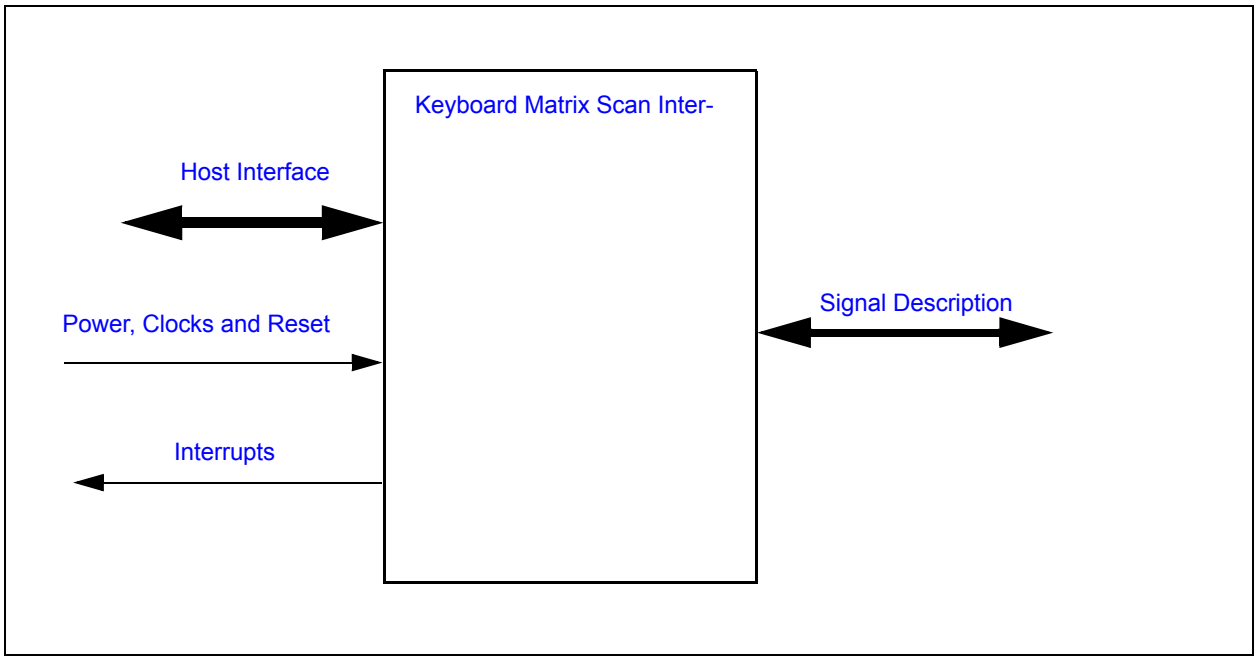
30.3 Terminology

There is no terminology defined for this section.

30.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 30-1: I/O DIAGRAM OF BLOCK



30.5 Signal Description

TABLE 30-1: SIGNAL DESCRIPTION

Name	Direction	Description
KSI[7:0]	Input	Column inputs from external keyboard matrix.
KSO[17:0]	Output	Row outputs to external keyboard matrix.

Note:	Pull-up resistors are required on both the KSI and KSO pins. Either external 10k ohm resistors or the internal resistors may be used. However, if the internal pull-ups are used then the PreDrive Mode must also be enabled.

30.6 Host Interface

The registers defined for the Keyboard Scan Interface are accessible by the various hosts as indicated in Section 30.11, "EC-Only Registers".

30.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

30.7.1 POWER DOMAINS

Name	Description
VTR	The logic and registers implemented in this block are powered by this power well.

30.7.2 CLOCK INPUTS

Name	Description
48 MHz Ring Oscillator	This is the clock source for Keyboard Scan Interface logic.

30.7.3 RESETS

Name	Description
nSYSRST	This signal resets all the registers and logic in this block to their default state.

30.8 Interrupts

This section defines the Interrupt Sources generated from this block.

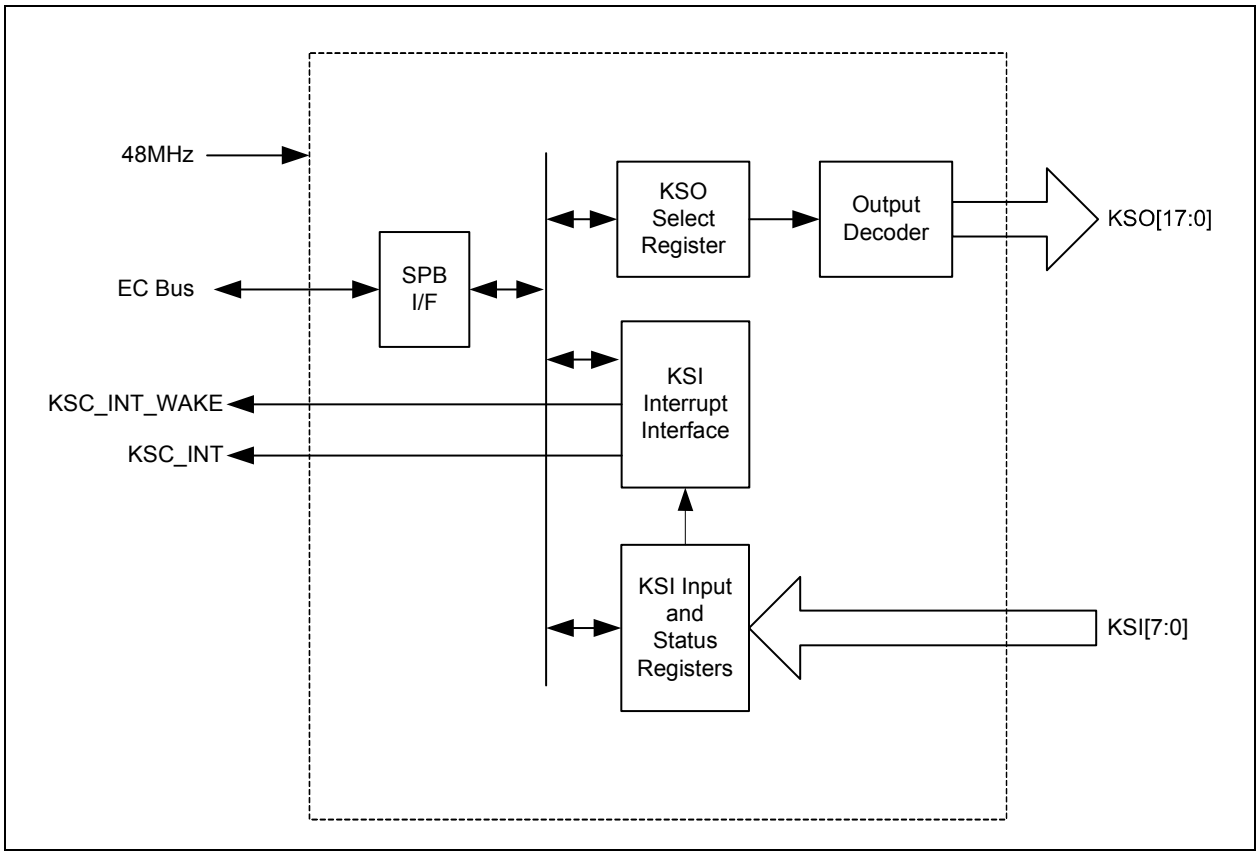
Source	Description
KSC_INT	Interrupt request to the Interrupt Aggregator.
KSC_INT_WAKE	Wake-up request to the Interrupt Aggregator’s wake-up interface.

30.9 Low Power Modes

The Keyboard Scan Interface automatically enters a low power mode whenever it is not actively scanning the keyboard matrix. The block is also placed in a low-power state when it is disabled by the KSEN bit. When the interface is in a low-power mode it will not prevent the chip from entering a sleep state. When the interface is active it will inhibit the chip sleep state until the interface has re-entered its low power mode.

30.10 Description

FIGURE 30-2: Keyboard Matrix Scan Interface Block Diagram



During scanning the firmware sequentially drives low one of the rows (KSO[17:0]) and then reads the column data line (KSI[7:0]). A key press is detected as a zero in the corresponding position in the matrix. Keys that are pressed are debounced by firmware. Once confirmed, the corresponding keycode is loaded into host data read buffer in the 8042 Host Interface module. Firmware may need to buffer keycodes in memory in case this interface is stalled or the host requests a Resend.

30.10.1 INITIALIZATION OF KSO PINS

If the Keyboard Scan Interface is not configured for PREDRIVE Mode, KSO pins should be configured as open-drain outputs. Internal or external pull-ups should be used so that the GPIO functions that share the pins do not have a floating input when the KSO pins are tri-stated.

If the Keyboard Scan Interface is configured for PREDRIVE Mode, KSO pins must be configured as push-pull outputs. Internal or external pull-ups should be used to protect the GPIO inputs associated with the KSO pins from floating inputs.

30.10.2 PREDRIVE MODE

There is an optional Predrive Mode that can be enabled to actively drive the KSO pins high before switching to open-drain operation. The PREDRIVE ENABLE bit in the Keyscan Extended Control Register is used to enable the PRE-DRIVE option. Timing for the Predive mode is shown in Section 43.16, Keyboard Scan Matrix Timing.

30.10.2.1 Predrive Mode Programming

The following precautions should be taken to prevent output pad damage during Predrive Mode Programming.

30.10.2.2 Asserting PREDRIVE_ENABLE

1. Disable Key Scan Interface (KSEN = '1')
2. Enable Predrive function (PREDRIVE_ENABLE = '1')
3. Program buffer type for all KSO pins to "push-pull”
4. Enable Keyscan Interface (KSEN ='0')

30.10.2.3 De-asserting PREDRIVE_ENABLE

1. Disable Key Scan Interface (KSEN = '1')
2. Program buffer type for all KSO pins to "open-drain”
3. Disable Predrive function (PREDRIVE_ENABLE = '0')
4. Enable Keyscan Interface (KSEN ='0')

30.10.3 INTERRUPT GENERATION

To support interrupt-based processing, an interrupt can optionally be generated on the high-to-low transition on any of the KSI inputs. A running clock is not required to generate interrupts.

30.10.3.1 Runtime interrupt

KSC_INT is the block’s runtime active-high level interrupt. It is connected to the interrupt interface of the Interrupt Aggregator, which then relays interrupts to the EC.

Associated with each KSI input is a status register bit and an interrupt enable register bit. A status bit is set when the associated KSI input goes from high to low. If the interrupt enable bit for that input is set, an interrupt is generated. An Interrupt is de-asserted when the status bit and/or interrupt enable bit is clear. A status bit cleared when written to a ‘1’.

Interrupts from individual KSIs are logically ORed together to drive the KSC_INT output port. Once asserted, an interrupt is not asserted again until either all KSI[7:0] inputs have returned high or the has changed.

30.10.3.2 Wake-up interrupt

KSC_INT_WAKE is the block’s wakeup interrupt. It is routed to the Interrupt Aggregator.

During sleep mode, i.e., when the bus clock is stopped, a high-to-low transition on any KSI whose interrupt enable bit is set causes the KSC_INT_WAKE to be asserted. Also set is the associated status bit in the EC Clock Required 2 Status Register (EC_CLK_REQ2_STS). KSC_WAKEUP_INT remains active until the bus clock is started.

The aforementioned transition on KSI also sets the corresponding status bit in the KSI STATUS Register. If enabled, a runtime interrupt is also asserted on KSC_INT when the bus clock resumes running.

30.10.4 WAKE PROGRAMMING

Using the Keyboard Scan Interface to ‘wake’ the MEC140x/1x can be accomplished using either the Keyboard Scan Interface wake interrupt, or using the wake capabilities of the GPIO Interface pins that are multiplexed with the Keyboard Scan Interface pins. Enabling the Keyboard Scan Interface wake interrupt requires only a single interrupt enable access and is recommended over using the GPIO Interface for this purpose.

30.11 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Keyboard Scan Interface. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 30-2: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
Keyboard Scan Interface	0	EC	32-bit internal address space	0000_9C00h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 30-3: EC-ONLY REGISTER SUMMARY

Offset	Register Name
0h	Reserved
4h	KSO Select Register
8h	KSI INPUT Register
Ch	KSI STATUS Register
10h	KSI INTERRUPT ENABLE Register
14h	Keyscan Extended Control Register

30.11.1 KSO SELECT REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	R	-	-
7	KSO_INVERT This bit controls the output level of KSO pins when selected. If all KSO pins are active by programming KSO_SELECT to a value greater than 11h the KSO_INVERT controls the state of the pins as follows: 1= All KSOs driven low when selected 0= All KSOs driven high when selected. Otherwise, if a single KSO line is selected via the KSO_SELECT bit field or all KSO lines are selected via the KSO_ALL bit then the KSO_INVERT controls the state of the pins as follows: 0= KSO[x] driven low when selected 1= KSO[x] driven high when selected. **Note:** The active state of the KSO pins is determined by the KSO_INVERT bit as is shown in Table 30-5, "Keyboard Scan Out Control Summary"	R/W	0b	nSYSRST
6	KSEN This field enables and disables keyboard scan 0= Keyboard scan enabled 1= Keyboard scan disabled. All KSO output buffers disabled.	R/W	1h	nSYSRST

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
5	KSO_ALL 0 = When key scan is enabled, KSO output controlled by the KSO_SELECT field. 1 = All KSO pins are active and the KSO_SELECT field is a don't care. **Note:** The active state is determined by the KSO_INVERT bit as is shown in Table 30-5, "Keyboard Scan Out Control Summary"	R/W	0b	nSYSRST
4:0	KSO_SELECT This field determines which KSO line(s) are active. 0_0000b = KSO00 Selected 0_0001b = KSO01 Selected . . . 1_0001b = KSO17 Selected 1_0010b - 1_1111b = All KSO pins selected **Note:** The full decode table is illustrated in Table 30-4, "KSO Select Decode" **Note:** The active state is determined by the KSO_INVERT bit as is shown in Table 30-5, "Keyboard Scan Out Control Summary"	R/W	0h	nSYSRST

TABLE 30-4: KSO SELECT DECODE

KSO Select [4:0]	KSO Selected
00h	KSO00
01h	KSO01
02h	KSO02
03h	KSO03
04h	KSO04
05h	KSO05
06h	KSO06
07h	KSO07
08h	KSO08
09h	KSO09
0Ah	KSO10

TABLE 30-4: KSO SELECT DECODE (CONTINUED)

KSO Select [4:0]	KSO Selected
0Bh	KSO11
0Ch	KSO12
0Dh	KSO13
0Eh	KSO14
0Fh	KSO15
10h	KSO16
11h	KSO17

TABLE 30-5: KEYBOARD SCAN OUT CONTROL SUMMARY

KSO_INVERT	KSEN	KSO_ALL	KSO_SELECT	Description
x	1	x	x	Keyboard Scan disabled. KSO[17:0] output buffers disabled.
0	0	0	10001b-00000b	KSO[Selected] driven low. All others driven high
1	0	0	10001b-00000b	KSO[Selected] driven high. All others driven low
0	0	0	11111b-10010b	All KSO's driven high
1	0	0	11111b-10010b	All KSO's driven low
0	0	1	x	All KSO's driven low
1	0	1	x	All KSO's driven high

30.11.2 KSI INPUT REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7:0	KSI This field returns the current state of the KSI pins.	R	0h	nSYSRST

30.11.3 KSI STATUS REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7:0	KSI_STATUS Each bit in this field is set on the falling edge of the corresponding KSI input pin. A KSI interrupt is generated when its corresponding status bit and interrupt enable bit are both set. KSI interrupts are logically ORed together to produce KSC_INT and KSC_INT_WAKE. Writing a '1' to a bit will clear it. Writing a '0' to a bit has no effect.	R/WC	0h	nSYSRST

30.11.4 KSI INTERRUPT ENABLE REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7:0	KSI_INT_EN Each bit in KSI_INT_EN enables interrupt generation due to high-to-low transition on a KSI input. An interrupt is generated when the corresponding bits in KSI_STATUS and KSI_INT_EN are both set.	R/W	0h	nSYSRST

30.11.5 KEYSCAN EXTENDED CONTROL REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
32:1	Reserved	R	-	-
0	PREDRIVE_ENABLE PREDRIVE_ENABLE enables the PREDRIVE mode to actively drive the KSO pins high for approximately 100 ns before switching to open-drain operation. 0=Disable predrive on KSO pins 1=Enable predrive on KSO pins.	RW	0	nSYSRST

31.0 BC-LINK MASTER

31.1 Overview

This block provides BC-Link connectivity to a slave device. The BC-Link protocol includes a start bit to signal the beginning of a message and a turnaround (TAR) period for bus transfer between the Master and Companion devices.

31.2 References

No references have been cited for this feature.

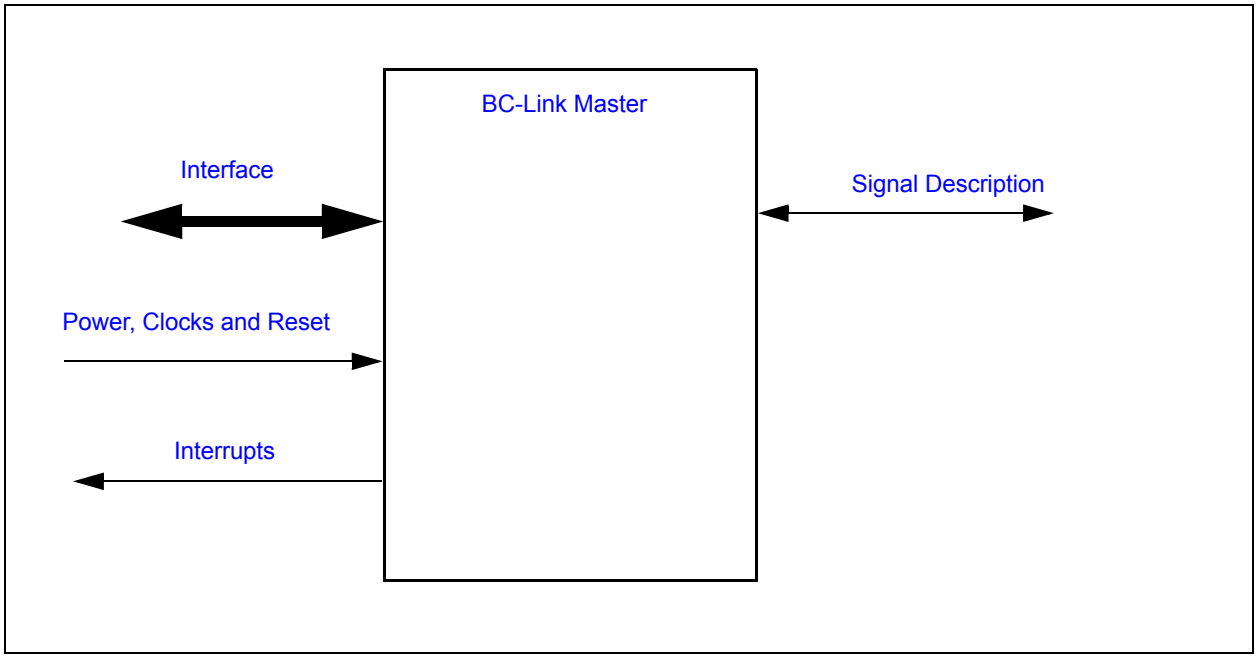
31.3 Terminology

There is no terminology defined for this section.

31.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 31-1: I/O DIAGRAM OF BLOCK



31.5 Signal Description

Note: 'x' in the Pin Name represents the peripheral instance number.

TABLE 31-1: SIGNAL DESCRIPTION

Name	Direction	Description
BCM_CLKx	Output	BC-Link output clock
BCM_DATx	Input/Output	Bidirectional data line
BCM_INTx#	Input	Input from the companion device

Note: A weak pull-up resistor is recommended on the data line (100KΩ).

The maximum speed at which the BC-Link Master Interface can operate reliably depends on the drive strength of the BC-Link BCM_CLK and BCM_DAT pins, as well as the nature of the connection to the Companion device (over ribbon cable or on a PC board). The following table shows the recommended maximum speeds over a PC board as well as a 12 inch ribbon cable for selected drive strengths. The frequency is set with the BC-Link Clock Select Register.

TABLE 31-2: BC-LINK MASTER PIN DRIVE STRENGTH VS. FREQUENCY

Pin Drive Strength	Max Freq on PC Board	Min Value in BC-Link Clock Select Register	Max Freq over Ribbon cable	Min Value in BC-Link Clock Select Register
16mA	24Mhz	1	16Mhz	2

31.6 Host Interface

The registers defined for the BC-Link Master Interface are accessible by the various hosts as indicated in Section 31.11, "EC-Only Registers".

31.7 Power, Clocks and Reset

31.7.1 POWER DOMAINS

Name	Description
VTR	The logic and registers implemented in this block are powered by this power well.

31.7.2 CLOCK INPUTS

Name	Description
48 MHz Ring Oscillator	This is the clock source for Keyboard Scan Interface logic.

31.7.3 RESETS

Name	Description
nSYSRST	This signal resets all the registers and logic in this block to their default state.

31.8 Interrupts

This section defines the Interrupt Sources generated from this block.

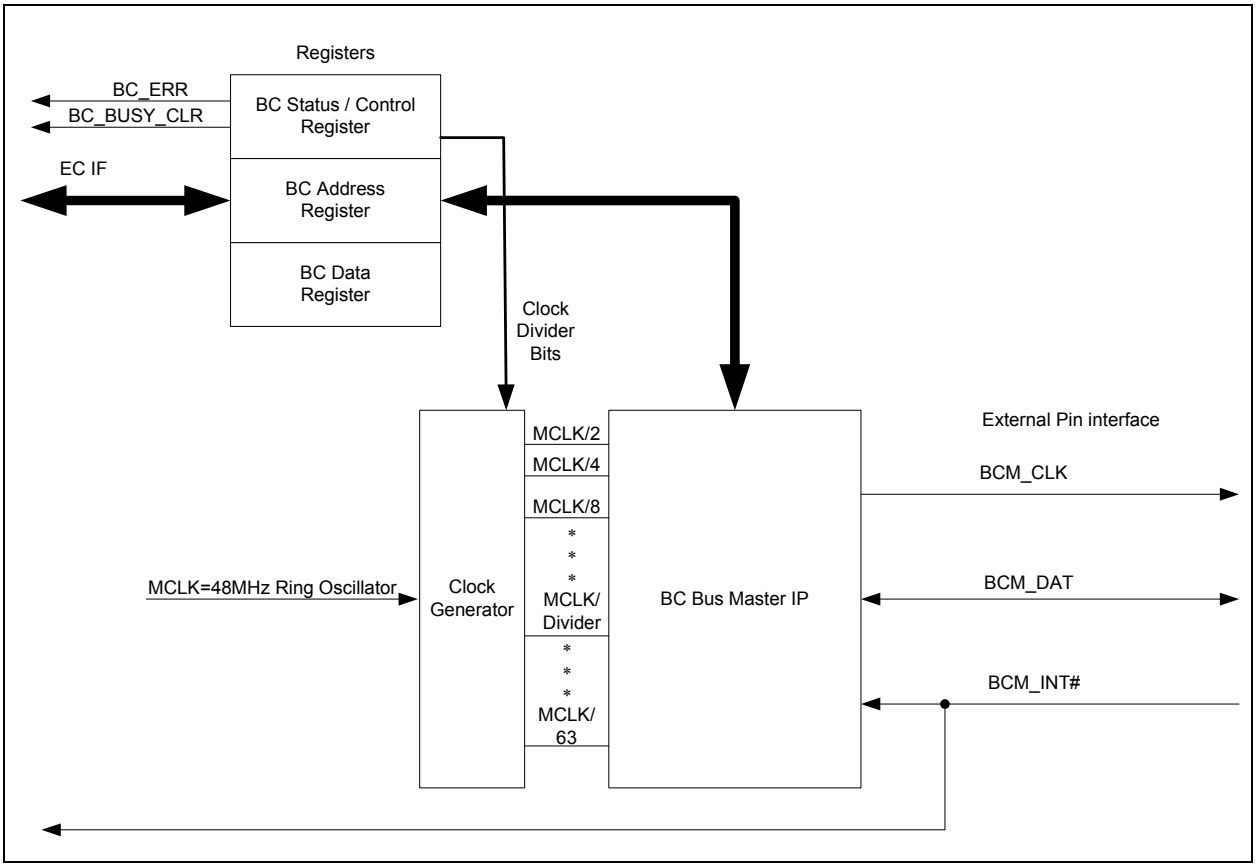
Source	Description
BCM_INT Busy	Interrupt request to the Interrupt Aggregator, generated from the status event BUSYdefined in the BC-Link Status Register.
BCM_INT Err	Interrupt request to the Interrupt Aggregator, generated from the status event defined in the BC-Link Status Register.
BC_INT_N_WK	Wake-up request to the Interrupt Aggregator's wake-up interface for BC-Link Master port. In order to enable BC-Link wakeup interrupts, the pin control registers for the BC_INT# pin must be programmed to Input, Falling Edge Triggered, non-inverted polarity detection.

31.9 Low Power Modes

The BC-Link Master Interface automatically enters a low power mode whenever it is not active (that is, whenever the BUSY bit in the BC-Link Status Register is '0'). When the interface is in a low-power mode it will not prevent the chip from entering a sleep state. When the interface is active it will inhibit the chip sleep state until the interface has re-entered its low power mode.

31.10 Description

FIGURE 31-2: BC-LINK MASTER BLOCK DIAGRAM



31.10.1 BC-LINK MASTER READ OPERATION

The BC-Link Read protocol requires two reads of the BC-Link Data Register. The two reads drive a two state-state machine: the two states are Read#1 and Read#2. The Read#1 of the Data Register starts the read protocol on the BC-Link pins and sets the BUSY bit in the BC-Link Status Register. The contents of the data read during Read#1 by the EC is stale and is not to be used. After the BUSY bit in the BC-Link Status Register autonomously clears to '0', the Read#2 of the Data Register transfers the data read from the peripheral/BC-Link companion chip to the EC.

1. Software starts by checking the status of the BUSY bit in the Status Register. If the BUSY bit is '0', proceed. If BUSY is '1', firmware must wait until it is '0'.
2. Software writes the address of the register to be read into the BC-Link Address Register.
3. Software then reads the Data Register. This read returns random data. The read activates the BC-Link Master state machine to transmit the read request packet to the BC-Link companion. When the transfer initiates, the hardware sets the BUSY bit to a '1'.
4. The BC-Link Companion reads the selected register and transmits the read response packet to the BC-Link Master. The Companion will ignore the read request if there is a CRC error; this will cause the Master state machine to time-out and issue a BC_ERR Interrupt.
5. The Master state machine loads the Data Register, issues a BUSY Bit Clear interrupt and clears the BUSY bit to '0'.
6. Software, after either receiving the Bit Clear interrupt, or polling the BUSY bit until it is '0', checks the BC_ERR bit in the Status Register.
7. Software can now read the Data Register which contains the valid data if there was no BC Bus error.
8. If a Bus Error occurs, firmware must issue a soft reset by setting the RESET bit in the Status Register to '1'.
9. The read can re-tried once BUSY is cleared.

Note: Steps 3 thorough 7 should be completed as a contiguous sequence. If not the interface could be presenting incorrect data when software thinks it is accessing a valid register read.

31.10.2 BC-LINK MASTER WRITE OPERATION

1. Software starts by checking the status of the BUSY bit in the BC-Link Status Register. If the BUSY bit is '0', proceed. If BUSY is '1', firmware must wait until it is '0'.
2. Software writes the address of the register to be written into the BC-Link Address Register.
3. Software writes the data to be written into the addressed register in to the BC-Link Data Register.
4. The write to the Data Register starts the BC_Link write operation. The Master state machine sets the BUSY bit.
5. The BC-Link Master Interface transmits the write request packet.
6. When the write request packet is received by the BC-Link companion, the CRC is checked and data is written to the addressed companion register.
7. The companion sends an ACK if the write is completed. A time-out will occur approximately 16 BC-Link clocks after the packet is sent by the Master state machine. If a time-out occurs, the state machine will set the BC_ERR bit in the Status Register to '1' approximately 48 clocks later and then clear the BUSY bit.
8. The Master state machine issues the Bit Clear interrupt and clears the BUSY bit after receiving the ACK from the Companion
9. If a Bus Error occurs, firmware must issue a soft reset by setting the RESET bit in the Status Register to '1'.
10. The write can re-tried once BUSY is cleared.\

31.11 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the BC-Link Master interface. The addresses of each register listed in this table are defined as a relative offset to the host "Base Address" defined in the EC-Only Register Base Address Table.

TABLE 31-3: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address ()
BC-LINK	0	EC	32-bit internal address space	0000_BC00h
BC-LINK	1	EC	32-bit internal address space	0000_BD00h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 31-4: EC-ONLY REGISTER SUMMARY

Register Name	EC Offset
BC-Link Status Register	00h
BC-Link Address Register	04h
BC-Link Data Register	08h
BC-Link Clock Select Register	0Ch

31.11.1 BC-LINK STATUS REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	R	-	-
7	RESET When this bit is ‘1’the BC_Link Master Interface will be placed in reset and be held in reset until this bit is cleared to ‘0’. Setting RESET to ‘1’ causes the BUSY bit to be set to ‘1’. The BUSY remains set to ‘1’ until the reset operation of the BC Interface is completed, which takes approximately 48 BC clocks. The de-assertion of the BUSY bit on reset will not generate an interrupt, even if the BC_BUSY_CLR_INT_EN bit is ‘1’. The BUSY bit must be polled in order to determine when the reset operation has completed.	R/W	1h	nSYSRST
6	BC_ERR This bit indicates that a BC Bus Error has occurred. If an error occurs this bit is set by hardware when the BUSY bit is cleared. This bit is cleared when written with a ’1’. An interrupt is generated If this bit is ‘1’ and BC_ERR_INT_EN bit is ‘1’. Errors that cause this interrupt are: • Bad Data received by the BASE (CRC Error) • Time-out caused by the COMPANION not responding. All COMPANION errors cause the COMPANION to abort the operation and the BASE to time-out.31.11.2	R/WC	0h	nSYSRST
5	BC_ERR_INT_EN This bit is an enable for generating an interrupt when the BC_ERR bit is set by hardware. When this bit is ‘1’, the interrupt signal is enabled. When this bit is ‘0’, the interrupt is disabled.	R/W	0b	nSYSRST
4	BC_BUSY_CLR_INT_EN This bit is an enable for generating an interrupt when the BUSY bit in this register is cleared by hardware. When this bit is set to ‘1’, the interrupt signal is enabled. When the this bit is cleared to ‘0’, the interrupt is disabled. When enabled, the interrupt occurs after a BC Bus read or write.	R/W	0h	nSYSRST
3:1	Reserved	R	-	-
0	BUSY This bit is asserted to ‘1’ when the BC interface is transferring data and on reset. Otherwise it is cleared to ‘0’. When this bit is cleared by hardware, an interrupt is generated if the BC_BUSY_CLR_INT_EN bit is set to ‘1’.	R	1h	nSYSRST

31.11.2 BC-LINK ADDRESS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7:0	ADDRESS Address in the Companion for the BC-Link transaction.	R/W	0h	nSYSRST

31.11.3 BC-LINK DATA REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7:0	DATA As described in Section 31.10.1, "BC-Link Master READ Operation" and Section 31.10.2, "BC-Link Master WRITE Operation", this register hold data used in a BC-Link transaction.	R/W	0h	nSYSRST

31.11.4 BC-LINK CLOCK SELECT REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7:0	DIVIDER The BC Clock is set to the Master Clock divided by this field, or 48MHz/ (Divider +1). The clock divider bits can only can be changed when the BC Bus is in soft RESET (when either the Reset bit is set by software or when the BUSY bit is set by the interface). Settings for DIVIDER are shown in Table 31-5, "Frequency Settings".	R/W	4h	nSYSRST

TABLE 31-5: FREQUENCY SETTINGS

Divider	Frequency
1	24MHz
2	16MHz
3	12MHz
4	9.6MHz
15	2.18MHz
2A	1.12MHz

32.0 TRACE FIFO DEBUG PORT (TFDP)

32.1 Introduction

The TFDP serially transmits Embedded Controller (EC)-originated diagnostic vectors to an external debug trace system.

32.2 References

No references have been cited for this chapter.

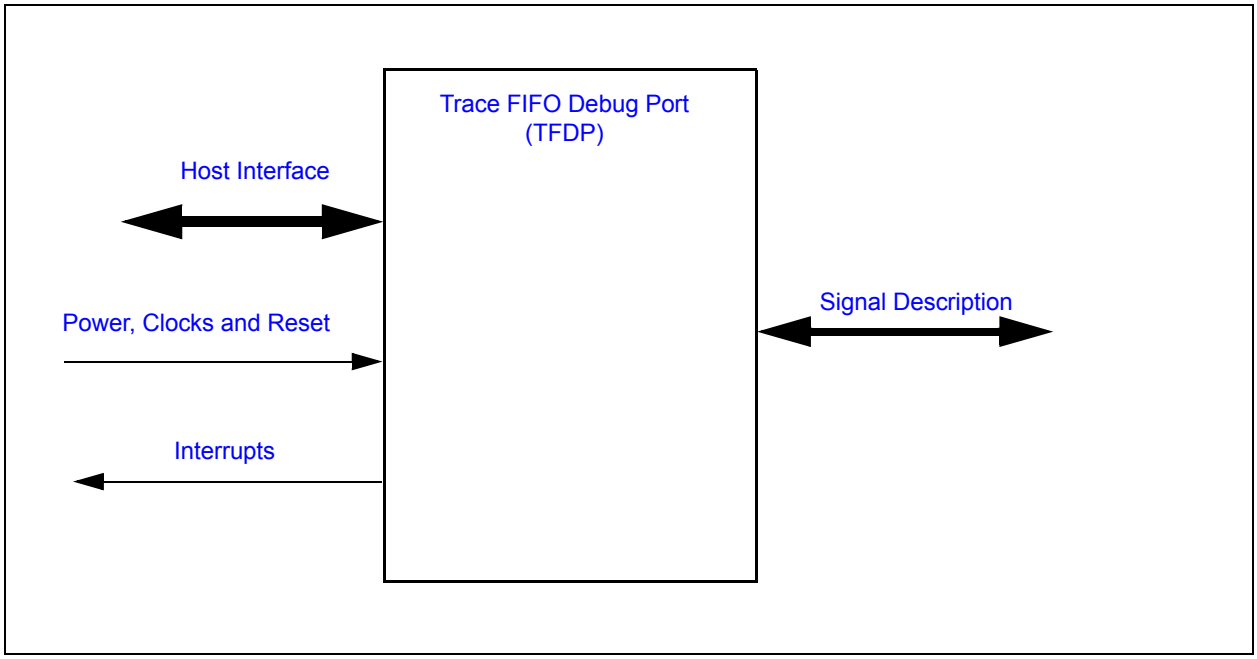
32.3 Terminology

There is no terminology defined for this chapter.

32.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 32-1: I/O DIAGRAM OF BLOCK



32.5 Signal Description

The Signal Description Table lists the signals that are typically routed to the pin interface.

TABLE 32-1: SIGNAL DESCRIPTION

Name	Direction	Description
TFDP Clk	Output	Derived from EC Bus Clock.
TFDP Data	Output	Serialized data shifted out by TFDP Clk.

32.6 Host Interface

The registers defined for the Trace FIFO Debug Port (TFDP) are accessible by the various hosts as indicated in Section 32.11, "EC-Only Registers".

32.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

32.7.1 POWER DOMAINS

Name	Description
VTR	This power well sources all of the registers and logic in this block.

32.7.2 CLOCK INPUTS

Name	Description
48 MHz Ring Oscillator	This clock input is used to derive the TFDP Clk.

32.7.3 RESETS

Name	Description
nSYSRST	This reset signal resets all of the registers and logic in this block.

32.8 Interrupts

There are no interrupts generated from this block.

32.9 Low Power Modes

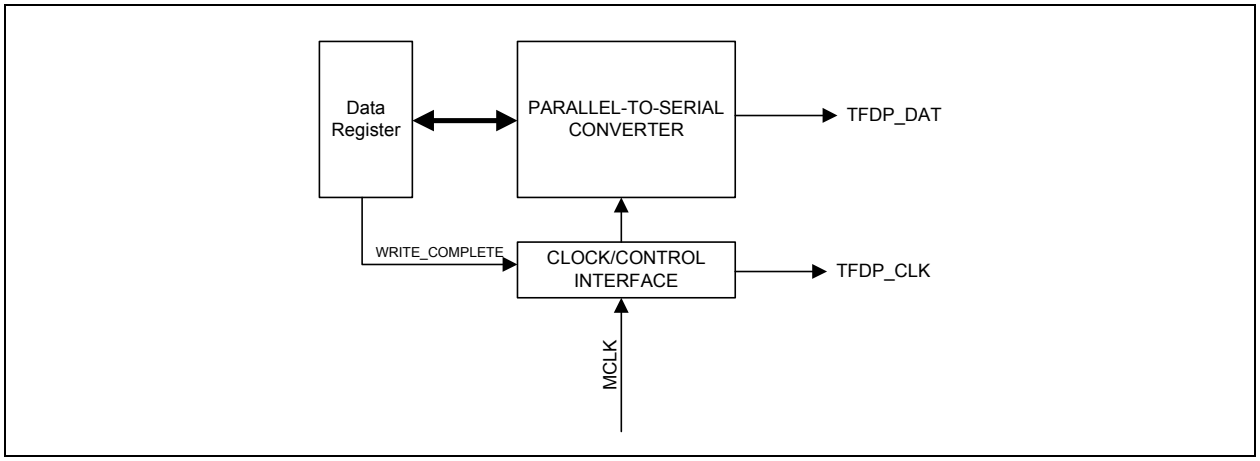
The Trace FIFO Debug Port (TFDP) may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

32.10 Description

The TFDP is a unidirectional (from processor to external world) two-wire serial, byte-oriented debug interface for use by processor firmware to transmit diagnostic information.

The TFDP consists of the Debug Data Register, Debug Control Register, a Parallel-to-Serial Converter, a Clock/Control Interface and a two-pin external interface (TFDP Clk, TFDP Data). See .

FIGURE 32-2: BLOCK DIAGRAM OF TFDP DEBUG PORT

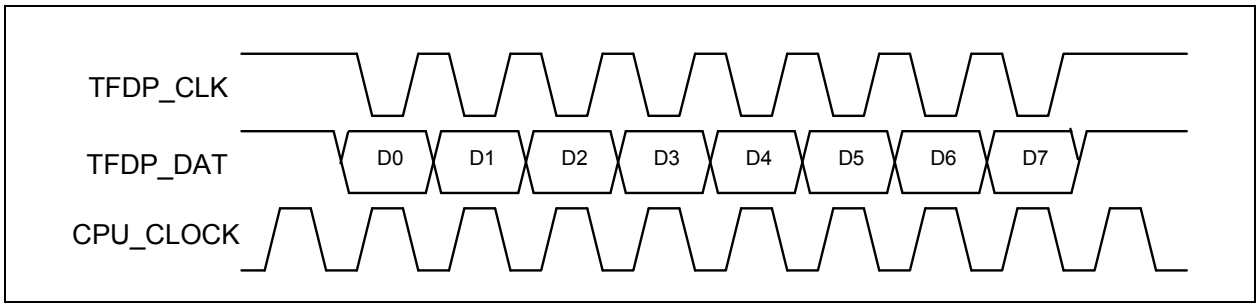


The firmware executing on the embedded controller writes to the Debug Data Register to initiate a transfer cycle (32.11). At first, data from the Debug Data Register is shifted into the LSB. Afterwards, it is transmitted at the rate of one byte per transfer cycle.

Data is transferred in one direction only from the Debug Data Register to the external interface. The data is shifted out at the clock edge. The clock edge is selected by the EDGE_SEL bit in the Debug Control Register. After being shifted out, valid data is guaranteed at the opposite edge of the TFDP_CLK. For example, when the EDGE_SEL bit is '0' (default), valid data is provided at the falling edge of TFDP_CLK. The Setup Time (to the falling edge of TFDP_CLK) is 10 ns, minimum. The Hold Time is 1 ns, minimum.

When the Serial Debug Port is inactive, the TFDP_CLK and TFDP_DAT outputs are '1.' The EC Bus Clock clock input is the transfer clock.

FIGURE 32-3: DATA TRANSFER



32.11 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Trace FIFO Debug Port (TFDP). The addresses of each register listed in this table are defined as a relative offset to the host "Base Address" defined in the EC-Only Register Base Address Table.

TABLE 32-2: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
TFDP Debug Port	0	EC	32-bit internal address space	0000_8C00h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 32-3: EC-ONLY REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	Debug Data Register
04h	Debug Control Register

32.11.1 DEBUG DATA REGISTER

The Debut Data Register is Read/Write. It always returns the last data written by the TFDP or the power-on default '00h'.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	DATA Debug data to be shifted out on the TFDP Debug port. While data is being shifted out, the Host Interface will 'hold-off' additional writes to the data register until the transfer is complete.	R/W	00h	nSYSRST

32.11.2 DEBUG CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	Reserved	R	-	-
6:4	IP_DELAY Inter-packet Delay. The delay is in terms of TFDP Debug output clocks. A value of 0 provides a 1 clock inter-packet period, while a value of 7 provides 8 clocks between packets:	R/W	000b	nSYSRST
3:2	DIVSEL Clock Divider Select. The TFDP Debug output clock is determined by this field, according to Table 32-4, "TFDP Debug Clocking":	R/W	00b	nSYSRST
1	EDGE_SEL 1= Data is shifted out on the falling edge of the debug clock 0= Data is shifted out on the rising edge of the debug clock (Default)	R/W	0b	nSYSRST
0	EN Enable. 1=Clock enabled 0=Clock is disabled (Default)	R/W	0b	nSYSRST

TABLE 32-4: TFDP DEBUG CLOCKING

divsel	TFDP Debug Clock
00	24 MHz
01	12 MHz
10	6 MHz
11	Reserved

33.0 PORT 80 BIOS DEBUG PORT

33.1 Overview

The Port 80 BIOS Debug Port emulates the functionality of a “Port 80” ISA plug-in card. In addition, a timestamp for the debug data can be optionally added.

Diagnostic data is written by the Host Interface to the Port 80 BIOS Debug Port, which is located in the Host I/O address space. The Port 80 BIOS Debug Port generates an interrupt to the EC when host data is available. The EC reads this data along with the timestamp, if enabled.

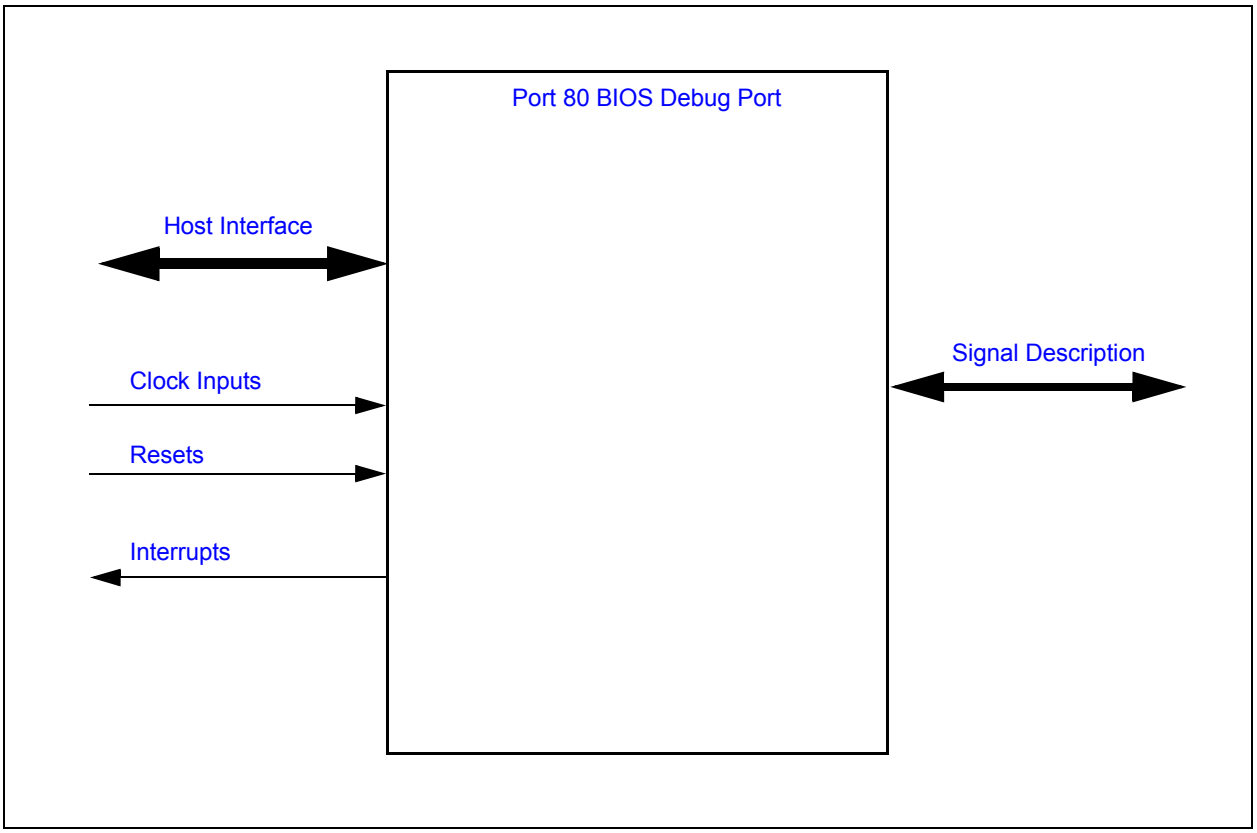
33.2 References

There are no references for this block.

33.3 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 33-1: I/O DIAGRAM OF BLOCK



33.4 Signal Description

There are no external signals for this block.

33.5 Host Interface

The Port 80 block is accessed by host software via a registered interface, as defined in Section 33.11, "Runtime Registers".

33.6 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

33.6.1 POWER DOMAINS

Name	Description
VTR	This Power Well is used to power the registers and logic in this block.

33.6.2 CLOCK INPUTS

Name	Description
48 MHz Ring Oscillator	This is the clock source for Port 80 block logic.

33.6.3 RESETS

Name	Description
nSYSRST	This signal is asserted when VTR is low, PWRGD is low, or Host Interface is reset.

33.7 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
BDP_INT	The Port 80 BIOS Debug Port generates an EC interrupt when the amount of data in the Port 80 FIFO equals or exceeds the FIFO Threshold defined in the Configuration Register. The interrupt signal is always generated by the Port 80 block if the block is enabled; the interrupt is enabled or disabled in the Interrupt Controller.

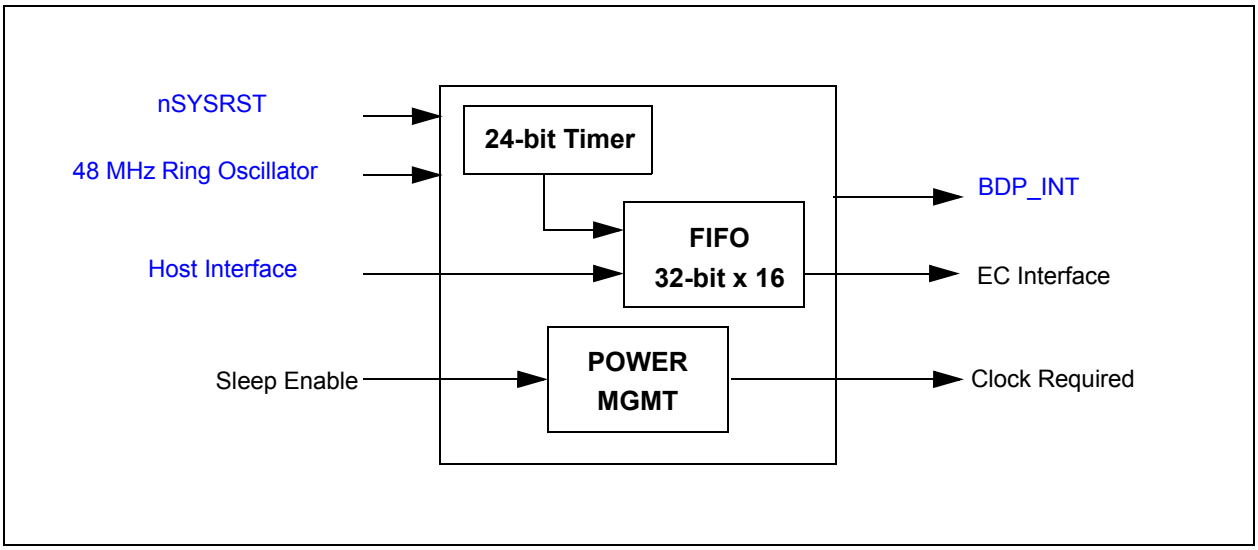
33.8 Low Power Modes

The Port 80 block may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

33.9 Description

33.9.1 BLOCK DIAGRAM

FIGURE 33-2: Port 80 BIOS Debug Port BLOCK DIAGRAM



The Port 80 BIOS Debug Port consists of a 32-bit wide x 16 deep FIFO and a 24-bit free running timer. Host and EC access to the Port 80 device is through a set of registers. The Host can write the FIFO via the Runtime Registers and the EC can read the FIFO can control the device via the EC-Only Registers.

Writes to the Host Data Register are concatenated with the 24-bit timestamp and written to the FIFO. Reads of the Host Data Register return zero. If writes to the Host Data Register overrun the FIFO, the oldest data are discarded and the OVERRUN status bit in the Status Register is asserted.

Only the EC can read data from the FIFO, using the EC Data Register. The use of this data is determined by EC Firmware alone.

Note:	The Port 80 BIOS Debug Port operates in byte mode. It does not support word writes when locating the two instances at contiguous base addresses.

33.10 Configuration Registers

The registers listed in the Configuration Register Summary table are for a single instance of the Port 80 BIOS Debug Port. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the Configuration Register Base Address Table.

FIGURE 33-3: CONFIGURATION REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
Port 80 BIOS Debug Port	0	LPC	Configuration Port	INDEX = 00h
		EC	32-bit internal address space	000F_5400h
	1	LPC	Configuration Port	INDEX = 00h
		EC	32-bit internal address space	000F_5800h

Each Configuration register access through the Host Access Port is via its LDN and its Host Access Port Index. EC access is a relative offset to the EC Base Address.

TABLE 33-1: CONFIGURATION REGISTER SUMMARY

EC Offset	Host Index	Register Name (Mnemonic)
330h	30h	Activate Register

33.10.1 ACTIVATE REGISTER

Offset	330h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	R	-	-
0	ACTIVATE When this bit is asserted ‘1’, the block is enabled. When this bit is ‘0’, writes by the Host interface to the Host Data Register are not claimed, the FIFO is flushed, the 24-bit Timer is reset, and the timer clock is stopped. Control bits in the Configuration Register are not affected by the state of ACTIVATE.	R/W	0h	nSYSRST

33.11 Runtime Registers

The registers listed in the Runtime Register Summary table are for two instances of the Port 80 BIOS Debug Port. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the Runtime Register Base Address Table.

Note:	The Runtime registers may be accessed by the EC but typically the Host will access the Runtime Registers and the EC will access just the EC-Only registers.

TABLE 33-2: RUNTIME REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
Port 80 BIOS Debug Port	0	LPC	I/O	Programmed BAR
		EC	32-bit internal address space	000F_5400h
	1	LPC	I/O	Programmed BAR
		EC	32-bit internal address space	000F_5800h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 33-3: RUNTIME REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	Host Data Register

33.11.1 HOST DATA REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7:0	HOST_DATA	W	0h	nSYSRST

33.12 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for two instances of the Port 80 BIOS Debug Port. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 33-4: EC-ONLY REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
Port 80 BIOS Debug Port	0	EC	32-bit internal address space	000F_5400h
	1	EC	32-bit internal address space	000F_5800h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 33-5: EC-ONLY REGISTER SUMMARY

Offset	Register Name (Mnemonic)
100h	EC Data Register
104h	Configuration Register
108h	Status Register
10Ch	Count Register

33.12.1 EC DATA REGISTER

Offset	100h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	TIME_STAMP	R	0h	nSYSRST
7:0	EC_DATA	R	0h	nSYSRST

33.12.2 CONFIGURATION REGISTER

Offset	104h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7:6	FIFO_THRESHOLD This field determines the threshold for the Port 80 BIOS Debug Port Interrupts. 3=14 entry threshold 2=8 entry threshold 1=4 entry threshold 0=1 entry threshold	R/W	0h	nSYSRST
5	TIMER_ENABLE When the TIMER_ENABLE bit is '1', the 24-bit Timer is actively counting at a rate determined by the TIMEBASE_SELECT bits. When the TIMER ENABLE bit is '0', counting is stopped.	R/W	0h	nSYSRST
4:3	TIMEBASE_SELECT The TIMEBASE SELECT bits determine the clock for the 24-bit Timer. 3=48 MHz Ring Oscillator/64 2=48 MHz Ring Oscillator/32 1=48 MHz Ring Oscillator/16 0=48 MHz Ring Oscillator/8	R/W	0h	nSYSRST
2	RESET_TIMESTAMP When this field is written with a '1', the 24-bit Timer is reset to '0'. Writing zero to the Count Register has the same effect. Writes of a '0' to this field have no effect. Reads always return '0'.	W	-	nSYSRST
1	FLUSH When this field is written with a '1', the FIFO is flushed. Writes of a '0' to this field have no effect. Reads always return '0'.	W	-	nSYSRST
0	Reserved	R	-	-

33.12.3 STATUS REGISTER

Offset	108h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	R	-	-
1	OVERRUN The OVERRUN bit is ‘1’ when the host writes the Host Data Register when the FIFO is full.	R	0h	nSYSRST
0	NOT_EMPTY The NOT EMPTY bit is ‘1’ when there is data in the FIFO. The NOT EMPTY bit is ‘0’ when the FIFO is empty.	R	0h	nSYSRST

33.12.4 COUNT REGISTER

Offset	10Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
32:8	COUNT Writes load data into the 24-bit Timer. Reads return the 24-bit Timer current value.	R/W	0h	–
7:0	Reserved	R	-	-

34.0 EC SUBSYSTEM REGISTERS

34.1 Introduction

This chapter defines a bank of registers associated with the EC Subsystem.

34.2 References

None

34.3 Interface

This block is designed to be accessed internally by the EC via the register interface.

34.4 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

34.4.1 POWER DOMAINS

Name	Description
VTR	The EC Subsystem Registers are all implemented on this single power domain.

34.4.2 CLOCK INPUTS

This block does not require any special clock inputs. All register accesses are synchronized to the host clock.

34.4.3 RESETS

Name	Description
VTR_RESET#	This reset signal, which is an input to this block, resets registers to their initial default state on a power-on-reset event only.
nSYSRST	This reset signal, which is an input to this block, resets registers to their initial default state any time the embedded controller is reset.

34.5 Interrupts

This block does not generate any interrupt events.

34.6 Low Power Modes

The EC Subsystem Registers may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. When this block is commanded to sleep it will still allow read/write access to the registers.

34.7 Description

The EC Subsystem Registers block is a block implemented for aggregating miscellaneous registers required by the Embedded Controller (EC) Subsystem that are not unique to a block implemented in the EC subsystem.

34.8 EC-Only Registers

TABLE 34-1: EC-ONLY REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
EC_REG_BANK	0	EC	32-bit internal address space	0000_FC00h

Note 34-1 The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 34-2: EC-ONLY REGISTER SUMMARY

Offset	Register Name
00h	Test
04h	Test
08h	Test
0Ch	Test
10h	Test
14h	AHB Error Control
18h	Comparator Control
1Ch	Test
20h	JTAG Enable
24h	Test
28h	WDT Event Count
2Ch	Reserved
30h	Reserved
34h	Test
38h	Reserved
3Ch	Test
40h	VREF_CPU DISABLE
44h	Test
48h	Power Regions Voltage Control

34.8.1 AHB ERROR CONTROL

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	R	-	-
0	AHB_ERROR_DISABLE 0: EC memory exceptions are enabled. 1: EC memory exceptions are disabled.	RW	0h	nSYSRST

34.8.2 COMPARATOR CONTROL

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	Reserved	R	-	-
5	Comparator 1 Threshold Input Select 0: Pin 1: DAC1	RW	0h	nSYSRST
4	Comparator 1 Enable 0: Disable Comparator 1 for operation 1: Enable Comparator 1 operation.	RW	0h	nSYSRST
3	Reserved	R	-	-
2	Comparator 0 Configuration Locked 0: Configuration Not Locked. Bits[2:0] are Read-Write 1: Configuration Locked. Bits[2:0] are Read-Only	R/W1X	0h	nSYSRST
1	Comparator 0 Threshold Input Select 0: Pin 1: DAC0	RW or RO (Note 1)	0h	nSYSRST
0	Comparator 0 Enable 0: Disable Comparator 0 for operation 1: Enable Comparator 0 operation.	RW or RO (Note 1)	0h	nSYSRST

Note 1: These bits become Read-Only by writing bit 2 Comparator 0 Configuration Locked bit

34.8.3 JTAG ENABLE

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	R	-	-
1	Boot ROM Configuration Ready This bit indicates to the ICSP debugger when the Boot ROM has finished its configuration sequence. The state of this bit is reflected in the MCHP_CMD <0x07> Read Status register. 0 = Boot ROM has not finished configuration sequence 1 = Boot ROM has finished configuration sequence.	R/W	0b	nSYSRST

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	JTAG_EN This bit enables the JTAG debug port. 0 = JTAG port disabled. JTAG cannot be enabled (i.e., the TRST# pin is ignored and the JTAG signals remain in their non-JTAG state). 1= JTAG port enabled. A high on TRST# enables JTAG	R/W	0b	nSYSRST

34.8.4 WDT EVENT COUNT

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	R	-	-
3:0	WDT_COUNT These EC R/W bits are cleared to 0 on VTR POR, but not on a WDT. **Note:** This field is written by Boot ROM firmware to indicate the number of times a WDT fired before loading a good EC code image.	R/W	0b	VTR_RESET#

34.8.5 VREF_CPU DISABLE

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	R	-	-
6:2	Test	R/W	0b	nSYSRST
1	VREF_CPU Disable 0: Enable 1: Disable **Note:** In order to achieve the lowest leakage current when both PECI and SB TSI are not used, set the VREF_CPU Disable bit to 1.	R/W	0b	nSYSRST
0	Test	R	0b	nSYSRST

34.8.6 POWER REGIONS VOLTAGE CONTROL

Offset	48h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	R	-	-
3	**VTR_LPC_ESPI_SEL18** 0 = 3.3V Operation (use for LPC interface) 1 = 1.8V Operation (use for eSPI Interface) **Note:** If the I2C interface is used as the host interface, the GPIOs on the LPC and eSPI interface may be configured to operate as either 1.8V or 3.3V GPIOs..	R/W	0b	nSYSRST
2	**Test** **Note:** Writing this register bit to a different value may cause unwanted results. This bit must always be set to 0.	R/W	0b	nSYSRST
1	**Test** **Note:** Writing this register bit to a different value may cause unwanted results. This bit must always be set to 0.	R/W	0b	nSYSRST
0	Reserved	R	-	-

35.0 VBAT REGISTER BANK

35.1 Introduction

This chapter defines a bank of registers powered by VBAT.

35.2 Interface

This block is designed to be accessed internally by the EC via the register interface.

35.3 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

35.3.1 POWER DOMAINS

Name	Description
VBAT	The VBAT Register Bank are all implemented on this single power domain.

35.3.2 CLOCK INPUTS

This block does not require any special clock inputs. All register accesses are synchronized to the host clock.

35.3.3 RESETS

Name	Description
VBAT_POR	This reset signal, which is an input to this block, resets all the logic and registers to their initial default state.

35.4 Interrupts

Name	Description
PFR_Status	This interrupt signal from the Power-Fail and Reset Status Register indicates VBAT RST and WDT events.

35.5 Low Power Modes

The VBAT Register Bank is designed to always operate in the lowest power consumption state.

35.6 Description

The VBAT Register Bank block is a block implemented for aggregating miscellaneous battery-backed registers required the host and by the Embedded Controller (EC) Subsystem that are not unique to a block implemented in the EC subsystem.

35.7 EC-Only Registers

TABLE 35-1: EC-ONLY REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
VBAT_REG_BANK	0	EC	32-bit internal address space	0000_A400h

Note 35-1 The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 35-2: RUNTIME REGISTER SUMMARY

Offset	Register Name
00h	Power-Fail and Reset Status Register
04h	Test Register
08h	Clock Enable Register
10h	Test Register
14h	Test Register
18h	Alternate Function VTR Control
1Ch	Test Register

35.7.1 POWER-FAIL AND RESET STATUS REGISTER

The Power-Fail and Reset Status Register collects and retains the VBAT RST and WDT event status when VTR is unpowered.

Address	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	VBAT_RST The VBAT RST bit is set to ‘1’ by hardware when a VBAT_POR is detected. This is the register default value. To clear VBAT RST EC firmware must write a ‘1’ to this bit; writing a ‘0’ to VBAT RST has no affect.	R/WC	1	VBAT_POR
6	Reserved	RES	-	-
5	WDT The WDT bit is asserted (‘1’) following a Watch-Dog Timer Forced Reset (WDT Event). To clear the WDT bit EC firmware must write a ‘1’ to this bit; writing a ‘0’ to the WDT bit has no affect.	R/WC	0 (Note 35-2)	VBAT_POR (Note 35-2)

Address	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
4:1	Reserved	RES	-	-
0	DET32K_IN 0 = No clock detected on the XTAL[1:2] pins. 1= Clock detected on the XTAL[1:2] pins.	R	X	VBAT_POR

Note 35-2 In the MEC140x/1x devices the WDT defaults to disabled, however the Boot ROM Exception Handler uses the WDT to generate a nSYSRST. The Boot ROM only touches the WDT if the BEV exception fires. In this case 0x5 is written to the EC Subsystem WDT_COUNT bit field. The WDT Status bit, located in the WDT EC-Only Register bank, and the WDT status bit located in the Power-Fail and Reset Status Register register are cleared before the WDT is enabled.

35.7.2 CLOCK ENABLE REGISTER

Address	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	RES	-	-
5	48MHz Oscillator Reference Select 0 = External 32KHz clock source is the 48MHz clock reference (default) 1 = Switched Clock Source (i.e., either internal 32kHz or external 32kHz clock) is the 48MHz clock reference **Note:** The external 32KHz clock source may be either the crystal or external single-ended 32kHz clock as selected by the XOSEL bit.	R/W	0b	VBAT_POR
4	32KHz Clock Switcher Control This bit disables the clock switcher logic. 0 = If the device is configured to operate on the external single-ended 32.768 KHz clock source and the clock switcher logic detects that the external clock is turned off, it will automatically switch to the internal 32k Hz clock source. It will remain operating on the internal 32k Hz clock source until it detects several good edges on the external clock input. Once it determines the external clock is on, the clock switcher will return control of the 32k Hz clock to the external pin. **Note:** Clock Switching only occurs when VTR is ON. The behavior of the 32kHz clock when VTR is OFF is determined by the INT_32K VTR Power Well Emulation bit. 1 = clock switching is disabled. The device will only operate on the clock enabled. See Table 35-3, "32kHz Clock Control" below.	R/W	0b (Note 35-3)	VBAT_POR
3	INT_32K VTR Power Well Emulation This bit determines the internal 32kHz clock behavior when VTR is off. 0 = VBAT Emulation. The internal 32k Hz clock remains ON when VTR is off. 1 = VTR Emulation. The internal 32k Hz clock is gated OFF when VTR is off.	R/W	0b (Note 35-3)	VBAT_POR

Address	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	INT_32K_OSC_EN 0 = Internal 32kHz oscillator is disabled 1 = Internal 32kHz oscillator is enabled. See Table 35-3, "32kHz Clock Control" below for determining the source of the 32kHz clock.	R/W	0b (Note 35-3)	VBAT_POR
1	EXT_32K_OSC_EN 0 = XOSEL control is disabled. All the External clock sources are disabled. 1 = External clock selected by XOSEL is enabled.	R/W	0b (Note 35-3)	VBAT_POR
0	XOSEL This bit controls whether a crystal or single ended clock source is used. 1= the 32.768 KHz Crystal Oscillator is driven by a single-ended 32.768 KHz clock source connected to the XTAL2 pin. 0= the 32.768 KHz Crystal Oscillator requires a 32.768 KHz parallel resonant crystal connected between the XTAL1 and XTAL2 pins (default).	R/W	0b (Note 35-3)	VBAT_POR

Note 35-3 If the Boot ROM exception handler runs, the Boot ROM will reconfigure this register to 04h, enabling the internal 32kHz clock source. This is done so the Boot ROM can issue a nSYSRST via a WDT Event.

TABLE 35-3: 32KHZ CLOCK CONTROL

XOSEL	32KHz Clock Switcher Control	EXT_32K_OSC_EN	INT_32K_OSC_EN	INT_32K VTR Power Well Emulation	32k Hz Clock Source
x	x	0	0	x	Disabled
1	x	1	0	x	Single-ended External Clock
0	x	1	x	x	External Crystal
x	x	0	1	0	Internal 32k Hz Clock - remains ON when VTR = 0V
x	x	0	1	1	Internal 32k Hz Clock - Turned OFF when VTR = 0V
1	0	1	1	0	Switched Clock Source: - Primary Clock is Single-ended External Clock when VTR is ON. - When the primary clock is OFF or VTR is OFF the internal 32k Hz clock is selected. **Note:** If VTR = 0V the internal 32kHz clock is ON.
1	0	1	1	1	Switched Clock Source: - Primary Clock is Single-ended External Clock when VTR is ON. - When the primary clock is OFF or VTR is OFF the internal 32k Hz clock is selected. **Note:** If VTR = 0V the internal 32kHz clock is OFF.
1	1	1	x	x	Single-ended External Clock

35.7.3 ALTERNATE FUNCTION VTR CONTROL

Address	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	RES	-	-
1	BGPO 0 = VTR Powered If VTR = ON, then the output is driven according to GPIO output control register If VTR = OFF, then the output pin is tristated 1 = VBAT Powered Output driven according to the BGPO bit located in	R/W	1b	VBAT_POR
0	VCI_OUT 0 = VTR Powered If VTR = ON, then the output is driven according to GPIO output control register If VTR = OFF, then the output pin is tristated 1 = VBAT Powered Output driven according to the VCI_OUT logic defined in Section 37.0, "VBAT-Powered Control Interface," on page 446	R/W	1b	VBAT_POR

36.0 VBAT-POWERED RAM

36.1 Overview

The VBAT Powered RAM provides a 64 Byte Random Accessed Memory that is operational while the main power rail is operational, and will retain its values powered by battery power while the main rail is unpowered.

36.2 References

No references have been cited for this feature.

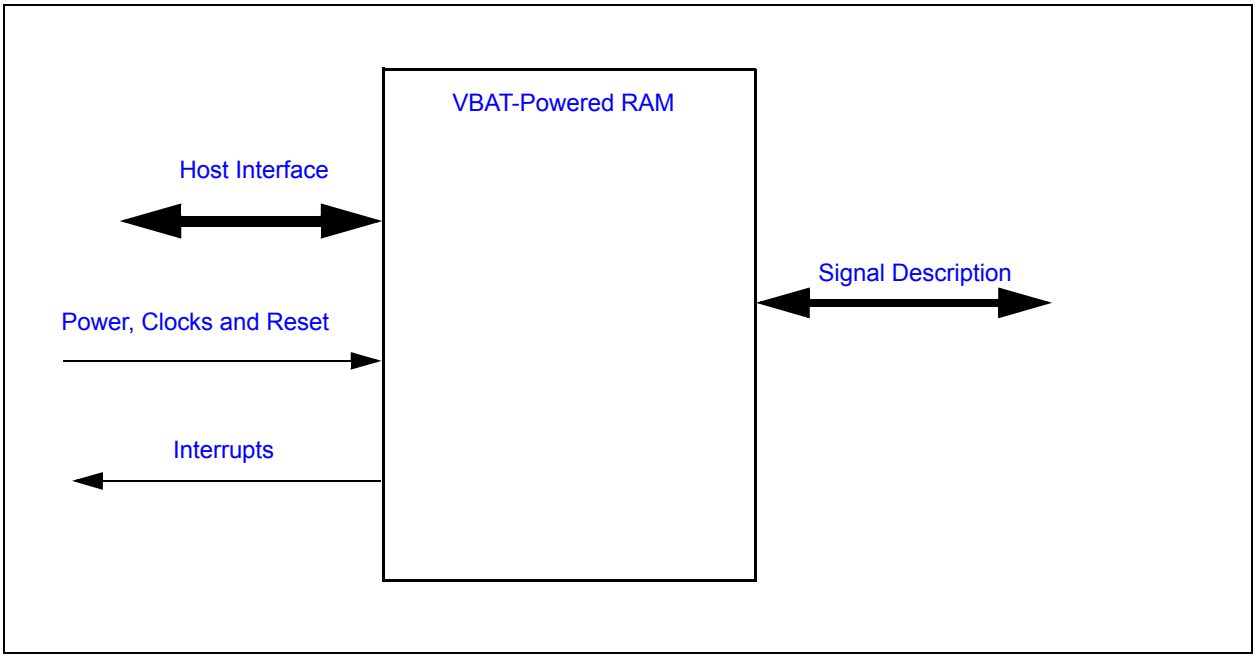
36.3 Terminology

There is no terminology defined for this section.

36.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 36-1: I/O DIAGRAM OF BLOCK



36.5 Signal Description

There are no external signals for this block.

36.6 Host Interface

The registers defined for the Keyboard Scan Interface are accessible by the various hosts as indicated in Section 36.11, "Registers".

36.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

36.7.1 POWER DOMAINS

Name	Description
VTR	The main power well used when the VBAT RAM is accessed by the EC.
VBAT	The power well used to retain memory state while the main power rail is unpowered.

36.7.2 CLOCK INPUTS

No special clocks are required for this block.

36.7.3 RESETS

Name	Description
VBAT_POR	This signal resets all the registers and logic in this block to their default state.

36.8 Interrupts

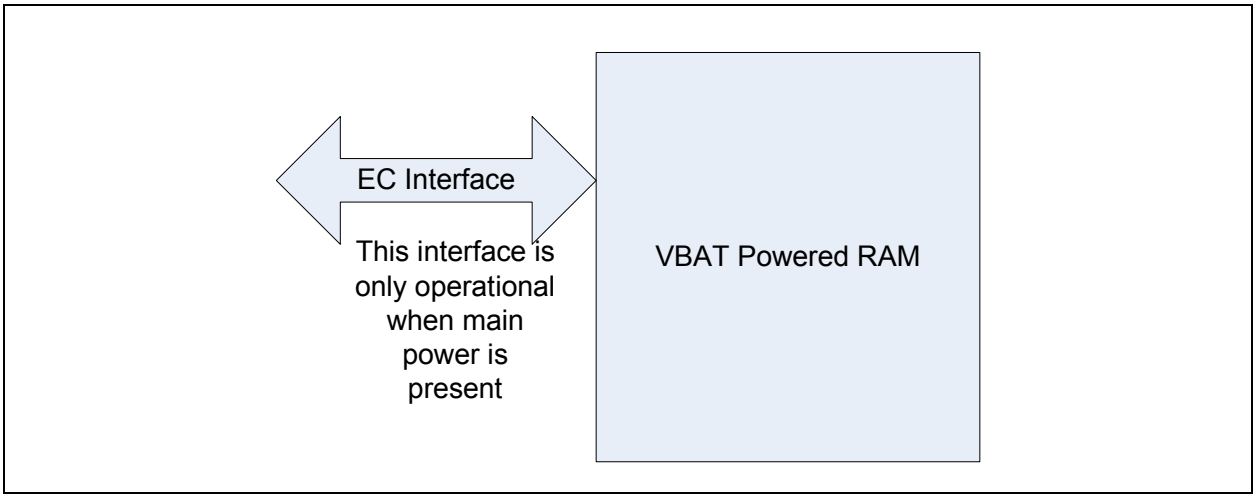
This block does not generate any interrupts.

36.9 Low Power Modes

The VBAT-Powered RAM automatically enters a low power mode whenever it is not being accessed by the EC. There is no chip-level Sleep Enable input.

36.10 Description

FIGURE 36-2: VBAT RAM BLOCK DIAGRAM



The VBAT Powered RAM provides a 64 Byte Random Accessed Memory that is operational while VTR is powered, and will retain its values powered by VBAT while VTR is unpowered. The RAM is organized as a 16 words x 32-bit wide for a total of 64 bytes.

36.11 Registers

36.11.1 REGISTERS SUMMARY

The registers listed in the Table 36-1, "EC-Only Register Base Address" are for a single instance of the Keyboard Scan Interface block. Each 32-bit RAM location is an offset from the EC base address.

TABLE 36-1: EC-ONLY REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
VBAT-Powered RAM	0	EC	32-bit internal address space	0000_A800h

Note: The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

37.0 VBAT-POWERED CONTROL INTERFACE

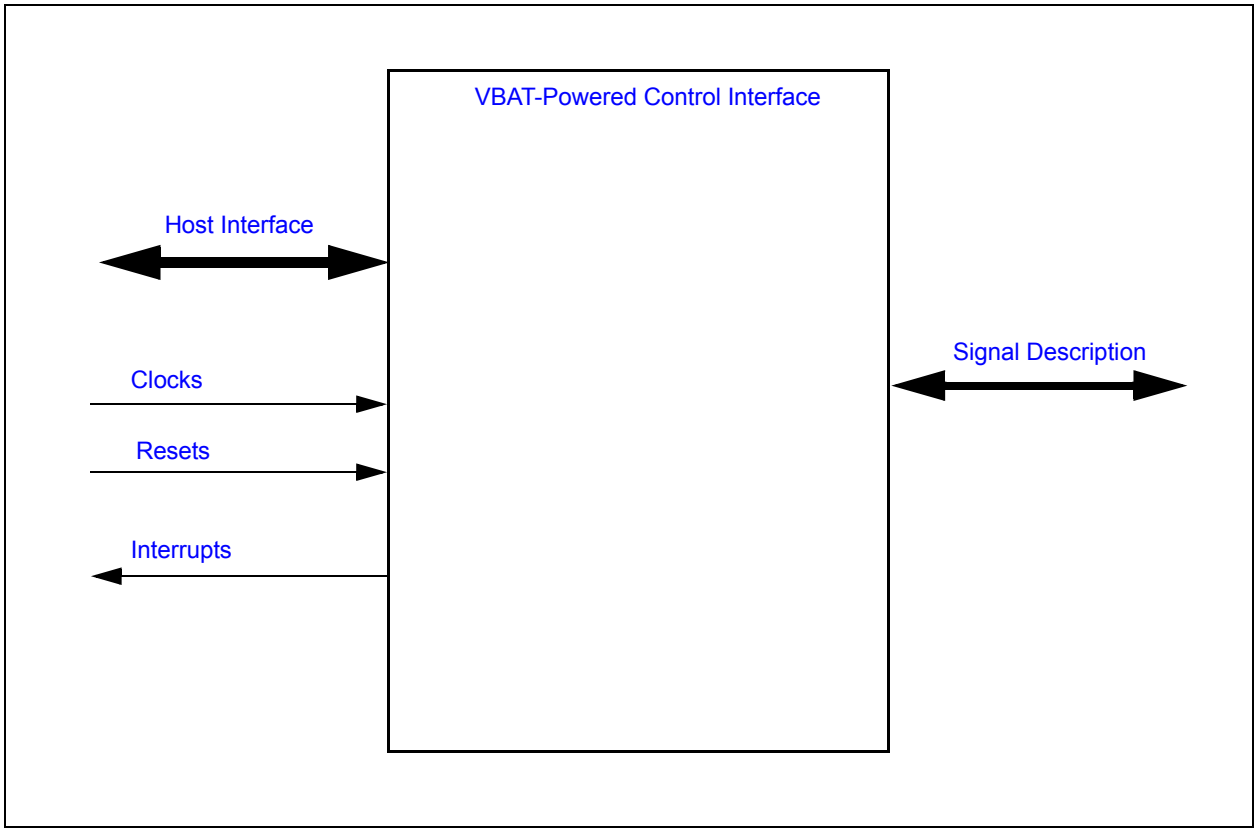
37.1 General Description

The VBAT-Powered Control Interface has VBAT powered combinational logic and input and output signal pins. The VBAT-Powered Control Interface block interfaces with the RTC/Week Timer on page 303.

37.2 Interface

This block's connections are entirely internal to the chip.

FIGURE 37-1: I/O DIAGRAM OF BLOCK



37.3 Signal Description

TABLE 37-1: EXTERNAL SIGNAL DESCRIPTION

Name	Direction	Description
VCI_OUT	OUTPUT	Output status driven by this block.
VCI_IN0#	INPUT	Input, active low
VCI_IN1#	INPUT	Input, active low
VCI_OVRD_IN	INPUT	Input, active high

TABLE 37-2: INTERNAL SIGNAL DESCRIPTION

Name	Direction	Description
POWER_UP_EVENT	INPUT	Signal from the RTC/Week Timer block. The POWER_UP_EVENT is asserted by the timer when either the Week_Alarm or the Sub-Week Alarm is asserted. The POWER_UP_EVENT can be suppressed if the SYSPWR_PRES pin indicates that system power is not available.
VTRGD	INPUT	Status signal for the state of the VTR power rail. This signal is high if the power rail is on, and low if the power rail is off.

37.4 Host Interface

The registers defined for the VBAT-Powered Control Interface are accessible only by the EC.

37.5 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

37.5.1 POWER DOMAINS

TABLE 37-3: POWER SOURCES

Name	Description
VBAT	This power well sources all of the internal registers and logic in this block.
VTR	This power well sources only bus communication. The block continues to operate internally while this rail is down.

37.5.2 CLOCKS

This block does not require clocks.

37.5.3 RESETS

TABLE 37-4: RESET SIGNALS

Name	Description
VBAT_POR	This reset signal is used reset all of the registers and logic in this block.
nSYSRST	This reset signal is used to inhibit the bus communication logic, and isolates this block from VTR powered circuitry on-chip. Otherwise it has no effect on the internal state.

37.6 Interrupts

TABLE 37-5: EC INTERRUPTS

Source	Description
VCI_IN0	This interrupt is routed to the Interrupt Controller It is only asserted when both VBAT and VTR are powered. Edge detection and assertion level for the interrupt are configured in the GPIO Pin Control Register for the GPIO that shares the pin with VCI_IN# input. This interrupt is equivalent to the GPIO interrupt for the GPIO that shares the pin, but appears on a different register in the Interrupt Aggregator.
VCI_IN1	This interrupt is routed to the Interrupt Controller It is only asserted when both VBAT and VTR are powered. Edge detection and assertion level for the interrupt are configured in the GPIO Pin Control Register for the GPIO that shares the pin with VCI_IN# input. This interrupt is equivalent to the GPIO interrupt for the GPIO that shares the pin, but appears on a different register in the Interrupt Aggregator.
VCI_OVRD_IN	This interrupt is routed to the Interrupt Controller It is only asserted when both VBAT and VTR are powered. Edge detection and assertion level for the interrupt are configured in the GPIO Pin Control Register for the GPIO that shares the pin with VCI_OVRD_IN input. This interrupt is equivalent to the GPIO interrupt for the GPIO that shares the pin, but appears on a different register in the Interrupt Aggregator.

37.7 Low Power Modes

The VBAT-powered Control Interface has no low-power modes. It runs continuously while the VBAT well is powered.

37.8 General Description

The VBAT-Powered Control Interface (VCI) is used to drive the VCI_OUT pin. The output pin can be controlled either by VBAT-powered inputs, or by firmware when the VTR is active and the EC is powered and running. When the VCI_OUT pin is controlled by hardware, either because VTR is inactive or because the VCI block is configured for hardware control, the VCI_OUT pin can be asserted by a number of inputs:

- When either the VCI_IN0# pin or the VCI_IN1# is asserted. By default, the VCI_IN# pins are active low, but firmware can switch each input individually to an active-high input. See Section 37.8.1, "Input Polarity".
- Wen the VCI_OVRD_IN pin is asserted. The VCI_OVRD_IN pin is always active high
- When the POWER_UP_EVENT from the RTC/Week Timer is asserted

Firmware can configure which of the hardware pin inputs contribute to the VCI_OUT output by setting the enable bits in the VCI Input Enable Register. Even if the input pins are not configured to affect VCI_OUT, firmware can monitor their current state through the status bits in the VCI Register. Firmware can also enable EC interrupts from the state of the input pins.

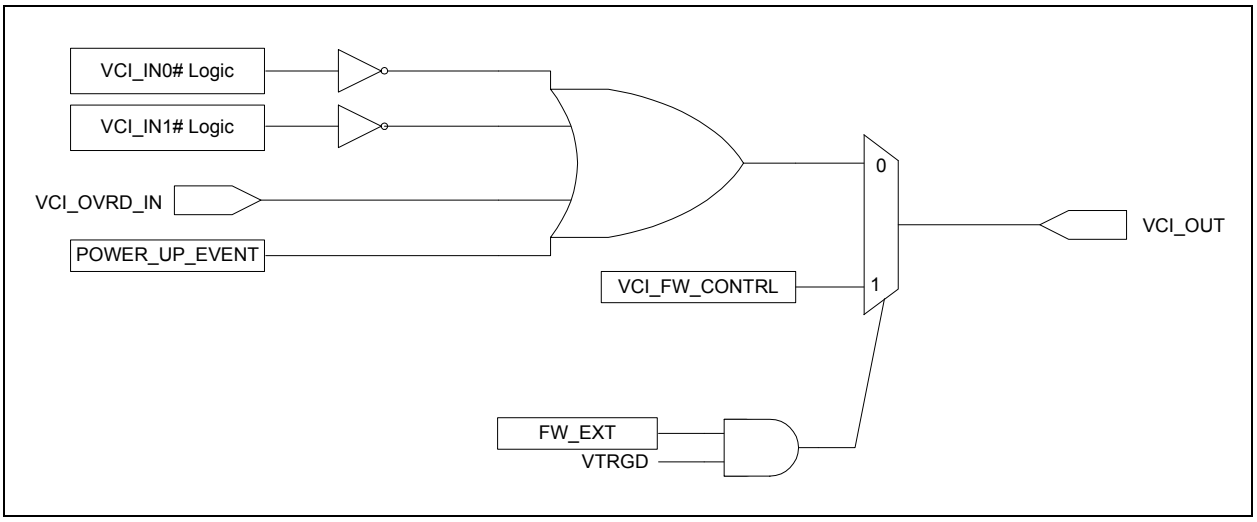
Each of the VCI_IN# pins can be configured for additional properties.

- By default, each of the VCI_IN# pins have an input glitch filter. All glitch filters can be disabled by the FILTERS_BYPASS bit in the VCI Register
- Assertions of each of the VCI_IN# pins can optionally be latched, so hardware can maintain the assertion of a VCI_IN# even after the physical pin is de-asserted, or so that firmware can determine which of the VCI_IN# inputs contributed to VCI_OUT assertion. See the Latch Enable Register and the Latch Resets Register.
- Rising edges and falling edges on the VCI_IN# pins are latched, so firmware can detect transitions on the VCI_IN# pins even if the transitions occurred while EC power was not available. See Section 37.8.2, "Edge Event Status".

When VTR power is present and the EC is operating, firmware can figure the VCI_OUT pin to operate as a general-purpose output pin. The VCI_OUT pin is firmware-controlled when the FW_EXT bit in the VCI Register is ‘1’. When firmware is controlling the output, the state of VCI_OUT is defined by the VCI_FW_CNTRL bit in the same register. When VTR is not present (the VTRGD input is low), the VCI_OUT pin is also determined by the hardware circuit.

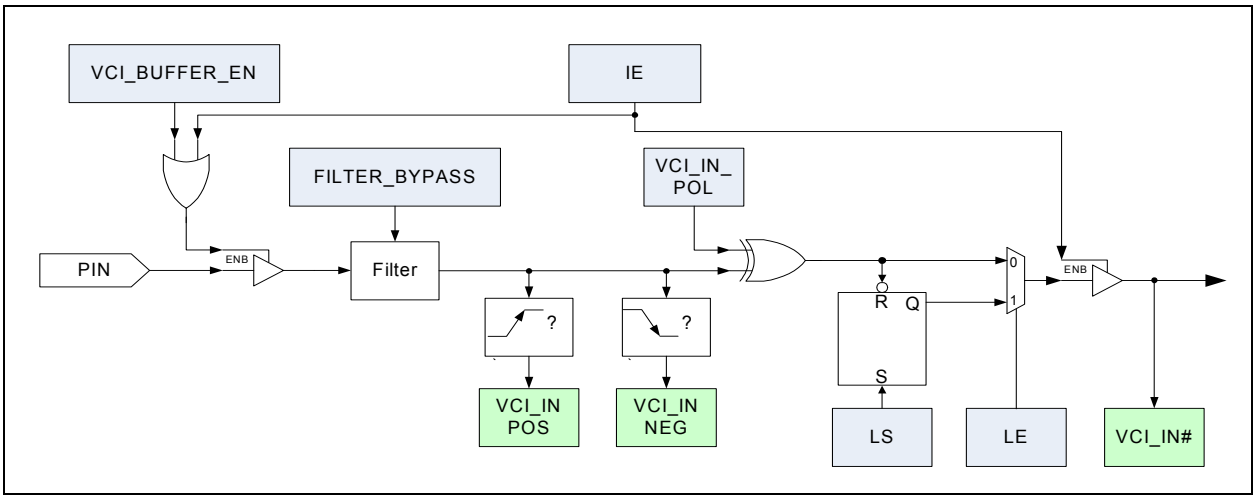
The following figures illustrate the VBAT-Power Control Interface logic:

FIGURE 37-2: VBAT-POWERED CONTROL INTERFACE BLOCK DIAGRAM



The VCI_INx# Logic in the block diagram is illustrated in the following figure:

FIGURE 37-3: VBAT-POWERED CONTROL INTERFACE BLOCK DIAGRAM



37.8.1 INPUT POLARITY

The VCI_IN# pins have an optional polarity inversion. The inversion takes place after any input filtering and before the VCI_IN signals are latched in the VCI_IN# status bits in the VCI Register. Edge detection occurs before the polarity inversion. The inversion is controlled by battery-backed configuration bits in the VCI Polarity Register.

37.8.2 EDGE EVENT STATUS

Each VCI_IN# input pin is associated with two register bits used to record edge transitions on the pins. The edge detection takes place after any input filtering, before polarity control and occurs even if the VCI_IN# input is not enabled as part of the VCI_OUT logic (the corresponding control bit in the VCI Input Enable Register is '0') or if the state of the VCI_IN# input is not latched (the corresponding control bit in the Latch Enable Register is '0'). One bit is set whenever there is a high-to-low transition on the VCI_IN# pin (the VCI Negedge Detect Register) and the other bit is set whenever there is a low-to-high transition on the VCI_IN# pin (the VCI Posedge Detect Register).

In order to minimize power drain on the VBAT circuit, the edge detection logic operates only when the input buffer for a VCI_IN# pin is enabled. The input buffer is enabled either when the VCI_IN# pin is configured to determine the VCI_OUT pin, as controlled by the VCI_IN[1:0]# field of the VCI Register, or when the input buffer is explicitly enabled in the VCI Input Enable Register. When the pins are not enabled transitions on the pins are ignored.

The VCI_OVRD input also has an Input Buffer Enable and an Input Enable bit associated with VCI_OUT. However, the VCI_OVRD input does not have any filtering, latching, input edge detection or polarity control.

37.8.3 VCI PIN MULTIPLEXING

Each of the VCI inputs, as well as VCI_OUT, are multiplexed with standard VTR-powered GPIOs. When VTR power is off, the mux control is disabled and the pin always reverts to the VCI function. The VCI_IN# function should be disabled in the VCI Input Enable Register for any pin that is intended to be used as a GPIO rather than a VCI_IN#, so that VCI_OUT is not affected by the state of the pin. The VCI_OVRD_IN function should similarly be disabled if the pin is to be used as a GPIO.

37.8.4 APPLICATION EXAMPLE

For this example, a mobile platform configures the VBAT-Powered Control Interface as follows:

- VCI_IN0# is wired to a power button on the mobile platform
- VCI_IN1# is wired to a power button on a dock
- VCI_OVRD_IN is wired so that it is asserted whenever AC power is present
- The VCI_OUT pin is connected to the regulator that sources the VTR power rail, the rail which powers the EC

The VBAT-Powered Control Interface can be used in a system as follows:

1. In the initial condition, there is no power on either the VTR or VBAT power rails. All registers in the VBAT-Powered Control Interface are in an indeterminate state
2. A coin cell battery is installed, causing a VBAT_POR. All registers in the interface are forced to their default conditions. The VCI_OUT pin is driven by hardware, input filters on the VCI_IN# pins are enabled, the VCI_IN# pins are all active low, all VCI inputs are enabled and all edge and status latches are in their non-asserted state
3. The power button on VCI_IN0# is pushed. This causes VCI_OUT to be asserted, powering the VTR rail. This causes the EC to boot and start executing EC firmware
4. The EC changes the VCI configuration so that firmware controls the VCI_OUT pin, and sets the output control so that VCI_OUT is driven high. With this change, the power button can be released without removing the EC power rail.
5. EC firmware re-configures the VCI logic so that the VCI_IN# input latches are enabled. This means that subsequent presses of the power button do not have to be held until EC firmware switches the VCI logic to firmware control
6. During this phase the VCI_OUT pin is driven by the firmware-controlled state bit and the VCI input pins are ignored. However, the EC can monitor the state of the pins, or generate inputs when their state changes
7. At some later point, EC firmware must enter a long-term power-down state.
   - Firmware configures the Week Timer for a Sub-Week Alarm once every 8 hours. This will turn on the EC power rail three times a day and enable the EC to perform low frequency housekeeping tasks even in its lowest-power state
   - Firmware de-asserts VCI_OUT. This action kills power to the EC and automatically returns control of the VCI_OUT pin to hardware.
   - The EC will remain in its lowest-power state until a power pin is pushed, AC power is connected, or the Sub-Week Alarm is active

37.9 EC-Only Registers

The addresses of each register listed in this section are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 37-6: EC-ONLY REGISTER BASE ADDRESS

INSTANCE NAME	INSTANCE NUMBER	HOST	ADDRESS SPACE	BASE ADDRESS
VBAT-Powered Control Interface	0	EC	32-bit internal address space	0000_D000h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 37-7: EC-ONLY REGISTER SUMMARY

REGISTER NAME	EC OFFSET
VCI Register	00h
Latch Enable Register	04h
Latch Resets Register	08h
VCI Input Enable Register	0Ch
Reserved	10h
VCI Polarity Register	14h
VCI Posedge Detect Register	18h
VCI Negedge Detect Register	1Ch
VCI Buffer Enable Register	20h

37.9.1 VCI REGISTER

Offset	00h			
Bits	**DESCRIPTION**	**TYPE**	**DEFAULT**	**RESET EVENT**
31:13	Reserved	R	-	-
12	FILTERS_BYPASS The Filters Bypass bit is used to enable and disable the input filters on the VCI_IN# pins. See Section 43.24, "VBAT-Powered Control Interface Timing," on page 531. 1=Filters disabled 0=Filters enabled (default)	R/W	0	VBAT_POR
11	FW_EXT This bit controls selecting between the external VBAT-Powered Control Interface inputs, or the VCI_FW_CNTRL bit output to control the VCI_OUT pin. 1=VCI_OUT is determined by the VCI_FW_CNTRL field, when VTR is active 0=VCI_OUT is determined by the external inputs.	R/W	0	nSYSRST & VBAT_POR
10	VCI_FW_CNTRL This bit can allow EC firmware to control the state of the VCI_OUT pin. For example, when VTRGD is asserted and the FW_EXT bit is '1', clearing the VCI_FW_CNTRL bit de-asserts the active high VCI_OUT pin. BIOS must set this bit to '1' prior to setting the FW_EXT bit to '1' on power up, in order to avoid glitches on the VCI_OUT pin.	R/W	0	
9	VCI_OUT This bit provides the current status of the VCI_OUT pin.	R	See Note 1	–
8	VCI_OVRD_IN This bit provides the current status of the VCI_OVRD_IN pin.	R	See Note 1	
7:2	Reserved	R	-	-
1:0	VCI_IN# These bits provide the latched state of the associated VCI_IN# pin, if latching is enabled or the current state of the pin if latching is not enabled. In both cases, the value is determined after the action of the VCI Polarity Register.	R	See Note 1	

Note 1: The VCI_IN[1:0]# and VCI_OVRD_IN bits default to the state of their respective input pins. The VCI_OUT bit is determined by the VCI hardware circuit.

37.9.2 LATCH ENABLE REGISTER

Offset	04h			
Bits	**DESCRIPTION**	**TYPE**	**DEFAULT**	**RESET EVENT**
31:2	Reserved	R	-	-
1:0	LE Latching Enables. Latching occurs after the Polarity configuration, so a VCI_IN# pin is asserted when it is '0' if VCI_IN_POL is '0', and asserted when it is '1 'if VCI_IN_POL is '1'. For each bit in the field: 1=Enabled. Assertions of the VCI_IN# pin are held until the latch is reset by writing the corresponding LS bit 0=Not Enabled. The VCI_IN# signal is not latched but passed directly to the VCI_OUT logic	R/W	00h	VBAT_POR

37.9.3 LATCH RESETS REGISTER

Offset	08h			
Bits	**DESCRIPTION**	**TYPE**	**DEFAULT**	**RESET EVENT**
31:2	Reserved	R	-	-
1:0	LS Latch Resets. When a Latch Resets bit is written with a '1', the corresponding VCI_IN# latch is de-asserted ('1'). The VCI_IN# input to the latch has priority over the Latch Reset input, so firmware cannot reset the latch while the VCI_IN# pin is asserted. Firmware should sample the state of the pin in the VCI Register before attempting to reset the latch. As noted in the Latch Enable Register, the assertion level is determined by the VCI_IN_POL bit. Reads of this register are undefined.	W	–	–

37.9.4 VCI INPUT ENABLE REGISTER

Offset	0Ch			
Bits	**DESCRIPTION**	**TYPE**	**DEFAULT**	**RESET EVENT**
31:9	Reserved	R	-	-
8	VCI_OVRD_INPUT_ENABLE After changing the input enable, firmware should clear any potential interrupt that may have been triggered by the input, as changing the enable may cause the internal status to change. 1=Enabled. This signal is not gated and toggling the pin will affect the VCI_OUT pin 0=Not Enabled. This signal is gated low and has no effect on the VCI_OUT pin	R/W	1h	VBAT_POR
7:2	Reserved	R	-	-
1:0	IE Input Enables for VCI_IN# signals. After changing the input enable for a VCI input, firmware should reset the input latch and clear any potential interrupt that may have been triggered by the input, as changing the enable may cause the internal status to change. For each bit in the field: 1=Enabled. The corresponding VCI_IN# input is not gated and toggling the pin will affect the VCI_OUT pin 0=Not Enabled. the corresponding VCI_IN# input does not affect the input status registers or the VCI_OUT pin, even if the input is ‘0.’ Latches are not asserted, even if the VCI_IN# pin is low, during a VBAT power transition	R/W	3h	VBAT_POR

37.9.5 VCI POLARITY REGISTER

Offset	14h			
Bits	**DESCRIPTION**	**TYPE**	**DEFAULT**	**RESET EVENT**
31:2	Reserved	R	-	-
1:0	VCI_IN_POL These bits determine the polarity of the VCI_IN input signals: For each bit in the field: 1=Active High. The value on the pins is inverted before use 0=Active Low (default)	RW	0	VBAT_POR

37.9.6 VCI POSEDGE DETECT REGISTER

Offset	18h			
Bits	**DESCRIPTION**	**TYPE**	**DEFAULT**	**RESET EVENT**
31:1	Reserved	R	-	-
1:0	VCI_IN_POS These bits record a low to high transition on the VCI_IN# pins. A “1” indicates a transition occurred. For each bit in the field: 1=Positive Edge Detected 0=No edge detected	RWC	0	VBAT_POR

37.9.7 VCI NEGEDGE DETECT REGISTER

Offset	1Ch			
Bits	**DESCRIPTION**	**TYPE**	**DEFAULT**	**RESET EVENT**
31:2	Reserved	R	-	-
1:0	VCI_IN_NEG These bits record a high to low transition on the VCI_IN# pins. A “1” indicates a transition occurred. For each bit in the field: 1=Negative Edge Detected 0=No edge detected	RWC	0	VBAT_POR

37.9.8 VCI BUFFER ENABLE REGISTER

Offset	20h			
Bits	**DESCRIPTION**	**TYPE**	**DEFAULT**	**RESET EVENT**
31:9	Reserved	R	-	-
8	VCI_OVRD_EN VCI_OVRD_IN Input Buffer Enable. After changing the buffer enable, firmware should clear any potential interrupt that may have been triggered by the input, as changing the buffer may cause the internal status to change. 1=VCI_OVRD_IN input buffer enabled independent of the VCI_OVRD_INPUT_ENABLE bit 0=VCI_OVRD_IN input buffer enabled by the VCI_OVRD_INPUT_ENABLE bit (default)	RW	0	VBAT_POR
7:2	Reserved	R	-	-
1:0	VCI_BUFFER_EN Input Buffer enable. After changing the buffer enable for a VCI input, firmware should reset the input latch and clear any potential interrupt that may have been triggered by the input, as changing the buffer may cause the internal status to change. For each bit in the field: 1=VCI_IN# input buffer enabled independent of the IE bit. The edge detection latches for this input are always enabled 0=VCI_IN# input buffer enabled by the IE bit. The edge detection latches are only enabled when the IE bit is ‘1’ (default)	RW	0	VBAT_POR

38.0 ANALOG TO DIGITAL CONVERTER

38.1 Introduction

This block is designed to convert external analog voltage readings into digital values. It consists of a single successive-approximation Analog-Digital Converter that can be shared with up to sixteen inputs. See Products on page 3 for the specific number of channels supported for a particular device.

Note: Transitions on ADC GPIOs are not permitted when Analog to Digital Converter readings are being taken.

38.2 References

No references have been cited for this chapter

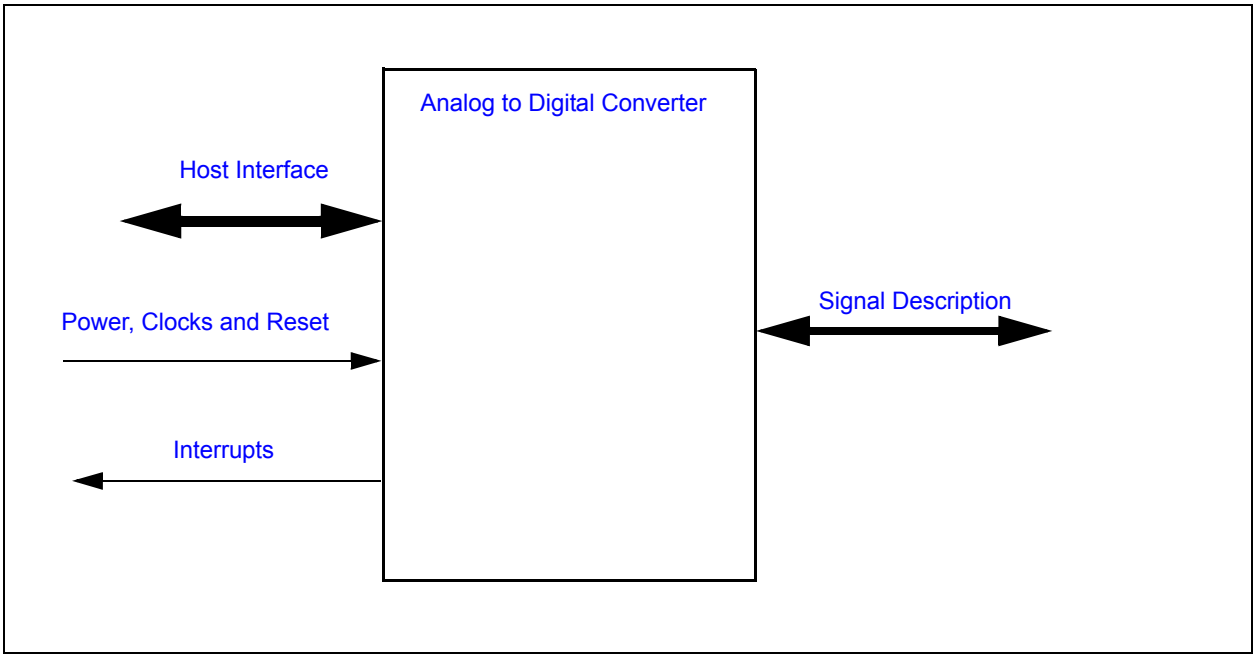
38.3 Terminology

No terminology is defined for this chapter

38.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 38-1: I/O DIAGRAM OF BLOCK



38.5 Signal Description

The Signal Description Table lists the signals that are typically routed to the pin interface.

TABLE 38-1: SIGNAL DESCRIPTION

Name	Direction	Description
ADC_VREF	Input	ADC Reference Voltage. This pin must either be connected to a very accurate 3.0V reference or connected to the same VTR power supply that is powering the ADC logic.
ADC 16:0	Input	ADC Analog Voltage Input 16:0 from pins Unused ports are connected to ground.

38.6 Host Interface

The registers defined for the Trace FIFO Debug Port are accessible by the various hosts as indicated in Section 38.11, "EC-Only Registers".

38.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

38.7.1 POWER DOMAINS

Name	Description
VTR	This power well sources the registers in this block.
VTR	This power well sources of the logic in this block, except where noted.
AVSS	This is the ground signal for the block.

38.7.2 CLOCK INPUTS

Name	Description
1.2MHz	This derived clock signal drives selected logic (1.2 MHz clock with a 50% duty cycle).

38.7.3 RESETS

Name	Description
nSYSRST	This reset signal resets all of the registers and logic in this block.

38.8 Interrupts

Source	Description
ADC_Single_Int	Interrupt signal from ADC controller to EC for Single-Sample ADC conversion.
ADC_Repeat_Int	Interrupt signal from ADC controller to EC for Repeated ADC conversion.

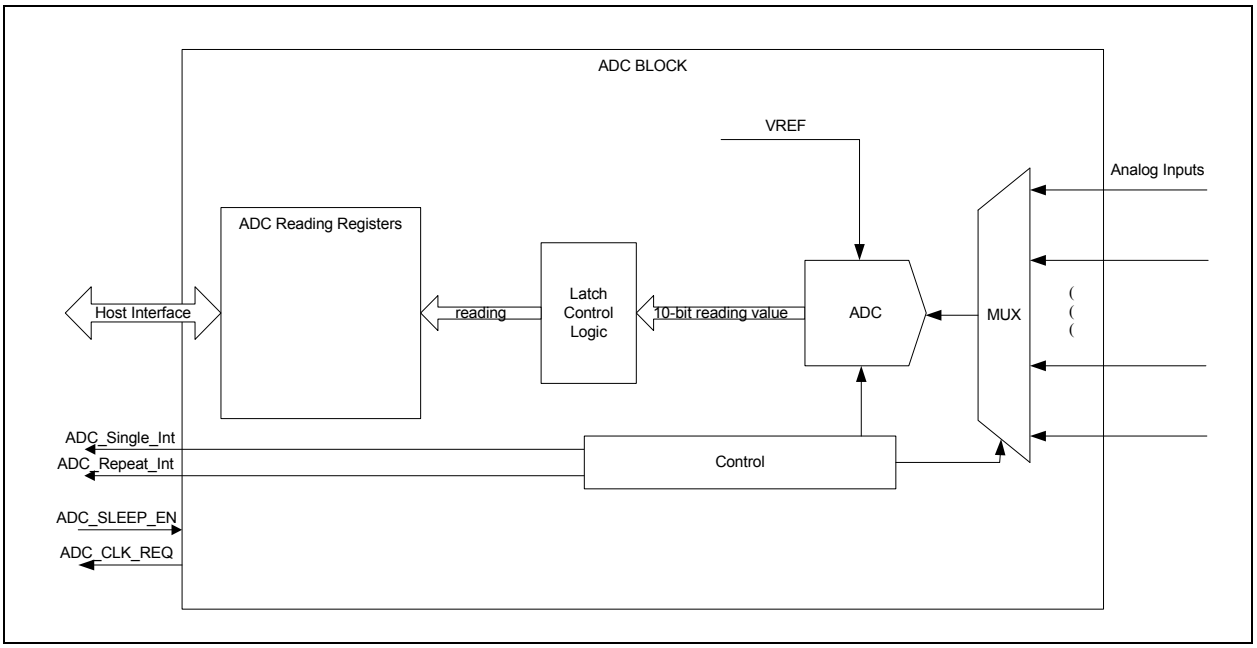
38.9 Low Power Modes

The ADC may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

The ADC is designed to conserve power when it is either sleeping or disabled. It is disabled via the Activate Bit and sleeps when the ADC_SLEEP_EN signal is asserted. The sleeping state only controls clocking in the ADC and does not power down the analog circuitry. For lowest power consumption, the ADC Activate bit must be set to '0.'

38.10 Description

FIGURE 38-2: ADC BLOCK DIAGRAM



The MEC140x/1x features successive approximation Analog to Digital Converter with up to sixteen channels. The ADC architecture features excellent linearity and converts analog signals to 10 bit words. Conversion takes less than 12 microseconds per 10-bit word. The sixteen channels are implemented with a single high speed ADC fed by a sixteen input analog multiplexer. The multiplexer cycles through the sixteen voltage channels, starting with the lowest-numbered channel and proceeding to the highest-number channel, selecting only those channels that are programmed to be active.

The input range on the voltage channels spans from 0V to the external voltage reference. With an external voltage reference of 3.0V, this provides resolutions of 2.9mV. The accuracy of any voltage reading depends on the accuracy and stability of the voltage reference input.

Note: The ADC pins are 3.3V tolerant.

The ADC conversion cycle starts either when the Start_Single bit in the ADC to set to 1 or when the ADC Repeat Timer counts down to 0. When the Start_Single is set to 1 the conversion cycle converts channels enabled by configuration bits in the ADC Single Register. When the Repeat Timer counts down to 0 the conversion cycle converts channels enabled by configuration bits in the ADC Repeat Register. When both the Start_Single bit and the Repeat Timer request conversions the Start_Single conversion is completed first.

Conversions always start with the lowest-numbered enabled channel and proceed to the highest-numbered enabled channel.

Note: If software repeatedly sets Start_Single to 1 at a rate faster than the Repeat Timer count down interval, the conversion cycle defined by the ADC Repeat Register will not be executed.

38.10.1 REPEAT MODE

- Repeat Mode will start a conversion cycle of all ADC channels enabled by bits Rpt_En[7:0] in the ADC Repeat Register. The conversion cycle will begin after a delay determined by Start_Delay[15:0] in the ADC Delay Register.
- After all channels enabled by Rpt_En[7:0] are complete, Repeat_Done_Status will be set to 1. This status bit is cleared when the next repeating conversion cycle begins to give a reflection of when the conversion is in progress.
- As long as Start_Repeat is 1 the ADC will repeatedly begin conversion cycles with a period defined by Repeat_Delay[15:0].
- If the delay period expires and a conversion cycle is already in progress because Start_Single was written with a 1, the cycle in progress will complete, followed immediately by a conversion cycle using Rpt_En[7:0] to control the channel conversions.

38.10.2 SINGLE MODE

- The Single Mode conversion cycle will begin without a delay. After all channels enabled by Single_En[7:0] are complete, Single_Done_Status will be set to 1. When the next conversion cycle begins the bit is cleared.
- If Start_Single is written with a 1 while a conversion cycle is in progress because Start_Repeat is set, the conversion cycle will complete, followed immediately by a conversion cycle using Single_En[7:0] to control the channel conversions.

38.11 EC-Only Registers

The registers listed in the Table 38-3, "Analog to Digital Converter Register Summary" are for a single instance of the Analog to Digital Converter block. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in Table 38-2, "Analog to Digital Converter Base Address".

TABLE 38-2: ANALOG TO DIGITAL CONVERTER BASE ADDRESS

Instance Name	Instance Number	Host	Address Space	Base Address
ADC	0	EC	32-bit internal address space	0000_7C00h

Note 38-1 The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 38-3: ANALOG TO DIGITAL CONVERTER REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	ADC Control Register
04h	ADC Delay Register
08h	ADC Status Register
0Ch	ADC Single Register
10h	ADC Repeat Register
14h	ADC Channel 0 Reading Register
18h	ADC Channel 1 Reading Register
1Ch	ADC Channel 2 Reading Register
20h	ADC Channel 3 Reading Register
24h	ADC Channel 4 Reading Register
28h	ADC Channel 5 Reading Register
2Ch	ADC Channel 6 Reading Register
30h	ADC Channel 7 Reading Register
34h	ADC Channel 8 Reading Register

Note: The unused channel reading registers are reserved. See Products on page 3 for the specific number of channels supported for a particular device.

38.11.1 ADC CONTROL REGISTER

The ADC Control Register is used to control the behavior of the Analog to Digital Converter.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	RESERVED	RES		
7	**Single_Done_Status** This bit is cleared when it is written with a 1. Writing a 0 to this bit has no effect. This bit can be used to generate an EC interrupt. 0: ADC single-sample conversion is not complete. This bit is cleared whenever an ADC conversion cycle begins for a single conversion cycle. 1: ADC single-sample conversion is completed. This bit is set to 1 when all enabled channels in the single conversion cycle.	R/WC	0h	nSYSRST

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
6	**Repeat_Done_Status** This bit is cleared when it is written with a 1. Writing a 0 to this bit has no effect. This bit can be used to generate an EC interrupt. 0: ADC repeat-sample conversion is not complete. This bit is cleared whenever an ADC conversion cycle begins for a repeating conversion cycle. 1: ADC repeat-sample conversion is completed. This bit is set to 1 when all enabled channels in a repeating conversion cycle complete.	R/WC	0h	nSYSRST
5	RESERVED	RES		
4	**Soft Reset** 1: writing one causes a reset of the ADC block hardware (not the registers) 0: writing zero takes the ADC block out of reset	R/W	0h	nSYSRST
3	**Power_Saver_Dis** 0: Power saving feature is enabled. The Analog to Digital Converter controller powers down the ADC between conversion sequences. 1: Power saving feature is disabled.	R/W	0h	nSYSRST
2	**Start_Repeat** 0: The ADC Repeat Mode is disabled. Note: This setting will not terminate any conversion cycle in process, but will inhibit any further periodic conversions. 1: The ADC Repeat Mode is enabled. This setting will start a conversion cycle of all ADC channels enabled by bits Rpt_En[7:0] in the ADC Repeat Register.	R/W	0h	nSYSRST
1	**Start_Single** 0: The ADC Single Mode is disabled. 1: The ADC Single Mode is enabled. This setting starts a single conversion cycle of all ADC channels enabled by bits Single_En[7:0] in the ADC Single Register. **Note:** This bit is self-clearing	R/W	0h	nSYSRST
0	**Activate** 0: The ADC is disabled and placed in its lowest power state. Note: Any conversion cycle in process will complete before the block is shut down, so that the reading registers will contain valid data but no new conversion cycles will begin. 1: ADC block is enabled for operation. Start_Single or Start_Repeat can begin data conversions by the ADC. Note: A reset pulse is sent to the ADC core when this bit changes from 0 to 1.	R/W	0h	nSYSRST

38.11.2 ADC DELAY REGISTER

The ADC Delay register determines the delay from setting Start_Repeat in the ADC Control Register and the start of a conversion cycle. This register also controls the interval between conversion cycles in repeat mode.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	**Repeat_Delay[15:0]** This field determines the interval between conversion cycles when Start_Repeat is 1. The delay is in units of 40µs. A value of 0 means no delay between conversion cycles, and a value of 0xFFFF means a delay of 2.6 seconds. This field has no effect when Start_Single is written with a 1.	R/W	0000h	nSYSRST
15:0	**Start_Delay[15:0]** This field determines the starting delay before a conversion cycle is begun when Start_Repeat is written with a 1. The delay is in units of 40µs. A value of 0 means no delay before the start of a conversion cycle, and a value of 0xFFFF means a delay of 2.6 seconds. This field has no effect when Start_Single is written with a 1.	R/W	0000h	nSYSRST

38.11.3 ADC STATUS REGISTER

The ADC Status Register indicates whether the ADC has completed a conversion cycle.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	RESERVED	RES		
7:0	**ADC_Ch_Status[7:0]** **All bits are cleared by being written with a '1'.** 0: conversion of the corresponding ADC channel is not complete 1: conversion of the corresponding ADC channel is complete Note: for enabled single cycles, the Single_Done_Status bit in the ADC Control Register is also set after all enabled channel conversion are done; for enabled repeat cycles, the Repeat_Done_Status in the ADC Control Register is also set after all enabled channel conversion are done. See Note 38-2.	R/WC	00h	nSYSRST

Note 38-2 Bits that correspond to the unused channels are reserved. See Products on page 3 for the specific number of channels supported for a particular device.

38.11.4 ADC SINGLE REGISTER

The ADC Single Register is used to control which ADC channel is captured during a Single-Sample conversion cycle initiated by the Start_Single bit in the ADC Control Register.

APPLICATION NOTE: Do not change the bits in this register in the middle of a conversion cycle to insure proper operation.

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	RESERVED	RES		
7:0	**Single_En[7:0]** 0: single cycle conversions for this channel are disabled 1: single cycle conversions for this channel are enabled Each bit in this field enables the corresponding ADC channel when a single cycle of conversions is started when the Start_Single bit in the ADC Control Register is written with a 1. See Note 38-2.	R/W	00h	nSYSRST

38.11.5 ADC REPEAT REGISTER

The ADC Repeat Register is used to control which ADC channels are captured during a repeat conversion cycle initiated by the Start_Repeat bit in the ADC Control Register.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	RESERVED	RES		
7:0	**Rpt_En[7:0]** 0: repeat conversions for this channel are disabled 1: repeat conversions for this channel are enabled Each bit in this field enables the corresponding ADC channel for each pass of the Repeated ADC Conversion that is controlled by bit Start_Repeat in the ADC Control Register. See Note 38-2.	R/W	00h	nSYSRST

38.11.6 ADC CHANNEL READING REGISTERS

All 8 ADC channels return their results into a 32-bit reading register. In each case the low 10 bits of the reading register return the result of the Analog to Digital conversion and the upper 22 bits return 0. Table 38-3, “Analog to Digital Converter Register Summary,” on page 461 shows the addresses of all the reading registers.

Note 38-3 The ADC Channel Reading Registers access require single 16, or 32 bit reads; i.e., two 8 bit reads cannot ensure data coherency.

Offset	See Table 38-3, "Analog to Digital Converter Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:10	RESERVED	RES		
9:0	**ADCx_[9:0]** This read-only field reports the 10-bit output reading of the Input ADCx.	R/W	000h	nSYSRST

39.0 DIGITAL TO ANALOG CONVERTER

39.1 Overview

The Digital to Analog Converter generates an analog output voltage based on a digital input.

39.2 References

No references have been cited for this feature.

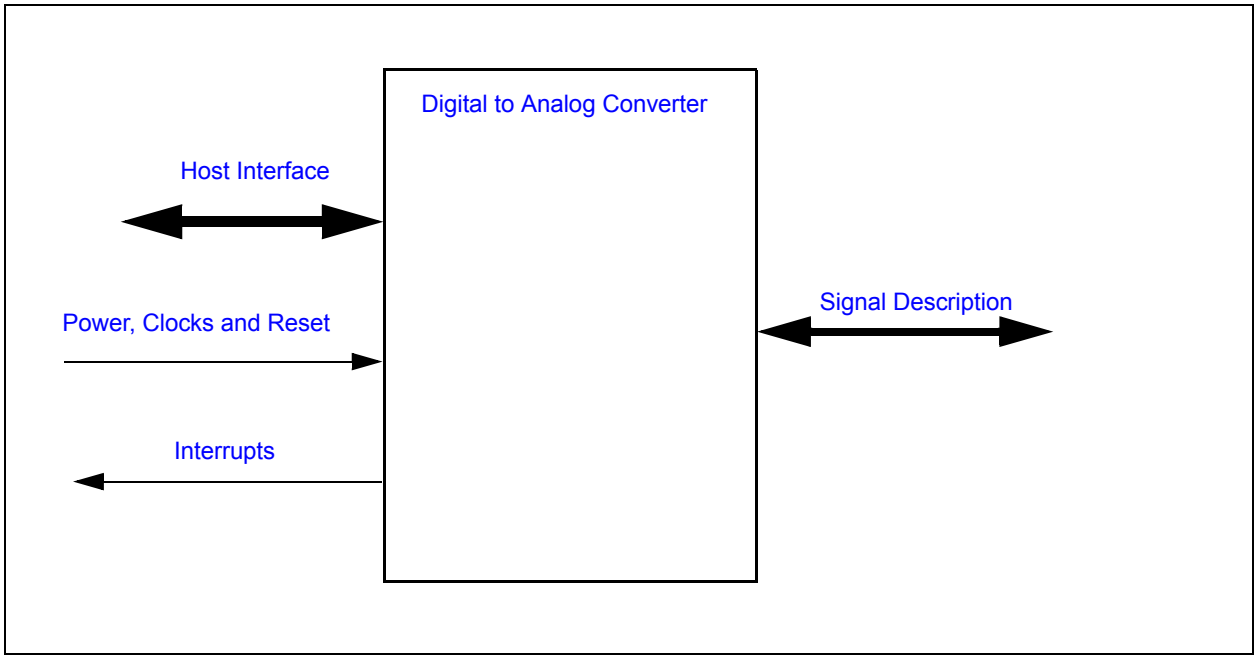
39.3 Terminology

There is no terminology defined for this section.

39.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 39-1: I/O DIAGRAM OF BLOCK



39.5 Signal Description

TABLE 39-1: SIGNAL DESCRIPTION

Name	Direction	Description
DAC_VREF	Input	DAC reference voltage
DAC	Output	DAC output pin

39.6 Host Interface

The registers defined for the Keyboard Scan Interface are accessible by the various hosts as indicated in Section 39.11, "EC-Only Registers".

39.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

39.7.1 POWER DOMAINS

Name	Description
VTR	The logic and registers implemented in this block are powered by this power well.

39.7.2 CLOCK INPUTS

Name	Description
48 MHz Ring Oscillator	This is the clock source for Keyboard Scan Interface logic.

39.7.3 RESETS

Name	Description
nSYSRST	This signal resets all the registers and logic in this block to their default state.
RESET_DAC	This signal resets all registers except the DAC Activate Register to their default state. It is asserted when either of the following is asserted: • nSYSRST • DAC_VREF SOFT_RESET

39.8 Interrupts

There are no interrupts from this block.

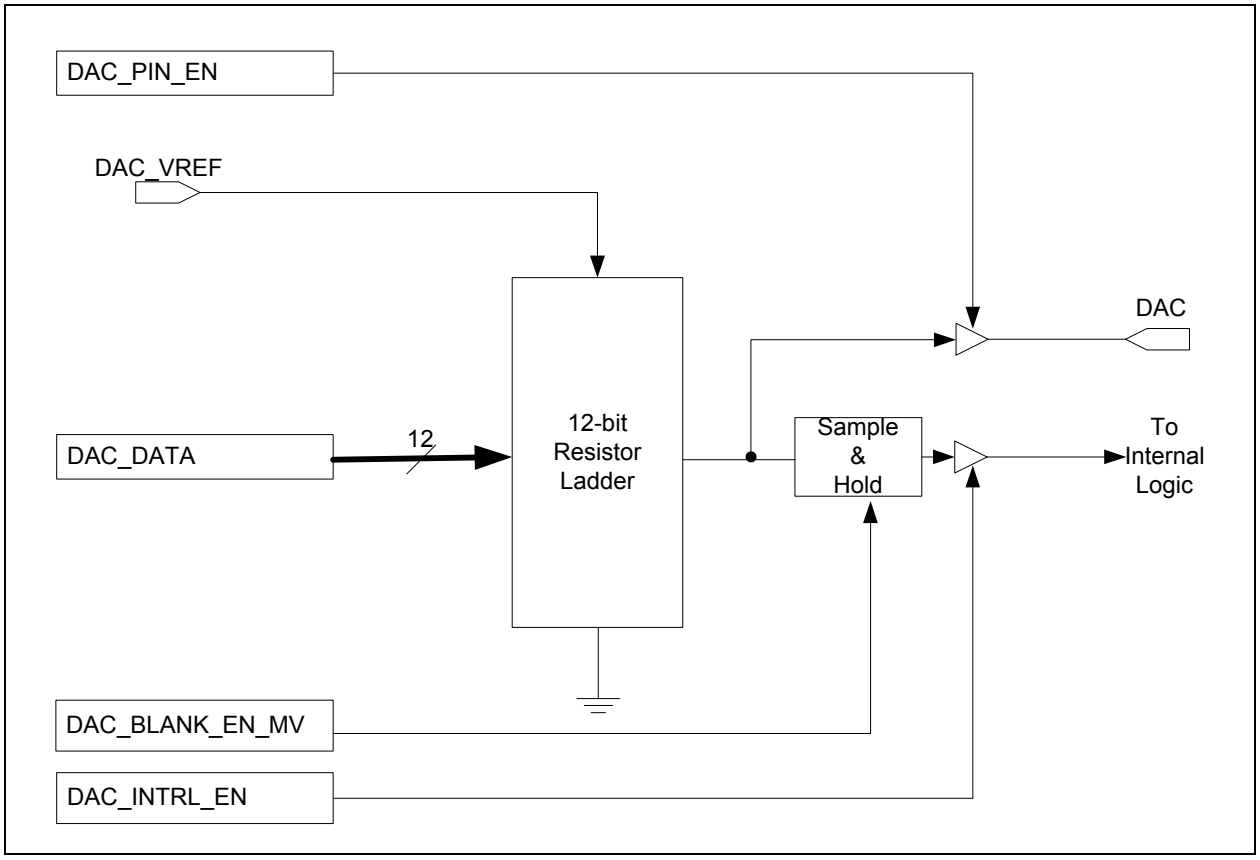
39.9 Low Power Modes

The DAC may be in the following power states:

- Deactivated. This mode is entered when the ACTIVATE i s'0'. The DAC analog circuitry is off and clocks are gated. Registers may be read or written
- Sleeping. This mode is entered when the EC asserts the SLEEP_EN for the DAC and DAC sleep is configured by the DAC_VREF SLEEP_CONTROL bit. The DAC analog circuitry is off and clocks are gated
- Off. The DAC_ON bit is '0'. Analog circuitry is off
- On. The DAC_ON bit is '1'. Analog circuitry is on and maintaining a constant value
- Converting. Either the DAC_ON bit or the DAC_DATA field have changed state.

39.10 Description

FIGURE 39-2: Digital to Analog Converter Block Diagram



The DAC generates an analog output voltage based on a digital input code. The output of the DAC may be routed to a pin, as well as to another internal device, such as an analog comparator. The DAC output is calculated according to the following formula:

$$DAC = DAC\_VREF \times \frac{DATA}{4095}$$

Where:

DAC = Output of the Digital Analog Converter, either on the pin or the internal logic
DAC_VREF = The voltage reference for the DAC
DATA = The contents of the DAC Data Register

The DAC features:

- Precision 12-bit resistor ladder
- 1M sample per second sample rates
- Reference input voltage from 0.5V to VTR
- Buffered output voltages
- A Sample and Hold circuit for reducing switching glitches on internal logic

39.10.1 DAC PROGRAMMING

The following sequence should be used to turn on the DAC:

1. Set the ACTIVATE bit to '1'
2. Program the DAC Configuration Register appropriately
3. Program the DAC Data Register with the required data value
4. Set the DAC Control Register bit to '1', enabling the DAC

The following sequence should be used to update the DAC output if the DAC is already enabled:

1. Program the DAC Data Register with the required data value. No other action is required.

The following sequence should be used to disable the DAC:

1. Set the DAC Control Register bit to '0',. No other action is required.

39.11 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Keyboard Scan Interface. The addresses of each register listed in this table are defined as a relative offset to the host "Base Address" defined in the EC-Only Register Base Address Table.

TABLE 39-2: EC-ONLY REGISTER BASE ADDRESS

Block Instance	Instance Number	Host	Address Space	Base Address
DAC	0	EC	32-bit internal address space	8000h
DAC	1	EC	32-bit internal address space	8040h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 39-3: EC-ONLY REGISTER SUMMARY

Offset	Register Name
0h	DAC Activate Register
4h	DAC Configuration Register
8h	DAC Control Register
Ch	DAC Data Register
10h	Microchip Reserved
14h	Microchip Reserved
18h	Microchip Reserved
1Ch	Microchip Reserved
20h	Microchip Reserved
24h	Microchip Reserved

TABLE 39-3: EC-ONLY REGISTER SUMMARY (CONTINUED)

Offset	Register Name
28h	Microchip Reserved
2Ch	Microchip Reserved
30h	Microchip Reserved

Microchip Reserved registers must not be modified.

39.11.1 DAC ACTIVATE REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	R	-	-
0	ACTIVATE 1=Block is active. The DAC may be turned on 0=Block disabled. The DAC is in its lowest power state and cannot be enabled	R/W	0h	nSYSRST

39.11.2 DAC CONFIGURATION REGISTER

Note: The DAC Configuration register can only be modified when the DACON bit in the DAC Control Register is zero.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	R	-	-
2	DAC_VREF SLEEP_CONTROL 1=The DAC responds to its Sleep_Enable input. This DAC output is tristated when the chip is sleeping. **Note:** If it is not desired to have the DAC start operating following a wake event, then it must be disabled prior going to sleep. 0=DAC ignores its Sleep_Enable input. The DAC output is remains unchanged when the chip is sleeping.	R/W	0h	RESET_DAC

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	DAC_VREF PIN_EN 1=DAC pin output buffer enabled; DAC output available on DAC pin 0=DAC pin output buffer disabled	R/W	0h	RESET_DAC
0	DAC_VREF INTRL_EN 1=DAC internal output buffer enabled; DAC output available to internal logic 0=DAC internal output buffer disabled	R/W	0h	RESET_DAC

39.11.3 DAC CONTROL REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	R	-	-
1	DAC_VREF SOFT_RESET This is a self-clearing bit. Setting this bit to '1' will reset all logic in the DAC block except the DAC Activate Register. Writing a '0' to this bit has no effect. Software should wait at least 150n after setting this bit to '1' before setting DAC_ON in this register to '1'.	W	0h	RESET_DAC
0	DAC_ON 1=DAC is turned out. The analog value of the DAC Data Register will be reflected on the DAC pin, if DAC_VREF PIN_EN is '1', and to internal logic, if DAC_VREF INTRL_EN is '1' 0=DAC is turned off	R/W	0h	RESET_DAC

39.11.4 DAC DATA REGISTER

Offset	Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:12	Reserved	R	-	-
11:0	DAC_DATA This data is converted by DAC to an analog voltage. All 12 bits must be written at the same time.	R/W	0h	RESET_DAC

40.0 ANALOG COMPARATOR

40.1 Overview

The Analog Comparator compares the analog voltage on an input pin to a reference voltage and generates an output that indicates the result of the comparison.

40.2 References

No references have been cited for this feature.

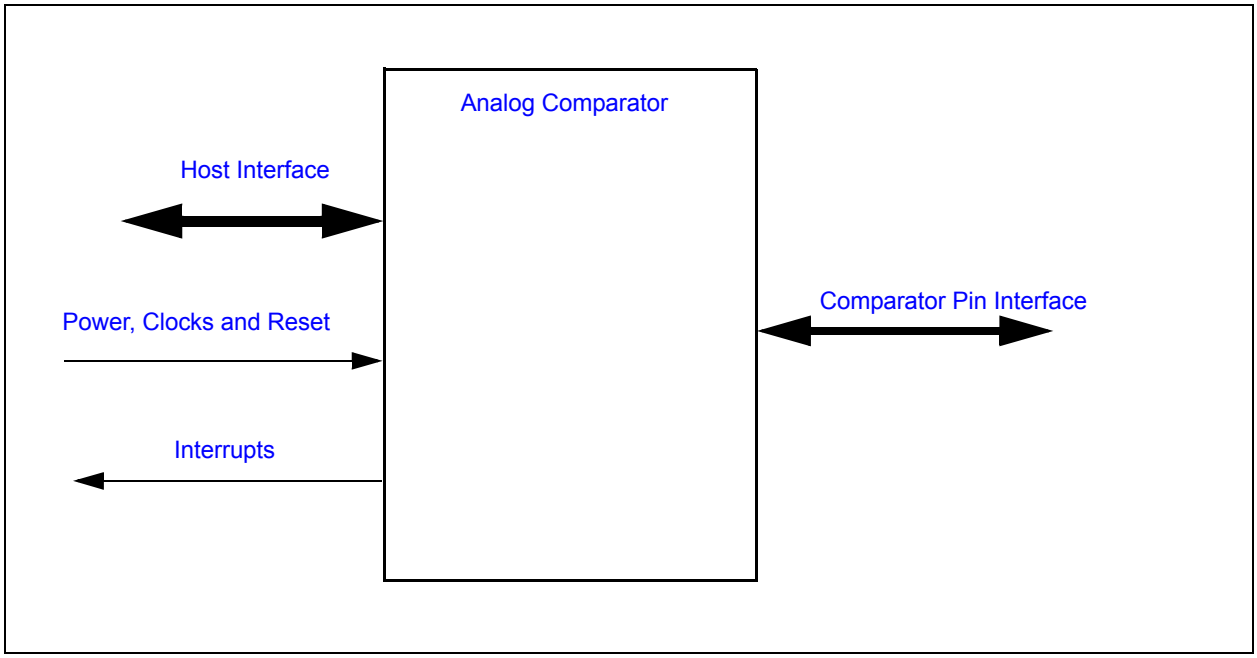
40.3 Terminology

There is no terminology defined for this section.

40.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 40-1: I/O DIAGRAM OF BLOCK



40.5 Comparator Pin Interface

TABLE 40-1: SIGNAL DESCRIPTION

Name	Direction	Description
CMP_VREF0	Input	Negative voltage input for Comparator 0
CMP_VREF1	Input	Negative voltage input for Comparator 1
CMP_VIN0	Input	Positive voltage input for Comparator 0
CMP_VIN1	Input	Positive voltage input for Comparator 1

TABLE 40-1: SIGNAL DESCRIPTION (CONTINUED)

Name	Direction	Description
CMP_VOUT0	Output	Comparator 0 output
CMP_VOUT1	Output	Comparator 1 output

40.6 Host Interface

The registers defined for the Comparator Interface are only accessible by the embedded controller. The Comparator Registers for both comparators are located in one register in the EC Subsystem register bank. See Section 34.8.2, "Comparator Control," on page 435.

40.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

40.7.1 POWER DOMAINS

Name	Description
VTR	The logic implemented in this block are powered by this power well.

40.7.2 CLOCK INPUTS

This component does not require a clock input.

40.7.3 RESETS

Name	Description
VTR_RESET#	This signal resets all the register in the EC Subsystem that interact with the comparators.

40.8 Interrupts

The comparators do not have a dedicated interrupt output event. An interrupt can be generated by the GPIO which shares the pin with the comparator output signal.

- GPIO124/CMP_VOUT0
- GPIO120/CMP_VOUT1

The GPIO interrupt is very configurable, thereby allowing CMP_VOUTx signal to generate an event when the CMP_VINx input is greater than the CMP_VREFx input or when it is less than the CMP_VREFx input. See the definition of Bits[7:4] of the Pin Control Register on page 329.

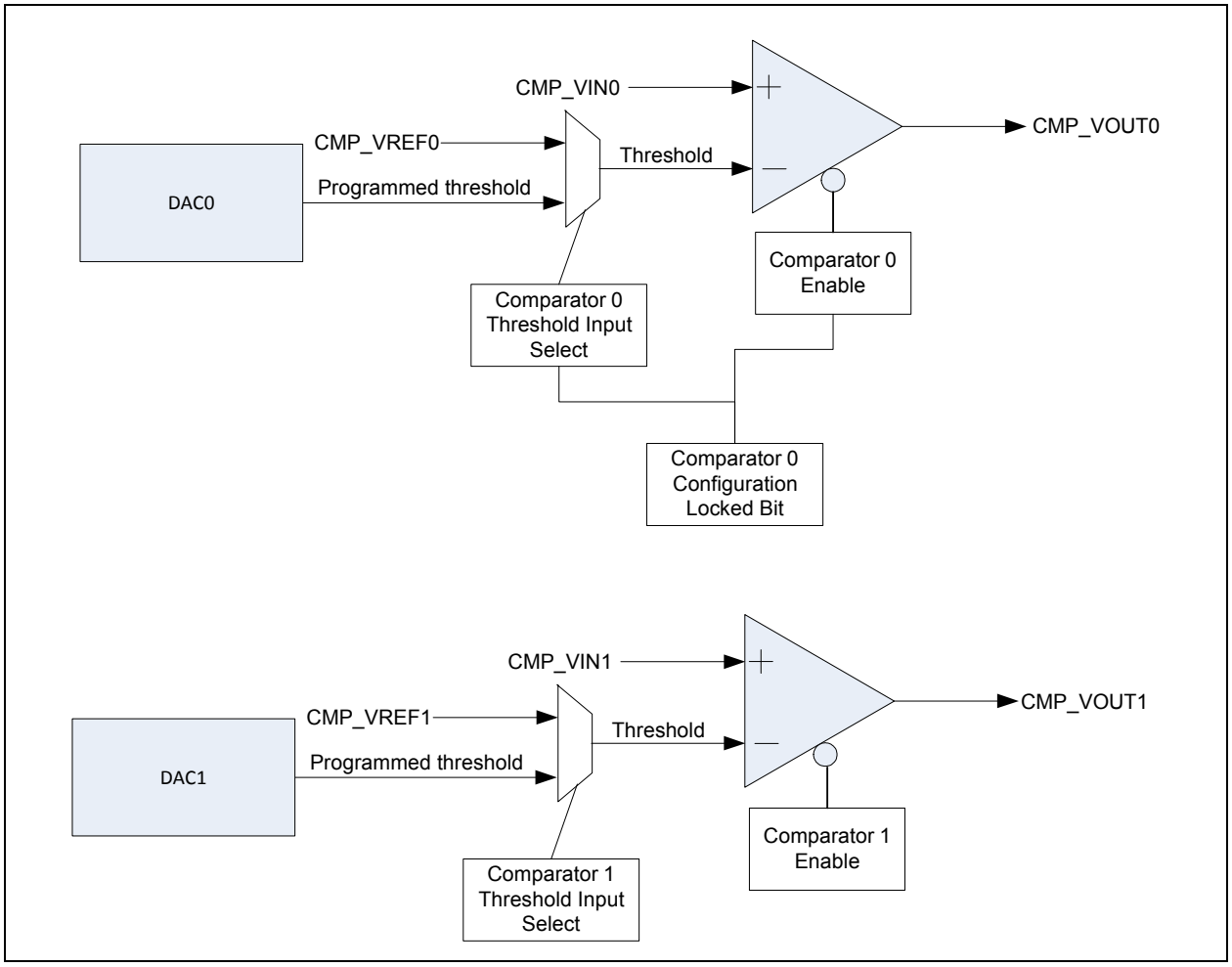
40.9 Low Power Modes

Each comparator is in its lowest powered state when its ENABLE bit is ‘0’.

40.10 Description

The Analog Comparator compares the analog voltage on an input pin to a reference voltage and generates an output that indicates the result of the comparison. The reference voltage can be derived either from an external pin or from the internal Digital Analog Converter.

FIGURE 40-2: COMPARATOR BLOCK DIAGRAM



The Analog Comparator compares the analog voltage on an input pin to a reference voltage and generates an output that indicates the result of the comparison. The reference voltage is derived either from an external source, on the CMP_VREFx input, or from the internal DAC, as configured by the COMPARATOR x THRESHOLD INPUT SELECT bit in the Comparator Control register.

The GPIO that shares a pin with the CMP_VOUT signal can be used to generate an interrupt to the EC when the pin multiplexer is configured for CMP_VOUT. The GPIO Pin Control Register is configured for the desired interrupt behavior (level or edge). Changes in the CMP_VOUT output signal will be reflected in the Interrupt Status register field for the GPIO, as configured in the GPIO Pin Control Register.

The control bits for Comparator 0 can be locked. The COMPARATOR 0 THRESHOLD INPUT SELECT and COMPARATOR 0 ENABLE bits are locked if the LOCK bit for Comparator 0 is set. Once the LOCK bit is set, neither COMPARATOR 0 THRESHOLD INPUT SELECT or COMPARATOR 0 ENABLE can be modified until the device is power cycled.

40.11 Comparator Registers

Control and status for both comparators are located in one register in the EC Subsystem register bank. See Section 34.8.2, "Comparator Control".

41.0 TEST MECHANISMS

41.1 Introduction

This device has the following test mechansims:

- 2-pin processor debug port (ICSP)
- 2-pin UART debug port
- 2-pin Trace FIFO port
- XNOR Chain for board connectivity test

This section defines the ICSP Controller and XNOR Chain for board test.

The UART is defined in Section 17.0, "UART," on page 267 and the Trace FIFO is defined in Section 32.0, "Trace FIFO Debug Port (TFDP)," on page 420.

41.2 References

No references have been cited for this chapter.

41.3 Terminology

Term	Definition
ICSP	In-Circuit Serial Programmer™

41.4 ICSP Controller

The ICSP Controller is the pin interface to the MIPs M14K EJTAG port.

41.4.1 INTERFACE

TABLE 41-1: ICSP 2-PIN PORT LIST

Signal Name	Direction	Description
ICSP_CLK	Input	Test Clock
ICSP_DATA	I/O	Bi-directional Test Data
ICSP_MCLR	Input	Test Reset, low active (Note 41-1). Also referred to as MCLR# **Note:** This signal has an internal pull-up.

Note 41-1 The ICSP_MCLR input provides the Reset. Note that the reset state of the ICSP port is only local to the port: its effect is to keep the port in an idle state and to disengage it from the rest of the system, so that it does not affect other on-chip logic in this state.

41.4.2 POWER, CLOCKS, AND RESET

41.4.2.1 Power Domains

Name	Description
VTR	The ICSP Controller logic and registers are implemented on this single power domain.

41.4.2.2 Clocks

The ICSP port runs internally from the externally-provided ICSP_CLK clock pulses only. There is no requirement for ICSP_CLK to be constantly running.

41.4.2.3 Reset

Name	Description
nSYSRST	Power On Reset for ICSP controller and registers
JTAG_RST#	Active-Low Test Reset Signal. Generated by toggling ICSP_MCLR low.

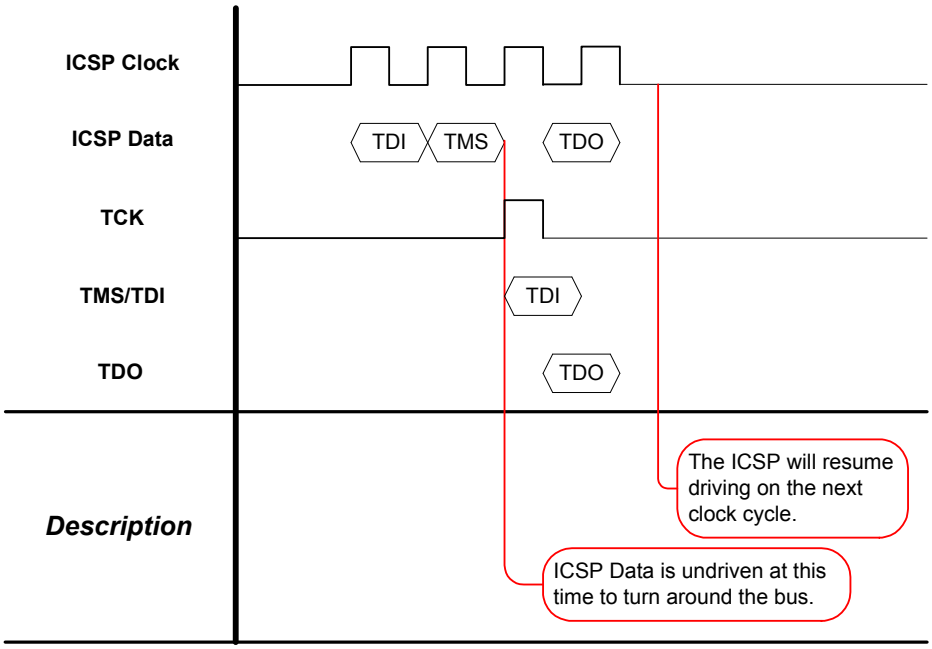
41.4.3 ICSP TEST MODES

The ICSP block supports TMOD0 .

- TMOD0 supports 2-wire ICSP JTAG

TMOD0 converts 2-wire ICSP signaling (Clock and Data) to standard 4-wire JTAG signaling (TCK, TMS, TDI and TDO). Doing this conversion has a cost of four clocks, therefore four ICSP_CLK pulses is equivalent to one JTAG clock (i.e., 4x slowdown).

FIGURE 41-1: ICSP-TO-JTAG CONVERSION TIMING (4 CLOCKS)



41.4.4 INSTRUCTION REGISTERS

TABLE 41-2: PUBLIC INSTRUCTIONS

Instruction	Description
IDCODE <0x01>	JTAG Standard IDCODE Register
SAMPLE/PRELOAD <0x02>	Not implemented, but reserved as required by JTAG standard.
SWTAP_CHIP <0x04>	Turn the Chip TAP back on and disable all other TAPs.
SWTAP <0x05>	Turn off the Chip TAP and enable all other TAPs behind it.
EXTEST <0x06>	Not implemented, but reserved as required by JTAG standard.
MCHP_CMD <0x07>	Chip Status interrogation and manual reset control.
BYPASS <0x1F>	Standard JTAG Bypass.

41.4.4.1 IDCODE

The MTAP JTAG ID Code is 0214_2445h.

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	VERSION	R	0h	nSYSRST
27:12	PartNumber	R	2142h	nSYSRST
11:1	ManufID	R	222h	nSYSRST
0	RESERVED Hard-coded to 1h.	R	1h	nSYSRST

41.4.4.2 SAMPLE/PRELOAD <0x02>

Not implemented, but reserved as required by JTAG standard.

41.4.4.3 SWTAP_CHIP <0x04>

Turn the Chip TAP back on and disable all other TAPs.

41.4.4.4 SWTAP <0x05>

Turn off the Chip TAP and enable all other TAPs behind it.

41.4.4.5 EXTEST <0x06>

Not implemented, but reserved as required by JTAG standard.

41.4.4.6 MCHP_CMD <0x07>

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	MCHP_CMD Microchip Command Register supports the following commands. • Command 0x00: Read Status Bit [7] Boot Into User Code Status (read-only) 0=Boot ROM will boot normally following a AssertDeviceReset command 1=Boot ROM will stall after setting the Boot Ready status bit allowing the ICSP debugger to load code into SRAM following a AssertDeviceReset command. Bit [6] BRDY; Boot Ready (read-only) 0 = eJTAG access is not enabled. 1 = Boot ROM is done initializing the device and has enabled eJTAG interface. This bit is cleared by H/W on nSYSRST. Bit [5:4] RESERVED Bit [3] CFGRDY; Configuration Ready (read-only) 0 = MTAP Device ID Not Valid 1 = MTAP Device ID Valid. Bit [2] RESERVED Bit [1] SLEEPING 0 = 48 MHz Ring Oscillator is running 1 = The device is sleeping. 48 MHz Ring Oscillator is not running Bit [0] DEVRST; Device Reset Status (read-only) 0 = MTAP Device Reset is deasserted 1 = MTAP Device Reset is asserted **Note:** The MTAP Device Reset is equivalent to a VTR POR, except the MTAP registers are not reset.	R/W	0h	JTAG_RST#

Offset	01h (continued)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Command 0x08: DeviceID Bit [31:16] Device ID Bit [15:8] Sub ID Bit [7:0] Revision ID The Device ID, Sub Id, and Revision ID are a reflection of the value in the same registers defined in Table 7-2, "Chip-Level (Global) Control/Configuration Registers," on page 149 starting at offset 1Ch. **Note:** This command only executes while Configuration Ready (CFGRDY) is set. Command 0x09: SetBootIntoUserCode Sets the MTAP Boot into User Code status bit. **Note:** This function only works while the Boot Ready status is set. **Command 0x0A: ClrBootIntoUserCode** Clears the MTAP Boot into User Code status bit. **Note:** This function only works while the Boot Ready status is set. Command 0xD1: AssertDeviceReset Causes a VTR POR. Test functions remain uneffected. **Note:** DEVRST, Device Reset Status, reflects the state of the reset event. Command 0xD0: DeassertDeviceReset Clears the AssertDeviceReset.	R/W	0h	JTAG_ RST#

41.4.4.7 BYPASS <0x1F>

Standard JTAG BYPASS. TDI connected to TDO via a 1-bit Bypass register.

41.4.5 TEST MODE ENTRY

The MCLR pin is used as MCLR for the ICSP interface. The device pulls this signal high internally. The debug connector must drive this signal correctly to enter ICSP modes.

41.4.5.1 Entry Sequence

To Enter ICSP:

1. Drive MCLR# High.
2. Drive ICSP_CLK and ICSP_DAT Low.
3. Drive MCLR# Low.
4. Send down 32 ICSP Clocks with the Test Mode Entry Code.
5. Drive MCLR# High

41.4.5.2 Test Mode Entry Codes

Test Mode	Test Mode Entry Code	Description
TMOD0	4D43 4850 “MCHP”	2-wire ICSP

41.4.5.3 Enabling EJTAG Interface

By default the EJTAG interface is disabled. It is gated by the MTAP (MCHP_TAP) controller. MTAP gates all other TAP controllers TDI so they always operate in BYPASS mode. There are two ISCP commands used to enable/disable the MTAP gating.

- IR SWTAP_CHIP (5'h04).
  - Enables the MTAP and gates the EJTAG interface behind it.
- IR SWTAP (5'h05).
  - Disables the MTAP and enables the EJTAG interface behind it.

The steps to enter EJTAG(M14K) are:

1. Drive MCLR# High.
2. Drive ICSP_CLK and ICSP_DAT Low.
3. Drive MCLR# Low.
4. Send down 32 ICSP Clocks with the following pattern on the ICSP_DAT pin (32'h4D434850).
5. Drive MCLR# High.
6. Send down the following IR SWTAP_CHIP (5'h04).
   - This will enable the MTAP.
7. Send down the following IR MCHP_CMD (5'h07).
   - This puts the DR in MTAP IR: MCHP_CMD
8. Poll 1 byte on the DR Shift until Bit [6] of the byte is 1. Always shift in 0x00.
   - The 0x00 shifted in the sub-command Read Status.
   - This is polling until the Boot ROM has opened up access to the part (JTAG Security).
9. Send down the following IR SWTAP (5'h05).
   - Disables the MTAP and enables the EJTAG behind it.
10. Run EJTAG program here.

41.5 XNOR Chain

41.5.1 OVERVIEW

The XNOR Chain test mode provides a means to confirm that all MEC140x/1x pins are in contact with the motherboard during assembly and test operations.

An example of an XNOR Chain test structure is illustrated below in . When the XNOR Chain test mode is enabled all pins, except for the Excluded Pins shown in Section 38.5.2, are disconnected from their internal functions and forced as inputs to the XNOR Chain. This allows a single input pin to toggle the XNOR Chain output if all other input pins are held high or low. The XNOR Chain output is the Test Output Pin (XNOR_OUT): GPIO027/KSO00/PVT_IO1.

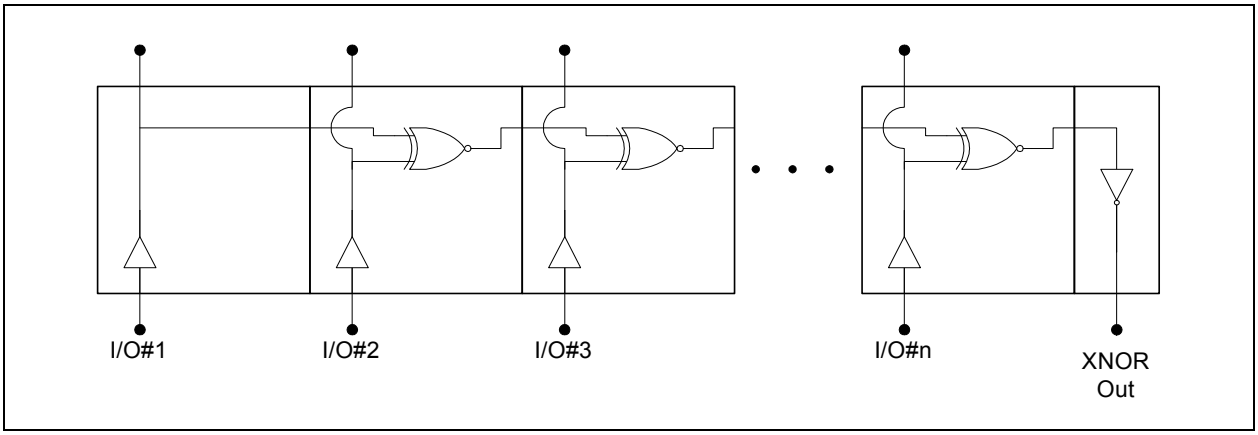
The tests that are performed when the XNOR Chain test mode is enabled require the board-level test hardware to control the device pins and observe the results at the XNOR Chain output pin; e.g., as described in Section 41.5.3, "Test Procedure," on page 482.

41.5.2 EXCLUDED PINS

All pins in the pinout are included in the XNOR chain, except the following:

- Power Pins (VTR, VTR_33_18, VBAT, VREF_CPU)
- Ground Pins (VSS, AVSS, VSS_VBAT)
- Voltage Regulator Capacitor (VR_CAP)
- Crystal pins (XTAL1, XTAL2)
- Test Output Pin (XNOR_OUT): GPIO027/KSO00/PVT_IO1
- Pins (ICSP_MCLR)

FIGURE 41-2: XNOR CHAIN TEST STRUCTURE



41.5.3 TEST PROCEDURE

41.5.3.1 Setup

Warning: Ensure power supply is off during Setup.

1. Connect ICSP_MCLR to ground.
2. Connect the VSS, AVSS, VSS_VBAT pins to ground.
3. Connect the VTR, VTR_33_18, VBAT pins to an unpowered 3.3V power source.
4. Connect the VREF_CPU pin to an unpowered 1.8V power source.
5. Connect an oscilloscope or voltmeter to the Test Output pin.
6. All other pins should be tied to ground.

Note:	There are 107 pins in the XNOR Chain in the 128-pin package.

41.5.3.2 Testing

1. Turn on the 3.3V power source.
2. Enable the XNOR Chain as defined in Section 38.5.3.3, "Procedure to Enable the XNOR Chain".

Note:	Note that at this point all inputs to the XNOR Chain are low, except for the ICSP_MCLR pin, and the output on the Test Output pin is non-inverted from its initial state, which is dependent on the number of pins in the chain. If the number of input pins in the chain is an even number, the initial state of the Test Output Pin (XNOR_OUT): GPIO027/KSO00/PVT_IO1 is low. If the number of input pins in the chain is an odd number, the initial state of the Test Output Pin (XNOR_OUT): GPIO027/KSO00/PVT_IO1 is high.

3. Bring one pin in the chain high. The output on the Test Output Pin (XNOR_OUT): GPIO027/KSO00/PVT_IO1 pin should toggle. Then individually toggle each of the remaining pins in the chain. Each time an input pin is toggled either high or low the Test Output Pin (XNOR_OUT): GPIO027/KSO00/PVT_IO1 pin should toggle.
4. Once the XNOR test is completed, exit the XNOR Chain Test Mode by cycling VTR power.

41.5.3.3 Procedure to Enable the XNOR Chain

Note:	The GPIO145(ICSP_CLOCK) pin is used as a clock in this test mode. This pin must never be toggled at a rate greater than 20Mhz.

```
//BEGIN PROCEDURE TO ENTER XNOR CHAIN
////////////////////////////////////
// Initialize IF
////////////////////////////////////
    force ICSP_MCLR = 1;
    force GPIO145(ICSP_CLOCK) = 0; //TCLk
    force GPIO146(ICSP_DATA) = 0; //TDI
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 1; //TMS
Wait 100 ns

////////////////////////////////////
// ICSP Reset
////////////////////////////////////
    force ICSP_MCLR = 1;
Wait 1000 ns;
    force ICSP_MCLR = 0;

////////////////////////////////////
// ICSP Bypass
////////////////////////////////////
    force ICSP_MCLR = 0;
    force GPIO146(ICSP_DATA) = 0; //TDI
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 1; //TMS

    repeat (40)
      begin
        force ICSP_MCLR = 1;
        force ICSP_MCLR = 0;
      end

Wait 1000 ns

////////////////////////////////////
// Come out of reset
////////////////////////////////////
```

```
    force ICSP_MCLR = 1;    //P 1
    force ICSP_MCLR = 0;
    force ICSP_MCLR = 1;    //P 2
    force ICSP_MCLR = 0;
Wait 100 ns

////////////////////////////////////
// Write IR with 0xD
////////////////////////////////////
    force ICSP_MCLR = 1; //P 3 (TEST_LOGIC_RESET)
    force ICSP_MCLR = 0; //1N
    force GPIO146(ICSP_DATA) = 0; //TDI
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS

    force ICSP_MCLR = 1; //P 4 (RUN_TEST_IDLE)
    force ICSP_MCLR = 0; //2N
    force GPIO146(ICSP_DATA) = 0; //TDI
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 1; //TMS

    force ICSP_MCLR = 1; //P 5 (SEL_DR)
    force ICSP_MCLR = 0; //3N
    force GPIO146(ICSP_DATA) = 0; //TDI
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 1; //TMS

    force ICSP_MCLR = 1; //P 6 (SEL_IR)
    force ICSP_MCLR = 0; //4N
    force GPIO146(ICSP_DATA) = 0; //TDI
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS

    force ICSP_MCLR = 1; //P 7 (CAP_IR)
    force ICSP_MCLR = 0; //5N
    force GPIO146(ICSP_DATA) = 0; //TDI
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS

////////////////////////////////////
//SHIFT IR 0xD
////////////////////////////////////
    force ICSP_MCLR = 1; //P 8 (SHIFT_IR)
    force ICSP_MCLR = 0; //6N
    force GPIO146(ICSP_DATA) = 1; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 9
    force ICSP_MCLR = 0; //7N
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
```

```
    force ICSP_MCLR= 1; //P 10
    force ICSP_MCLR = 0; //8N
    force GPIO146(ICSP_DATA) = 1; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 11
    force ICSP_MCLR = 0; //9N
    force GPIO146(ICSP_DATA) = 1; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 1; //TMS //Next will be EXIT1_IR

    force ICSP_MCLR = 1; //P 12 (EXIT1_IR)
    force ICSP_MCLR 0; //10N
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 1; //TMS //Next will be UPDATE_IR

    force ICSP_MCLR = 1; //P 13 (UPDATE_IR)
    force ICSP_MCLR = 0; //11N
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS //Next will be IDLE

    force ICSP_MCLR = 1; //P 14 (IDLE)
    force ICSP_MCLR = 0; //12N
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS //Next will be IDLE
Wait 1000 ns

////////////////////////////////////
// DIR=0, CMD[2:0]=1, DATA[7:0]=01\h, ADDR[7:0]=88\h
////////////////////////////////////
    force ICSP_MCLR = 1; //P 15
    force ICSP_MCLR = 0; //1N
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 1; //TMS
    force ICSP_MCLR = 1; //P 16
    force ICSP_MCLR = 0; //2N
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 17
    force ICSP_MCLR = 0; //3N
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS

////////////////////////////////////
//DIR 0 - Write
////////////////////////////////////
    force ICSP_MCLR = 1; //P 18
```

```
    force ICSP_MCLR = 0; //N (DR1)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS

////////////////////////////////////
//CMD 1 - Test
////////////////////////////////////
    force ICSP_MCLR = 1; //P 19
**Verify JTAG_TDO = 1
    force ICSP_MCLR = 0;//N (DR2)
    force GPIO146(ICSP_DATA) = 1; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force `ICSP_MCLR = 1; //P 20
**Verify JTAG_TDO = 1
    force ICSP_MCLR = 0; //N (DR3)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 21
**Verify JTAG_TDO = 1
    force ICSP_MCLR = 0; //N (DR4)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS

////////////////////////////////////
//DATA 0x01 - XNOR_EN
////////////////////////////////////
    force ICSP_MCLR = 1; //P 22
**Verify JTAG_TDO = 1
    force ICSP_MCLR = 0; //N (DR5)
    force GPIO146(ICSP_DATA) = 1; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P23
**Verify JTAG_TDO = 1
    force ICSP_MCLR = 0; //N (DR6)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 24
**Verify JTAG_TDO = 0
    force ICSP_MCLR = 0; //N (DR7)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 25
**Verify JTAG_TDO = 0
    force ICSP_MCLR = 0; //N (DR8)
```

```
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 26
**Verify JTAG_TDO = 0
    force ICSP_MCLR = 0; //N (DR9)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 27
**Verify JTAG_TDO = 1
    force ICSP_MCLR = 0; //N (DR10)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force `ICSP_MCLR = 1; //P 28
**Verify JTAG_TDO = 0
    force ICSP_MCLR = 0; //N (DR11)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 29
**Verify JTAG_TDO = 0
    force ICSP_MCLR = 0; //N (DR12)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS

////////////////////////////////////
//ADDRESS 0x88 - Customer Control
////////////////////////////////////
    force ICSP_MCLR = 1; //P 30
**Verify JTAG_TDO = 0
    force ICSP_MCLR = 0; //N (DR13)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 31
**Verify JTAG_TDO = 1
    force ICSP_MCLR = 0; //N (DR14)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 32
**Verify JTAG_TDO = 0
    force ICSP_MCLR = 0; //N (DR15)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 33
**Verify JTAG_TDO = 0
    force ICSP_MCLR = 0; //N (DR16)
```

```
    force GPIO146(ICSP_DATA) = 1; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 34
**Verify JTAG_TDO = 0
    force ICSP_MCLR = 0; //N (DR17)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 35
**Verify JTAG_TDO = 1
    force ICSP_MCLR = 0; //N (DR18)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 36
**Verify JTAG_TDO = 0
    force ICSP_MCLR = 0; //N (DR19)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 37
**Verify JTAG_TDO = 0
    force ICSP_MCLR = 0; //N (DR20)
    force GPIO146(ICSP_DATA) = 1; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 1; //TMS
    force ICSP_MCLR= 1; //P 38
**Verify JTAG_TDO = 0
    force ICSP_MCLR = 0; //N (E1_DR)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 1; //TMS
    force ICSP_MCLR = 1; //P 39
    force ICSP_MCLR = 0; //N (UP_DR)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 40
    force ICSP_MCLR = 0; //N (EXTRA CLK)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
    force ICSP_MCLR = 1; //P 41
    force ICSP_MCLR= 0; //N (EXTRA CLK)
    force GPIO146(ICSP_DATA) = 0; //TDI;
    force GPIO130/SMB03_DATA/SMB03_DATA18 = 0; //TMS
Wait 1000 ns

////////////////////////////////////////////////////////////////////////////
//FINISHED PROCEDURE TO ENTER XNOR
////////////////////////////////////////////////////////////////////////////
```

42.0 ELECTRICAL SPECIFICATIONS

42.1 Maximum Ratings*

*Stresses exceeding those listed could cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other condition above those indicated in the operation sections of this specification is not implied.

Note:	When powering this device from laboratory or system power supplies, it is important that the Absolute Maximum Ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes on their outputs when the AC power is switched on or off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists, it is suggested that a clamp circuit be used.

42.1.1 ABSOLUTE MAXIMUM THERMAL RATINGS

TABLE 42-1: ABSOLUTE MAXIMUM THERMAL RATINGS

Parameter	Maximum Limits
Operating Temperature Range	0^{o}C to +70^{o}C Commercial -40^{o}C to +85^{o}C Industrial
Storage Temperature Range	-55^{o} to +150^{o}C
Lead Temperature Range	Refer to JEDEC Spec J-STD-020B

42.1.2 ABSOLUTE MAXIMUM SUPPLY VOLTAGE RATINGS

TABLE 42-2: ABSOLUTE POWER SUPPLY RATINGS

Symbol	Parameter	Maximum Limits
VBAT	3.0V Battery Backup Power Supply with respect to ground	-0.3V to +3.63V
VTR	3.3V Suspend Power Supply with respect to ground	-0.3V to +3.465V
VTR_33_18	3.3V or 1.8V Power Supply with respect to ground	-0.3V to +3.465V
VCC	3.3V Main Power Supply with respect to ground (Connected to VCC_PWRGD pin)	-0.3V to +3.465V

42.1.3 ABSOLUTE MAXIMUM I/O VOLTAGE RATINGS

Parameter	Maximum Limits
Voltage with respect to ground on any pin without backdrive protection	-0.3V to (Power Supply used to power the buffer) + 0.3V (Note 42-1)

Note 42-1 The Power Supply used to power the buffer is shown in the Signal Power Well column of the Pin Multiplexing Tables in **Section 2.0 "Pin Configuration"**.

42.2 Operational Specifications

42.2.1 POWER SUPPLY OPERATIONAL CHARACTERISTICS

TABLE 42-3: POWER SUPPLY OPERATING CONDITIONS

Symbol	Parameter	MIN	TYP	MAX	Units
VBAT	Battery Backup Power Supply	2.0	3.0	3.6	V
VTR	Suspend Power Supply	3.135	3.3	3.465	V
VTR_33_18	3.3V Power Supply	3.135	3.3	3.465	V
	1.8V Power Supply	1.71	1.80	1.89	V

Note: The specification for the VTR & VTR_33_18 supplies are +/- 5%.

42.2.2 AC ELECTRICAL SPECIFICATIONS

The AC Electrical Specifications for the clock input time are defined in Section 43.2, "Clocking AC Timing Characteristics," on page 504. The clock rise and fall times use the standard input thresholds of 0.8V and 2.0V unless otherwise specified and the capacitive values listed in Section 42.2.2, "AC Electrical Specifications," on page 490.

42.2.3 CAPACITIVE LOADING SPECIFICATIONS

The following table defines the maximum capacitive load validated for the buffer characteristics listed in Table 42-5, “DC Electrical Characteristics,” on page 491.

CAPACITANCE T_A = 25°C; fc = 1MHz; V_{cc} = 3.3 VDC

Note: All output pins, except pin under test, tied to AC ground.

TABLE 42-4: MAXIMUM CAPACITIVE LOADING

Parameter	Symbol	Limits			Unit	Notes
		MIN	TYP	MAX		
Input Capacitance of PCI_I and PCI_IO pins	C_{IN}			Note 42-2	pF	
Input Capacitance of PCI_CLK pin	C_{IN}			Note 42-2	pF	
Output Load Capacitance supported by PCI_IO, PCI_O, and PCI_OD	C_{OUT}			Note 42-2	pF	
SUSCLK Input Capacitance	C_{IN}			10	pF	
Input Capacitance of PECI_I and PECI_IO	C_{IN}			10	pF	
Output Load Capacitance supported by PECI_IO and OD_PH	C_{OUT}			10	pF	

TABLE 42-4: MAXIMUM CAPACITIVE LOADING (CONTINUED)

Parameter	Symbol	Limits			Unit	Notes
		MIN	TYP	MAX		
Input Capacitance (SPI pins)	C_{IN}			6	pF	Note 42-3
Output Capacitance (SPI pins)	C_{OUT}			8	pF	Note 42-3
Input Capacitance (all other input pins)	C_{IN}			10	pF	Note 42-4
Output Capacitance (all other output pins)	C_{OUT}			20	pF	Note 42-5

Note 42-2 The PCI buffers are designed to meet the defined PCI Local Bus Specification, Rev. 2.1, electrical requirements.

Note 42-3 This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

Note 42-4 All input buffers can be characterized by this capacitance unless otherwise specified.

Note 42-5 All output buffers can be characterized by this capacitance unless otherwise specified.

42.2.4 DC ELECTRICAL CHARACTERISTICS FOR I/O BUFFERS

TABLE 42-5: DC ELECTRICAL CHARACTERISTICS

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
PIO Type Buffer						
All PIO Buffers						Internal PU/PD selected via the GPIO Pin Control Register.
Pull-up current	I_{PU}	39	84	162	µA	
Pull-down current	I_{PD}	39	65	105	µA	
I Type Input Buffer						TTL Compatible Schmitt Trigger Input
Low Input Level	V_{ILI}			0.3x VTR	V	
High Input Level	V_{IHI}	0.7x VTR			V	
Tolerance				3.63	V	This buffer is not 5V tolerant.
Schmitt Trigger Hysteresis	V_{HYS}		400		mV	
O-2 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 2 mA
High Output Level	V_{OH}	VTR-0.4			V	I_{OH} = -2 mA
Tolerance						This buffer is not 5V tolerant.

TABLE 42-5: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
IO-2 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an O-2mA.
OD-2 mA Type Buffer Low Output Level Tolerance	V_{OL}			0.4	V	V_{OL} = 2 mA This buffer is not 5V tolerant.
IOD-2 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an OD-2mA.
O-4 mA Type Buffer Low Output Level High Output Level Tolerance	V_{OL} V_{OH}	 VTR-0.4		0.4	V V	I_{OL} = 4 mA I_{OH} = -4 mA This buffer is not 5V tolerant.
IO-4 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an O-4mA.
OD-4 mA Type Buffer Low Output Level Tolerance	V_{OL}			0.4	V	V_{OL} = 4 mA This buffer is not 5V tolerant.
IOD-4 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an OD-4mA.
O-8 mA Type Buffer Low Output Level High Output Level Tolerance	V_{OL} V_{OH}	 VTR-0.4		0.4	V V	I_{OL} = 8 mA I_{OH} = -8 mA This buffer is not 5V tolerant.
IO-8 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an O-8mA.
OD-8 mA Type Buffer Low Output Level Tolerance	V_{OL}			0.4	V	V_{OL} = 8 mA This buffer is not 5V tolerant.
IOD-8 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an OD-8mA.

TABLE 42-5: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
O-12 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 12mA
High Output Level	V_{OH}	VTR-0.4			V	I_{OH} = -12mA
Tolerance						This buffer is not 5V tolerant.
IO-12 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an O-12mA.
OD-12 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	IOL = 12mA
Tolerance						This buffer is not 5V tolerant.
IOD-12 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an OD-12mA.
ILLK Low Leakage Input Buffer						TTL Levels
Low Input Level	V_{ILI}			0.8	V	
High Input Level	V_{IH}	2.0			V	
Input Leakage	I_{IL}	-500		+500	nA	V_{IN} = 0n, VBAT = 3.0 VDC & VTR = 0 VDC
I_AN Type Buffer						
I_AN Type Buffer (Analog Input Buffer)	I_AN					Voltage range on pins: -0.3V to +3.63V These buffers are not 5V tolerant buffers and they are not backdrive protected
PCI_PIO Type Buffer						
All PCI_PIO Buffers						Internal PU is selected via the GPIO Pin Control Register.
Pull-up current	I_{PU}	0.6	1	1.5	mA	
PCI_CLK Type Buffer	PCI_ICLK					See *PCI Local Bus Specification Rev. 2.1*
PCI_IO Type Buffers	PCI_IO PCI_O PCI_I					
PCI_OD Type Buffer	PCI_OD					These buffers are not not 5V tolerant buffers and they are not backdrive protected.

TABLE 42-5: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
PECI Type Buffer						
VREF_CPU						Connects to CPU Voltage pin (Processor dependent)
PECI Bus Voltage	V_{BUS}	0.95		1.26	V	
SBTSI Bus Voltage	V_{BUS}	1.28		1.9	V	
Input current	IDC			100	µA	
Input Low Current	ILEAK	-10		+10	µA	This buffer is not 5V tolerant This buffer is not backdrive protected.
PECI_I Buffer						All input and output voltages are a function of Vref, which is connected to CPU_VREF input.
Input voltage range	VIn	-0.3		+Vref 0.3	V	
Low Input Level	VIL			0.275× Vref	V	
High Input Level	VIH	0.725× Vref			V	This buffer is not 5V tolerant This buffer is not backdrive protected.
PECI_IO						All input and output voltages are a function of Vref, which is connected to CPU_VREF input.
Input voltage range	VIn	-0.3		+Vref 0.3	V	See PECI Specification.
Hysteresis	VHYS	0.1 ×Vref	0.2× Vref		V	
Low Input Level	VIL			0.275× Vref	V	
High Input Level	VIH	0.725× Vref			V	
Low Output Level	VOL			0.25× Vref	V	0.5mA < IOL < 1mA
High Output Level	VOH	0.75 ×Vref			V	IOH = -6mA
Tolerance				3.63	V	This buffer is not 5V tolerant This buffer is not backdrive protected.

TABLE 42-5: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
Crystal oscillator						
XTAL1 (OCLK)	The MEC140x/1x crystal oscillator design requires a 32.768 KHz parallel resonant crystal with load caps in the range 4-18pF. Refer to “Application Note PCB Layout Guide for MEC140x/1x” for more information.					
XTAL2 (ICLK)						
Low Input Level	V_{ILI}			0.4	V	
High Input Level	V_{ILH}	2.0			V	VIN = 0 to VTR
ADC, DAC, and Comparator Reference Pins						
ADC_VREF						
Voltage (Option A)	V		VTR		V	connect to same power supply as VTR
Voltage (Option B)	V	2.97	3.0	3.03	V	
Input Impedance	R_{REF}		75		kΩ	
Input Low Current	ILEAK	-0.05		+0.05	µA	This buffer is not 5V tolerant This buffer is not backdrive protected.
DAC_VREF						
Voltage	V	0.5		VTR	V	
Input Impedance	R_{REF}	100			kΩ	
Input Low Current	ILEAK	-10		+10	µA	This buffer is not 5V tolerant This buffer is not backdrive protected.
CMP_VREF						
Voltage	V	0	-	VTR	V	
Input Current (comparator enabled)	I_{DC}			30	µA	
Input Low Current	ILEAK	-0.05		+0.05	µA	This buffer is not 5V tolerant This buffer is not backdrive protected.

42.2.4.1 Max Voltage Tolerance

All the functional pins are 3.63V tolerant, except for the 1.8V I/O signals defined in Section 2.6, "1.8V or 3.3V I/O Pins," on page 21. The 1.8V I/O signals can only tolerate up to +/-10% I/O operation (or +1.98V max)

42.2.4.2 Pin Leakage

Leakage characteristics for all pins, except for the battery powered pins with the ILLK buffer type, is shown in the following table:

TABLE 42-6: PIN LEAKAGE

(T_A = 0°C to +85°C)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
Leakage Current	I_{IL}			+/-2	µA	VIN=0V to VTR

42.2.4.3 Backdrive Protection

All signal pins are Backdrive Protected except those listed in the Pin Configuration chapter as non-backdrive protected.

TABLE 42-7: BACKDRIVE PROTECTION

(T_A = 0°C to +85°C)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
Input Leakage	I_{IL}	-2		+2	µA	VIN=3.47V@VTR=0V

42.2.5 ADC ELECTRICAL CHARACTERISTICS

TABLE 42-8: ADC CHARACTERISTICS

Parameter	MIN	TYP	MAX	Units
Analog Supply Voltage (powered by VTR)	3.135	3.3	3.465	V
Resolution	–	–	10	Bits
Accuracy	–	2	4	LSB
Differential Non Linearity, DNL	-1	–	+1	LSB
Integral Non Linearity, INL	-1.5	–	+1.5	LSB
Gain Error, E_{GAIN}	-2	–	2	LSB
Offset Error, E_{OFFSET}	-2	–	2	LSB
Conversion Time	–	–	12	µs/channel
Input Impedance	4	4.5	5.3	MΩ
V_{REF} Input Impedance	630	750	830	kΩ

42.2.6 DAC ELECTRICAL CHARACTERISTICS

TABLE 42-9: DC CHARACTERISTICS: DAC

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise noted)				
Symbol	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Comments**
V_{RNG}	Output Voltage Range for Guaranteed Specifications	10%	–	90%	V	Range of (input-AVSS)
RES	Resolution	12	–	–	Bits	Guaranteed Monotonic
ACC	Accuracy	–	2	4	LSB	
INL	Integral Nonlinearity	–	±2	±4	LSB	Guaranteed Monotonic
DNL	Differential Nonlinearity	-1	±1	<+2	LSB	Guaranteed Monotonic
E_{OFF}	Offset Error	-5	20	36	mV	
E_G	Gain Error	-2	0	+2	% of FS	
R_{IN}	V_{REF} Input Impedance	–	130	–	kΩ	For each DAC
		–	65	–	kΩ	With both DACs enabled
DAC Output Buffer						
RL	Resistive Output Load impedance	2	–	–	kΩ	Minimum external load resistance
CL	Output Load capacitance	–	–	50	pF	Maximum external load capacitance
I_{OUT}	Output current drive strength	1.5	–	–	mA	Sink and Source
R_{OUT}	Output impedance	3.34	3.60	4.01	Ω	DAC0
		2.67	2.90	3.34	Ω	DAC1

TABLE 42-10: AC CHARACTERISTICS: DAC

AC CHARACTERISTICS		Standard Operating Conditions (unless otherwise noted)				
Symbol	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Comments**
F_S	Sampling Frequency	—	—	1	MSPS	Frequency for a small variation of input (1 LSB)
T_{ST}	Settling Time	—	1	2	µS	

42.2.7 COMPARATOR ELECTRICAL CHARACTERISTICS

TABLE 42-11: AC AND DC CHARACTERISTICS: COMPARATOR

CHARACTERISTICS		Standard Operating Conditions (unless otherwise noted)				
Symbol	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Comments**
V_{IN}	Input Voltage Range	0	-	VTR	V	
V_{HYST}	Input Hysteresis Voltage	15	30	45	mV	
CMRR	Common mode rejection ratio	44	-	-	dB	
T_{RESP}	Large signal response time	—	100	160	ns	VCM = VDD/2 100 mV step
TS_{RESP}	Small signal response time	—	160	320	ns	VCM = VDD/2 100 mV step
T_{ON}	Comparator Enable to Valid Output	—	—	0.1	µS	Note 42-6

Note 42-6 To prevent getting glitches on the comparator output, it is recommended to enable the comparator and wait for the output to be valid and stable before configuring the pin for the CMP_VOUTx function.

42.2.8 THERMAL CHARACTERISTICS

TABLE 42-12: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min.	Typical	Max.	Unit
Consumer Temperature Devices					
Operating Junction Temperature Range	TJ	—	—	+125 (Note 1)	°C
Operating Ambient Temperature Range - Commercial	TA	0	—	+70	°C
Operating Ambient Temperature Range - Industrial	TA	-40	—	+85	°C
Power Dissipation: Internal Chip Power Dissipation: PINT = VTR x IVTR from Table 42-14 and Table 42-15 (e.g., 3.45V x 12.50mA = 43mW)	PD	PINT + PI/O			W
I/O Pin Power Dissipation: I/O = S (({VTR– VOH} x IOH) + S (VOL x IOL))					
Maximum Allowed Power Dissipation	PDMAX	(TJ – TA)/θJA			W
Note 1: TJ Max value is at ambient of 70°C					

TABLE 42-13: THERMAL PACKAGING CHARACTERISTICS

Characteristics	Symbol	Typical	Max.	Unit	Notes
Package Thermal Resistance, 128-pin VTQFP	θJA	51.0	—	°C/W	1
	θJC	25.0	—	°C/W	1
Package Thermal Resistance, 144-pin WFBGA	θJA	50.0	—	°C/W	1
	θJC	17.0	—	°C/W	1
Note 1: Junction to ambient thermal resistance, Theta-JA (θJA) and Junction to case thermal resistance, Theta-JC (θJC) numbers are achieved by package simulations.					

42.3 Power Consumption

TABLE 42-14: VTR SUPPLY CURRENT, I_VTR

VCC	VTR	LPC Clock	48 MHz Ring Oscillator Frequency	Typical (3.3V, 25° C)	Max (3.45V, 70° C)	Max (3.45V, 85° C)	Units	Comments
On	On	On	48MHz	10.75	12.50	13.75	mA	FULL ON
On	On	On	12MHz	5.50	6.75	7.50	mA	FULL ON
On	On	On	3MHz	3.55	4.75	5.25	mA	FULL ON
On	On	On	1MHz	3.25	4.50	5.00	mA	FULL ON
On	On	On	12MHz	2.85	4.00	4.50	mA	Heavy Sleep 1
On	On	On	Off	1.05	2.25	2.50	mA	Heavy Sleep 2
On	On	On	Off	0.75	1.95	2.15	mA	Heavy Sleep 3
Off	On	Off	48MHz	9.75	11.25	12.50	mA	FULL ON
Off	On	Off	12MHz	4.50	6.00	6.60	mA	FULL ON
Off	On	Off	3MHz	2.65	4.00	4.40	mA	FULL ON
Off	On	Off	1MHz	2.35	3.75	4.15	mA	FULL ON
Off	On	Off	48MHz	9.50	10.75	11.85	mA	EC running, all peripherals Off
Off	On	Off	12MHz	4.25	5.50	6.00	mA	EC running, all peripherals Off
Off	On	Off	3MHz	2.50	3.65	4.00	mA	EC running, all peripherals Off
Off	On	Off	1MHz	2.25	3.50	3.75	mA	EC running, all peripherals Off
Off	On	Off	12MHz	2.00	3.25	3.60	mA	Heavy Sleep 1
Off	On	Off	Off	0.80	2.00	2.25	mA	Heavy Sleep 2
Off	On	Off	Off	0.52	1.70	1.90	mA	Heavy Sleep 3
Off	On	Off	Off	0.47	1.65	1.85	mA	Deepest Sleep

Note 1: FULL ON is defined as follows: The processor is not sleeping, the Core regulator and the Ring Oscillator remain powered, and at least one block is not sleeping.

2: The sleep states are defined in the System Sleep Control Register in the Power, Clocks and Resets Chapter. See Table 3-12, “System Sleep Control Bit Encoding,” on page 82.

3: In order to achieve the lowest leakage current when both PECI and SB TSI are not used, set the VREF_CPU Disable bit to 1. This bit is defined in Section 34.8.5, VREF_CPU DISABLE.

4: All values are taken with DAC 0, Comparator 0, DAC 1, Comparator 1, ADC all disabled. See Table 42-15 for additional I_{VTR} with these blocks enabled.

TABLE 42-15: ADDITIONAL VTR SUPPLY CURRENT WITH ANALOG BLOCKS ENABLED

VCC	VTR	LPC Clock	48 MHz Ring Oscillator Frequency	Typical (3.3V, 25° C)	Max (3.45V, 70° C)	Max (3.45V, 85° C)	Units	Comments
Off	On	Off	48MHz	0.90	1.00	1.10	mA	Additional I_{VTR} with DAC 0 enabled
Off	On	Off	48MHz	1.00	1.10	1.20	mA	Additional I_{VTR} with DAC 0 & Comparator 0 enabled
Off	On	Off	48MHz	0.90	1.00	1.10	mA	Additional I_{VTR} with DAC 1 enabled
Off	On	Off	48MHz	1.00	1.10	1.20	mA	Additional I_{VTR} with DAC 1 & Comparator 1 enabled
Off	On	Off	48MHz	0.60	0.70	0.80	mA	Additional I_{VTR} with ADC enabled
Note 1: The values in this table are added to the values in Table 42-14 excluding the sleep states.								

TABLE 42-16: VBAT SUPPLY CURRENT, I_VBAT (VBAT=3.0V)

VCC	VTR	LPC Clock	48 MHz Ring Oscillator Frequency	Typical (3.0V, 25° C)	Max (3.0V, 25° C)	Units	Comments
Off	Off	Off	Off	12.00	14.50	uA	Internal 32kHz oscillator
Off	Off	Off	Off	4.75	7.00	uA	32kHz crystal oscillator
Off	Off	Off	Off	4.00	6.50	uA	External 32kHz clock on XTAL2 pin

TABLE 42-17: VBAT SUPPLY CURRENT, I_VBAT (VBAT=3.3V)

VCC	VTR	LPC Clock	48 MHz Ring Oscillator Frequency	Typical (3.0V, 25° C)	Max (3.0V, 25° C)	Units	Comments
Off	Off	Off	Off	13.00	15.50	uA	Internal 32kHz oscillator
Off	Off	Off	Off	5.50	8.00	uA	32kHz crystal oscillator
Off	Off	Off	Off	4.75	7.50	uA	External 32kHz clock on XTAL2 pin

43.0 TIMING DIAGRAMS

Note: Timing values are preliminary and may change after characterization.

43.1 Voltage Thresholds and Power Good Timing

43.1.1 VTR THRESHOLD AND VTRGD TIMING

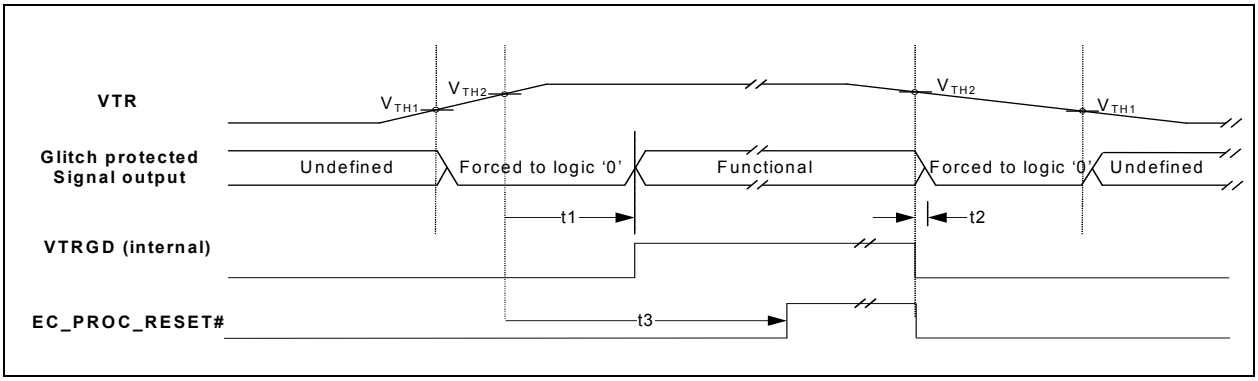


TABLE 43-1: VTR POWER GOOD TIMING

Parameters	Symbol	MIN	TYP	MAX	Unit	Notes
VTR Threshold for Pin Glitch Protection active	V_{TH1}	0.9	1	1.1	V	
VTR Power Good Threshold	V_{TH2}	2.16	2.4	2.64	V	
VTR Rise Time (Off to VTR =$V_{Threshold}$)	V_{Rise}	200			µs	
VTR Fall Time (VTR =$V_{Threshold}$) to Off	V_{Fall}	200			µs	
VTR > V_{TH2} to VTRGD (internal) asserted	t1		600		µs	
VTR < V_{TH2} to VTRGD (internal) deasserted	t2		100		ns	
VTR > V_{TH2} to EC_PROC_RESET# deasserted	t3		1		ms	Note:

Note: The Embedded Controller starts executing instructions when EC_PROC_ RESET deasserts.

43.1.2 VTR_33_18 TIMING

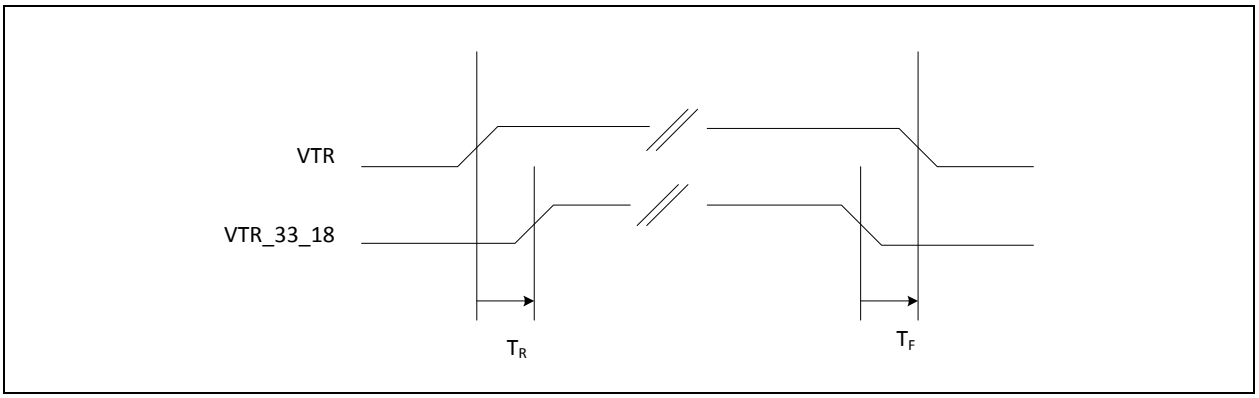


TABLE 43-2: VTR_33_18 TIMING

Parameters	Symbol	MIN	TYP	MAX	Unit	Notes
VTR On to VTR_33_18 On	T_R	0	-	-	us	
VTR_33_18 off to VTR Off	T_F	0	-	-	us	
VTR_33_18 Rise Time	V_{Rise}	10	-	-	us	
VTR_33_18 Fall Time	V_{Rise}	10	-	-	us	

Note: VTR_33_18 may be connected to the same VTR Power Rail or the system 1.8V power rail. If the VTR_33_18 power rail is connected to 1.8V rail this rail must be powered on only while the 3.3V VTR rail is present. The 1.8V rail must not be applied to the device when the 3.3V VTR rail is off.

43.1.3 VBAT THRESHOLDS AND VBAT_POR

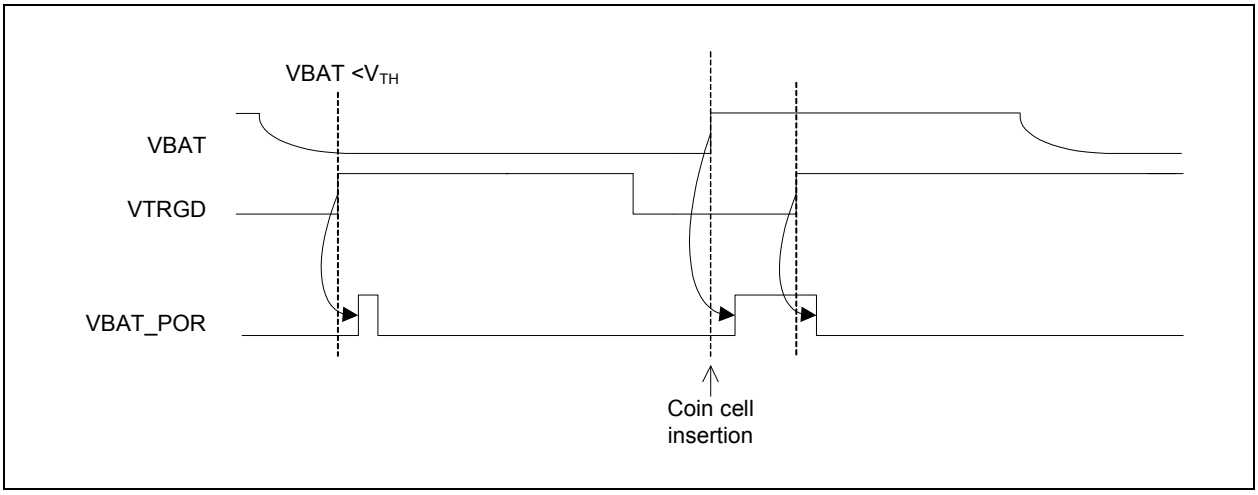


TABLE 43-3: VBAT THRESHOLDS AND VBAT_POR

Parameters	Symbol	MIN	TYP	MAX	Units	Notes
VBAT Power On Reset Threshold	V_{TH}	1.125	1.25	1.375	V	Note 43-1
VBAT Rise Time (Off to VBAT $=V_{Threshold}$)	V_{Rise}	0.100		30	ms	

Note 43-1 VBAT is monitored on two events: coin cell insertion and VTRGD assertion. If VBAT is below VTH when VTRGD is asserted a VBAT_POR is generated.

43.1.4 VCC_PWRGD TIMING

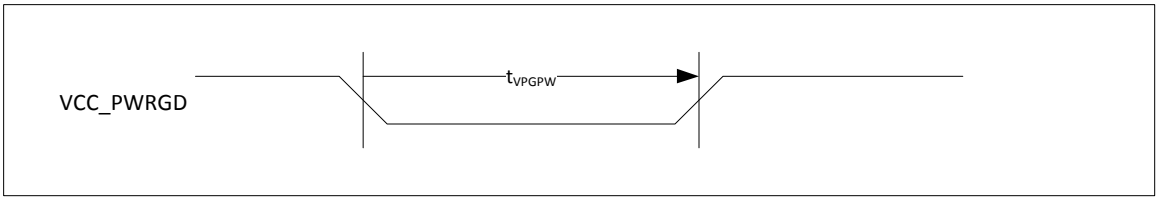


FIGURE 43-1: VCC_PWRGD TIMING DIAGRAM

TABLE 43-4: VCC_PWRGD INPUT TIMING

Parameters	Symbol	MIN	TYP	MAX	Units	Notes
VCC_PWRGD Pulse Width	t_{VPGPW}	31	–	–	ns	

43.2 Clocking AC Timing Characteristics

FIGURE 43-2: CLOCK TIMING DIAGRAM

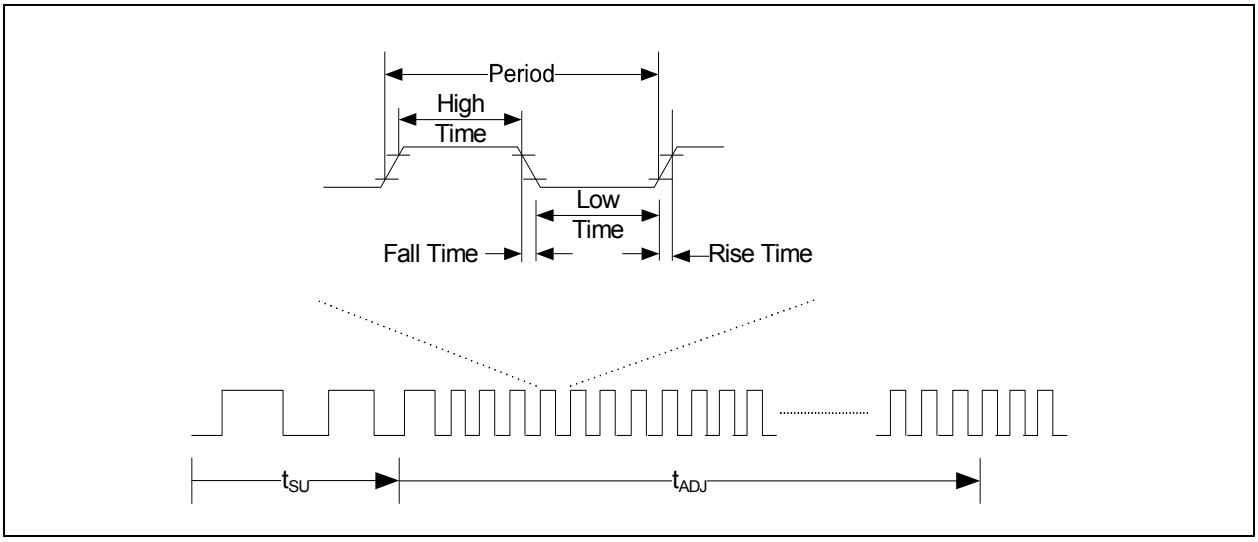


TABLE 43-5: CLOCK TIMING PARAMETERS

Clock	Parameters	Symbol	MIN	TYP	MAX	Units
48 MHz Ring Oscillator	Start-up accuracy (without 32 kHz present)	f_{SU}	22	-	53	MHz
	Start-up delay from 0 MHz to Start-up accuracy	t_{SU}	-	-	6	µs
	Operating Frequency (with external 32kHz clock source present after frequency lock to 48 MHz)	f_{OP}	47.04	48	48.95	MHz
	Operating Frequency (with internal 32 kHz oscillator present after frequency lock to 48 MHz) Note 43-3	f_{OP}	46.08	48	49.92	MHz
	Adjustment Delay from Start-up accuracy to Operating accuracy (time to attain frequency lock - with 32 kHz present)	t_{ADJ}	0.03	-	4 (Note 43-3)	ms
	Adjustment Delay when resuming from Heavy Sleep 3 and System Deepest Sleep state. (time to re-attain frequency lock - with external 32 kHz present)	t_{ADJ}	-	0.60	1 (Note 43-3)	ms
	Operating Frequency (with external 32 kHz removed after frequency locked to 48 MHz)	f_{OP}	43.2 (Note 43-5)	-	52.8 (Note 43-5)	MHz

TABLE 43-5: CLOCK TIMING PARAMETERS (CONTINUED)

Clock	Parameters	Symbol	MIN	TYP	MAX	Units
SUSCLK	Operating Frequency	-	-	32.768	-	kHz
	Period	-	(Note 43-4)	30.52	(Note 43-4)	µs
	High Time	-	10			us
	Low Time	-	10			us
	Fall Time	-	-	-	1	us
	Rise Time	-	-	-	1	us
32kHz_INT_OSC	Operating Frequency	-	32.112	32.768	33.424	kHz
	Start-up delay from 0k Hz to Operating Frequency	-	150			us

Note 43-2 Without the external clock, the 48MHz clock will vary up to +/-4% which may affect the timing parameters of certain blocks. In particular it may not be accurate enough to ensure that the UART will work, depending on the accuracy of the clock of the external device.

Note 43-3 This time applies if the external 32KHz clock input is available or if the Internal Oscillator clock source is enabled via the Clock Enable Register on page 440.

Note 43-4 SUSCLK is required to have an accuracy of +/- 100 ppm.

Note 43-5 The drift in frequency after frequency lock if the 32kHz clock is removed is determined by varying temperature while voltage is held constant.

Note 43-6 Firmware can use the INT_32K_RDY status bit listed in the Section 10.0, "Jump Table Vectored Interrupt Controller (JTVIC)" to indicate when the internal 32k Hz clock has reach operating frequency.

Note 43-7 This timing assumes the oscillator was powered and reach oscillator-lock prior to entering a sleep state. See also Table 3-12, “System Sleep Control Bit Encoding,” on page 82.

43.3 Generic Strap Option Sampling Timing

TABLE 43-6: GENERIC STRAP OPTION SAMPLING TIMING

Parameters	Symbol	MIN	TYP	MAX	Units	Notes
EC_PROC_RESET# deasserted to Strap Sampled	t_{SAMPLE}	–	500	600	µs	

43.4 Pin Reset Timing

43.4.1 RESET_IN# TIMING

Note: The GPIO pins will float after the RESET_IN# pin is asserted low in less than 9 usec (max).

FIGURE 43-3: RESET_IN# TIMING

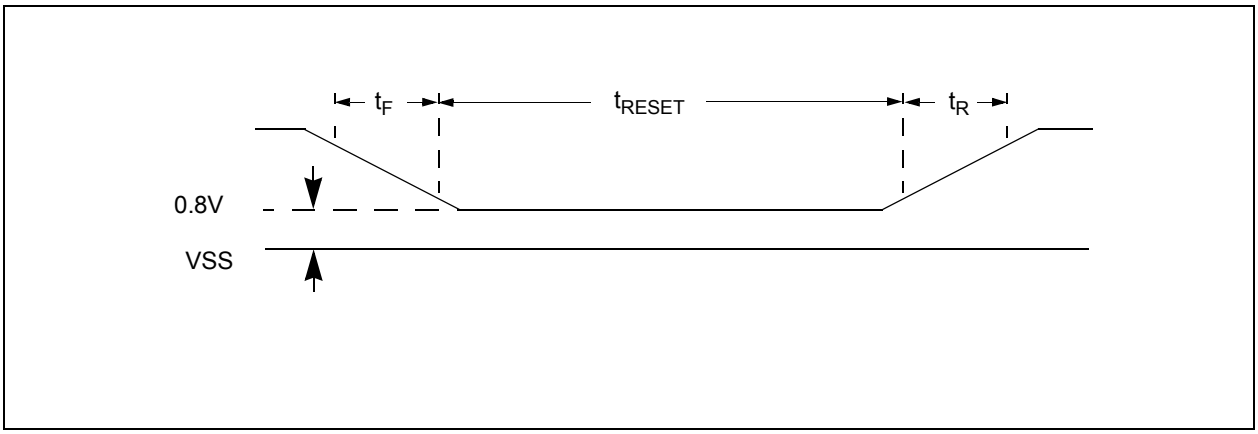


TABLE 43-7: RESETI# TIMING PARAMETERS

Symbol	Parameter	Limits		Units	Comments
		Min	Max		
t_F	RESET_IN# Fall time	0	10	µs	
t_R	RESET_IN# Rise time	0	10	µs	
t_{RESET}	Minimum Reset Time	1		µs	Note 43-8

Note 43-8 The RESET_IN# input can tolerate glitches of no more than 50ns.

43.4.2 RESET_OUT# TIMING

The minimum reset output time is determined by the RESET_IN# pin.

43.5 GPIO Timings

FIGURE 43-4: GPIO TIMING

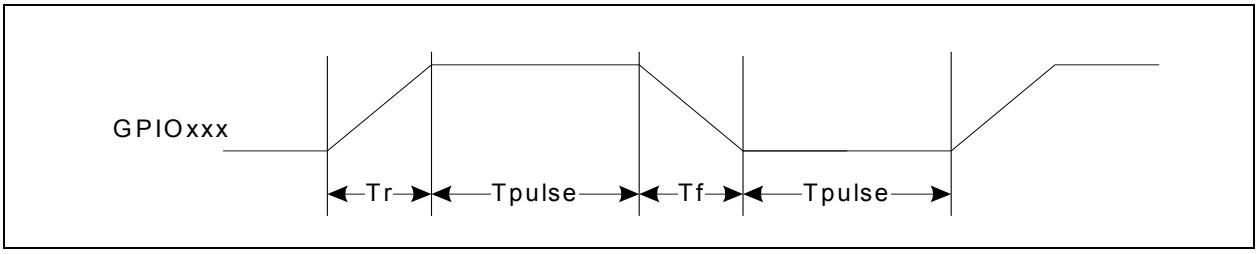


TABLE 43-8: GPIO TIMING PARAMETERS

Symbol	Parameter	MIN	TYP	MAX	Unit	Notes
t_R	GPIO Rise Time (push-pull)	0.54		1.31	ns	Pad type = IO2 C_L=2pF
t_F	GPIO Fall Time	0.52		1.27	ns	
t_R	GPIO Rise Time (push-pull)	0.58		1.46	ns	Pad type = IO4 C_L=5pF
t_F	GPIO Fall Time	0.62		1.48	ns	
t_R	GPIO Rise Time (push-pull)	0.80		2.00	ns	Pad type = IO8 C_L=10pF
t_F	GPIO Fall Time	0.80		1.96	ns	
t_R	GPIO Rise Time (push-pull)	1.02		2.46	ns	Pad type = IO12 C_L=20pF
t_F	GPIO Fall Time	1.07		2.51	ns	
t_{pulse}	GPIO Pulse Width	60			ns	

43.6 eSPI Timing

See Microchip “eSPI Controller” Specification.

43.7 A20M Timing

Same timing as GPIO signals. See Section 43.5, "GPIO Timings," on page 507.

43.8 8042 Emulation CPU_Reset Timing

FIGURE 43-5: CPU_RESET TIMING

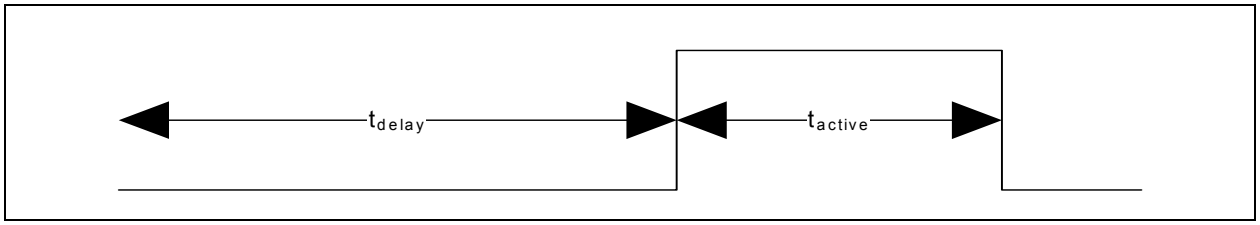


TABLE 43-9: CPU_RESET TIMING PARAMETERS

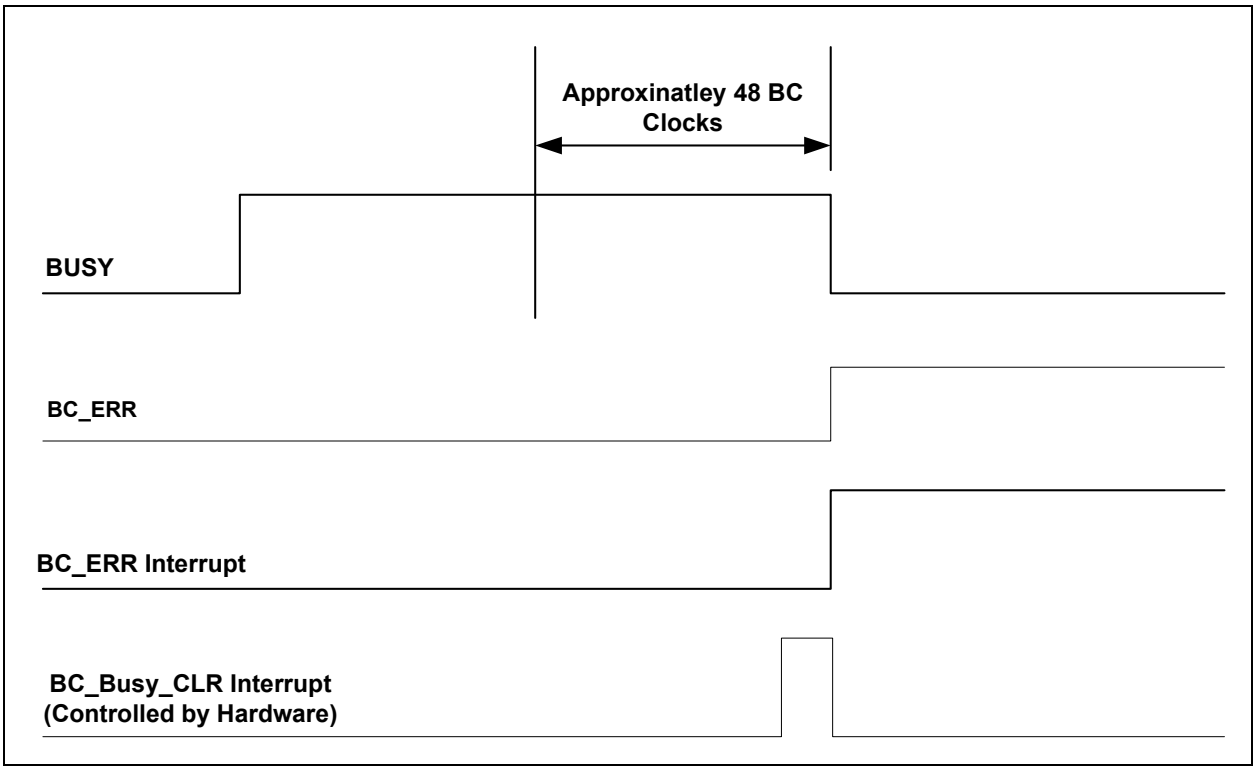
Name	Description	MIN	TYP	MAX	Units
t_{delay}	Delay prior to active pulse	14	15	15.5	µs
t_{active}	Active pulse width	6	8	8.5	µs

Note 43-1 FIGURE 43-5: and TABLE 43-9: refer to in which CPU_RESET is the inverse of ALT_RST# & KRESET.

Note 43-2 The KBRST pin function is the output of CPU_RESET described in Section 16.11.2, "CPU_RESET Hardware Speed-Up," on page 255.

43.9 BC-Link Master Interrupt Timing

FIGURE 43-6: BC-LINK ERR INTERRUPT TIMING



43.10 BC-Link Master Timing

FIGURE 43-7: BC-LINK READ TIMING

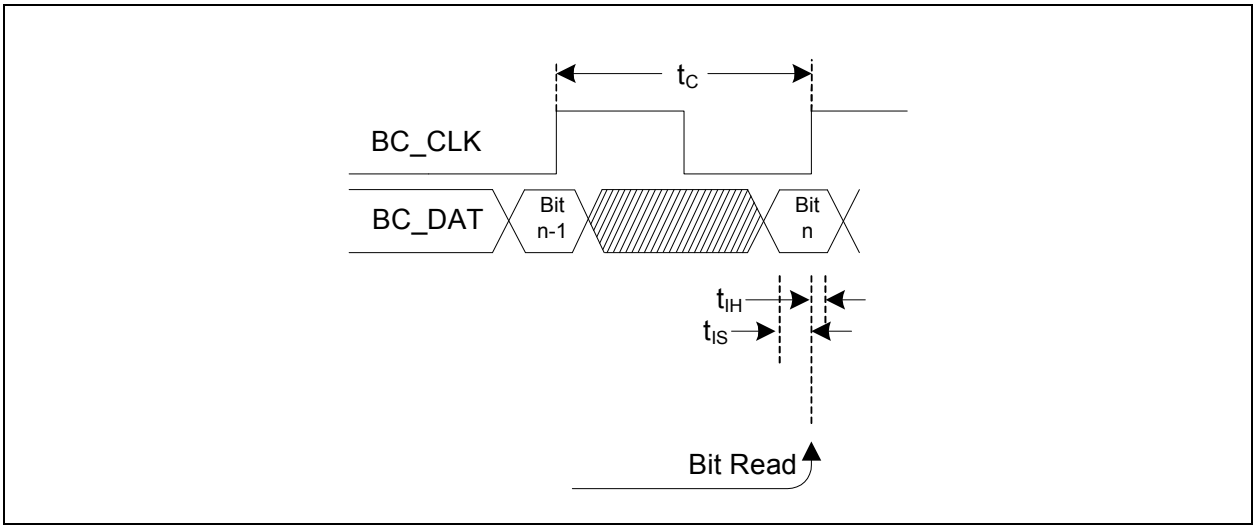


FIGURE 43-8: BC-LINK WRITE TIMING

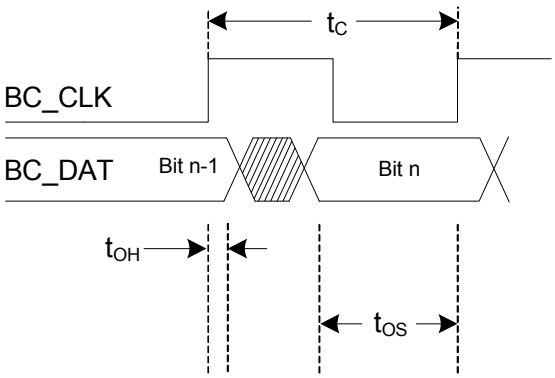


TABLE 43-10: BC-LINK MASTER TIMING DIAGRAM PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_c(High Speed)	High Spec BC Clock Frequency	23.5	24	24.5	MHz
	High Spec BC Clock Period	40.8	41.67	42.5	ns
t_{OS}	BC-Link Master DATA output setup time before rising edge of CLK.			t_c-t_{OH-MAX}	nsec
t_{OH}	BC-Link Master Data hold time after falling edge of CLK			10	nsec
t_{IS}	BC-Link Master DATA input setup time before rising edge of CLK.	15			nsec
t_{IH}	BC-Link Master DATA input hold time after rising edge of CLK.	0			nsec

Note 43-3 The (t_{IH} in TABLE 43-10:) BC-Link Master DATA input must be stable before next rising edge of CLK.

Note 43-4 The BC-Link Clock frequency is limited by the application usage model (see BC-Link Master Section 31.5, Signal Description). The BC-Link Clock frequency is controlled by the BC-Link Clock Select Register. The tc(High Speed) parameter implies both BC-link master and companion devices are located on the same circuit board and a high speed clock setting is possible.

Note: The timing budget equation is as follows for data from BC-Link slave to master:

Tc > TOD(master-clk) + Tprop(clk) +TOD(slave) + Tprop(slave data) + TIS(master).

43.11 Blinking/Breathing PWM Timing

FIGURE 43-9: BLINKING/BREATHING PWM OUTPUT TIMING

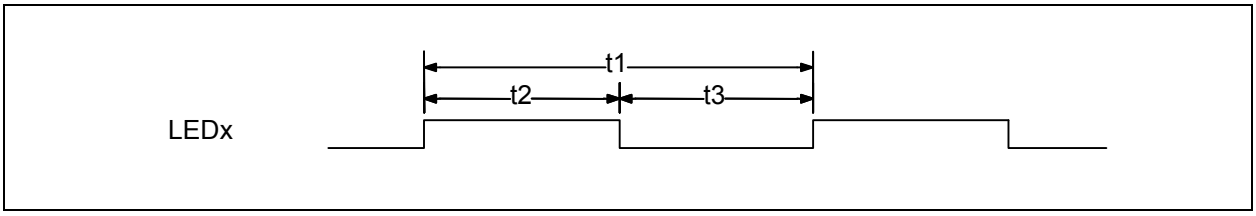


TABLE 43-11: BLINKING/BREATHING PWM TIMING PARAMETERS, BLINKING MODE

Name	Description	MIN	TYP	MAX	Units
t1	Period	7.8ms		32sec	
t_f	Frequency	0.03125		128	Hz
t2	High Time	0		16	sec
t3	Low Time	0		16	sec
t_d	Duty cycle	0		100	%

TABLE 43-12: BLINKING/BREATHING PWM TIMING PARAMETERS, GENERAL PURPOSE

Name	Description	MIN	TYP	MAX	Units
t1	Period	5.3us		21.8ms	
t_f	Frequency	45.8Hz		187.5kHz	
t2	High Time	0		10.9	ms
t3	Low Time	0		10.9	ms
t_d	Duty cycle	0		100	%

43.12 Fan Tachometer Timing

FIGURE 43-10: FAN TACHOMETER INPUT TIMING

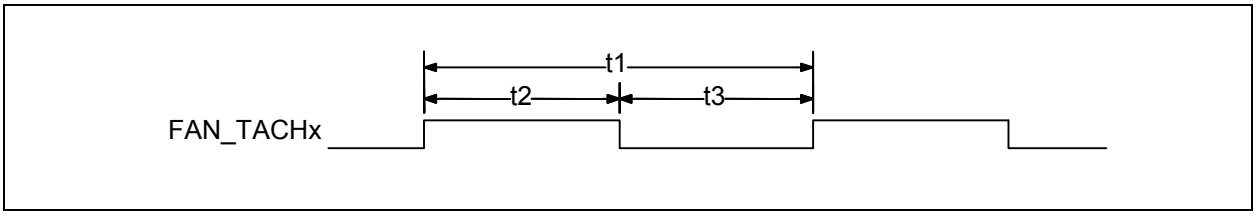


TABLE 43-13: FAN TACHOMETER INPUT TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Pulse Time	100			µsec
t2	Pulse High Time	20			
t3	Pulse Low Time	20			

Note 43-5 t_{TACH} is the clock used for the tachometer counter. It is 30.52 * prescaler, where the prescaler is programmed in the Fan Tachometer Timebase Prescaler register.

43.13 I2C/SMBus Timing

FIGURE 43-11: I2C/SMBUS TIMING

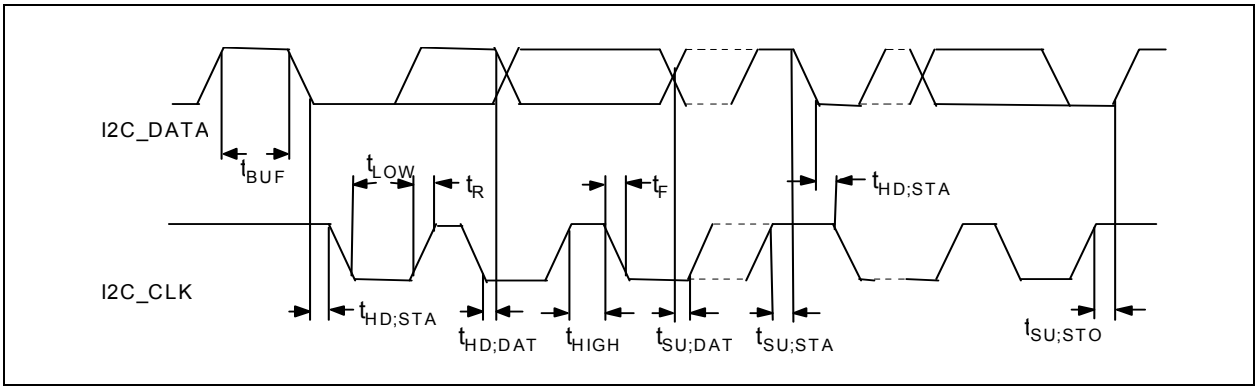


TABLE 43-14: I2C/SMBUS TIMING PARAMETERS

Symbol	Parameter	Standard-Mode		Fast-Mode		Fast-Mode Plus		Units
		MIN	MAX	MIN	MAX	MIN	MAX	
f_{SCL}	SCL Clock Frequency		100		400		1000	kHz
t_{BUF}	Bus Free Time	4.7		1.3		0.5		µs
$t_{SU;STA}$	START Condition Set-Up Time	4.7		0.6		0.26		µs
$t_{HD;STA}$	START Condition Hold Time	4.0		0.6		0.26		µs
t_{LOW}	SCL LOW Time	4.7		1.3		0.5		µs
t_{HIGH}	SCL HIGH Time	4.0		0.6		0.26		µs
t_R	SCL and SDA Rise Time		1.0		0.3		0.12	µs
t_F	SCL and SDA Fall Time		0.3		0.3		0.12	µs
$t_{SU;DAT}$	Data Set-Up Time	0.25		0.1		0.05		µs
$t_{HD;DAT}$	Data Hold Time	0		0		0		µs
$t_{SU;STO}$	STOP Condition Set-Up Time	4.0		0.6		0.26		µs

43.14 ICSP Interface Timing

FIGURE 43-12: ICSP POWER-UP & ASYNCHRONOUS RESET TIMING

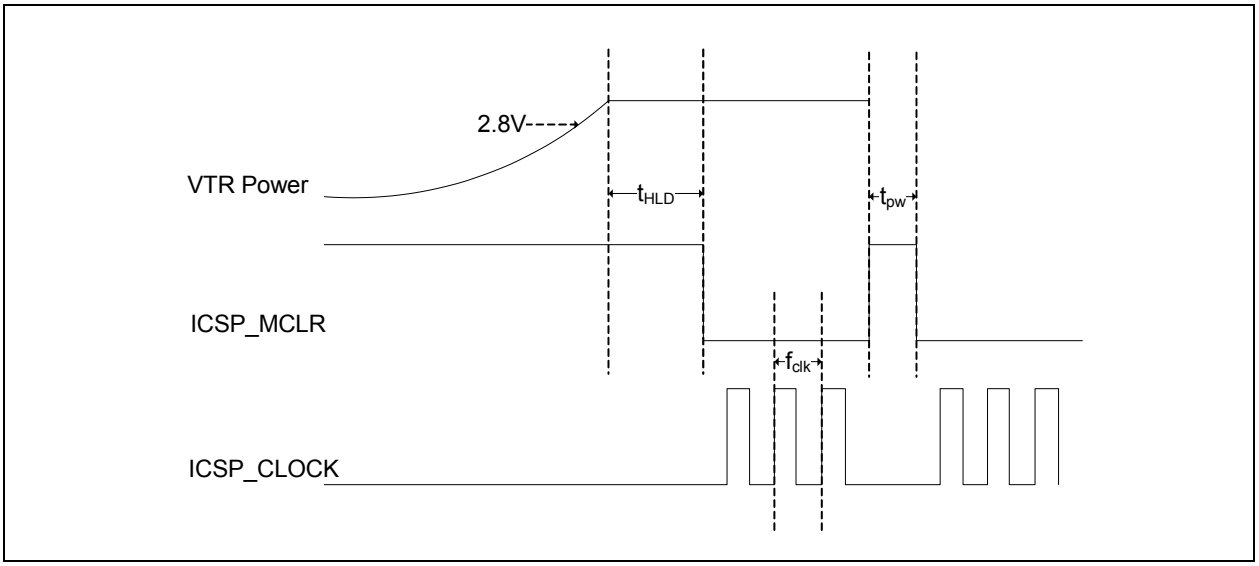


FIGURE 43-13: ICSP SETUP & HOLD PARAMETERS

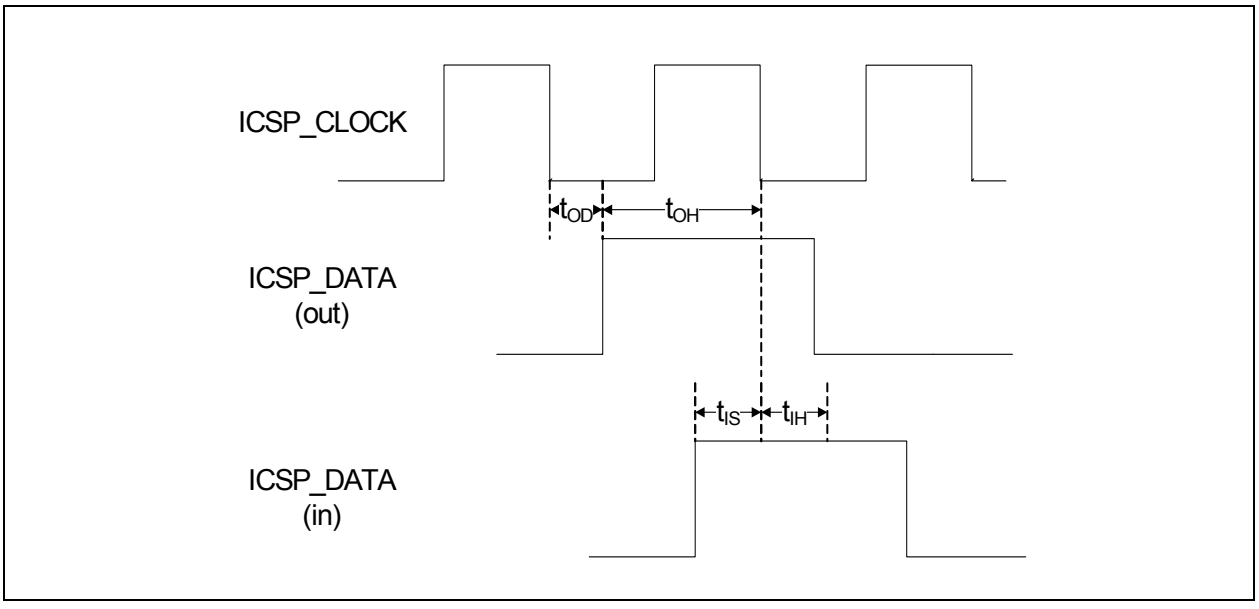


TABLE 43-15: ICSP INTERFACE TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_{HLD}	ICSP_MCLR de-assertion after VTR power is applied	5			ms
t_{pw}	ICSP_MCLR assertion pulse width	500			nsec
f_{clk}	ICSP_CLOCK frequency (see note)			48	MHz
t_{OD}	ICSP_DATA output delay after falling edge of TCLK.	5		10	nsec
t_{OH}	ICSP_DATA hold time after falling edge of TCLK	1 TCLK - t_{OD}			nsec
t_{IS}	ICSP_DATA input setup time before falling edge of TCLK.	5			nsec
t_{IH}	ICSP_DATA hold time after falling edge of TCLK.	5			nsec

Note: f_{clk} is the maximum frequency to access ICSP accessible test registers.

43.15 Test Port - XNOR

XNOR test mode is entered and exiting via the ICSP test port. Therefore, XNOR test mode must abide by the ICSP timing.

43.16 Keyboard Scan Matrix Timing

TABLE 43-16: ACTIVE PRE DRIVE MODE TIMING

Parameter	Symbol	Value			Units	Notes
		MIN	TYP	MAX		
Active Predrive Mode	$t_{PREDRIVE}$	40.87	41.7	42.5	ns	

43.17 LPC Interface Timing

43.17.1 LPC LCLK TIMING

FIGURE 43-14: LPC CLOCK TIMING

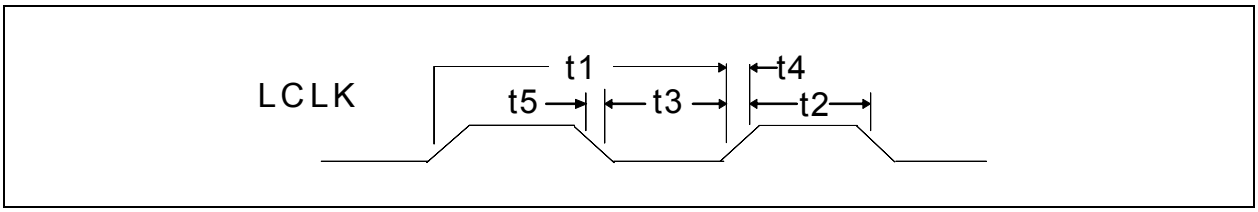


TABLE 43-17: LPC CLOCK TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Period	30		57.3 (Note 4 3-1)	nsec
t2	High Time	11			
t3	Low Time				
t4	Rise Time			3	
t5	Fall Time				

Note 43-1 The standard clock frequency supported is 33MHz (max 33.3 ns period). Setting the Handshake bit in the Host Interface allows the LPC to support 19.2 MHz (max 45.8 ns period) and 24 MHz (max 57.3 ns period) PCI clock rates.

43.17.2 LPC RESET# TIMING

FIGURE 43-15: RESET TIMING

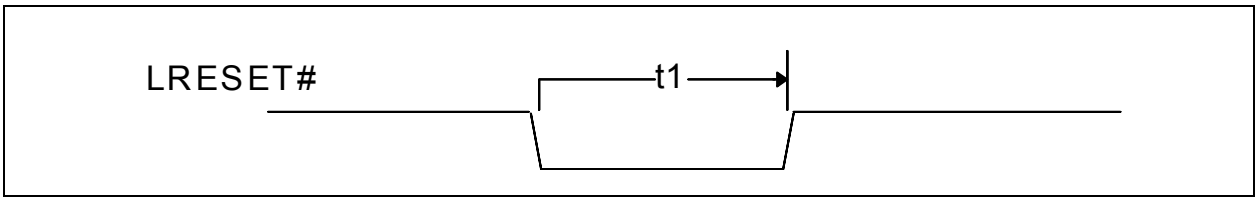


TABLE 43-18: RESET TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	LRESET# width	1			ms

43.17.3 LPC BUS TIMING

FIGURE 43-16: OUTPUT TIMING MEASUREMENT CONDITIONS, LPC SIGNALS

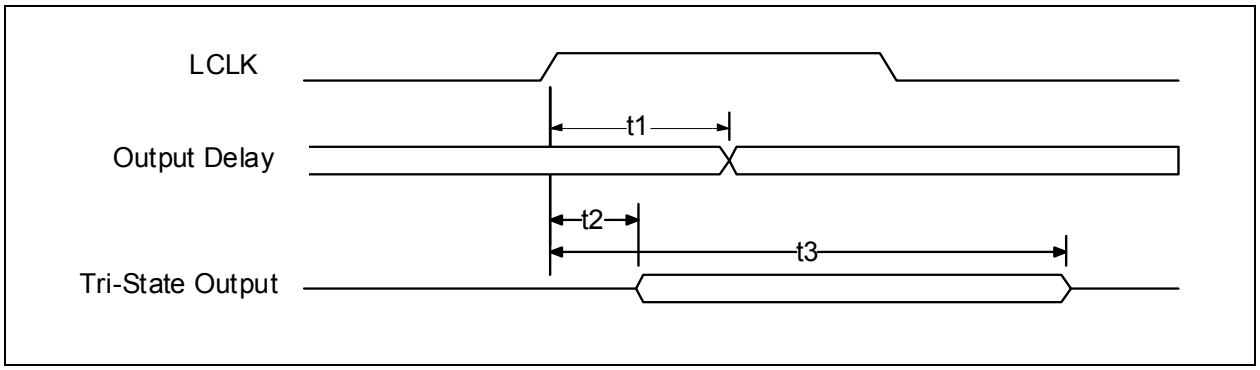


TABLE 43-19: OUTPUT TIMING MEASUREMENT CONDITIONS, LPC SIGNALS PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	LCLK to Signal Valid Delay – Bused Signals	2		11	ns
t2	Float to Active Delay				
t3	Active to Float Delay			28	

43.17.4 LPC INPUT TIMING

FIGURE 43-17: INPUT TIMING MEASUREMENT CONDITIONS, LPC SIGNALS

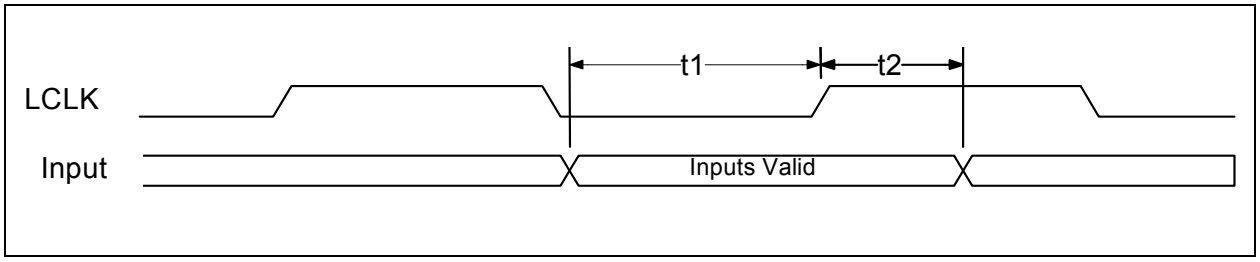
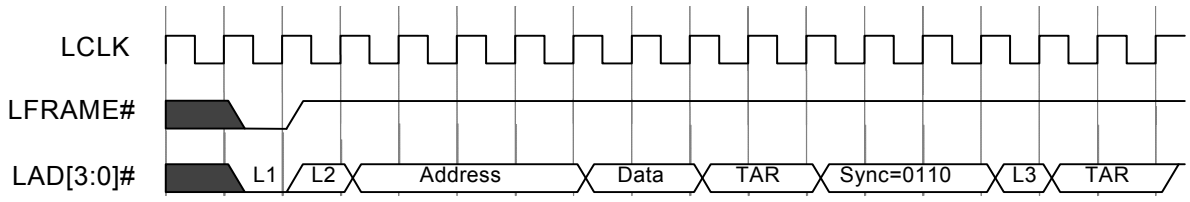


TABLE 43-20: INPUT TIMING MEASUREMENT CONDITIONS, LPC SIGNALS PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Input Set Up Time to LCLK – Bused Signals	7			ns
t2	Input Hold Time from LCLK	0			

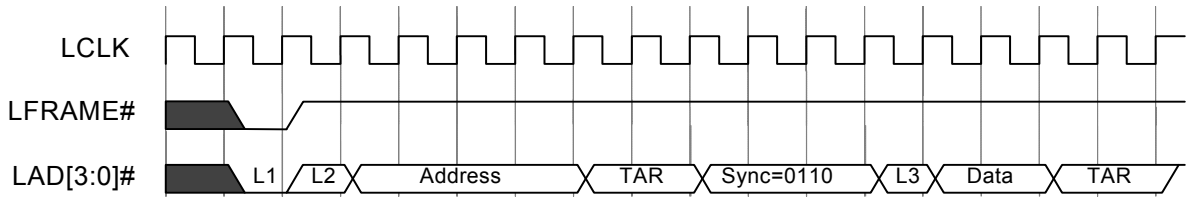
43.17.5 LPC I/O TIMING

FIGURE 43-18: I/O WRITE



Note: L1=Start; L2=CYCTYP+DIR; L3=Sync of 0000

FIGURE 43-19: I/O READ



Note: L1=Start; L2=CYCTYP+DIR; L3=Sync of 0000

43.17.6 SERIAL IRQ TIMING

FIGURE 43-20: SETUP AND HOLD TIME

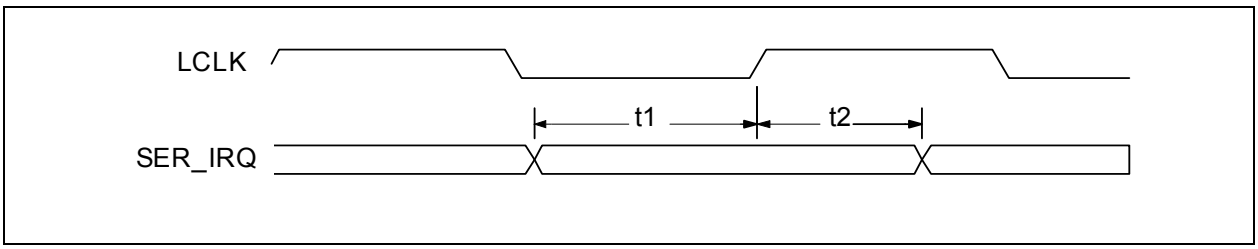


TABLE 43-21: SETUP AND HOLD TIME

Name	Description	MIN	TYP	MAX	Units
t1	SER_IRQ Setup Time to LCLK Rising	7			nsec
t2	SER_IRQ Hold Time to LCLK Rising	0			

43.17.7 nEC_SCI TIMING

nEC_SCI pin has the same minimum timing requirements as GPIO signals. See Section 43.5, "GPIO Timings," on page 507.

43.18 Serial Port (UART) Timing

43.18.1 SERIAL PORT (UART) DATA TIMING

FIGURE 43-21: SERIAL PORT DATA

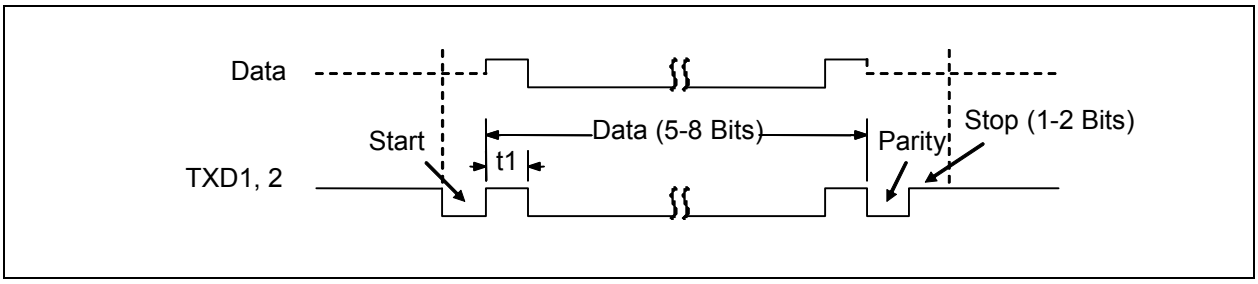


TABLE 43-22: SERIAL PORT DATA PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Serial Port Data Bit Time		t_{BR} (Note 43-1)		nsec

Note 43-1 tBR is 1/Baud Rate. The Baud Rate is programmed through the Baud_Rate_Divisor bits located in the Programmable Baud Rate Generator registers. The selectable baud rates are listed in Table 17-2, "UART Baud Rates using Clock Source 1.8432MHz_Clk" and Table 17-3, "UART Baud Rates using Clock Source 24MHz_Clk". Some of the baud rates have some percentage of error because the clock does not divide evenly. This error can be determined from the values in these baud rate tables.

43.18.2 UART_CLK TIMING

FIGURE 43-22: UART_CLK EXTERNAL CLOCK TIMING

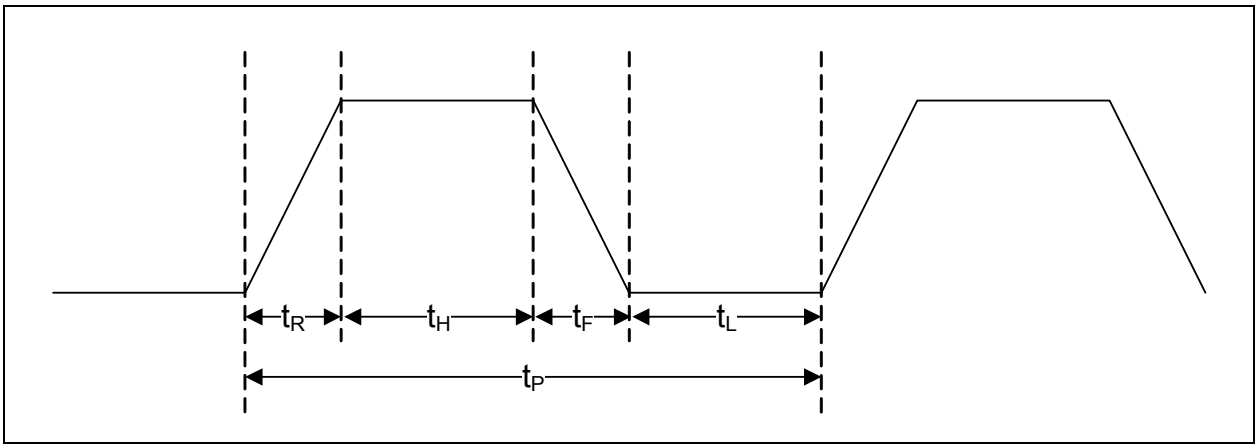


TABLE 43-23: UART_CLK EXTERNAL CLOCK TIMING PARAMETERS

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_P	Period	553.6	542.5	553.6	nsec
t_H	High Time	200			
t_L	Low Time				
t_R	Rise Time			10	
t_F	Fall Time				

43.19 PECI Interface

Name	Description	MIN	MAX	Units	Notes
t_{BIT}	Bit time (overall time evident on PECI pin) Bit time driven by an originator	0.495 0.495	500 250	µsec µsec	43-1
$t_{BIT,jitter}$	Bit time jitter between adjacent bits in a PECI message header or data bytes after timing has been negotiated	-	-	%	
$t_{BIT,drift}$	Change in bit time across a PECI address or PECI message bits as driven by the originator. This limit only applies across t_{BIT-A} bit drift and t_{BIT-M} drift.	-	-	%	
t_{H1}	High level time for logic 1	0.6	0.8	t_{BIT}	43-2
t_{H0}	High level time for logic 0	0.2	0.4	t_{BIT}	
t_{PECIR}	Rise time (measured from V_{OL} to $V_{IH,min}$, $Vtt_{(nom)}$–5%)	-	30 + (5 x #nodes)	ns	43-3
t_{PECIF}	Fall time (measured from V_{OH} to $V_{IL,max}$, $Vtt_{(nom)}$+5%)	-	(30 x #nodes)	ns	43-3

Note 43-1 The originator must drive a more restrictive time to allow for quantized sampling errors by a client yet still attain the minimum time less than 500 µsec. t_{BIT} limits apply equally to t_{BIT-A} and t_{BIT-M} . The MEC140x/1x is designed to support 2 MHz, or a 500ns bit time. See the PECI 3.0 specification from Intel Corp. for further details.

Note 43-2 The minimum and maximum bit times are relative to t_{BIT} defined in the Timing Negotiation pulse. See the PECI 3.0 specification from Intel Corp. for further details.

Note 43-3 “#nodes” is the number of nodes on the PECI bus; host and client nodes are counted as one each. Extended trace lengths may appear as extra nodes. Refer also to Table 25-2, "PECI Routing Guidelines". See the PECI 3.0 specification from Intel Corp. for further details.

43.20 PS/2 Timing

FIGURE 43-23: PS/2 TRANSMIT TIMING

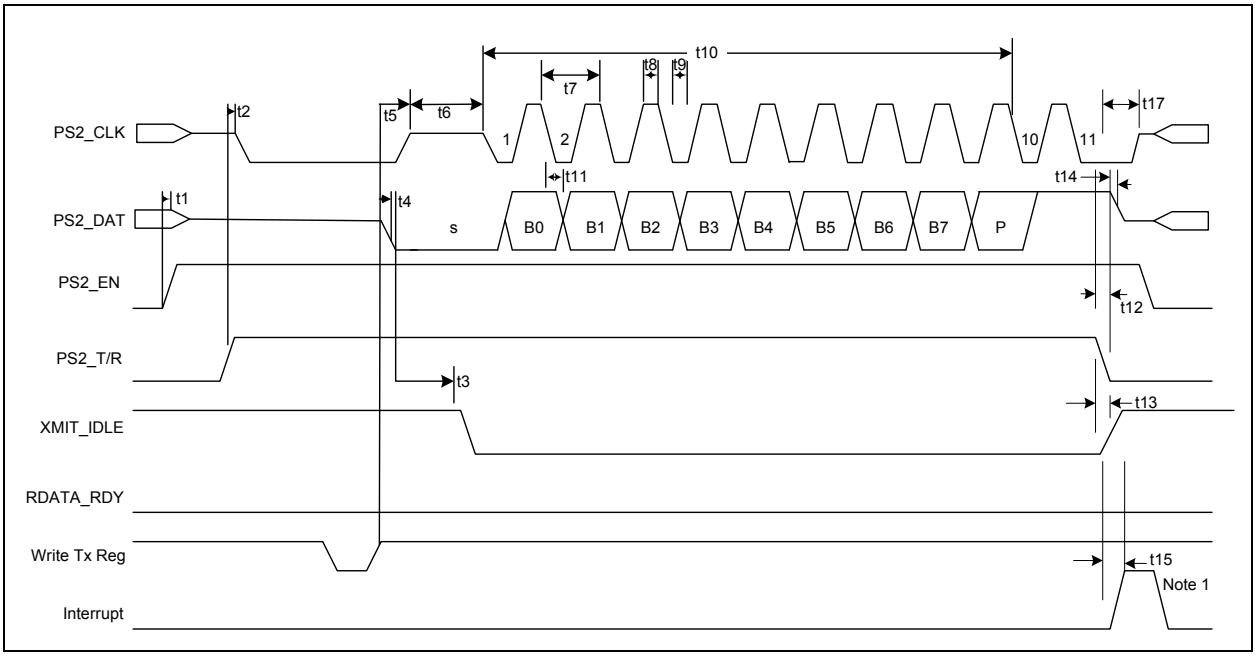


TABLE 43-24: PS/2 CHANNEL TRANSMISSION TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	The PS/2 Channel's CLK and DATA lines are floated following PS2_EN=1 and PS2_T/R=0.			1000	ns
t2	PS2_T/R bit set to CLK driven low preparing the PS/2 Channel for data transmission.				
t3	CLK line floated to XMIT_IDLE bit deasserted.			1.7	
t4	Trailing edge of WR to Transmit Register to DATA line driven low.	45		90	
t5	Trailing edge of EC WR of Transmit Register to CLK line floated.	90		130	ns
t6	Initiation of Start of Transmit cycle by the PS/2 channel controller to the auxiliary peripheral's responding by latching the Start bit and driving the CLK line low.	0.002		25.003	ms
t7	Period of CLK	60		302	µs
t8	Duration of CLK high (active)	30		151	
t9	Duration of CLK low (inactive)				

TABLE 43-24: PS/2 CHANNEL TRANSMISSION TIMING PARAMETERS (CONTINUED)

Name	Description	MIN	TYP	MAX	Units
t10	Duration of Data Frame. Falling edge of Start bit CLK (1st clk) to falling edge of Parity bit CLK (10th clk).			2.002	ms
t11	DATA output by MEC140x/1x following the falling edge of CLK. The auxiliary peripheral device samples DATA following the rising edge of CLK.			1.0	µs
t12	Rising edge following the 11th falling clock edge to PS_T/R bit driven low.	3.5		7.1	µs
t13	Trailing edge of PS_T/R to XMIT_IDLE bit asserted.			500	ns
t14	DATA released to high-Z following the PS2_T/R bit going low.				
t15	XMIT_IDLE bit driven high to interrupt generated. Note1- Interrupt is cleared by writing a 1 to the status bit in the GIRQ17 source register.				
t17	Trailing edge of CLK is held low prior to going high-Z				

FIGURE 43-24: PS/2 RECEIVE TIMING

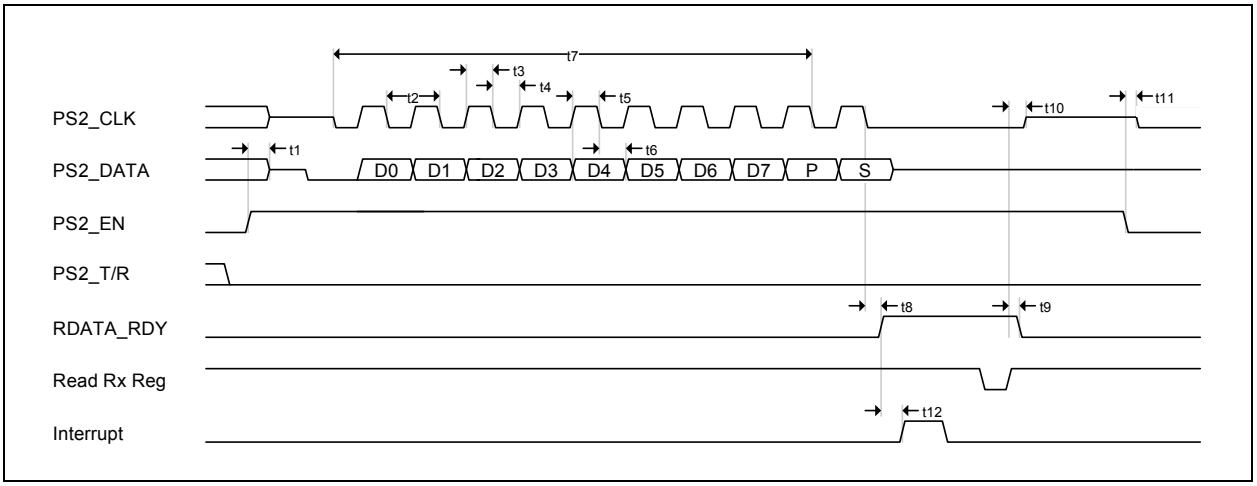


TABLE 43-25: PS/2 CHANNEL RECEIVE TIMING DIAGRAM PARAMETERS

Name	Descritpion	MIN	TYP	MAX	Units
t1	The PS/2 Channel's CLK and DATA lines are floated following PS2_EN=1 and PS2_T/R=0.			1000	ns
t2	Period of CLK	60		302	µs
t3	Duration of CLK high (active)	30		151	
t4	Duration of CLK low (inactive)				
t5	DATA setup time to falling edge of CLK. MEC140x/1x samples the data line on the falling CLK edge.	1			
t6	DATA hold time from falling edge of CLK. MEC140x/1x samples the data line on the falling CLK edge.	2			
t7	Duration of Data Frame. Falling edge of Start bit CLK (1st clk) to falling edge of Parity bit CLK (10th clk).			2.002	ms
t8	Falling edge of 11th CLK to RDATA_RDY asserted.			1.6	µs

TABLE 43-25: PS/2 CHANNEL RECEIVE TIMING DIAGRAM PARAMETERS (CONTINUED)

Name	Descritpion	MIN	TYP	MAX	Units
t9	Trailing edge of the EC's RD signal of the Receive Register to RDATA_RDY bit de-asserted.			500	ns
t10	Trailing edge of the EC's RD signal of the Receive Register to the CLK line released to high-Z.				
t11	PS2_CLK is "Low" and PS2_DATA is "Hi-Z" when PS2_EN is de-asserted.				
t12	RDATA_RDY asserted an interrupt is generated.				

43.21 PWM Timing

FIGURE 43-25: PWM OUTPUT TIMING

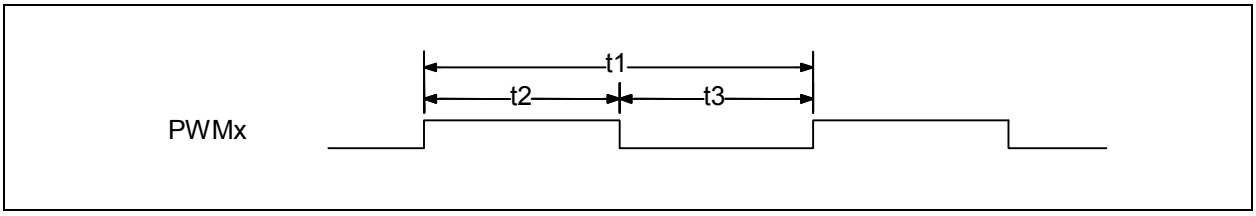


TABLE 43-26: PWM TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Period	42ns		23.3sec	
t_f	Frequency	0.04Hz		24MHz	
t2	High Time	0		11.65	sec
t3	Low Time	0		11.65	sec
t_d	Duty cycle	0		100	%

43.22 Serial Debug Port Timing

FIGURE 43-26: SERIAL DEBUG PORT TIMING PARAMETERS

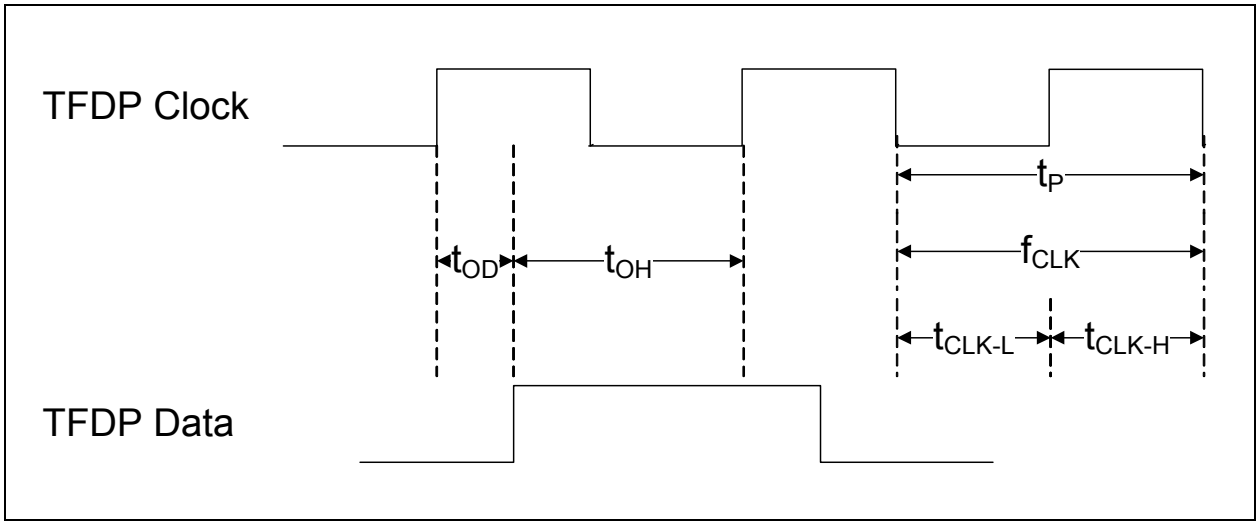


TABLE 43-27: SERIAL DEBUG PORT INTERFACE TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
f_{clk}	TFDP Clock frequency (see note)	6	-	24	MHz
t_P	TFDP Clock Period.	1/fclk			µs
t_{OD}	TFDP Data output delay after falling edge of MSCLK.			5	nsec
t_{OH}	TFDP Data hold time after falling edge of TFDP Clock	$t_P - t_{OD}$			nsec
t_{CLK-L}	TFDP Clock Low Time	$t_P/2 - 3$		$t_P/2 + 3$	nsec
t_{CLK-H}	TFDP Clock high Time (see Note 43-1)	$t_P/2 - 3$		$t_P/2 + 3$	nsec

Note 43-1 When the clock divider for the embedded controller is an odd number value greater than 2h, then $t_{CLK-L} = t_{CLK-H} + 15$ ns. When the clock divider for the embedded controller is 0h, 1h, or an even number value greater than 2h, then $t_{CLK-L} = t_{CLK-H}$.

43.23 Serial Peripheral Interface (SPI) Timings

FIGURE 43-27: SPI CLOCK TIMING

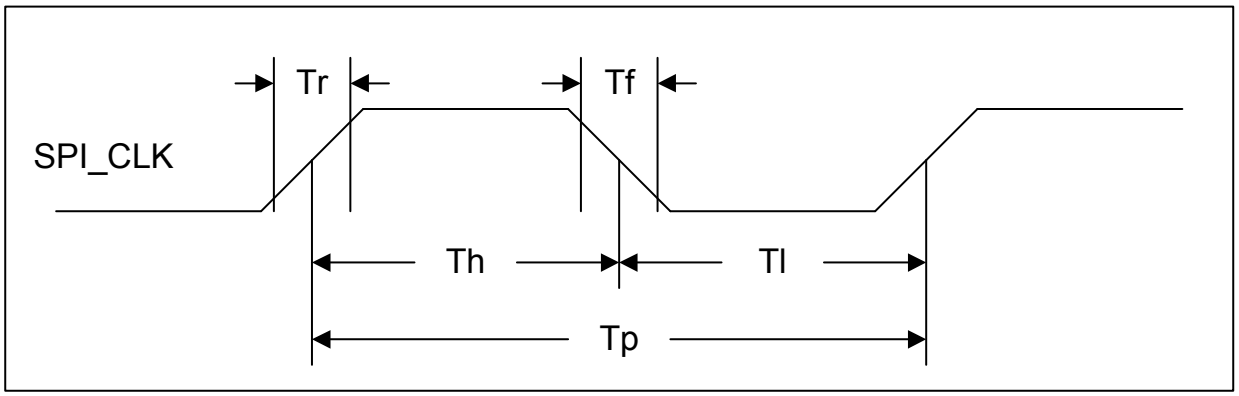
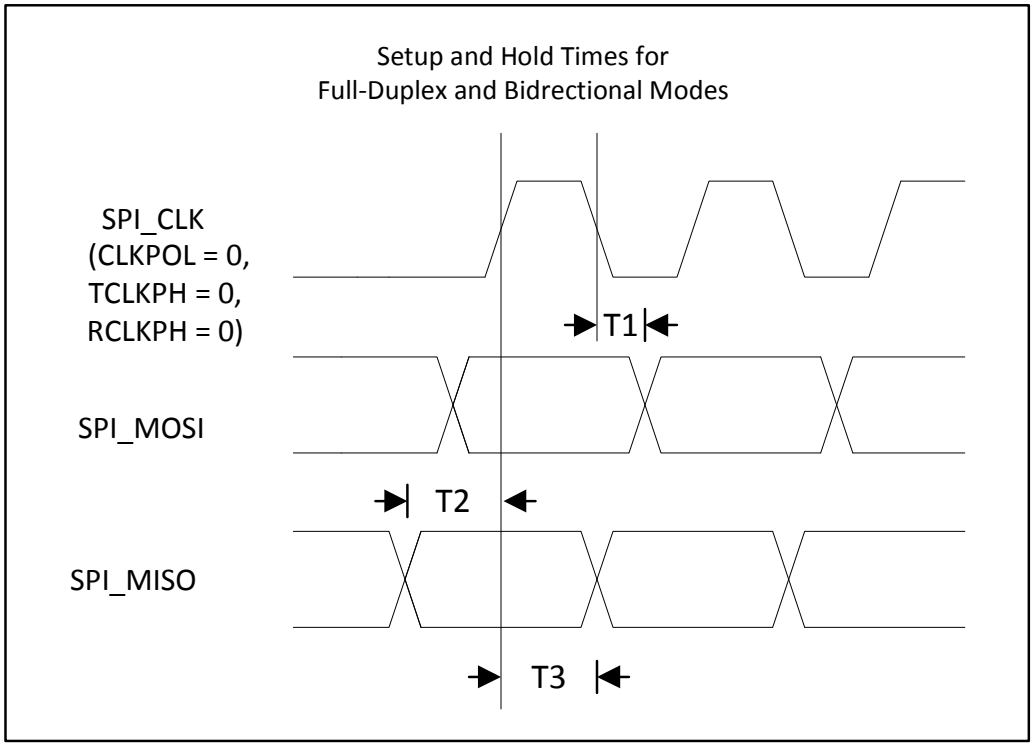


TABLE 43-28: SPI CLOCK TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units	Note
Tr	SPI Clock Rise Time. Measured from 10% to 90%.			3	ns	Note 43-2
Tf	SPI Clock Fall Time. Measured from 90% to 10%.			3	ns	Note 43-2
Th/Tl	SPI Clock High Time/SPI Clock Low Time	40% of SPCLK Period	50% of SPCLK Period	60% of SPCLK Period	ns	
Tp	SPI Clock Period – As selected by SPI Clock Generator Register	20.8		5,333	ns	

Note 43-2 Test conditions are as follows: output load is C_L=30pF, pin drive strength setting is 4mA and slew rate setting is slow.

FIGURE 43-28: SPI SETUP AND HOLD TIMES



Note: SPI_IO[3:0] obey the SPI_MOSI and SPI_MISO timing. In the 2-pin SPI Interface implementation, SPI_IO0 pin is the SPI Master-Out/Slave-In (MOSI) pin and the SPI_IO1 pin is the Master-In/Slave-out (MISO) pin.

TABLE 43-29: SPI SETUP AND HOLD TIMES PARAMETERS

Name	Description	MIN	TYP	MAX	Units
T1	Data Output Delay			2	ns
T2	Data IN Setup Time	3			ns
T3	Data IN Hold Time	0			ns

43.24 VBAT-Powered Control Interface Timing

43.24.1 VCI INPUT TIMING

FIGURE 43-29: VCI INPUT TIMING

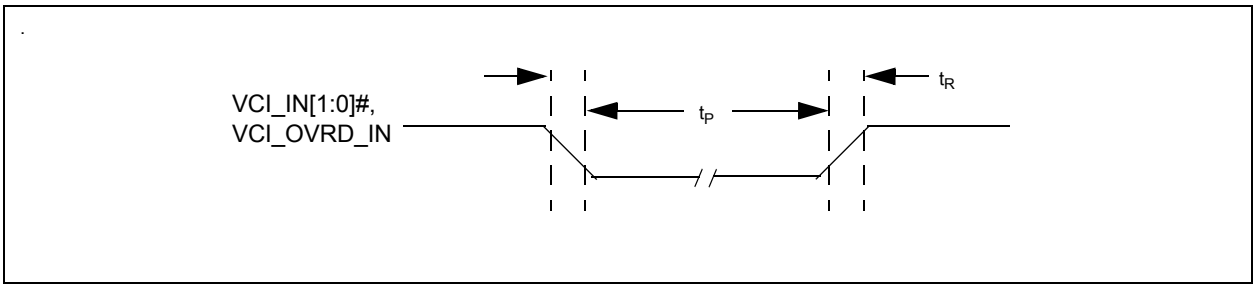


TABLE 43-30: VCI INPUT TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_F	Input fall time	–	–	1	µsec
t_R	Input rise time	–	–	1	
t_P	Pulse width of spikes suppressed by input filter	50	–	140	µsec

44.0 REGISTER MEMORY MAP

TABLE 44-1: REGISTER MEMORY MAP

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
400	Watchdog Timer Interface	0	WDT Registers	WDT Load Register	2
404	Watchdog Timer Interface	0	WDT Registers	WDT Control Register	1
408	Watchdog Timer Interface	0	WDT Registers	WDT Kick Register	1
40C	Watchdog Timer Interface	0	WDT Registers	WDT Count Register	2
C00	Basic Timer	0	Basic_Timer_EC_Only	Timer Count	4
C04	Basic Timer	0	Basic_Timer_EC_Only	Timer Preload	4
C08	Basic Timer	0	Basic_Timer_EC_Only	Timer Status	4
C0C	Basic Timer	0	Basic_Timer_EC_Only	Timer Interrupt Enable	4
C10	Basic Timer	0	Basic_Timer_EC_Only	Timer Control	4
C20	Basic Timer	1	Basic_Timer_EC_Only	Timer Count	4
C24	Basic Timer	1	Basic_Timer_EC_Only	Timer Preload	4
C28	Basic Timer	1	Basic_Timer_EC_Only	Timer Status	4
C2C	Basic Timer	1	Basic_Timer_EC_Only	Timer Interrupt Enable	4
C30	Basic Timer	1	Basic_Timer_EC_Only	Timer Control	4
C40	Basic Timer	2	Basic_Timer_EC_Only	Timer Count	4
C44	Basic Timer	2	Basic_Timer_EC_Only	Timer Preload	4
C48	Basic Timer	2	Basic_Timer_EC_Only	Timer Status	4
C4C	Basic Timer	2	Basic_Timer_EC_Only	Timer Interrupt Enable	4
C50	Basic Timer	2	Basic_Timer_EC_Only	Timer Control	4
C60	Basic Timer	3	Basic_Timer_EC_Only	Timer Count	4
C64	Basic Timer	3	Basic_Timer_EC_Only	Timer Preload	4
C68	Basic Timer	3	Basic_Timer_EC_Only	Timer Status	4
C6C	Basic Timer	3	Basic_Timer_EC_Only	Timer Interrupt Enable	4
C70	Basic Timer	3	Basic_Timer_EC_Only	Timer Control	4
1800	SMB Device Interface	0	SMB_EC_Only	Status Register	1
1800	SMB Device Interface	0	SMB_EC_Only	Control Register	1
1801	SMB Device Interface	0	SMB_EC_Only	Reserved	3
1804	SMB Device Interface	0	SMB_EC_Only	Own Address Register	2
1806	SMB Device Interface	0	SMB_EC_Only	Reserved	2
1808	SMB Device Interface	0	SMB_EC_Only	Data	1
1809	SMB Device Interface	0	SMB_EC_Only	Reserved	3
180C	SMB Device Interface	0	SMB_EC_Only	SMBus Master Command Register	4
1810	SMB Device Interface	0	SMB_EC_Only	SMBus Slave Command Register	4
1814	SMB Device Interface	0	SMB_EC_Only	PEC Register	1
1815	SMB Device Interface	0	SMB_EC_Only	Reserved	3

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
1818	SMB Device Interface	0	SMB_EC_Only	DATA_TIMING2	1
1819	SMB Device Interface	0	SMB_EC_Only	Reserved	3
1820	SMB Device Interface	0	SMB_EC_Only	Completion Register	4
1824	SMB Device Interface	0	SMB_EC_Only	Idle Scaling Register	4
1828	SMB Device Interface	0	SMB_EC_Only	Configuration Register	4
182C	SMB Device Interface	0	SMB_EC_Only	Bus Clock Register	2
182E	SMB Device Interface	0	SMB_EC_Only	Reserved	2
1830	SMB Device Interface	0	SMB_EC_Only	Block ID Register	1
1831	SMB Device Interface	0	SMB_EC_Only	Reserved	3
1834	SMB Device Interface	0	SMB_EC_Only	Revision Register	1
1835	SMB Device Interface	0	SMB_EC_Only	Reserved	3
1838	SMB Device Interface	0	SMB_EC_Only	Bit-Bang Control Register	1
1839	SMB Device Interface	0	SMB_EC_Only	Reserved	3
1840	SMB Device Interface	0	SMB_EC_Only	Data Timing Register	4
1844	SMB Device Interface	0	SMB_EC_Only	Time-Out Scaling Register	4
1848	SMB Device Interface	0	SMB_EC_Only	SMBus Slave Transmit Buffer Register	1
1849	SMB Device Interface	0	SMB_EC_Only	Reserved	3
184C	SMB Device Interface	0	SMB_EC_Only	SMBus Slave Receive Buffer Register	1
184D	SMB Device Interface	0	SMB_EC_Only	Reserved	3
1850	SMB Device Interface	0	SMB_EC_Only	SMBus Master Transmit Bufer Register	1
1851	SMB Device Interface	0	SMB_EC_Only	Reserved	3
1854	SMB Device Interface	0	SMB_EC_Only	SMBus Master Receive Buffer Register	1
1855	SMB Device Interface	0	SMB_EC_Only	Reserved	3
1860	SMB Device Interface	0	SMB_EC_Only	Wake Status register	4
1864	SMB Device Interface	0	SMB_EC_Only	Wake Enable register	4
2400	DMA	0	DMA Main	DMA Main Control Register	1
2401	DMA	0	DMA Main	DMA Reserved	3
2404	DMA	0	DMA Main	DMA AFIFO Data Register	4
2440	DMA	0	DMA_CH0	DMA Activate Register	4
2444	DMA	0	DMA_CH0	DMA Memory Start Address Register	4
2448	DMA	0	DMA_CH0	DMA Memory End Address Register	4
244C	DMA	0	DMA_CH0	AHB Address Register	4
2450	DMA	0	DMA_CH0	DMA Control Register	4
2454	DMA	0	DMA_CH0	DMA Channel Interrupt Status	4
2458	DMA	0	DMA_CH0	DMA Channel Interrupt Enable	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
2460	DMA	0	DMA_CH0_CRC	DMA Channel 0 CRC Enable Register	4
2464	DMA	0	DMA_CH0_CRC	DMA Channel 0 CRC Data Register	4
2468	DMA	0	DMA_CH0_CRC	DMA Channel 0 CRC Post Status Register	4
2480	DMA	0	DMA_CH1	DMA Activate Register	4
2484	DMA	0	DMA_CH1	DMA Memory Start Address Register	4
2488	DMA	0	DMA_CH1	DMA Memory End Address Register	4
248C	DMA	0	DMA_CH1	AHB Address Register	4
2490	DMA	0	DMA_CH1	DMA Control Register	4
2494	DMA	0	DMA_CH1	DMA Channel Interrupt Status	4
2498	DMA	0	DMA_CH1	DMA Channel Interrupt Enable	4
24A0	DMA	0	DMA_CH1_NOCRC	Reserved	22
24C0	DMA	0	DMA_CH2	DMA Activate Register	4
24C4	DMA	0	DMA_CH2	DMA Memory Start Address Register	4
24C8	DMA	0	DMA_CH2	DMA Memory End Address Register	4
24CC	DMA	0	DMA_CH2	AHB Address Register	4
24D0	DMA	0	DMA_CH2	DMA Control Register	4
24D4	DMA	0	DMA_CH2	DMA Channel Interrupt Status	4
24D8	DMA	0	DMA_CH2	DMA Channel Interrupt Enable	4
24E0	DMA	0	DMA_CH2_NOCRC	Reserved	22
2500	DMA	0	DMA_CH3	DMA Activate Register	4
2504	DMA	0	DMA_CH3	DMA Memory Start Address Register	4
2508	DMA	0	DMA_CH3	DMA Memory End Address Register	4
250C	DMA	0	DMA_CH3	AHB Address Register	4
2510	DMA	0	DMA_CH3	DMA Control Register	4
2514	DMA	0	DMA_CH3	DMA Channel Interrupt Status	4
2518	DMA	0	DMA_CH3	DMA Channel Interrupt Enable	4
2520	DMA	0	DMA_CH3_NOCRC	Reserved	22
2540	DMA	0	DMA_CH4	DMA Activate Register	4
2544	DMA	0	DMA_CH4	DMA Memory Start Address Register	4
2548	DMA	0	DMA_CH4	DMA Memory End Address Register	4
254C	DMA	0	DMA_CH4	AHB Address Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
2550	DMA	0	DMA_CH4	DMA Control Register	4
2554	DMA	0	DMA_CH4	DMA Channel Interrupt Status	4
2558	DMA	0	DMA_CH4	DMA Channel Interrupt Enable	4
2560	DMA	0	DMA_CH4_NOCRC	Reserved	22
2580	DMA	0	DMA_CH5	DMA Activate Register	4
2584	DMA	0	DMA_CH5	DMA Memory Start Address Register	4
2588	DMA	0	DMA_CH5	DMA Memory End Address Register	4
258C	DMA	0	DMA_CH5	AHB Address Register	4
2590	DMA	0	DMA_CH5	DMA Control Register	4
2594	DMA	0	DMA_CH5	DMA Channel Interrupt Status	4
2598	DMA	0	DMA_CH5	DMA Channel Interrupt Enable	4
25A0	DMA	0	DMA_CH5_NOCRC	Reserved	22
25C0	DMA	0	DMA_CH6	DMA Activate Register	4
25C4	DMA	0	DMA_CH6	DMA Memory Start Address Register	4
25C8	DMA	0	DMA_CH6	DMA Memory End Address Register	4
25CC	DMA	0	DMA_CH6	AHB Address Register	4
25D0	DMA	0	DMA_CH6	DMA Control Register	4
25D4	DMA	0	DMA_CH6	DMA Channel Interrupt Status	4
25D8	DMA	0	DMA_CH6	DMA Channel Interrupt Enable	4
25E0	DMA	0	DMA_CH6_NOCRC	Reserved	22
5400	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Mode	4
5404	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Control	4
5408	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI IF Control	4
540C	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Execute	4
5410	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Status	4
5414	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Buffer Count Status	4
5418	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Buffer Trigger	4
541C	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Interrupt Enable	4
5420	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Rx Buffer	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
5424	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Tx Buffer	4
5430	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Description Buffer 0	4
5434	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Description Buffer 1	4
5438	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Description Buffer 2	4
543C	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Description Buffer 3	4
5440	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Description Buffer 4	4
5800	PWM	0	PWM_EC_Only	PWM Counter ON Time Register	4
5804	PWM	0	PWM_EC_Only	PWM Counter OFF Time Register	4
5808	PWM	0	PWM_EC_Only	PWM Configuration Register	4
580C	PWM	0	PWM_EC_Only	Reserved	4
5810	PWM	1	PWM_EC_Only	PWM Counter ON Time Register	4
5814	PWM	1	PWM_EC_Only	PWM Counter OFF Time Register	4
5818	PWM	1	PWM_EC_Only	PWM Configuration Register	4
581C	PWM	1	PWM_EC_Only	Reserved	4
5820	PWM	2	PWM_EC_Only	PWM Counter ON Time Register	4
5824	PWM	2	PWM_EC_Only	PWM Counter OFF Time Register	4
5828	PWM	2	PWM_EC_Only	PWM Configuration Register	4
582C	PWM	2	PWM_EC_Only	Reserved	4
5830	PWM	3	PWM_EC_Only	PWM Counter ON Time Register	4
5834	PWM	3	PWM_EC_Only	PWM Counter OFF Time Register	4
5838	PWM	3	PWM_EC_Only	PWM Configuration Register	4
583C	PWM	3	PWM_EC_Only	Reserved	4
5840	PWM	4	PWM_EC_Only	PWM Counter ON Time Register	4
5844	PWM	4	PWM_EC_Only	PWM Counter OFF Time Register	4
5848	PWM	4	PWM_EC_Only	PWM Configuration Register	4
584C	PWM	4	PWM_EC_Only	Reserved	4
5850	PWM	5	PWM_EC_Only	PWM Counter ON Time Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
5854	PWM	5	PWM_EC_Only	PWM Counter OFF Time Register	4
5858	PWM	5	PWM_EC_Only	PWM Configuration Register	4
585C	PWM	5	PWM_EC_Only	Reserved	4
5860	PWM	6	PWM_EC_Only	PWM Counter ON Time Register	4
5864	PWM	6	PWM_EC_Only	PWM Counter OFF Time Register	4
5868	PWM	6	PWM_EC_Only	PWM Configuration Register	4
586C	PWM	6	PWM_EC_Only	Reserved	4
5870	PWM	7	PWM_EC_Only	PWM Counter ON Time Register	4
5874	PWM	7	PWM_EC_Only	PWM Counter OFF Time Register	4
5878	PWM	7	PWM_EC_Only	PWM Configuration Register	4
587C	PWM	7	PWM_EC_Only	Reserved	4
6000	TACH	0	TACH_EC_ONLY	TACH Control Register	4
6004	TACH	0	TACH_EC_ONLY	TACH Status Register	4
6008	TACH	0	TACH_EC_ONLY	TACH High Limit Register	4
600C	TACH	0	TACH_EC_ONLY	TACH Low Limit Register	4
6010	TACH	1	TACH_EC_ONLY	TACH Control Register	4
6014	TACH	1	TACH_EC_ONLY	TACH Status Register	4
6018	TACH	1	TACH_EC_ONLY	TACH High Limit Register	4
601C	TACH	1	TACH_EC_ONLY	TACH Low Limit Register	4
6400	PECI	0	PECI_EC_Only	PECI Write Data Register	4
6404	PECI	0	PECI_EC_Only	PECI Read Data Register	4
6408	PECI	0	PECI_EC_Only	PECI Control Register	4
640C	PECI	0	PECI_EC_Only	PECI Status 1 Register	4
6410	PECI	0	PECI_EC_Only	PECI Status 2 Register	4
6414	PECI	0	PECI_EC_Only	PECI Error Register	4
6418	PECI	0	PECI_EC_Only	PECI Interrupt Enable 1 Register	4
641C	PECI	0	PECI_EC_Only	PECI Interrupt Enable 2 Register	4
6420	PECI	0	PECI_EC_Only	PECI Optimal Bit Time (Low Byte) Register	4
6424	PECI	0	PECI_EC_Only	PECI Optimal Bit Time (High Byte) Register	4
6430	PECI	0	PECI_EC_Only	PECI Reserved	16
6440	PECI	0	PECI_EC_Only	PECI Block ID Register	4
6444	PECI	0	PECI_EC_Only	Block Revision	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
7400	RTOS Timer	0	RTOS Registers	RTOS Timer Count Value	4
7404	RTOS Timer	0	RTOS Registers	RTOS Timer Pre-Load	4
7408	RTOS Timer	0	RTOS Registers	Timer Control	4
7C00	ADC	0	ADC Registers	ADC Control Register	4
7C04	ADC	0	ADC Registers	ADC Delay Register	4
7C08	ADC	0	ADC Registers	ADC Status Register	4
7C0C	ADC	0	ADC Registers	ADC Single Register	4
7C10	ADC	0	ADC Registers	ADC Repeat Register	4
7C14	ADC	0	ADC Registers	ADC Channel 0 Reading Registers	4
7C18	ADC	0	ADC Registers	ADC Channel 1 Reading Registers	4
7C1C	ADC	0	ADC Registers	ADC Channel 2 Reading Registers	4
7C20	ADC	0	ADC Registers	ADC Channel 3 Reading Registers	4
7C24	ADC	0	ADC Registers	ADC Channel 4 Reading Registers	4
7C28	ADC	0	ADC Registers	ADC Channel 5 Reading Registers	4
7C2C	ADC	0	ADC Registers	ADC Channel 6 Reading Registers	4
7C30	ADC	0	ADC Registers	ADC Channel 7 Reading Registers	4
8000	DAC	0	DAC Registers	DAC Activate Register	4
8004	DAC	0	DAC Registers	DAC Configuration Register	4
8008	DAC	0	DAC Registers	DAC Control Register	4
800C	DAC	0	DAC Registers	DAC Data Register	4
8040	DAC	1	DAC Registers	DAC Activate Register	4
8044	DAC	1	DAC Registers	DAC Configuration Register	4
8048	DAC	1	DAC Registers	DAC Control Register	4
804C	DAC	1	DAC Registers	DAC Data Register	4
8C00	Trace FIFO Debug Port	0	TFDP	Data	4
8C04	Trace FIFO Debug Port	0	TFDP	Control	4
9000	PS/2	0	Registers	PS/2 Transmit Buffer Register	1
9000	PS/2	0	Registers	PS/2 Receive Buffer Register	1
9004	PS/2	0	Registers	PS/2 Control Register	1
9008	PS/2	0	Registers	PS/2 Status Register	1
9040	PS/2	1	Registers	PS/2 Transmit Buffer Register	1
9040	PS/2	1	Registers	PS/2 Receive Buffer Register	1
9044	PS/2	1	Registers	PS/2 Control Register	1

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
9048	PS/2	1	Registers	PS/2 Status Register	1
9800	Hibernation Timer	0	Registers	HTimer x Preload Register	2
9804	Hibernation Timer	0	Registers	Hibernation Timer x Control Register	2
9808	Hibernation Timer	0	Registers	Hibernation Timer x Count Register	2
9C00	Keyboard Matrix Scan Support	0	Registers	Reserved	4
9C04	Keyboard Matrix Scan Support	0	Registers	KSO Select Register	4
9C08	Keyboard Matrix Scan Support	0	Registers	KSI Input Register	4
9C0C	Keyboard Matrix Scan Support	0	Registers	KSI Status Register	4
9C10	Keyboard Matrix Scan Support	0	Registers	KSI Interrupt Enable Register	4
9C14	Keyboard Matrix Scan Support	0	Registers	Keyscan Extended Control Register	4
A400	VBAT Registers	0	VBAT_REG_BANK	Power-Fail and Reset Status Register	1
A408	VBAT Registers	0	VBAT_REG_BANK	Clock Enable Register	1
A418	VBAT Registers	0	VBAT_REG_BANK	Alternate Function VTR Control	4
A800	VBAT Powered RAM	0	Registers	VBAT Backed Memory	64
AC00	SMB Device Interface	1	SMB_EC_Only	Status Register	1
AC00	SMB Device Interface	1	SMB_EC_Only	Control Register	1
AC01	SMB Device Interface	1	SMB_EC_Only	Reserved	3
AC04	SMB Device Interface	1	SMB_EC_Only	Own Address Register	2
AC06	SMB Device Interface	1	SMB_EC_Only	Reserved	2
AC08	SMB Device Interface	1	SMB_EC_Only	Data	1
AC09	SMB Device Interface	1	SMB_EC_Only	Reserved	3
AC0C	SMB Device Interface	1	SMB_EC_Only	SMBus Master Command Register	4
AC10	SMB Device Interface	1	SMB_EC_Only	SMBus Slave Command Register	4
AC14	SMB Device Interface	1	SMB_EC_Only	PEC Register	1
AC15	SMB Device Interface	1	SMB_EC_Only	Reserved	3
AC18	SMB Device Interface	1	SMB_EC_Only	DATA_TIMING2	1
AC19	SMB Device Interface	1	SMB_EC_Only	Reserved	3
AC20	SMB Device Interface	1	SMB_EC_Only	Completion Register	4
AC24	SMB Device Interface	1	SMB_EC_Only	Idle Scaling Register	4
AC28	SMB Device Interface	1	SMB_EC_Only	Configuration Register	4
AC2C	SMB Device Interface	1	SMB_EC_Only	Bus Clock Register	2

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
AC2E	SMB Device Interface	1	SMB_EC_Only	Reserved	2
AC30	SMB Device Interface	1	SMB_EC_Only	Block ID Register	1
AC31	SMB Device Interface	1	SMB_EC_Only	Reserved	3
AC34	SMB Device Interface	1	SMB_EC_Only	Revision Register	1
AC35	SMB Device Interface	1	SMB_EC_Only	Reserved	3
AC38	SMB Device Interface	1	SMB_EC_Only	Bit-Bang Control Register	1
AC39	SMB Device Interface	1	SMB_EC_Only	Reserved	3
AC40	SMB Device Interface	1	SMB_EC_Only	Data Timing Register	4
AC44	SMB Device Interface	1	SMB_EC_Only	Time-Out Scaling Register	4
AC48	SMB Device Interface	1	SMB_EC_Only	SMBus Slave Transmit Buffer Register	1
AC49	SMB Device Interface	1	SMB_EC_Only	Reserved	3
AC4C	SMB Device Interface	1	SMB_EC_Only	SMBus Slave Receive Buffer Register	1
AC4D	SMB Device Interface	1	SMB_EC_Only	Reserved	3
AC50	SMB Device Interface	1	SMB_EC_Only	SMBus Master Transmit Bufer Register	1
AC51	SMB Device Interface	1	SMB_EC_Only	Reserved	3
AC54	SMB Device Interface	1	SMB_EC_Only	SMBus Master Receive Buffer Register	1
AC55	SMB Device Interface	1	SMB_EC_Only	Reserved	3
AC60	SMB Device Interface	1	SMB_EC_Only	Wake Status register	4
AC64	SMB Device Interface	1	SMB_EC_Only	Wake Enable register	4
B000	SMB Device Interface	2	SMB_EC_Only	Control Register	1
B000	SMB Device Interface	2	SMB_EC_Only	Status Register	1
B001	SMB Device Interface	2	SMB_EC_Only	Reserved	3
B004	SMB Device Interface	2	SMB_EC_Only	Own Address Register	2
B006	SMB Device Interface	2	SMB_EC_Only	Reserved	2
B008	SMB Device Interface	2	SMB_EC_Only	Data	1
B009	SMB Device Interface	2	SMB_EC_Only	Reserved	3
B00C	SMB Device Interface	2	SMB_EC_Only	SMBus Master Command Register	4
B010	SMB Device Interface	2	SMB_EC_Only	SMBus Slave Command Register	4
B014	SMB Device Interface	2	SMB_EC_Only	PEC Register	1
B015	SMB Device Interface	2	SMB_EC_Only	Reserved	3
B018	SMB Device Interface	2	SMB_EC_Only	DATA_TIMING2	1
B019	SMB Device Interface	2	SMB_EC_Only	Reserved	3
B020	SMB Device Interface	2	SMB_EC_Only	Completion Register	4
B024	SMB Device Interface	2	SMB_EC_Only	Idle Scaling Register	4
B028	SMB Device Interface	2	SMB_EC_Only	Configuration Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
B02C	SMB Device Interface	2	SMB_EC_Only	Bus Clock Register	2
B02E	SMB Device Interface	2	SMB_EC_Only	Reserved	2
B030	SMB Device Interface	2	SMB_EC_Only	Block ID Register	1
B031	SMB Device Interface	2	SMB_EC_Only	Reserved	3
B034	SMB Device Interface	2	SMB_EC_Only	Revision Register	1
B035	SMB Device Interface	2	SMB_EC_Only	Reserved	3
B038	SMB Device Interface	2	SMB_EC_Only	Bit-Bang Control Register	1
B039	SMB Device Interface	2	SMB_EC_Only	Reserved	3
B040	SMB Device Interface	2	SMB_EC_Only	Data Timing Register	4
B044	SMB Device Interface	2	SMB_EC_Only	Time-Out Scaling Register	4
B048	SMB Device Interface	2	SMB_EC_Only	SMBus Slave Transmit Buffer Register	1
B049	SMB Device Interface	2	SMB_EC_Only	Reserved	3
B04C	SMB Device Interface	2	SMB_EC_Only	SMBus Slave Receive Buffer Register	1
B04D	SMB Device Interface	2	SMB_EC_Only	Reserved	3
B050	SMB Device Interface	2	SMB_EC_Only	SMBus Master Transmit Bufer Register	1
B051	SMB Device Interface	2	SMB_EC_Only	Reserved	3
B054	SMB Device Interface	2	SMB_EC_Only	SMBus Master Receive Buffer Register	1
B055	SMB Device Interface	2	SMB_EC_Only	Reserved	3
B060	SMB Device Interface	2	SMB_EC_Only	Wake Status register	4
B064	SMB Device Interface	2	SMB_EC_Only	Wake Enable register	4
B800	LED	0	EC-Only Registers	LED Configuration	4
B804	LED	0	EC-Only Registers	LED Limits	4
B808	LED	0	EC-Only Registers	LED Delay	4
B80C	LED	0	EC-Only Registers	LED Update Stepsize	4
B810	LED	0	EC-Only Registers	LED Update Interval	4
B900	LED	1	EC-Only Registers	LED Configuration	4
B904	LED	1	EC-Only Registers	LED Limits	4
B908	LED	1	EC-Only Registers	LED Delay	4
B90C	LED	1	EC-Only Registers	LED Update Stepsize	4
B910	LED	1	EC-Only Registers	LED Update Interval	4
BA00	LED	2	EC-Only Registers	LED Configuration	4
BA04	LED	2	EC-Only Registers	LED Limits	4
BA08	LED	2	EC-Only Registers	LED Delay	4
BA0C	LED	2	EC-Only Registers	LED Update Stepsize	4
BA10	LED	2	EC-Only Registers	LED Update Interval	4
BC00	BC-Link Master	0	Registers	BC-Link Status Register	1

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
BC04	BC-Link Master	0	Registers	BC-Link Address Register	1
BC08	BC-Link Master	0	Registers	BC-Link Data Register	1
BC0C	BC-Link Master	0	Registers	BC-Link Clock Select Register	1
BD00	BC-Link Master	1	Registers	BC-Link Status Register	1
BD04	BC-Link Master	1	Registers	BC-Link Address Register	1
BD08	BC-Link Master	1	Registers	BC-Link Data Register	1
BD0C	BC-Link Master	1	Registers	BC-Link Clock Select Register	1
CC80	Week Timer	0	Registers	Control Register	1
CC84	Week Timer	0	Registers	28-bit Up-Counter Timer Register	4
CC88	Week Timer	0	Registers	28-bit Comparator Register	4
CC8C	Week Timer	0	Registers	15-Bit Clock Divider Reading Register	2
CC90	Week Timer	0	Registers	Sub-second Programmable Interrupt Select Register	1
CC94	Week Timer	0	Registers	Sub-Week Control Register	2
CC98	Week Timer	0	Registers	Sub-Week Timer Register	4
D000	VBAT-Powered Control Interface	0	Registers	VCI Register	4
D004	VBAT-Powered Control Interface	0	Registers	Latch Enable Register	4
D008	VBAT-Powered Control Interface	0	Registers	Latch Resets Register	4
D00C	VBAT-Powered Control Interface	0	Registers	VCI Input Enable Register	4
D014	VBAT-Powered Control Interface	0	Registers	VCI Polarity Register	1
D018	VBAT-Powered Control Interface	0	Registers	VCI Posedge Detect Register	1
D01C	VBAT-Powered Control Interface	0	Registers	VCI Negedge Detect Register	1
D020	VBAT-Powered Control Interface	0	Registers	VCI Buffer Enable Register	1
FC14	EC_REG_BANK	0	EC_REG_BANK	AHB Error Control	1
FC18	EC_REG_BANK	0	EC_REG_BANK	Comparator Control	4
FC20	EC_REG_BANK	0	EC_REG_BANK	JTAG Enable	4
FC28	EC_REG_BANK	0	EC_REG_BANK	WDT Count	4
FC48	EC_REG_BANK	0	EC_REG_BANK	Power Regions Voltage Control	4
80100	PCR	0	EC-Only Registers	Chip Sleep Enable Register	4
80104	PCR	0	EC-Only Registers	Chip Clock Required Register	4
80108	PCR	0	EC-Only Registers	EC Sleep Enables Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
8010C	PCR	0	EC-Only Registers	EC Clock Required Status Register	4
80110	PCR	0	EC-Only Registers	Host Sleep Enables Register	4
80114	PCR	0	EC-Only Registers	Host Clock Required Status Register	4
80118	PCR	0	EC-Only Registers	CHIP_PCR_ADDR_SYS-_SLEEP_CTRL_0	4
80120	PCR	0	EC-Only Registers	Processor Clock Control	4
80124	PCR	0	EC-Only Registers	EC Sleep Enable 2 Register	4
80128	PCR	0	EC-Only Registers	EC Clock Required 2 Status Register	4
8012C	PCR	0	EC-Only Registers	Slow Clock Control	4
80134	PCR	0	EC-Only Registers	CHIP_PWR_RST_STS	4
80138	PCR	0	EC-Only Registers	Chip Reset Enable	4
8013C	PCR	0	EC-Only Registers	Host Reset Enable	4
80140	PCR	0	EC-Only Registers	EC Reset Enable	4
80144	PCR	0	EC-Only Registers	EC Reset Enable 2	4
80148	PCR	0	EC-Only Registers	Power Reset Control	4
81004	GPIO	0	GPIO Registers	GPIO001 Pin Control	4
81008	GPIO	0	GPIO Registers	GPIO002 Pin Control	4
8100C	GPIO	0	GPIO Registers	GPIO003 Pin Control	4
81010	GPIO	0	GPIO Registers	GPIO004 Pin Control	4
81014	GPIO	0	GPIO Registers	GPIO005 Pin Control	4
81018	GPIO	0	GPIO Registers	GPIO006 Pin Control	4
8101C	GPIO	0	GPIO Registers	GPIO007 Pin Control	4
81020	GPIO	0	GPIO Registers	GPIO010 Pin Control	4
81024	GPIO	0	GPIO Registers	GPIO011 Pin Control	4
81028	GPIO	0	GPIO Registers	GPIO012 Pin Control	4
8102C	GPIO	0	GPIO Registers	GPIO013 Pin Control	4
81030	GPIO	0	GPIO Registers	GPIO014 Pin Control	4
81034	GPIO	0	GPIO Registers	GPIO015 Pin Control	4
81038	GPIO	0	GPIO Registers	GPIO016 Pin Control	4
8103C	GPIO	0	GPIO Registers	GPIO017 Pin Control	4
81040	GPIO	0	GPIO Registers	GPIO020 Pin Control	4
81044	GPIO	0	GPIO Registers	GPIO021 Pin Control	4
81048	GPIO	0	GPIO Registers	GPIO022 Pin Control	4
8104C	GPIO	0	GPIO Registers	GPIO023 Pin Control	4
81050	GPIO	0	GPIO Registers	GPIO024 Pin Control	4
81054	GPIO	0	GPIO Registers	GPIO025 Pin Control	4
81058	GPIO	0	GPIO Registers	GPIO026 Pin Control	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
8105C	GPIO	0	GPIO Registers	GPIO027 Pin Control	4
81060	GPIO	0	GPIO Registers	GPIO030 Pin Control	4
81064	GPIO	0	GPIO Registers	GPIO031 Pin Control	4
81068	GPIO	0	GPIO Registers	GPIO032 Pin Control	4
8106C	GPIO	0	GPIO Registers	GPIO033 Pin Control	4
81070	GPIO	0	GPIO Registers	GPIO034 Pin Control	4
81074	GPIO	0	GPIO Registers	GPIO035 Pin Control	4
81078	GPIO	0	GPIO Registers	GPIO036 Pin Control	4
81080	GPIO	0	GPIO Registers	GPIO040 Pin Control	4
81084	GPIO	0	GPIO Registers	GPIO041 Pin Control	4
81088	GPIO	0	GPIO Registers	GPIO042 Pin Control	4
8108C	GPIO	0	GPIO Registers	GPIO043 Pin Control	4
81090	GPIO	0	GPIO Registers	GPIO044 Pin Control	4
81094	GPIO	0	GPIO Registers	GPIO045 Pin Control	4
81098	GPIO	0	GPIO Registers	GPIO046 Pin Control	4
8109C	GPIO	0	GPIO Registers	GPIO047 Pin Control	4
810A0	GPIO	0	GPIO Registers	GPIO050 Pin Control	4
810A4	GPIO	0	GPIO Registers	GPIO051 Pin Control	4
810A8	GPIO	0	GPIO Registers	GPIO052 Pin Control	4
810AC	GPIO	0	GPIO Registers	GPIO053 Pin Control	4
810B0	GPIO	0	GPIO Registers	GPIO054 Pin Control	4
810B4	GPIO	0	GPIO Registers	GPIO055 Pin Control	4
810B8	GPIO	0	GPIO Registers	GPIO056 Pin Control	4
810BC	GPIO	0	GPIO Registers	GPIO057 Pin Control	4
810C0	GPIO	0	GPIO Registers	GPIO060 Pin Control	4
810C4	GPIO	0	GPIO Registers	GPIO061 Pin Control	4
810C8	GPIO	0	GPIO Registers	GPIO062 Pin Control	4
810CC	GPIO	0	GPIO Registers	GPIO063 Pin Control	4
810D0	GPIO	0	GPIO Registers	GPIO064 Pin Control	4
810D4	GPIO	0	GPIO Registers	GPIO065 Pin Control	4
810D8	GPIO	0	GPIO Registers	GPIO066 Pin Control	4
810DC	GPIO	0	GPIO Registers	GPIO067 Pin Control	4
81100	GPIO	0	GPIO Registers	GPIO100 Pin Control	4
81104	GPIO	0	GPIO Registers	GPIO101 Pin Control	4
81108	GPIO	0	GPIO Registers	GPIO102 Pin Control	4
8110C	GPIO	0	GPIO Registers	GPIO103 Pin Control	4
81110	GPIO	0	GPIO Registers	GPIO104 Pin Control	4
81114	GPIO	0	GPIO Registers	GPIO105 Pin Control	4
81118	GPIO	0	GPIO Registers	GPIO106 Pin Control	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
8111C	GPIO	0	GPIO Registers	GPIO107 Pin Control	4
81120	GPIO	0	GPIO Registers	GPIO110 Pin Control	4
81124	GPIO	0	GPIO Registers	GPIO111 Pin Control	4
81128	GPIO	0	GPIO Registers	GPIO112 Pin Control	4
8112C	GPIO	0	GPIO Registers	GPIO113 Pin Control	4
81130	GPIO	0	GPIO Registers	GPIO114 Pin Control	4
81134	GPIO	0	GPIO Registers	GPIO115 Pin Control	4
81138	GPIO	0	GPIO Registers	GPIO116 Pin Control	4
8113C	GPIO	0	GPIO Registers	GPIO117 Pin Control	4
81140	GPIO	0	GPIO Registers	GPIO120 Pin Control	4
81144	GPIO	0	GPIO Registers	GPIO121 Pin Control	4
81148	GPIO	0	GPIO Registers	GPIO122 Pin Control	4
8114C	GPIO	0	GPIO Registers	GPIO123 Pin Control	4
81150	GPIO	0	GPIO Registers	GPIO124 Pin Control	4
81154	GPIO	0	GPIO Registers	GPIO125 Pin Control	4
81158	GPIO	0	GPIO Registers	GPIO126 Pin Control	4
8115C	GPIO	0	GPIO Registers	GPIO127 Pin Control	4
81160	GPIO	0	GPIO Registers	GPIO130 Pin Control	4
81164	GPIO	0	GPIO Registers	GPIO131 Pin Control	4
81168	GPIO	0	GPIO Registers	GPIO132 Pin Control	4
8116C	GPIO	0	GPIO Registers	GPIO133 Pin Control	4
81170	GPIO	0	GPIO Registers	GPIO134 Pin Control	4
81174	GPIO	0	GPIO Registers	GPIO135 Pin Control	4
81178	GPIO	0	GPIO Registers	GPIO136 Pin Control	4
81180	GPIO	0	GPIO Registers	GPIO140 Pin Control	4
81184	GPIO	0	GPIO Registers	GPIO141 Pin Control	4
81188	GPIO	0	GPIO Registers	GPIO142 Pin Control	4
8118C	GPIO	0	GPIO Registers	GPIO143 Pin Control	4
81190	GPIO	0	GPIO Registers	GPIO144 Pin Control	4
81194	GPIO	0	GPIO Registers	GPIO145 Pin Control	4
81198	GPIO	0	GPIO Registers	GPIO146 Pin Control	4
8119C	GPIO	0	GPIO Registers	GPIO147 Pin Control	4
811A0	GPIO	0	GPIO Registers	GPIO150 Pin Control	4
811A4	GPIO	0	GPIO Registers	GPIO151 Pin Control	4
811A8	GPIO	0	GPIO Registers	GPIO152 Pin Control	4
811AC	GPIO	0	GPIO Registers	GPIO153 Pin Control	4
811B0	GPIO	0	GPIO Registers	GPIO154 Pin Control	4
811B4	GPIO	0	GPIO Registers	GPIO155 Pin Control	4
811B8	GPIO	0	GPIO Registers	GPIO156 Pin Control	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
811BC	GPIO	0	GPIO Registers	GPIO157 Pin Control	4
811C0	GPIO	0	GPIO Registers	GPIO160 Pin Control	4
811C4	GPIO	0	GPIO Registers	GPIO161 Pin Control	4
811C8	GPIO	0	GPIO Registers	GPIO162 Pin Control	4
811CC	GPIO	0	GPIO Registers	GPIO163 Pin Control	4
811D0	GPIO	0	GPIO Registers	GPIO164 Pin Control	4
811D4	GPIO	0	GPIO Registers	GPIO165 Pin Control	4
811D8	GPIO	0	GPIO Registers	GPIO166 Pin Control	4
81280	GPIO	0	GPIO Registers	Output GPIO[000:036]	4
81284	GPIO	0	GPIO Registers	Output GPIO[040:076]	4
81288	GPIO	0	GPIO Registers	Output GPIO[100:136]	4
8128C	GPIO	0	GPIO Registers	Output GPIO[140:176]	4
81300	GPIO	0	GPIO Registers	Input GPIO[000:036]	4
81304	GPIO	0	GPIO Registers	Input GPIO[040:076]	4
81308	GPIO	0	GPIO Registers	Input GPIO[100:136]	4
8130C	GPIO	0	GPIO Registers	Input GPIO[140:176]	4
813F0	GPIO	0	GPIO Registers	GPIO Lock 3	4
813F4	GPIO	0	GPIO Registers	GPIO Lock 2	4
813F8	GPIO	0	GPIO Registers	GPIO Lock 1	4
813FC	GPIO	0	GPIO Registers	GPIO Lock 0	4
81504	GPIO	0	GPIO Registers	GPIO001 Pin Control 2	4
81508	GPIO	0	GPIO Registers	GPIO002 Pin Control 2	4
8150C	GPIO	0	GPIO Registers	GPIO003 Pin Control 2	4
81510	GPIO	0	GPIO Registers	GPIO004 Pin Control 2	4
81514	GPIO	0	GPIO Registers	GPIO005 Pin Control 2	4
81518	GPIO	0	GPIO Registers	GPIO006 Pin Control 2	4
8151C	GPIO	0	GPIO Registers	GPIO007 Pin Control 2	4
81520	GPIO	0	GPIO Registers	GPIO010 Pin Control 2	4
81524	GPIO	0	GPIO Registers	GPIO011 Pin Control 2	4
81528	GPIO	0	GPIO Registers	GPIO012 Pin Control 2	4
8152C	GPIO	0	GPIO Registers	GPIO013 Pin Control 2	4
81530	GPIO	0	GPIO Registers	GPIO014 Pin Control 2	4
81534	GPIO	0	GPIO Registers	GPIO015 Pin Control 2	4
81538	GPIO	0	GPIO Registers	GPIO016 Pin Control 2	4
8153C	GPIO	0	GPIO Registers	GPIO017 Pin Control 2	4
81540	GPIO	0	GPIO Registers	GPIO020 Pin Control 2	4
81544	GPIO	0	GPIO Registers	GPIO021 Pin Control 2	4
81548	GPIO	0	GPIO Registers	GPIO022 Pin Control 2	4
8154C	GPIO	0	GPIO Registers	GPIO023 Pin Control 2	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
81550	GPIO	0	GPIO Registers	GPIO024 Pin Control 2	4
81554	GPIO	0	GPIO Registers	GPIO025 Pin Control 2	4
81558	GPIO	0	GPIO Registers	GPIO026 Pin Control 2	4
8155C	GPIO	0	GPIO Registers	GPIO027 Pin Control 2	4
81560	GPIO	0	GPIO Registers	GPIO030 Pin Control 2	4
81564	GPIO	0	GPIO Registers	GPIO031 Pin Control 2	4
81568	GPIO	0	GPIO Registers	GPIO032 Pin Control 2	4
8156C	GPIO	0	GPIO Registers	GPIO033 Pin Control 2	4
81570	GPIO	0	GPIO Registers	GPIO034 Pin Control 2	4
81574	GPIO	0	GPIO Registers	GPIO035 Pin Control 2	4
81578	GPIO	0	GPIO Registers	GPIO036 Pin Control 2	4
81580	GPIO	0	GPIO Registers	GPIO040 Pin Control 2	4
81584	GPIO	0	GPIO Registers	GPIO041 Pin Control 2	4
81588	GPIO	0	GPIO Registers	GPIO042 Pin Control 2	4
8158C	GPIO	0	GPIO Registers	GPIO043 Pin Control 2	4
81590	GPIO	0	GPIO Registers	GPIO044 Pin Control 2	4
81594	GPIO	0	GPIO Registers	GPIO045 Pin Control 2	4
81598	GPIO	0	GPIO Registers	GPIO046 Pin Control 2	4
8159C	GPIO	0	GPIO Registers	GPIO047 Pin Control 2	4
815A0	GPIO	0	GPIO Registers	GPIO050 Pin Control 2	4
815A4	GPIO	0	GPIO Registers	GPIO051 Pin Control 2	4
815A8	GPIO	0	GPIO Registers	GPIO052 Pin Control 2	4
815AC	GPIO	0	GPIO Registers	GPIO053 Pin Control 2	4
815B0	GPIO	0	GPIO Registers	GPIO054 Pin Control 2	4
815B4	GPIO	0	GPIO Registers	GPIO055 Pin Control 2	4
815B8	GPIO	0	GPIO Registers	GPIO056 Pin Control 2	4
815BC	GPIO	0	GPIO Registers	GPIO057 Pin Control 2	4
815C0	GPIO	0	GPIO Registers	GPIO060 Pin Control 2	4
815C4	GPIO	0	GPIO Registers	GPIO061 Pin Control 2	4
815C8	GPIO	0	GPIO Registers	GPIO062 Pin Control 2	4
815CC	GPIO	0	GPIO Registers	GPIO063 Pin Control 2	4
815D0	GPIO	0	GPIO Registers	GPIO064 Pin Control 2	4
815D4	GPIO	0	GPIO Registers	GPIO065 Pin Control 2	4
815D8	GPIO	0	GPIO Registers	GPIO066 Pin Control 2	4
815DC	GPIO	0	GPIO Registers	GPIO067 Pin Control 2	4
815E0	GPIO	0	GPIO Registers	GPIO100 Pin Control 2	4
815E4	GPIO	0	GPIO Registers	GPIO101 Pin Control 2	4
815E8	GPIO	0	GPIO Registers	GPIO102 Pin Control 2	4
815EC	GPIO	0	GPIO Registers	GPIO103 Pin Control 2	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
815F0	GPIO	0	GPIO Registers	GPIO104 Pin Control 2	4
815F4	GPIO	0	GPIO Registers	GPIO105 Pin Control 2	4
815F8	GPIO	0	GPIO Registers	GPIO106 Pin Control 2	4
815FC	GPIO	0	GPIO Registers	GPIO107 Pin Control 2	4
81600	GPIO	0	GPIO Registers	GPIO110 Pin Control 2	4
81604	GPIO	0	GPIO Registers	GPIO111 Pin Control 2	4
81608	GPIO	0	GPIO Registers	GPIO112 Pin Control 2	4
8160C	GPIO	0	GPIO Registers	GPIO113 Pin Control 2	4
81610	GPIO	0	GPIO Registers	GPIO114 Pin Control 2	4
81614	GPIO	0	GPIO Registers	GPIO115 Pin Control 2	4
81618	GPIO	0	GPIO Registers	GPIO116 Pin Control 2	4
8161C	GPIO	0	GPIO Registers	GPIO117 Pin Control 2	4
81620	GPIO	0	GPIO Registers	GPIO120 Pin Control 2	4
81624	GPIO	0	GPIO Registers	GPIO121 Pin Control 2	4
81628	GPIO	0	GPIO Registers	GPIO122 Pin Control 2	4
8162C	GPIO	0	GPIO Registers	GPIO123 Pin Control 2	4
81630	GPIO	0	GPIO Registers	GPIO124 Pin Control 2	4
81634	GPIO	0	GPIO Registers	GPIO125 Pin Control 2	4
81638	GPIO	0	GPIO Registers	GPIO126 Pin Control 2	4
8163C	GPIO	0	GPIO Registers	GPIO127 Pin Control 2	4
81640	GPIO	0	GPIO Registers	GPIO130 Pin Control 2	4
81644	GPIO	0	GPIO Registers	GPIO131 Pin Control 2	4
81648	GPIO	0	GPIO Registers	GPIO132 Pin Control 2	4
8164C	GPIO	0	GPIO Registers	GPIO133 Pin Control 2	4
81650	GPIO	0	GPIO Registers	GPIO134 Pin Control 2	4
81654	GPIO	0	GPIO Registers	GPIO135 Pin Control 2	4
81658	GPIO	0	GPIO Registers	GPIO136 Pin Control 2	4
81660	GPIO	0	GPIO Registers	GPIO140 Pin Control 2	4
81664	GPIO	0	GPIO Registers	GPIO141 Pin Control 2	4
81668	GPIO	0	GPIO Registers	GPIO142 Pin Control 2	4
8166C	GPIO	0	GPIO Registers	GPIO143 Pin Control 2	4
81670	GPIO	0	GPIO Registers	GPIO144 Pin Control 2	4
81674	GPIO	0	GPIO Registers	GPIO145 Pin Control 2	4
81678	GPIO	0	GPIO Registers	GPIO146 Pin Control 2	4
8167C	GPIO	0	GPIO Registers	GPIO147 Pin Control 2	4
81680	GPIO	0	GPIO Registers	GPIO150 Pin Control 2	4
81684	GPIO	0	GPIO Registers	GPIO151 Pin Control 2	4
81688	GPIO	0	GPIO Registers	GPIO152 Pin Control 2	4
8168C	GPIO	0	GPIO Registers	GPIO153 Pin Control 2	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
81690	GPIO	0	GPIO Registers	GPIO154 Pin Control 2	4
81694	GPIO	0	GPIO Registers	GPIO155 Pin Control 2	4
81698	GPIO	0	GPIO Registers	GPIO156 Pin Control 2	4
8169C	GPIO	0	GPIO Registers	GPIO157 Pin Control 2	4
816A0	GPIO	0	GPIO Registers	GPIO160 Pin Control 2	4
816A4	GPIO	0	GPIO Registers	GPIO161 Pin Control 2	4
816A8	GPIO	0	GPIO Registers	GPIO162 Pin Control 2	4
816AC	GPIO	0	GPIO Registers	GPIO163 Pin Control 2	4
816B0	GPIO	0	GPIO Registers	GPIO164 Pin Control 2	4
816B4	GPIO	0	GPIO Registers	GPIO165 Pin Control 2	4
816D8	GPIO	0	GPIO Registers	GPIO166 Pin Control 2	4
F0000	IMAP	0	EMI_RUNTIME	EMI Host-to-EC Mailbox Register	1
F0001	IMAP	0	EMI_RUNTIME	EC-to-Host Mailbox Register	1
F0002	IMAP	0	EMI_RUNTIME	EC Address Register	2
F0004	IMAP	0	EMI_RUNTIME	EC Data Register	4
F0008	IMAP	0	EMI_RUNTIME	Interrupt Source Register	2
F000A	IMAP	0	EMI_RUNTIME	Interrupt Mask Register	2
F000C	IMAP	0	EMI_RUNTIME	Application ID Register	1
F0100	IMAP	0	EMI_EC_ONLY	EMI Host-to-EC Mailbox Register	1
F0101	IMAP	0	EMI_EC_ONLY	EC-to-Host Mailbox Register	1
F0104	IMAP	0	EMI_EC_ONLY	Memory Base Address 0 Register	4
F0108	IMAP	0	EMI_EC_ONLY	Memory Read Limit 0 Register	2
F010A	IMAP	0	EMI_EC_ONLY	Memory Write Limit 0 Register	2
F010C	IMAP	0	EMI_EC_ONLY	Memory Base Address 1 Register	4
F0110	IMAP	0	EMI_EC_ONLY	Memory Read Limit 1 Register	2
F0112	IMAP	0	EMI_EC_ONLY	Memory Write Limit 1 Register	2
F0114	IMAP	0	EMI_EC_ONLY	Interrupt Set Register	2
F0116	IMAP	0	EMI_EC_ONLY	Host Clear Enable Register	2
F0400	8042 Host Interface	0	KBC_Runtime	EC_Host Data/Aux Register (Read)	1
F0400	8042 Host Interface	0	KBC_Runtime	Host_EC Data Register (Write)	1
F0404	8042 Host Interface	0	KBC_Runtime	Keyboard Status Read Register	1
F0404	8042 Host Interface	0	KBC_Runtime	Host_EC Command Register (Write)	1
F0500	8042 Host Interface	0	KBC_EC_Only	Host_EC Data/Cmd Register	1
F0500	8042 Host Interface	0	KBC_EC_Only	EC_Host Data Register	1
F0504	8042 Host Interface	0	KBC_EC_Only	Keyboard Status Read Register	1

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
F0508	8042 Host Interface	0	KBC_EC_Only	Keyboard Control Register	1
F050C	8042 Host Interface	0	KBC_EC_Only	EC_Host Aux Register	1
F0514	8042 Host Interface	0	KBC_EC_Only	PCOBF Register	1
F0730	8042 Host Interface	0	KBC_Configuration	Activate Register	1
F0C00	ACPI EC Interface	0	ACPI_Runtime	ACPI OS Data Register Byte 0 - Read	1
F0C00	ACPI EC Interface	0	ACPI_Runtime	ACPI OS Data Register Byte 0 - Write	1
F0C01	ACPI EC Interface	0	ACPI_Runtime	ACPI OS Data Register Byte 1 - Read	1
F0C01	ACPI EC Interface	0	ACPI_Runtime	ACPI OS Data Register Byte 1 - Write	1
F0C02	ACPI EC Interface	0	ACPI_Runtime	ACPI OS Data Register Byte 2 - Read	1
F0C02	ACPI EC Interface	0	ACPI_Runtime	ACPI OS Data Register Byte 2 - Write	1
F0C03	ACPI EC Interface	0	ACPI_Runtime	ACPI OS Data Register Byte 3 - Read	1
F0C03	ACPI EC Interface	0	ACPI_Runtime	ACPI OS Data Register Byte 3 - Write	1
F0C04	ACPI EC Interface	0	ACPI_Runtime	ACPI OS Command Register	1
F0C04	ACPI EC Interface	0	ACPI_Runtime	STATUS OS-Register	1
F0C05	ACPI EC Interface	0	ACPI_Runtime	Byte Control OS-Register	1
F0D00	ACPI EC Interface	0	ACPI_EC_Only	EC2OS Data EC-Register Byte 0	1
F0D01	ACPI EC Interface	0	ACPI_EC_Only	EC2OS Data EC-Register Byte 1	1
F0D02	ACPI EC Interface	0	ACPI_EC_Only	EC2OS Data EC-Register Byte 2	1
F0D03	ACPI EC Interface	0	ACPI_EC_Only	EC2OS Data EC-Register Byte 3	1
F0D04	ACPI EC Interface	0	ACPI_EC_Only	STATUS EC-Register	1
F0D05	ACPI EC Interface	0	ACPI_EC_Only	Byte Control EC-Register	1
F0D08	ACPI EC Interface	0	ACPI_EC_Only	OS2EC Data EC-Register Byte 0 - CMD	1
F0D08	ACPI EC Interface	0	ACPI_EC_Only	OS2EC Data EC-Register Byte 0 - DATA	1
F0D09	ACPI EC Interface	0	ACPI_EC_Only	OS2EC Data EC-Register Byte 1	1
F0D0A	ACPI EC Interface	0	ACPI_EC_Only	OS2EC Data EC-Register Byte 2	1
F0D0B	ACPI EC Interface	0	ACPI_EC_Only	OS2EC Data EC-Register Byte 3	1

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
F1000	ACPI EC Interface	1	ACPI_Runtime	ACPI OS Data Register Byte 0 - Read	1
F1000	ACPI EC Interface	1	ACPI_Runtime	ACPI OS Data Register Byte 0 - Write	1
F1001	ACPI EC Interface	1	ACPI_Runtime	ACPI OS Data Register Byte 1 - Read	1
F1001	ACPI EC Interface	1	ACPI_Runtime	ACPI OS Data Register Byte 1 - Write	1
F1002	ACPI EC Interface	1	ACPI_Runtime	ACPI OS Data Register Byte 2 - Read	1
F1002	ACPI EC Interface	1	ACPI_Runtime	ACPI OS Data Register Byte 2 - Write	1
F1003	ACPI EC Interface	1	ACPI_Runtime	ACPI OS Data Register Byte 3 - Read	1
F1003	ACPI EC Interface	1	ACPI_Runtime	ACPI OS Data Register Byte 3 - Write	1
F1004	ACPI EC Interface	1	ACPI_Runtime	ACPI OS Command Register	1
F1004	ACPI EC Interface	1	ACPI_Runtime	STATUS OS-Register	1
F1005	ACPI EC Interface	1	ACPI_Runtime	Byte Control OS-Register	1
F1100	ACPI EC Interface	1	ACPI_EC_Only	EC2OS Data EC-Register Byte 0	1
F1101	ACPI EC Interface	1	ACPI_EC_Only	EC2OS Data EC-Register Byte 1	1
F1102	ACPI EC Interface	1	ACPI_EC_Only	EC2OS Data EC-Register Byte 2	1
F1103	ACPI EC Interface	1	ACPI_EC_Only	EC2OS Data EC-Register Byte 3	1
F1104	ACPI EC Interface	1	ACPI_EC_Only	STATUS EC-Register	1
F1105	ACPI EC Interface	1	ACPI_EC_Only	Byte Control EC-Register	1
F1108	ACPI EC Interface	1	ACPI_EC_Only	OS2EC Data EC-Register Byte 0 - CMD	1
F1108	ACPI EC Interface	1	ACPI_EC_Only	OS2EC Data EC-Register Byte 0 - DATA	1
F1109	ACPI EC Interface	1	ACPI_EC_Only	OS2EC Data EC-Register Byte 1	1
F110A	ACPI EC Interface	1	ACPI_EC_Only	OS2EC Data EC-Register Byte 2	1
F110B	ACPI EC Interface	1	ACPI_EC_Only	OS2EC Data EC-Register Byte 3	1
F1400	ACPI PM1	0	PM1_Runtime	PM1 Status 1	1
F1401	ACPI PM1	0	PM1_Runtime	PM1 Status 2	1
F1402	ACPI PM1	0	PM1_Runtime	PM1 Enable 1	1
F1403	ACPI PM1	0	PM1_Runtime	PM1 Enable 2	1
F1404	ACPI PM1	0	PM1_Runtime	PM1 Control 1	1

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
F1405	ACPI PM1	0	PM1_Runtime	PM1 Control 2	1
F1406	ACPI PM1	0	PM1_Runtime	PM2 Control 1	1
F1407	ACPI PM1	0	PM1_Runtime	PM2 Control 2	1
F1500	ACPI PM1	0	PM1_EC_Only	PM1 Status 1	1
F1501	ACPI PM1	0	PM1_EC_Only	PM1 Status 2	1
F1502	ACPI PM1	0	PM1_EC_Only	PM1 Enable 1	1
F1503	ACPI PM1	0	PM1_EC_Only	PM1 Enable 2	1
F1504	ACPI PM1	0	PM1_EC_Only	PM1 Control 1	1
F1505	ACPI PM1	0	PM1_EC_Only	PM1 Control 2	1
F1506	ACPI PM1	0	PM1_EC_Only	PM2 Control 1	1
F1507	ACPI PM1	0	PM1_EC_Only	PM2 Control 2	1
F1510	ACPI PM1	0	PM1_EC_Only	PM1 EC PM Status	1
F1800	8042 Host Interface	0	Legacy_Runtime	PORT92 Register	1
F1900	8042 Host Interface	0	Legacy_EC_Only	GATEA20 Control Register	1
F1908	8042 Host Interface	0	Legacy_EC_Only	SETGA20L Register	1
F190C	8042 Host Interface	0	Legacy_EC_Only	RSTGA20L Register	1
F1B30	8042 Host Interface	0	Legacy_Configuration	PORT92 Enable Register	1
F1C00	M16C550A UART	0	UART_Runtime	UART Receive Buffer Register	1
F1C00	M16C550A UART	0	UART_Runtime	UART Transmit Buffer Register	1
F1C00	M16C550A UART	0	UART_Runtime	UART Programmable BAUD Rate Generator (LSB) Register	1
F1C01	M16C550A UART	0	UART_Runtime	UART Interrupt Enable Register	1
F1C01	M16C550A UART	0	UART_Runtime	UART Programmable BAUD Rate Generator (MSB) Register	1
F1C02	M16C550A UART	0	UART_Runtime	UART FIFO Control Register	1
F1C02	M16C550A UART	0	UART_Runtime	UART Interrupt Identification Register	1
F1C03	M16C550A UART	0	UART_Runtime	UART Line Control Register	1
F1C04	M16C550A UART	0	UART_Runtime	UART Modem Control Register	1
F1C05	M16C550A UART	0	UART_Runtime	UART Line Status Register	1
F1C06	M16C550A UART	0	UART_Runtime	UART Modem Status Register	1
F1C07	M16C550A UART	0	UART_Runtime	UART Scratchpad Register	1
F1F30	M16C550A UART	0	UART_Config	UART Activate Register	1
F1FF0	M16C550A UART	0	UART_Config	UART Config Select Register	1
F2400	Mailbox Registers Interface	0	MBX_Runtime	MBX_Index Register	1
F2401	Mailbox Registers Interface	0	MBX_Runtime	MBX_Data_Register	1
F2500	Mailbox Registers Interface	0	MBX_EC_Only (140x)	HOST-to-EC Mailbox Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
F2504	Mailbox Registers Interface	0	MBX_EC_Only (140x)	EC-to-Host Mailbox Register	4
F2508	Mailbox Registers Interface	0	MBX_EC_Only (140x)	SMI Interrupt Source Register	4
F250C	Mailbox Registers Interface	0	MBX_EC_Only (140x)	SMI Interrupt Mask Register	4
F2510	Mailbox Registers Interface	0	MBX_EC_Only (140x)	Mailbox Register [3:0]	4
F2514	Mailbox Registers Interface	0	MBX_EC_Only (140x)	Mailbox Register [7:4]	4
F2518	Mailbox Registers Interface	0	MBX_EC_Only (140x)	Mailbox Register [B:8]	4
F251C	Mailbox Registers Interface	0	MBX_EC_Only (140x)	Mailbox Register [F:C]	4
F2520	Mailbox Registers Interface	0	MBX_EC_Only (140x)	Mailbox Register [13:10]	4
F2524	Mailbox Registers Interface	0	MBX_EC_Only (140x)	Mailbox Register [17:14]	4
F2528	Mailbox Registers Interface	0	MBX_EC_Only (140x)	Mailbox Register [1B:18]	4
F252C	Mailbox Registers Interface	0	MBX_EC_Only (140x)	Mailbox Register [1F:1C]	4
F2800	ACPI EC Interface	2	ACPI_Runtime	ACPI OS Data Register Byte 0 - Read	1
F2800	ACPI EC Interface	2	ACPI_Runtime	ACPI OS Data Register Byte 0 - Write	1
F2801	ACPI EC Interface	2	ACPI_Runtime	ACPI OS Data Register Byte 1 - Read	1
F2801	ACPI EC Interface	2	ACPI_Runtime	ACPI OS Data Register Byte 1 - Write	1
F2802	ACPI EC Interface	2	ACPI_Runtime	ACPI OS Data Register Byte 2 - Read	1
F2802	ACPI EC Interface	2	ACPI_Runtime	ACPI OS Data Register Byte 2 - Write	1
F2803	ACPI EC Interface	2	ACPI_Runtime	ACPI OS Data Register Byte 3 - Read	1
F2803	ACPI EC Interface	2	ACPI_Runtime	ACPI OS Data Register Byte 3 - Write	1
F2804	ACPI EC Interface	2	ACPI_Runtime	ACPI OS Command Register	1
F2804	ACPI EC Interface	2	ACPI_Runtime	STATUS OS-Register	1
F2805	ACPI EC Interface	2	ACPI_Runtime	Byte Control OS-Register	1
F2900	ACPI EC Interface	2	ACPI_EC_Only	EC2OS Data EC-Register Byte 0	1
F2901	ACPI EC Interface	2	ACPI_EC_Only	EC2OS Data EC-Register Byte 1	1

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
F2902	ACPI EC Interface	2	ACPI_EC_Only	EC2OS Data EC-Register Byte 2	1
F2903	ACPI EC Interface	2	ACPI_EC_Only	EC2OS Data EC-Register Byte 3	1
F2904	ACPI EC Interface	2	ACPI_EC_Only	STATUS EC-Register	1
F2905	ACPI EC Interface	2	ACPI_EC_Only	Byte Control EC-Register	1
F2908	ACPI EC Interface	2	ACPI_EC_Only	OS2EC Data EC-Register Byte 0 - CMD	1
F2908	ACPI EC Interface	2	ACPI_EC_Only	OS2EC Data EC-Register Byte 0 - DATA	1
F2909	ACPI EC Interface	2	ACPI_EC_Only	OS2EC Data EC-Register Byte 1	1
F290A	ACPI EC Interface	2	ACPI_EC_Only	OS2EC Data EC-Register Byte 2	1
F290B	ACPI EC Interface	2	ACPI_EC_Only	OS2EC Data EC-Register Byte 3	1
F2C00	ACPI EC Interface	3	ACPI_Runtime	ACPI OS Data Register Byte 0 - Read	1
F2C00	ACPI EC Interface	3	ACPI_Runtime	ACPI OS Data Register Byte 0 - Write	1
F2C01	ACPI EC Interface	3	ACPI_Runtime	ACPI OS Data Register Byte 1 - Read	1
F2C01	ACPI EC Interface	3	ACPI_Runtime	ACPI OS Data Register Byte 1 - Write	1
F2C02	ACPI EC Interface	3	ACPI_Runtime	ACPI OS Data Register Byte 2 - Read	1
F2C02	ACPI EC Interface	3	ACPI_Runtime	ACPI OS Data Register Byte 2 - Write	1
F2C03	ACPI EC Interface	3	ACPI_Runtime	ACPI OS Data Register Byte 3 - Read	1
F2C03	ACPI EC Interface	3	ACPI_Runtime	ACPI OS Data Register Byte 3 - Write	1
F2C04	ACPI EC Interface	3	ACPI_Runtime	ACPI OS Command Register	1
F2C04	ACPI EC Interface	3	ACPI_Runtime	STATUS OS-Register	1
F2C05	ACPI EC Interface	3	ACPI_Runtime	Byte Control OS-Register	1
F2D00	ACPI EC Interface	3	ACPI_EC_Only	EC2OS Data EC-Register Byte 0	1
F2D01	ACPI EC Interface	3	ACPI_EC_Only	EC2OS Data EC-Register Byte 1	1
F2D02	ACPI EC Interface	3	ACPI_EC_Only	EC2OS Data EC-Register Byte 2	1
F2D03	ACPI EC Interface	3	ACPI_EC_Only	EC2OS Data EC-Register Byte 3	1
F2D04	ACPI EC Interface	3	ACPI_EC_Only	STATUS EC-Register	1

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
F2D05	ACPI EC Interface	3	ACPI_EC_Only	Byte Control EC-Register	1
F2D08	ACPI EC Interface	3	ACPI_EC_Only	OS2EC Data EC-Register Byte 0 - CMD	1
F2D08	ACPI EC Interface	3	ACPI_EC_Only	OS2EC Data EC-Register Byte 0 - DATA	1
F2D09	ACPI EC Interface	3	ACPI_EC_Only	OS2EC Data EC-Register Byte 1	1
F2D0A	ACPI EC Interface	3	ACPI_EC_Only	OS2EC Data EC-Register Byte 2	1
F2D0B	ACPI EC Interface	3	ACPI_EC_Only	OS2EC Data EC-Register Byte 3	1
F3000	LPC	0	LPC_Runtime	Configuration Port Index Register	1
F3001	LPC	0	LPC_Runtime	Configuration Port Data Register	1
F3100	LPC	0	LPC_EC_Only	Reserved	4
F3104	LPC	0	LPC_EC_Only	LPC Bus Monitor Register	4
F3108	LPC	0	LPC_EC_Only	Host Bus Error Register	4
F310C	LPC	0	LPC_EC_Only	EC SERIRQ Register	4
F3110	LPC	0	LPC_EC_Only	EC Clock Control Register	4
F3120	LPC	0	LPC_EC_Only	BAR Inhibit Register	4
F3130	LPC	0	LPC_EC_Only	LPC BAR Init Register	2
F3140	LPC	0	LPC_EC_Only	Memory BAR Inhibit	8
F31FC	LPC	0	LPC_EC_Only	Memory Host Configuration Register	4
F3330	LPC	0	LPC_Config (MEC140x)	LPC Activate	1
F3340	LPC	0	LPC_Config (MEC140x)	SIRQ0 Interrupt Configuration Register	1
F3341	LPC	0	LPC_Config (MEC140x)	SIRQ1 Interrupt Configuration Register	1
F3342	LPC	0	LPC_Config (MEC140x)	SIRQ2 Interrupt Configuration Register	1
F3343	LPC	0	LPC_Config (MEC140x)	SIRQ3 Interrupt Configuration Register	1
F3344	LPC	0	LPC_Config (MEC140x)	SIRQ4 Interrupt Configuration Register	1
F3345	LPC	0	LPC_Config (MEC140x)	SIRQ5 Interrupt Configuration Register	1
F3346	LPC	0	LPC_Config (MEC140x)	SIRQ6 Interrupt Configuration Register	1
F3347	LPC	0	LPC_Config (MEC140x)	SIRQ7 Interrupt Configuration Register	1
F3348	LPC	0	LPC_Config (MEC140x)	SIRQ8 Interrupt Configuration Register	1

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
F3349	LPC	0	LPC_Config (MEC140x)	SIRQ9 Interrupt Configuration Register	1
F334A	LPC	0	LPC_Config (MEC140x)	SIRQ10 Interrupt Configuration Register	1
F334B	LPC	0	LPC_Config (MEC140x)	SIRQ11 Interrupt Configuration Register	1
F334C	LPC	0	LPC_Config (MEC140x)	SIRQ12 Interrupt Configuration Register	1
F334D	LPC	0	LPC_Config (MEC140x)	SIRQ13 Interrupt Configuration Register	1
F334E	LPC	0	LPC_Config (MEC140x)	SIRQ14 Interrupt Configuration Register	1
F334F	LPC	0	LPC_Config (MEC140x)	SIRQ15 Interrupt Configuration Register	1
F3360	LPC	0	LPC_Config (MEC140x)	LPC Interface BAR Register	4
F3364	LPC	0	LPC_Config (MEC140x)	EM Interface 0 BAR	4
F3368	LPC	0	LPC_Config (MEC140x)	Keyboard Controller BAR	4
F336C	LPC	0	LPC_Config (MEC140x)	ACPI EC Interface 0 BAR	4
F3370	LPC	0	LPC_Config (MEC140x)	ACPI EC Interface 1 BAR	4
F3374	LPC	0	LPC_Config (MEC140x)	ACPI PM1 Interface BAR	4
F3378	LPC	0	LPC_Config (MEC140x)	Legacy (GATEA20) Interface BAR	4
F337C	LPC	0	LPC_Config (MEC140x)	UART 0 BAR Register	4
F3380	LPC	0	LPC_Config (MEC140x)	Mailbox Registers Interface BAR	4
F3384	LPC	0	LPC_Config (MEC140x)	ACPI EC Interface 2 BAR	4
F3388	LPC	0	LPC_Config (MEC140x)	ACPI EC Interface 3 BAR	4
F338C	LPC	0	LPC_Config (MEC140x)	Port 80 BIOS Debug Port 0 CONFIG BAR	4
F3390	LPC	0	LPC_Config (MEC140x)	Port 80 BIOS Debug Port 1 CONFIG BAR	4
F33A0	LPC	0	LPC_Config (MEC140x)	SRAM Memory BAR	4
F33A4	LPC	0	LPC_Config (MEC140x)	SRAM Memory BAR Configuration	4
F33C0	LPC	0	LPC_Config (MEC140x)	EM Interface 0 Memory BAR	6
F33C6	LPC	0	LPC_Config (MEC140x)	ACPI EC Interface 0 Memory BAR	6
F33CC	LPC	0	LPC_Config (MEC140x)	ACPI EC Interface 1 Memory BAR	6
F33D2	LPC	0	LPC_Config (MEC140x)	Mailbox Registers I/F Memory BAR	6
F33D8	LPC	0	LPC_Config (MEC140x)	ACPI EC2 Memory BAR (Internal Component)	16

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
F33DE	LPC	0	LPC_Config (MEC140x)	ACPI EC3 Memory BAR (Internal Component)	16
F5400	BIOS Debug Port	0	BDP_Runtime	Host Data Register	1
F5500	BIOS Debug Port	0	BDP_EC_Only	EC Data Register	1
F5504	BIOS Debug Port	0	BDP_EC_Only	Configuration Register	4
F5508	BIOS Debug Port	0	BDP_EC_Only	Status Register	4
F550C	BIOS Debug Port	0	BDP_EC_Only	Count Register	4
F5730	BIOS Debug Port	0	BDP_Configuration	Activate Register	1
F5800	BIOS Debug Port	1	BDP_Runtime	Host Data Register	1
F5900	BIOS Debug Port	1	BDP_EC_Only	EC Data Register	1
F5904	BIOS Debug Port	1	BDP_EC_Only	Configuration Register	4
F5908	BIOS Debug Port	1	BDP_EC_Only	Status Register	4
F590C	BIOS Debug Port	1	BDP_EC_Only	Count Register	4
F5B30	BIOS Debug Port	1	BDP_Configuration	Activate Register	1
FFF00	Global Configuration Registers	0	GCR	GCR Reserved Registers	7
FFF07	Global Configuration Registers	0	GCR	Logical Device Number Register	1
FFF1C	Global Configuration Registers	0	GCR	Device Revision	1
FFF1D	Global Configuration Registers	0	GCR	Device Sub ID	1
FFF1E	Global Configuration Registers	0	GCR	Device ID[7:0]	1
FFF1F	Global Configuration Registers	0	GCR	Device ID[15:8]	1
FFF22	Global Configuration Registers	0	GCR	GCR Reserved	2
FFF24	Global Configuration Registers	0	GCR	Device Mode	1
FFF25	Global Configuration Registers	0	GCR	GCR Reserved	3
FFF2A	Global Configuration Registers	0	GCR	GCR Reserved Registers	2
1FFFC000	JTVIC	0	JTVIC Registers	GIRQ8 Source Register	4
1FFFC004	JTVIC	0	JTVIC Registers	GIRQ8 Enable Set Register	4
1FFFC008	JTVIC	0	JTVIC Registers	GIRQ8 Enable Clear Register	4
1FFFC00C	JTVIC	0	JTVIC Registers	GIRQ8 Result Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
1FFFC 010	JTVIC	0	JTVIC Registers	GIRQ9 Source Register	4
1FFFC 014	JTVIC	0	JTVIC Registers	GIRQ9 Enable Set Register	4
1FFFC 018	JTVIC	0	JTVIC Registers	GIRQ9 Enable Clear Register	4
1FFFC 01C	JTVIC	0	JTVIC Registers	GIRQ9 Result Register	4
1FFFC 020	JTVIC	0	JTVIC Registers	GIRQ10 Source Register	4
1FFFC 024	JTVIC	0	JTVIC Registers	GIRQ10 Enable Set Register	4
1FFFC 028	JTVIC	0	JTVIC Registers	GIRQ10 Enable Clear Register	4
1FFFC 02C	JTVIC	0	JTVIC Registers	GIRQ10 Result Register	4
1FFFC 030	JTVIC	0	JTVIC Registers	GIRQ11 Source Register	4
1FFFC 034	JTVIC	0	JTVIC Registers	GIRQ11 Enable Set Register	4
1FFFC 038	JTVIC	0	JTVIC Registers	GIRQ11 Enable Clear Register	4
1FFFC 03C	JTVIC	0	JTVIC Registers	GIRQ11 Result Register	4
1FFFC 040	JTVIC	0	JTVIC Registers	GIRQ12 Source Register	4
1FFFC 044	JTVIC	0	JTVIC Registers	GIRQ12 Enable Set Register	4
1FFFC 048	JTVIC	0	JTVIC Registers	GIRQ12 Enable Clear Register	4
1FFFC 04C	JTVIC	0	JTVIC Registers	GIRQ12 Result Register	4
1FFFC 050	JTVIC	0	JTVIC Registers	GIRQ13 Source Register	4
1FFFC 054	JTVIC	0	JTVIC Registers	GIRQ13 Enable Set Register	4
1FFFC 058	JTVIC	0	JTVIC Registers	GIRQ13 Enable Clear Register	4
1FFFC 05C	JTVIC	0	JTVIC Registers	GIRQ14 Result Register	4
1FFFC 05C	JTVIC	0	JTVIC Registers	GIRQ13 Result Register	4
1FFFC 060	JTVIC	0	JTVIC Registers	GIRQ14 Source Register	4
1FFFC 064	JTVIC	0	JTVIC Registers	GIRQ14 Enable Set Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
1FFFC 068	JTVIC	0	JTVIC Registers	GIRQ14 Enable Clear Register	4
1FFFC 070	JTVIC	0	JTVIC Registers	GIRQ15 Source Register	4
1FFFC 074	JTVIC	0	JTVIC Registers	GIRQ15 Enable Set Register	4
1FFFC 078	JTVIC	0	JTVIC Registers	GIRQ15 Enable Clear Register	4
1FFFC 07C	JTVIC	0	JTVIC Registers	GIRQ15 Result Register	4
1FFFC 080	JTVIC	0	JTVIC Registers	GIRQ16 Source Register	4
1FFFC 084	JTVIC	0	JTVIC Registers	GIRQ16 Enable Set Register	4
1FFFC 088	JTVIC	0	JTVIC Registers	GIRQ16 Enable Clear Register	4
1FFFC 08C	JTVIC	0	JTVIC Registers	GIRQ16 Result Register	4
1FFFC 090	JTVIC	0	JTVIC Registers	GIRQ17 Source Register	4
1FFFC 094	JTVIC	0	JTVIC Registers	GIRQ17 Enable Set Register	4
1FFFC 098	JTVIC	0	JTVIC Registers	GIRQ17 Enable Clear Register	4
1FFFC 09C	JTVIC	0	JTVIC Registers	GIRQ17 Result Register	4
1FFFC 0A0	JTVIC	0	JTVIC Registers	GIRQ18 Source Register	4
1FFFC 0A4	JTVIC	0	JTVIC Registers	GIRQ18 Enable Set Register	4
1FFFC 0A8	JTVIC	0	JTVIC Registers	GIRQ18 Enable Clear Register	4
1FFFC 0AC	JTVIC	0	JTVIC Registers	GIRQ18 Result Register	4
1FFFC 0B0	JTVIC	0	JTVIC Registers	GIRQ19 Source Register	4
1FFFC 0B4	JTVIC	0	JTVIC Registers	GIRQ19 Enable Set Register	4
1FFFC 0B8	JTVIC	0	JTVIC Registers	GIRQ19 Enable Clear Register	4
1FFFC 0BC	JTVIC	0	JTVIC Registers	GIRQ19 Result Register	4
1FFFC 0C0	JTVIC	0	JTVIC Registers	GIRQ20 Source Register	4
1FFFC 0C4	JTVIC	0	JTVIC Registers	GIRQ20 Enable Set Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
1FFFC 0C8	JTVIC	0	JTVIC Registers	GIRQ20 Enable Clear Register	4
1FFFC 0CC	JTVIC	0	JTVIC Registers	GIRQ20 Result Register	4
1FFFC 0D0	JTVIC	0	JTVIC Registers	GIRQ21 Source Register	4
1FFFC 0D4	JTVIC	0	JTVIC Registers	GIRQ21 Enable Set Register	4
1FFFC 0D8	JTVIC	0	JTVIC Registers	GIRQ21 Enable Clear Register	4
1FFFC 0DC	JTVIC	0	JTVIC Registers	GIRQ21 Result Register	4
1FFFC 0E0	JTVIC	0	JTVIC Registers	GIRQ22 Source Register	4
1FFFC 0E4	JTVIC	0	JTVIC Registers	GIRQ22 Enable Set Register	4
1FFFC 0E8	JTVIC	0	JTVIC Registers	GIRQ22 Enable Clear Register	4
1FFFC 0EC	JTVIC	0	JTVIC Registers	GIRQ22 Result Register	4
1FFFC 0F0	JTVIC	0	JTVIC Registers	GIRQ23 Source Register	4
1FFFC 0F4	JTVIC	0	JTVIC Registers	GIRQ23 Enable Set Register	4
1FFFC 0F8	JTVIC	0	JTVIC Registers	GIRQ23 Enable Clear Register	4
1FFFC 0FC	JTVIC	0	JTVIC Registers	GIRQ23 Result Register	4
1FFFC 100	JTVIC	0	JTVIC Registers	GIRQ24 Source Register	4
1FFFC 104	JTVIC	0	JTVIC Registers	GIRQ24 Enable Set Register	4
1FFFC 108	JTVIC	0	JTVIC Registers	GIRQ24 Enable Clear Register	4
1FFFC 10C	JTVIC	0	JTVIC Registers	GIRQ24 Result Register	4
1FFFC 110	JTVIC	0	JTVIC Registers	GIRQ25 Source Register	4
1FFFC 114	JTVIC	0	JTVIC Registers	GIRQ25 Enable Set Register	4
1FFFC 118	JTVIC	0	JTVIC Registers	GIRQ25 Enable Clear Register	4
1FFFC 11C	JTVIC	0	JTVIC Registers	GIRQ25 Result Register	4
1FFFC 120	JTVIC	0	JTVIC Registers	GIRQ26 Source Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
1FFFC 124	JTVIC	0	JTVIC Registers	GIRQ26 Enable Set Register	4
1FFFC 128	JTVIC	0	JTVIC Registers	GIRQ26 Enable Clear Register	4
1FFFC 12C	JTVIC	0	JTVIC Registers	GIRQ26 Result Register	4
1FFFC 200	JTVIC	0	JTVIC Registers	GIRQ8 Aggregator Control Register	4
1FFFC 204	JTVIC	0	JTVIC Registers	GIRQ9 Aggregator Control Register	4
1FFFC 208	JTVIC	0	JTVIC Registers	GIRQ10 Aggregator Control Register	4
1FFFC 20C	JTVIC	0	JTVIC Registers	GIRQ11 Aggregator Control Register	4
1FFFC 210	JTVIC	0	JTVIC Registers	GIRQ12 Aggregator Control Register	4
1FFFC 214	JTVIC	0	JTVIC Registers	GIRQ13 Aggregator Control Register	4
1FFFC 218	JTVIC	0	JTVIC Registers	GIRQ14 Aggregator Control Register	4
1FFFC 21C	JTVIC	0	JTVIC Registers	GIRQ15 Aggregator Control Register	4
1FFFC 220	JTVIC	0	JTVIC Registers	GIRQ16 Aggregator Control Register	4
1FFFC 224	JTVIC	0	JTVIC Registers	GIRQ17 Aggregator Control Register	4
1FFFC 228	JTVIC	0	JTVIC Registers	GIRQ18 Aggregator Control Register	4
1FFFC 22C	JTVIC	0	JTVIC Registers	GIRQ19 Aggregator Control Register	4
1FFFC 230	JTVIC	0	JTVIC Registers	GIRQ20 Aggregator Control Register	4
1FFFC 234	JTVIC	0	JTVIC Registers	GIRQ21 Aggregator Control Register	4
1FFFC 238	JTVIC	0	JTVIC Registers	GIRQ22 Aggregator Control Register	4
1FFFC 23C	JTVIC	0	JTVIC Registers	GIRQ23 Aggregator Control Register	4
1FFFC 240	JTVIC	0	JTVIC Registers	GIRQ24 Aggregator Control Register	4
1FFFC 244	JTVIC	0	JTVIC Registers	GIRQ25 Aggregator Control Register	4
1FFFC 248	JTVIC	0	JTVIC Registers	GIRQ26 Aggregator Control Register	4
1FFFC 300	JTVIC	0	JTVIC Registers	GIRQ8 [7:0] Interrupt Priority Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
1FFFC 304	JTVIC	0	JTVIC Registers	GIRQ8 [15:8] Interrupt Priority Register	4
1FFFC 308	JTVIC	0	JTVIC Registers	GIRQ8 [23:16] Interrupt Priority Register	4
1FFFC 30C	JTVIC	0	JTVIC Registers	GIRQ8 [31:24] Interrupt Priority Register	4
1FFFC 310	JTVIC	0	JTVIC Registers	GIRQ9 [7:0] Interrupt Priority Register	4
1FFFC 314	JTVIC	0	JTVIC Registers	GIRQ9 [15:8] Interrupt Priority Register	4
1FFFC 318	JTVIC	0	JTVIC Registers	GIRQ9 [23:16] Interrupt Priority Register	4
1FFFC 31C	JTVIC	0	JTVIC Registers	GIRQ9 [31:24] Interrupt Priority Register	4
1FFFC 320	JTVIC	0	JTVIC Registers	GIRQ10 [7:0] Interrupt Priority Register	4
1FFFC 324	JTVIC	0	JTVIC Registers	GIRQ10 [15:8] Interrupt Priority Register	4
1FFFC 328	JTVIC	0	JTVIC Registers	GIRQ10 [23:16] Interrupt Priority Register	4
1FFFC 32C	JTVIC	0	JTVIC Registers	GIRQ10 [31:24] Interrupt Priority Register	4
1FFFC 330	JTVIC	0	JTVIC Registers	GIRQ11 [7:0] Interrupt Priority Register	4
1FFFC 334	JTVIC	0	JTVIC Registers	GIRQ11 [15:8] Interrupt Priority Register	4
1FFFC 338	JTVIC	0	JTVIC Registers	GIRQ11 [23:16] Interrupt Priority Register	4
1FFFC 33C	JTVIC	0	JTVIC Registers	GIRQ11 [31:24] Interrupt Priority Register	4
1FFFC 340	JTVIC	0	JTVIC Registers	GIRQ12 [7:0] Interrupt Priority Register	4
1FFFC 344	JTVIC	0	JTVIC Registers	GIRQ12 [15:8] Interrupt Priority Register	4
1FFFC 348	JTVIC	0	JTVIC Registers	GIRQ12 [23:16] Interrupt Priority Register	4
1FFFC 34C	JTVIC	0	JTVIC Registers	GIRQ12 [31:24] Interrupt Priority Register	4
1FFFC 350	JTVIC	0	JTVIC Registers	GIRQ13 [7:0] Interrupt Priority Register	4
1FFFC 354	JTVIC	0	JTVIC Registers	GIRQ13 [15:8] Interrupt Priority Register	4
1FFFC 358	JTVIC	0	JTVIC Registers	GIRQ13 [23:16] Interrupt Priority Register	4
1FFFC 35C	JTVIC	0	JTVIC Registers	GIRQ13 [31:24] Interrupt Priority Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
1FFFC 360	JTVIC	0	JTVIC Registers	GIRQ14 [7:0] Interrupt Priority Register	4
1FFFC 364	JTVIC	0	JTVIC Registers	GIRQ14 [15:8] Interrupt Priority Register	4
1FFFC 368	JTVIC	0	JTVIC Registers	GIRQ14 [23:16] Interrupt Priority Register	4
1FFFC 36C	JTVIC	0	JTVIC Registers	GIRQ14 [31:24] Interrupt Priority Register	4
1FFFC 370	JTVIC	0	JTVIC Registers	GIRQ15 [7:0] Interrupt Priority Register	4
1FFFC 374	JTVIC	0	JTVIC Registers	GIRQ15 [15:8] Interrupt Priority Register	4
1FFFC 378	JTVIC	0	JTVIC Registers	GIRQ15 [23:16] Interrupt Priority Register	4
1FFFC 37C	JTVIC	0	JTVIC Registers	GIRQ15 [31:24] Interrupt Priority Register	4
1FFFC 380	JTVIC	0	JTVIC Registers	GIRQ16 [7:0] Interrupt Priority Register	4
1FFFC 384	JTVIC	0	JTVIC Registers	GIRQ16 [15:8] Interrupt Priority Register	4
1FFFC 388	JTVIC	0	JTVIC Registers	GIRQ16 [23:16] Interrupt Priority Register	4
1FFFC 38C	JTVIC	0	JTVIC Registers	GIRQ16 [31:24] Interrupt Priority Register	4
1FFFC 390	JTVIC	0	JTVIC Registers	GIRQ17 [7:0] Interrupt Priority Register	4
1FFFC 394	JTVIC	0	JTVIC Registers	GIRQ17 [15:8] Interrupt Priority Register	4
1FFFC 398	JTVIC	0	JTVIC Registers	GIRQ17 [23:16] Interrupt Priority Register	4
1FFFC 39C	JTVIC	0	JTVIC Registers	GIRQ17 [31:24] Interrupt Priority Register	4
1FFFC 3A0	JTVIC	0	JTVIC Registers	GIRQ18 [7:0] Interrupt Priority Register	4
1FFFC 3A4	JTVIC	0	JTVIC Registers	GIRQ18 [15:8] Interrupt Priority Register	4
1FFFC 3A8	JTVIC	0	JTVIC Registers	GIRQ18 [23:16] Interrupt Priority Register	4
1FFFC 3AC	JTVIC	0	JTVIC Registers	GIRQ18 [31:24] Interrupt Priority Register	4
1FFFC 3B0	JTVIC	0	JTVIC Registers	GIRQ19 [7:0] Interrupt Priority Register	4
1FFFC 3B4	JTVIC	0	JTVIC Registers	GIRQ19 [15:8] Interrupt Priority Register	4
1FFFC 3B8	JTVIC	0	JTVIC Registers	GIRQ19 [23:16] Interrupt Priority Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
1FFFC 3BC	JTVIC	0	JTVIC Registers	GIRQ19 [31:24] Interrupt Priority Register	4
1FFFC 3C0	JTVIC	0	JTVIC Registers	GIRQ20 [7:0] Interrupt Priority Register	4
1FFFC 3C4	JTVIC	0	JTVIC Registers	GIRQ20 [15:8] Interrupt Priority Register	4
1FFFC 3C8	JTVIC	0	JTVIC Registers	GIRQ20 [23:16] Interrupt Priority Register	4
1FFFC 3CC	JTVIC	0	JTVIC Registers	GIRQ20 [31:24] Interrupt Priority Register	4
1FFFC 3D0	JTVIC	0	JTVIC Registers	GIRQ21 [7:0] Interrupt Priority Register	4
1FFFC 3D4	JTVIC	0	JTVIC Registers	GIRQ21 [15:8] Interrupt Priority Register	4
1FFFC 3D8	JTVIC	0	JTVIC Registers	GIRQ21 [23:16] Interrupt Priority Register	4
1FFFC 3DC	JTVIC	0	JTVIC Registers	GIRQ21 [31:24] Interrupt Priority Register	4
1FFFC 3E0	JTVIC	0	JTVIC Registers	GIRQ22 [7:0] Interrupt Priority Register	4
1FFFC 3E4	JTVIC	0	JTVIC Registers	GIRQ22 [15:8] Interrupt Priority Register	4
1FFFC 3E8	JTVIC	0	JTVIC Registers	GIRQ22 [23:16] Interrupt Priority Register	4
1FFFC 3EC	JTVIC	0	JTVIC Registers	GIRQ22 [31:24] Interrupt Priority Register	4
1FFFC 3F0	JTVIC	0	JTVIC Registers	GIRQ23 [7:0] Interrupt Priority Register	4
1FFFC 3F4	JTVIC	0	JTVIC Registers	GIRQ23 [15:8] Interrupt Priority Register	4
1FFFC 3F8	JTVIC	0	JTVIC Registers	GIRQ23 [23:16] Interrupt Priority Register	4
1FFFC 3FC	JTVIC	0	JTVIC Registers	GIRQ23 [31:24] Interrupt Priority Register	4
1FFFC 400	JTVIC	0	JTVIC Registers	GIRQ24 [7:0] Interrupt Priority Register	4
1FFFC 404	JTVIC	0	JTVIC Registers	GIRQ24 [15:8] Interrupt Priority Register	4
1FFFC 408	JTVIC	0	JTVIC Registers	GIRQ24 [23:16] Interrupt Priority Register	4
1FFFC 40C	JTVIC	0	JTVIC Registers	GIRQ24 [31:24] Interrupt Priority Register	4
1FFFC 410	JTVIC	0	JTVIC Registers	GIRQ25 [7:0] Interrupt Priority Register	4
1FFFC 414	JTVIC	0	JTVIC Registers	GIRQ25 [15:8] Interrupt Priority Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
1FFFC 418	JTVIC	0	JTVIC Registers	GIRQ25 [23:16] Interrupt Priority Register	4
1FFFC 41C	JTVIC	0	JTVIC Registers	GIRQ25 [31:24] Interrupt Priority Register	4
1FFFC 420	JTVIC	0	JTVIC Registers	GIRQ26 [7:0] Interrupt Priority Register	4
1FFFC 424	JTVIC	0	JTVIC Registers	GIRQ26 [15:8] Interrupt Priority Register	4
1FFFC 428	JTVIC	0	JTVIC Registers	GIRQ26 [23:16] Interrupt Priority Register	4
1FFFC 42C	JTVIC	0	JTVIC Registers	GIRQ26 [31:24] Interrupt Priority Register	4
1FFFC 500	JTVIC	0	JTVIC Registers	JTVIC Control Register	4
1FFFC 504	JTVIC	0	JTVIC Registers	Interrupt Pending Register	4
1FFFC 508	JTVIC	0	JTVIC Registers	Aggregated Group Enable Set Register	4
1FFFC 50C	JTVIC	0	JTVIC Registers	Aggregated Group Enabled Clear Register	4
1FFFC 510	JTVIC	0	JTVIC Registers	GIRQ Active Register	4

APPENDIX A: DATA SHEET REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision Level	Section/Figure/Entry	Correction
DS00001956E (10-05-16)	Section 2.3, "Notes for Tables in this Chapter"	Added Note 18.
	Section 3.9.12, "Host Reset Enable Register (HOST_RST_EN)"	Updated to remove unimplemented bits.
	Section 14.12.1, "ACPI OS Data Register Byte 0 Register"	Added Note.
	Section 33.9, "Description"	Added Note.
	Table 42-1, "Absolute Maximum Thermal Ratings" Table 42-6, "Pin Leakage" Table 42-7, "Backdrive Protection" Table 42-12, "Thermal Operating Conditions" Table 42-14, "VTR Supply Current, I_VTR" "Product Identification System"	Updates for Industrial temperature
DS00001956D (02-11-16)	Table 3-3, “Power Good Signal Definitions,” on page 66	Updated VTRGD power good signal description to remove the VTR_33_18 power supply as a source.
DS00001956C (02-09-16)	Mailbox Interface	Updated host index of mailbox registers.
	Section 2.3, "Notes for Tables in this Chapter"	Note 17 added.
	Section 1.2, "Initialize Host Interface" and Section 1.4.2, "eSPI Host System Block Diagram"	Updated description to include eSPI Flash Channel. Updated block digram to show SPI interface as optional.
	Section 1.2.2, "Configure eSPI Interface"	Updated GPIO numbers of eSPI pins.
	Section 2.3, "Notes for Tables in this Chapter"	Added note that the GPIO123/SHD_CS# pin must be used as RSMRST# if the eSPI Flash Channel is used for booting.
	Section 2.4.2, "MEC141X Pin List" and Table 2-3, "MEC141X Pin Multiplexing"	Added BSS_STRAP on the GPIO123/SHD_CS# pin
	Section 2.12.1, "Boot Source Select Straps"	Renamed and updated Crisis Recovery Strap section to include description of the BSS_STRAP on the GPIO123/SHD_CS# pin and a note that the GPIO123/SHD_CS# pin must be used as RSMRST# if the eSPI Flash Channel is used for booting.

TABLE A-1: REVISION HISTORY (CONTINUED)

Revision Level	Section/Figure/Entry	Correction
DS00001956B (06-19-15)	Table 42-12, "Thermal Operating Conditions"	TJ max value changed from “+75” to “+125”
	Section 2.4.1, "MEC140x Pin List" and Section 2.4.2, "MEC141X Pin List"	Updated pins L2 and L4 in the 144 WFBGA package
	Table 42-14, "VTR Supply Current, I_VTR"	Updated Heavy Sleep 1 entries to show the 48 MHz Ring Oscillator Frequency at 12MHz
DS00001956A (05-27-15)	Initial Release	

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

PRODUCT IDENTIFICATION SYSTEM

Not all of the possible combinations of Device, Temperature Range and Package may be offered for sale. To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.[1]	-	[X/]	XXX[2]	-	[XX]	-	[X][3]
Device		Temperature Range	Package		ROM Version		Tape and Reel Option

Note: [ ] indicate designators that have blank options

Devices:	MEC1404[1] = 128KB SRAM, LPC Interface MEC1406[1] = 160KB SRAM, LPC Interface MEC1408[1] = 192KB SRAM, LPC Interface MEC1414[1] = 128KB SRAM, eSPI or LPC Interface MEC1416[1] = 160KB SRAM, eSPI or LPC Interface MEC1418[1] = 192KB SRAM, eSPI or LPC Interface		
Temperature Range Option:	Blank	=	0°C to +70°C (Commercial)
	I/	=	-40°C to +85°C (Industrial)
Package:	NU	=	128 pin VTQFP[2]
	SZ	=	144 pin WFBGA[2]
ROM Version:	Blank	=	Standard ROM
Tape and Reel Option:	Blank	=	Tray packaging
	TR	=	Tape and Reel[3]

Examples:

a) MEC1404-NU = 128KB SRAM, LPC Interface, Commercial temperature, 128 VTQFP

b) MEC1406-SZ = 160KB SRAM, LPC Interface, Commercial temperature, 144 WFBGA

c) MEC1418-NU-TR = 192KB SRAM, eSPI or LPC Interface, Commercial temperature, 128 VTQFP, tape and

Note 1: These products meet the halogen maximum concentration values per IEC61249-2-21.

Note 2: All package options are RoHS compliant. For RoHS compliance and environmental information, please visit http://www.microchip.com/pagehandler/en-us/aboutus/ehs.html.

Note 3: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

ISBN: 9781522409939

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

06/23/16

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Microchip:

MEC1418-NU MEC1418-SZ MEC1416-NU MEC1414-NU MEC1414-I/NU MEC1416-I/NU MEC1418-I/NU MEC1418-I/SZ

LifeElectronics
Живое партнерство

ООО «ЛайфЭлектроникс» **“LifeElectronics” LLC**

ИНН 7805602321 КПП 780501001 Р/С 40702810122510004610 ФАКБ "АБСОЛЮТ БАНК" (ЗАО) в г.Санкт-Петербурге К/С 30101810900000000703 БИК 044030703

Компания «Life Electronics» занимается поставками электронных компонентов импортного и отечественного производства от производителей и со складов крупных дистрибьюторов Европы, Америки и Азии.

С конца 2013 года компания активно расширяет линейку поставок компонентов по направлению коаксиальный кабель, кварцевые генераторы и конденсаторы (керамические, пленочные, электролитические), за счёт заключения дистрибьюторских договоров

Мы предлагаем:

- *Конкурентоспособные цены и скидки постоянным клиентам.*
- *Специальные условия для постоянных клиентов.*
- *Подбор аналогов.*
- *Поставку компонентов в любых объемах, удовлетворяющих вашим потребностям.*

- *Приемлемые сроки поставки, возможна ускоренная поставка.*
- *Доставку товара в любую точку России и стран СНГ.*
- *Комплексную поставку.*
- *Работу по проектам и поставку образцов.*
- *Формирование склада под заказчика.*

- *Сертификаты соответствия на поставляемую продукцию (по желанию клиента).*
- *Тестирование поставляемой продукции.*
- *Поставку компонентов, требующих военную и космическую приемку.*
- *Входной контроль качества.*
- *Наличие сертификата ISO.*

В составе нашей компании организован Конструкторский отдел, призванный помогать разработчикам, и инженерам.

Конструкторский отдел помогает осуществить:

- *Регистрацию проекта у производителя компонентов.*
- *Техническую поддержку проекта.*
- *Защиту от снятия компонента с производства.*
- *Оценку стоимости проекта по компонентам.*
- *Изготовление тестовой платы монтаж и пусконаладочные работы.*

Тел: +7 (812) 336 43 04 (многоканальный)
Email: org@lifeelectronics.ru

www.lifeelectronics.ru