# Low Latency 40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide

Last updated for Quartus Prime Design Suite: 16.0
**UG-01172**
2017.12.28

101 Innovation Drive
San Jose, CA 95134
www.altera.com

Subscribe
Send Feedback

now part of Intel

# Contents


**About the Low Latency 40- and 100-Gbps Ethernet MAC and PHY IP Core... 1-1**

Low Latency 40- and 100-Gbps Ethernet MAC and PHY IP Core Supported Features....1-3
Low Latency 40-100GbE IP Core Device Family and Speed Grade Support....1-4
Device Family Support....1-5
Low Latency 40-100GbE IP Core Device Speed Grade Support....1-5
IP Core Verification....1-6
Simulation Environment....1-7
Compilation Checking....1-7
Hardware Testing....1-7
Performance and Resource Utilization....1-7
Stratix V Resource Utilization for Low Latency 40-100GbE IP Cores....1-8
Arria 10 Resource Utilization for Low Latency 40-100GbE IP Cores....1-9
Release Information....1-10

**Getting Started.... 2-1**

Installation and Licensing for LL 40-100GbE IP Core for Stratix V Devices....2-2
Licensing IP Cores....2-3
OpenCore Plus IP Evaluation....2-3
Specifying the Low Latency 40-100GbE IP Core Parameters and Options....2-4
IP Core Parameters....2-5
Files Generated for Stratix V Variations....2-14
Files Generated for Arria 10 Variations....2-15
Integrating Your IP Core in Your Design....2-18
Pin Assignments....2-19
External Transceiver Reconfiguration Controller Required in Stratix V Designs....2-19
Transceiver PLL Required in Arria 10 Designs....2-20
External Time-of-Day Module for Variations with 1588 PTP Feature....2-22
Clock Requirements for 40GBASE-KR4 Variations....2-23
External TX MAC PLL....2-23
Placement Settings for the Low Latency 40-100GbE IP Core....2-23
Low Latency 40-100GbE IP Core Testbenches....2-23
Low Latency 40-100GbE IP Core Testbench Overview....2-24
Understanding the Testbench Behavior....2-27
Simulating the Low Latency 40-100GbE IP Core With the Testbenches....2-28
Generating the Low Latency 40-100GbE Testbench....2-29
Optimizing the Low Latency 40-100GbE IP Core Simulation With the Testbenches....2-30
Simulating with the Modelsim Simulator....2-30
Simulating with the NCSim Simulator....2-31
Simulating with the VCS Simulator....2-31
Testbench Output Example: Low Latency 40-100GbE IP Core....2-31

Compiling the Full Design and Programming the FPGA....2-32
Initializing the IP Core....2-32

**Functional Description....3-1**
High Level System Overview....3-2
Low Latency 40-100GbE MAC and PHY Functional Description....3-2
Low Latency 40-100GbE IP Core TX Datapath....3-3
Low Latency 40-100GbE IP Core TX Data Bus Interfaces....3-6
Low Latency 40-100GbE IP Core RX Datapath....3-17
Low Latency 40-100GbE IP Core RX Data Bus Interface....3-20
Low Latency 100GbE CAUI–4 PHY....3-28
External Reconfiguration Controller....3-28
External Transceiver PLL....3-28
External TX MAC PLL....3-28
Congestion and Flow Control Using Pause Frames....3-29
Pause Control and Generation Interface....3-32
Pause Control Frame Filtering....3-33
Link Fault Signaling Interface....3-33
Statistics Counters Interface....3-35
1588 Precision Time Protocol Interfaces....3-39
PHY Status Interface....3-55
Transceiver PHY Serial Data Interface....3-55
Low Latency 40GBASE-KR4 IP Core Variations....3-55
Control and Status Interface....3-56
Arria 10 Transceiver Reconfiguration Interface....3-58
Clocks....3-58
Resets....3-61
Signals....3-62
Low Latency 40-100GbE IP Core Signals....3-62
Software Interface: Registers....3-72
Low Latency 40-100GbE IP Core Registers....3-76
LL 40-100GbE Hardware Design Example Registers....3-114
Ethernet Glossary....3-116

**Debugging the Link....4-1**
Creating a SignalTap II Debug File to Match Your Design Hierarchy....4-2

**Arria 10 10GBASE-KR Registers....A-1**
10GBASE-KR PHY Register Definitions....A-1

**Differences Between Low Latency 40-100GbE IP Core and 40-100GbE IP Core v15.1....B-1**

**Additional Information....C-1**

Low Latency 40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide Archives....C-1
Low Latency 40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide Revision History....C-1
How to Contact Altera....C-11
Typographic Conventions....C-12

# About the Low Latency 40- and 100-Gbps Ethernet MAC and PHY IP Core

1

2017.12.28

UG-01172 Subscribe Send Feedback

The Altera® Low Latency 40- and 100-Gbps Ethernet (40GbE and 100GbE) media access controller (MAC) and PHY MegaCore® functions offer the lowest round-trip latency and smallest size to implement the *IEEE 802.3ba 40G and 100G Ethernet Standard* with an option to support the *IEEE 802.3ap-2007 Backplane Ethernet Standard*.

**Note:** This user guide documents the 16.0 version of the Altera Low Latency 40- and 100-Gbps Ethernet MAC and PHY IP core that targets a Stratix® V device or an Arria® 10 device. For the 16.1 release and beyond, two IP core user guides are available to document the Low Latency 40-Gbps Ethernet IP core and the Low Latency 100-Gbps Ethernet IP core separately. These two user guides document the variations that target an Arria 10 device. As of 2017.12.28, the 16.0 version of the Stratix V Low Latency 40-100GbE IP core is the most recent Stratix V Low Latency 40-100GbE IP core available in the Self-Service Licensing Center and this user guide provides its most current documentation.

The version of this product that supports Arria 10 devices is included in the Altera MegaCore IP Library and available from the Quartus® Prime IP Catalog.

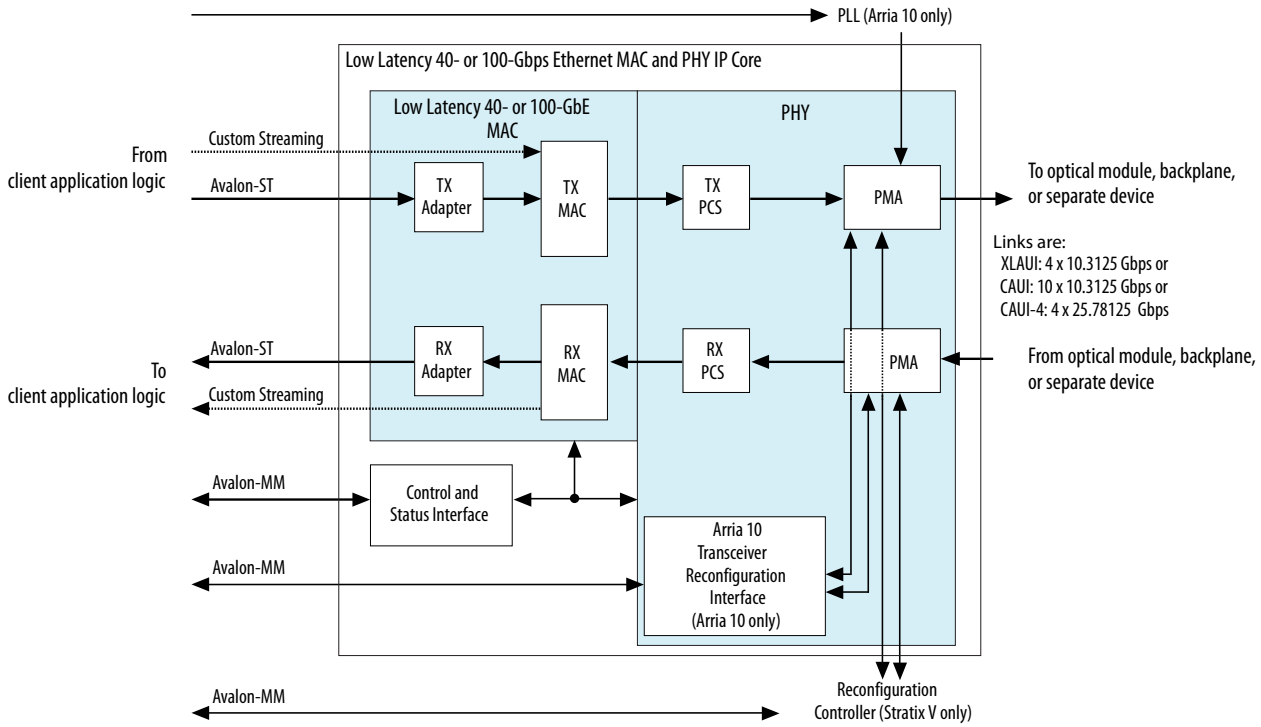
**Note:** The full product name, Low Latency 40- and 100-Gbps Ethernet MAC and PHY MegaCore Function, is shortened to Low Latency (LL) 40-100GbE IP core in this document. In addition, although multiple variations are available from the parameter editor, this document refers to this product as a single IP core, because all variations are configurable from the same parameter editor.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

**Figure 1-1: Low Latency 40GbE and 100GbE MAC and PHY IP Cores**

Main blocks, internal connections, and external block requirements.




As illustrated, on the MAC client side you can choose a wide, standard Avalon® Streaming (Avalon-ST) interface, or a narrower, custom streaming interface. Depending on the variant you choose, the MAC client side Avalon Streaming (Avalon-ST) interface is either 256 or 512 bits of data mapped to either four or ten 10.3125 Gbps transceiver PHY links, depending on data rate, or to four 25.78125 Gbps transceiver PHY links.

The 40GbE (XLAUI) interface has 4x10.3125 Gbps links. The 100GbE (CAUI) interface has 10x10.3125 Gbps links. For Arria 10 devices only, you can configure a 40GbE 40GBASE-KR4 variation to support Backplane Ethernet. For Arria 10 GT devices only, you can configure a 100GbE CAUI-4 option, with 4x25.78125 Gbps links.

The FPGA serial transceivers are compliant with the IEEE 802.3ba standard XLAUI, CAUI, and CAUI-4 specifications. The IP core configures the transceivers to implement the relevant specification for your IP core variation. You can connect the transceiver interfaces directly to an external physical medium dependent (PMD) optical module or to another device.

The IP core provides standard MAC and physical coding sublayer (PCS) functions with a variety of configuration and status registers. You can exclude the statistics registers. If you exclude these registers, you can monitor the statistics counter increment vectors that the IP core provides at the client side interface and maintain your own counters.

**Related Information**

- **Low Latency 40-100GbE MAC and PHY Functional Description** on page 3-2
  Provides detailed descriptions of LL 40-100GbE IP core operation and functions.

- **Introduction to Altera IP Cores**
  Provides general information about all Altera IP cores, including parameterizing, generating, upgrading, and simulating IP.
- **Creating Version-Independent IP and Qsys Simulation Scripts**
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- **Project Management Best Practices**
  Guidelines for efficient management and portability of your project and IP files.
- **Low Latency 40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide Archives** on page 7-1
- **Low Latency 40G Ethernet Example Design User Guide**
- **Low Latency 100G Ethernet Example Design User Guide**
- **Low Latency 40-Gbps Ethernet IP Core User Guide**
  Documents the current release of the Low Latency 40-Gbps Ethernet IP core that targets an Arria 10 device.
- **Low Latency 100-Gbps Ethernet IP Core User Guide**
  Documents the current release of the Low Latency 100-Gbps Ethernet IP core that targets an Arria 10 device.

## Low Latency 40- and 100-Gbps Ethernet MAC and PHY IP Core Supported Features

All LL 40-100GbE IP core variations include both a MAC and a PHY, and all variations are in full-duplex mode. These IP core variations offer the following features:

- Designed to the *IEEE 802.3ba-2010 High Speed Ethernet Standard* available on the IEEE website (www.ieee.org).
- Soft PCS logic that interfaces seamlessly to Altera 10.3125 Gbps and 25.78125 Gbps serial transceivers.
- Standard XLAUI or CAUI external interface consisting of FPGA hard serial transceiver lanes operating at 10.3125 Gbps , or the CAUI-4 external interface consisting of four FPGA hard serial transceiver lanes operating at 25.78125 Gbps.
- Supports 40GBASE-KR4 PHY based on 64B/66B encoding with data striping and alignment markers to align data from multiple lanes.
- Supports 40GBASE-KR4 PHY and forward error correction (FEC) option for interfacing to backplanes.
- Supports Synchronous Ethernet (Sync-E) by providing an optional CDR recovered clock output signal to the device fabric.
- Avalon Memory-Mapped (Avalon-MM) management interface to access the IP core control and status registers.
- Avalon-ST data path interface connects to client logic with the start of frame in the most significant byte (MSB) when optional adapters are used. Interface has data width 256 or 512 bits depending on the data rate.
- Optional custom streaming data path interface with narrower bus width and a start frame possible on 64-bit word boundaries without the optional adapters. Interface has data width 128 or 256 bits depending on the data rate.
- Support for jumbo packets.
- TX and RX CRC pass-through control.
- Optional TX CRC generation and insertion.

- RX CRC checking and error reporting.
- TX error insertion capability supports test and debug.
- RX and TX preamble pass-through options for applications that require proprietary user management information transfer.
- TX automatic frame padding to meet the 64-byte minimum Ethernet frame length at the LL 40-100GbE Ethernet connection.
- Hardware and software reset control.
- Pause frame filtering control.
- Received control frame type indication.
- MAC provides cut-through frame processing.
- Optional deficit idle counter (DIC) options to maintain a finely controlled 8-byte or 12-byte inter-packet gap (IPG) minimum average.
- Optional IEEE 802.3 Clause 31 Ethernet flow control operation using the pause registers or pause interface.
- Optional priority-based flow control that complies with the *IEEE Standard 802.1Qbb-2011—Amendment 17: Priority-based Flow Control*, using the pause registers for fine control.
- RX PCS lane skew tolerance that exceeds the IEEE 802.3-2012 Ethernet standard clause 82.2.12 requirements: 1900 bits RX lane skew tolerance for LL 40GbE IP cores and 1000 bits RX lane skew tolerance for LL 100GbE IP cores.
- Optional support for the IEEE Standard 1588-2008 Precision Clock Synchronization Protocol (1588 PTP).
- Optional statistics counters.
- Optional fault signaling: detects and reports local fault and generates remote fault, with *IEEE 802.3ba-2012 Ethernet Standard* Clause 66 support.
- Optional serial PMA loopback (TX to RX) at the serial transceiver for self-diagnostic testing.
- Optional access to Altera Debug Master Endpoint (ADME) for debugging or monitoring PHY signal integrity.

The LL 40-100GbE IP core can support full wire line speed with a 64-byte frame length and back-to-back or mixed length traffic with no dropped packets.

For a detailed specification of the Ethernet protocol refer to the *IEEE 802.3ba-2010 High Speed Ethernet Standard*.

**Related Information**

**IEEE website**

The *IEEE 802.3ba-2010 High Speed Ethernet Standard* and the *IEEE Standard 802.1Qbb-2011—Amendment 17: Priority-based Flow Control* are available on the IEEE website.

# Low Latency 40-100GbE IP Core Device Family and Speed Grade Support

The following sections list the device family and device speed grade support offered by the Low Latency 40-100GbE IP core:

**Device Family Support** on page 1-5

**Low Latency 40-100GbE IP Core Device Speed Grade Support** on page 1-5

# Device Family Support

**Table 1-1: Altera IP Core Device Support Levels**

| Device Support Level | Definition |
|---|---|
| **Preliminary** | The IP core is verified with preliminary timing models for this device family. The IP core meets all functional requirements, but might still be undergoing timing analysis for the device family. It can be used in production designs with caution. |
| **Final** | The IP core is verified with final timing models for this device family. The IP core meets all functional and timing requirements for the device family and can be used in production designs. |

**Table 1-2: Low Latency 40-100GbE IP Core Device Family Support**

Shows the level of support offered by the Low Latency 40-100GbE IP core for each Altera device family.

| Device Family | Support |
|---|---|
| Stratix V (GX, GT, and GS) | Final |
| Arria 10 (GX, GT, and GS) | Refer to What's New in IP webpage |
| Other device families | Not supported |

**Related Information**

- **Low Latency 40-100GbE IP Core Device Speed Grade Support** on page 1-5
- **What's New in IP**
  Information about the device support level in the current release of the Quartus Prime software.

# Low Latency 40-100GbE IP Core Device Speed Grade Support

**Table 1-3: Slowest Supported Device Speed Grades**

Lists the slowest supported device speed grades for standard variations of the Low Latency 40-100GbE IP core. IP core variations that include a 1588 PTP module might require Quartus Prime seed sweeping to achieve a comfortable timing margin.

| MegaCore Function | Device Family | Supported Speed Grades |
|---|---|---|
| 40GbE | Stratix V (GX) | I3, C3 |
| | Stratix V (GT) | I3, C2 |
| | Stratix V (GS) | I3, C3 |
| | Arria 10 (GX, GT, GS) | I2, C2 |
| 40GbE (40GBASE-KR4 option) | Arria 10 (GX, GT, GS) | I2, C2 |
| 100GbE | Stratix V (GX) | I2, C2 |
| | Stratix V (GT) | I2, C2 |
| | Stratix V (GS) | I2, C2 |
| | Arria 10 (GX, GT, GS) | I2, C2 |
| 100GbE (CAUI–4 option) | Arria 10 GT | I2, C2 |

# IP Core Verification

To ensure functional correctness of the Low Latency 40-100GbE IP core, Altera performs extensive validation through both simulation and hardware testing. Before releasing a version of the Low Latency 40- and 100-Gbps Ethernet MAC and PHY IP core, Altera runs comprehensive regression tests in the current or associated version of the Quartus Prime software.

**Related Information**

- **Knowledge Base Errata for Low Latency 40-100GbE IP core**
  Exceptions to functional correctness are documented in the Low Latency 40-100GbE IP core errata.
- **Altera IP Release Notes**
  Changes to the Low Latency 40-100GbE IP core are noted in the Altera IP Release Notes starting from the Quartus II software v14.0 Arria 10 Edition.

## Simulation Environment

Altera performs the following tests on the Low Latency 40-100GbE MAC and PHY IP core in the simulation environment using internal and third party standard bus functional models (BFM):

- Constrained random tests that cover randomized frame size and contents
- Randomized error injection tests that inject Frame Check Sequence (FCS) field errors, runt packets, and corrupt control characters, and then check for the proper response from the IP core
- Assertion based tests to confirm proper behavior of the IP core with respect to the specification
- Extensive coverage of our runtime configuration space and proper behavior in all possible modes of operation

## Compilation Checking

Altera performs compilation testing on an extensive set of Low Latency 40-100GbE MAC and PHY IP core variations and designs that target different devices, to ensure the Quartus Prime software places and routes the IP core ports correctly.

## Hardware Testing

Altera performs hardware testing of the key functions of the Low Latency 40-100GbE MAC and PHY IP core using standard 40-100Gbps Ethernet network test equipment and optical modules. The Altera hardware tests of the Low Latency 40-100GbE IP core also ensure reliable solution coverage for hardware related areas such as performance, link synchronization, and reset recovery. The IP core is tested with Stratix V devices.

# Performance and Resource Utilization

The following sections provide performance and resource utilization data for the Low Latency 40GbE and 100GbE IP cores.

**Table 1-4: IP Core Variation Encoding for Resource Utilization Tables**

"On" indicates the parameter is turned on. The symbol "—" indicates the parameter is turned off or not available.

| IP Core Variation / Parameter | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| **Data interface** | Custom-ST | Avalon-ST | Avalon-ST | Avalon-ST | Avalon-ST | Avalon-ST |
| **Flow control mode** | No flow control | No flow control | Standard flow control | Standard flow control | No flow control | No flow control |
| **Average interpacket gap** | 12 | 12 | 12 | 12 | 12 | 12 |
| **Enable 1588 PTP** | — | — | — | On | — | — |

| IP Core Variation / Parameter | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| **Enable link fault generation** | — | — | On | On | — | — |
| **Enable TX CRC insertion** | — | On | On | On | On | On |
| **Enable preamble passthrough** | — | — | On | On | — | — |
| **Enable alignment EOP on FCS word** | — | On | On | On | On | On |
| **Enable TX statistics** | — | On | On | On | On | On |
| **Enable RX statistics** | — | On | On | On | On | On |
| **Enable KR4** | — | — | — | — | On | On |
| **Include FEC sublayer** | — | — | — | — | — | On |

## Stratix V Resource Utilization for Low Latency 40-100GbE IP Cores

Resource utilization changes depending on the parameter settings you specify in the Low Latency 40-100GbE parameter editor. For example, if you turn on pause functionality or statistics counters in the LL 40-100GbE parameter editor, the IP core requires additional resources to implement the additional functionality.

### Table 1-5: IP Core FPGA Resource Utilization in Stratix V Devices

Lists the resources and expected performance for selected variations of the Low Latency 40-100GbE IP cores in a Stratix V device.

These results were obtained using the Quartus II v14.1 software.

- The numbers of ALMs and logic registers are rounded up to the nearest 100.
- The numbers of ALMs, before rounding, are the **ALMs needed** numbers from the Quartus II Fitter Report.

| 40GbE Variation | ALMs | Dedicated Logic Registers | Memory M20K |
|---|---|---|---|
| 40GbE variation A | 5300 | 12800 | 13 |
| 40GbE variation B | 9900 | 21500 | 13 |
| 40GbE variation C | 10900 | 24100 | 13 |
| 40GbE variation D | 14000 | 31000 | 17 |

| 100GbE Variation | ALMs | Dedicated Logic Registers | Memory M20K |
|---|---|---|---|
| 100GbE variation A | 9500 | 23000 | 29 |
| 100GbE variation B | 20900 | 48400 | 61 |
| 100GbE variation C | 22100 | 52500 | 61 |
| 100GbE variation D | 26900 | 63700 | 65 |

**Related Information**

**Fitter Resources Reports in the Quartus Prime Help**
Information about Quartus Prime resource utilization reporting, including **ALMs needed**.

## Arria 10 Resource Utilization for Low Latency 40-100GbE IP Cores

Resource utilization changes depending on the parameter settings you specify in the Low Latency 40-100GbE parameter editor. For example, if you turn on pause functionality or statistics counters in the LL 40-100GbE parameter editor, the IP core requires additional resources to implement the additional functionality.

### Table 1-6: IP Core FPGA Resource Utilization in Arria 10 Devices

Lists the resources and expected performance for selected variations of the Low Latency 40-100GbE IP cores in an Arria 10 device.

These results were obtained using the Quartus II v14.1 software.

- The numbers of ALMs and logic registers are rounded up to the nearest 100.
- The numbers of ALMs, before rounding, are the **ALMs needed** numbers from the Quartus II Fitter Report.

| 40GbE Variation | ALMs | Dedicated Logic Registers | Memory M20K |
|---|---|---|---|
| 40GbE variation A | 5400 | 12800 | 13 |
| 40GbE variation B | 10100 | 21200 | 13 |
| 40GbE variation C | 11000 | 24100 | 13 |
| 40GbE variation D | 14200 | 31100 | 17 |
| 40GbE variation E | 14400 | 28200 | 26 |
| 40GbE variation F | 16300 | 29300 | 26 |

| 100GbE Variation | ALMs | Dedicated Logic Registers | Memory M20K |
|---|---|---|---|
| 100GbE variation A | 13100 | 29000 | 29 |
| 100GbE variation B | 21200 | 47600 | 61 |
| 100GbE variation C | 22500 | 51800 | 61 |
| 100GbE variation D | 27000 | 63200 | 65 |
| **CAUI-4 Variation** | **ALMs** | **Dedicated Logic Registers** | **Memory M20K** |
| CAUI-4 variation B | 22700 | 51300 | 61 |

**Related Information**

**Fitter Resources Reports in the Quartus Prime Help**
Information about Quartus Prime resource utilization reporting, including **ALMs needed**.

# Release Information

**Table 1-7: Low Latency 40-100GbE IP Core Current Release Information**

| Item | Description |
|---|---|
| Version | 16.0 |
| Release Date | 2016.05.02 |
| Ordering Codes | Low Latency 40G Ethernet MAC and PHY: IP-40GEUMACPHY<br>Low Latency 40G Ethernet MAC and PHY with 1588: IP-40GEUMACPHYF<br>Low Latency 100G Ethernet MAC and PHY: IP-100GEUMACPHY<br>Low Latency 100G Ethernet MAC and PHY with 1588: IP-100GEUMACPHYF<br>Low Latency 40G Ethernet MAC and 40GBASE-KR4 PHY with FEC: IP-40GBASEKR4PHY |
| Product ID | Low Latency 40G Ethernet MAC and PHY: 011B<br>Low Latency 40G Ethernet MAC and PHY with 1588: 011C<br>Low Latency 100G Ethernet MAC and PHY: 011A<br>Low Latency 100G Ethernet MAC and PHY with 1588: 011D<br>Low Latency 40G Ethernet MAC and 40GBASE-KR4 PHY with FEC: 0113 |

| Item | Description |
|---|---|
| Vendor ID | 6AF7 |

# Getting Started 2

2017.12.28

UG-01172

Subscribe Send Feedback

The following sections explain how to install, parameterize, simulate, and initialize the Low Latency 40-100GbE IP core:

**Installation and Licensing for LL 40-100GbE IP Core for Stratix V Devices** on page 2-2
The Low Latency 40-100GbE IP core that targets a Stratix V device is an extended IP core which is not included with the Quartus Prime release. This section provides a general overview of the Altera extended IP core installation process to help you quickly get started with any Altera extended IP core.

**Licensing IP Cores** on page 2-3
The Low Latency 40-100GbE IP core that targets an Arria 10 device is a standard Altera IP core in the Altera IP Library.

**Specifying the Low Latency 40-100GbE IP Core Parameters and Options** on page 2-4
The LL 40-100GbE IP core for Arria 10 devices supports a standard customization and generation process from the Quartus Prime IP Catalog. After you install and integrate the extended IP core in the ACDS release, the LL 40-100GbE IP core for Stratix V devices also supports the standard customization and generation process. The Low Latency 40-100GbE IP core is not supported in Qsys.

**IP Core Parameters** on page 2-5
The Low Latency 40-100GbE parameter editor provides the parameters you can set to configure the Low Latency 40-100GbE IP core and simulation and hardware design examples.

**Files Generated for Stratix V Variations** on page 2-14
The Quartus Prime software generates the following output for your Stratix V LL 40-100GbE IP core.

**Files Generated for Arria 10 Variations** on page 2-15
The Quartus Prime software generates the following IP core output file structure when targeting Arria 10 devices.

**Integrating Your IP Core in Your Design** on page 2-18

**Low Latency 40-100GbE IP Core Testbenches** on page 2-23
Altera provides a testbench , a hardware design example, and a compilation-only example design with most variations of the Low Latency 40-100GbE IP core. The testbench is available for simulation of your IP core, and the hardware design example can be run on hardware. You can run the testbench to observe the IP core behavior on the various interfaces in simulation.

**Simulating the Low Latency 40-100GbE IP Core With the Testbenches** on page 2-28

**Compiling the Full Design and Programming the FPGA** on page 2-32

**Initializing the IP Core** on page 2-32

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

**Related Information**

- **Introduction to Altera IP Cores**
  Provides general information about all Altera IP cores, including parameterizing, generating, upgrading, and simulating IP.
- **Creating Version-Independent IP and Qsys Simulation Scripts**
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- **Project Management Best Practices**
  Guidelines for efficient management and portability of your project and IP files.
- **Introduction to Altera IP Cores**
  More information about generating an Altera IP core and integrating it in your Quartus Prime project.

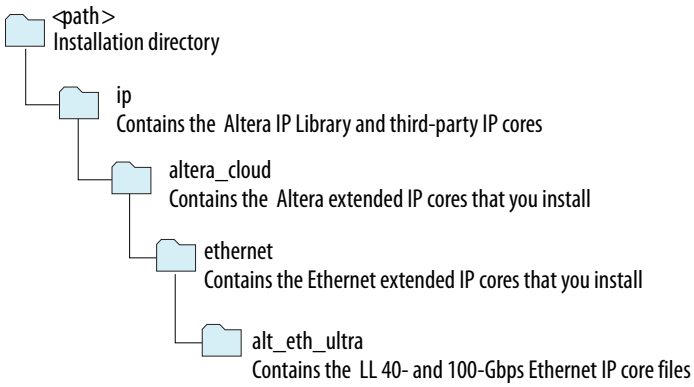
# Installation and Licensing for LL 40-100GbE IP Core for Stratix V Devices

The Low Latency 40-100GbE IP core that targets the Stratix V device family is an extended IP core which is not included with the Quartus Prime release. This section provides a general overview of the Altera extended IP core installation process to help you quickly get started with any Altera extended IP core.

The Altera extended IP cores are available from the Altera Self-Service Licensing Center (SSLC). Refer to Related Links below for the correct link for this IP core.

**Figure 2-1: IP Core Directory Structure**

Directory structure after you install the Low Latency 40-100GbE IP core. The default installation directory <*path*> on Windows is **C:\altera**\< *version number* >; on Linux it is **/opt/altera**< *version number* >**.**




You can evaluate an IP core in simulation and in hardware until you are satisfied with its functionality and performance. You must purchase a license for the IP core when you want to take your design to production. After you purchase a license for an Altera IP core, you can request a license file from the Altera Licensing page of the Altera website and install the license on your computer.

**Related Information**

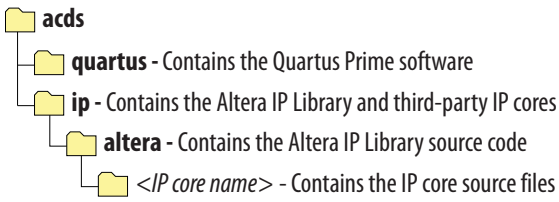
- **Altera website**
- **Altera Licensing website**

- **Altera Self-Service Licensing Center**
  After you purchase the Low Latency 40-100GbE IP core that supports Stratix V devices, the IP core is available for download from the SSLC page in your myAltera account. Altera requires that you create a myAltera account if you do not have one already, and log in to access the SSLC. On the SSLC page, click Run for this IP core. The SSLC provides an installation dialog box to guide your installation of the IP core.

# Licensing IP Cores

The Altera IP Library provides many useful IP core functions for your production use without purchasing an additional license. Some Altera MegaCore IP functions require that you purchase a separate license for production use. However, the OpenCore® feature allows evaluation of any Altera IP core in simulation and compilation in the Quartus Prime software. After you are satisfied with functionality and performance, visit the Self Service Licensing Center to obtain a license number for any Altera product.

**Figure 2-2: IP Core Installation Path**




**Note:** The default IP installation directory on Windows is `<drive>:\altera\<version number>`; on Linux the IP installation directory is `<home directory>/altera/<version number>`.

## OpenCore Plus IP Evaluation

Altera's free OpenCore Plus feature allows you to evaluate licensed MegaCore IP cores in simulation and hardware before purchase. You only need to purchase a license for MegaCore IP cores if you decide to take your design to production. OpenCore Plus supports the following evaluations:

- Simulate the behavior of a licensed IP core in your system.
- Verify the functionality, size, and speed of the IP core quickly and easily.
- Generate time-limited device programming files for designs that include IP cores.
- Program a device with your IP core and verify your design in hardware.

OpenCore Plus evaluation supports the following two operation modes:

- Untethered—run the design containing the licensed IP for a limited time.
- Tethered—run the design containing the licensed IP for a longer time or indefinitely. This requires a connection between your board and the host computer.

**Note:** All IP cores that use OpenCore Plus time out simultaneously when any IP core in the design times out.

**Related Information**

- **Altera Licensing Site**
- **Altera Software Installation and Licensing Manual**

# Specifying the Low Latency 40-100GbE IP Core Parameters and Options

The Low Latency 40-100GbE parameter editor allows you to quickly configure your custom IP variation. Use the following steps to specify IP core options and parameters in the Quartus Prime software.

1. In the IP Catalog (**Tools** > **IP Catalog**), select a target device family.
2. In the IP Catalog, locate and double-click the name of the IP core to customize. The New IP Variation window appears.
3. Specify a top-level name for your custom IP variation. The parameter editor saves the IP variation settings in a file named <*your_ip*>`.qsys` (for Arria 10 variations) or <*your_ip*>`.qip` (for Stratix V variations).
4. If your IP core targets the Arria 10 device family, you must select a specific device in the **Device** field or maintain the default device the Quartus Prime software lists. If you target a specific Altera development kit, the hardware design example overwrites the selection with the device on the target board.
5. Click **OK**. The parameter editor appears.
6. Specify the parameters and options for your IP variation in the parameter editor, including one or more of the following. Refer to your IP core user guide for information about specific IP core parameters.
   - Specify parameters defining the IP core functionality, port configurations, and device-specific features.
   - Specify options for processing the IP core files in other EDA tools.
7. For Arria 10 variations, follow these steps:
   a. Optionally, to generate a simulation testbench or example project, follow the instructions in **Generating the Low Latency 40-100GbE Testbench** on page 2-29.
   b. Click **Generate HDL**. The **Generation** dialog box appears.
   c. Specify output file generation options, and then click **Generate**. The IP variation files generate according to your specifications.
   d. Click **Finish**. The parameter editor adds the top-level `.qsys` file to the current project automatically. If you are prompted to manually add the `.qsys` file to the project, click **Project** > **Add/Remove Files in Project** to add the file.
8. For Stratix V variations, follow these steps:
   a. Click **Finish**.
   b. Optionally, to generate a simulation testbench or example project, follow the instructions in **Generating the Low Latency 40-100GbE Testbench** on page 2-29.
   After you click Finish and optionally follow the additional step to generate a simulation testbench and example project, if available for your IP core variation, the parameter editor adds the top-level `.qsys` file or top-level `.qip` file to the current project automatically. If you are prompted to manually add the `.qip` file to the project, click **Project** > **Add/Remove Files in Project** to add the file.
9. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports.

# IP Core Parameters

The Low Latency 40-100GbE parameter editor provides the parameters you can set to configure the Low Latency 40-100GbE IP core and simulation and hardware design examples.

LL 40-100GbE IP core variations that target an Arria 10 device include an **Example Design** tab.

### Table 2-1: Low Latency 40-100GbE Parameters: Main Tab

Describes the parameters for customizing the 40-100GbE IP core on the Main tab of the 40-100GbE parameter editor.

| Parameter | Type | Range | Default Setting | Parameter Description |
|---|---|---|---|---|
| **General Options** | | | | |
| **Device family** | String | • Stratix V<br>• Arria 10 | According to the setting in the project or IP Catalog settings. | Selects the device family. |
| **Protocol speed** | String | • 40 GbE<br>• 100 GbE | 100 GbE | Selects the MAC datapath width. |
| **Data interface** | String | • Custom–ST<br>• Avalon–ST | Avalon–ST | Selects the Avalon–ST interface or the narrower, custom streaming client interface to the MAC.<br>If you select the custom streaming client interface, the **Flow control mode** and **Enable 1588 PTP** parameters are not available. |
| **PCS/PMA Options** | | | | |
| **Enable CAUI4 PCS** (1)(2) | Boolean | • True<br>• False | False | If turned on, the IP core is a 100GbE CAUI-4 variation, with four 25.78125 Gbps transceiver PHY links. |

(1) The **Enable CAUI4 PCS** parameter is disabled when **Protocol speed** is set to 100GbE and **Device family** is not Arria 10, and when **Protocol speed** is set to 40GbE. If the parameter is disabled, the IP core is configured with the regular 100 Gbps PHY link option of 10 x 10.3125 Gbps.

(2) For the **Device family** parameter, the CAUI-4 option requires the Arria 10 device.

| Parameter | Type | Range | Default Setting | Parameter Description |
|---|---|---|---|---|
| **Enable SyncE** | Boolean | • True<br>• False | False | Exposes the RX recovered clock as an output signal. This feature supports the Synchronous Ethernet standard described in the ITU-T G. 8261, G.8262, and G.8264 recommendations.<br>This parameter is available only in variations that target an Arria 10 device. |
| **PHY reference frequency** | Integer (encoding) | • 322.265625 MHz<br>• 644.53125 MHz | 644.53125 MHz | Sets the expected incoming PHY `clk_ref` reference frequency. The input clock frequency must match the frequency you specify for this parameter (± 100ppm). |
| **Use external TX MAC PLL** | Boolean | • True<br>• False | False | If you turn this option on, the IP core is configured to expect an input clock to drive the TX MAC. The input clock signal is `clk_txmac_in`. |
| **Flow Control Options** | | | | |
| **Flow control mode** | String | • No flow control<br>• Standard flow control<br>• Priority-based flow control | No flow control | Configures the flow control mechanism the IP core implements. Standard flow control is Ethernet standard flow control.<br>If you select the custom streaming client interface, the IP core must be configured with no flow control, and this parameter is not available. |
| **Number of PFC queues** | Integer | 1–8 | 8 | Number of distinct priority queues for priority-based flow control. This parameter is available only if you set **Flow control mode** to **Priority-based flow control**. |

| Parameter | Type | Range | Default Setting | Parameter Description |
|---|---|---|---|---|
| **Average interpacket gap** | String | • Disable deficit idle counter<br>• 8<br>• 12 | 12 | If you set the value of this parameter to **8** or to **12**, the IP core includes a deficit idle counter (DIC), which maintains an average interpacket gap (IPG) of 8 or 12, as you specify. If you set the value of this parameter to **Disable deficit idle counter**, the IP core is configured without the DIC, and does not maintain the required minimum average IPG. The Ethernet standard requires a minimum average IPG of 12. Turning off **Average interpacket gap** increases bandwidth. |
| **MAC Options** | | | | |
| **Enable 1588 PTP** | Boolean | • True<br>• False | False | If turned on, the IP core supports the IEEE Standard 1588-2008 Precision Clock Synchronization Protocol, by providing the hooks to implement the Precise Timing Protocol (PTP).<br>If you select the custom streaming client interface, the IP core must be configured without 1588 support, and this parameter is not available. |
| **Enable 96b Time of Day Format** | Boolean | • True<br>• False | True | Include the 96-bit interface to the TOD module. If you turn on this parameter, the TOD module that is generated with the IP core has a matching 96-bit timestamp interface.<br>If **Enable 1588 PTP** is turned on, you must turn on at least one of **Enable 96b Time of Day Format** and **Enable 64b Time of Day Format**. You can turn on both **Enable 96b Time of Day Format** and **Enable 64b Time of Day Format** to generate a TOD interface for each format.<br>This parameter is available only in variations with **Enable 1588 PTP** turned on. |

| Parameter | Type | Range | Default Setting | Parameter Description |
|---|---|---|---|---|
| **Enable 64b Time of Day Format** | Boolean | • True<br>• False | False | Include the 64-bit interface to the TOD module. If you turn on this parameter, the TOD module that is generated with the IP core has a matching 64-bit timestamp interface.<br>If **Enable 1588 PTP** is turned on, you must turn on at least one of **Enable 96b Time of Day Format** and **Enable 64b Time of Day Format**. You can turn on both **Enable 96b Time of Day Format** and **Enable 64b Time of Day Format** to generate a TOD interface for each format.<br>This parameter is available only in variations with **Enable 1588 PTP** turned on. |
| **Timestamp fingerprint width** | Integer | 1–16 | 1 | Specifies the number of bits in the fingerprint that the IP core handles.<br>This parameter is available only in variations with **Enable 1588 PTP** turned on. |
| **Enable link fault generation** | Boolean | • True<br>• False | False | If turned on, the IP core includes the link fault signaling modules and relevant signals. If turned off, the IP core is configured without these modules and without these signals. Turning on link fault signaling provides your design a tool to improve reliability, but increases resource utilization. |

| Parameter | Type | Range | Default Setting | Parameter Description |
|---|---|---|---|---|
| **Enable TX CRC insertion** | Boolean | • True<br>• False | True | If turned on, the IP core inserts a 32-bit Frame Check Sequence (FCS), which is a CRC-32 checksum, in outgoing Ethernet frames. If turned off, the IP core does not insert the CRC-32 sequence in outgoing Ethernet communication. Turning on TX CRC insertion improves reliability but increases resource utilization and latency through the IP core.<br>If you turn on flow control, the IP core must be configured with TX CRC insertion, and this parameter is not available. |
| **Enable preamble passthrough** | Boolean | • True<br>• False | False | If turned on, the IP core is in RX and TX preamble pass-through mode. In RX preamble pass-through mode, the IP core passes the preamble and SFD to the client instead of stripping them out of the Ethernet packet. In TX preamble pass-through mode, the client specifies the preamble to be sent in the Ethernet frame. |
| **Enable alignment EOP on FCS word** | Boolean | • True<br>• False | True | If turned on, the IP core aligns the 32-bit Frame Check Sequence (FCS) error signal with the assertion of the EOP by delaying the RX data bus to match the latency of the FCS computation. If turned off, the IP core does not delay the RX data bus to match the latency of the FCS computation. If the parameter is turned off, the FCS error signal, in the case of an FCS error, is asserted in a later clock cycle than the relevant assertion of the EOP signal.<br>Altera recommends that you turn on this option. Otherwise, the latency between the EOP indication and assertion of the FCS error signal is non-deterministic.<br>You must turn on this parameter if your design relies on the `rx_inc_octetsOK` signal.. |

| Parameter | Type | Range | Default Setting | Parameter Description |
|---|---|---|---|---|
| **Enable TX statistics** | Boolean | • True<br>• False | True | If turned on, the IP core includes built–in TX statistics counters. If turned off, the IP core is configured without TX statistics counters. In any case, the IP core is configured with TX statistics counter increment output vectors. |
| **Enable RX statistics** | Boolean | • True<br>• False | True | If turned on, the IP core includes built–in RX statistics counters. If turned off, the IP core is configured without RX statistics counters. In any case, the IP core is configured with RX statistics counter increment output vectors. |
| **Configuration, Debug and Extension Options** | | | | |
| **Enable Altera Debug Master Endpoint (ADME)** | Boolean | • True<br>• False | False | If turned on, the IP core turns on the following features in the Arria 10 PHY IP core that is included in the LL 40-100GbE IP core:<br>• **Enable Altera Debug Master Endpoint (ADME)**<br>• **Enable capability registers**<br>If turned off, the IP core is configured without these features.<br>This parameter is available only in variations that target an Arria 10 device. For information about these Arria 10 features, refer to the **Arria 10 Transceiver PHY User Guide**. |

**Table 2-2: LL 40-100GbE Parameters: 40GBASE-KR4 Tab**

Describes the parameters for customizing a 40GBASE-KR4 Low Latency 40-100GbE IP core, on the 40GBASE-KR4 tab of the LL 40-100GbE parameter editor. The parameters on this tab are available only if the following conditions hold:

- Your IP core targets an Arria 10 device. You set the target device family for your Quartus Prime project or in the Quartus Prime software before you acess the IP Catalog.
- You select the value of **40GbE** for the **Protocol speed** parameter on the Main tab.
- You turn off the **Enable 1588 PTP** parameter on the Main tab.

| Parameter | Type | Range | Default Setting | Parameter Description |
|---|---|---|---|---|
| **KR4 General Options** | | | | |

| Parameter | Type | Range | Default Setting | Parameter Description |
|---|---|---|---|---|
| **Enable KR4** | Boolean | • True<br>• False | False | If this parameter is turned on, the IP core is a 40GBASE-KR4 variation. If this parameter is turned off, the IP core is not a 40GBASE-KR4 variation, and the other parameters on this tab are not available. |
| **Status clock rate** | Frequency range | 100–125 MHz | 100 MHz | Sets the expected incoming `clk_status` frequency. The input clock frequency must match the frequency you specify for this parameter.<br>The IP core is configured with this information:<br>• To ensure the IP core measures the link fail inhibit time accurately. Determines the value of the Link Fail Inhibit timer (IEEE 802.3 clause 73.10.2) correctly.<br>• If `clk_status` frequency is not 100 MHz, to adjust the PHY clock monitors to report accurate frequency information.<br>This parameter determines the **PHY Management clock (MGMT_CLK) frequency in MHz** parameter of the underlying 10GBASE-KR PHY IP core. However, the default value of the **Status clock rate** parameter is not identical to the default value of the PHY IP core **PHY Management clock (MGMT_CLK) frequency in MHz** parameter. |
| **Auto-Negotiation** | | | | |
| **Enable Auto-Negotiation** | Boolean | • True<br>• False | True | If this parameter is turned on, the IP core includes logic to implement auto-negotiation as defined in Clause 73 of *IEEE Std 802.3ap–2007*. If this parameter is turned off, the IP core does not include auto-negotiation logic and cannot perform auto-negotiation.<br>Currently the IP core can only negotiate to KR4 mode. |
| **Link fail inhibit time for 40Gb Ethernet** | Integer (Unit: ms) | 500–510 ms | 504 ms | Specifies the time before link status is set to FAIL or OK. A link fails if the time duration specified by this parameter expires before link status is set to OK. For more information, refer to *Clause 73 Auto-Negotiation for Backplane Ethernet* in *IEEE Standard 802.3ap–2007*.<br>The 40GBASE-KR4 IP core asserts the `rx_pcs_ready` signal to indicate link status is OK. |

| Parameter | Type | Range | Default Setting | Parameter Description |
|---|---|---|---|---|
| **Auto-Negotiation Master** | String | • Lane 0<br>• Lane 1<br>• Lane 2<br>• Lane 3 | Lane 0 | Selects the master channel for auto-negotiation. |
| **Pause ability–C0** | Boolean | • True<br>• False | True | If this parameter is turned on, the IP core indicates on the Ethernet link that it supports symmetric pauses as defined in *Annex 28B* of Section 2 of *IEEE Std 802.3–2008*. |
| **Pause ability–C1** | Boolean | • True<br>• False | True | If this parameter is turned on, the IP core indicates on the Ethernet link that it supports asymmetric pauses as defined in *Annex 28B* of Section 2 of *IEEE Std 802.3–2008*. |
| **Link Training: PMA Parameters** | | | | |
| **VMAXRULE** | Integer | 0–31 | 30 | Specifies the maximum $V_{OD}$. The default value, 60, represents 1200 mV. |
| **VMINRULE** | Integer | 0–31 | 6 | Specifies the minimum $V_{OD}$. The default value, 9, represents 165 mV. |
| **VODMINRULE** | Integer | 0–31 | 14 | Specifies the minimum $V_{OD}$ for the first tap. The default value, 24, represents 440 mV. |
| **VPOSTRULE** | Integer | 0–25 | 25 | Specifies the maximum value that the internal algorithm for pre-emphasis will ever test in determining the optimum post-tap setting. |
| **VPRERULE** | Integer | 0–16 | 16 | Specifies the maximum value that the internal algorithm for pre-emphasis will ever test in determining the optimum pre-tap setting. |
| **PREMAINVAL** | Integer | 0–31 | 30 | Specifies the Preset $V_{OD}$ value. This value is set by the Preset command of the link training protocol, defined in Clause 72.6.10.2.3.1 of *IEEE Std 802.3ap–2007*. |
| **PREPOSTVAL** | Integer | 0–25 | 0 | Specifies the preset Post-tap value. |
| **PREPREVAL** | Integer | 0–16 | 0 | Specifies the preset Pre-tap value. |
| **INITMAINVAL** | Integer | 0–31 | 25 | Specifies the initial $V_{OD}$ value. This value is set by the Initialize command of the link training protocol, defined in Clause 72.6.10.2.3.2 of *IEEE Std 802.3ap–2007*. |

| Parameter | Type | Range | Default Setting | Parameter Description |
|---|---|---|---|---|
| **INITPOSTVAL** | Integer | 0–25 | 13 | Specifies the initial Post-tap value. |
| **INITPREVAL** | Integer | 0–16 | 3 | Specifies the initial Pre-tap value. |
| **Link Training: General** | | | | |
| **Enable Link Training** | Boolean | • True<br>• False | True | If this parameter is turned on, the IP core includes the link training module, which configures the remote link partner TX PMD for the lowest Bit Error Rate (BER). LT is defined in Clause 72 of *IEEE Std 802.3ap–2007*. |
| **Maximum bit error count** | Integer | $2^n - 1$ for n an integer in the range 4–10. | 511 | Specifies the maximum number of errors on a lane before the `Link Training Error` bit (40GBASE-KR4 register offset 0xD2, bit 4, 12, 20, or 28, depending on the lane) is set, indicating an unacceptable bit error rate.<br>n is the width of the Bit Error Counter that is configured in the IP core. The value to which you set this parameter determines n, and thus the width of the Bit Error Counter. Because the default value of this parameter is 511, the default width of the Bit Error Counter is 10 bits.<br>You can use this parameter to tune PMA settings. For example, if you see no difference in error rates between two different sets of PMA settings, you can increase the width of the bit error counter to determine if a larger counter enables you to distinguish between PMA settings. |
| **Number of frames to send before sending actual data** | Integer | • 127<br>• 255 | 127 | Specifies the number of additional training frames the local link partner delivers to ensure that the link partner can correctly detect the local receiver state. |
| **FEC Options** | | | | |
| **Include FEC sublayer** | Boolean | • True<br>• False | False | If this parameter is turned on, the IP core includes logic to implement FEC |
| **Set FEC_Ability bit on power up or reset** | Boolean | • True<br>• False | True | If this parameter is turned on, the IP core sets the FEC ability bit (40GBASE-KR4 register offset 0xB0, bit 16: `KR FEC enable`) on power up and reset.<br>This parameter is available if you turn on **Include FEC sublayer**. |

| Parameter | Type | Range | Default Setting | Parameter Description |
|---|---|---|---|---|
| **Set FEC_Enable bit on power up or reset** | Boolean | • True<br>• False | True | If this parameter is turned on, the IP core sets the FEC enable bit (40GBASE-KR4 register offset 0xB0, bit 18: `KR FEC request`) on power up and reset. If you turn on this parameter but do not turn on **Set FEC_ability bit on power up or reset**, this parameter has no effect: the IP core cannot specify the value of 1 for FEC Requested without specifying the value of 1 for FEC Ability.<br>This parameter is available if you turn on **Include FEC sublayer**. |

**Table 2-3: Low Latency 40-100GbE PHY Parameter Settings**

Lists the PHY parameters that are configured automatically based on parameter values you select in the Low Latency 40G/100G Ethernet parameter editor.

| Parameter | 40GbE Value<br>40GBASE-KR4 Value | 100GbE Value | 100GbE at CAUI–4 |
|---|---|---|---|
| Lanes | 4 | 10 | 4 |
| Data rate per lane | 10312.5 Mbps | 10312.5 Mbps | 25781.25 Mbps |
| Available PHY reference clock frequencies | 322.265625 MHz<br>644.53125 MHz | 322.265625 MHz<br>644.53125 MHz | 322.265625 MHz<br>644.53125 MHz |

**Related Information**

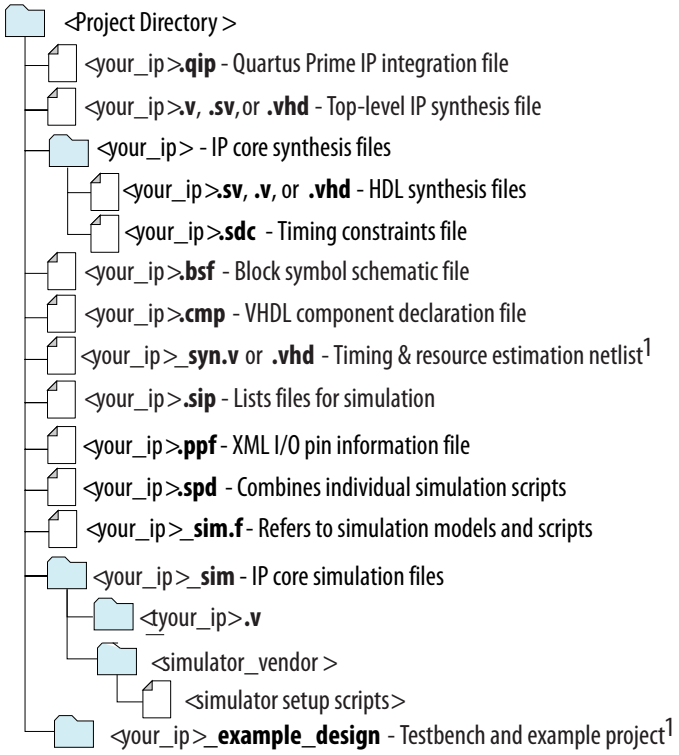
**Clocks** on page 3-58

The **PHY reference frequency** value is the required frequency of the transceiver reference clock or transceiver reference clocks.

# Files Generated for Stratix V Variations

The Quartus Prime software generates the following output for your Stratix V LL 40-100GbE IP core.

**Figure 2-3: IP Core Generated Files**




Notes:
1. If generated for your IP variation

# Files Generated for Arria 10 Variations

The Quartus Prime software generates the following IP core output file structure when targeting Arria 10 devices.

**Figure 2-4: IP Core Generated Files**

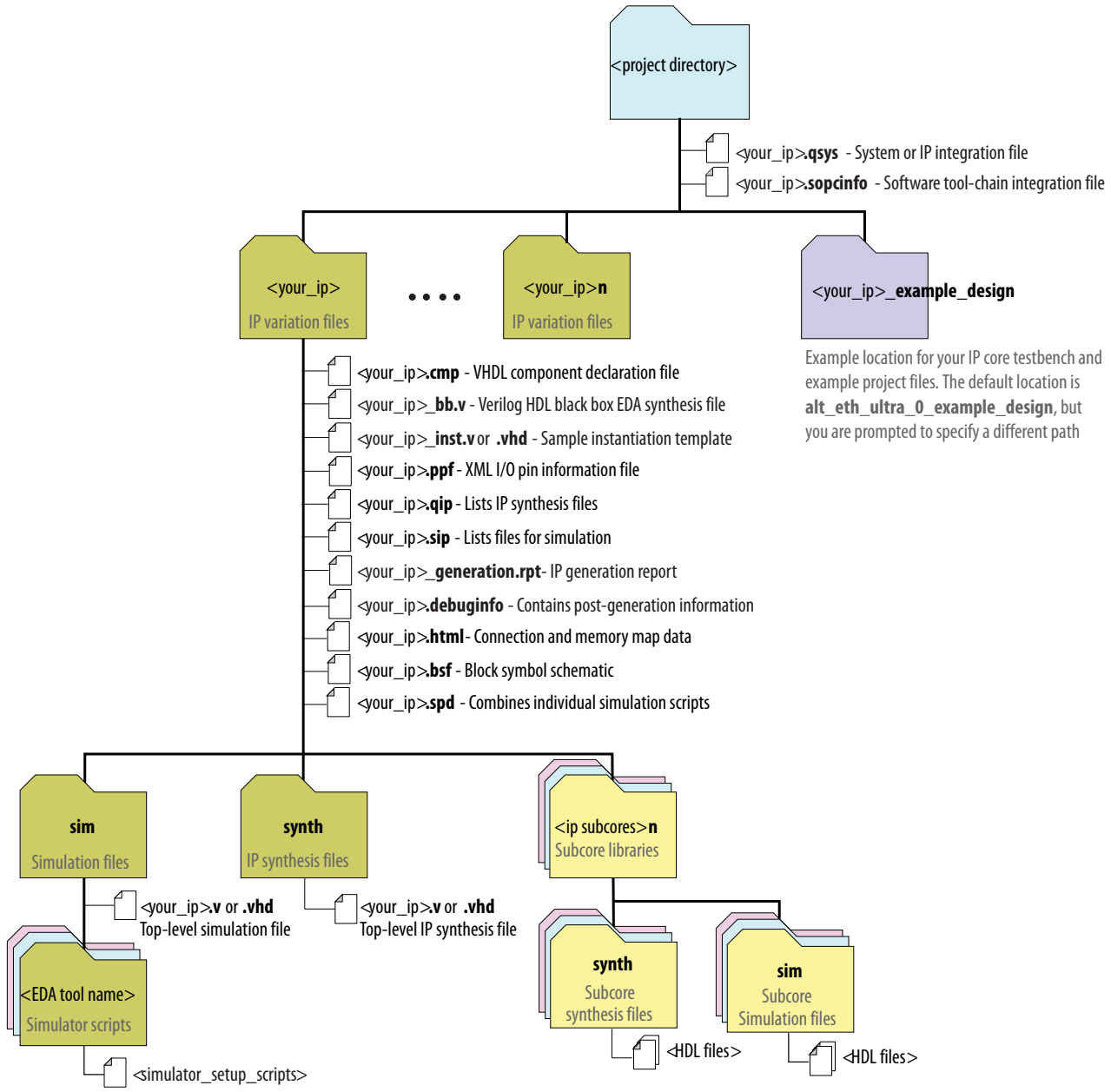



**Table 2-4: IP Core Generated Files (Arria 10 Variations)**

| File Name | Description |
|---|---|
| *<my_ip>*.qsys | The Qsys system or top-level IP variation file. *<my_ip>* is the name that you give your IP variation. |

| File Name | Description |
|---|---|
| `<system>.sopcinfo` | Describes the connections and IP component parameterizations in your Qsys system. You can parse its contents to get requirements when you develop software drivers for IP components.<br>Downstream tools such as the Nios II tool chain use this file. The `.sopcinfo` file and the `system.h` file generated for the Nios II tool chain include address map information for each slave relative to each master that accesses the slave. Different masters may have a different address map to access a particular slave component. |
| `<my_ip>.cmp` | The VHDL Component Declaration (**.cmp**) file is a text file that contains local generic and port definitions that you can use in VHDL design files. |
| `<my_ip>.html` | A report that contains connection information, a memory map showing the address of each slave with respect to each master to which it is connected, and parameter assignments. |
| `<my_ip>_generation.rpt` | IP or Qsys generation log file. A summary of the messages during IP generation. |
| `<my_ip>.debuginfo` | Contains post-generation information. Used to pass System Console and Bus Analyzer Toolkit information about the Qsys interconnect. The Bus Analysis Toolkit uses this file to identify debug components in the Qsys interconnect. |
| `<my_ip>.qip` | Contains all the required information about the IP component to integrate and compile the IP component in the Quartus Prime software. |
| `<my_ip>.csv` | Contains information about the upgrade status of the IP component. |
| `<my_ip>.bsf` | A Block Symbol File (**.bsf**) representation of the IP variation for use in Quartus Prime Block Diagram Files (**.bdf**). |
| `<my_ip>.spd` | Required input file for `ip-make-simscript` to generate simulation scripts for supported simulators. The **.spd** file contains a list of files generated for simulation, along with information about memories that you can initialize. |
| `<my_ip>.ppf` | The Pin Planner File (**.ppf**) stores the port and node assignments for IP components created for use with the Pin Planner. |
| `<my_ip>_bb.v` | You can use the Verilog black-box (**_bb.v**) file as an empty module declaration for use as a black box. |
| `<my_ip>.sip` | Contains information required for NativeLink simulation of IP components. You must add the **.sip** file to your Quartus Prime project. |
| `<my_ip>_inst.v` or `_inst.vhd` | HDL example instantiation template. You can copy and paste the contents of this file into your HDL file to instantiate the IP variation. |

| File Name | Description |
|---|---|
| `<my_ip>.regmap` | If IP contains register information, **.regmap** file generates. The **.regmap** file describes the register map information of master and slave interfaces. This file complements the **.sopcinfo** file by providing more detailed register information about the system. This enables register display views and user customizable statistics in the System Console. |
| `<my_ip>.svd` | Allows HPS System Debug tools to view the register maps of peripherals connected to HPS within a Qsys system.<br>During synthesis, the **.svd** files for slave interfaces visible to System Console masters are stored in the **.sof** file in the debug section. System Console reads this section, which Qsys can query for register map information. For system slaves, Qsys can access the registers by name. |
| `<my_ip>.v`<br>or<br>`<my_ip>.vhd` | HDL files that instantiate each submodule or child IP core for synthesis or simulation. |
| `mentor/` | Contains a ModelSim® script `msim_setup.tcl` to set up and run a simulation. |
| `aldec/` | Contains a Riviera-PRO script `rivierapro_setup.tcl` to setup and run a simulation. |
| `/synopsys/vcs`<br>`/synopsys/vcsmx` | Contains a shell script `vcs_setup.sh` to set up and run a VCS® simulation.<br>Contains a shell script `vcsmx_setup.sh` and `synopsys_sim.setup` file to set up and run a VCS MX® simulation. |
| `/cadence` | Contains a shell script `ncsim_setup.sh` and other setup files to set up and run an NCSIM simulation. |
| `/submodules` | Contains HDL files for the IP core submodule. |
| `<child IP cores>/` | For each generated child IP core directory, Qsys generates `/synth` and `/sim` sub-directories. |

# Integrating Your IP Core in Your Design

When you integrate your IP core instance in your design, you must pay attention to the following items:

**Pin Assignments** on page 2-19

**External Transceiver Reconfiguration Controller Required in Stratix V Designs** on page 2-19

**Transceiver PLL Required in Arria 10 Designs** on page 2-20

**External Time-of-Day Module for Variations with 1588 PTP Feature** on page 2-22

**Clock Requirements for 40GBASE-KR4 Variations** on page 2-23

**External TX MAC PLL** on page 2-23

**Placement Settings for the Low Latency 40-100GbE IP Core** on page 2-23

**Related Information**

- **Low Latency 40G Ethernet Example Design User Guide**
- **Low Latency 100G Ethernet Example Design User Guide**

## Pin Assignments

When you integrate your Low Latency 40-100GbE IP core instance in your design, you must make appropriate pin assignments. You can create a virtual pin to avoid making specific pin assignments for top-level signals while you are simulating and not ready to map the design to hardware.

For the Arria 10 device family, you must configure a transceiver PLL that is external to the 40-100GbE IP core. The transceiver PLLs you configure are physically present in the device transceivers, but the LL 40-100GbE IP core does not configure and connect them. The required number of transceiver PLLs depends on the distribution of your Ethernet data pins in the different Arria 10 transceiver blocks.

**Related Information**

- **Transceiver PLL Required in Arria 10 Designs** on page 2-20
- **Quartus Prime Help**
  For information about the Quartus Prime software, including virtual pins and the IP Catalog.

## External Transceiver Reconfiguration Controller Required in Stratix V Designs

Low Latency 40-100GbE IP cores that target Stratix V devices require an external reconfiguration controller to compile and to function correctly in hardware. Low Latency 40-100GbE IP cores that target Arria 10 devices include a reconfiguration controller block in the PHY component and do not require an external reconfiguration controller.

You can use the IP Catalog to generate an Altera Transceiver Reconfiguration Controller.

When you configure the Altera Transceiver Reconfiguration Controller, you must specify the number of reconfiguration interfaces. The number of reconfiguration interfaces required for the Low Latency 40GbE and 100GbE IP cores depends on the IP core variation.

**Table 2-5: Number of Reconfiguration Interfaces**

Lists the number of reconfiguration interfaces you should specify for the Altera Transceiver Reconfiguration Controller for your Stratix V Low Latency 40-100GbE IP core. Low Latency 40-100GbE IP cores that target Arria 10 devices include a reconfiguration controller block in the PHY component and do not require any external reconfiguration controllers.

| PHY Configuration | Number of Reconfiguration Interfaces |
|---|---|
| LL 40GbE (4x10.3125 lanes) | 8 |
| LL 100GbE (10x10.3125 lanes) | 20 |

You can configure your reconfiguration controller with additional interfaces if your design connects with multiple transceiver IP cores. You can leave other options at the default settings or modify them for your preference.

You should connect the `reconfig_to_xcvr`, `reconfig_from_xcvr`, and `reconfig_busy` ports of the Low Latency 40-100GbE IP core to the corresponding ports of the reconfiguration controller.

You must also connect the `mgmt_clk_clk` and `mgmt_rst_reset` ports of the Altera Transceiver Reconfiguration Controller. The `mgmt_clk_clk` port must have a clock setting in the range of 100–125MHz; this setting can be shared with the Low Latency 40-100GbE IP core `clk_status` port. The `mgmt_rst_reset` port must be deasserted before, or deasserted simultaneously with, the Low Latency 40-100GbE IP core `reset_async` port.

Refer to the example project for RTL that connects the Altera transceiver reconfiguration controller to the IP core..

**Table 2-6: External Altera Transceiver Reconfiguration Controller Ports for Connection to Low Latency 40-100GbE IP Core**

| Signal Name | Direction | Description |
|---|---|---|
| `reconfig_to_xcvr[559:0]`(40GbE)<br>`reconfig_to_xcvr[1399:0]`(100GbE) | Input | The Low Latency 40-100GbE IP core reconfiguration controller to transceiver port in Stratix V devices. |
| `reconfig_from_xcvr[367:0]`(40GbE)<br>`reconfig_from_xcvr[919:0]`(100GbE) | Output | The Low Latency 40-100GbE IP core reconfiguration controller from transceiver port in Stratix V devices. |
| `reconfig_busy` | Input | Indicates the reconfiguration controller is still in the process of reconfiguring the transceiver. |

**Related Information**

**Altera Transceiver PHY IP Core User Guide**
For more information about the Altera Transceiver Reconfiguration Controller.

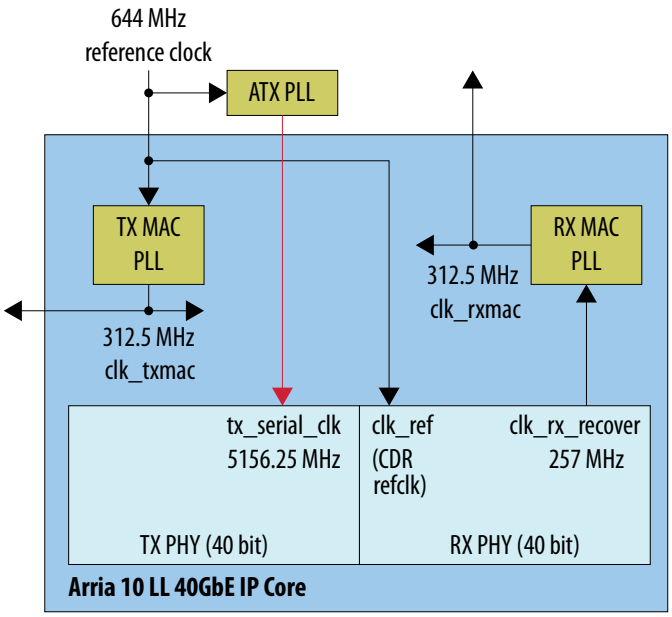
## Transceiver PLL Required in Arria 10 Designs

Low Latency 40-100GbE IP cores that target Arria 10 devices require an external TX transceiver PLL to compile and to function correctly in hardware.

**Figure 2-5: PLL Configuration Example**

In this example, the TX transceiver PLL is instantiated with an Altera ATX PLL IP core. The TX transceiver PLL must always be instantiated outside the LL 40-100GbE IP core.

In this example, **Use external TX MAC PLL** is turned off. Therefore, the TX MAC PLL is in the IP core. If you turn on the **Use external TX MAC PLL** parameter you must also instantiate and connect a TX MAC PLL outside the LL 40-100GbE IP core.




You can use the IP Catalog to create a transceiver PLL.

- Select **Arria 10 Transceiver ATX PLL** or **Arria 10 Transceiver CMU PLL**.
- In the parameter editor, set the following parameter values:
  - **PLL output frequency** to **5156.25 MHz** for standard LL 40-100GbE IP core variations or to **12890.625 MHz** for CAUI-4 variations. The transceiver performs dual edge clocking, using both the rising and falling edges of the input clock from the PLL. Therefore, this PLL output frequency setting supports a 10.3125 or 25.78125 Gbps data rate through the transceiver.
  - **PLL reference clock frequency** to the value you specified for the **PHY reference frequency** parameter.

When you generate a Low Latency 40-100GbE IP core, the software also generates the HDL code for an ATX PLL, in the file *<variation_name>*`/arria10_atx_pll.v`. However, the HDL code for the Low Latency 40-100GbE IP core does not instantiate the ATX PLL. If you choose to use the ATX PLL provided with the Low Latency 40-100GbE IP core, you must instantiate and connect the instances of the ATX PLL with the LL 40-100GbE IP core in user logic.

**Note:** If your Arria 10 design includes multiple instances of the LL 40-100GbE IP core, do not use the ATX PLL HDL code provided with the IP core. Instead, generate new TX PLL IP cores to connect in your design.

The number of external PLLs you must generate or instantiate depends on the distribution of your Ethernet TX serial lines across physical transceiver channels and banks. You specify the clock network to which each PLL output connects by setting the clock network in the PLL parameter editor. The example

project demonstrates one possible choice, which is compatible with the ATX PLL provided with the LL 40-100GbE IP core.

You must connect the `tx_serial_clk` input pin for each Low Latency 40-100GbE IP core PHY link to the output port of the same name in the corresponding external PLL. You must connect the `pll_locked` input pin of the Low Latency 40-100GbE IP core to the logical AND of the `pll_locked` output signals of the external PLLs for all of the PHY links.

User logic must provide the AND function and connections. Refer to the example project for example working user logic including one correct method to instantiate and connect an external PLL.

**Related Information**

- **External Transceiver PLL** on page 3-28
- **Pin Assignments** on page 2-19
- **Arria 10 Transceiver PHY User Guide**
  Information about the correspondence between PLLs and transceiver channels, and information about how to configure an external transceiver PLL for your own design. You specify the clock network to which the PLL output connects by setting the clock network in the PLL parameter editor.

## External Time-of-Day Module for Variations with 1588 PTP Feature

Low Latency 40-100GbE IP cores that include the 1588 PTP module require an external time-of-day (TOD) module to provide a continuous flow of current time-of-day information. The TOD module must update the time-of-day output value on every clock cycle, and must provide the TOD value in the V2 format (96 bits) or the 64-bit TOD format, or both..

The example project you can generate for your IP core PTP variation includes a TOD module, implemented as two distinct, simple TOD modules, one connected to the TX MAC and one connected to the RX MAC.

### Table 2-7: TOD Module Required Connections

Required connections for TOD module, listed using signal names for TOD modules that provide both a 96-bit TOD and a 64-bit TOD. If you create your own TOD module it must have the output signals required by the LL 40-100GbE IP core. However, its signal names could be different than the TOD module signal names in the table. The signals that that the IP core includes depend on the **Enable 96b Time of Day Format** and **Enable 64b Time of Day Format** parameters. For example, an RX TOD module might require only a 96-bit TOD out signal.

| TOD Module Signal | LL 40-100GbE IP Core Signal |
|---|---|
| `rst_txmac` (input) | Drive this signal from the same source as the `reset_async` input signal to the LL 40-100GbE IP core. |
| `rst_rxmac` (input) | Drive this signal from the same source as the `reset_async` input signal to the LL 40-100GbE IP core. |
| `tod_txmclk_96b[95:0]` (output) | `tx_time_of_day_96b_data[95:0]` (input) |
| `tod_txmclk_64b[63:0]` (output) | `tx_time_of_day_64b_data[63:0]` (input) |
| `tod_rxmclk_96b[95:0]` (output) | `rx_time_of_day_96b_data[95:0]` (input) |
| `tod_rxmclk_64b[63:0]` (output) | `rx_time_of_day_64b_data[63:0]` (input) |
| `clk_txmac` (input) | `clk_txmac` (output) |
| `clk_rxmac` (input) | `clk_rxmac` (output) |

**Related Information**

- **PTP Timestamp and TOD Formats** on page 3-45
- **External Time-of-Day Module for 1588 PTP Variations** on page 3-45
- **1588 Precision Time Protocol Interfaces** on page 3-39

## Clock Requirements for 40GBASE-KR4 Variations

In 40GBASE-KR4 IP core designs, you must drive the clocks for the two IP core register interfaces (`reconfig_clk` and `clk_status`) from the same clock source.

## External TX MAC PLL

If you turn on **Use external TX MAC PLL** in the LL 40-100GbE parameter editor, you must connect the `clk_txmac_in` input port to a clock source, usually a PLL on the device.

The `clk_txmac_in` signal drives the `clk_txmac` clock in the IP core TX MAC and PHY. If you turn off this parameter, the `clk_txmac_in` input clock signal is not available.

The required TX MAC clock frequency is 312.5 MHz for 40GbE variations, and 390.625 MHz for 100GbE variations. User logic must drive `clk_txmac_in` from a PLL whose input is the PHY reference clock, `clk_ref`.

## Placement Settings for the Low Latency 40-100GbE IP Core

The Quartus Prime software provides the options to specify design partitions and LogicLock™ regions for incremental compilation, to control placement on the device. To achieve timing closure for your design, you might need to provide floorplan guidelines using one or both of these features.

The appropriate floorplan is always design-specific, and depends on your full design.

**Related Information**

**Quartus Prime Standard Edition Handbook Volume 2: Design Implementation and Optimization**
Describes incremental compilation, design partitions, and LogicLock regions.

# Low Latency 40-100GbE IP Core Testbenches

Altera provides a testbench , a hardware design example, and a compilation-only example design with most variations of the Low Latency 40-100GbE IP core. The testbench is available for simulation of your IP core, and the hardware design example can be run on hardware. You can run the testbench to observe the IP core behavior on the various interfaces in simulation.

Altera offers testbenches for all Avalon-ST client interface variations that generate their own TX MAC clock(**Use external TX MAC PLL** is turned off).

Currently, the IP core can generate a testbench and example project for variations that use an external TX MAC PLL, but these testbenches and example projects do not function correctly. Non-functional testbenches and example projects provide an example of the connections you must create in your design to ensure the LL 40-100GbE IP core functions correctly. However, you cannot simulate them nor run them in hardware.

To generate the testbench, in the Low Latency 40-100GbE parameter editor, you must first set the parameter values for the IP core variation you intend to generate. If you do not set the parameter values identically, the testbench you generate might not exercise the IP core variation you generate. If your IP core variation does not meet the criteria for a testbench, the generation process does not create a testbench (with the exception of the non-functional testbench generated if an IP core requires an external TX MAC clock signal).

You can simulate the testbench that you generate with your IP core variation. The testbench illustrates packet traffic, in addition to providing information regarding the transceiver PHY. The 40GBASE-KR4 testbench exercises auto-negotiation and link training, in addition to generating and checking packet traffic.

**Related Information**

- **Generating the Low Latency 40-100GbE Testbench** on page 2-29
- **Simulating the Low Latency 40-100GbE IP Core With the Testbenches** on page 2-28
  Instructions to simulate the 40GbE or 100GbE IP core with the IP core appropriate testbench you can generate, including simulation parameters and supported simulators.

## Low Latency 40-100GbE IP Core Testbench Overview

The non-40GBASE-KR4 testbenches send traffic through the IP core in transmit-to-receive loopback mode, exercising the transmit side and receive side of the IP core in the same data flow. These testbenches send traffic to allow the Ethernet lanes to lock, and then send packets to the transmit client data interface and check the data as it returns through the receive client data interface.

**Figure 2-6: Low Latency 40-100GbE non-40GBASE-KR4 IP Core Testbench**

Illustrates the top-level modules of the Low Latency 40GbE and 100GbE example testbenches.

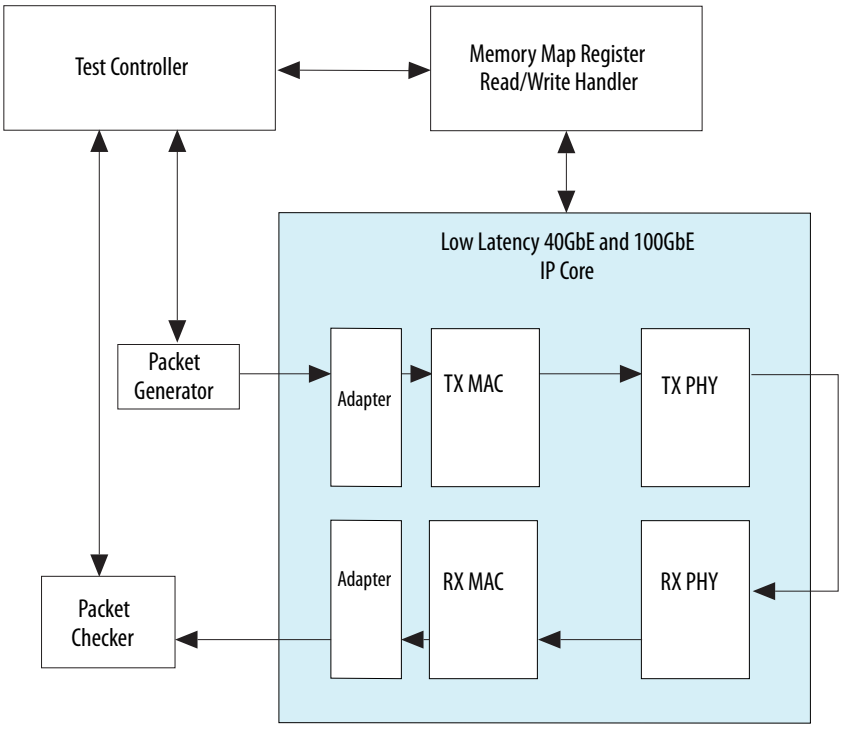


**Figure 2-7: 40GBASE-KR4 LL 40GbE IP Core Testbench**

Illustrates the top-level modules of the LL 40GBASE-KR4 example testbench. To support the simulation of auto-negotiation, the testbench uses two instances of the IP core instead of configuring the IP core in loopback mode.

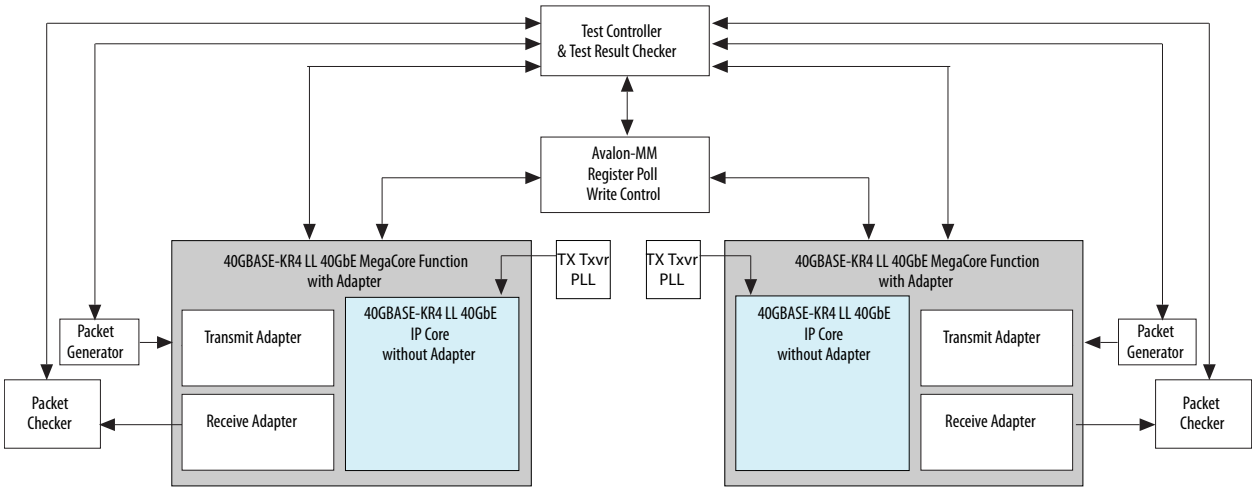



**Table 2-8: Low Latency 40-100GbE IP Core Testbench File Descriptions**

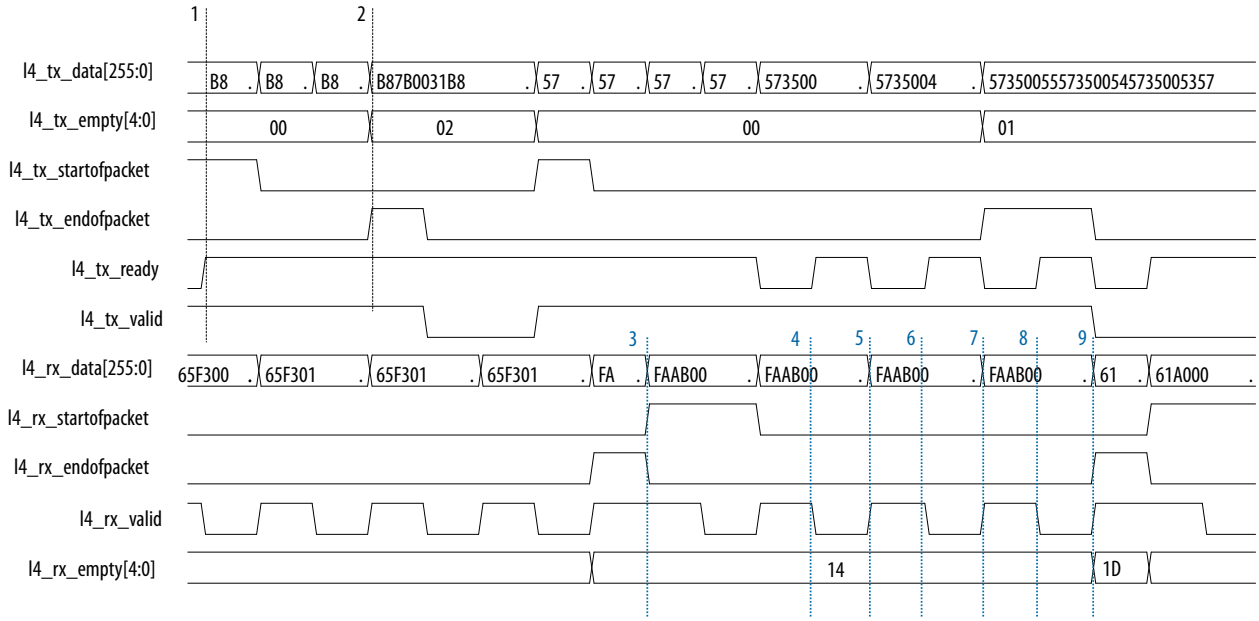
Lists the key files that implement the example testbenches.

| File Names | Description |
|---|---|
| Testbench and Simulation Files | |
| `basic_avl_tb_top.v` | Top-level testbench file for non-40GBASE-KR4 variations. The testbench instantiates the DUT and runs Verilog HDL tasks to generate and accept packets. |
| `alt_e40_avalon_kr4_tb.sv` | Top-level testbench file for 40GBASE-KR4 variations. |
| `alt_e40_avalon_tb_packet_gen.v`, `alt_e40_avalon_tb_packet_gen_sanity_check.v`, `alt_e40_stat_cntr_1port.v` | Packet generator and checkers for 40GBASE-KR4 variations. |
| Testbench Scripts | |
| `run_vsim.do` | The ModelSim script to run the testbench. |
| `run_vcs.sh` | The Synopsys VCS script to run the testbench. |
| `run_ncsim.sh` | The Cadence NCSim script to run the testbench. |

**Figure 2-8: Typical 40GbE Traffic on the Avalon-ST Interface**

Shows typical traffic from the simulation testbench created using the `run_vsim.do` script in ModelSim. In this case the IP core is a 40GbE IP core with adapters. In Stratix V variations the script is found in *<instance_name>*`_example_design/alt_eth_ultra/example_testbench/run_vsim.do` and in Arria 10 variations the script is found in *<example_design_directory>*`/example_testbench/run_vsim.do`.

**Note:** Client logic must maintain the `l4_tx_valid` signal asserted while asserting SOP, through the assertion of EOP. Client logic should not pull this signal low during a packet transmission.




The markers in the figure show the following sequence of events:

1. At marker 1, the application asserts `l4_tx_startofpacket`, indicating the beginning of a TX packet.
2. At marker 2, the application asserts `l4_tx_endofpacket`, indicating the end of the TX packet. The value on `l4_tx_empty[4:0]` indicates that the 2 least significant bytes of the last data cycle are empty.
3. At marker 3, the IP core asserts `l4_rx_startofpacket`, indicating the beginning of an RX packet. A second transfer has already started on the TX bus.
4. At marker 4, the 40GbE IP core deasserts `l4_rx_valid`, indicating that the IP core does not have new valid data to send to the client on `l4_rx_data[255:0]`. `l4_rx_data[255:0]` remains unchanged for a second cycle, but because the `l4_rx_valid` signal is deasserted, the client should ignore the value on the RX signals.
5. A marker 5, the 40GbE IP core asserts `l4_rx_valid`, indicating that it has valid data to send to the client on `l4_rx_data[255:0]`.
6. At marker 6, the 40GbE IP core deasserts `l4_rx_valid`, indicating that it does not have new valid data to send to the client on `l4_rx_data[255:0]`. `l4_rx_data[255:0]` remains unchanged for a second cycle.

7. At marker 7, the 40GbE IP core asserts `l4_rx_valid`, indicating that it has valid data to send to the client on `l4_rx_data[255:0]`.
8. At marker 8, the 40GbE IP core deasserts `l4_rx_valid`, indicating that the 40GbE IP core does not have new valid data to send to the client on `l4_rx_data[255:0]`. `l4_rx_data[255:0]` remains unchanged for a second cycle.
9. At marker 9, the IP core asserts `l4_rx_endofpacket`, indicating the end of the RX packet. `l4_rx_empty[4:0]` has a value of 0x1D, indicating that 29 least significant bytes of the last cycle of the RX packet empty.

**Note:** The ready latency on the Avalon-ST TX client interface is 0.

**Related Information**

**Avalon Interface Specifications**
For more information about the Avalon-ST protocol.

## Understanding the Testbench Behavior

The non-40GBASE-KR4 testbenches send traffic through the IP core in transmit-to-receive loopback mode, exercising the transmit side and receive side of the IP core in the same data flow. These testbenches send traffic to allow the Ethernet lanes to lock, and then send packets to the transmit client data interface and check the data as it returns through the receive client data interface.

The 40GBASE-KR4 testbench sends traffic through the two IP cores in each direction, exercising the receive and transmit sides of both IP cores. This testbench exercises auto-negotiation and link training, and then sends and checks packets in data mode.

The Low Latency 40-100GbE IP core implements virtual lanes as defined in the*IEEE 802.3ba-2010 40G and 100G Ethernet Standard*. The 40GbE IP cores are fixed at four virtual lanes and each lane is sent over a 10 Gbps physical lane. The 100GbE IP cores are fixed at 20 virtual lanes; the 20 virtual lanes are typically bit-interleaved over ten 10-Gbps physical lanes. When the lanes arrive at the receiver the lane streams are in an undefined order. Each lane carries a periodic PCS-VLANE alignment tag to restore the original ordering. The simulation establishes a random permutation of the physical lanes that is used for the remainder of the simulation.

Within each virtual lane stream, the data is 64B/66B encoded. Each word has two framing bits which are always either 01 or 10, never 00 or 11. The RX logic uses this pattern to lock onto the correct word boundaries in each serial stream. The process is probabilistic due to false locks on the pseudo-random scrambled stream. To reduce hardware costs, the receiver does not test alignments in parallel; consequently, the process can be somewhat time-consuming in simulation.

In the 40GBASE-KR4 testbench, some register values are set to produce a shorter runtime. For example, timeout counters and the number of steps used in link training are set to smaller values than would be prudent in hardware. To override this behavior and use the normal settings in simulation, add the following line to your IP core variation top-level file or to the testbench top-level file, `alt_e40_avalon_kr4_tb.sv`:

```
`define ALTERA_RESERVED_XCVR_FULL_KR_TIMERS
```

Both the word lock and the alignment marker lock implement hysteresis as defined in the *IEEE 802.3ba-2010 40G and 100G Ethernet Standard*. Multiple successes are required to acquire lock and multiple failures are required to lose lock. The “fully locked” messages in the simulation log indicate the point at which a physical lane has successfully identified the word boundary and virtual lane assignment.

In the event of a catastrophic error, the RX PCS automatically attempts to reacquire alignment. The MAC properly identifies errors in the datastream.

# Simulating the Low Latency 40-100GbE IP Core With the Testbenches

You can simulate the Low Latency 40-100GbE IP core using the Altera-supported versions of the Mentor Graphics ModelSim SE, Cadence NCSim, and Synopsys VCS simulators for the current version of the Quartus Prime software. The ModelSim-AE simulator does not have the capacity to simulate this IP core.

The example testbenches simulate packet traffic at the digital level. The testbenches do not require special SystemVerilog class libraries.

The top-level testbench file for non-40GBASE-KR4 variations consists of a simple packet generator and checker and one IP core in a loopback configuration.

The top-level testbench file for 40GBASE-KR4 variations consists of a symmetric arrangement with two IP cores and traffic between them. For each IP core there is a packet generator to send traffic on the TX side of the IP core and a packet checker to check the packets it receives from the other IP core. The two IP cores communicate with each other through their Ethernet link, in which the testbench injects random skew. The 40GBASE-KR4 testbench connects each IP core to a transceiver TX PLL, and exercises auto-negotiation, link training, and data mode.

The example testbenches contain the test files and run scripts for the ModelSim, Cadence, and Synopsys simulators. The run scripts use the file lists in the wrapper files. When you launch a simulation from the original directory, the relative filenames in the wrapper files allow the run script to locate the files correctly. When you generate the testbench for a Low Latency 40-100GbE IP core that targets an Arria 10 device, the software generates a copy of the IP core variation with a specific relative path from the testbench scripts.

The following sections provide directions for generating the testbench and running tests with the ModelSim, Cadence, and Synopsys simulators.

**Generating the Low Latency 40-100GbE Testbench** on page 2-29

**Optimizing the Low Latency 40-100GbE IP Core Simulation With the Testbenches** on page 2-30

**Simulating with the Modelsim Simulator** on page 2-30

**Simulating with the NCSim Simulator** on page 2-31

**Simulating with the VCS Simulator** on page 2-31

**Testbench Output Example: Low Latency 40-100GbE IP Core** on page 2-31

**Related Information**

- **Low Latency 40G Ethernet Example Design User Guide**
- **Low Latency 100G Ethernet Example Design User Guide**
- **Low Latency 40-100GbE IP Core Testbenches** on page 2-23

Altera provides a testbench , a hardware design example, and a compilation-only example design with most variations of the Low Latency 40-100GbE IP core. The testbench is available for simulation of your IP core, and the hardware design example can be run on hardware. You can run the testbench to observe the IP core behavior on the various interfaces in simulation.

# Generating the Low Latency 40-100GbE Testbench

A single procedure generates both the testbench and the example project. The procedure varies depending on your target device. To generate the testbench and example project:

1. Follow the steps in **Specifying the Low Latency 40-100GbE IP Core Parameters and Options** on page 2-4 to parameterize your IP core.
2. If your IP core variation targets a Stratix V device, go to step 4.
3. If your IP core variation targets an Arria 10 device, in the Low Latency 40-100GbE parameter editor, click the **Generate Example Design** button to generate the testbench and example project for the IP core variation you intend to generate.

   **Tip:** You are prompted to locate the new testbench and example project in the directory *<working directory>*/alt_eth_ultra_0_example_design. You can accept the default path or modify the path to the new testbench and example project.
4. Generate the IP core by clicking **Generate HDL** for Arria 10 variations or **Generate** for Stratix V variations.

   **Note:** If your IP core variation targets a Stratix V device, when prompted at the start of generation, you must turn on **Generate example design**. Turning on **Generate example design** is the only process that generates a functional testbench and a functional example project for Stratix V variations.

   When the IP core is generated in *<working directory>*, the testbench and example project are generated in different locations depending on the device family your IP core variation targets.

   - For Stratix V variations, the testbench and example project are generated in *<working directory>*/*<IP core variation>*_example_design/alt_eth_ultra.
   - For Arria 10 variations, the testench and the compilation-only and hardware design examples are generated in the directory you specify in Step 2. If you do not modify the location text at the prompt, they are generated in *<working directory>*/alt_eth_ultra_0_example_design.

The directory with the testbench and design example has four subdirectories for Arria 10 variations and three subdirectories for Stratix V variations:

- example_testbench
- compilation_test_design
- hardware_test_design
- ex_40g, ex_100g, or ex_100g_caui4, for Arria 10 variations only

The ex_40g, ex_100g, or ex_100g_caui4 directory contains a copy of the IP core variation. The testbench and design examples (compilation-only and hardware design example) for your Arria 10 IP core variation connect to the copy in this directory rather than to the copy you generate in *<working directory>*.

**Related Information**

- **Low Latency 40-100GbE IP Core Testbenches** on page 2-23
  Altera provides a testbench , a hardware design example, and a compilation-only example design with most variations of the Low Latency 40-100GbE IP core. The testbench is available for simulation of your IP core, and the hardware design example can be run on hardware. You can run the testbench to observe the IP core behavior on the various interfaces in simulation.
- **Low Latency 40G Ethernet Example Design User Guide**
  Provides a demonstration on IP generation, simulation, compliation, and hardware testing.
- **Low Latency 100G Ethernet Example Design User Guide**
  Provides a demonstration on IP generation, simulation, compliation, and hardware testing.

# Optimizing the Low Latency 40-100GbE IP Core Simulation With the Testbenches

In the testbench file for non-40GBASE-KR4 variations, you can set the `FASTSIM` parameter and force values in the `RO_SELS`, `BPOS`, and `WPOS` parameters to enable simulation optimization. The testbench also specifies another IP core simulation optimization setting that you might need to modify for simulation in your own environment: `AM_CNT_BITS`.

To derive the values to which to force the `RO_SELS`, `BPOS`, and `WPOS` parameters, you must run your first simulation with some additional testbench code to display the simulation-derived values. For subsequent simulation runs with the same hardware design and simulator, you can force the values to these simulation-derived values to avoid the lengthy simulation required to achieve lane alignment. The process is particularly slow for this IP core because the IP core includes a soft processor that drives the lane alignment process. Forcing the parameters to the correct values for your design and simulator bypasses this process, increasing simulation efficiency.

The testbench file generated with the Low Latency 40-100GbE IP core includes an `initial` block that displays the required force values. You can view the appropriate `initial` block for your IP core variation in the top-level testbench file you generate with your example design.

When you run simulation, this initial block prints values for the three parameters after lane alignment. Copy the values from standard output or from your log file and add the following lines to the testbench file, overwriting other forced values for these parameters if necessary:

```
defparam dut.top_inst.FASTSIM = 1;
defparam dut.<variation_name>_inst.FORCE_BPOS = <required BPOS value>;
defparam dut.<variation_name>_inst.FORC_WPOS  = <required WPOS value>;
defparam dut.<variation_name>_inst.FORCE_RO_SELS = <required RO_SELS value>;
```

**Note:** Whether you use the Altera-provided testbench or your own custom testbench, you must update the testbench file with these lines and the derived values. The testbench file generated with the Low Latency 40-100GbE IP core does not include the correct values, which depend on your IP core variation and simulation tool.

The `AM_CNT_BITS` parameter specifies the interval between expected alignment markers. The default value of this parameter is 14; this value specifies that alignment markers are inserted every $2^{14}$ blocks, in compliance with the Ethernet specification. However, the testbench sets this parameter to the value of 6, to speed up simulation. In your own simulation environment, you must set this parameter to match the interval between incoming alignment markers to the IP core.

## Simulating with the Modelsim Simulator

To run the simulation in the supported versions of the ModelSim simulation tool, follow these steps:

1. Change directory to the `<example_design_install_dir>/example_testbench` directory.
2. In the command line, type: `vsim -c -do run_vsim.do`

The example testbench will run and pass.

The ModelSim-AE simulator does not have the capacity to simulate this IP core.

**Related Information**

**Optimizing the Low Latency 40-100GbE IP Core Simulation With the Testbenches** on page 2-30
Instructions to improve simulation performance.

## Simulating with the NCSim Simulator

To run the simulation in the supported versions of the NCSim simulation tool, follow these steps:

1. Change directory to the `<example_design_install_dir>/example_testbench` directory.
2. In the command line, type: `sh run_ncsim.sh`

   The example testbench will run and pass.

**Related Information**

**Optimizing the Low Latency 40-100GbE IP Core Simulation With the Testbenches** on page 2-30
Instructions to improve simulation performance.

## Simulating with the VCS Simulator

To run the simulation in the supported versions of the VCS simulation tool, follow these steps:

1. Change directory to the `<example_design_install_dir>/example_testbench` directory.
2. In the command line, type: `sh run_vcs.sh`

The example testbench will run and pass.

**Related Information**

**Optimizing the Low Latency 40-100GbE IP Core Simulation With the Testbenches** on page 2-30
Instructions to improve simulation performance.

## Testbench Output Example: Low Latency 40-100GbE IP Core

This section shows successful simulation using the Low Latency 40-100GbE IP core testbench (*<variation_name>*`_example_design/alt_eth_ultra/example_testbench/basic_avl_tb_top.v` for Stratix V variations, or *<example_design_directory>*`/example_testbench/basic_avl_tb_top.v` for Arria 10 variations). The testbench connects the Ethernet TX lanes to the Ethernet RX lanes, so that the IP core is in an external loopback configuration. In simulation, the testbench resets the IP core and waits for lane alignment and deskew to complete successfully. The packet generator sends ten packets on the Ethernet TX lanes and the packet checker checks the packets when the IP core receives them on the Ethernet RX lanes.

The successful testbench run displays the following output:

```
# ****************************************
# ** Starting TX traffic...
# **
# **
# ** Sending Packet           1...
# ** Sending Packet           2...
# ** Sending Packet           3...
# ** Sending Packet           4...
# ** Sending Packet           5...
# ** Sending Packet           6...
# ** Sending Packet           7...
# ** Sending Packet           8...
# ** Sending Packet           9...
# ** Sending Packet          10...
# ** Received Packet          1...
# ** Received Packet          2...
# ** Received Packet          3...
# ** Received Packet          4...
# ** Received Packet          5...
```

```
# ** Received Packet          6...
# ** Received Packet          7...
# ** Received Packet          8...
# ** Received Packet          9...
# ** Received Packet         10...
# **
# ** Testbench complete.
# **
# ****************************************
```

## Compiling the Full Design and Programming the FPGA

You can use the **Start Compilation** command on the Processing menu in the Quartus Prime software to compile your design. After successfully compiling your design, program the targeted Altera device with the Programmer and verify the design in hardware.

**Related Information**

- **Incremental Compilation for Hierarchical and Team-Based Design**
- **Programming Altera Devices**

## Initializing the IP Core

The testbench initializes the IP core. However, when you simulate or run your own design in hardware, you must implement the initialization steps yourself.

To initialize the 40-100GbE IP core in your own design, follow these steps:

1. Drive the clock ports.
2. Reset the IP core.

**Related Information**

- **Clocks** on page 3-58
  In step 1, drive the input clock ports as specified here.
- **Resets** on page 3-61
  In step 2, reset the IP core.

# Functional Description 3

2017.12.28

UG-01172 Subscribe Send Feedback

This chapter provides a detailed description of the Low Latency 40-100GbE IP core. The chapter begins with a high-level overview of typical Ethernet systems and then provides detailed descriptions of the MAC, transmit (TX) and receive (RX) datapaths, signals, register descriptions, and an Ethernet glossary. This chapter includes the following sections:

**High Level System Overview** on page 3-2

**Low Latency 40-100GbE MAC and PHY Functional Description** on page 3-2

**Signals** on page 3-62

**Software Interface: Registers** on page 3-72

**Ethernet Glossary** on page 3-116

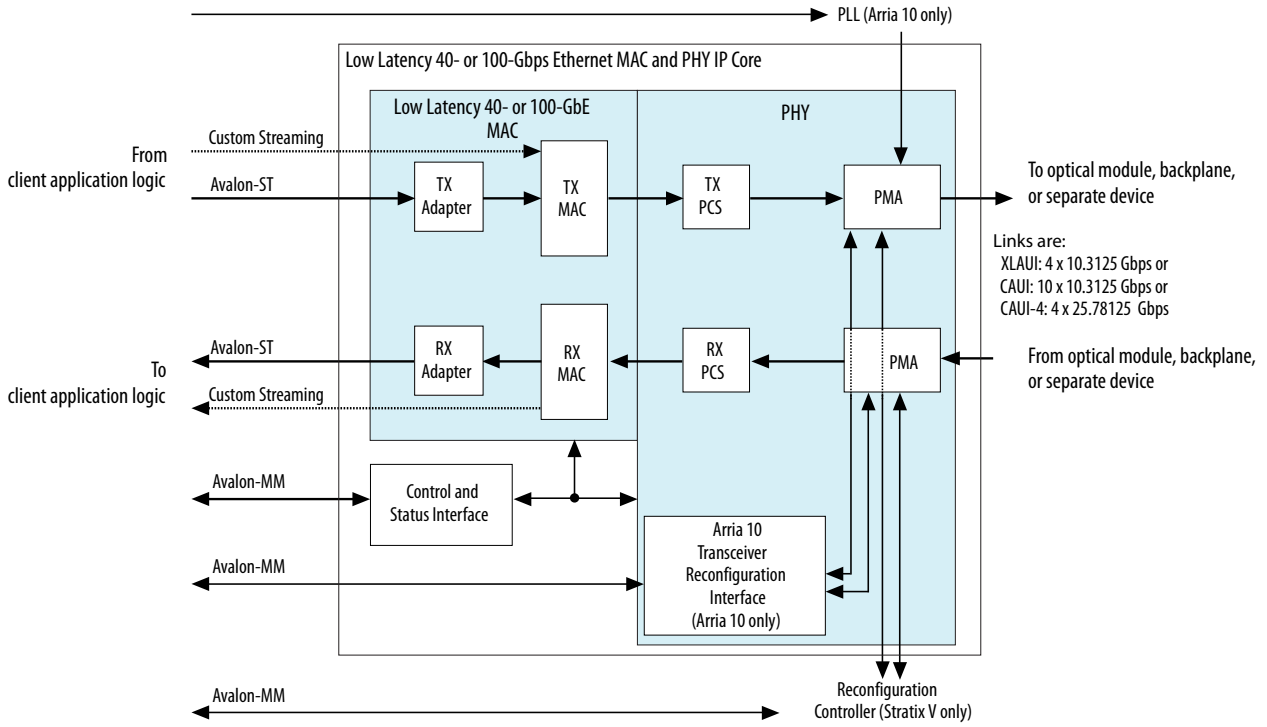
Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

# High Level System Overview

**Figure 3-1: Low Latency 40GbE and 100GbE MAC and PHY IP Cores**

Main blocks, internal connections, and external block requirements.




# Low Latency 40-100GbE MAC and PHY Functional Description

The Altera Low Latency 40-100GbE IP core implements the 40-100GbE Ethernet MAC in accordance with the *IEEE 802.3ba 2010 40G and 100G Ethernet Standard*. This IP core handles the frame encapsulation and flow of data between a client logic and Ethernet network via a 40-100GbE Ethernet PCS and PMA (PHY).

In the transmit direction, the MAC accepts client frames, and inserts inter-packet gap (IPG), preamble, start of frame delimiter (SFD), padding, and CRC bits before passing them to the PHY. The PHY encodes the MAC frame as required for reliable transmission over the media to the remote end.

In the receive direction, the PHY passes frames to the MAC. The MAC accepts frames from the PHY, performs checks, updates statistics counters, strips out the CRC, preamble, and SFD, and passes the rest of the frame to the client. In RX preamble pass-through mode, the MAC passes on the preamble and SFD to the client instead of stripping them out. In RX CRC pass-through mode (bit 1 of the `CRC_CONFIG` register has the value of 1), the MAC passes on the CRC bytes to the client and asserts the EOP signal in the same clock cycle with the final CRC byte.

The Low Latency 40-100GbE IP core includes the following interfaces:

- Datapath client-interface–The following options are available:
  - 40GbE with adapters—Avalon-ST, 256 bits
  - 40GbE—Custom streaming, 128 bits
  - 100GbE with adapters—Avalon-ST, 512 bits
  - 100GbE—Custom streaming, 256 bits
- Management interface—Avalon-MM host slave interface for MAC management. This interface has a data width of 32 bits and an address width of 16 bits.
- Datapath Ethernet interface–The following options are available:
  - 40GbE—Four 10.3125 Gbps serial links
  - 100GbE—Ten 10.3125 Gbps serial links
  - 100GbE CAUI–4—Four 25.78125 Gbps serial links
- In Arria 10 variations, an Arria 10 dynamic reconfiguration interface—an Avalon-MM interface to read and write the Arria 10 Native PHY IP core registers. This interface supports dynamic reconfiguration of the transceiver. Low Latency 40-100GbE IP cores that target an Arria 10 device use the Arria 10 Native PHY IP core to configure the Ethernet link serial transceivers on the device. This interface has a data width of 32 bits. This interface has an address width of 12 bits for 40GbE and 100GbE CAUI-4 variations, and an address width of 14 bits for standard 100GbE variations.

## Low Latency 40-100GbE IP Core TX Datapath

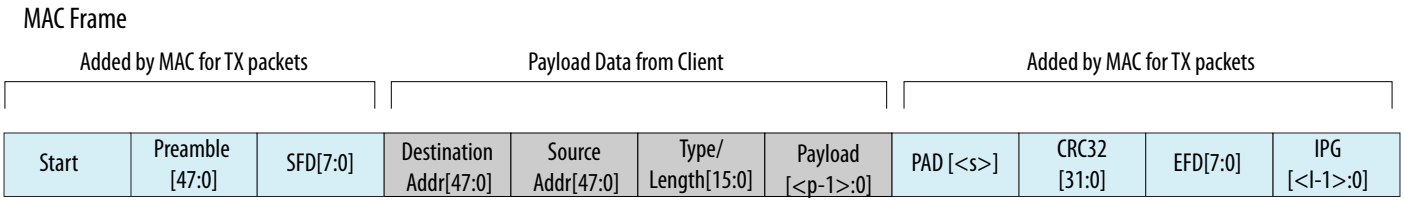
The TX MAC module receives the client payload data with the destination and source addresses and then adds, appends, or updates various header fields in accordance with the configuration specified. The MAC does not modify the destination address, the source address, or the payload received from the client. However, the TX MAC module adds a preamble (if the IP core is not configured to receive the preamble from user logic), pads the payload of frames greater than eight bytes to satisfy the minimum Ethernet frame payload of 46 bytes, and if you set **Enable TX CRC insertion** or turn on flow control, calculates the CRC over the entire MAC frame. (If padding is added, it is also included in the CRC calculation. If you turn off **Enable TX CRC insertion**, the client must provide the CRC bytes and must provide frames that have a minimum size of 64 bytes and therefore do not require padding). If you set **Average interpacket gap** to 8 or 12, the TX MAC module inserts IDLE bytes to maintain an average IPG. In addition, the TX MAC inserts an error in the Ethernet frame if the client requests to insert an error.

The Low Latency 40-100GbE IP core does not process incoming frames of less than nine bytes correctly. You must ensure such frames do not reach the TX client interface.

**Figure 3-2: Typical Client Frame at the Transmit Interface**

Illustrates the changes that the TX MAC makes to the client frame. This figure uses the following notational conventions:

- <*p*> = payload size, which is arbitrarily large.
- <*s*> = padding bytes = 0–46 bytes.
- <*l*> = number of IPG bytes




The following sections describe the functions that the TX module performs:

**Preamble, Start, and SFD Insertion** on page 3-4

**Address Insertion** on page 3-4

**Length/Type Field Processing** on page 3-4

**Frame Padding** on page 3-5

**Frame Check Sequence (CRC-32) Insertion** on page 3-5

**Inter-Packet Gap Generation and Insertion** on page 3-5

**Error Insertion Test and Debug Feature** on page 3-5

## Preamble, Start, and SFD Insertion

In the TX datapath the MAC appends an eight-byte preamble that begins with a Start byte (0xFB) to the client frame. If you turn on **Enable link fault generation**, this MAC module also incorporates the functions of the reconciliation sublayer.

The source of the preamble depends on whether you turn on the preamble pass-through feature by turning on **Enable preamble passthrough** in the Low Latency 40-100GbE parameter editor.

If the preamble pass-through feature is turned on, the client provides the eight-byte preamble (including Start byte) on the data bus. The client is responsible for providing the correct Start byte.

## Address Insertion

The client provides the destination MAC address and the source address of the local MAC.

## Length/Type Field Processing

This two-byte header represents either the length of the payload or the type of MAC frame. When the value of this field is equal to or greater than 1536 (0x600) it indicates a type field. Otherwise, this field

provides the length of the payload data that ranges from 0–1500 bytes. The TX MAC does not modify this field before forwarding it to the network.

## Frame Padding

When the length of client frame is less than 64 bytes (meaning the payload is less than 46 bytes) and greater than eight bytes, the TX MAC module inserts pad bytes (0x00) after the payload to create a frame length equal to the minimum size of 64 bytes.

**Caution:** The Low Latency 40-100GbE IP core does not process incoming (egress) frames of less than nine bytes correctly. You must ensure such frames do not reach the TX client interface.

## Frame Check Sequence (CRC-32) Insertion

The TX MAC computes and inserts a CRC32 checksum in the transmitted MAC frame. The frame check sequence (FCS) field contains a 32-bit CRC value. The MAC computes the CRC32 over the frame bytes that include the source address, destination address, length, data, and pad (if applicable). The CRC checksum computation excludes the preamble, SFD, and FCS. The encoding is defined by the following generating polynomial:

```
FCS(X) = X32 +X26 +X23 +X22 +X16 +X12 +X11 +X10 +X8 +X7 +X5 +X4 +X2 +X1 +1
```

CRC bits are transmitted with MSB (X32) first.

If you configure your Low Latency 40-100GbE IP core with no flow control, you can configure your IP core TX MAC to implement TX CRC insertion or not, by turning **Enable TX CRC insertion** on or off in the Low Latency 40-100GbE parameter editor. By default, the CRC insertion feature is enabled. In variations with flow control, CRC insertion is enabled.

**Related Information**

**Order of Transmission** on page 3-12
Illustrations of the byte order and octet transmission order on the Avalon-ST client interface.

## Inter-Packet Gap Generation and Insertion

If you set **Average interpacket gap** to **12** in the Low Latency 40-100GbE parameter editor, the TX MAC maintains the minimum inter-packet gap (IPG) between transmitted frames required by the IEEE 802.3 Ethernet standard. The standard requires an average minimum IPG of 96 bit times (or 12 byte times). The deficit idle counter maintains the average IPG of 12 bytes.

If you set **Average interpacket gap** to **8**, the TX MAC maintains a minimum average IPG of 8 bytes. This option is provided as an intermediate option to allow you to enforce an IPG that does not conform to the Ethernet standard, but which increases the throughput of your IP core.

If you set **Average interpacket gap** to **Disable deficit idle counter**, the IP core transmits Ethernet packets as soon as the data is available, without maintaining the 12-byte IPG. In this case the IP core maintains only a minimum 1-byte IPG. If you select this parameter value, you optimize the IP core throughput.

## Error Insertion Test and Debug Feature

The client can specify the insertion of a TX error in a specific packet. If the client specifies the insertion of a TX error, the LL 40-100GbE IP core inserts an error in the frame it transmits on the Ethernet link. The

error appears as a 66-bit error block that consists of eight /E/ characters (EBLOCK_T) in the Ethernet frame.

To direct the IP core to insert a TX error in a packet, the client should assert the TX error insertion signal as follows, depending on the TX client interface:

- On the Avalon-ST TX client interface, assert the l<*n*>_tx_error signal in the EOP cycle of the packet.
- On the custom streaming TX client interface, assert bit N of the tx_error[<w>-1:0] signal in the same cycle in which bit N of din_eop[<w>-1:0] is asserted.

The IP core overwrites Ethernet frame data with an EBLOCK_T error block when it transmits the Ethernet frame that corresponds to the packet EOP.

This feature supports test and debug of your IP core. In loopback mode, when the IP core receives a deliberately errored packet on the Ethernet link, the IP core recognizes it as a malformed packet.

**Related Information**

**LL 40-100GbE IP Core Malformed Packet Handling** on page 3-18

## Low Latency 40-100GbE IP Core TX Data Bus Interfaces

This section describes the TX data bus at the user interface and includes the following topics:

**Low Latency 40-100GbE IP Core User Interface Data Bus** on page 3-6

**Low Latency 40-100GbE IP Core TX Data Bus with Adapters (Avalon-ST Interface)** on page 3-7

**Low Latency 40-100GbE IP Core TX Data Bus Without Adapters (Custom Streaming Interface)** on page 3-10

**Bus Quantization Effects With Adapters** on page 3-12

**User Interface to Ethernet Transmission** on page 3-12

### Low Latency 40-100GbE IP Core User Interface Data Bus

**Table 3-1: User Interface Width Depends on IP Core Variation**

The Low Latency 40-100GbE IP core provides two different client interfaces: the Avalon-ST interface and a custom interface. The Avalon-ST interface requires adapters and the custom streaming interface does not require adapters.

| Client Interface | Data Bus Width (Bits) | |
|---|---|---|
| | LL 40GbE IP Core | LL 100GbE IP Core |
| Custom streaming interface (no adapters) | **128** | **256** |
| Avalon-ST interface (with adapters) | **256** | **512** |

## Low Latency 40-100GbE IP Core TX Data Bus with Adapters (Avalon-ST Interface)

The Low Latency 40-100GbE IP core TX datapath with adapters employs the Avalon-ST protocol. The Avalon-ST protocol is a synchronous point-to-point, unidirectional interface that connects the producer of a data stream (source) to a consumer of data (sink). The key properties of this interface include:

- Start of packet (SOP) and end of packet (EOP) signals delimit frame transfers.
- A valid signal qualifies signals from source to sink.
- The sink applies backpressure to the source by using the ready signal. The source typically responds to the deassertion of the ready signal from the sink by driving the same data until the sink can accept it. The `readyLatency` defines the relationship between assertion and deassertion of the ready signal, and cycles which are considered to be `ready` for data transfer.The `readyLatency` on the TX client interface is zero cycles.

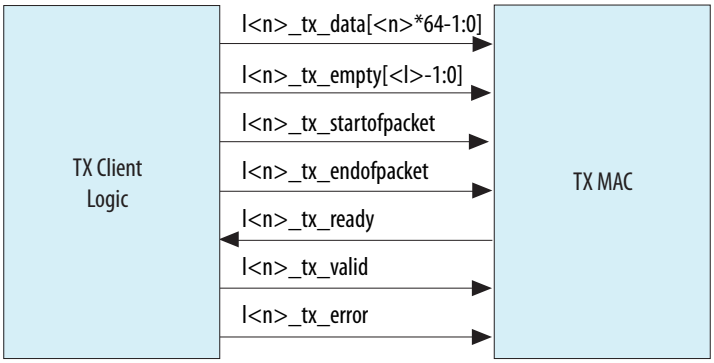
Altera provides an Avalon-ST interface with adapters for both the LL 40GbE and LL 100GbE IP cores. The Avalon-ST interface requires that the start of packet (SOP) always be in the MSB, simplifying the interpretation and processing of incoming data.

In the LL 40GbE IP core, the interface width is 256 bits, and in the LL 100GbE IP core, the interface width is 512 bits. The LL 40GbE Avalon-ST interface operates at 312.5 MHz and the LL 100GbE Avalon-ST interface operates at 390.625 MHz.

The client acts as a source and the TX MAC acts as a sink in the transmit direction.

### Figure 3-3: TX Client to MAC Interface with Adapters (Avalon-ST)

The Avalon-ST interface bus width varies with the IP core variation. In the figure, <*n*> = 4 for the 40GbE IP core and <*n*> = 8 for the 100GbE IP core. <*l*> is $\log_2(8*<n>)$.




### Table 3-2: Signals of the TX Client Interface

In the table, <*n*> = 4 for the 40GbE IP core and <*n*> = 8 for the 100GbE IP core. <*l*> is $\log_2(8*<n>)$. All interface signals are clocked by the `clk_txmac` clock.

| Signal Name | Direction | Description |
|---|---|---|
| l<*n*>_tx_data[<*n*>*64-1:0] | Input | TX data. If the preamble pass-through feature is enabled, data begins with the preamble.<br>The Low Latency 40-100GbE IP core does not process incoming frames of less than nine bytes correctly. You must ensure such frames do not reach the TX client interface.<br>You must send each TX data packet without intermediate idle cycles. Therefore, you must ensure your application can provide the data for a single packet in consecutive clock cycles. If data might not be available otherwise, you must buffer the data in your design and wait to assert l<*n*>_tx_startofpacket when you are assured the packet data to send on l<*n*>_tx_data[<*n*>*64-1:0] is available or will be available on time. |
| l<*n*>_tx_empty[<*l*>-1:0] | Input | Indicates the number of empty bytes on l<*n*>_tx_data when l<*n*>_tx_endofpacket is asserted. |
| l<*n*>_tx_startofpacket | Input | When asserted, indicates the start of a packet. The packet starts on the MSB. |
| l<*n*>_tx_endofpacket | Input | When asserted, indicates the end of packet. |
| l<*n*>_tx_ready | Output | When asserted, the MAC is ready to receive data. The l<*n*>_tx_ready signal acts as an acknowledge. The source drives l<*n*>_tx_valid and l<*n*>_tx_data[<*n*>*64-1:0], then waits for the sink to assert l<*n*>_tx_ready. The readyLatency is zero cycles, so that the IP core accepts valid data in the same cycle in which it asserts l<*n*>_tx_ready.<br>The l<*n*>_tx_ready signal indicates the MAC is ready to receive data in normal operational model. However, the l<*n*>_tx_ready signal might not be an adequate indication following reset. To avoid sending packets before the Ethernet link is able to transmit them reliably, you should ensure that the application does not send packets on the TX client interface until after the tx_lanes_stable signal is asserted. |
| l<*n*>_tx_valid | Input | When asserted l<*n*>_tx_data is valid. This signal must be continuously asserted between the assertions of l<*n*>_tx_startofpacket and l<*n*>_tx_endofpacket for the same packet. |

| Signal Name | Direction | Description |
|---|---|---|
| l<*n*>_tx_error | Input | When asserted in an EOP cycle (while l<*n*>_tx_endofpacket is asserted), directs the IP core to insert an error in the packet before sending it on the Ethernet link.<br>This signal is a test and debug feature. In loopback mode, the IP core recognizes the packet upon return as a malformed packet. |

**Figure 3-4: Traffic on the TX and RX Avalon-ST Client Interface for Low Latency 40GbE IP Core**

Shows typical traffic for the TX and RX Avalon-ST interface Low Latency 40GbE IP core.

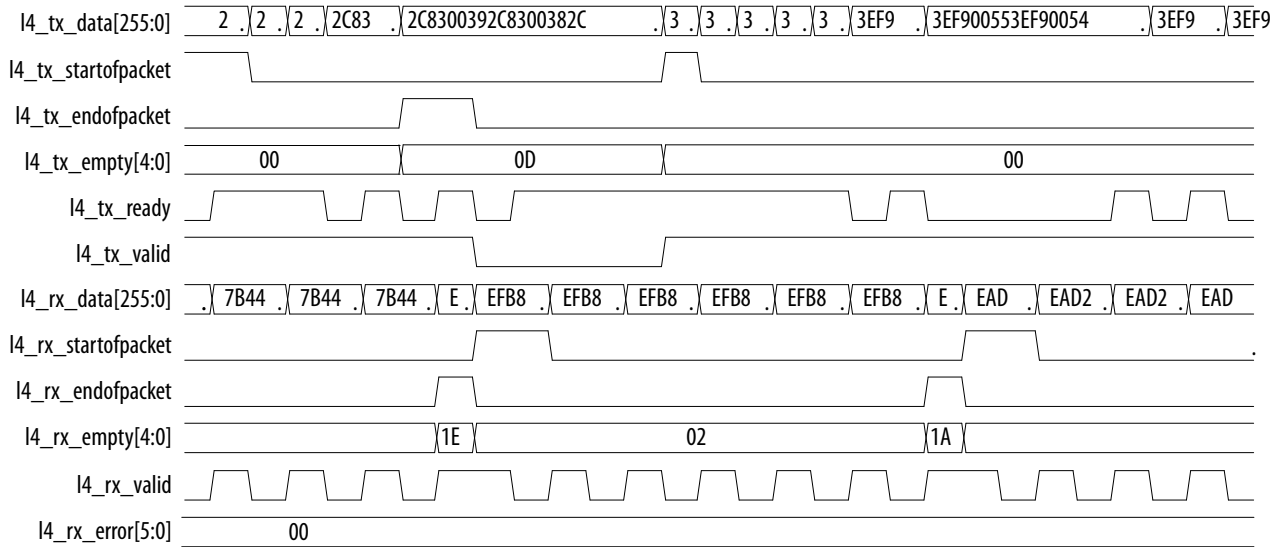
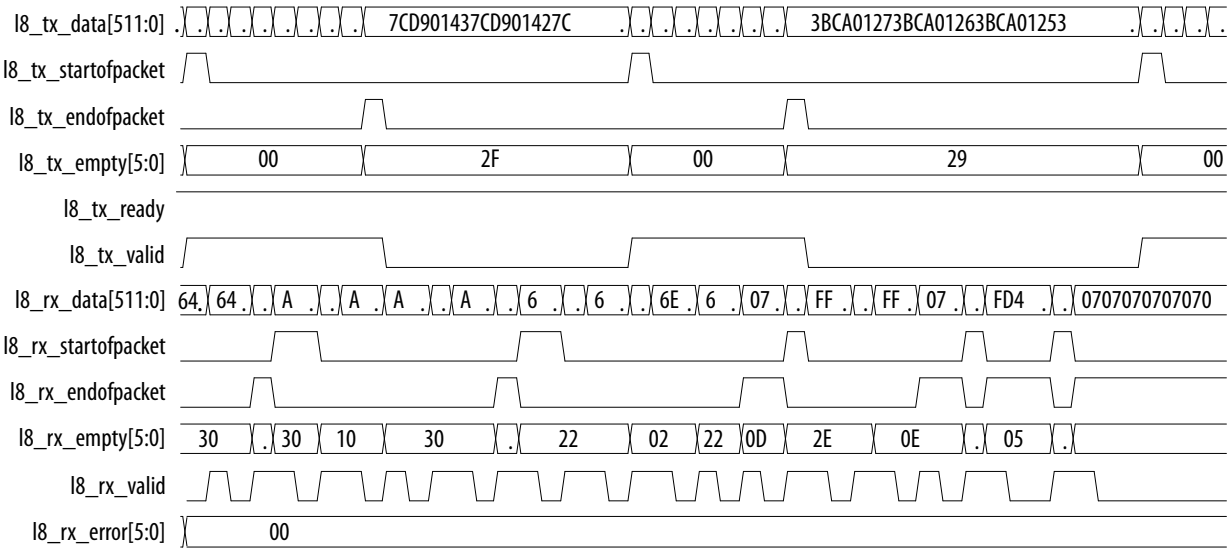


**Figure 3-5: Traffic on the TX and RX Avalon-ST Client Interface for Low Latency 100GbE IP Core**

Shows typical traffic for the TX and RX Avalon-ST interface of the 100GbE IP core.




**Related Information**

**Avalon Interface Specifications**

For more information about the Avalon-ST interface.

## Low Latency 40-100GbE IP Core TX Data Bus Without Adapters (Custom Streaming Interface)

When no adapters are used, the LL 40GbE custom interface bus width is 2 words (128 bits) and the LL 100GbE custom interface bus width is 4 words (256 bits). The LL 40GbE custom interface operates at 312.5 MHz and the LL 100GbE custom interface operates at 390.625 MHz.

**Figure 3-6: TX Client to MAC Interface Without Adapters**

The custom streaming interface bus width varies with the IP core variation. In the figure, <*w*> = 2 for the 40GbE IP core and <*w*> = 4 for the 100GbE IP core.

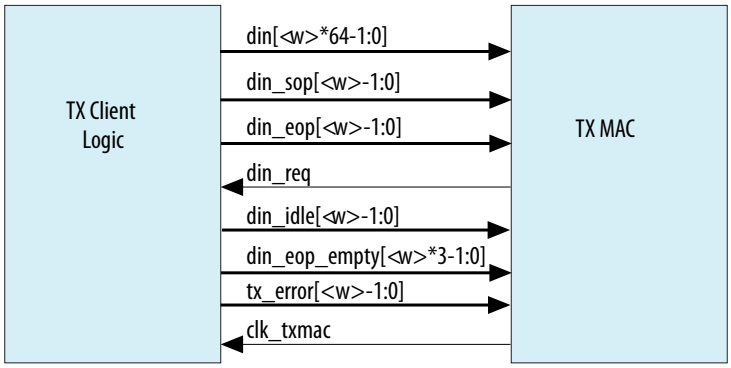



**Table 3-3: Signals of the TX Client Interface Without Adapters**

In the table, <*w*> = 2 for the 40GbE IP core and <*w*> = 4 for the 100GbE IP core. The signals are clocked by `clk_txmac`.

| Signal Name | Direction | Description |
|---|---|---|
| din[<w>*64-1:0] | Input | Data bytes to send in big-endian mode.<br>Most significant 64-bit word is in the higher-order bits. In 40GbE variations, the most significant word is in bits [127:64] and in 100GbE variations, the most significant word is in bits [255:192].<br>The Low Latency 40-100GbE IP core does not process incoming frames of less than nine bytes correctly. You must ensure such frames do not reach the TX client interface. |
| din_sop[<w>-1:0] | Input | Start of packet (SOP) location in the TX data bus. Only the most significant byte of each 64-bit word may be a start of packet. Bit 63 or 127 are possible for the 40GbE and bits 255, 191, 127, or 63 are possible for 100 GbE.<br>Bit 0 of din_sop corresponds to the data word in din[63:0]. |
| din_eop[<w>-1:0] | Input | End of packet location in the TX data bus. Indicates the 64-bit word that holds the end-of-packet byte. Any byte may be the last byte in a packet.<br>Bit 0 of din_eop corresponds to the data word in din[63:0]. |
| din_eop_empty[<w>*3-1:0] | Input | Indicates the number of empty (invalid) bytes in the end-of-packet byte in the word indicated by din_eop.<br>If din_eop[z] has the value of 0, then the value of din_eop_empty[(z+2):z] does not matter. However, if din_eop[z] has the value of 1, then you must set the value of din_eop_empty[(z+2):z] to the number of empty (invalid) bytes in the end-of-packet word z.<br>For example, if you have a 100GbE IP core and want to indicate that in the current clk_txmac clock cycle, byte 6 in word 2 of din is an end-of-packet byte, and no other words hold an end-of-packet byte in the current clock cycle, you must set the value of din_eop to 4'b0100 and the value of din_eop_empty to 12'b000_110_000_000. |
| din_idle[<w>-1:0] | Input | Indicates the words in din that hold Idle bytes or control information rather than Ethernet data. One-hot encoded. |
| din_req | Output | Indicates that input data was accepted by the IP core. |
| tx_error[<w>-1:0] | Input | When asserted in an EOP cycle (while din_eop[<w>-1:0] is non-zero) , directs the IP core to insert an error in the corresponding packet before sending it on the Ethernet link.<br>This signal is a test and debug feature. In loopback mode, the IP core recognizes the packet upon return as a malformed packet. |
| clk_txmac | Output | TX MAC clock. The clock frequency should be 312.5 MHz in LL 40GbE IP cores, and 390.625 MHz in LL 100GbE IP cores. The clk_txmac clock and the clk_rxmac clock (which clocks the RX datapath) are assumed to have the same frequency. |

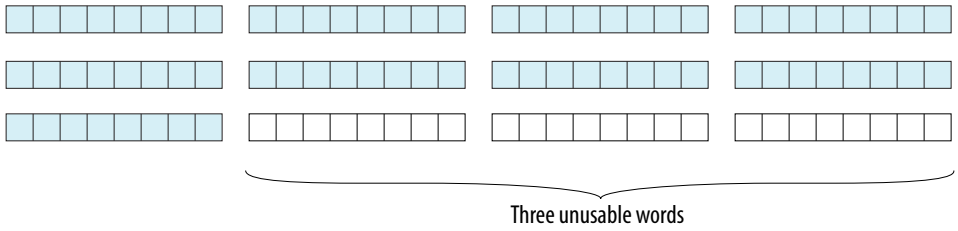
The IP core reads the bytes in big endian order. A packet may start in the most significant byte of any word. A packet may end on any byte.

## Bus Quantization Effects With Adapters

The TX custom streaming interface allows a packet to start at any of two or four positions to maximize utilization of the link bandwidth. The TX Avalon-ST interface only allows start of packet (SOP) to be placed at the most significant position. If the SOP were restricted to the most significant position in the client logic data bus in the custom streaming interface, bus bandwidth would be reduced.

**Figure 3-7: Reduced Bandwidth With Left-Aligned SOP Requirement**

Illustrates the reduction of bandwidth that would be caused by left-aligning the SOP for the 100GbE IP core.




The example shows a nine-word packet, which is the worst case for bandwidth utilization. Assuming another packet is waiting for transmission, the effective ingress bandwidth is reduced by 25%. Running the MAC portion of the logic slightly faster than is required can mitigate this loss of bandwidth. Additional increases in the MAC frequency can provide further mitigation, although increases in frequency make timing closure more difficult. The wider data bus for the Avalon-ST interface also helps to compensate for the Avalon-ST left-aligned SOP requirement.

## User Interface to Ethernet Transmission

The IP core reverses the bit stream for transmission per Ethernet requirements. The transmitter handles the insertion of the inter-packet gap, frame delimiters, and padding with zeros as necessary. The transmitter also handles FCS computation and insertion.

The Ethernet MAC and PHY transmit complete packets. After transmission begins, it must complete with no IDLE insertions. Between the end of one packet and the beginning of the next packet, the data input is not considered and the transmitter sends IDLE characters. An unbounded number of IDLE characters can be sent between packets.

### Order of Transmission

The IP core transmits bytes on the Ethernet link starting with the preamble and ending with the FCS in accordance with the IEEE 802.3 standard. Transmit frames the IP core receives on the client interface are big-endian. Frames the MAC sends to the PHY on the XGMII/CGMII between the MAC and the PHY are little-endian; the MAC TX transmits frames on this interface beginning with the least significant byte.

**Figure 3-8: Byte Order on the Client Interface Lanes Without Preamble Pass-Through**

Describes the byte order on the Avalon-ST interface when the preamble pass-through feature is turned off. Destination Address[40] is the broadcast/multicast bit (a type bit), and Destination Address[41] is a locally administered address bit.

| | Destination Address (DA) | | | | | | Source Address (SA) | | | | | | Type/ Length (TL) | | Data (D) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Octet | 5 | 4 | 3 | 2 | 1 | 0 | 5 | 4 | 3 | 2 | 1 | 0 | 1 | 0 | 00 | ... | NN |
| Bit | [47:40] | [39:32] | [31:24] | [23:16] | [15:8] | [7:0] | [47:40] | [39:32] | [31:24] | [23:16] | [15:8] | [7:0] | [15:8] | [7:0] | MSB[7 :0] | ... | LSB[7:0] |

For example, the destination MAC address includes the following six octets AC-DE-48-00-00-80. The first octet transmitted (octet 0 of the MAC address described in the 802.3 standard) is AC and the last octet transmitted (octet 7 of the MAC address) is 80. The first bit transmitted is the low-order bit of AC, a zero. The last bit transmitted is the high order bit of 80, a one.

The preceding table and the following figure show that in this example, 0xAC is driven on `DA5 (DA[47:40])` and 0x80 is driven on `DA0 (DA[7:0])`.

**Figure 3-9: Octet Transmission on the Avalon-ST Signals Without Preamble Pass-Through**

Illustrates how the octets of the client frame are transferred over the TX datapath when preamble pass-through is turned off.

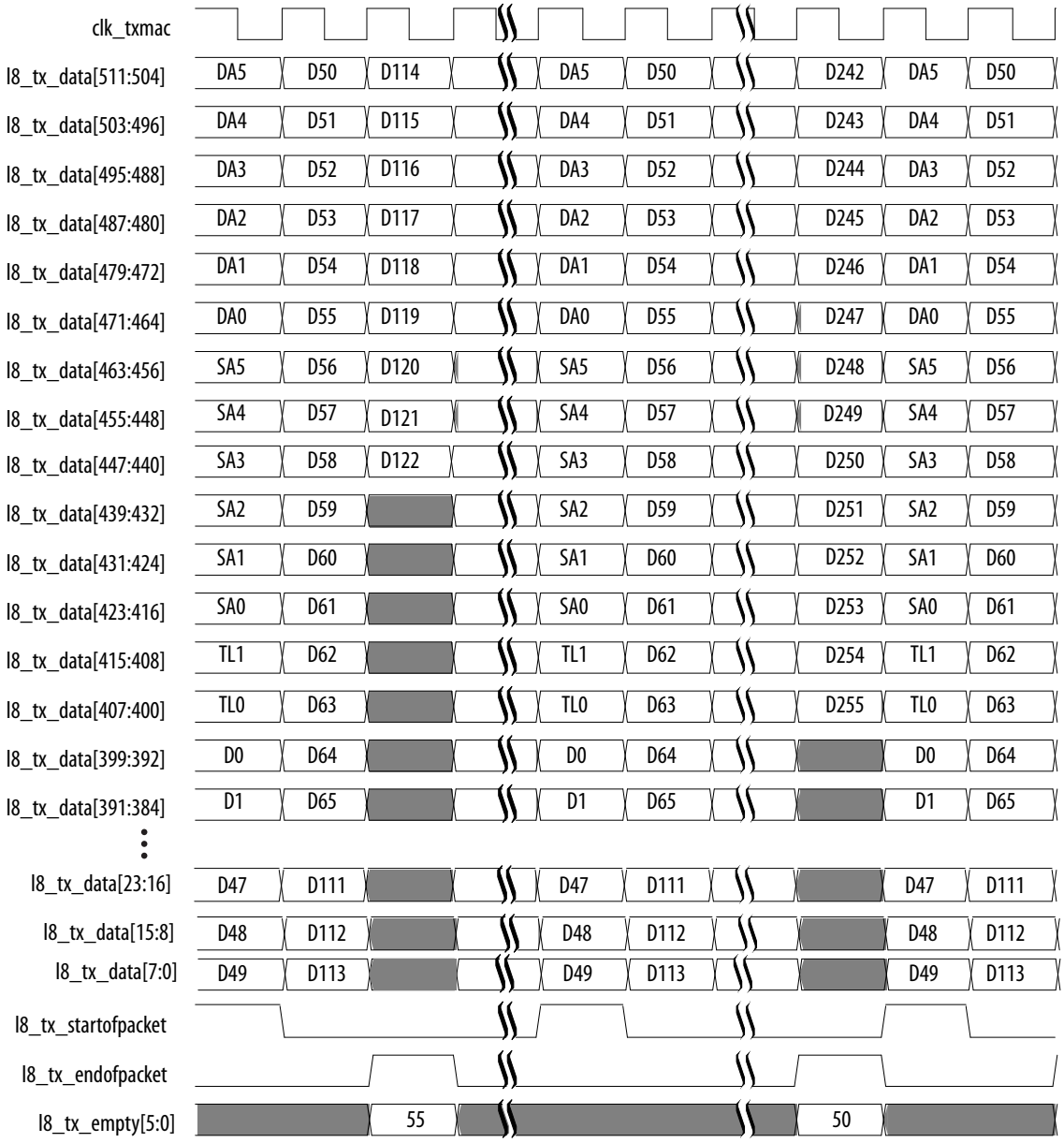


**Figure 3-10: Byte Order on the Avalon-ST Interface Lanes With Preamble Pass-Through**

Describes the byte order on the Avalon-ST interface when the preamble pass-through feature is turned on.

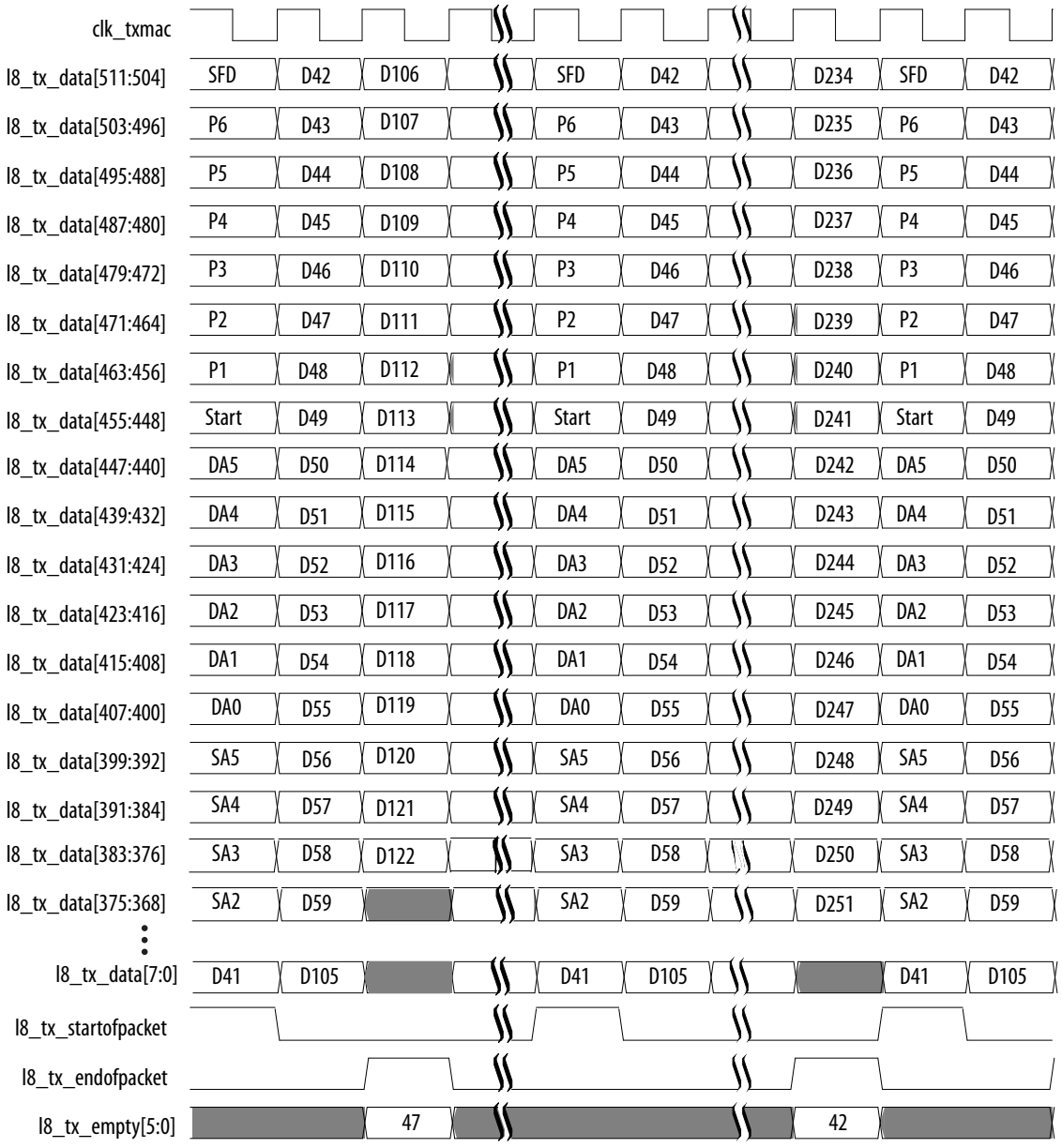
Destination Address[40] is the broadcast/multicast bit (a type bit), and Destination Address[41] is a locally administered address bit.

| | SFD | Preamble | | | | | | Start | Destination Address (DA) | | | | | | Source Address (SA) | | | | | | Type/ Length | | Data (D) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Octet | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 5 | 4 | 3 | 2 | 1 | 0 | 5 | 4 | 3 | 2 | 1 | 0 | 1 | 0 | 00 | ... | NN |
| Bit | [63:56] | [55:48] | [47:40] | [39:32] | [31:24] | [23:16] | [15:8] | [7:0] | [47:40] | [39:32] | [31:24] | [23:16] | [15:8] | [7:0] | [47:40] | [39:32] | [31:24] | [23:16] | [15:8] | [7:0] | [15:8] | [7:0] | MSB[7:0] | ... | LSB[7:0] |

**Figure 3-11: Octet Transmission on the Avalon-ST Signals With Preamble Pass-Through**

Illustrates how the octets of the client frame are transferred over the TX datapath when preamble pass-through is turned on. The eight preamble bytes precede the destination address bytes. The preamble bytes are reversed: the application must drive the Start byte on `l8_tx_data[455:448]` and the SFD byte on `l8_tx_data[511:504]`.

The destination address and source address bytes follow the preamble pass-through in the same order as in the case without preamble pass-through.



# Low Latency 40-100GbE IP Core RX Datapath

The Low Latency 40-100GbE RX MAC receives Ethernet frames from the PHY and forwards the payload with relevant header bytes to the client after performing some MAC functions on header bytes.

**Figure 3-12: Flow of Frame Through the MAC RX Without Preamble Pass-Through**

Illustrates the typical flow of frame through the MAC RX when the preamble pass-through feature is turned off. In this figure, <*p*> is payload size (0–1500 bytes), and <*s*> is the number of pad bytes (0–46 bytes).

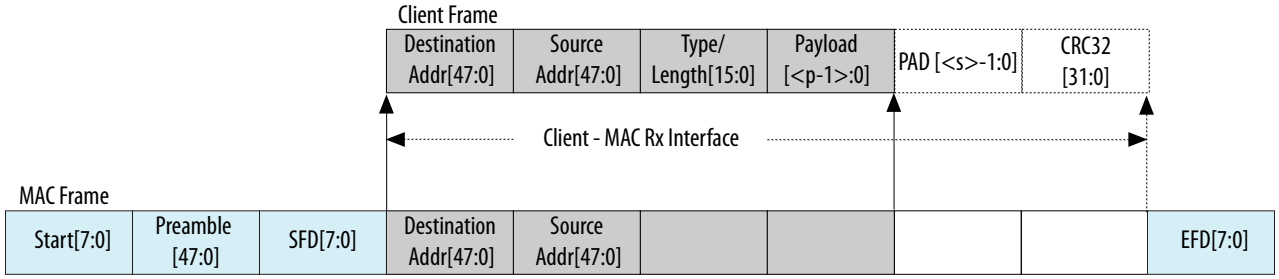
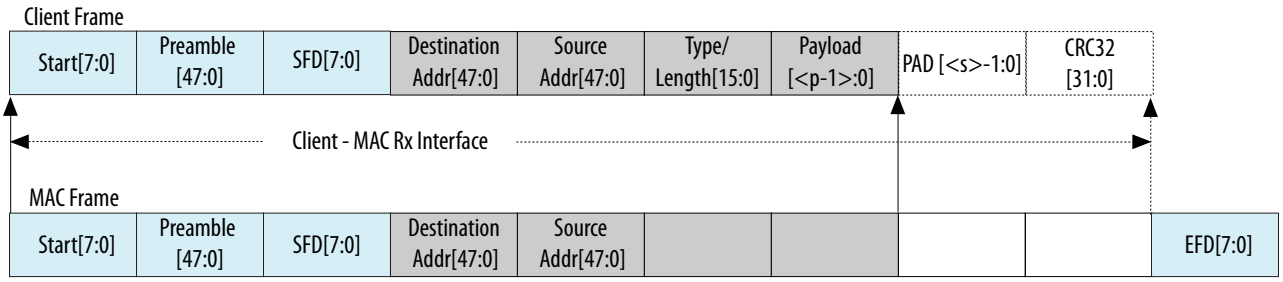



**Figure 3-13: Flow of Frame Through the MAC RX With Preamble Pass-Through Turned On**

Illustrates the typical flow of frame through the MAC RX when the preamble pass-through feature is turned on. In this figure, <*p*> is payload size (0–1500 bytes), and <*s*> is the number of pad bytes (0–46 bytes).




The following sections describe the functions performed by the RX MAC:

**Low Latency 40-100GbE IP Core RX Filtering** on page 3-18

**40-100GbE IP Core Preamble Processing** on page 3-18

**40-100GbE IP Core FCS (CRC-32) Removal** on page 3-18

**40-100GbE IP Core CRC Checking** on page 3-18

**LL 40-100GbE IP Core Malformed Packet Handling** on page 3-18

**RX CRC Forwarding** on page 3-19

**Inter-Packet Gap** on page 3-19

**Pause Ignore** on page 3-19

**Control Frame Identification** on page 3-20

## Low Latency 40-100GbE IP Core RX Filtering

The Low Latency 40-100GbE IP core processes all incoming valid frames. However, the IP core does not forward pause frames to the Avalon-ST RX client interface by default.

If you set the `cfg_fwd_ctrl` bit of the `RX_PAUSE_FWD` register to the value of 1, the IP core forwards pause frames to the Avalon-ST RX client interface.

## 40-100GbE IP Core Preamble Processing

The preamble sequence is Start, six preamble bytes, and SFD. If this sequence is incorrect the frame is ignored. The Start byte must be on receive lane 0 (most significant byte). The IP core uses the SFD byte (0xD5) to identify the last byte of the preamble. The MAC RX looks for the Start, six preamble bytes and SFD.

By default, the MAC RX removes all Start, SFD, preamble, and IPG bytes from accepted frames. However, if you turn on **Enable preamble passthrough** in the Low Latency 40-100GbE parameter editor, the MAC RX does not remove the eight-byte preamble sequence.

## 40-100GbE IP Core FCS (CRC-32) Removal

Independent user configuration register bits control FCS CRC removal at runtime. CRC removal supports both narrow and wide bus options. Bit 0 of the `MAC_CRC_CONFIG` register enables and disables CRC removal; by default, CRC removal is enabled.

In the user interface, the EOP signal (`l<n>_rx_endofpacket` or `dout_eop` ) indicates the end of CRC bytes if CRC is not removed. When CRC is removed, the EOP signal indicates the final byte of payload.

The IP core signals an FCS error by asserting the FCS error output signal `l<n>_rx_fcs_error` and the `l<n>_rx_fcs_valid` (or `rx_fcs_error` and the `rx_fcs_valid`) output signals in the same clock cycle. The `l<n>_rx_error[1]` or `rx_error[1]` also signals an FCS error.

If you turn on **Enable alignment EOP on FCS word** in the parameter editor, the IP core asserts `l<n>_rx_fcs_error` (or `rx_fcs_error`) and the EOP signal on the same clock cycle if the current frame has an FCS error. However, if you turn off **Enable alignment EOP on FCS word**, the IP core asserts `l<n>_rx_fcs_error` in a later clock cycle than the EOP signal.

## 40-100GbE IP Core CRC Checking

The 32-bit CRC field is received in the order: X32, X30, . . . X1, and X0 , where X32 is the most significant bit of the FCS field and occupies the least significant bit position in the first FCS byte.

If a CRC32 error is detected, the RX MAC marks the frame invalid by asserting the `l<n>_rx_fcs_error` and `l<n>_rx_fcs_valid` (or `rx_fcs_error` and `rx_fcs_valid`) signals, as well as the `l<n>_rx_error[1]` (or `rx_error[1]`) signal.

## LL 40-100GbE IP Core Malformed Packet Handling

While receiving an incoming packet from the Ethernet link, the LL 40-100GbE IP core expects to detect a terminate character at the end of the packet. When it detects an expected terminate character, the IP core generates an EOP on the client interface. However, sometimes the IP core detects an unexpected control

character when it expects a terminate character. The Low Latency 40-100GbE IP core detects and handles the following forms of malformed packets:

- If the IP core detects an Error character, it generates an EOP, asserts a malformed packet error (`rx_error[0]` or `l<n>_rx_error[0]`), and asserts an FCS error (`rx_fcs_error` (`l<n>_rx_fcs_error`) and `rx_error[1]` (`l<n>_rx_error[1]`). If the IP core subsequently detects a terminate character, it does not generate another EOP indication.
- If the IP core detects any other control character when it is waiting for an EOP indication (terminate character), the IP core generates an EOP indication (for example, an IDLE or Start character), asserts a malformed packet error (`rx_error[0]` or `l<n>_rx_error[0]`), and asserts an FCS error (`rx_fcs_error` (`l<n>_rx_fcs_error`) and `rx_error[1]` (`l<n>_rx_error[1]`). If the IP core subsequently detects a terminate character, it does not generate another EOP indication.

The IP core ignores a Start control character it receives on any lane other than Lane 0.

When the IP core receives a packet that contains an error deliberately introduced on the Ethernet link using the LL 40-100GbE TX error insertion feature, the IP core identifies it as a malformed packet.

**Related Information**

**Error Insertion Test and Debug Feature** on page 3-5

## RX CRC Forwarding

The CRC-32 field is forwarded to the client interface after the final byte of data, if the CRC removal option is not enabled.

**Related Information**

**40-100GbE IP Core FCS (CRC-32) Removal** on page 3-18

## Inter-Packet Gap

The MAC RX removes all IPG octets received, and does not forward them to the Avalon-ST client interface. If you configure your IP core with a custom streaming client interface, the MAC RX does not remove IPG octets. The MAC RX fowards all received IPG octets to the custom client interface.

## Pause Ignore

If you turn on flow control by setting **Flow control mode** to the value of **Standard flow control** or **Priority-based flow control** in the Low Latency 40-100GbE parameter editor, the IP core processes incoming pause frames by default. However, when the pause frame receive enable bit (for standard flow control) or bits (for priority-based flow control) is or are not set, the IP core does not process incoming pause frames. In this case, the MAC TX traffic is not affected by the valid pause frames.

If you turn off flow control by setting **Flow control mode** to the value of **No flow control** in the LL 40-100GbE parameter editor, the IP core does not process incoming pause frames.

**Related Information**

- **Congestion and Flow Control Using Pause Frames** on page 3-29
- **Pause Control and Generation Interface** on page 3-32
- **Pause Registers** on page 3-94

### Control Frame Identification

The mechanisms to process flow control frames from the Ethernet link are specific to the flow control mode of the LL 40-100GbE IP core. Therefore, control frames that are inappropriate to the IP core mode, simply pass through the RX MAC to the RX client interface.

If you turn off flow control by setting **Flow control mode** to the value of **No flow control** in the LL 40-100GbE parameter editor, the IP core might nevertheless receive a flow control request on the Ethernet link. Similarly, if you set **Flow control mode** to **Standard flow control**, the IP core might nevertheless receive a priority-based flow control request on the Ethernet link, and if you set **Flow control mode** to **Priority-based flow control**, the IP core might nevertheless receive a standard flow control request.

The IP core provides a three-bit RX status flag that identifies the control frames that appear on the RX client interface. The flag is one-hot encoded to identify whether the control frame is a standard flow control frame, a priority-based flow control frame, or a different type of control frame. If the flag has the value of 3'b000, the current packet on the RX client interface is not a control frame. You can use this flag to identify control frames on the RX client interface that the client prefers to ignore.

**Related Information**

- **Low Latency 40-100GbE IP Core RX Data Bus** on page 3-21
- **Low Latency 40-100GbE IP Core RX Data Bus Without Adapters (Custom Streaming Interface)** on page 3-25

## Low Latency 40-100GbE IP Core RX Data Bus Interface

This section describes the RX data bus at the user interface and includes the following topics:

**Low Latency 40-100GbE IP Core User Interface Data Bus** on page 3-6

**Low Latency 40-100GbE IP Core RX Data Bus** on page 3-21

**Low Latency 40-100GbE IP Core RX Data Bus Without Adapters (Custom Streaming Interface)** on page 3-25

### Low Latency 40-100GbE IP Core User Interface Data Bus

**Table 3-4: User Interface Width Depends on IP Core Variation**

The Low Latency 40-100GbE IP core provides two different client interfaces: the Avalon-ST interface and a custom interface. The Avalon-ST interface requires adapters and the custom streaming interface does not require adapters.

| Client Interface | Data Bus Width (Bits) | |
|---|---|---|
| | LL 40GbE IP Core | LL 100GbE IP Core |
| Custom streaming interface (no adapters) | **128** | **256** |
| Avalon-ST interface (with adapters) | **256** | **512** |

## Low Latency 40-100GbE IP Core RX Data Bus

The Low Latency 40-100GbE IP core RX datapath employs the Avalon-ST protocol. The Avalon-ST protocol is a synchronous point-to-point, unidirectional interface that connects the producer of a data stream (source) to a consumer of data (sink). The key properties of this interface include:

- Start of packet (SOP) and end of packet (EOP) signals delimit frame transfers.
- A valid signal qualifies signals from source to sink.

Altera provides an Avalon-ST interface for both the LL 40GbE and LL 100GbE IP cores. In the 40GbE IP core, the interface width is 256 bits, and in the 100GbE IP core, the interface width is 512 bits. In the LL 40GbE IP core, the client interface operates at a frequency of 312.5 MHz, and in the LL 100GbE IP core, the client interface operates at a frequency of 390.625 MHz. The Avalon-ST interface requires that the start of packet (SOP) always be in the MSB.

The RX MAC acts as a source and the client acts as a sink in the receive direction.

**Figure 3-14: RX MAC to Client Interface**

The Avalon-ST interface bus width varies with the IP core variation. In the figure, <*n*> = 4 for the 40GbE IP core and <*n*> = 8 for the 100GbE IP core. <*l*> is $\log_2(8*<n>)$.

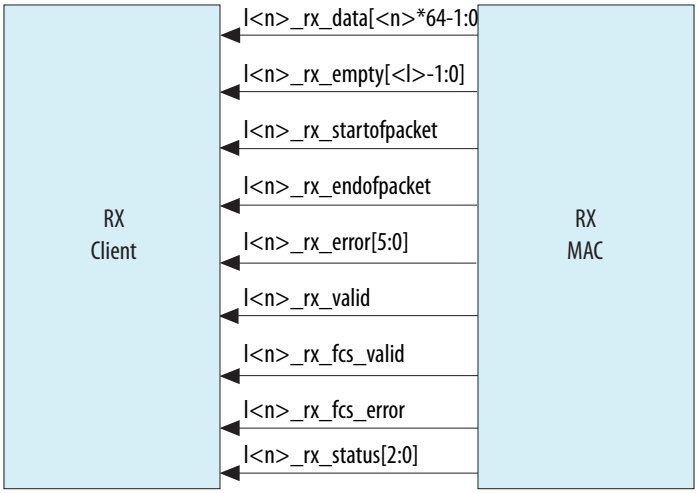



**Table 3-5: Signals of the RX Client Interface**

In the table, <*n*> = 4 for the 40GbE IP core and <*n*> = 8 for the 100GbE IP core. <*l*> is $\log_2(8*<n>)$. The signals are clocked by `clk_rxmac`.

| Name | Direction | Description |
|---|---|---|
| `l<n>_rx_data[<n>*64-1:0]` | Output | RX data. |
| `l<n>_rx_empty[<l>-1:0]` | Output | Indicates the number of empty bytes on `l<n>_rx_data` when `l<n>_rx_endofpacket` is asserted, starting from the least significant byte (LSB). |
| `l<n>_rx_startofpacket` | Output | When asserted, indicates the start of a packet. The packet starts on the MSB. |

| Name | Direction | Description |
|---|---|---|
| l<n>_rx_endofpacket | Output | When asserted, indicates the end of packet. |
| l<*n*>_rx_error[5:0] | Output | Reports certain types of errors in the Ethernet frame whose contents are currently being transmitted on the client interface. This signal is valid in EOP cycles only. To ensure you can identify the corresponding packet, you must turn on **Enable alignment EOP on FCS word** in the LL 40-100GbE parameter editor.<br>The individual bits report different types of errors:<br>• Bit [0]: Malformed packet error. If this bit has the value of 1, the packet is malformed. The IP core identifies a malformed packet when it receives a control character that is not a terminate character, while receiving the packet.<br>• Bit [1]: CRC error. If this bit has the value of 1, the IP core detected a CRC error in the frame.<br>If you turn on **Enable alignment EOP on FCS word**, this bit and the l<*n*>_rx_fcs_error signal behave identically.<br>• Bit [2]: undersized payload. If this bit has the value of 1, the frame size is between nine and 63 bytes, inclusive. The IP core does not recognize an incoming frame of size eight bytes or less as a frame, and those cases are not reported here.The l<n>_rx_error[1] or rx_error[1] also signals an FCS error.<br>• Bit [3]: oversized payload. If this bit has the value of 1, the frame size is greater than the maximum frame size programmed in the MAX_RX_SIZE_CONFIG register at offset 0x506.<br>• Bit [4]: payload length error. If this bit has the value of 1, the payload received in the frame did not match the length field value, and the value in the length field is less than 1536 bytes. This bit only reports errors if you set bit [0] of the CFG_PLEN_CHECK register at offset 0x50A to the value of 1.<br>• Bit [5} Reserved. |
| l<*n*>_rx_valid | Output | When asserted, indicates that RX data is valid. Only valid between the l<*n*>_rx_startofpacket and l<*n*>_rx_endofpacket signals. |
| l<*n*>_rx_fcs_valid | Output | When asserted, indicates that FCS is valid. |

| Name | Direction | Description |
|---|---|---|
| l<*n*>_rx_fcs_error | Output | When asserted, indicates an FCS error condition. The IP core asserts the l<*n*>_rx_fcs_error signal only when it asserts the l<*n*>_rx_fcs valid signal.<br><br>Runt frames always force an FCS error condition. However, if a packet is eight bytes or smaller, it is considered a decoding error and not a runt frame, and the IP core does not flag it as a runt. |
| l<*n*>_rx_status[2:0] | Output | Indicates the IP core received a control frame on the Ethernet link. This signal identifies the type of control frame the IP core is passing through to the client interface.<br><br>This signal is valid in EOP cycles only. To ensure you can identify the corresponding packet, you must turn on **Enable alignment EOP on FCS word** in the LL 40-100GbE parameter editor.<br><br>The individual bits report different types of received control frames:<br>• Bit [0]: Indicates the IP core received a standard flow control frame. If the IP core is in standard flow control mode and the cfg_fwd_ctrl bit of the RX_PAUSE_FWD register has the value of 0, this bit maintains the value of 0.<br>• Bit [1]: Indicates the IP core received a priority flow control frame. If the IP core is in priority flow control mode and the cfg_fwd_ctrl bit of the RX_PAUSE_FWD register has the value of 0, this bit maintains the value of 0.<br>• Bit [2]: Indicates the IP core received a control frame that is not a flow control frame. |

**Figure 3-15: Traffic on the TX and RX Avalon-ST Client Interface for Low Latency 40GbE IP Core**

Shows typical traffic for the TX and RX Avalon-ST interface Low Latency 40GbE IP core.

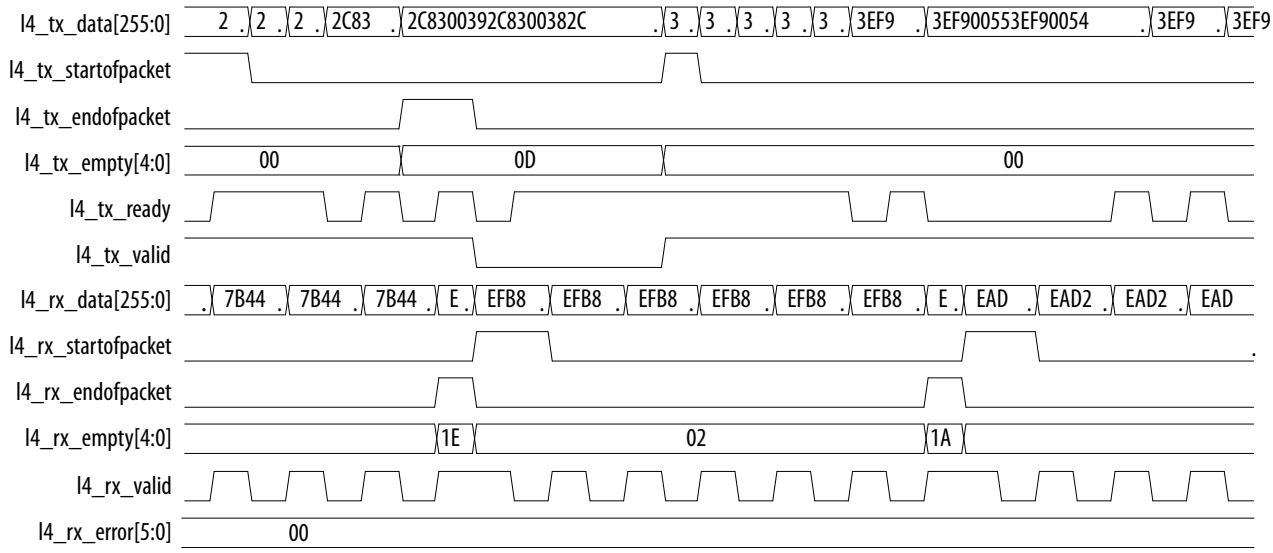
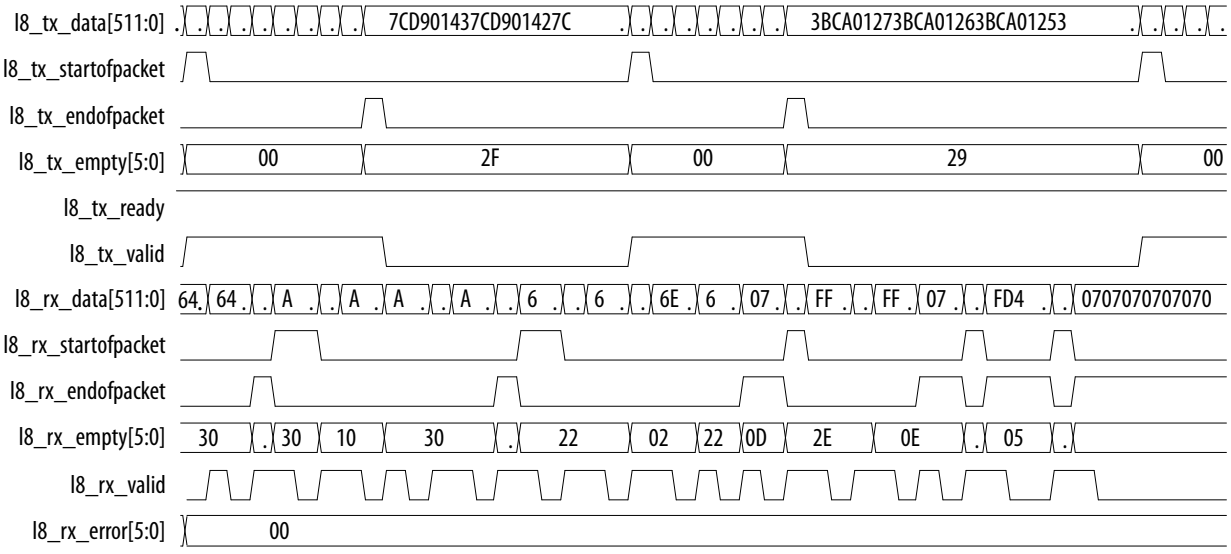



**Figure 3-16: Traffic on the TX and RX Avalon-ST Client Interface for Low Latency 100GbE IP Core**

Shows typical traffic for the TX and RX Avalon-ST interface of the 100GbE IP core.




**Related Information**

- **Low Latency 40-100GbE IP Core MAC Configuration Registers** on page 3-92
  Describes the `MAX_RX_SIZE_CONFIG` and `CFG_PLEN_CHECK` registers.
- **Control Frame Identification** on page 3-20
- **Avalon Interface Specifications**
  For more information about the Avalon-ST interface.

## Low Latency 40-100GbE IP Core RX Data Bus Without Adapters (Custom Streaming Interface)

The RX bus without adapters consists of four 8-byte words, or 256 bits, operating at a frequency of 390.625. for the 100GbE IP core or two 8-byte words, or 128 bits, for the 40GbE IP core, nominally at 312.5 MHz. This bus drives data from the RX MAC to the RX client.

**Figure 3-17: RX MAC to Client Interface Without Adapters**

The custom streaming interface bus width varies with the IP core variation. In the figure, <*w*> = 2 for the 40GbE IP core and <*w*> = 4 for the 100GbE IP core.

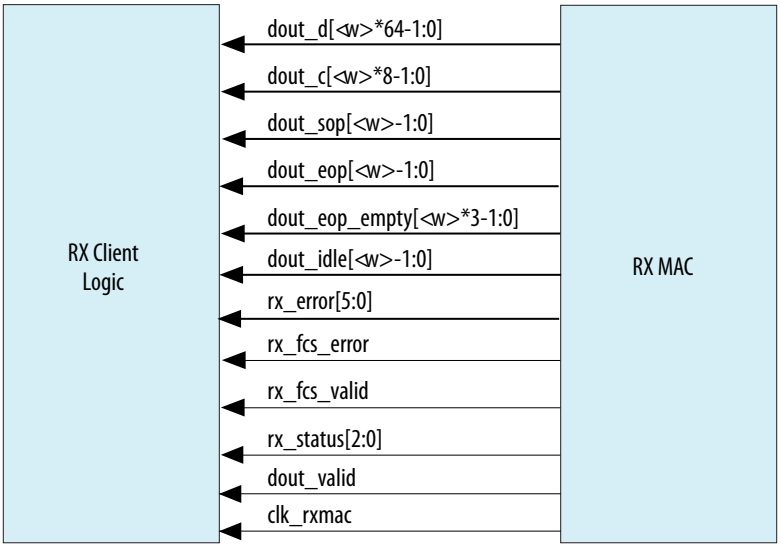



**Table 3-6: Signals of the RX Client Interface Without Adapters**

In the table, <*w*> = 2 for the 40GbE IP core and <*w*> = 4 for the 100GbE IP core. The signals are clocked by `clk_rxmac`.

| Signal Name | Direction | Description |
|---|---|---|
| `dout_d[<w>*64-1:0]` | Output | Received data and Idle bytes. In RX preamble pass-through mode, this bus also carries the preamble. |
| `dout_c[<w>*8-1:0]` | Output | Indicates control bytes on the data bus. Each bit of dout_c indicates whether the corresponding byte of `dout_d` is a control byte. A bit is asserted high if the corresponding byte on `dout_d` is an Idle byte or the Start byte, and has the value of zero if the corresponding byte is a data byte or, in preamble pass-through mode, a preamble or SFD byte. |
| `dout_sop[<w>-1:0]` | Output | Indicates the first data word of a frame, in the current `clk_rxmac` cycle. In RX preamble pass-through mode, the first data word is the word that contains the preamble. When the RX preamble pass-through feature is turned off, the first data word is the first word of Ethernet data that follows the preamble. This signal is one-hot encoded. |

| Signal Name | Direction | Description |
|---|---|---|
| dout_eop[<w>-1:0] | Output | Indicates the final word of a frame in the current clk_rxmac cycle. If CRC removal is disabled, this signal indicates the word with the final CRC byte. If CRC removal is enabled, this signal indicates the final word with data. This signal is one-hot encoded. |
| dout_eop_empty[<w>*3-1:0] | Output | Indicates the number of empty (invalid) bytes in the end-of-packet byte in the word indicated by dout_eop.<br>If dout_eop[z] has the value of 0, then the IP core sets the value of dout_eop_empty[(z+2):z] to 0. However, if dout_eop[z] has the value of 1, then you must use the value of dout_eop_empty[(z+2):z] to determine the number of empty (invalid) bytes in the end-of-packet word (and therefore, the end-of-packet byte).<br>For example, if you have a 100GbE IP core and you observe that in the current clk_rxmac clock cycle, dout_eop has the value of 4'b0100 and dout_eop_empty has the value of 12'b000_110_000_000, you can conclude that byte 6 in word 2 of dout_d is an end-of-packet byte. |
| dout_idle[<w>-1:0] | Output | Indicates the words in dout_d that hold Idle bytes or control information rather than Ethernet data. This signal is one-hot encoded. |
| rx_error[5:0] | Output | Reports certain types of errors in the Ethernet frame whose contents are currently being transmitted on the client interface. This signal is valid in EOP cycles only. To ensure you can identify the corresponding packet, you must turn on **Enable alignment EOP on FCS word** in the LL 40-100GbE parameter editor.<br>The individual bits report different types of errors:<br>• Bit [0]: Malformed packet error. If this bit has the value of 1, the packet is malformed. The IP core identifies a malformed packet when it receives a control character that is not a terminate character, while receiving the packet.<br>• Bit [1]: CRC error. If this bit has the value of 1, the IP core detected a CRC error in the frame.<br>If you turn on **Enable alignment EOP on FCS word**, this bit and the rx_fcs_error signal behave identically.<br>• Bit [2]: undersized payload. If this bit has the value of 1, the frame size is between nine and 63 bytes, inclusive. The IP core does not recognize an incoming frame of size eight bytes or less as a frame, and those cases are not reported here.<br>• Bit [3]: oversized payload. If this bit has the value of 1, the frame size is greater than the maximum frame size programmed in the MAX_RX_SIZE_CONFIG register at offset 0x506.<br>• Bit [4]: payload length error. If this bit has the value of 1, the payload received in the frame did not match the length field value, and the value in the length field is less than 1536 bytes. This bit only reports errors if you set bit [0] of the CFG_PLEN_CHECK register at offset 0x50A to the value of 1.<br>• Bit [5} Reserved. |

| Signal Name | Direction | Description |
|---|---|---|
| rx_fcs_error | Output | The current or most recent EOP byte is part of a frame with an incorrect FCS (CRC-32) value. By default, the IP core asserts rx_fcs_error in the same cycle as the dout_eop signal. However, if you turn off **Enable alignment EOP on FCS word** in the LL 40-100GbE parameter editor, the rx_fcs_error signal might lag the dout_eop signal for the frame.<br>Runt frames always force an FCS error condition. However, if a packet is eight bytes or smaller, it is considered a decoding error and not a runt frame, and the IP core does not flag it as a runt. |
| rx_fcs_valid | Output | When set, indicates that rx_fcs_error has a valid value in the current clock cycle.. |
| rx_status[2:0] | Output | Indicates the IP core received a control frame on the Ethernet link. This signal identifies the type of control frame the IP core is passing through to the client interface.<br>This signal is valid in EOP cycles only. To ensure you can identify the corresponding packet, you must turn on **Enable alignment EOP on FCS word** in the LL 40-100GbE parameter editor.<br>The individual bits report different types of received control frames:<br>• Bit [0]: Indicates the IP core received a standard flow control frame. If the IP core is in standard flow control mode and the cfg_fwd_ctrl bit of the RX_PAUSE_FWD register has the value of 0, this bit maintains the value of 0.<br>• Bit [1]: Indicates the IP core received a priority flow control frame. If the IP core is in priority flow control mode and the cfg_fwd_ctrl bit of the RX_PAUSE_FWD register has the value of 0, this bit maintains the value of 0.<br>• Bit [2]: Indicates the IP core received a control frame that is not a flow control frame. |
| dout_valid | Output | The dout_d bus contents are valid. This signal is occasionally deasserted due to clock crossing. |
| clk_rxmac | Output | RX MAC clock. The clock frequency should be 312.5 MHz in LL 40GbE IP cores, and 390.625 MHz in LL 100GbE IP cores. The clk_rxmac clock is derived from the recovered CDR clock. |

The data bytes use 100 Gigabit Media Independent Interface (CGMII-like) encoding. For packet payload bytes, the dout_c bit is set to 0 and the dout_d byte is the packet data. You can use this information to transmit out-of-spec data such as customized preambles when implementing non-standard variants of the *IEEE 802.3ba-2010 100G Ethernet Standard*.

In RX preamble pass-through mode, dout_c has the value of 1 while the start byte of the preamble is presented on the RX interface, and dout_c has the value of 0 while the remainder of the preamble sequence (six-byte preamble plus SFD byte) is presented on the RX interface.

**Related Information**

- **Low Latency 40-100GbE IP Core MAC Configuration Registers** on page 3-92
  Describes the `MAX_RX_SIZE_CONFIG` and `CFG_PLEN_CHECK` registers.
- **Control Frame Identification** on page 3-20

## Low Latency 100GbE CAUI–4 PHY

The Low Latency 100GbE PHY IP core configured in an Arria 10 GT device supports CAUI-4 PCS and PMA at 4 x 25.78125 Gbps.

**Related Information**

**Low Latency 40-100GbE IP Core Device Speed Grade Support** on page 1-5
Information about the device speed grades that support the 100GbE CAUI–4 IP core variation.

## External Reconfiguration Controller

Low Latency 40GbE and 100GbE IP cores that target a Stratix V device require an external reconfiguration controller.

Altera recommends that you configure an Altera Transceiver Reconfiguration Controller for your Stratix V 40-100GbE IP core.

**Related Information**

**External Transceiver Reconfiguration Controller Required in Stratix V Designs** on page 2-19
Information about configuring and connecting the Altera Transceiver Reconfiguration Controller. Includes signal descriptions.

## External Transceiver PLL

Low Latency 40GbE and 100GbE IP cores that target an Arria 10 device require an external transceiver PLL. The number and type of transceiver PLLs your design requires depends on the transceiver channels and available clock networks.

**Related Information**

- **Transceiver PLL Required in Arria 10 Designs** on page 2-20
  Information about configuring and connecting the external PLLs. Includes signal descriptions.
- **Arria 10 Transceiver PHY User Guide**
  Information about the correspondence between transceiver PLLs and transceiver channels, and information about how to configure an external transceiver PLL for your own design.

## External TX MAC PLL

If you turn on **Use external TX MAC PLL** in the LL 40-100GbE parameter editor, the IP core has an extra input port, `clk_txmac_in`, which drives the TX MAC clock. You must connect this input port to a clock source, usually a PLL on the device.

The port is expected to receive the clock from the external TX MAC PLL and drives the internal clock `clk_txmac`. The required TX MAC clock frequency is 312.5 MHz for 40GbE variations, and 390.625 MHz for 100GbE variations. User logic must drive `clk_txmac_in` from a PLL whose input is the PHY reference clock, `clk_ref`.

## Congestion and Flow Control Using Pause Frames

If you turn on flow control by setting **Flow control mode** to the value of **Standard flow control** or **Priority-based flow control** in the Low Latency 40-100GbE parameter editor, the Low Latency 40-100GbE IP core provides flow control to reduce congestion at the local or remote link partner. When either link partner experiences congestion, the respective transmit control sends pause frames. The pause frame instructs the remote transmitter to stop sending data for the duration that the congested receiver specified in an incoming XOFF frame. In the case of priority-based flow control, the pause frame applies to only the indicated priority class.

When the IP core receives the XOFF pause control frame, if the following conditions all hold, the IP core behavior depends on the flow control scheme.

- In priority-based flow control, the IP core informs the TX client of the received XOFF frame. The IP core continues to process packets it receives on the TX client interface. The IP core tracks the pause quanta countdown internally and informs the TX client when the pause period is ended.

  **Note:** In the priority-based flow control sceme, the client is responsible to throttle the flow of data to the TX client interface after the IP core informs the client of the received XOFF frame.
- In standard flow control, the IP core TX MAC does not process TX packets and in fact, prevents the client from sending additional data to the TX client interface, for the duration of the pause quanta of the incoming pause frame.

The IP core responds to an incoming XOFF pause control frame if the following conditions all hold:

- The relevant `cfg_enable` bit of the `RX_PAUSE_ENABLE` register has the value of 1.
- In the case of standard flow control, bit [0] of the `TX_XOF_EN` register also has the value of 1.
- Address matching is positive.

The pause quanta can be configured in the pause quanta register of the device sending XOFF frames. If the pause frame is received in the middle of a frame transmission, the transmitter finishes sending the current frame and then suspends transmission for a period specified by the pause quanta. Data transmission resumes when a pause frame with quanta of zero is received or when the timer has expired. The pause quanta received overrides any counter currently stored. When more than one pause quanta is sent, the value of the pause is set to the last quanta received.

XOFF pause frames stop the remote transmitter. XON pause frames let the remote transmitter resume data transmission.

One pause quanta fraction is equivalent to 512 bit times. The duration of a pause quantum depends on the client interface datapath width (256 bits for Low Latency 40GbE IP cores and 512 bits for Low Latency 100GbE IP cores) and on the system clock (`clk_txmac`) frequency (312.5 MHz for Low Latency 40GbE IP cores and 390.625 MHz for Low Latency 100GbE IP cores).

**Figure 3-18: The XOFF and XON Pause Frames for Standard Flow Control**

| XOFF Frame | | XON Frame |
|---|---|---|
| START[7:0] | | START[7:0] |
| PREAMBLE[47:0] | | PREAMBLE[47:0] |
| SFD[7:0] | | SFD[7:0] |
| DESTINATION ADDRESS[47:0] = 0x010000C28001 (3) | | DESTINATION ADDRESS[47:0] = 0x010000C28001 |
| SOURCE ADDRESS[47:0] | | SOURCE ADDRESS[47:0] |
| TYPE[15:0] = 0x8808 | | TYPE[15:0] = 0x8808 |
| OPCODE[15:0] = 0x001 (standard FC) | | OPCODE[15:0] = 0x001 (standard FC) |
| PAUSE QUANTA[15:0] = 0xP1, 0xP2 (4) | | PAUSE QUANTA[15:0] = 0x0000 |
| PAD[335:0] | | PAD[335:0] |
| CRC[31:0] | | CRC[31:0] |

(3) This is a multicast destination address.
(4) The bytes P1 and P2 are filled with the value configured in the `TX_PAUSE_QUANTA` register.

**Figure 3-19: The XOFF and XON Pause Frames for Priority Flow Control**

| XOFF Frame | XON Frame |
|---|---|
| START[7:0] | START[7:0] |
| PREAMBLE[47:0] | PREAMBLE[47:0] |
| SFD[7:0] | SFD[7:0] |
| DESTINATION ADDRESS[47:0] = 0x010000C28001 (3) | DESTINATION ADDRESS[47:0] = 0x010000C28001 |
| SOURCE ADDRESS[47:0] | SOURCE ADDRESS[47:0] |
| TYPE[15:0] = 0x8808 | TYPE[15:0] = 0x8808 |
| OPCODE[15:0] = 0x0101 (PFC) | OPCODE[15:0] = 0x0101 (PFC) |
| PRIORITY_ENABLE[15:0](5) | PRIORITY_ENABLE[15:0] (6) |
| TIME0[15:0] (7) | TIME0[15:0] = 0x0000 |
| ... | ... |
| TIME7[15:0] (7) | TIME7[15:0] = 0x0000 |
| PAD[207:0] | PAD[207:0] |
| CRC[31:0] | CRC[31:0] |

## Conditions Triggering XOFF Frame Transmission

The LL 40-100GbE IP core supports retransmission. In retransmission mode, the IP core retransmits a XOFF frame periodically, extending the pause time, based on signal values.

The TX MAC transmits XOFF frames when one of the following conditions occurs:

- Client requests XOFF transmission—A client can explicitly request that XOFF frames be sent using the pause control interface signal. When `pause_insert_tx` is asserted, an XOFF frame is sent to the Ethernet network when the current frame transmission completes.
- Host (software) requests XOFF transmission—Setting the pause request register triggers a request that an XOFF frame be sent.
- Retransmission mode—If the retransmit hold-off enable bit has the value of 1, and the `pause_insert_tx` signal remains asserted or the pause request register value remains high, when the time duration specified in the hold-off quanta register has lapsed after the previous XOFF transmission, the TX MAC sends another XOFF frame to the Ethernet network. While the IP core is paused in retransmission mode, you cannot use either of the other two methods to trigger a new XOFF frame: the signal or register value is already high.

**Note:** Altera recommends that you use the `pause_insert_tx` signal to backpressure the remote Ethernet node.

(5) Bit [n] has the value of 1 if the TIMEn field is valid.
(6) Bit [n] has the value of 1 if the XON request applies to priority queue n.
(7) The TIMEn field is filled with the value available in the `TX_PAUSE_QUANTA`register when the `TX_PAUSE_QNUMBER` register holds the value of n.

**Note:** Altera recommends that you set and maintain the value of the retransmit hold-off enable bit at 1 to control the rate of XOFF pause frame transmission.

### Conditions Triggering XON Frame Transmission

The TX MAC transmits XON frames when one of the following conditions occurs:

- Client requests XON transmission—A client can explicitly request that XON frames be sent using the pause control interface signal. When `pause_insert_tx` is de-asserted, an XON frame is sent to the Ethernet network when the current frame transmission completes.
- Host (software) requests XON transmission—Resetting the pause request register triggers a request that an XON frame be sent.

## Pause Control and Generation Interface

The pause control interface implements flow control as specified by the *IEEE 802.3ba 2010 100G Ethernet Standard*. If you turn on priority-based flow control, the interface implements the *IEEE Standard 802.1Qbb*.The pause logic, upon receiving a pause packet, temporarily stops packet transmission, and can pass the pause packets through as normal traffic or drop the pause control frames in the RX direction.

**Table 3-7: Pause Control and Generation Signals**

Describes the signals that implement pause control. These signals are available only if you turn on flow control in the LL 40-100GbE parameter editor or you upgrade your IP core from an earlier release in which the signals are available in any case.

| Signal Name | Direction | Description |
|---|---|---|
| `pause_insert_tx[N-1:0]`[8] | Input | Level signal which directs the IP core to insert a pause frame for priority traffic class [n] on the Ethernet link. If bit [n] of the `TX_PAUSE_EN` register has the value of 1, the IP core transmits an XOFF frame when this signal is first asserted. If you enable retransmission, the IP core continues to transmit XOFF frames periodically until the signal is de-asserted. When the signal is deasserted, the IP core inserts an XON frame. |
| `pause_receive_rx[N-1:0]`[8] | Output | Asserted to indicate an RX pause signal match. The IP core asserts bit [n] of this signal when it receives a pause request with an address match, to signal the TX MAC to throttle its transmissions from priority queue [n] on the Ethernet link. |

**Related Information**

- **Pause Control Frame Filtering** on page 3-33
  Information about enabling and disabling the pause packets pass-through.
- **Pause Registers** on page 3-94
  You can access the pause functionality using the pause registers for any 40-100GbE IP core variation that includes a MAC component. Values you program in the registers specify the pause quanta.

[8] N is the number of priority queues. If the IP core implements Ethernet standard flow control, N is 1.

## Pause Control Frame Filtering

The Low Latency 40GbE and 100GbE IP cores support options to enable or disable the following features for incoming pause control frames. These options are available if you set the **Flow control mode** parameter to the value of **Standard flow control** or **Priority-based flow control**.

- Processing—You can enable or disable pause frame processing. If you disable pause frame processing, the IP core does not modify its behavior in response to incoming pause frames on the Ethernet link. You can enable or disable pause frame processing with the `cfg_enable` bit of the `RX_PAUSE_ENABLE` register. By default, RX pause frame processing is enabled.
- Filtering—If pause frame processing is enabled, the IP core automatically performs address filtering on incoming pause control frames before processing them. You set the matching address value in these registers:
  - `RX_PAUSE_DADDRL[31:0]` at offset 0x707
  - `RX_PAUSE_DADDRH[15:0]` at offset 0x708

  The 48-bit address ({`RX_PAUSE_DADDRH[15:0]`,`RX_PAUSE_DADDRL[31:0]`} can be an individual MAC address, a multicast address, or a broadcast address.
- TX MAC filtering—For standard flow control only, Altera provides an additional level of filtering to enable or disable the TX MAC from responding to notification from the RX MAC that it received an incoming pause frame with an address match. Even if the RX MAC processes an incoming pause frame, you can separately set the TX MAC to ignore the RX MAC request to pause outgoing frames, by setting bit [0] of the `TX_XOF_EN` register to the value of 0. By default this register field has the value of 1.
- Pass-through—The Low Latency 40GbE and 100GbE MAC IP cores can pass the matching pause packets through as normal traffic or drop these pause control frames in the RX direction. You can enable and disable pass-through with the `cfg_fwd_ctrl` bit of the `RX_PAUSE_FWD` register. By default, pass-through is disabled. All non-matching pause frames are passed through to the RX client interface irrespective of the `cfg_fwd_ctrl` setting.

The following rules define pause control frames filtering control:

1. If you have disabled pause frame processing, by setting the `cfg_enable` bit of the `RX_PAUSE_ENABLE` register to the value of 0, the IP core drops packets that enter the RX MAC and match the destination address, length, and type of 0x8808 with an opcode of 0x1 (pause packets)
2. If you have enabled pause frame processing, and the destination address in the pause frame is a match, when the RX MAC receives a pause packet it passes a pause request to the TX MAC. The RX MAC only processes pause packets with a valid packet multicast address or a destination address matching the destination address specified in the `RX_PAUSE_DADDR1` and `RX_PAUSE_DADDR0` registers, or in the `RX_PAFC_DADDRH` and `RX_PFC_DADDRL` registers, as appropriate for the flow control mode . If you have turned on pause frame pass-through, the RX MAC also forwards the pause frame to the RX client interface. If you have not turned on pause frame pass-through, the RX MAC does not forward the matching pause frame to the RX client interface.
3. In priority-based flow control, or in standard flow control if you have enabled TX MAC filtering, when the TX MAC receives a pause request from the RX MAC, it pauses transmission on the TX Ethernet link.

Pause packet pass-through does not affect the pause functionality in the TX or RX MAC.

## Link Fault Signaling Interface

If you turn on **Enable link fault generation** in the Low Latency 40-100GbE parameter editor, the Low Latency 40-100GbE IP core provides link fault signaling as defined in the *IEEE 802.3ba-2010 100G Ethernet Standard* and Clause 66 of the *IEEE 802.3-2012 Ethernet Standard*, based on the `LINK_FAULT_CONFIG` register settings.. The Low Latency 40GbE and 100GbE MAC include a Reconciliation Sublayer (RS) located between the MAC and the XLGMII or CGMII to manage local and remote

faults. Link fault signaling on the Ethernet link is disabled by default but can be enabled by bit [0] of the LINK_FAULT_CONFIG register. When the LINK_FAULT_CONFIG register bits [1:0] have the value of 2'b01, link fault signaling is enabled in normal bidirectional mode. In this mode, the local RS TX logic transmits remote fault sequences in case of a local fault and transmits IDLE control words in case of a remote fault.

If you turn on bit [1] of the LINK_FAULT_CONFIG register, the IP core conforms to Clause 66 of the *IEEE 802.3-2012 Ethernet Standard*. When LINK_FAULT_CONFIG[1:0] has the value of 2'b11, the IP core transmits the fault sequence ordered sets in the interpacket gaps according to the clause requirements.

The RS RX logic sets remote_fault_status or local_fault_status to 1 when the RS RX block receives remote fault or local fault sequence ordered sets. When valid data is received in more than 127 columns, the RS RX logic resets the relevant fault status (remote_fault_status or local_fault_status) to 0.

The IEEE standard specifies RS monitoring of RXC<7:0> and RXD<63:0> for Sequence ordered_sets. For more information, refer to *Figure 81–9—Link Fault Signaling state diagram* and *Table 81-5—Sequence* ordered_sets in the *IEEE 802.3ba 2010 100G Ethernet Standard*. The variable link_fault is set to indicate the value of an RX Sequence ordered_set when four fault_sequences containing the same fault value are received with fault sequences separated by less than 128 columns and with no intervening fault_sequences of different fault values. The variable link_fault is set to OK following any interval of 128 columns not containing a remote fault or local fault Sequence ordered_set.

### Table 3-8: Signals of the Link Fault Signaling Interface

These signals are available only if you turn on **Enable link fault generation** in the LL 40-100GbE parameter editor.

| Signal Name | Direction | Description |
|---|---|---|
| remote_fault_status | Output | Asserted by the IP core when it detects a real-time remote fault in the RX MAC. The IP core asserts this signal regardless of the settings in the LINK_FAULT_CONFIG register.<br>This signal is clocked by clk_rxmac. |
| local_fault_status | Output | Asserted by the IP core when it detects a real-time local fault in the RX MAC. The IP core asserts this signal regardless of the settings in the LINK_FAULT_CONFIG register.<br>This signal is clocked by clk_rxmac. |
| unidirectional_en | Output | The IP core asserts this signal if it includes Clause 66 support for remote link fault reporting on the Ethernet link. Connects to the Unidir Enable field in bit [1] of the LINK_FAULT_CONFIG register at offset 0x405.<br>This signal is clocked by clk_txmac. |
| link_fault_gen_en | Output | The IP core asserts this signal if the PCS is enabled to generate a remote fault sequence on the Ethernet link when appropriate. Connects to the Link Fault Reporting Enable field in bit [0] of the LINK_FAULT_CONFIG register at offset 0x405.<br>This signal is clocked by clk_txmac. |

**Related Information**

- **Link Fault Signaling Registers** on page 3-79
  Information about the `Link Fault Reporting Enable` register and the `Unidir Enable` register.
- **IEEE website**
  The *IEEE 802.3ba –2010 100G Ethernet Standard* and the *IEEE 802.3 –2012 Ethernet Standard* are available on the IEEE website.

## Statistics Counters Interface

The statistics counters modules are synthesis options that you select in the Low Latency 40-100GbE parameter editor. However, the statistics status bit output vectors are provided whether you select the statistics counters module option or not.

The increment vectors are brought to the top level as output ports.The increment vectors also function internally as input ports to the control and status registers (CSR).

**Table 3-9: Statistics Counters Increment Vectors**

The TX statistics counter increment vectors are clocked by the `clk_txmac` clock, and the RX statistics counter increment vectors are clocked by the `clk_rxmac` clock.

| Name | Signal Direction | Description |
|---|---|---|
| TX Statistics Counter Increment Vectors | | |
| `tx_inc_64` | Output | Asserted for one cycle when a 64-byte TX frame is transmitted. |
| `tx_inc_127` | Output | Asserted for one cycle when a 65–127 byte TX frame is transmitted. |
| `tx_inc_255` | Output | Asserted for one cycle when a 128–255 byte TX frame is transmitted. |
| `tx_inc_511` | Output | Asserted for one cycle when a 256–511 byte TX frame is transmitted. |
| `tx_inc_1023` | Output | Asserted for one cycle when a 512–1023 byte TX frame is transmitted. |
| `tx_inc_1518` | Output | Asserted for one cycle when a 1024–1518 byte TX frame is transmitted. |
| `tx_inc_max` | Output | Asserted for one cycle when a maximum-size TX frame is transmitted.<br>You program the maximum number of bytes in the `MAX_TX_SIZE_CONFIG` register. |
| `tx_inc_over` | Output | Asserted for one cycle when an oversized TX frame is transmitted.<br>You program the maximum number of bytes in the `MAX_TX_SIZE_CONFIG` register. |
| `tx_inc_mcast_data_err` | Output | Asserted for one cycle when an errored multicast TX frame, excluding control frames, is transmitted. |

| Name | Signal Direction | Description |
|---|---|---|
| tx_inc_mcast_data_ok | Output | Asserted for one cycle when a valid multicast TX frame, excluding control frames, is transmitted. |
| tx_inc_bcast_data_err | Output | Asserted for one cycle when an errored broadcast TX frame, excluding control frames, is transmitted. |
| tx_inc_bcast_data_ok | Output | Asserted for one cycle when a valid broadcast TX frame, excluding control frames, is transmitted. |
| tx_inc_ucast_data_err | Output | Asserted for one cycle when an errored unicast TX frame, excluding control frames, is transmitted. |
| tx_inc_ucast_data_ok | Output | Asserted for one cycle when a valid unicast TX frame, excluding control frames, is transmitted. |
| tx_inc_mcast_ctrl | Output | Asserted for one cycle when a valid multicast TX frame is transmitted. |
| tx_inc_bcast_ctrl | Output | Asserted for one cycle when a valid broadcast TX frame is transmitted. |
| tx_inc_ucast_ctrl | Output | Asserted for one cycle when a valid unicast TX frame is transmitted. |
| tx_inc_pause | Output | Asserted for one cycle when a valid pause TX frame is transmitted. |
| tx_inc_fcs_err | Output | Asserted for one cycle when a TX packet with FCS errors is transmitted. |
| tx_inc_fragment | Output | Asserted for one cycle when a TX frame less than 64 bytes and reporting a CRC error is transmitted. |
| tx_inc_jabber | Output | Asserted for one cycle when an oversized TX frame reporting a CRC error is transmitted. |
| tx_inc_sizeok_fcserr | Output | Asserted for one cycle when a valid TX frame with FCS errors is transmitted. |
| RX Statistics Counter Increment Vectors | | |
| rx_inc_runt | Output | Asserted for one cycle when an RX runt packet is received. |
| rx_inc_64 | Output | Asserted for one cycle when a 64-byte RX frame is received. |
| rx_inc_127 | Output | Asserted for one cycle when a 65–127 byte RX frame is received. |

| Name | Signal Direction | Description |
|---|---|---|
| rx_inc_255 | Output | Asserted for one cycle when a 128–255 byte RX frame is received. |
| rx_inc_511 | Output | Asserted for one cycle when a 256–511 byte RX frame is received. |
| rx_inc_1023 | Output | Asserted for one cycle when a 512–1023 byte RX frame is received. |
| rx_inc_1518 | Output | Asserted for one cycle when a 1024–1518 byte RX frame is received. |
| rx_inc_max | Output | Asserted for one cycle when a maximum-size RX frame is received.<br>You program the maximum number of bytes in the MAX_RX_SIZE_CONFIG register. |
| rx_inc_over | Output | Asserted for one cycle when an oversized RX frame is received.<br>You program the maximum number of bytes in the MAX_RX_SIZE_CONFIG register. |
| rx_inc_mcast_data_err | Output | Asserted for one cycle when an errored multicast RX frame, excluding control frames, is received. |
| rx_inc_mcast_data_ok | Output | Asserted for one cycle when valid a multicast RX frame, excluding control frames, is received. |
| rx_inc_bcast_data_err | Output | Asserted for one cycle when an errored broadcast RX frame, excluding control frames, is received. |
| rx_inc_bcast_data_ok | Output | Asserted for one cycle when a valid broadcast RX frame, excluding control frames, is received. |
| rx_inc_ucast_data_err | Output | Asserted for one cycle when an errored unicast RX frame, excluding control frames, is received. |
| rx_inc_ucast_data_ok | Output | Asserted for one cycle when a valid unicast RX frame, excluding control frames, is received. |
| rx_inc_mcast_ctrl | Output | Asserted for one cycle when a valid multicast RX frame is received. |
| rx_inc_bcast_ctrl | Output | Asserted for one cycle when a valid broadcast RX frame is received. |
| rx_inc_ucast_ctrl | Output | Asserted for one cycle when a valid unicast RX frame is received. |
| rx_inc_pause | Output | Asserted for one cycle when valid RX pause frames are received. |

| Name | Signal Direction | Description |
|---|---|---|
| rx_inc_fcs_err | Output | Asserted for one cycle when a RX packet with FCS errors is received. |
| rx_inc_fragment | Output | Asserted for one cycle when a RX frame less than 64 bytes and reporting a CRC error is received. |
| rx_inc_jabber | Output | Asserted for one cycle when an oversized RX frame reporting a CRC error is received. |
| rx_inc_sizeok_fcserr | Output | Asserted for one cycle when a valid RX frame with FCS errors is received. |
| rx_inc_pause_ctrl_err | Output | Asserted for one cycle when an errored pause RX frame is received. |
| rx_inc_mcast_ctrl_err | Output | Asserted for one cycle when an errored multicast RX control frame is received. |
| rx_inc_bcast_ctrl_err | Output | Asserted for one cycle when an errored broadcast RX control frame is received. |
| rx_inc_ucast_ctrl_err | Output | Asserted for one cycle when an errored unicast RX control frame is received. |

**Related Information**

- **TX Statistics Registers** on page 3-101
- **RX Statistics Registers** on page 3-107

## OctetOK Count Interface

The statistics counters include two 64-bit counters, `RxOctetsOK` and `TxOctetsOK`, which count the payload bytes (octets) in frames with no FCS, undersized, oversized, or payload length errors. The `RxOctetsOK` register maintains a cumulative count of the payload bytes in the qualifying received frames, and complies with section 5.2.1.14 of the *IEEE Standard 802.3-2008*. The `TxOctetsOK` register maintains a cumulative count of the payload bytes in the qualifying transmitted frames, and complies with section 5.2.2.18 of the *IEEE Standard 802.3-2008*.

To support payload size checking per frame, the LL 40-100GbE IP core provides an octetOK count interface instead of standard increment vectors. For most purposes, the number of payload bytes per frame is of more interest than the cumulative count, and an increment vector that pulses when the counter increments would be difficult to track. For each of these two registers, the IP core maintains two signals. A 16-bit signal provides a count of the payload bytes in the current frame, and the other signal pulses to indicate when the first signal is valid. The per-frame count is valid only when the valid signal is asserted. All signals in this interface are functional even if you do not turn on the corresponding statistics module.

**Table 3-10: OctetOK Count Interface Signals**

The signals for received frames are clocked by the `clk_rxmac` clock. The signals for transmitted frames are clocked by the `clk_txmac` clock.

| Name | Signal Direction | Description |
|---|---|---|
| `tx_inc_octetsOK[15:0]` | Output | When `tx_inc_octetsOK_valid` is asserted, `tx_inc_octetsOK[15:0]` holds the count of payload bytes in the current valid frame. |
| `tx_inc_octetsOK_valid` | Output | Pulses to indicate that `tx_inc_octetsOK[15:0]` currently holds the number of payload bytes for the current transmitted frame, and that the current frame is a qualifying frame. A qualifying frame has no FCS errors, no oversized error, no undersized error, and no payload length error. |
| `rx_inc_octetsOK[15:0]` | Output | When `rx_inc_octetsOK_valid` is asserted, `rx_inc_octetsOK[15:0]` holds the count of payload bytes in the current valid frame. |
| `rx_inc_octetsOK_valid` | Output | Pulses to indicate that `rx_inc_octetsOK[15:0]` currently holds the number of payload bytes for the current received frame, and that the current frame is a qualifying frame. A qualifying frame has no FCS errors, no oversized error, no undersized error, and no payload length error. |

## 1588 Precision Time Protocol Interfaces

If you turn on **Enable 1588 PTP**, the Low Latency 40-100GbE IP core processes and provides 1588 Precision Time Protocol (PTP) timestamp information as defined in the *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard*.

1588 PTP packets carry timestamp information. The Low Latency 40-100GbE IP core updates the incoming timestamp information in a 1588 PTP packet to transmit a correct updated timestamp with the data it transmits on the Ethernet link, using a one-step or two-step clock.

A fingerprint can accompany a 1588 PTP packet. You can use this information for client identification and other client uses. If provided fingerprint information, the IP core passes it through unchanged.

The IP core connects to a time-of-day (TOD) module that continuously provides the current time of day based on the input clock frequency. Because the module is outside the LL 40-100GbE IP core, you can use the same module to provide the current time of day for multiple modules in your system.

**Related Information**

- **External Time-of-Day Module for Variations with 1588 PTP Feature** on page 2-22
- **1588 PTP Registers** on page 3-112
- **IEEE website**
  The *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard* is available on the IEEE website.

### Implementing a 1588 System That Includes a LL 40-100GbE IP Core

The 1588 specification in *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard* describes various systems you can implement in hardware and

software to synchronize clocks in a distributed system by communicating offset and frequency correction information between master and slave clocks in arbitrarily complex systems. A 1588 system that includes the LL 40-100GbE IP core with 1588 PTP functionality uses the incoming and outgoing timestamp information from the IP core and the other modules in the system to synchronize clocks across the system.

The LL 40-100GbE IP core with 1588 PTP functionality provides the timestamp manipulation and basic update capabilities required to integrate your IP core in a 1588 system. You can specify that packets are PTP packets, and how the IP core should update incoming timestamps from the client interface before transmitting them on the Ethernet link. The IP core does not implement the event messaging layers of the protocol, but rather provides the basic hardware capabilities that support a system in implementing the full 1588 protocol.

**Figure 3-20: Example Ethernet System with Ordinary Clock Master and Ordinary Clock Slave**

You can implement both master and slave clocks using the Altera LL 40-100GbE IP core with 1588 PTP functionality.

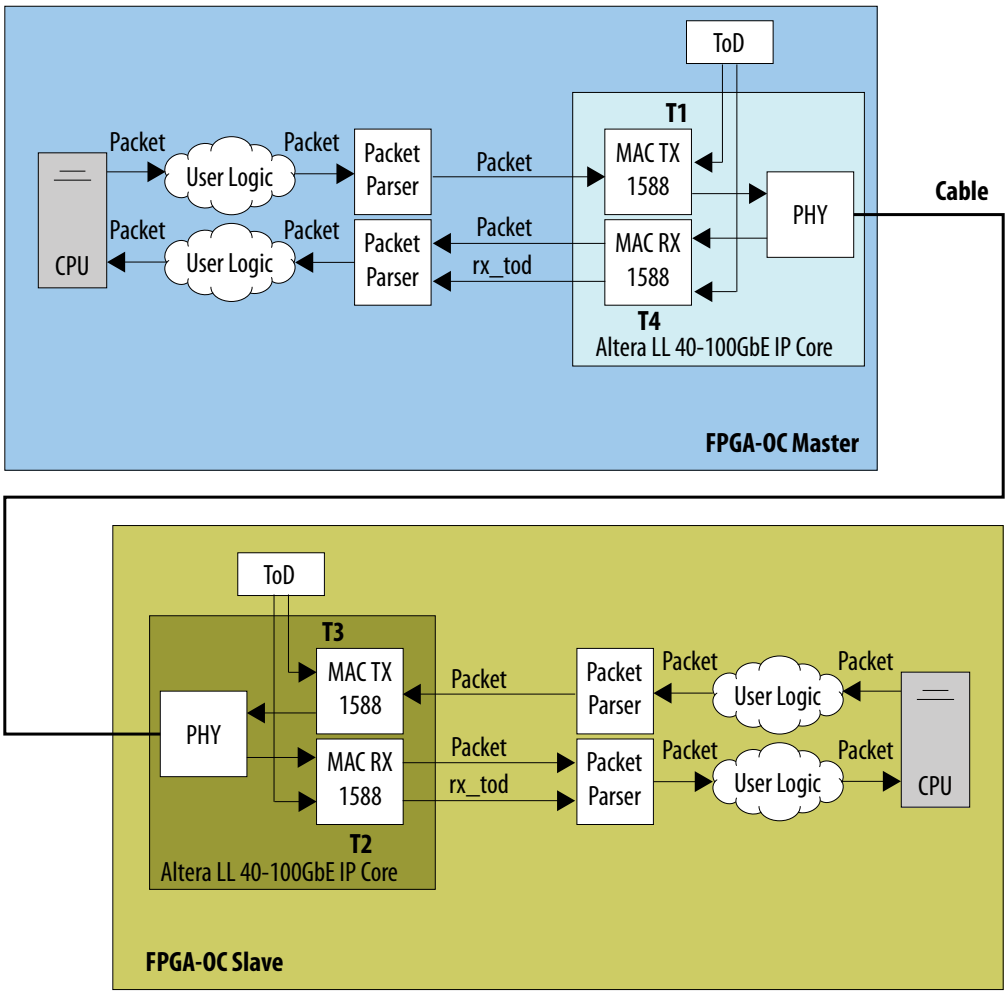


**Figure 3-21: Hardware Configuration Example Using Altera LL 40-100GbE IP core in a 1588 System in Transparent Clock Mode**

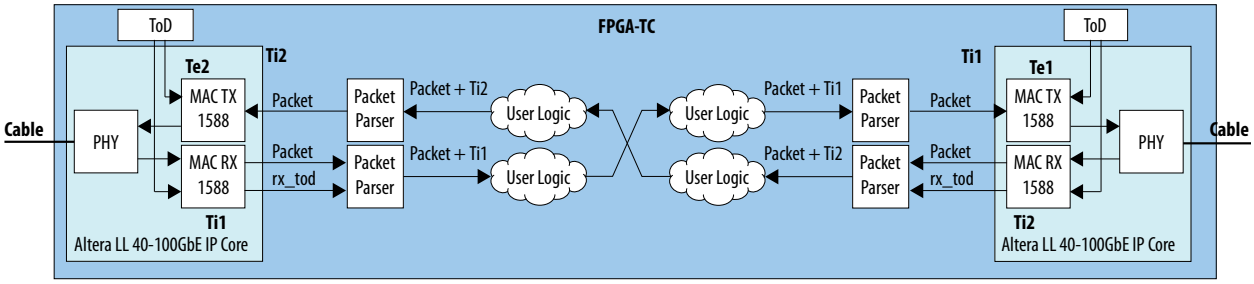



**Figure 3-22: Software Flow Using Transparent Clock Mode System**

This figure from the 1588 standard is augmented with the timestamp labels shown in the transparent clock system figure. A precise description of the software requirements is beyond the scope of this document. Refer to the 1588 standard.

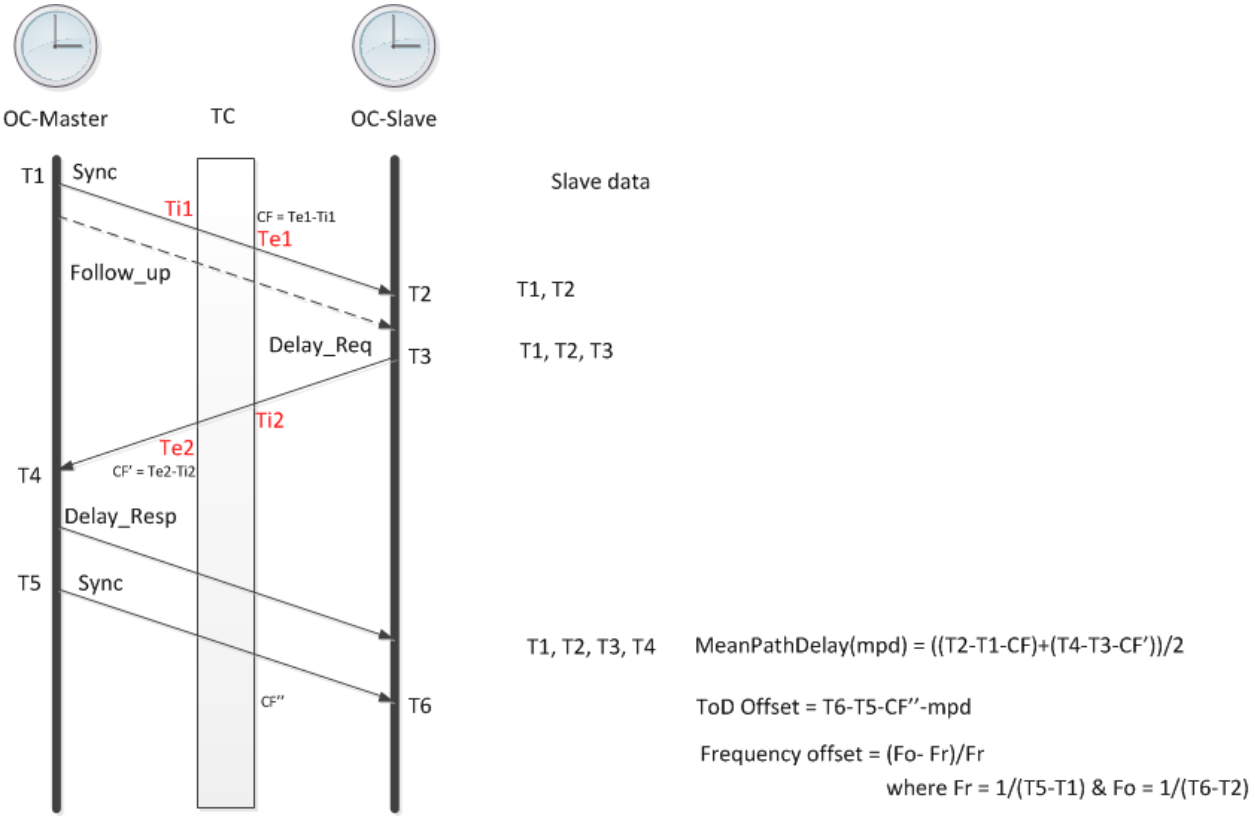
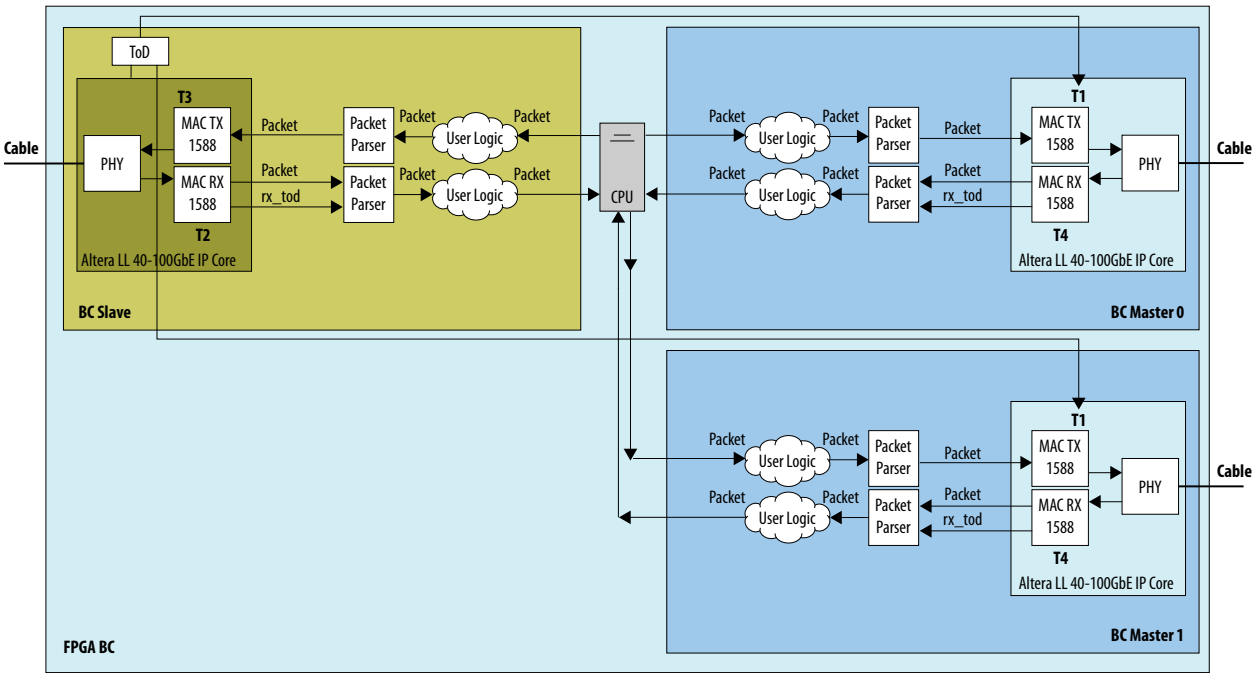


**Figure 3-23: Example Boundary Clock with One Slave Port and Two Master Ports**

You can implement a 1588 system in boundary clock mode using the LL 40-100GbE IP core with 1588 PTP functionality.




**Related Information**

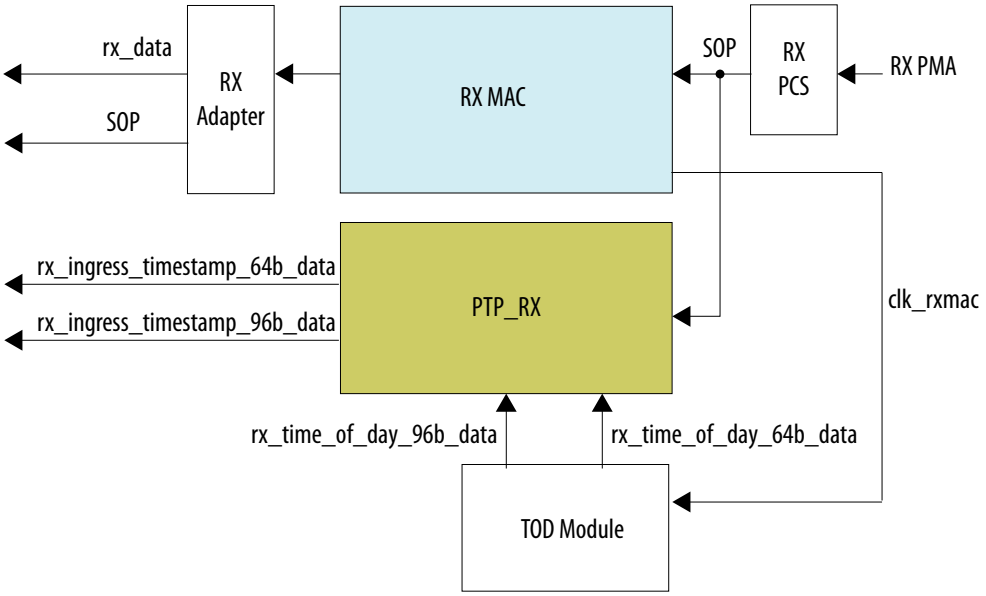
**IEEE website**

The *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard* is available on the IEEE website.

## PTP Receive Functionality

If you turn on **Enable 1588 PTP** in the Low Latency 40-100GbE parameter editor, the IP core provides a 96-bit (V2 format) or 64-bit timestamp with every packet on the RX client interface, whether it is a 1588 PTP packet or not. The value on the timestamp bus (`rx_ingress_timestamp_96b_data[95:0]` or `rx_ingress_timestamp_64b_data[63:0]` or both, if present) is valid in the same clock cycle as the RX SOP signal. The value on the timestamp bus is not the current timestamp; instead, it is the timestamp from the time when the IP core received the packet on the Ethernet link. The IP core captures the time-of-day from the TOD module on `rx_time_of_data_96b_data` or `rx_time_of_day_64b_data` at the time it receives the packet on the Ethernet link, and sends that timestamp to the client on the RX SOP cycle on the timestamp bus `rx_ingress_timestamp_96b_data[95:0]` or `rx_ingress_timestamp_64b_data[63:0]` or both, if present. User logic can use this timestamp or ignore it.

**Figure 3-24: PTP Receive Block Diagram**




**Related Information**

**IEEE website**

The *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard* is available on the IEEE website.

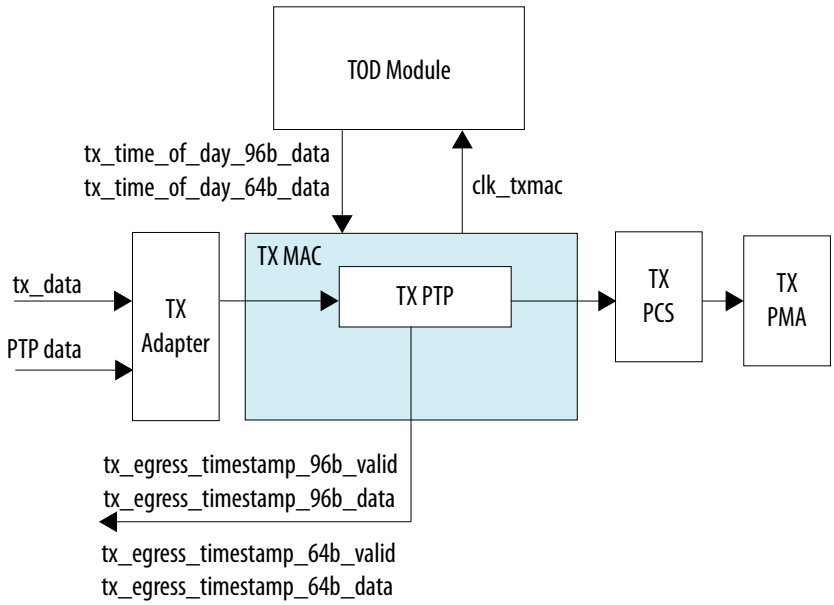
## PTP Transmit Functionality

When you send a 1588 PTP packet to a Low Latency 40-100GbE IP core with **Enable 1588 PTP** turned on in the parameter editor, you must assert one and only one of the following input signals with the TX SOP signal to tell the IP core the incoming packet is a 1588 PTP packet:

- `tx_egress_timestamp_request_valid`: assert this signal to tell the IP core to process the current packet in two-step processing mode.
- `tx_etstamp_ins_ctrl_timestamp_insert`: assert this signal to tell the IP core to process the current packet in one-step processing mode and to insert the exit timestamp for the packet in the packet (insertion mode).
- `tx_etstamp_ins_ctrl_residence_time_update`: assert this signal to tell the IP core to process the current packet in one-step processing mode and to update the timestamp in the packet by adding the latency through the IP core (the residence time in the IP core) to the cumulative delay field maintained in the packet (correction mode). This mode supports transparent clock systems.

The IP core transmits the 1588 PTP packet in an Ethernet frame after PTP processing.

**Figure 3-25: PTP Transmit Block Diagram**




In one-step mode, the IP core either overwrites the timestamp information provided at the user-specified offset with the packet exit timestamp (insertion mode), or adds the residence time in this system to the value at the specified offset (correction mode). You tell the IP core how to process the timestamp by asserting the appropriate signal with the TX SOP signal. You must specify the offset of the timestamp in the packet (`tx_etstamp_ins_ctrl_offset_timestamp`) in insertion mode, or the offset of the correction field in the packet (`tx_etstamp_ins_ctrl_offset_correction_field`) in correction mode. In addition, the IP core zeroes out or updates the UDP checksum, or leaves the UDP checksum as is, depending on the mutually exclusive `tx_etstamp_ins_ctrl_checksum_zero` and `tx_etstamp_ins_ctrl_checksum_correct` signals.

Two-step PTP processing ignores the values on the one-step processing signals. In two-step processing mode, the IP core does not modify the current timestamp in the packet. Instead, the IP core transmits a two-step derived timestamp on the separate `tx_egress_timestamp_96b_data[95:0]` or `tx_egress_timestamp_64b_data[63:0]` bus, when it begins transmitting the Ethernet frame. The value on the `tx_egress_timestamp_{96b,64b}_data` bus is the packet exit timestamp. The `tx_egress_timestamp_{96b,64b}_data` bus holds a valid value when the corresponding `tx_egress_timestamp_{96b,64b}_valid` signal is asserted.

In addition, to help the client to identify the packet, you can specify a fingerprint to be passed by the IP core in the same clock cycle with the timestamp. To specify the number of distinct fingerprint values the IP core can handle, set the **Timestamp fingerprint width** parameter to the desired number of bits `W`. You provide the fingerprint value to the IP core in the `tx_egress_timestamp_request_fingerprint[(W-1):0]` signal. The IP core then drives the fingerprint on the appropriate `tx_egress_timestamp_{96b,64b}_fingerprint[(W-1):0]` port with the corresponding output timestamp, when it asserts the `tx_egress_timestamp_{96b,64b}_valid` signal.

The IP core calculates the packet exit timestamp.

exit TOD = entry TOD + IP core maintained expected latency + user-specified extra latency

- entry TOD is the value in `tx_time_of_day_96b_data` or `tx_time_of_day_64b_data` when the packet enters the IP core.
- The expected latency through the IP core is a static value. The IP core maintains this value internally.
- The IP core reads the user-specified extra latency from the `TX_PTP_EXTRA_LATENCY` register. This option is provided for user flexibility.

The IP core provides the exit TOD differently in different processing modes.

- In two-step mode, the IP core drives the exit TOD on `tx_egress_timestamp_96b_data` and on `tx_egress_timestamp_64b_data`, as available.
- In one-step processing insertion mode, the IP core inserts the exit TOD in the timestamp field of the packet at the offset you specify in `tx_etstamp_ins_ctrl_offset_timestamp`.
- In one-step processing correction mode, the IP core calculates the exit TOD and uses it only to calculate the residence time.

In one-step processing correction mode, the IP core calculates the updated correction field value:

exit correction field value = entry correction field value + residence time + asymmetry extra latency

- residence time = exit TOD – entry (ingress) timestamp.
- entry (ingress) timestamp is the value on `tx_etstamp_ins_ctrl_ingress_timestamp_{95,64}b` in the SOP cycle when the IP core received the packet on the TX client interface. The application is responsible to drive this signal with the correct value for the cumulative calculation. The correct value depends on system configuration.
- The IP core reads the asymmetry extra latency from the `TX_PTP_ASYM_DELAY` register if the `tx_egress_asymmetry_update` signal is asserted. This option is provided for additional user-defined precision. You can set the value of this register and set the `tx_egress_asymmetry_update` signal to indicate the register value should be included in the latency calculation.

**Related Information**

- **1588 PTP Registers** on page 3-112
- **IEEE website**
  The *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard* is available on the IEEE website.

## External Time-of-Day Module for 1588 PTP Variations

Low Latency 40-100GbE IP cores that include the 1588 PTP module require an external time-of-day (TOD) module to provide the current time-of-day in each clock cycle, based on the incoming clock. The TOD module must update the time-of-day output value on every clock cycle, and must provide the TOD value in the V2 format (96 bits) or the 64-bit TOD format, or both.

The example project you can generate for your IP core PTP variation includes two TOD modules, one for the RX MAC and one for the TX MAC, and demonstrates their connections to the IP core.

**Related Information**

**External Time-of-Day Module for Variations with 1588 PTP Feature** on page 2-22

## PTP Timestamp and TOD Formats

The Low Latency 40-100GbE IP core supports a 96-bit timestamp (V2 format) or a 64-bit timestamp (V1 format) in PTP packets. It supports the PHY operating speed random error with timestamp accuracy of ± 7 ns. The 64-bit timestamp and TOD signals of the IP core are in an Altera-defined 64-bit format that is distinct from the V1 format, for improved efficiency in one-step processing correction mode. Therefore, if

your system need not handle any packets in one-step processing correction mode, Altera recommends that you turn off the **Enable 64b Time of Day Format** parameter.

You control the format or formats the IP core supports with the **Enable 96b Time of Day Format** and **Enable 64b Time of Day Format** parameters. If you turn on **Enable 96b Time of Day Format**, your IP core can support two-step processing mode, one-step processing insertion mode, and one-step processing correction mode, and can support both V1 and V2 formats. You can turn on **Enable 64b Time of Day Format** and turn off **Enable 96b Time of Day Format** to support one-step processing correction mode more efficiently. However, if you do so, your IP core variation cannot support two-step processing mode and cannot support one-step processing insertion mode. If you turn on both of these parameters, the value you drive on the `tx_estamp_ins_ctrl_timestamp_format` or `tx_etstamp_ins_ctrl_residence_time_calc_format` signal determines the format the IP core supports for the current packet.

The IP core completes all internal processing in the V2 format. However, if you specify V1 format for a particular PTP packet in one-step insertion mode, the IP core inserts the appropriate V1-format timestamp in the outgoing packet on the Ethernet link.

### V2 Format

The IP core maintains the time-of-day (TOD) in V2 format according to the IEEE specification::

- Bits [95:48]: Seconds (48 bits).
- Bits [47:16]: Nanoseconds (32 bits). This field overflows at 1 billion.
- Bits [15:0]: Fractions of nanosecond (16 bits). This field is a true fraction; it overflows at 0xFFFF.

The IP core can receive time-of-day information from the TOD module in V2 format or in 64-bit TOD format, or both, depending on your settings for the **Enable 64b Time of Day Format** and **Enable 96b Time of Day Format** parameters.

### V1 Format

V1 timestamp format is specified in the IEEE specification:

- Bits [63:32]: Seconds (32 bits).
- Bits [31:0]: Nanoseconds (32 bits). This field overflows at 1 billion.

### Altera 64-Bit TOD Format

The Altera 64-bit TOD format is distinct from the V1 format and supports a longer time delay. It is intended for use in transparent clock systems, in which each node adds its own residence time to a running total latency through the system. This format matches the format of the correction field in the packet, as used in transparent clock mode.

- Bits [63:16]: Nanoseconds (48 bits). This field can specify a value greater than 4 seconds.
- Bits [15:0]: Fractions of nanosecond (16 bits). This field is a true fraction; it overflows at 0xFFFF.

The TOD module provides 64-bit TOD information to the IP core in this 64-bit TOD format. The expected format of all 64-bit input timestamp and TOD signals to the IP core is the Altera 64-bit TOD format. The format of all 64-bit output timestamp and TOD signals from the IP core is the Altera 64-bit TOD format. If you build your own TOD module that provides 64-bit TOD information to the IP core, you must ensure it provides TOD information in the Altera 64-bit TOD format.

**Related Information**

**IEEE website**

The *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard* is available on the IEEE website.

## 1588 PTP Interface Signals

### Table 3-11: Signals of the 1588 Precision Time Protocol Interface

Signals are clocked by `clk_rxmac` or `clk_txmac`, as specified. All 64-bit output signals are in the Altera 64-bit TOD format, and you are expected to drive all 64-bit input signals in this format.

| Signal Name | Direction | Description |
|---|---|---|
| **PTP Interface to TOD module** | | |
| `tx_time_of_day_96b_data[95:0]` | Input | Current V2-format (96-bit) TOD in `clk_txmac` clock domain. Connect this signal to the external TOD module.<br>This signal is available only if you turn on the **Enable 96b Time of Day Format** parameter. |
| `tx_time_of_day_64b_data[63:0]` | Input | Current 64-bit TOD in `clk_txmac` clock domain. Connect this signal to the external TOD module.<br>This signal is available only if you turn on the **Enable 64b Time of Day Format** parameter. |
| `rx_time_of_day_96b_data[95:0]` | Input | Current V2-format (96-bit) TOD in `clk_rxmac` clock domain. Connect this signal to the external TOD module.<br>This signal is available only if you turn on the **Enable 96b Time of Day Format** parameter. |
| `rx_time_of_day_64b_data[63:0]` | Input | Current 64-bit TOD in `clk_rxmac` clock domain. Connect this signal to the external TOD module.<br>This signal is available only if you turn on the **Enable 64b Time of Day Format** parameter. |
| **PTP Interface to Client** | | |
| TX Signals Related to One Step Processing | | |

| Signal Name | Direction | Description |
|---|---|---|
| tx_etstamp_ins_ctrl_timestamp_insert | Input | Indicates the current packet on the TX client interface is a 1588 PTP packet, and directs the IP core to process the packet in one-step processing insertion mode. In this mode, the IP core overwrites the timestamp of the packet with the timestamp when the packet appears on the TX Ethernet link.<br>The TX client must assert and deassert this signal synchronously with the TX SOP signal for the 1588 PTP packet.<br>If the TX client asserts this signal simultaneously with either of tx_etstamp_ins_ctrl_residence_time_update or tx_egress_timestamp_request_valid, the results are undefined. |
| tx_etstamp_ins_ctrl_residence_time_update | Input | Indicates the current packet on the TX client interface is a 1588 PTP packet, and directs the IP core to process the packet in one-step processing correction mode. In this mode, the IP core adds the latency through the IP core (residence time) to the current contents of the timestamp field.<br>The TX client must assert and deassert this signal synchronously with the TX SOP signal for the 1588 PTP packet.<br>If the TX client asserts this signal simultaneously with either of tx_etstamp_ins_ctrl_timestamp_insert or tx_egress_timestamp_request_valid, the results are undefined. |
| tx_etstamp_ins_ctrl_ingress_timestamp_96b[95:0] | Input | Iindicates the V2-format TOD when the packet entered the system.<br>The TX client must ensure this signal is valid in each TX SOP cycle when it asserts tx_etstamp_ins_ctrl_residence_time_update. The TX client must maintain the desired value on this signal while the TX SOP signal is asserted. This signal is useful only in transparent clock mode when the TX client asserts tx_etstamp_ins_ctrl_residence_time_update.<br>This signal is available only if you turn on the **Enable 96b Time of Day Format** parameter. |

| Signal Name | Direction | Description |
|---|---|---|
| tx_etstamp_ins_ctrl_ingress_timestamp_64b[63:0] | Input | Indicates the TOD (in Altera 64-bit format) when the packet entered the system.<br><br>The TX client must ensure this signal is valid in each TX SOP cycle when it asserts tx_etstamp_ins_ctrl_residence_time_update. The TX client must maintain the desired value on this signal while the TX SOP signal is asserted. This signal is useful only in transparent clock mode when the TX client asserts tx_etstamp_ins_ctrl_residence_time_update.<br><br>This signal is available only if you turn on the **Enable 64b Time of Day Format** parameter. |
| tx_etstamp_ins_ctrl_timestamp_format | Input | Specifies the timestamp format (V1 or V2 format) for the current packet if the TX client simultaneously asserts tx_etstamp_ins_ctrl_timestamp_insert. Values are:<br><br>• 1'b0: 96-bit timestamp format (V2)<br>• 1'b1: 64-bit timestamp format (V1)<br><br>The TX client must maintain the desired value on this signal while the TX SOP signal is asserted.<br><br>If the client specifies the V1 format, you read and write the V1 format TOD (32 bits of seconds and 32 bits of nanoseconds) in bits [79:16] of the 96-bit timestamp and TOD signals.<br><br>**Note:** If you do not turn on the **Enable 96b Time of Day Format** parameter, the results of asserting tx_etstamp_ins_ctrl_timestamp_insert are undefined. Therefore, the timestamp in either case maps to the 96-bit signals. |

| Signal Name | Direction | Description |
|---|---|---|
| tx_etstamp_ins_ctrl_residence_time_calc_format | Input | Specifies the TOD format (Altera 64-bit TOD format or the V2 96-bit format) for the current packet if the TX client simultaneously asserts tx_etstamp_ins_ctrl_residence_time_update. Values are:<br>• 1'b0: 96-bit TOD format (V2)<br>• 1'b1: 64-bit TOD format<br>The TX client must maintain the desired value on this signal while the TX SOP signal is asserted.<br>If you turned on **Enable 96b Time of Day Format** and the client specifies the 64-bit format, the IP core maps the 64-bit TOD format time-of-day (32 bits of seconds and 32 bits of nanoseconds) as is in bits [79:16] of the 96-bit timestamp and TOD signals.<br>If you turned off **Enable 96b Time of Day Format** and the client specifies the 96-bit format (V2), the results are undefined. |
| tx_etstamp_ins_ctrl_offset_timestamp[15:0] | Input | Specifies the byte offset of the timestamp information in the current packet if the TX client simultaneously asserts tx_etstamp_ins_ctrl_timestamp_insert. The IP core overwrites the value at this offset. The TX client must maintain the desired value on this signal while the TX SOP signal is asserted.<br>If the packet supports V2 format, the timestamp has 96 bits. In this case, the IP core inserts ten bytes (bits [95:16]) of the timestamp at this offset and the remaining two bytes (bits [15:0]) of the timestamp at the offset specified in tx_etstamp_ins_ctrl_offset_correction_field.<br>The offset must support inclusion of the entire timestamp in the first 256 bytes of the packet.<br>The TX client must ensure that the timestamp bytes and the UDP checksum bytes in the packet do not overlap. The IP core does not check for overlap. If these two fields overlap, the results are undefined. |

| Signal Name | Direction | Description |
|---|---|---|
| tx_etstamp_ins_ctrl_offset_correction_field[15:0] | Input | If the TX client simultaneously asserts tx_etstamp_ins_ctrl_residence_time_update, this signal specifies the byte offset of the correction field in the the current packet.<br>If the TX client simultaneously asserts tx_etstamp_ins_ctrl_timestamp_insert and deasserts (sets to the value of 0) the tx_etstamp_ins_ctrl_timestamp_format signal, this signal specifies the byte offset of bits [15:0]] of the timestamp.<br>The TX client must maintain the desired value on this signal while the TX SOP signal is asserted. The offset must support inclusion of the entire timestamp in the first 256 bytes of the packet.<br>The TX client must ensure that the correction field bytes and the UDP checksum bytes in the packet do not overlap. The IP core does not check for overlap. If these two fields overlap, the results are undefined. |
| tx_etstamp_ins_ctrl_checksum_zero | Input | The TX client asserts this signal during a TX SOP cycle to tell the IP core to zero the UDP checksum in the current packet.<br>If the TX client asserts the tx_etstamp_ins_ctrl_checksum_correct signal, it cannot assert this signal. This signal is meaningful only in two-step clock mode.<br>A zeroed UDP checksum indicates the checksum value is not necessarily correct. This information is useful to tell the application to skip checksum checking of UDP IPv4 packets. This function is illegal for UDP IPv6 packets. |
| tx_etstamp_ins_ctrl_offset_checksum_field[15:0] | Input | Indicates the byte offset of the UDP checksum in the current packet. The TX client must ensure this signal has a valid value during each TX SOP cycle when it also asserts the tx_etstamp_ins_ctrl_checksum_zero signal. Holds the byte offset of the two bytes in the packet that the IP core should reset. This signal is meaningful only in two-step clock mode.<br>The TX client must ensure that the UDP checksum bytes and the timestamp bytes in the packet do not overlap. The IP core does not check for overlap. If these two fields overlap, the results are undefined. |

| Signal Name | Direction | Description |
|---|---|---|
| tx_etstamp_ins_ctrl_checksum_correct | Input | The TX client asserts this signal during a TX SOP cycle to tell the IP core to update (correct) the UDP checksum in the current packet.<br>If the TX client asserts the tx_etstamp_ins_ctrl_checksum_zero signal, it cannot assert this signal. This signal is meaningful only in two-step clock mode.<br>The application must assert this signal for correct processing of UDP IPv6 packets. |
| tx_etstamp_ins_ctrl_offset_checksum_correction[15:0] | Input | Indicates the byte offset of the UDP checksum in the current packet. The TX client must ensure this signal has a valid value during each TX SOP cycle when it also asserts the tx_etstamp_ins_ctrl_checksum_correct signal. Holds the byte offset of the two bytes in the packet that the IP core should correct. This signal is meaningful only in two-step clock mode.<br>The TX client must ensure that the UDP checksum bytes and the timestamp bytes in the packet do not overlap. The IP core does not check for overlap. If these two fields overlap, the results are undefined. |
| tx_egress_asymmetry_update | Input | Indicates the IP core should include the value in the TX_PTP_ASYM_DELAY register in its correction calculations. The TX client must maintain the desired value on this signal while the TX SOP signal is asserted.<br>This option is useful in two-step correction mode. |

TX Signals Related to Two Step Processing

| Signal Name | Direction | Description |
|---|---|---|
| tx_egress_timestamp_request_valid | Input | Indicates the current packet on the TX client interface is a 1588 PTP packet, and directs the IP core to process the packet in two-step processing mode. In this mode, the IP core outputs the timestamp of the packet when it exits the IP core, and does not modify the packet timestamp information.<br>The TX client must assert and deassert this signal synchronously with the TX SOP signal for the 1588 PTP packet.<br>If the TX client asserts this signal simultaneously with either of tx_etstamp_ins_ctrl_timestamp_insert or tx_etstamp_ins_ctrl_residence_time_update, the results are undefined. |
| tx_egress_timestamp_96b_data[95:0] | Output | Provides the V2-format timestamp when a 1588 PTP frame begins transmission on the Ethernet link. Value is valid when the tx_egress_timestamp_96b_valid signal is asserted. This signal is meaningful only in two-step clock mode.<br>This signal is available only if you turn on the **Enable 96b Time of Day Format** parameter. |
| tx_egress_timestamp_96b_valid | Output | Indicates that the tx_egress_timestamp_96b_data and tx_egress_timestamp_96b_fingerprint signals are valid in the current clk_txmac clock cycle. This signal is meaningful only in two-step clock mode.<br>This signal is available only if you turn on the **Enable 96b Time of Day Format** parameter. |
| tx_egress_timestamp_64b_data[63:0] | Output | Provides the timestamp when a V1-format 1588 PTP frame begins transmission on the Ethernet link. Value is valid when the tx_egress_timestamp_64b_valid signal is asserted. This signal is meaningful only in two-step clock mode.<br>This signal is available only if you turn on the **Enable 64b Time of Day Format** parameter. |

| Signal Name | Direction | Description |
|---|---|---|
| tx_egress_timestamp_64b_valid | Output | Indicates that the tx_egress_timestamp_64b_data and tx_egress_timestamp_64b_fingerprint signals are valid in the current clk_txmac clock cycle. This signal is meaningful only in two-step clock mode.<br>This signal is available only if you turn on the **Enable 64b Time of Day Format** parameter. |
| tx_egress_timestamp_request_fingerprint[(W–1):0]<br>where W is the value between 1 and 16, inclusive, that you specify for the **Timestamp fingerprint width** parameter | Input | Fingerprint of the current packet.<br>The TX client must assert and deassert this signal synchronously with the TX SOP signal for the 1588 PTP packet. |
| tx_egress_timestamp_96b_fingerprint[(W–1):0]<br>where W is the value between 1 and 16, inclusive, that you specify for the **Timestamp fingerprint width** parameter | Output | Provides the fingerprint of the V2-format 1588 PTP frame currently beginning transmission on the Ethernet link. Value is valid when the tx_egress_timestamp_96b_valid signal is asserted.<br>This signal is available only if you turn on the **Enable 96b Time of Day Format** parameter. |
| tx_egress_timestamp_64b_fingerprint[(W–1):0]<br>where W is the value between 1 and 16, inclusive, that you specify for the **Timestamp fingerprint width** parameter | Output | Provides the fingerprint of the V1-format 1588 PTP frame currently beginning transmission on the Ethernet link. Value is valid when the tx_egress_timestamp_64b_valid signal is asserted.<br>This signal is available only if you turn on the **Enable 64b Time of Day Format** parameter. |
| RX Signals | | |
| rx_ingress_timestamp_96b_data[95:0] | Output | Whether or not the current packet on the RX client interface is a 1588 PTP packet, indicates the V2-format timestamp when the IP core received the packet on the Ethernet link. The IP core provides a valid value on this signal in the same cycle it asserts the RX SOP signal for 1588 PTP packets.<br>This signal is available only if you turn on the **Enable 96b Time of Day Format** parameter. |
| rx_ingress_timestamp_96b_valid | Output | Indicates that the rx_ingress_timestamp_96b_data signal is valid in the current cycle. This signal is redundant with the RX SOP signal for 1588 PTP packets.<br>This signal is available only if you turn on the **Enable 96b Time of Day Format** parameter. |

| Signal Name | Direction | Description |
|---|---|---|
| rx_ingress_timestamp_64b_data[63:0] | Output | Whether or not the current packet on the RX client interface is a 1588 PTP packet, indicates the 64-bit TOD (in Altera 64-bit format) when the IP core received the packet on the Ethernet link. The IP core provides a valid value on this signal in the same cycle it asserts the RX SOP signal for 1588 PTP packets.<br>This signal is available only if you turn on the **Enable 64b Time of Day Format** parameter. |
| rx_ingress_timestamp_64b_valid | Output | Indicates that the rx_ingress_timestamp_64b_data signal is valid in the current cycle. This signal is redundant with the RX SOP signal for 1588 PTP packets.<br>This signal is available only if you turn on the **Enable 64b Time of Day Format** parameter. |

## PHY Status Interface

The rx_pcs_ready output signal is available to provide status information to user logic. This signal is asserted when the RX lanes are fully aligned and ready to receive data.

The tx_lanes_stable output signal is available to provide status information to user logic. This signal is asserted when the TX lanes are fully aligned and ready to transmit data.

## Transceiver PHY Serial Data Interface

The core uses an <*n*> lane digital interface to send data to the TX high-speed serial I/O pins operating at 10.3125 Gbps in the standard 40GbE and 100GbE variations and at 25.78125 Gbps in the CAUI–4 variations. The rx_serial and tx_serial ports connect to the 10.3125 Gbps or 25.78125 Gbps pins. Virtual lanes 0 and 1 transmit data on tx_serial[0].

## Low Latency 40GBASE-KR4 IP Core Variations

The LL 40GBASE-KR4 IP core supports low-level control of analog transceiver properties for link training and auto-negotiation in the absence of a predetermined environment for the IP core. For example, an Ethernet IP core in a backplane may have to communicate with different link partners at different times. When it powers up, the environment parameters may be different than when it ran previously. The environment can also change dynamically, necessitating reset and renegotiation of the Ethernet link.

The LL 40-100GbE IP core 40GBASE-KR4 variations implement the *IEEE Backplane Ethernet Standard 802.3ap-2007*. The LL 40-100GbE IP core provides this reconfiguration functionality in Arria 10 devices by configuring each physical Ethernet lane with an Altera Backplane Ethernet 10GBASE-KR PHY IP core if you turn on **Enable KR4** in the 40-100GbE parameter editor. The parameter is available in variations parameterized with these values:

- Device family: Arria 10
- Protocol speed: 40GbE
- **Enable 1588 PTP**: Off

The IP core includes the option to implement the following features:

- KR auto-negotiation provides a process to explore coordination with a link partner on a variety of different common features. The 40GBASE-KR4 variations of the LL 40-100GbE IP core can auto-negotiate only to a 40GBASE-KR4 configuration. Turn on the **Enable KR4 Reconfiguration** and **Enable Auto-Negotiation** parameters to configure support for auto-negotiation.
- Link training provides a process for the IP core to train the link to the data frequency of incoming data, while compensating for variations in process, voltage, and temperature. Turn on the **Enable KR4 Reconfiguration** and **Enable Link Training** parameters to configure support for link training.
- After the link is up and running, forward error correction (FEC) provides an error detection and correction mechanism to enable noisy channels to achieve the Ethernet-mandated bit error rate (BER) of $10^{-12}$. Turn on the **Include FEC sublayer** parameter to configure support for FEC.

The LL 40GBASE-KR4 IP core variations include separate link training and FEC modules for each of the four Ethernet lanes, and a single auto-negotiation module. You specify the master lane for performing auto-negotiation in the parameter editor, and the IP core also provides register support to modify the selection dynamically.

Altera provides a testbench for LL 40GBASE-KR4 IP core variations with an Avalon-ST client interface that generate their own TX MAC clock (**Use external TX MAC PLL** is turned off). Altera provides an example project for all LL 40GBASE-KR4 IP core variations that generate their own TX MAC clock, to assist you in integrating your LL 40GBASE-KR4 IP core into your complete design. You can examine the testbench and example project for an example of how to drive and connect the 40GBASE-KR4 IP core.

IP core FEC functionality relies on register settings in the LL 40GBASE-KR4 registers and on some specific register fields in the Arria 10 device registers.

To simulate correctly and to run correctly in hardware, you must drive the `reconfig_clk` and the `clk_status` inputs from the same source clock.

**Related Information**

**Arria 10 Transceiver PHY User Guide**

The 40GBASE-KR4 variations of the LL 40-100GbE IP core use the Altera Arria 10 10GBASE-KR PHY IP core. Information about this PHY IP core, including functional descriptions of the listed features, is available in the *10GBASE-KR PHY IP Core* section of the *Arria 10 Transceiver PHY User Guide*. In this section, functional descriptions of the AN and LT features are available in the "Auto Negotiation, IEEE 802.3 Clause 73" and "Link Training (LT), IEEE 802.3 Clause 72" sections.

## Control and Status Interface

The control and status interface provides an Avalon-MM interface to the IP core control and status registers. The Avalon-MM interface implements a standard memory-mapped protocol. You can connect an embedded processor or JTAG Avalon master to this bus to access the control and status registers.

### Table 3-12: Avalon-MM Control and Status Interface Signals

The `clk_status` clocks the signals on the LL 40-100GbE IP core control and status interface. The synchronous `reset_status` reset signal resets the interface.

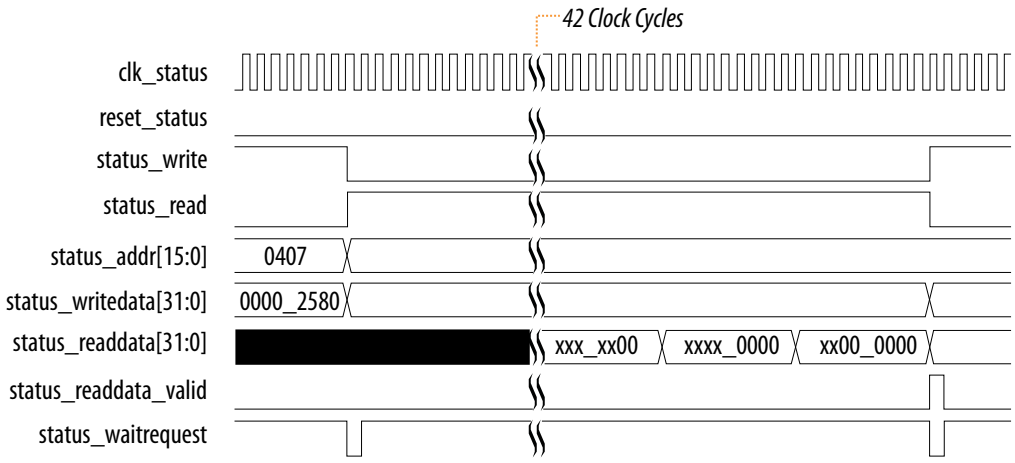
| Signal Name | Direction | Description |
|---|---|---|
| `status_addr [15:0]` | Input | Address for reads and writes |
| `status_read` | Input | Read command |

| Signal Name | Direction | Description |
|---|---|---|
| status_write | Input | Write command |
| status_writedata [31:0] | Input | Data to be written |
| status_readdata [31:0] | Output | Read data |
| status_readdata_valid | Output | Read data is ready for use |
| status_waitrequest | Output | Busy signal indicating control and status interface cannot currently respond to requests |
| status_read_timeout | Output | Timeout signal indicating read data did not arrive when expected. Hardwired timeout counter is set so that this timeout should only occur in the presence of an error condition, such as status_addr with an undefined value. This signal is not an Avalon-MM defined signal |

The status interface is designed to operate at a low frequencies, typically 100 MHz, so that control and status logic does not compete for resources with the surrounding high speed datapath.

**Figure 3-26: Read and Write Register Access Example**

The waveform demonstrates a write-read-write sequence of back-to-back register accesses. The delay from the time the application asserts status_read until the IP core asserts status_readdata_valid and deasserts status_waitrequest is approximately 80 clk_status cycles in this example.




**Related Information**

- **Software Interface: Registers** on page 3-72
- **Low Latency 40-100GbE IP Core Registers** on page 3-76
- **Avalon Interface Specifications**
  For more information about the Avalon-MM protocol, including timing diagrams, refer to the *Avalon Memory-Mapped Interfaces* chapter.

## Arria 10 Transceiver Reconfiguration Interface

Arria 10 variations provide a dedicated Avalon-MM interface, called the Arria 10 transceiver reconfiguration interface, to access the transceiver registers. You access the Arria 10 Native PHY IP core registers through this dedicated interface and not through the IP core general purpose control and status interface.

The Avalon-MM interface implements a standard memory-mapped protocol. You can connect an embedded processor or JTAG Avalon master to this bus to access the registers of the embedded Arria 10 Native PHY IP core.

**Table 3-13: Avalon-MM Arria 10 Reconfiguration Interface Signals**

The `reconfig_clk` clocks the signals on the Low Latency 40-100GbE IP core Arria 10 transceiver reconfiguration interface. The synchronous `reconfig_reset` reset signal resets the interface.

| Signal Name | Direction | Description |
|---|---|---|
| `reconfig_address [11:0]` (40GbE and CAUI-4)<br>`reconfig_address [13:0]` (100GbE) | Input | Address for reads and writes |
| `reconfig_read` | Input | Read command |
| `reconfig_write` | Input | Write command |
| `reconfig_writedata [31:0]` | Input | Data to be written |
| `reconfig_readdata [31:0]` | Output | Read data |
| `reconfig_waitrequest` | Output | Interface busy signal |

The Arria 10 reconfiguration interface is designed to operate at a low frequency, 100 MHz to 125 MHz, so that the user's Arria 10 transceiver reconfiguration logic does not compete for resources with the surrounding high speed datapath.

**Related Information**

- **Avalon Interface Specifications**
  For more information about the Avalon-MM protocol, including timing diagrams, refer to the *Avalon Memory-Mapped Interfaces* chapter.
- **Arria 10 Transceiver PHY User Guide**
  Information about the Arria 10 Native PHY IP core hard PCS registers that you can program through the Arria 10 transceiver reconfiguration interface.
- **Arria 10 Transceiver Registers**
  Information about the Arria 10 transceiver registers.

## Clocks

You must set the transceiver reference clock (`clk_ref`) frequency to a value that the IP core supports. The Low Latency 40-100GbE IP core supports `clk_ref` frequencies of 644.53125 MHz ±100 ppm and

322.265625 MHz ± 100 ppm. The ±100ppm value is required for any clock source providing the transceiver reference clock.

Sync–E IP core variations are IP core variations for which you turn on **Enable SyncE** in the parameter editor. These variations provide the RX recovered clock as a top-level output signal. This option is available only for IP core variations that target an Arria 10 device.

The Synchronous Ethernet standard, described in the ITU-T G.8261, G.8262, and G.8264 recommendations, requires that the TX clock be filtered to maintain synchronization with the RX reference clock through a sequence of nodes. The expected usage is that user logic drives the TX PLL reference clock with a filtered version of the RX recovered clock signal, to ensure the receive and transmit functions remain synchronized. In this usage model, a design component outside the LL 40-100GbE IP core performs the filtering.

**Table 3-14: Clock Inputs**

Describes the input clocks that you must provide.

| Signal Name | Description |
|---|---|
| `clk_status` | Clocks the control and status interface. The clock quality and pin chosen are not critical. `clk_status` is expected to be a 100–125 MHz clock.<br>In LL 40GBASE-KR4 variations, you must drive `clk_status` and `reconfig_status` from a single clock source. |
| `reconfig_clk` | Clocks the Arria 10 transceiver reconfiguration interface. The clock quality and pin chosen are not critical. `reconfig_clk` is expected to be a 100 MHz clock; the allowed frequency range depends on Arria 10 transceiver requirements and is not IP core specific.<br>In LL 40GBASE-KR4 variations, you must drive `clk_status` and `reconfig_clk` from a single clock source. |
| `clk_ref` | In IP core variations that target an Arria 10 device, `clk_ref` is the reference clock for the transceiver RX CDR PLL. In other IP core variations, `clk_ref` is the reference clock for the transceiver TX PLL and the RX CDR PLL.<br>The frequency of this input clock must match the value you specify for **PHY reference frequency** in the IP core parameter editor, with a ±100 ppm accuracy per the *IEEE 802.3ba-2010 100G Ethernet Standard.*.<br>In addition, `clk_ref` must meet the jitter specification of the *IEEE 802.3ba-2010 100G Ethernet Standard*.<br>The PLL and clock generation logic use this reference clock to derive the transceiver and PCS clocks. The input clock should be a high quality signal on the appropriate dedicated clock pin. Refer to the relevant device datasheet for transceiver reference clock phase noise specifications. |

| Signal Name | Description |
|---|---|
| clk_txmac_in | If you turn on **Use external TX MAC PLL** in the LL 40-100GbE parameter editor, this clock drives the TX MAC. The port is expected to receive the clock from the external TX MAC PLL and drives the internal clock clk_txmac. The required TX MAC clock frequency is 312.5 MHz for 40GbE variations, and 390.625 MHz for 100GbE variations. User logic must drive clk_txmac_in from a PLL whose input is the PHY reference clock, clk_ref. |
| tx_serial_clk[3:0] (for 40GbE and CAUI-4 variations that target an Arria 10 device)<br>tx_serial_clk[9:0] (for standard 100GbE variations that target an Arria 10 device) | These input clocks are present only in variations that target an Arria 10 device. They are part of the external PLL interface to these variations. Each clock targets a single transceiver PHY link. You must drive these clocks from one or more TX transceiver PLLs that you configure separately from the Low Latency 40-100GbE IP core. |

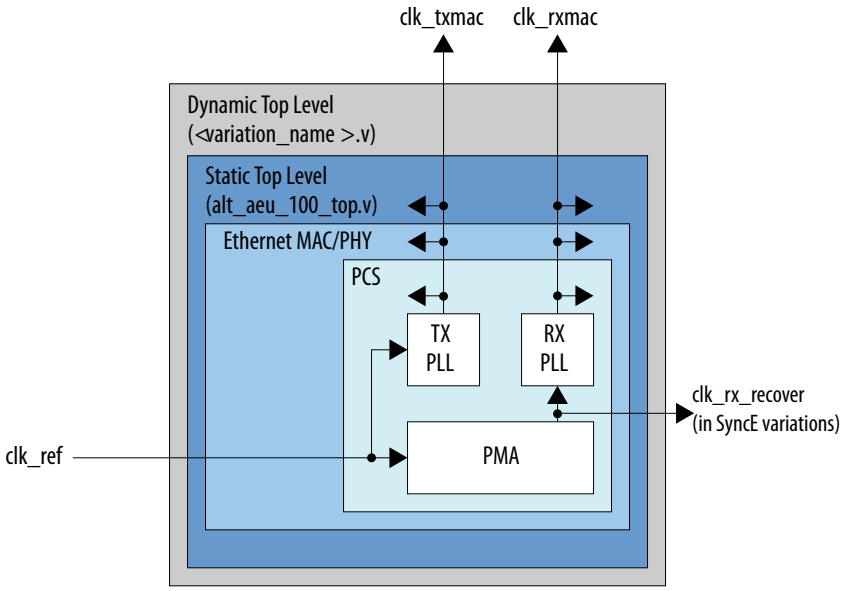
**Table 3-15: Clock Outputs**

Describes the output clocks that the IP core provides. In most cases these clocks participate in internal clocking of the IP core as well.

| Signal Name | Description |
|---|---|
| clk_txmac | The TX clock for the IP core is clk_txmac. The TX MAC clock frequency is 312.5 MHz for 40GbE IP core variations and 390.625 MHz for 100GbE IP core variations.<br>If you turn on **Use external TX MAC PLL** in the LL 40-100GbE parameter editor, the clk_txmac_in input clock drives clk_txmac. |
| clk_rxmac | The RX clock for the IP core is clk_rxmac. The RX MAC clock frequency is 312.5 MHz for 40GbE IP core variations and 390.625 MHz for 100GbE IP core variations.<br>This clock is only reliable when rx_pcs_ready has the value of 1. The IP core generates clk_rxmac from a recovered clock that relies on the presence of incoming RX data. |
| clk_rx_recover | RX recovered clock. This clock is available only if you turn on **Enable SyncE** in the LL 40-100GbE parameter editor.<br>The RX recovered clock frequency is 257.81 MHz in 40GbE IP core variations and 322.265625 MHz in 100GbE IP core variations.<br>The expected usage is that you drive the TX transceiver PLL reference clock with a filtered version of clk_rx_recover, to ensure the receive and transmit functions remain synchronized in your Synchronous Ethernet system. To do so you must instantiate an additional component in your design. The IP core does not provide filtering. |

**Figure 3-27: Clock Generation Circuitry**

Provides a high-level view of the clock generation circuitry and clock distribution to the transceiver. In Arria 10 variations, the TX transceiver PLL is configured outside the 40-100GbE IP core, and `clk_ref` drives the RX CDR PLL. In Arria 10 variations, you can connect a separate reference clock for the external TX transceiver PLL.




**Related Information**

- **Transceiver PLL Required in Arria 10 Designs** on page 2-20
  Information about configuring and connecting the external PLLs. Includes signal descriptions.
- **Arria 10 Device Datasheet**
  Provides transceiver reference clock phase noise specifications.
- **Stratix V Device Datasheet**
  Provides transceiver reference clock phase noise specifications.
- **Arria V Device Datasheet**
  Provides Arria V GZ transceiver reference clock phase noise specifications.

## Resets

The Low Latency 40-100GbE IP core has a single asynchronous reset signal. Asserting this signal resets the full IP core. You must hold the reset signal asserted for ten `clk_status` clock cycles to ensure proper hold time.

You should not release the reset signal until after you observe that the reference clock is stable. If the reference clock is generated from an fPLL, wait until after the fPLL locks. Ten `clk_status` cycles should be sufficient for the fPLL to lock and the reference clock to stabilize.

**Table 3-16: Asynchronous Reset Signal**

| Signal Name | Direction | Description |
|---|---|---|
| `reset_async` | Input | Low Latency 40-100GbE IP core asynchronous reset signal |

In addition, the Low Latency 40-100GbE IP core has one or two of the following synchronous reset signals:

- `reset_status`—Resets the IP core control and status interface, an Avalon-MM interface. Associated clock is the `clk_status` clock, which clocks the control and status interface.
- `reconfig_reset`—Resets the IP core Arria 10 transceiver reconfiguration interface, an Avalon-MM interface. Associated clock is the `reconfig_clk`, which clocks the Arria 10 transceiver reconfiguration interface. This signal is available only in Arria 10 IP core variations.

# Signals

This section lists the external signals of the different 40-100GbE IP core variations.

**Low Latency 40-100GbE IP Core Signals** on page 3-62

## Low Latency 40-100GbE IP Core Signals

The signals of the Low Latency 40-100GbE IP core are described in the following formats:

- The figure identifies the IP core interfaces.
- Tables list the signals and the interfaces to which they belong.
- Links guide you to descriptions for the individual signals, by interface. The links are available only if you are viewing this topic in the context of the Functional Description chapter of the *Low Latency 40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide*. The links appear in Related Links, below.

**Figure 3-28: Top-Level Signals of the Low Latency 40-100GbE IP Cores**

In the figure, <*n*> = 4 for the 40GbE IP cores and <*n*> = 8 for the 100GbE IP cores. <*l*> is $\log_2(8^*<n>)$. In the custom streaming client interface, <*w*> = 2 for the 40GbE IP core and <*w*> = 4 for the 100GbE IP core. <*v*> is the number of transceiver PHY links (4 for 40GbE and CAUI-4 IP cores, and 10 for standard 100GbE IP cores). <*N*> is the number of priority flow control queues. (<*N*> = 1 for IP cores configured with standard flow control or with no flow control, and <*N*> = the value of the relevant parameter for IP cores configured with priority-based flow control). In the 1588 PTP signals, <*W*> is the width of the fingerprint.

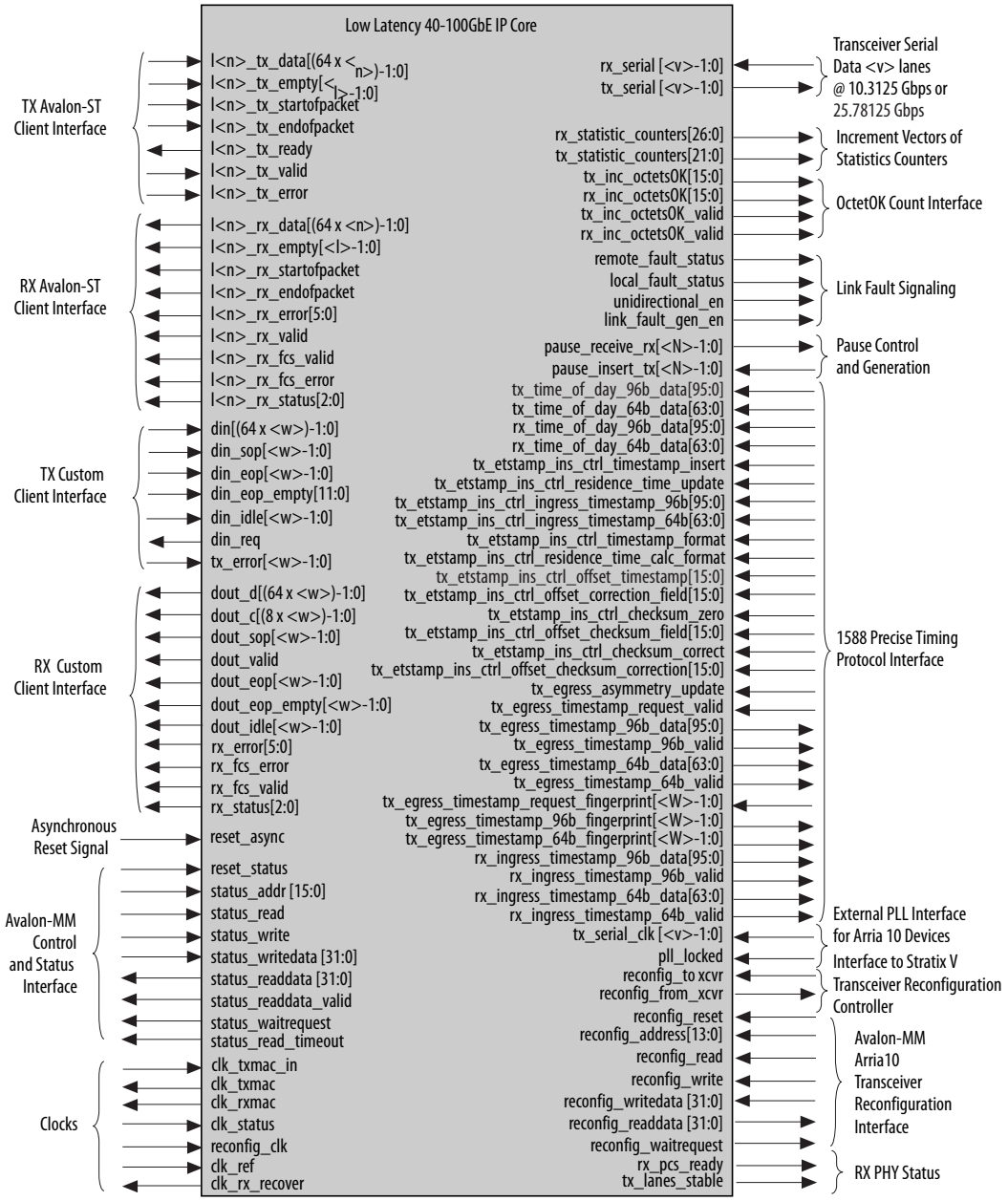


**Table 3-17: Low Latency 40-100GbE MAC and PHY IP Core Signals**

| Signal Name | Direction | Interface |
|---|---|---|
| clk_ref | Input | Clocks |
| clk_rx_recover | Output | Clocks<br>This signal is only available if you turn on **Enable SyncE** in the parameter editor. |
| reset_async | Input | Reset |
| tx_serial[3:0] (40GbE and CAUI–4)<br>tx_serial[9:0] (100GbE) | Output | Transceiver PHY serial data interface |
| rx_pcs_ready | Output | PHY status |
| tx_lanes_stable | Output | |
| clk_txmac_in | Input | Clocks<br>Interface to external TX MAC PLL<br>This signal is only available if you turn on **Enable external TX MAC PLL** in the parameter editor. |
| clk_txmac | Output | Clocks<br>TX client interface |
| l<*n*>_tx_data[<*n*>*64-1:0] | Input | Avalon-ST TX client interface<br>Each IP core instance has Avalon-ST TX and RX client interfaces, or custom streaming TX and RX client interfaces. |
| l<*n*>_tx_empty[<*l*>-1:0] | Input | |
| l<*n*>_tx_startofpacket | Input | |
| l<*n*>_tx_endofpacket | Input | |
| l<*n*>_tx_ready | Output | |
| l<*n*>_tx_valid | Input | |
| l<*n*>_tx_error | Input | |
| din[<w>*64-1:0] | Input | Custom streaming TX client interface |
| din_sop[<w>-1:0] | Input | |

| Signal Name | Direction | Interface |
|---|---|---|
| din_eop[<w>-1:0] | Input | Each IP core instance has Avalon-ST TX and RX client interfaces, or custom streaming TX and RX client interfaces. |
| din_eop_empty[<w>*3-1:0] | Input | |
| din_idle[<w>-1:0] | Input | |
| din_req | Output | |
| tx_error[<w>-1:0] | Input | |
| clk_rxmac | Output | Clocks<br>RX client interface |
| l<n>_rx_data[<n>*64-1:0] | Output | Avalon-ST RX client interface<br>Each IP core instance has Avalon-ST TX and RX client interfaces, or custom streaming TX and RX client interfaces. |
| l<n>_rx_empty[<l>-1:0] | Output | |
| l<n>_rx_startofpacket | Output | |
| l<n>_rx_endofpacket | Output | |
| l<n>_rx_error[5:0] | Output | |
| l<n>_rx_valid | Output | |
| l<n>_rx_fcs_valid | Output | |
| l<n>_rx_fcs_error | Output | |
| l<n>_rx_status[2:0] | Output | |
| dout_d[<w>*64-1:0] | Output | Custom streaming RX client interface<br>Each IP core instance has Avalon-ST TX and RX client interfaces, or custom streaming TX and RX client interfaces. |
| dout_c[<w>*8-1:0] | Output | |
| dout_sop[<w>-1:0] | Output | |
| dout_eop[<w>-1:0] | Output | |
| dout_eop_empty[<w>*3-1:0] | Output | |
| dout_idle[<w>-1:0] | Output | |
| rx_error[5:0] | Output | |
| rx_fcs_error | Output | |
| rx_fcs_valid | Output | |

| Signal Name | Direction | Interface |
|---|---|---|
| rx_status[2:0] | Output | |
| dout_valid | Output | |
| pause_insert_tx[<N>-1:0] | Input | Pause control and generation interface<br>These signals are available in new IP core variations that include a flow control module. In IP core variations that do not include a flow control module, but which you upgraded to the v14.1 IP core, these signals are tied low. |
| pause_receive_rx[<N>-1:0] | Output | |
| remote_fault_status | Output | Link fault signaling interface<br>These signals are available only in IP core variations that include the link fault signaling module. |
| local_fault_status | Output | |
| unidirectional_en | Output | Link fault signaling interface, Clause 66 status<br>These signals are available only in IP core variations that include the link fault signaling module. |
| link_fault_gen_en | Output | |
| tx_inc_64 | Output | TX statistics counter increment vectors<br>These signals are available whether or not your IP core includes TX statistics counters. |
| tx_inc_127 | Output | |
| tx_inc_255 | Output | |
| tx_inc_511 | Output | |
| tx_inc_1023 | Output | |
| tx_inc_1518 | Output | |
| tx_inc_max | Output | |
| tx_inc_over | Output | |
| tx_inc_mcast_data_err | Output | |
| tx_inc_mcast_data_ok | Output | |
| tx_inc_bcast_data_err | Output | |

| Signal Name | Direction | Interface |
|---|---|---|
| tx_inc_bcast_data_ok | Output | |
| tx_inc_ucast_data_err | Output | |
| tx_inc_ucast_data_ok | Output | |
| tx_inc_mcast_ctrl | Output | |
| tx_inc_bcast_ctrl | Output | |
| tx_inc_ucast_ctrl | Output | |
| tx_inc_pause | Output | |
| tx_inc_fcs_err | Output | |
| tx_inc_fragment | Output | |
| tx_inc_jabber | Output | |
| tx_inc_sizeok_fcserr | Output | |
| rx_inc_runt | Output | RX statistics counter increment vectors<br>These signals are available whether or not your IP core includes RX statistics counters. |
| rx_inc_64 | Output | |
| rx_inc_127 | Output | |
| rx_inc_255 | Output | |
| rx_inc_511 | Output | |
| rx_inc_1023 | Output | |
| rx_inc_1518 | Output | |
| rx_inc_max | Output | |
| rx_inc_over | Output | |
| rx_inc_mcast_data_err | Output | |
| rx_inc_mcast_data_ok | Output | |
| rx_inc_bcast_data_err | Output | |
| rx_inc_bcast_data_ok | Output | |

| Signal Name | Direction | Interface |
|---|---|---|
| rx_inc_ucast_data_err | Output | |
| rx_inc_ucast_data_ok | Output | |
| rx_inc_mcast_ctrl | Output | |
| rx_inc_bcast_ctrl | Output | |
| rx_inc_ucast_ctrl | Output | |
| rx_inc_pause | Output | |
| rx_inc_fcs_err | Output | |
| rx_inc_fragment | Output | |
| rx_inc_jabber | Output | |
| rx_inc_sizeok_fcserr | Output | |
| rx_inc_pause_ctrl_err | Output | |
| rx_inc_mcast_ctrl_err | Output | |
| rx_inc_bcast_ctrl_err | Output | |
| rx_inc_ucast_ctrl_err | Output | |
| tx_inc_octetsOK[15:0] | Output | OctetOK Count interface |
| tx_inc_octetsOK_valid | Output | |
| rx_inc_octetsOK[15:0] | Output | |
| rx_inc_octetsOK_valid | Output | |
| clk_status | Input | Clocks<br>Control and status interface |
| reset_status | Input | Resets<br>Control and status interface |

| Signal Name | Direction | Interface |
|---|---|---|
| status_addr[15:0] | Input | Control and status interface |
| status_read | Input | |
| status_write | Input | |
| status_writedata[31:0] | Input | |
| status_readdata[31:0] | Output | |
| status_readdata_valid | Output | |
| status_waitrequest | Output | |
| status_read_timeout | Output | |
| tx_time_of_day_96b_data[95:0] | Input | 1588 PTP interface<br>These signals are available only if 1588 PTP functionality is included in the IP core. |
| tx_time_of_day_64b_data[63:0] | Input | |
| rx_time_of_day_96b_data[95:0] | Input | |
| rx_time_of_day_64b_data[63:0] | Input | |
| tx_etstamp_ins_ctrl_timestamp_insert | Input | |
| tx_etstamp_ins_ctrl_residence_time_update | Input | |
| tx_etstamp_ins_ctrl_ingress_timestamp_96b[95:0] | Input | |
| tx_etstamp_ins_ctrl_ingress_timestamp_64b[63:0] | Input | |
| tx_etstamp_ins_ctrl_timestamp_format | Input | |
| tx_etstamp_ins_ctrl_residence_time_calc_format | Input | |
| tx_etstamp_ins_ctrl_offset_timestamp[15:0] | Input | |
| tx_etstamp_ins_ctrl_offset_correction_field[15:0] | Input | |
| tx_etstamp_ins_ctrl_checksum_zero | Input | |

| Signal Name | Direction | Interface |
|---|---|---|
| tx_etstamp_ins_ctrl_offset_checksum_field[15:0] | Input | |
| tx_etstamp_ins_ctrl_checksum_correct | Input | |
| tx_etstamp_ins_ctrl_offset_checksum_correction[15:0] | Input | |
| tx_egress_asymmetry_update | Input | |
| tx_egress_timestamp_request_valid | Input | |
| tx_egress_timestamp_96b_data[95:0] | Output | |
| tx_egress_timestamp_96b_valid | Output | |
| tx_egress_timestamp_64b_data[63:0] | Output | |
| tx_egress_timestamp_64b_valid | Output | |
| tx_egress_timestamp_request_fingerprint[(W-1):0] | Input | |
| tx_egress_timestamp_96b_fingerprint[(W-1):0] | Output | |
| tx_egress_timestamp_64b_fingerprint[(W-1):0] | Output | |
| rx_ingress_timestamp_96b_data[95:0] | Output | |
| rx_ingress_timestamp_96b_valid | Output | |
| rx_ingress_timestamp_64b_data[63:0] | Output | |
| rx_ingress_timestamp_64b_valid | Output | |

| | | |
|---|---|---|
| reconfig_to_xcvr[559:0] (40GbE)<br>reconfig_to_xcvr[1399:0] (100GbE) | Input | Interface to Stratix V reconfiguration controller<br>These signals are available in Stratix V devices only. |
| reconfig_from_xcvr[367:0] (40GbE)<br>reconfig_from_xcvr[919:0] (100GbE) | Output | |
| reconfig_busy | Input | |
| reconfig_clk | Input | Clocks<br>Arria 10 Native PHY IP core reconfiguration interface<br>This signal is available in Arria 10 devices only. |
| reconfig_reset | Input | Resets<br>Arria 10 Native PHY IP core reconfiguration interface<br>This signal is available in Arria 10 devices only. |
| reconfig_address [11:0] (40GbE and CAUI-4)<br>reconfig_address [13:0] (100GbE) | Input | Arria 10 Native PHY IP core reconfiguration interface<br>These signals are available in Arria 10 devices only. |
| reconfig_read | Input | |
| reconfig_write | Input | |
| reconfig_writedata[31:0] | Input | |
| reconfig_readdata[31:0] | Output | |
| reconfig_waitrequest | Output | |
| pll_locked | Input | External transceiver PLL interface. These signals are available in Arria 10 devices only. |
| tx_serial_clk[3:0] (40GbE and CAUI–4)<br>tx_serial_clk[9:0] (100GbE) | Input | |

**Related Information**

- **Clocks** on page 3-58
  Overview of IP core clocks. Includes list of clock signals and recommended and required frequencies.
- **Resets** on page 3-61
- **Transceiver PHY Serial Data Interface** on page 3-55
- **PHY Status Interface** on page 3-55
- **Low Latency 40-100GbE IP Core TX Data Bus with Adapters (Avalon-ST Interface)** on page 3-7
  Describes the TX client interface.
- **Low Latency 40-100GbE IP Core TX Data Bus Without Adapters (Custom Streaming Interface)** on page 3-10
- **Low Latency 40-100GbE IP Core RX Data Bus** on page 3-21
  Describes the RX client interface.
- **Low Latency 40-100GbE IP Core RX Data Bus Without Adapters (Custom Streaming Interface)** on page 3-25
- **Pause Control and Generation Interface** on page 3-32
  Describes the pause signals available in the Low Latency 40-100GbE IP core.
- **Link Fault Signaling Interface** on page 3-33
- **Statistics Counters Interface** on page 3-35
- **Control and Status Interface** on page 3-56
- **1588 Precision Time Protocol Interfaces** on page 3-39
- **External Reconfiguration Controller** on page 3-28
- **Arria 10 Transceiver Reconfiguration Interface** on page 3-58
- **External Transceiver PLL** on page 3-28
- **Low Latency 40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide**
  If you read this topic in the full user guide, links guide you to the descriptions of the individual signals, by interface. Otherwise, refer to the user guide to find information about the signals.

# Software Interface: Registers

This section provides information about the memory-mapped registers. You access these registers using the IP core control and status interface. The registers use 32-bit addresses; they are not byte addressable.

Write operations to a read-only register field have no effect. Read operations that address a Reserved register return an unspecified constant. Write operations to Reserved registers have no effect. Accesses to registers that do not exist in your IP core variation have an unspecified result.

The following tables list the memory mapped registers for the Low Latency 40-100GbE IP core.

### Table 3-18: Low Latency 40-100GbE IP Core Register Map Overview

Lists the main ranges of the memory mapped registers for the Low Latency 40-100GbE IP core. Addresses are word addresses. Register address range 0x000–0x2FF is Reserved.

| Word Offset | Register Category |
|---|---|
| 0x0B0–0x0FF | 40GBASE-KR4 registers |
| 0x300–0x3FF | PHY registers |
| 0x400–0x4FF | TX MAC registers |

| Word Offset | Register Category |
|---|---|
| 0x500–0x5FF | RX MAC registers |
| 0x600–0x6FF | TX flow control (pause functionality) registers ((If modify to byte addresses, this will become 0x1800–0x1BFC; in fact, the final defined register in this range will now be |
| 0x700–0x7FF | RX flow control (pause functionality) registers |
| 0x800–0x8FF | TX statistics counters |
| 0x900–0x9FF | RX statistics counters |
| 0xA00–0xAFF | TX 1588 PTP registers |
| 0xB00–0xBFF | RX 1588 PTP registers |

**Table 3-19: Low Latency 40-100GbE IP Core Address Map**

Lists the memory mapped registers for the Low Latency 40-100GbE IP core. Each register is 32 bits, and the addresses (word offsets) each address a full word. This table does not list the Revision ID, scratch, and name registers present in each of the address ranges at offsets 0x00, 0x01, and 0x02–0x04 respectively. However, those registers appear in the detailed listings with descriptions.

| Word Offset | Register Description |
|---|---|
| 0x0B0–0x0BD | 40GBASE-KR4 top-level and FEC registers<br>Accessible only if you turn on **Enable KR**. FEC registers are accessible only if you also turn on **Include FEC sublayer**. |
| 0x0BE–0x0BF | Reserved. |
| 0x0C0–0x0CC | 40GBASE-KR4 auto-negotiation registers<br>Accessible only if you turn on **Enable KR** and **Enable Auto-Negotiation**. |
| 0x0CD–0x0CF | Reserved. |
| 0x0D0–0x0EB | 40GBASE-KR4 link training registers<br>Accessible only if you turn on **Enable KR** and **Enable Link Training**. |
| 0x0EC–0x0FF | Reserved. |
| 0x310–0x344 | PHY registers available in all IP core variations |
| 0x405 | Link fault signaling register `LINK_FAULT_CONFIG`<br>Accessible only if you turn on **Enable link fault generation** |

| Word Offset | Register Description |
|---|---|
| 0x406 | IPG column removal register `IPG_COL_REM`<br>Accessible only if you turn on **Average interpacket gap** |
| 0x407 | TX maximum size Ethernet frame (in bytes) `MAX_TX_SIZE_CONFIG`. Value determines whether the IP core increments the `CNTR_TX_OVERSIZE` register |
| 0x506 | RX maximum size Ethernet frame (in bytes) `MAX_RX_SIZE_CONFIG`. Value determines whether the IP core increments the `CNTR_RX_OVERSIZE` register. |
| 0x507 | RX CRC forwarding configuration register `MAC_CRC_CONFIG` |
| 0x508 | Link fault status register<br>Provides link fault status information if you turn on **Enable link fault generation**. Returns zeroes if you turn off **Enable link fault generation**. |
| 0x50A | Enable RX payload length checking register<br>Provides enable bit to determine whether the RX error signal flags payload lengths that do not match the length field.. |
| 0x605–0x60A | Transmit side pause registers<br>Accessible only if you set **Flow control mode** to the value **Standard flow control** or **Priority-based flow control**. |
| 0x700–0x703 | Receive side pause registers<br>Accessible only if you set **Flow control mode** to the value **Standard flow control** or **Priority-based flow control**. |
| 0x800–0x837, 0x860–0x861 | Transmit side statistics registers<br>Accessible only if you turn on **Enable TX statistics** |
| 0x845 | Transmit statistics counters configuration register `CNTR_TX_CONFIG`<br>Accessible only if you turn on **Enable TX statistics** |
| 0x846 | Transmit statistics counters status register<br>Accessible only if you turn on **Enable TX statistics** |
| 0x900–0x937, 0x960–0x961 | Receive side statistics registers<br>Accessible only if you turn on **Enable RX statistics** |

| Word Offset | Register Description |
|---|---|
| 0x945 | Receive statistics counters configuration register `CNTR_RX_CONFIG`<br>Accessible only if you turn on **Enable RX statistics** |
| 0x946 | Receive statistics counters status register<br>Accessible only if you turn on **Enable RX statistics** |
| 0xA00–0xA0C | TX 1588 PTP registers<br>Accessible only if you turn on **Enable 1588 PTP** |
| 0xB00–0xB06 | RX 1588 PTP registers<br>Accessible only if you turn on **Enable 1588 PTP** |

**Table 3-20: LL 40-100GbE Hardware Design Example Registers**

Lists the memory mapped register ranges for the LL 40-100GbE IP core hardware design example

| Word Offset | Register Category |
|---|---|
| 0x1000–0x1016 | Packet client registers |
| 0x2004–0x2023 | Reserved |
| 0x4000–0x4C00 | Arria 10 dynamic reconfiguration register base addresses for four-lane variations. Register base address is 0x4000 for Lane 0, 0x4400 for Lane 1, 0x4800 for Lane 2, and 0x4C00 for Lane 3. (Bits [11:10] specify the lane). |
| 0x4000–0x4C00 | Arria 10 dynamic reconfiguration register base addresses for ten-lane variations. Register base address is 0x4000 for Lane 0, 0x4400 for Lane 1, 0x4800 for Lane 2, and 0x4C00 for Lane 3,. 0x5000 for Lane 4, ... 0x6400 for Lane 9. (Bits [13:10] specify the lane). |

**Related Information**

- **Control and Status Interface** on page 3-56
- **PHY Registers** on page 3-76
- **Link Fault Signaling Registers** on page 3-79
  Describes the fault link signaling and fault status signal registers.
- **Low Latency 40-100GbE IP Core MAC Configuration Registers** on page 3-92
- **Pause Registers** on page 3-94
- **TX Statistics Registers** on page 3-101
- **RX Statistics Registers** on page 3-107
- **1588 PTP Registers** on page 3-112
- **LL 40-100GbE Hardware Design Example Registers** on page 3-114

- **Arria 10 Transceiver PHY User Guide**
  The 40GBASE-KR4 variations of the LL 40-100GbE IP core use the Arria 10 10GBASE-KR PHY IP core PHY registers at internal offsets 0x4B0–0x4FF (at IP core register map offsets 0xB0–0xFF) and Arria 10 FEC error insertion device registers. Information about this 10GBASE-KR PHY IP core, including register descriptions, is available in the *10GBASE-KR PHY IP Core* section in the *Arria 10 Transceiver PHY User Guide*.. The register descriptions are also duplicated in **Arria 10 10GBASE-KR Registers**.

## Low Latency 40-100GbE IP Core Registers

The following sections describe the registers included in the Low Latency 40-100GbE IP core.

**PHY Registers** on page 3-76

**Link Fault Signaling Registers** on page 3-79

**LL 40GBASE-KR4 Registers** on page 3-81

**Low Latency 40-100GbE IP Core MAC Configuration Registers** on page 3-92

**Pause Registers** on page 3-94

**TX Statistics Registers** on page 3-101

**RX Statistics Registers** on page 3-107

**1588 PTP Registers** on page 3-112

**Related Information**

**Control and Status Interface** on page 3-56

### PHY Registers

**Table 3-21: PHY Registers**

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x300 | PHY_REVID | [31:0] | IP core PHY module revision ID. | 0x02062015 | RO |
| 0x301 | PHY_SCRATCH | [31:0] | Scratch register available for testing. | 32'b0 | RW |
| 0x302 | PHY_NAME_0 | [31:0] | First 4 characters of IP core variation identifier string " 40GE pcs " or "100GE pcs ". | | RO |
| 0x303 | PHY_NAME_1 | [31:0] | Next 4 characters of IP core variation identifier string " 40GE pcs " or "100GE pcs ". | | RO |
| 0x304 | PHY_NAME_2 | [31:0] | Final 4 characters of IP core variation identifier string " 40GE pcs " or "100GE pcs ". | | RO |

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x310 | PHY_CONFIG | [5] | set_data_lock: Directs the PLL to lock to data. | 6'b0 | RW |
| | | [4] | set_ref_lock: Directs the PLL to lock to the reference clock. | | |
| | | [3] | rxp_ignore_freq: Directs the IP core to proceed with the internal reset sequence (to reset the RX PLL) without waiting for the RX CDR PLL to lock. | | |
| | | [2] | soft_rxp_rst: RX PLL soft reset | | |
| | | [1] | soft_txp_rst: TX PLL soft reset | | |
| | | [0] | eio_sys_rst: PMA system reset. Set this bit to start the internal reset sequence. | | |
| 0x313 | PHY_PMA_SLOOP | [3:0] or [9:0] | Serial PMA loopback. Each bit that is asserted directs the IP core to connect the corresponding TX–RX lane pair on the internal loopback path, by setting the corresponding transceiver in serial loopback mode.<br>This option is not available for CAUI-4 variations. | 0 | RW |
| 0x314 | PHY_PCS_INDIRECT_ADDR | [2:0] | Supports indirect addressing of individual FIFO flags in the 10G PCS Native PHY IP core. Program this register with the encoding for a specific FIFO flag. The flag values (one per transceiver) are then accessible in the PHY_PCS_INDIRECT_DATA register.<br>The value in the PHY_PCS_INDIRECT_ADDR register directs the IP core to make available the following FIFO flag:<br>• 3'b111: RX partial empty<br>• 3'b110: RX partial full<br>• 3'b101: RX empty<br>• 3'b100: RX full<br>• 3'b011: TX partial empty<br>• 3'b010: TX partial full<br>• 3'b001: TX empty<br>• 3'b000: TX full | 3'b0 | RW |

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x315 | PHY_PCS_INDIRECT_DATA | [3:0] or [9:0] | PCS indirect data. To read a FIFO flag, set the value in the PHY_PCS_INDIRECT_ADDR register to indicate the flag you want to read. After you set the specific flag indication in the PHY_PCS_INDIRECT_ADDR register, each bit [n] in the PHY_PCS_INDIRECT_DATA register has the value of that FIFO flag for the transceiver channel for lane [n].<br>This register has four valid bits [3:0] in LL 40GbE IP core variations, and ten valid bits [9:0] in standard LL 100GbE IP core variations. This register is not valid for CAUI-4 variations. | TX full flags | RO |
| 0x320 | PHY_TX_PLL_LOCKED | [9:0] | Each bit that is asserted indicates that the corresponding lane TX PLL is locked. | 10'b0 | RO |
| 0x321 | PHY_EIOFREQ_LOCKED | [9:0] | Each bit that is asserted indicates that the corresponding lane RX CDR PLL is locked. | 10'b0 | RO<br>RO |
| 0x322 | PHY_TX_COREPLL_LOCKED | [2] | RX PLL is locked. | 3'b0 | RO |
| | | [1] | TX PLL is locked. | | |
| | | [0] | TX PCS is ready. | | |
| 0x323 | PHY_FRAME_ERROR | [3:0] or [19:0] | Each bit that is asserted indicates that the corresponding virtual lane has a frame error. If you read a non-zero value in this register,<br>You can clear these bits with the PHY_SCLR_FRAME_ERROR register.<br>Altera recommends that you set bit [0] of the PHY_SCLR_FRAME_ERROR register and read the PHY_FRAME_ERROR register again to determine if the PHY frame error is generated continuously.<br>These bits are not sticky: the IP core clears them more than once while attempting to achieve alignment. | 0xF or 0xFFFFF | RO |
| 0x324 | PHY_SCLR_FRAME_ERROR | [0] | Synchronous clear for PHY_FRAME_ERROR register. Write the value of 1 to this register to clear the PHY_FRAME_ERROR register. | 1'b0 | RW |
| 0x325 | PHY_EIO_SFTRESET | [1] | Set this bit to clear the RX FIFO. | 2'b00 | RW |
| | | [0] | RX PCS reset: set this bit to reset the RX PCS. | | |

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x326 | PHY_RXPCS_STATUS | [1:0] | Indicates the RX PCS is fully aligned and ready to accept traffic.<br>• Bit [1]: HI BER status (bit error rate is high according to Ethernet standard definition). This bit maintains the value of 1'b0 unless you turn on **Enable link fault generation**.<br>• Bit [0]: RX PCS fully aligned status. | 0 | RO |
| 0x340 | PHY_REFCLK_KHZ | [31:0] | Reference clock frequency in KHz, assuming the clk_status clock has the frequency of 100 MHz. The reference clock frequency is the value in the PHY_REFCLK_KHZ register times the frequency of the clk_status clock, divided by 100. | | RO |
| 0x341 | PHY_RXCLK_KHZ | [31:0] | RX clock (clk_rxmac) frequency in KHz, assuming the clk_status clock has the frequency of 100 MHz. The RX clock frequency is the value in the PHY_RXCLK_KHZ register times the frequency of the clk_status clock, divided by 100. | | RO |
| 0x342 | PHY_TXCLK_KHZ | [31:0] | TX clock (clk_txmac) frequency in KHz, assuming the clk_status clock has the frequency of 100 MHz. The TX clock frequency is the value in the PHY_TXCLK_KHZ register times the frequency of the clk_status clock, divided by 100. | | RO |
| 0x343 | PHY_RECCLK_KHZ | [31:0] | RX recovered clock frequency in KHz, assuming the clk_status clock has the frequency of 100 MHz. The RX recovered clock frequency is the value in the PHY_RECCLK_KHZ register times the frequency of the clk_status clock, divided by 100. | | RO |
| 0x344 | PHY_TXIOCLK_KHZ | [31:0] | TX PMA clock frequency in KHz, assuming the clk_status clock has the frequency of 100 MHz. The TX PMA clock frequency is the value in the PHY_TX10CLK_KHZ register times the frequency of the clk_status clock, divided by 100. | | RO |

## Link Fault Signaling Registers

### Table 3-22: LINK_FAULT_CONFIG Register—Offset 0x405

| Name | Bit | Description | Reset Value | Access |
|---|---|---|---|---|
| Unidir Enable | [1] | When asserted, the IP core includes Clause 66 support for remote link fault reporting on the Ethernet link. | 1'b0 | RW |
| Link Fault Reporting Enable | [0] | • 1'b1: The PCS generates a proper fault sequence on the Ethernet link, if conditions are met.<br>• 1'b0: The PCS does not generate any fault sequences.<br>Whether this bit has the value of 0 or 1, the RX PCS always reports real-time link status on the `local_fault_status` and `remote_fault_status` output signals and in the `Local Fault Status` register at offset 0x508. | 1'b1 | RW |

**Table 3-23: Link Fault Status Register—Offset 0x508**

| Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|
| `Remote Fault Status` | [1] | The remote fault status register.The IP core sets this bit when it detects real-time remote faults in the RX MAC. The IP core sets this bit regardless of the settings in the `LINK_FAULT_CONFIG` register. | 1'b0 | RO |
| `Local Fault Status` | [0] | The local fault status register. The IP core sets this bit when it detects real-time local faults in the RX MAC. The IP core sets this bit regardless of the settings in the `LINK_FAULT_CONFIG` register. | 1'b1 | RO |

**Related Information**

**Link Fault Signaling Interface** on page 3-33
Describes how the IP core uses the register values.

## LL 40GBASE-KR4 Registers

Most LL 40GBASE-KR4 registers are 10GBASE-KR PHY registers of the Arria 10 10GBASE-KR PHY IP core, documented in the **Arria 10 Transceiver PHY User Guide**. Exceptions are:

- The register offsets of the 10GBASE-KR PHY registers are offset by negative 0x400 in the LL 40GBASE-KR4 variations of the LL 40-100GbE IP core. The Arria 10 10GBASE-KR PHY IP core registers begin at offset 0x4B0. In the LL 40GBASE-KR4 IP core, these registers begin at offset 0x0B0.
- The LL 40GBASE-KR4 variations of the LL 40-100GbE IP core have additional 40GBASE-KR4 related registers and register fields.
- The FEC error insertion feature requires that you program some Arria 10 device registers through the Arria 10 dynamic reconfiguration interface. The FEC error count is collected in other Arria 10 device registers that you access through the Arria 10 dynamic reconfiguration interface. You access the relevant Arria 10 device registers at offsets 0xBD through 0xE3 for Lane 0, 0x4BD through 0x4E3 for Lane 1, 0x8BD through 0x8E3 for Lane 2, and 0xCBD through 0xCE3 for Lane 3. The descriptions of the LL 40GBASE-KR4 registers that depend on these Arria 10 device registers provide the individual A10 register information.

For your convenience, the LL 40-100GbE IP core user guide includes an appendix with the 10GBASE-KR PHY register descriptions: **Arria 10 10GBASE-KR Registers**.

### Table 3-24: LL 40-100GbE IP Core 40GBASE-KR4 Registers and Register Fields Not in Arria 10 10GBASE-KR PHY IP Core

Documents the differences between the Arria 10 10GBASE-KR PHY register definitions and the 40GBASE-KR4 registers of the LL 40-100GbE IP core. All 10GBASE-KR PHY registers and register fields not listed in the table are available in the 40GBASE-KR4 variations of the LL 40-100GbE IP core.

The 10GBASE-KR PHY register listings are available in the 0xB0**Arria 10 10GBASE-KR Registers** appendix and in the *10GBASE-KR PHY Register Defintions* section of the *10GBASE-KR PHY IP Core* section in the *Arria 10 Transceiver PHY User Guide*. The information in these two sources should be identical. Refer to **Arria 10 10GBASE-KR Registers** for details.

Where the 10GBASE-KR PHY register definitions list 10GBASE-R, substitute 40GBASE-KR4 with auto-negotiation and link training both turned off, and where the 10GBASE-KR PHY register definitions list 10GBASE-KR (except in the description of 0xCB[24:0]), substitute 40GBASE-KR4. Where a register field description in the 10GBASE-KR PHY register definitions refers to link training or FEC in the single-lane 10GBASE-KR PHY IP core, substitute link training or FEC on Lane 0 of the 40GBASE-KR4 IP core variation.

Registers and register bits not described in either location should be considered Reserved.

To modify a field value in any LL 40GBASE-KR4 specific register, whether an underlying 10GBASE-KR PHY IP core register or one of the registers defined in this table, you must perform a read-modify-write operation to ensure you do not modify the values of any other fields in the register.

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x0B0 | SEQ Force Mode[3:0] | [7:4] | Forces the sequencer to a specific protocol. Must write the `Reset SEQ` bit (bit [0]) to 1 for the | 4'b0 | RW |

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| | | | Force to take effect. The following encodings are defined:<br>• 0000: No force<br>• 0001: GigE mode (unsupported)<br>• 0010: XAUI mode (unsupported)<br>• 0100: 40GBASE-R4 mode (without auto-negotiation and without link training)<br>• 0101: 40GBASE-KR4 mode<br>• 1100: 40GBASE-R4 mode with FEC (without auto-negotiation and without link training) | | |
| | `Enable Arria 10 Calibration` | [8] | When set to 1, it enables the Arria 10 HSSI reconfiguration calibration as part of the PCS dynamic reconfiguration. 0 skips the calibration when the PCS is reconfigured. | 1'b1 | RW |
| | `LT Failure Response` | [12] | When set to 1, LT failure causes the PHY to go into data mode. When set to 0, LT failure restarts auto-negotiation (if enabled). If auto-negotiation is not enabled, the PHY will restart LT. | 1'b1 in simulation; 1'b0 in hardware | RW |
| | Reserved | [17] | Reserved<br>Access FEC error indication selection using the Arria 10 dynamic reconfiguration interface. Refer to the descriptions of register 0xB2. | | |
| | Reserved | [19] | Reserved | | |
| 0x0B1 | SEQ Reconfig Mode[5:0] | [13:8] | Specifies the Sequencer mode for PCS reconfiguration. The following modes are defined:<br>• Bit 8, mode[0]: AN mode<br>• Bit 9, mode[1]: LT Mode<br>• Bit 10, mode[2]: 40G data mode<br>• Bit 11, mode[3]: GigE data mode<br>• Bit 12, mode[4]: Reserved for XAUI<br>• Bit13, mode[5]: 40G FEC mode | | |
| | `FEC Block Lock` | [23:20] | FEC Block Lock for lanes [3:0]: bit [20] is FEC block lock for lane 0, bit [21] is FEC block lock for lane 1, bit [22] is FEC block lock for lane 2, and bit [23] is FEC block lock for lane 3. | 4'b0 | RO |

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| | KR FEC TX Error Insert, Lane 0 | 11 | Writing a 1 inserts one error pulse into the TX FEC for lane 0, depending on the Transcoder and Burst error settings for lane 0.<br>You must select these settings through the Arria 10 dynamic reconfiguration interface to the Arria 10 device registers before you write a 1 to the KR FEC TX Error Insert, Lane 0 bit. To select these settings for Lane 0, perform a read-modify-write operation sequence at register offset 0xBD.<br>You select a Transcoder error by setting the `transcode_err` bit, resetting the `burst_err` bit, resetting the `burst_err_len` field, and leaving the remaining bits at their previous values.<br>You select a Burst error by setting the `burst_err` bit, specifying the burst error length in the `burst_err_len` field, resetting the `transcode_err` bit, and leaving the remaining bits at their previous values. | 1'b0 | RWSC |
| 0xB2 | RCLR_ERRBLK_CNT, Lane 0 | 12 | Writing a 1 resets the error block counters.Writing a 0 causes counting to resume.<br>Each lane has a 32-bit corrected error block counter and a 32-bit uncorrected error block counter in the Arria 10 device registers. Refer to *Clause 74.8.4.1* and *Clause 74.8.4.2* of *IEEE Std 802.3ap-2007*.<br>For Lane 0, the corrected error block counter is in the Arria 10 device registers you access through the Arria 10 dynamic reconfiguration interface at offsets 0xDC to 0xDF: `blkcnt_corr[31:0]` is in {0xDF[7:0],0xDE[7:0], 0xDD[7:0],0xDC[7:0]}.<br>For Lane 0, the uncorrected error block counter is in the Arria 10 device registers you access through the Arria 10 dynamic reconfiguration interface at offsets 0xE0 to 0xE3: `blkcnt_uncorr[31:0]` is in {0xE3[7:0],0xE2[7:0], 0xE1[7:0],0xE0[7:0]}. | | RW |
| | Reserved | [31:13] | Reserved | | |
| 0x0B5 | Register 0xB2 refers to Lane 0. This register is the equivalent of register 0xB2 for Lane 1. The relevant FEC error Arria 10 device registers for Lane 1 are at 0x4BD through 0x4E3 (additional offset of 0x400). | | | | RW |

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x0B8 | This register is the equivalent of register 0xB2 for Lane 2. The relevant FEC error Arria 10 device registers for Lane 2 are at 0x8BD through 0x8E3 (additional offset of 0x800 compared to the Lane 0 device registers). | | | | RW |
| 0x0BB | This register is the equivalent of register 0xB2 for Lane 3. The relevant FEC error Arria 10 device registers for Lane 3 are at 0xCBD through 0xCE3 (additional offset of 0xC00 compared to the Lane 0 device registers). | | | | RW |
| 0x0C0 | `Override AN Channel Enable` | [6] | Overrides the auto-negotiation master channel that you set with the **Auto-Negotiation Master** parameter, setting the new master channel according to the value in register 0xCC[3:0].<br>While 0x0C0[6] has the value of 1, the channel encoded in 0xCC[3:0] is the master channel. While 0xC0[6] has the value of 0, the master channel is the channel that you set with the **Auto-Negotiation Master** parameter. | 1'b0 | RW |
| 0x0C2 | KR4 AN Link Ready [5:0] | [17:12] | Provides a one-hot encoding of an_receive_idle = true and link status for the supported link as described in Clause 73.10.1. The following encodings are defined:<br>• 6'b000000: 1000BASE-KX<br>• 6'b000001: 10GBASE-KX4<br>• 6'b000100: 10GBASE-KR<br>• 6'b001000: 40GBASE-KR4<br>• 6'b010000: 40GBASE-CR4<br>• 6'b100000: 100GBASE-CR10<br>The only valid value for the LL 40GBASE-KR4 IP core is 6'b001000: 40GBASE-KR4. | 6'b001000 | RO |
| 0x0CB | `AN LP ADV Tech_A[24:0]` | [24:0] | Received technology ability field bits of Clause 73 Auto Negotiation. The following protocols are defined:<br>• A0 1000BASE-KX<br>• A1 10GBASE-KX4<br>• A2 10GBASE-KR<br>• A3 40GBASE-KR4<br>• A4 40GBASE-CR4<br>• A5 100GBASE-CR10<br>• A24:6 are reserved<br>The only valid value for the LL 40GBASE-KR4 IP core is A3: 40GBASE-KR4.. | 25'b0 | RO |

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| | | | For more information, refer to Clause 73.6.4 and AN LP base page ability registers (7.19-7.21) of Clause 45 of *IEEE 802.3ap-2007*. | | |
| 0x0CC | `Override AN Channel Select` | [3:0] | If you set the value of the Override AN Channel Enable register field (0xC0[6]) to the value of 1, then while 0xC0[6] has the value of 1, the value in this register field (0xCC[3:0])overrides the master channel you set with the **Auto-Negotiation Master** parameter.<br>This register field has the following valid values:<br>• 4'b0001: Selects Lane 0<br>• 4'b0010: Selects Lane 1<br>• 4'b0100: Selects Lane 2<br>• 4'b1000: Selects Lane 3<br>All other values are invalid. The new master channel is encoded with one-hot encoding. | 4'b0 | RW |
| 0x0D0 | Reserved | [3:2] | Reserved | | |
| | `Ovride LP Coef Enable` | [16] | When set to 1, overrides the link partner's equalization coefficients; software changes the update commands sent to the link partner TX equalizer coefficients. When set to 0, uses the Link Training logic to determine the link partner coefficients. Used with 0x0D1 bits [7:4] and bits[7:0] of 0x0D4 through 0x0D7. | 1'b0 | RW |
| | `Ovride Local RX Coef Enable` | [17] | When set to 1, overrides the local device equalization coefficients generation protocol. When set, the software changes the local TX equalizer coefficients. When set to 0, uses the update command received from the link partner to determine local device coefficients. Used with 0x0D1 bits [11:8] and bits[23:16] of 0x0D4 through 0x0D7. | 1'b0 | RW |
| | Reserved | [31:22] | Reserved | | |
| 0x0D1 | `Restart Link training, Lane 1` | [1] | When set to 1, resets the 40GBASE-KR4 start-up protocol. When set to 0, continues normal operation. This bit self clears. Refer to the state variable `mr_restart_training` as defined in *Clause 72.6.10.3.1* and 10GBASE-KR PMD control register bit (1.150.0) in *IEEE Std 802.3ap-2007*. | 1'b0 | RW SC |

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| | | | Register bit 0xD1[0] refers to Lane 0. This bit is the equivalent of register 0xD1[0] for Lane 1. | | |
| | `Restart Link training, Lane 2` | [2] | This bit is the equivalent of register 0xD1[0] for Lane 2. | 1'b0 | RW SC |
| | `Restart Link training, Lane 3` | [3] | This bit is the equivalent of register 0xD0[1] for Lane 3. | 1'b0 | RW SC |
| 0x0D1 | `Updated TX Coef new, Lane 1` | [5] | When set to 1, indicates that new link partner coefficients are available to send. The LT logic starts sending the new values set in 0xD4[7:0] to the remote device. When set to 0, continues normal operation. This bit self clears.<br>This override of normal operation can only occur if 0xD0[16] (`Ovride LP Coef enable`) has the value of 1. If 0xD0[16] has the value of 0, this register field (0xD1[5]) has no effect.<br>Register bit 0xD1[4] refers to Lane 0. This bit is the equivalent of register 0xD1[4] for Lane 1. | 1'b0 | RW SC |
| | `Updated TX Coef new, Lane 2` | [6] | This bit is the equivalent of register 0xD1[5] for Lane 2. | 1'b0 | RW SC |
| | `Updated TX Coef new, Lane 3` | [7] | This bit is the equivalent of register 0xD1[5] for Lane 3. | 1'b0 | RW SC |
| 0x0D1 | `Updated RX Coef new, Lane 1` | [9] | When set to 1, indicates that new local device coefficients are available for Lane 1. The LT logic changes the local TX equalizer coefficients as specified in 0xE1[23:16]. When set to 0, continues normal operation. This bit self clears.<br>This override of normal operation can only occur if 0xD0[17] (`Ovride Local RX Coef enable`) has the value of 1. If 0xD0[17] has the value of 0, this register field (0xD1[9]) has no effect.<br>Register bit 0xD1[8] refers to Lane 0. This bit is the equivalent of register 0xD1[8] for Lane 1. | 1'b0 | RW |

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| | Updated RX Coef new, Lane 2 | [10] | When set to 1, indicates that new local device coefficients are available for Lane 2. The LT logic changes the local TX equalizer coefficients as specified in 0xE5[23:16]. When set to 0, continues normal operation. This bit self clears.<br>This override of normal operation can only occur if 0xD0[17] (Ovride Local RX Coef enable) has the value of 1.<br>This bit is the equivalent of register 0xD1[9] for Lane 2. | 1'b0 | RW |
| | Updated RX Coef new, Lane 3 | [11] | When set to 1, indicates that new local device coefficients are available for lane 3. The LT logic changes the local TX equalizer coefficients as specified in 0xE9[23:16]. When set to 0, continues normal operation. This bit self clears.<br>This override of normal operation can only occur if 0xD0[17] (Ovride Local RX Coef enable) has the value of 1.<br>This bit is the equivalent of register 0xD1[9] for Lane 3. | 1'b0 | RW |
| 0x0D2 | Reserved | [7:6] | Reserved | | |
| | | [13:8] | Register bits 0xD2[5:0] refer to Lane 0. These bits are the equivalent of 0xD2[5:0] for Lane 1.<br>For Link Training Frame lock Error, Lane 1, if the tap settings specified by the fields of 0xE2 are the same as the initial parameter value, the frame lock error was unrecoverable. | | RO |
| | | [21:16] | These bits are the equivalent of 0xD2[5:0] for Lane 2.<br>For Link Training Frame lock Error, Lane 2, if the tap settings specified by the fields of 0xE6 are the same as the initial parameter value, the frame lock error was unrecoverable. | | RO |
| | | [29:24] | These bits are the equivalent of 0xD2[5:0] for Lane 3.<br>For Link Training Frame lock Error, Lane 3, if the tap settings specified by the fields of 0xEA are the same as the initial parameter value, the frame lock error was unrecoverable. | | RO |

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x0D4 | `LD coefficient update[5:0], Lane 0` | [5:0] | Reflects the contents of the first 16-bit word of the training frame sent to Lane 0 from the local device control channel. Normally, the bits in this register are read-only; however, when you override training by setting the `Ovride LP Coef enable` control bit (0x0D0 bit [16]), these bits become writeable. The following fields are defined:<br>• [5: 4]: Coefficient (+1) update<br>  • 2'b11: Reserved<br>  • 2'b01: Increment<br>  • 2'b10: Decrement<br>  • 2'b00: Hold<br>• [3:2]: Coefficient (0) update (same encoding as [5:4])<br>• [1:0]: Coefficient (-1) update (same encoding as [5:4])<br>Before you can send these bits, you must enable the override in 0x0D0 bit [16] and also signal a new word in 0x0D1 bit [4].<br>For more information, refer to bit 10G BASE-KR LD coefficient update register bits (1.154.5:0) in Clause 45.2.1.80.3 of *IEEE 802.3ap-2007*. | | RO/ RW |

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| | LP Coefficient Update[5:0], Lane 0 | [21:16] | Reflects the contents of the first 16-bit word of the training frame most recently received on Lane 0 from the control channel.<br>Normally the bits in this register are read only; however, when training is disabled by setting low the Link Training enable control bit (bit 0 at offset 0xD0), these bits become writeable. The following fields are defined:<br>• [5: 4]: Coefficient (+1) update<br>  • 2'b11: Reserved<br>  • 2'b01: Increment<br>  • 2'b10: Decrement<br>  • 2'b00: Hold<br>• [3:2]: Coefficient (0) update (same encoding as [5:4])<br>• [1:0]: Coefficient (-1) update (same encoding as [5:4])<br>Before you can send these bits, you must enable the override in 0x0D0 bit [17] and also signal a new word in 0x0D2 bit [8].<br>For more information, refer to bit 10G BASE-KR LP coefficient update register bits (1.152.5:0) in Clause 45.2.1.78.3 of *IEEE 802.3ap-2007*. | | |
| 0x0D5 | Reserved | [31:21] | Reserved | | |
| 0x0D6 | LT VODMAX ovrd, Lane 0 | [4:0] | Override value for the **VMAXRULE** parameter on Lane 0. When enabled, this value substitutes for the **VMAXRULE** to allow channel-by-channel override of the device settings. This only effects the local device TX output for the channel specified.<br>This value must be greater than the **INITMAINVAL** parameter for proper operation. Note this will also override the **PREMAINVAL** parameter value. | 0x1C (28 decimal) for simulation; 0 for compilation | RW |
| | LT VODMAX ovrd Enable, Lane 0 | [5] | When set to 1, enables the override value for the **VMAXRULE** parameter stored in the LT VODMAX ovrd, Lane 0 register field. | 1 for simulation; 0 for compilation | RW |

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| | `LT VODMin ovrd, Lane 0` | [12:8] | Override value for the **VODMINRULE** parameter on Lane 0. When enabled, this value substitutes for the **VMINRULE** to allow channel-by-channel override of the device settings. This override only effects the local device TX output for this channel.<br>The value to be substituted must be less than the **INITMAINVAL** parameter and greater than the **VMINRULE** parameter for proper operation. | 0x19 (25 decimal) for simulation; 0 for compilation | RW |
| | `LT VODMin ovrd Enable, Lane 0` | [13] | When set to 1, enables the override value for the **VODMINRULE** parameter stored in the `LT VODMin ovrd, Lane 0` register field. | 1 for simulation; 0 for compilation | RW |
| | `LT VPOST ovrd, Lane 0` | [21:16] | Override value for the **VPOSTRULE** parameter on Lane 0. When enabled, this value substitutes for the **VPOSTRULE** to allow channel-by-channel override of the device settings. This override only effects the local device TX output for this channel.<br>The value to be substituted must be greater than the **INITPOSTVAL** parameter for proper operation. | 6 for simulation; 0 for compilation | RW |
| | `LT VPOST ovrd Enable, Lane 0` | [22] | When set to 1, enables the override value for the **VPOSTRULE** parameter stored in the `LT VPOST ovrd, Lane 0` register field. | 1 for simulation; 0 for compilation | RW |
| | `LT VPre ovrd, Lane 0` | [28:24] | Override value for the **VPRERULE** parameter on Lane 0. When enabled, this value substitutes for the **VPOSTRULE** to allow channel-by-channel override of the device settings. This override only effects the local device TX output for this channel.<br>The value to be substituted must be greater than the **INITPREVAL** parameter for proper operation. | 4 for simulation; 0 for compilation | RW |

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| | `LT VPre ovrd Enable, Lane 0` | [29] | When set to 1, enables the override value for the **VPRERULE** parameter stored in the `LT VPre ovrd, Lane 0` register field. | 1 for simulation; 0 for compilation | RW |
| 0xE0 | Register 0xD3 refers to Lane 0. This register, register 0xE0, is the equivalent of register 0xD3 for Lane 1 link training. | | | | RW |
| 0xE1 | Register 0xD4 refers to Lane 0. This register, register 0xE1, is the equivalent of register 0xD4 for Lane 1 link training. | | | | RW |
| 0xE2 | Register 0xD5 refers to Lane 0. This register, register 0xE2, is the equivalent of register 0xD5 for Lane 1 link training. | | | | RO |
| 0xE3 | Register 0xD6 refers to Lane 0. This register, register 0xE3, is the equivalent of register 0xD6 for Lane 1 link training.. | | | | RW |
| 0xE4 | This register is the equivalent of register 0xD3 for Lane 2 link training. | | | | RW |
| 0xE5 | This register is the equivalent of register 0xD4 for Lane 2 link training. | | | | R / RW |
| 0xE6 | This register is the equivalent of register 0xD5 for Lane 2 link training. | | | | RO |
| 0xE7 | This register is the equivalent of register 0xD6 for Lane 2 link training. | | | | RW |
| 0xE8 | This register is the equivalent of register 0xD3 for Lane 3 link training. | | | | RW |
| 0xE9 | This register is the equivalent of register 0xD4 for Lane 3 link training. | | | | R / RW |
| 0xEA | This register is the equivalent of register 0xD5 for Lane 3 link training. | | | | RO |
| 0xEB | This register is the equivalent of register 0xD6 for Lane 3 link training. | | | | RW |

**Related Information**

**Altera Transceiver PHY IP Core User Guide**

The 40GBASE-KR4 variations of the 40-100GbE IP core use the 10GBASE-KR PHY IP core PHY registers at internal offsets 0xB0–0xFF (at IP core register map offsets 0xB0–0xFF), in addition to the registers listed in this section. Information about this PHY IP core, including register descriptions, is available in the *Backplane Ethernet 10GBASE-KR PHY IP Core with FEC Option* chapter of the *Altera Transceiver PHY IP Core User Guide*..

## Low Latency 40-100GbE IP Core MAC Configuration Registers

The MAC configuration registers control the following MAC features in the RX and TX datapaths:

- Fault link signaling on the Ethernet link (TX)
- Local and remote fault status signals (RX)
- CRC forwarding (RX)
- Inter-packet gap IDLE removal (TX)
- Maximum frame sizes for the `CNTR_RX_OVERSIZE` and `CNTR_TX_OVERSIZE` counters (RX and TX)

The fault link signaling and fault status signal registers are documented separately. Refer to Related Links below.

### Table 3-25: TX MAC Configuration Registers

This table documents the non-fault link signaling registers in the TX MAC address space. The `LINK_FAULT_CONFIG` register at address 0x405 is documented with the link fault signaling registers. The `LINK_FAULT_CONFIG` register is available only if you turn on **Enable link fault generation** in the Low Latency 40-100GbE parameter editor.

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x400 | `TXMAC_REVID` | [31:0] | TX MAC revision ID. | 0x02062015 | RO |
| 0x401 | `TXMAC_SCRATCH` | [31:0] | Scratch register available for testing. | 32'b0 | RW |
| 0x402 | `TXMAC_NAME_0` | [31:0] | First 4 characters of IP core variation identifier string "40gMACTxCSR" or "100gMACTxCSR". | | RO |
| 0x403 | `TXMAC_NAME_1` | [31:0] | Next 4 characters of IP core variation identifier string "40gMACTxCSR" or "100gMACTxCSR". | | RO |
| 0x404 | `TXMAC_NAME_2` | [31:0] | Final 4 characters of IP core variation identifier string "40gMACTxCSR" or "100gMACTxCSR". | | RO |
| 0x406 | `IPG_COL_REM` | [7:0] | Specifies the number of IDLE columns to be removed in every Alignment Marker period to compensate for alignment marker insertion. You can program this register to a larger value than the default value, for clock compensation.<br>This register is not present if you set the value of the **Average interpacket gap** parameter to **Disable deficit idle counter** in the Low Latency 40-100GbE parameter editor. | 20 (decimal) in 100GbE variations; 4 in 40GbE variations | RW |

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x407 | MAX_TX_SIZE_CONFIG | [15:0] | Maximum size of Ethernet frames for CNTR_TX_OVERSIZE.<br>If the IP core transmits an Ethernet frame of size greater than the number of bytes specified in this register, and the IP core includes TX statistics registers, the IP core increments the 64-bit CNTR_TX_OVERSIZE register. | 9600 (decimal) | RW |

### Table 3-26: RX MAC Configuration Registers

This table documents the non-fault link signaling registers in the RX MAC address space. The local and remote fault status register at address 0x508 is documented with the link fault signaling registers. The local and remote fault status register is available only if you turn on **Enable link fault generation** in the Low Latency 40-100GbE parameter editor.

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x500 | RXMAC_REVID | [31:0] | RX MAC revision ID. | 0x0206 2015 | RO |
| 0x501 | RXMAC_SCRATCH | [31:0] | Scratch register available for testing. | 32'b0 | RW |
| 0x502 | RXMAC_NAME_0 | [31:0] | First 4 characters of IP core variation identifier string "40gMACRxCSR" or "100gMACRxCSR". | | RO |
| 0x503 | RXMAC_NAME_1 | [31:0] | Next 4 characters of IP core variation identifier string "40gMACRxCSR" or "100gMACRxCSR". | | RO |
| 0x504 | RXMAC_NAME_2 | [31:0] | Final 4 characters of IP core variation identifier string "40gMACRxCSR" or "100gMACRxCSR". | | RO |
| 0x506 | MAX_RX_SIZE_CONFIG | [15:0] | Maximum size of Ethernet frames for CNTR_RX_OVERSIZE and for rx_error[3] or l<n>_rx_error[3] and for .the RxOctetsOK register.<br>If the IP core receives an Ethernet frame of size greater than the number of bytes specified in this register, and the IP core includes RX statistics registers, the IP core increments the 64-bit CNTR_RX_OVERSIZE register.<br>An Ethernet frame of size greater than the number of bytes specified in this register is considered oversized and therefore does not contribute to the value in the RxOctetsOK register.and does cause the assertion of rx_error[3] or l<n>_rx_error[3]. | 9600 (decimal) | RW |

| Address | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x507 | MAC_CRC_CONFIG | [0] | The RX CRC forwarding configuration register. Possible values are:<br>• 1'b0: remove RX CRC— do not forward it to the RX client interface.<br>• 1'b1: retain RX CRC—forward it to the RX client interface.<br>In either case, the IP core checks the incoming RX CRC and flags errors. | 1'b0 | RW |
| 0x50A | CFG_PLEN_CHECK | [0] | Enables payload length checking. If you set this bit to the value of 1, bit[4] of the rx_error signal flags any payload lengths that do not match the length field. | 1'b1 | RW |

**Related Information**

**Link Fault Signaling Registers** on page 3-79
Describes the fault link signaling and fault status signal registers.

## Pause Registers

The pause registers together with the pause signals implement the pause functionality defined in the *IEEE 802.3ba-2010 100G Ethernet Standard*. You can program the pause registers to control the insertion and decoding of pause frames, to help reduce traffic in congested networks.

### Table 3-27: TX Ethernet Flow Control (Pause Functionality) Registers

Some registers are different depending on whether you select **Standard flow control** or **Priority-based flow control** in the Low Latency 40-100GbE parameter editor. Where the difference is only whether the register refers to the single standard flow control priority class or whether distinct bits in the register refer to the individual priority queues, the two uses are documented together. In that case we understand that standard flow control effectively supports a single priority queue.

When your IP core implements priority-based flow control, the following registers provide an access window into an internal table of values, one per priority queue. The entry currently accessible in the registers is determined by the value you write to the TX_PAUSE_QNUMBER register.

- RETRANSMIT_XOFF_HOLDOFF_QUANTA at offset 0x608
- TX_PAUSE_QUANTA at offset 0x609

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x600 | TXSFC_REVID | [31:0] | TX standard flow control module revision ID. | 0x01282014 | RO |
| 0x601 | TXSFC_SCRATCH | [31:0] | Scratch register available for testing. | 32'b0 | RW |

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x602 | TXSFC_NAME_0 | [31:0] | First 4 characters of IP core variation identifier string "40GSFCTxCSR" or "100GSFCTxCSR". | | RO |
| 0x603 | TXSFC_NAME_1 | [31:0] | Next 4 characters of IP core variation identifier string "40GSFCTxCSR" or "100GSFCTxCSR". | | RO |
| 0x604 | TXSFC_NAME_2 | [31:0] | Final 4 characters of IP core variation identifier string "40GSFCTxCSR" or "100GSFCTxCSR". | | RO |
| 0x605 | TX_PAUSE_EN | [N-1:0] (9) | Enable the IP core to transmit pause frames on the Ethernet link in response to a client request through the pause_insert_tx input signal or the TX_PAUSE_REQUEST register. If your IP core implements priority-based flow control, each bit of this field enables TX pause functionality for the corresponding priority queue.<br>Altera recommends that you signal a pause request using the pause_insert_tx signal rather than using the TX_PAUSE_REQUEST register. | N'b1 ...1 (1'b1 in each defined bit) | RW |

(9) N is the number of priority queues. If the IP core implements Ethernet standard flow control, N is 1.

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x606 | TX_PAUSE_REQUEST | [N-1:0][9] | Pause request. If bit [n] of the TX_PAUSE_EN register has the value of 1, setting bit [n] of the TX_PAUSE_REQUEST register field to the value of 1 triggers a XOFF pause packet insertion into the TX data stream on the Ethernet link. If the IP core implements priority-based flow control, the XOFF pause packet includes identity information for the corresponding priority queue.<br><br>If RETRANSMIT_XOFF_HOLDOFF_EN is turned on for the associated priority queue, as long as the value in TX_PAUSE_REQUEST bit [n] remains high, the IP core retransmits the XOFF pause packet at intervals determined by the retransmit hold-off value associated with this priority queue.<br><br>If bit [n] of the TX_PAUSE_EN register has the value of 1, resetting bit [n] of the TX_PAUSE_REQUEST register field to the value of 0 triggers an XON pause packet insertion into the TX data stream on the Ethernet link. If the IP core implements priority-based flow control, the XON pause packet includes identity information for the corresponding priority queue.<br><br>Other pause registers, described in this table, specify the properties of the pause packets.<br><br>Altera recommends that you signal a pause request using the pause_insert_tx signal rather than using the TX_PAUSE_REQUEST register. | 0 | RW |
| 0x607 | RETRANSMIT_XOFF_HOLDOFF_EN | [N-1:0] [9] | Enable XOFF pause frame retransmission hold-off functionality. If your IP core implements priority-based flow control with multiple priority queues, this register provides access to one bit for each priority queue.<br><br>Altera recommends that you maintain this register at the value of all ones.<br><br>If your IP core implements priority-based flow control, refer also to the description of the CFG_RETRANSMIT_HOLDOFF_EN and CFG_RETRANSMIT_HOLDOFF_QUANTA registers. | N'b1...1 (1'b1 in each defined bit) | RW |

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x608 | RETRANSMIT_XOFF_HOLDOFF_QUANTA | [15:0] | Specifies hold-off time from XOFF pause frame transmission until retransmission, if retransmission hold-off functionality is enabled and pause request remains at the value of 1. If your IP core implements priority-based flow control with multiple priority queues, this register provides access to an internal table of retransmit-hold-off times, one for each priority queue. Accesses are indexed by the value in the TX_PAUSE_QNUMBER register.<br><br>Unit is quanta. One quanta is 512 bit times, which varies according to the datapath width (256 or 512 depending on the IP core variation) and the clk_txmac frequency. Note that in the case of 100GbE IP cores, one quanta is effectively one clk_txmac clock cycle.<br><br>Pause request can be either of the TX_PAUSE_REQUEST register pause request bit and the pause_insert_tx signal. | 0xFFFF | RW |
| 0x609 | TX_PAUSE_QUANTA | [15:0] | Specifies the pause time to be included in XOFF frames.<br><br>If your IP core implements priority-based flow control with multiple priority queues, this register provides access to an internal table of pause times, one for each priority queue. Accesses are indexed by the value in the TX_PAUSE_QNUMBER register.<br><br>Unit is quanta. One quanta is 512 bit times, which varies according to the datapath width (256 or 512 depending on the IP core variation) and the clk_txmac frequency. In 100GbE IP cores, a quanta is effectively a single clk_txmac clock cycle. | 0xFFFF | RW |

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x60A if you set the value of **Flow control mode** to **Standard flow control** in the LL 40-100GbE parameter editor. | TX_XOF_EN | [0] | Enable the TX MAC to pause outgoing Ethernet traffic in response to a pause frame received on the RX Ethernet link and forwarded to the TX MAC by the RX MAC.<br>If the cfg_enable bit of the RX_PAUSE_ENABLE register has the value of 1, the RX MAC processes incoming pause frames. When the RX MAC processes an incoming pause frame with an address match, it notifies the TX MAC to pause outgoing traffic on the TX Ethernet link. The TX MAC pauses outgoing traffic on the TX Ethernet link in response to this notification only if bit [0] of the TX_XOF_EN register has the value of 1.<br>The value in the TX_XOF_EN register is only relevant when the cfg_enable bit of the RX_PAUSE_ENABLE register has the value of 1. If the cfg_enable bit of the RX_PAUSE_ENABLE register has the value of 0, pause frames received on the RX Ethernet link do not reach the TX MAC. | 1'b0 | RW |
| 0x60A if you set the value of **Flow control mode** to **Priority-based flow control** in the LL 40-100GbE parameter editor. | TX_PAUSE_QNUMBER | [2:0] | Queue number (index to internal table) of queue whose relevant values are currently accessible (readable and writeable) in these registers:<br>• RETRANSMIT_XOFF_HOLDOFF_QUANTA at offset 0x608<br>• TX_PAUSE_QUANTA at offset 0x609 | 0 | RW |

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x60B | CFG_RETRANSMIT_HOLDOFF_EN | [0] | The CFG_RETRANSMIT_HOLDOFF_EN and CFG_RETRANSMIT_HOLDOFF_QUANTA registers provide a mechanism to specify a uniform retransmission hold-off delay for all priority queues. If CFG_RETRANSMIT_HOLDOFF_EN has the value of 1, the IP core enforces a retransmission hold-off delay for each priority queue that is the longer of the queue-specific retransmission hold-off delay accessible in the RETRANSMIT_XOFF_HOLDOFF_QUANTA register (if enabled) and the retransmission hold-off delay specified in the CFG_RETRANSMIT_HOLDOFF_QUANTA register.<br><br>CFG_RETRANSMIT_HOLDOFF_QUANTA unit is quanta. One quanta is 512 bit times, which varies according to the datapath width (256 or 512 depending on the IP core variation) and the clk_txmac frequency. In 100GbE IP cores, a quanta is effectively a single clk_txmac clock cycle.<br><br>These registers are present only if you set the value of **Flow control mode** to **Priority-based flow control** in the LL 40-100GbE parameter editor. | 1'b0 | RW |
| 0x60C | CFG_RETRANSMIT_HOLDOFF_QUANTA | [15:0] | | | |
| 0x60D | TX_PFC_DADDRL | [31:0] | TX_PFC_DADDRH contains the 16 most significant bits of the destination address for PFC pause frames.<br><br>TX_PFC_DADDRL contains the 32 least significant bits of the destination address for PFC pause frames.<br><br>This feature allows you to program a destination address other than the standard multicast address for PFC frames, for debug or proprietary purposes.<br><br>{TX_PFC_DADDRH[15:0],TX_PFC_DADDRL[31:0]} can be a broadcast, muilticast, or unicast address.<br><br>These registers are present only if you set the value of **Flow control mode** to **Priority-based flow control** in the LL 40-100GbE parameter editor. | 0xC200_0001 | RW |
| 0x60E | TX_PFC_DADDRH | [15:0] | | 0x0180 | RW |

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x60F | TX_PFC_SADDRL | [31:0] | TX_PFC_SADDRH contains the 16 most significant bits of the source address for PFC pause frames.<br>TX_PFC_SADDRL contains the 32 least significant bits of the source address for PFC pause frames.<br>These registers are present only if you set the value of **Flow control mode** to **Priority-based flow control** in the LL 40-100GbE parameter editor. | 0xCBFC_5ADD | RW |
| 0x610 | TX_PFC_SADDRH | [15:0] | | 0xE100 | RW |

**Table 3-28: RX Ethernet Flow Control (Pause Functionality) Registers**

Some registers are different depending on whether you select **Standard flow control** or **Priority-based flow control** in the Low Latency 40-100GbE parameter editor. Where the difference is only whether the register refers to the single standard flow control priority class or whether distinct bits in the register refer to the individual priority queues, the two uses are documented together. In that case we understand that standard flow control effectively supports a single priority queue.

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x700 | RXSFC_REVID | [31:0] | RX standard flow control module revision ID. | 0x01282014 | RO |
| 0x701 | RXSFC_SCRATCH | [31:0] | Scratch register available for testing. | 32'b0 | RW |
| 0x702 | RXSFC_NAME_0 | [31:0] | First 4 characters of IP core variation identifier string "40gSFCRxCSR" or "100gSFCRxCSR". | | RO |
| 0x703 | RXSFC_NAME_1 | [31:0] | Next 4 characters of IP core variation identifier string "40gSFCRxCSR" or "100gSFCRxCSR". | | RO |
| 0x704 | RXSFC_NAME_2 | [31:0] | Final 4 characters of IP core variation identifier string "40gSFCRxCSR" or "100gSFCRxCSR". | | RO |
| 0x705 | RX_PAUSE_ENABLE | [N-1:0] | cfg_enable bits. When bit [n] has the value of 1, the RX MAC processes the incoming pause frames for priority class n whose address matches {DADDR1[31:0],DADDR0[15:0]}. When bit [n] has the value of 0, the RX MAC does not process any incoming pause frames for priority class n.<br>When the RX MAC processes an incoming pause frame with an address match, it notifies | N'b1...1 (1'b1 in each defined bit) | RW |

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| | | | the TX MAC to pause outgoing traffic on the TX Ethernet link. If you implement priority-based flow control, the TX MAC pauses outgoing traffic from the indicated priority queue to the TX Ethernet link in response to this notification. If you implement standard flow control, the TX MAC pauses outgoing traffic on the TX Ethernet link in response to this notification only if bit [0] of the TX_XOF_EN register has the value of 1. | | |
| 0x706 | RX_PAUSE_FWD | [0] | cfg_fwd_ctrl bit. When this bit has the value of 1, the RX MAC forwards matching pause frames to the RX client interface. When this bit has the value of 0, the RX MAC does not forward matching pause frames to the RX client interface. In both cases, the RX MAC forwards non-matching pause frames to the RX client interface. | 1'b0 | RW |
| 0x707 | RX_PAUSE_DADDRL | [31:0] | RX_PAUSE_DADDRL contains the 32 least significant bits of the destination address for pause frame matching.<br>RX_PAUSE_DADDRH contains the 16 most significant bits of the destination address for pause frame matching.<br>When pause frame processing is turned on, if {RX_PAUSE_DADDRH[15:0],RX_PAUSE_DADDRL[31:0]} matches the incoming pause frame destination address, the IP core processes the pause frame. if there is no match, the IP core does not process the pause frame.<br>{RX_PAUSE_DADDRH[15:0],RX_PAUSE_DADDRL[31:0]} can be a broadcast, muilticast, or unicast address. | 0xC200_0001 | RW |
| 0x708 | RX_PAUSE_DADDRH | [15:0] | | 0x0180 | RW |

**Related Information**

**Pause Control and Generation Interface** on page 3-32
Describes the pause signals available in the Low Latency 40-100GbE IP core.

## TX Statistics Registers

The TX statistics registers count TX Ethernet traffic and errors. The 64-bit statistics registers are designed to roll over, to ensure timing closure on the FPGA. However, these registers should never roll over if the

link is functioning properly. The statistics registers check the size of frames, which includes the following fields:

- Size of the destination address
- Size of the source address
- Size of the data
- Four bytes of CRC

The TX statistics counters module is a synthesis option. The statistics registers are counters that are implemented inside the CSR. When you turn on the **Enable TX statistics** parameter in the Low Latency 40-100GbE parameter editor, the counters are implemented in the CSR. When you turn off the **Enable TX statistics** parameter in the Low Latency 40-100GbE parameter editor, the counters are not implemented in the CSR, and read access to the counters returns read data equal to 0.

Reading the value of a statistics register does not affect its value. A configuration register at offset 0x845 allows you to clear all of the TX statistics counters.

To ensure that the counters you read are consistent, you should issue a shadow request to create a snapshot of all of the TX statistics registers, by setting bit [2] of the configuration register at offset 0x845. Until you reset this bit, the counters continue to increment but the readable values remain constant.

**Table 3-29: Transmit Side Statistics Registers**

| Address | Name- | Description | Access |
|---|---|---|---|
| 0x800 | CNTR_TX_FRAGMENTS_LO | Number of transmitted frames less than 64 bytes and reporting a CRC error (lower 32 bits) | RO |
| 0x801 | CNTR_TX_FRAGMENTS_HI | Number of transmitted frames less than 64 bytes and reporting a CRC error (upper 32 bits) | RO |
| 0x802 | CNTR_TX_JABBERS_LO | Number of transmitted oversized frames reporting a CRC error (lower 32 bits) | RO |
| 0x803 | CNTR_TX_JABBERS_HI | Number of transmitted oversized frames reporting a CRC error (upper 32 bits) | RO |
| 0x804 | CNTR_TX_FCS_LO | Number of transmitted packets with FCS errors. (lower 32 bits) | RO |
| 0x805 | CNTR_TX_FCS_HI | Number of transmitted packets with FCS errors. (upper 32 bits) | RO |
| 0x806 | CNTR_TX_CRCERR_LO | Number of transmitted frames with a frame of length at least 64 reporting a CRC error (lower 32 bits) | RO |
| 0x807 | CNTR_TX_CRCERR_HI | Number of transmitted frames with a frame of length at least 64 reporting a CRC error (upper 32 bits) | RO |
| 0x808 | CNTR_TX_MCAST_DATA_ERR_LO | Number of errored multicast frames transmitted, excluding control frames (lower 32 bits) | RO |

| Address | Name- | Description | Access |
|---|---|---|---|
| 0x809 | CNTR_TX_MCAST_DATA_ERR_HI | Number of errored multicast frames transmitted, excluding control frames (upper 32 bits) | RO |
| 0x80A | CNTR_TX_BCAST_DATA_ERR_LO | Number of errored broadcast frames transmitted, excluding control frames (lower 32 bits) | RO |
| 0x80B | CNTR_TX_BCAST_DATA_ERR_HI | Number of errored broadcast frames transmitted, excluding control frames (upper 32 bits) | RO |
| 0x80C | CNTR_TX_UCAST_DATA_ERR_LO | Number of errored unicast frames transmitted, excluding control frames (lower 32 bits) | RO |
| 0x80D | CNTR_TX_UCAST_DATA_ERR_HI | Number of errored unicast frames transmitted, excluding control frames (upper 32 bits) | RO |
| 0x80E | CNTR_TX_MCAST_CTRL_ERR_LO | Number of errored multicast control frames transmitted (lower 32 bits) | RO |
| 0x80F | CNTR_TX_MCAST_CTRL_ERR_HI | Number of errored multicast control frames transmitted (upper 32 bits) | RO |
| 0x810 | CNTR_TX_BCAST_CTRL_ERR_LO | Number of errored broadcast control frames transmitted (lower 32 bits) | RO |
| 0x811 | CNTR_TX_BCAST_CTRL_ERR_HI | Number of errored broadcast control frames transmitted (upper 32 bits) | RO |
| 0x812 | CNTR_TX_UCAST_CTRL_ERR_LO | Number of errored unicast control frames transmitted (lower 32 bits) | RO |
| 0x813 | CNTR_TX_UCAST_CTRL_ERR_HI | Number of errored unicast control frames transmitted (upper 32 bits) | RO |
| 0x814 | CNTR_TX_PAUSE_ERR_LO | Number of errored pause frames transmitted (lower 32 bits) | RO |
| 0x815 | CNTR_TX_PAUSE_ERR_HI | Number of errored pause frames transmitted (upper 32 bits) | RO |
| 0x816 | CNTR_TX_64B_LO | Number of 64-byte transmitted frames (lower 32 bits), including the CRC field but excluding the preamble and SFD bytes | RO |
| 0x817 | CNTR_TX_64B_HI | Number of 64-byte transmitted frames (upper 32 bits), including the CRC field but excluding the preamble and SFD bytes | RO |

| Address | Name- | Description | Access |
|---|---|---|---|
| 0x818 | CNTR_TX_65to127B_LO | Number of transmitted frames between 65–127 bytes (lower 32 bits) | RO |
| 0x819 | CNTR_TX_65to127B_HI | Number of transmitted frames between 65–127 bytes (upper 32 bits) | RO |
| 0x81A | CNTR_TX_128to255B_LO | Number of transmitted frames between 128 –255 bytes (lower 32 bits) | RO |
| 0x81B | CNTR_TX_128to255B_HI | Number of transmitted frames between 128 –255 bytes (upper 32 bits) | RO |
| 0x81C | CNTR_TX_256to511B_LO | Number of transmitted frames between 256 –511 bytes (lower 32 bits) | RO |
| 0x81D | CNTR_TX_256to511B_HI | Number of transmitted frames between 256 –511 bytes (upper 32 bits) | RO |
| 0x81E | CNTR_TX_512to1023B_LO | Number of transmitted frames between 512–1023 bytes (lower 32 bits) | RO |
| 0x81F | CNTR_TX_512to1023B_HI | Number of transmitted frames between 512 –1023 bytes (upper 32 bits) | RO |
| 0x820 | CNTR_TX_1024to1518B_LO | Number of transmitted frames between 1024–1518 bytes (lower 32 bits) | RO |
| 0x821 | CNTR_TX_1024to1518B_HI | Number of transmitted frames between 1024–1518 bytes (upper 32 bits) | RO |
| 0x822 | CNTR_TX_1519toMAXB_LO | Number of transmitted frames of size between 1519 bytes and the number of bytes specified in the MAX_TX_SIZE_CONFIG register (lower 32 bits) | RO |
| 0x823 | CNTR_TX_1519toMAXB_HI | Number of transmitted frames of siz between 1519 bytes and the number of bytes specified in the MAX_TX_SIZE_CONFIG register (upper 32 bits) | RO |
| 0x824 | CNTR_TX_OVERSIZE_LO | Number of oversized frames (frames with more bytes than the number specified in the MAX_TX_SIZE_CONFIG register) transmitted (lower 32 bits) | RO |
| 0x825 | CNTR_TX_OVERSIZE_HI | Number of oversized frames (frames with more bytes than the number specified in the MAX_TX_SIZE_CONFIG register) transmitted (upper 32 bits) | RO |
| 0x826 | CNTR_TX_MCAST_DATA_OK_LO | Number of valid multicast frames transmitted, excluding control frames (lower 32 bits) | RO |

| Address | Name- | Description | Access |
|---|---|---|---|
| 0x827 | CNTR_TX_MCAST_DATA_OK_HI | Number of valid multicast frames transmitted, excluding control frames (upper 32 bits) | RO |
| 0x828 | CNTR_TX_BCAST_DATA_OK_LO | Number of valid broadcast frames transmitted, excluding control frames (lower 32 bits) | RO |
| 0x829 | CNTR_TX_BCAST_DATA_OK_HI | Number of valid broadcast frames transmitted, excluding control frames (upper 32 bits) | RO |
| 0x82A | CNTR_TX_UCAST_DATA_OK_LO | Number of valid unicast frames transmitted, excluding control frames (lower 32 bits) | RO |
| 0x82B | CNTR_TX_UCAST_DATA_OK_HI | Number of valid unicast frames transmitted, excluding control frames (upper 32 bits) | RO |
| 0x82C | CNTR_TX_MCAST_CTRL_LO | Number of valid multicast frames transmitted, excluding data frames (lower 32 bits) | RO |
| 0x82D | CNTR_TX_MCAST_CTRL_HI | Number of valid multicast frames transmitted, excluding data frames (upper 32 bits) | RO |
| 0x82E | CNTR_TX_BCAST_CTRL_LO | Number of valid broadcast frames transmitted, excluding data frames (lower 32 bits) | RO |
| 0x82F | CNTR_TX_BCAST_CTRL_HI | Number of valid broadcast frames transmitted, excluding data frames (upper 32 bits) | RO |
| 0x830 | CNTR_TX_UCAST_CTRL_LO | Number of valid unicast frames transmitted, excluding data frames (lower 32 bits) | RO |
| 0x831 | CNTR_TX_UCAST_CTRL_HI | Number of valid unicast frames transmitted, excluding data frames (upper 32 bits) | RO |
| 0x832 | CNTR_TX_PAUSE_LO | Number of valid pause frames transmitted (lower 32 bits) | RO |
| 0x833 | CNTR_TX_PAUSE_HI | Number of valid pause frames transmitted (upper 32 bits) | RO |
| 0x834 | CNTR_TX_RUNT_LO | Number of transmitted runt packets (lower 32 bits). The IP core does not transmit frames of length less than nine bytes. The IP core pads frames of length nine bytes to 64 bytes to extend them to 64 bytes. | RO |
| 0x835 | CNTR_TX_RUNT_HI | Number of transmitted runt packets (upper 32 bits). The IP core does not transmit frames of length less than nine bytes. The IP core pads frames of length nine bytes to 64 bytes to extend them to 64 bytes. | RO |

| Address | Name- | Description | Access |
|---|---|---|---|
| 0x836 | CNTR_TX_ST_LO | Number of transmitted frame starts (lower 32 bits) | RO |
| 0x837 | CNTR_TX_ST_HI | Number of transmitted frame starts (upper 32 bits) | RO |
| 0x838–0x83F | Reserved | | |
| 0x840 | TXSTAT_REVID | TX statistics module revision ID. | RO |
| 0x841 | TXSTAT_SCRATCH | Scratch register available for testing. Default value is 0x08. | RW |
| 0x842 | TXSTAT_NAME_0 | First 4 characters of IP core variation identifier string "040gMacStats" or "100gMacStats" | RO |
| 0x843 | TXSTAT_NAME_1 | Next 4 characters of IP core variation identifier string "040gMacStats" or "100gMacStats" | RO |
| 0x844 | TXSTAT_NAME_2 | Final 4 characters of IP core variation identifier string "040gMacStats" or "100gMacStats" | RO |
| 0x845 | CNTR_TX_CONFIG | Bits [2:0]: Configuration of TX statistics counters:<br>• Bit [2]: Shadow request (active high): When set to the value of 1, TX statistics collection is paused. The underlying counters continue to operate, but the readable values reflect a snapshot at the time the pause flag was activated. Write a 0 to release.<br>• Bit [1]: Parity-error clear. When software sets this bit, the IP core clears the parity bit CNTR_TX_STATUS[0]. This bit (CNTR_TX_CONFIG[1]) is self-clearing.<br>• Bit [0]: Software can set this bit to the value of 1 to reset all of the TX statistics registers at the same time. This bit is self-clearing.<br>Bits [31:3] are Reserved. | RW |
| 0x846 | CNTR_TX_STATUS | • Bit [1]: Indicates that the TX statistics registers are paused (while CNTR_TX_CONFIG[2] is asserted) .<br>• Bit [0]: Indicates the presence of at least one parity error in the TX statistics counters.<br>Bits [31:2] are Reserved. | RO |
| 0x860 | TxOctetsOK_LO | Number of transmitted payload bytes in frames with no FCS, undersized, oversized, or payload length errors. This register is compliant with section 5.2.2.18 of the *IEEE Standard 802.3-2008*.<br>This register corresponds to the signals tx_inc_octetsOK[15:0] and tx_inc_octetsOK_valid. | RO |
| 0x861 | TxOctetsOK_HI | | RO |

**Related Information**

Statistics Counters Interface on page 3-35

## RX Statistics Registers

The RX statistics registers count RX Ethernet traffic and errors. The 64-bit statistics registers are designed to roll over, to ensure timing closure on the FPGA. However, these registers should never roll over if the link is functioning properly. The statistics registers check the size of frames, which includes the following fields:

- Size of the destination address
- Size of the source address
- Size of the data
- Four bytes of CRC

The RX statistics counters module is a synthesis option. The statistics registers are counters that are implemented inside the CSR. When you turn on the **Enable RX statistics** parameter in the Low Latency 40-100GbE parameter editor, the counters are implemented in the CSR. When you turn off the **Enable RX statistics** parameter in the Low Latency 40-100GbE parameter editor, the counters are not implemented in the CSR, and read access to the counters returns read data equal to 0.

Reading the value of a statistics register does not affect its value. A configuration register at offset 0x945 allows you to clear all of the RX statistics counters.

To ensure that the counters you read are consistent, you should issue a shadow request to create a snapshot of all of the RX statistics registers, by setting bit [2] of the configuration register at offset 0x945. Until you reset this bit, the counters continue to increment but the readable values remain constant.

**Table 3-30: Receive Side Statistics Registers**

| Address | Name- | Description | Access |
|---|---|---|---|
| 0x900 | `CNTR_RX_FRAGMENTS_LO` | Number of received frames less than 64 bytes and reporting a CRC error (lower 32 bits) | RO |
| 0x901 | `CNTR_RX_FRAGMENTS_HI` | Number of received frames less than 64 bytes and reporting a CRC error (upper 32 bits) | RO |
| 0x902 | `CNTR_RX_JABBERS_LO` | Number of received oversized frames reporting a CRC error (lower 32 bits) | RO |
| 0x903 | `CNTR_RX_JABBERS_HI` | Number of received oversized frames reporting a CRC error (upper 32 bits) | RO |
| 0x904 | `CNTR_RX_FCS_LO` | Number of received packets with FCS errors. This register maintains a count of the number of pulses on the `l<n>_rx_fcs_error` or `rx_fcs_error` output signal (lower 32 bits) | RO |
| 0x905 | `CNTR_RX_FCS_HI` | Number of received packets with FCS errors. This register maintains a count of the number of pulses on the `l<n>_rx_fcs_error` output signal (upper 32 bits) | RO |
| 0x906 | `CNTR_RX_CRCERR_LO` | Number of received frames with a frame of length at least 64, with CRC error (lower 32 bits) | RO |

| Address | Name- | Description | Access |
|---|---|---|---|
| 0x907 | CNTR_RX_CRCERR_HI | Number of received frames with a frame of length at least 64, with CRC error (upper 32 bits) | RO |
| 0x908 | CNTR_RX_MCAST_DATA_ERR_LO | Number of errored multicast frames received, excluding control frames (lower 32 bits) | RO |
| 0x909 | CNTR_RX_MCAST_DATA_ERR_HI | Number of errored multicast frames received, excluding control frames (upper 32 bits) | RO |
| 0x90A | CNTR_RX_BCAST_DATA_ERR_LO | Number of errored broadcast frames received, excluding control frames (lower 32 bits) | RO |
| 0x90B | CNTR_RX_BCAST_DATA_ERR_HI | Number of errored broadcast frames received, excluding control frames (upper 32 bits) | RO |
| 0x90C | CNTR_RX_UCAST_DATA_ERR_LO | Number of errored unicast frames received, excluding control frames (lower 32 bits) | RO |
| 0x90D | CNTR_RX_UCAST_DATA_ERR_HI | Number of errored unicast frames received, excluding control frames (upper 32 bits) | RO |
| 0x90E | CNTR_RX_MCAST_CTRL_ERR_LO | Number of errored multicast control frames received (lower 32 bits) | RO |
| 0x90F | CNTR_RX_MCAST_CTRL_ERR_HI | Number of errored multicast control frames received (upper 32 bits) | RO |
| 0x910 | CNTR_RX_BCAST_CTRL_ERR_LO | Number of errored broadcast control frames received (lower 32 bits) | RO |
| 0x911 | CNTR_RX_BCAST_CTRL_ERR_HI | Number of errored broadcast control frames received (upper 32 bits) | RO |
| 0x912 | CNTR_RX_UCAST_CTRL_ERR_LO | Number of errored unicast control frames received (lower 32 bits) | RO |
| 0x913 | CNTR_RX_UCAST_CTRL_ERR_HI | Number of errored unicast control frames received (upper 32 bits) | RO |
| 0x914 | CNTR_RX_PAUSE_ERR_LO | Number of errored pause frames received (lower 32 bits) | RO |
| 0x915 | CNTR_RX_PAUSE_ERR_HI | Number of errored pause frames received (upper 32 bits) | RO |
| 0x916 | CNTR_RX_64B_LO | Number of 64-byte received frames (lower 32 bits), including the CRC field but excluding the preamble and SFD bytes | RO |

| Address | Name- | Description | Access |
|---|---|---|---|
| 0x917 | CNTR_RX_64B_HI | Number of 64-byte received frames (upper 32 bits), including the CRC field but excluding the preamble and SFD bytes | RO |
| 0x918 | CNTR_RX_65to127B_LO | Number of received frames between 65–127 bytes (lower 32 bits) | RO |
| 0x919 | CNTR_RX_65to127B_HI | Number of received frames between 65–127 bytes (upper 32 bits) | RO |
| 0x91A | CNTR_RX_128to255B_LO | Number of received frames between 128 –255 bytes (lower 32 bits) | RO |
| 0x91B | CNTR_RX_128to255B_HI | Number of received frames between 128 –255 bytes (upper 32 bits) | RO |
| 0x91C | CNTR_RX_256to511B_LO | Number of received frames between 256 –511 bytes (lower 32 bits) | RO |
| 0x91D | CNTR_RX_256to511B_HI | Number of received frames between 256 –511 bytes (upper 32 bits) | RO |
| 0x91E | CNTR_RX_512to1023B_LO | Number of received frames between 512–1023 bytes (lower 32 bits) | RO |
| 0x91F | CNTR_RX_512to1023B_HI | Number of received frames between 512 –1023 bytes (upper 32 bits) | RO |
| 0x920 | CNTR_RX_1024to1518B_LO | Number of received frames between 1024–1518 bytes (lower 32 bits) | RO |
| 0x921 | CNTR_RX_1024to1518B_HI | Number of received frames between 1024–1518 bytes (upper 32 bits) | RO |
| 0x922 | CNTR_RX_1519toMAXB_LO | Number of received frames between 1519 bytes and the maximum size defined in the MAX_RX_SIZE_CONFIG register (lower 32 bits) | RO |
| 0x923 | CNTR_RX_1519toMAXB_HI | Number of received frames between 1519 bytes and the maximum size defined in the MAX_RX_SIZE_CONFIG register (upper 32 bits) | RO |
| 0x924 | CNTR_RX_OVERSIZE_LO | Number of oversized frames (frames with more bytes than the number specified in the MAX_RX_SIZE_CONFIG register) received (lower 32 bits) | RO |

| Address | Name- | Description | Access |
|---|---|---|---|
| 0x925 | CNTR_RX_OVERSIZE_HI | Number of oversized frames (frames with more bytes than the number specified in the MAX_RX_SIZE_CONFIG register) received (upper 32 bits) | RO |
| 0x926 | CNTR_RX_MCAST_DATA_OK_LO | Number of valid multicast frames received, excluding control frames (lower 32 bits) | RO |
| 0x927 | CNTR_RX_MCAST_DATA_OK_HI | Number of valid multicast frames received, excluding control frames (upper 32 bits) | RO |
| 0x928 | CNTR_RX_BCAST_DATA_OK_LO | Number of valid broadcast frames received, excluding control frames (lower 32 bits) | RO |
| 0x929 | CNTR_RX_BCAST_DATA_OK_HI | Number of valid broadcast frames received, excluding control frames (upper 32 bits) | RO |
| 0x92A | CNTR_RX_UCAST_DATA_OK_LO | Number of valid unicast frames received, excluding control frames (lower 32 bits) | RO |
| 0x92B | CNTR_RX_UCAST_DATA_OK_HI | Number of valid unicast frames received, excluding control frames (upper 32 bits) | RO |
| 0x92C | CNTR_RX_MCAST_CTRL_LO | Number of valid multicast frames received, excluding data frames (lower 32 bits) | RO |
| 0x92D | CNTR_RX_MCAST_CTRL_HI | Number of valid multicast frames received, excluding data frames (upper 32 bits) | RO |
| 0x92E | CNTR_RX_BCAST_CTRL_LO | Number of valid broadcast frames received, excluding data frames (lower 32 bits) | RO |
| 0x92F | CNTR_RX_BCAST_CTRL_HI | Number of valid broadcast frames received, excluding data frames (upper 32 bits) | RO |
| 0x930 | CNTR_RX_UCAST_CTRL_LO | Number of valid unicast frames received, excluding data frames (lower 32 bits) | RO |
| 0x931 | CNTR_RX_UCAST_CTRL_HI | Number of valid unicast frames received, excluding data frames (upper 32 bits) | RO |
| 0x932 | CNTR_RX_PAUSE_LO | Number of valid pause frames received (lower 32 bits) | RO |
| 0x933 | CNTR_RX_PAUSE_HI | Number of valid pause frames received (upper 32 bits) | RO |

| Address | Name- | Description | Access |
|---|---|---|---|
| 0x934 | CNTR_RX_RUNT_LO | Number of received runt packets (lower 32 bits)<br>A run is a packet of size less than 64 bytes but greater than eight bytes. If a packet is eight bytes or smaller, it is considered a decoding error and not a runt frame, and the IP core does not flag it nor count it as a runt. | RO |
| 0x935 | CNTR_RX_RUNT_HI | Number of received runt packets (upper 32 bits)<br>A run is a packet of size less than 64 bytes but greater than eight bytes. If a packet is eight bytes or smaller, it is considered a decoding error and not a runt frame, and the IP core does not flag it nor count it as a runt. | RO |
| 0x936 | CNTR_RX_ST_LO | Number of received frame starts (lower 32 bits) | RO |
| 0x937 | CNTR_RX_ST_HI | Number of received frame starts (upper 32 bits) | RO |
| 0x938–0x93F | Reserved | | |
| 0x940 | RXSTAT_REVID | RX statistics module revision ID. | RO |
| 0x941 | RXSTAT_SCRATCH | Scratch register available for testing. Default value is 0x09. | RW |
| 0x942 | RXSTAT_NAME_0 | First 4 characters of IP core variation identifier string "040gMacStats" or "100gMacStats" | RO |
| 0x943 | RXSTAT_NAME_1 | Next 4 characters of IP core variation identifier string "040gMacStats" or "100gMacStats" | RO |
| 0x944 | RXSTAT_NAME_2 | Final 4 characters of IP core variation identifier string "040gMacStats" or "100gMacStats" | RO |
| 0x945 | CNTR_RX_CONFIG | Bits [2:0]: Configuration of RX statistics counters:<br>• Bit [2]: Shadow request (active high): When set to the value of 1, RX statistics collection is paused. The underlying counters continue to operate, but the readable values reflect a snapshot at the time the pause flag was activated. Write a 0 to release.<br>• Bit [1]: Parity-error clear. When software sets this bit, the IP core clears the parity bit CNTR_RX_STATUS[0]. This bit (CNTR_RX_CONFIG[1]) is self-clearing.<br>• Bit [0]: Software can set this bit to the value of 1 to reset all of the RX statistics registers at the same time. This bit is self-clearing.<br>Bits [31:3] are Reserved. | RW |

| Address | Name- | Description | Access |
|---|---|---|---|
| 0x946 | CNTR_RX_STATUS | • Bit [1]: Indicates that the RX statistics registers are paused (while CNTR_RX_CONFIG[2] is asserted) .<br>• Bit [0]: Indicates the presence of at least one parity error in the RX statistics counters.<br>Bits [31:2] are Reserved. | RO |
| 0x960 | RxOctetsOK_LO | Number of received payload bytes in frames with no FCS, undersized, oversized, or payload length errors. This register is compliant with section 5.2.1.14 of the *IEEE Standard 802.3-2008*.<br>This register corresponds to the signals rx_inc_octetsOK[15:0] and rx_inc_octetsOK_valid. | RO |
| 0x961 | RxOctetsOK_HI | | RO |

**Related Information**

**Statistics Counters Interface** on page 3-35

## 1588 PTP Registers

The 1588 PTP registers together with the 1588 PTP signals process and provide Precision Time Protocol (PTP) timestamp information as defined in the *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard*. The 1588 PTP module provides you the support to implement the 1588 Precision Time Protocol in your design.

**Table 3-31: TX 1588 PTP Registers**

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0xA00 | TXPTP_REVID | [31:0] | IP core revision ID. | | RO |
| 0xA01 | TXPTP_SCRATCH | [31:0] | Scratch register available for testing. | 32'b0 | RW |
| 0xA02 | TXPTP_NAME_0 | [31:0] | First 4 characters of IP core variation identifier string "40gPTPTxCSR" or "100gPTPTxCSR" | | RO |
| 0xA03 | TXPTP_NAME_1 | [31:0] | Next 4 characters of IP core variation identifier string "40gPTPTxCSR" or "100gPTPTxCSR" | | RO |
| 0xA04 | TXPTP_NAME_2 | [31:0] | Final 4 characters of IP core variation identifier string "40gPTPTxCSR" or "100gPTPTxCSR" | | RO |

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0xA05 | TX_PTP_CLK_PERIOD | [19:0] | clk_txmac clock period.<br>Bits [19:16]: nanoseconds<br>Bits [15:0]: fraction of nanosecond | This value is set to the correct clock period for the required TX MAC clock frequency. The clock period is different for 40GbE variations and 100GbE variations. | RW |
| 0xA06–0xA09 | Reserved | [95:0] | Reserved | 96'b0 | RO |
| 0xA0A | TX_PTP_EXTRA_LATENCY | [31:0] | User-defined extra latency the IP core adds to outgoing timestamps.<br>Bits [31:16]: nanoseconds<br>Bits [15:0]: fraction of nanosecond | 32'b0 | RW |
| 0xA0B | TX_PTP_ASYM_DELAY | [18:0] | Asymmetry adjustment as required for delay measurement. The IP core adds this value to the final delay. | 19'b0 | RW |
| 0xA0C | TX_PTP_PMA_LATENCY | [31:0] | Latency through the TX PMA. This is the delay from the TX MAC output to the tx_serial pin .<br>The IP core sets this register to a value that is sufficiently accurate in most cases. Altera recommends that you modify this value with extreme caution. | | RW |

**Table 3-32: RX 1588 PTP Registers**

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0xB00 | RXPTP_REVID | [31:0] | IP core revision ID. | | RO |
| 0xB01 | RXPTP_SCRATCH | [31:0] | Scratch register available for testing. | 32'b0 | RW |
| 0xB02 | RXPTP_NAME_0 | [31:0] | First 4 characters of IP core variation identifier string "40GPTPRxCSR" or "100GPTPRxCSR" | | RO |

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0xB03 | RXPTP_NAME_1 | [31:0] | Next 4 characters of IP core variation identifier string "40GPTPRxCSR" or "100GPTPRxCSR" | | RO |
| 0xB04 | RXPTP_NAME_2 | [31:0] | Final 4 characters of IP core variation identifier string "40GPTPRxCSR" or "100GPTPRxCSR" | | RO |
| 0xB05 | RX_PTP_CLK_PERIOD | [19:0] | clk_rxmac clock period.<br>Bits [19:16]: nanoseconds<br>Bits [15:0]: fraction of nanosecond | This value is set to the correct clock period for the required RX MAC clock frequency. The clock period is different for 40GbE variations and 100GbE variations. | RW |
| 0xB06 | RX_PTP_PMA_LATENCY | [31:0] | Latency through the RX PMA. This is the delay from the rx_serial pin to the RX MAC input.<br>The IP core sets this register to a value that is sufficiently accurate in most cases. Altera recommends that you modify this value with extreme caution. | | RW |

**Related Information**

- **1588 Precision Time Protocol Interfaces** on page 3-39
- **PTP Transmit Functionality** on page 3-43

## LL 40-100GbE Hardware Design Example Registers

The following sections describe the registers that are included in the LL 40-100GbE hardware design example and are not a part of the LL 40-100GbE IP core.

**Related Information**

- **Low Latency 40G Ethernet Example Design User Guide**
- **Low Latency 100G Ethernet Example Design User Guide**

### Packet Client Registers

You can customize the LL 40-100GbE hardware design example by programming the packet client registers.

**Table 3-33: Packet Client Registers**

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x1000 | PKT_CL_SCRATCH | [31:0] | Scratch register available for testing. | | RW |
| 0x1001 | PKT_CL_CLNT | [31:0] | Four characters of IP block identification string "CLNT" | | RO |
| 0x1002 | PKT_CL_FEATURE | [9:0] | Feature vector to match DUT. Bits [8:3] have the value of 0 to indicate the DUT does not have the property or the value of 1 to indicate the DUT has the property.<br>• Bit [0]: Has the value of 1 to indicate the DUT targets an Arria 10 device.<br>• Bit [1]: Has the value of 0 if the DUT is a LL 40GbE IP core; has the value of 1 if the DUT is a LL 100GbE IP core.<br>• Bit [2]: Reference clock frequency. Has the value 0 for 322 MHz; has the value of 1 for 644 MHz.<br>• Bit [3]: Indicates whether the DUT is a LL 40-GBASE KR4 IP core.<br>• Bit [4]: Inidcates whether the DUT is a CAUI-4 IP core.<br>• Bit [5]: Indicates whether the DUT includes PTP support.<br>• Bit [6]: Indicates whether the DUT includes pause support<br>• Bit [7]: Indicates whether the DUT provides local fault signaling.<br>• Bit [8]: Indicates whether the DUT has **Use external MAC TX PLL** turned on. Must have the value of 0.<br>• Bit [9]: Value 0 if the DUT has a custom streaming client interface; value 1 if the DUT has an Avalon-ST client interface. Must have the value of 1. | | RO |
| 0x1006 | PKT_CL_TSD | [7:0] | Arria 10 device temperature sensor diode readout in Fahrenheit. | | RO |

| Addr | Name | Bit | Description | HW Reset Value | Access |
|---|---|---|---|---|---|
| 0x1010 | PKT_GEN_TX_CTRL | [3:0] | • Bit [0]: Reserved.<br>• Bit [1]: Packet generator disable bit. set this bit to the value of 1 to turn off the packet generator, and reset it to the value of 0 to turn on the packet generator.<br>• Bit [2]: Reserved.<br>• Bit [3]: Has the value of 1 if the IP core is in MAC loopback mode; has the value of 0 if the packet client uses the packet generator. | 4'b0101 | RW |
| 0x1015 | PKT_CL_LOOPBACK_FIFO_ERR_CLR | [2:0] | Reports MAC loopback errors.<br>• Bit [0]: FIFO underflow. Has the value of 1 if the FIFO has underflowed. This bit is sticky. Has the value of 0 if the FIFO has not underflowed.<br>• Bit [1]: FIFO overflow. Has the value of if the FIFO has overflowed. This bit is sticky. Has the value of 0 if the FIFO has not overflowed.<br>• Bit [2]: Assert this bit to clear bits [0] and [1]. | 3'b0 | RO |
| 0x1016 | PKT_CL_LOOPBACK_RESET | [0] | MAC loopback reset. Set to the value of 1 to reset the design example MAC loopback. | 1'b0 | RW |

# Ethernet Glossary

**Table 3-34: Ethernet Glossary**

Provides definitions for some terms associated with the Ethernet protocol.

| Term | Definition |
|---|---|
| BIP | Bit Interleaved Parity. A diagonal parity field which is carried in the periodic alignment markers on each virtual lane, allowing isolation of a bit error to a physical channel. |
| CAUI | 100 gigabit attachment unit interface. (C is the symbol in Roman Numerals for 100). This is an electrical interface that which is based on a 10-lane interface with a bandwidth of 10 Gbps per lane. (In this implementation, the PMA multiplexes the 20 PCS lanes into 10 physical lanes. |

| Term | Definition |
|---|---|
| CGMII | 100 gigabit media independent interface. (C is the symbol in Roman Numerals for 100). This is the byte-oriented interface protocol that connects the PCS and MAC. |
| DIC | Deficit Idle Counter. A rule for inserting and deleting idles as necessary to maintain the average IPG. The alternative is to always insert idles which could lead to reduced bandwidth. |
| FCS | Frame Check Sequence. A CRC-32 with bit reordering and inversion. |
| Frame | Ethernet formatted packet. A frame consists of a start delimiter byte, a 7 byte preamble, variable length data, 4-byte FCS, and an end delimiter byte. |
| IPG | Inter Packet Gap. Includes the end of frame delimiter and subsequent IDLE bytes up to, but not including the next start of frame delimiter. The protocol requires an average gap of 12 bytes. |
| MAC | Media Access Control. Formats a user packet stream into proper Ethernet frames for delivery to the PCS. The MAC generates the FCS and checks and maintains the IPG. |
| MII | Media Independent Interface. The byte-oriented protocol used by the PCS. Sometimes distinguished with roman numerals, XGMII (10), XLGMII (40), CGMII (100). |
| Octet | Byte. Note that Ethernet specifications primarily use least significant bit first ordering which is opposite from the default behavior of most contemporary CAD tools. |
| PCS | Physical Coding Sublayer. Presents the underlying hardware as a byte-oriented communication channel. |
| XLAUI | 40 gigabit attachment unit interface. (XL is the symbol in Roman Numerals for 40). This is an electrical interface that which is based on a 4-lane interface with a bandwidth of 10 Gbps per lane. |
| XLGMII | 40 gigabit media independent interface. (XL is the symbol in Roman Numerals for 40). This is the byte-oriented interface protocol that connects the PCS and MAC. |

# Debugging the Link 4

2017.12.28

UG-01172

Subscribe Send Feedback

Begin debugging your link at the most basic level, with word lock. Then, consider higher level issues.

The following steps should help you identify and resolve common problems that occur when bringing up a IP core link:

1. Establish word lock—The RX lanes should be able to achieve word lock even in the presence of extreme bit error rates. If unable to achieve word lock, check the transceiver clocking and data rate configuration. Check for cabling errors such as the reversal of the TX and RX lanes. Check the clock frequency monitors in the Control and Status registers.

   To check for word lock: Clear the `FRM_ERR` register by writing the value of 1 followed by another write of 0 to the `SCLR_FRM_ERR` register at offset 0x324.Then read the `FRM_ERR` register at offset 0x323. If the value is zero, the core has word lock. If non-zero the status is indeterminate
2. When having problems with word lock, check the `EIO_FREQ_LOCK` register at address 0x321. The values of the this register define the status of the recovered clock. In normal operation, all the bits should be asserted. A non-asserted (value-0) or toggling logic value on indicates a clock recovery problem. Clock recovery difficulties are typically caused by the following problems:
   - A bit error rate (BER)
   - Failure to establish the link
   - Incorrect clock inputs to the IP core
3. Check the PMA FIFO levels by selecting appropriate bits in the `EIO_FLAG_SEL` register and reading the values in `EIO_FLAGS` register. During normal operation, the TX and RX FIFOs should be nominally filled. Observing a the TX FIFO is either empty or full typically indicates a problem with clock frequencies. The RX FIFO should never be full, although an empty RX FIFO can be tolerated.
4. Establish lane integrity—When operating properly, the lanes should not experience bit errors at a rate greater than roughly one per hour per day. Bit errors within data packets are identified as FCS errors. Bit errors in control information, including IDLE frames, generally cause errors in XL/CGMII decoding.
5. Verify packet traffic—The Ethernet protocol includes automatic lane reordering so the higher levels should follow the PCS. If the PCS is locked, but higher level traffic is corrupted, there may be a problem with the remote transmitter virtual lane tags.
6. Tuning—You can adjust analog parameters to minimize any remaining bit error rate. IDLE traffic is representative for analog purposes.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

In addition, your IP core can experience loss of signal on the Ethernet link after it is established. In this case, the TX functionality is unaffected, but the RX functionality is disrupted. The following symptoms indicate a loss of signal on the Ethernet link:

- The IP core deasserts the `rx_pcs_ready` signal, indicating the IP core has lost alignment marker lock.
- The IP core deasserts the RX PCS fully aligned status bit (bit [0]) of the `RX_PCS_FULLY_ALIGNED_S` register at offset 0x326. This change is linked to the change in value of the `rx_pcs_ready` signal.
- If **Enable link fault generation** is turned on, the IP core sets `local_fault_status` to the value of 1.
- The IP core asserts the `Local Fault Status` bit (bit [0]) of the `Link_Fault` register at offset 0x308. This change is linked to the change in value of the `local_fault_status` signal.
- The IP core triggers the RX digital reset process by asserting `soft_rxp_rst`.

**Related Information**

**Arria 10 Transceiver PHY User Guide**
For information about the analog parameters for Arria 10 devices.

# Creating a SignalTap II Debug File to Match Your Design Hierarchy

For Arria 10 devices, the Quartus Prime Standard Edition software generates two files, `build_stp.tcl` and `<ip_core_name>.xml`. You can use these files to generate a SignalTap® II file with probe points matching your design hierarchy.

The Quartus Prime software stores these files in the *IP core Diretory*`/synth/debug/stp/` directory.

## Before you begin

Synthesize your design using the Quartus Prime software.

1. To open the Tcl console, click **View** > **Utility Windows** > **Tcl Console**.
2. Type the following command in the Tcl console:
   `source <IP core Directory>/synth/debug/stp/build_stp.tcl`
3. To generate the STP file, type the following command:
   `main -stp_file <output stp file name>.stp -xml_file <input xml_file name>.xml -mode build`
4. To add this SignalTap II file (**.stp**) to your project, select **Project** > **Add/Remove Files in Project**. Then, compile your design.
5. To program the FPGA, click **Tools** > **Programmer**.
6. To start the SignalTap II Logic Analyzer, click **Quartus Prime** > **Tools** > **SignalTap II Logic Analyzer**.
   The software generation script may not assign the SignalTap II acquisition clock in `<output stp file name>.stp`. Consequently, the Quartus Prime software automatically creates a clock pin called `auto_stp_external_clock`. You may need to manually substitute the appropriate clock signal as the SignalTap II sampling clock for each STP instance.
7. Recompile your design.
8. To observe the state of your IP core, click **Run Analysis**.
   You may see signals or SignalTap II instances that are red, indicating they are not available in your design. In most cases, you can safely ignore these signals and instances.They are present because software generates wider buses and some instances that your design does not include.

# Arria 10 10GBASE-KR Registers

# A

2017.12.28

UG-01172

Subscribe Send Feedback

This appendix duplicates the 10GBASE-KR PHY register listings from the **Arria 10 Transceiver PHY User Guide**. Altera provides this appendix as a convenience to make the full LL 40GBASE-KR4 register information available in the LL 40-100GbE IP core user guide. While Altera makes every attempt to keep the information in the appendix up-to-date, the most up-to-date information is always found in the **Arria 10 Transceiver PHY User Guide**, and the appendix is not guaranteed to be up-to-date at any particular time.

Most LL 40GBASE-KR4 registers are 10GBASE-KR PHY registers of the Arria 10 10GBASE-KR PHY IP core, documented in the **Arria 10 Transceiver PHY User Guide** and duplicated, with a potential time lag for updates, in this appendix. The register offsets differ by 0x400 in the 40GBASE-KR4 variations of the 40-100GbE IP core. The LL 40GBASE-KR4 variations of the LL 40-100GbE IP core have additional LL 40GBASE-KR4 related registers and register fields.

**LL 40GBASE-KR4 Registers** documents the differences between the 10GBASE-KR PHY register definitions in the *10GBASE-KR PHY Register Defintions* section of the *10GBASE-KR PHY IP Core* section in the *Arria 10 Transceiver PHY User Guide* and the 40GBASE-KR4 registers of the Low Latency 40-100GbE IP core. All Arria 10 10GBASE-KR PHY registers and register fields not listed in **LL 40GBASE-KR4 Registers** are available in the 40GBASE-KR4 variations of the LL 40-100GbE IP core.

Where the *Arria 10 Transceiver PHY User Guide* and this appendix list 10GBASE-R, substitute 40GBASE-KR4 with auto-negotiation and link training both turned off, and where they list 10GBASE-KR (except in the description of 0x4CB[24:0]), substitute 40GBASE-KR4. Where a register field description in the *Altera Transceiver PHY IP Core User Guide* and this appendix refers to link training or FEC in the single-lane 10GBASE-KR PHY IP core, substitute link training or FEC on Lane 0 of the LL 40GBASE-KR4 IP core variation.

**Related Information**

**Arria 10 Transceiver PHY User Guide**

The 40GBASE-KR4 variations of the LL 40-100GbE IP core use the 10GBASE-KR PHY IP core PHY registers at internal offsets 0x4B0–0x4FF (at IP core register map offsets 0xB0–0xFF), in addition to the registers listed in **LL 40GBASE-KR4 Registers**. Information about this PHY IP core, including up-to-date register descriptions, is available in the *10GBASE-KR PHY IP Core* section of the *Arria 10 Transceiver PHY User Guide*..

## 10GBASE-KR PHY Register Definitions

The Avalon-MM slave interface signals provide access to the control and status registers.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

The following table specifies the control and status registers that you can access over the Avalon-MM PHY management interface. A single address space provides access to all registers.

**Note:** Unless otherwise indicated, the default value of all registers is 0.

**Note:** Writing to reserved or undefined register addresses may have undefined side effects.

**Table A-1: 10GBASE-KR Register Definitions**

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4B0 | 0 | RW | `Reset SEQ` | When set to 1, resets the 10GBASE-KR sequencer (auto rate detect logic), initiates a PCS reconfiguration, and may restart Auto-Negotiation, Link Training or both if AN and LT are enabled (10GBASE-KR mode). SEQ Force Mode[2:0] forces these modes. This reset self clears. |
| | 1 | RW | `Disable AN Timer` | Auto-Negotiation disable timer. If disabled ( `Disable AN Timer = 1`) , AN may get stuck and require software support to remove the ABILITY_DETECT capability if the link partner does not include this feature. In addition, software may have to take the link out of loopback mode if the link is stuck in the ACKNOWLEDGE_DETECT state. To enable this timer set `Disable AN Timer` = 0. |
| | 2 | RW | `Disable LF Timer` | When set to 1, disables the Link Fail timer. When set to 0, the Link Fault timer is enabled. |
| | 3 | RW | `fail_lt_if_ber` | When set to 1, the last LT measurement is a non-zero number. Treat this as a failed run. 0 = normal. |
| | 7:4 | RW | `SEQ Force Mode[3:0]` | Forces the sequencer to a specific protocol. Must write the `Reset SEQ` bit to 1 for the Force to take effect. The following encodings are defined:<br>• 0000: No force<br>• 0001: GigE<br>• 0010: XAUI<br>• 0100: 10GBASE-R<br>• 0101: 10GBASE-KR<br>• 1100: 10GBASE-KR FEC |
| | 8 | RW | `Enable Arria 10 Calibration` | When set to 1, it enables the Arria 10 HSSI reconfiguration calibration as part of the PCS dynamic reconfiguration. 0 skips the calibration when the PCS is reconfigured. |
| | 11:9 | RW | Reserved | — |
| | 12 | RW | `LT failure response` | When set to 1, LT failure causes the PHY to go into data mode. When set to 0, LT failure restarts auto-negotiation (if enabled). If auto-negotiation is not enabled, the PHY will restart LT. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4B0 | 16 | RW | KR FEC enable 171.0 | When set to 1, FEC is enabled. When set to 0, FEC is disabled. Resets to the CAPABLE_FEC parameter value. |
| | 17 | RW | KR FEC enable err ind 171.1 | When set to 1, KR PHY FEC decoding errors are signaled to the PCS. When set to 0, FEC errors are not signaled to the PCS. See *Clause 74.8.3 of IEEE 802.3ap-2007* for details. |
| | 18 | RW | KR FEC request | When set to 1, enables the FEC request. When this bit changes, you must assert the Reset SEQ bit (0x4B0[0]) to renegotiate with the new value. When set to 0, disables the FEC request. |
| 0x4B1 | 0 | R | SEQ Link Ready | When asserted, the sequencer is indicating that the link is ready. |
| | 1 | R | SEQ AN timeout | When asserted, the sequencer has had an Auto Negotiation timeout. This bit is latched and is reset when the sequencer restarts Auto Negotiation. |
| | 2 | R | SEQ LT timeout | When set, indicates that the Sequencer has had a timeout. |
| | 13:8 | R | SEQ Reconfig Mode[5:0] | Specifies the Sequencer mode for PCS reconfiguration. The following modes are defined:<br>• Bit 8, mode[0]: AN mode<br>• Bit 9, mode[1]: LT Mode<br>• Bit 10, mode[2]: 10G data mode<br>• Bit 11, mode[3]: Gige data mode<br>• Bit 12, mode[4]: Reserved for XAUI<br>• Bit 13, mode[5]: 10G FEC mode |
| | 16 | R | KR FEC ability 170.0 | When set to 1, indicates that the 10GBASE-KR PHY supports FEC. Set as parameter SYNTH_FEC. For more information, refer to *Clause 45.2.1.84 of IEEE 802.3ap-2007.* |
| | 17 | R | KR FEC err ind ability 170.0 | When set to 1, indicates that the 10GBASE-KR PHY is capable of reporting FEC decoding errors to the PCS. For more information, refer to *Clause 74.8.3 of IEEE 802.3ap-2007.* |
| 0x4B2 | 0:10 | — | Reserved | — |
| | 11 | RW | KR FEC TX Error Insert | Writing a 1 inserts one error pulse into the TX FEC depending on the Transcoder and Burst error settings. This bit self clears. |
| | 31:12 | — | Reserved | — |
| 0x4B5 to 0x4BF | | | Reserved for 40G KR | Intentionally left empty for address compatibility with 40G MAC + PHY KR solutions. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4C0 | 0 | RW | AN enable | When set to 1, enables Auto Negotiation function. The default value is 1. For additional information, refer to 7.0.12 in Clause 73.8 Management Register Requirements, of *IEEE 802.3ap-2007*. |
| | 1 | RW | AN base pages ctrl | When set to 1, the user base pages are enabled. You can send any arbitrary data via the user base page low/high bits. When set to 0, the user base pages are disabled and the state machine generates the base pages to send. |
| | 2 | RW | AN next pages ctrl | When set to 1, the user next pages are enabled. You can send any arbitrary data via the user next page low/high bits. When set to 0, the user next pages are disabled. The state machine generates the null message to send as next pages. |
| | 3 | RW | Local device remote fault | When set to 1, the local device signals Remote Faults in the Auto Negotiation pages. When set to 0, a fault has not occurred. |
| | 4 | RW | Force TX nonce value | When set to 1, forces the TX nonce value to support some UNH testing modes. When set to 0, this is normal operation. |
| | 5 | RW | Override AN Parameters Enable | When set to 1, overrides the AN_TECH, AN_FEC, and AN_PAUSE parameters and uses the bits in 0xC3 instead. You must reset the Sequencer to reconfigure and restart into Auto Negotiation mode. When set to 0, this is normal operation and is used with 0x4B0 bit 0 and 0x4C3 bits[30:16]. |
| 0x4C1 | 0 | RW | Reset AN | When set to 1, resets all the 10GBASE-KR Auto Negotiation state machines. This bit is self-clearing. |
| | 4 | RW | Restart AN TX SM | When set to 1, restarts the 10GBASE-KR TX state machine. This bit self clears. This bit is active only when the TX state machine is in the Auto Negotiation state. For more information, refer to 7.0.9 in Clause 73.8 Management Register Requirements of *IEEE 802.3ap-2007*. |
| | 8 | RW | AN Next Page | When asserted, new next page info is ready to send. The data is in the XNP TX registers. When 0, the TX interface sends null pages. This bit self clears. Next Page (NP) is encoded in bit D15 of Link Codeword. For more information, refer to Clause 73.6.9 and 7.16.15 of Clause 45.2.7.6 of *IEEE 802.3ap-2007*. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4C2 | 1 | RO | AN page received | When set to 1, a page has been received. When 0, a page has not been received. The current value clears when the register is read. For more information, refer to 7.1.6 in Clause 73.8 of *IEEE 802.3ap-2007*. |
| | 2 | RO | AN Complete | When asserted, Auto-Negotiation has completed. When 0, Auto Negotiation is in progress. For more information, refer to 7.1.5 in Clause 73.8 of *IEEE 802.3ap-2007*. |
| | 3 | RO | AN ADV Remote Fault | When set to 1, fault information has been sent to the link partner. When 0, a fault has not occurred. The current value clears when the register is read. Remote Fault (RF) is encoded in bit D13 of the base Link Codeword. For more information, refer to Clause 73.6.7 of and 7.16.13 of *IEEE 802.3ap-2007*. |
| | 4 | RO | AN RX SM Idle | When set to 1, the Auto-Negotiation state machine is in the idle state. Incoming data is not Clause 73 compatible. When 0, the Auto-Negotiation is in progress. |
| | 5 | RO | AN Ability | When set to 1, the transceiver PHY is able to perform Auto Negotiation. When set to 0, the transceiver PHY i s not able to perform Auto Negotiation. If your variant includes Auto Negotiation, this bit is tied to 1. For more information, refer to 7.1.3 and 7.48.0 of Clause 45 of *IEEE 802.3ap-2007*. |
| | 6 | RO | AN Status | When set to 1, link is up. When 0, the link is down. The current value clears when the register is read. For more information, refer to 7.1.2 of Clause 45 of *IEEE 802.3ap-2007*. |
| | 7 | RO | LP AN Ability | When set to 1, the link partner is able to perform Auto Negotiation. When 0, the link partner is not able to perform Auto-Negotiation. For more information, refer to 7.1.0 of Clause 45 of *IEEE 802.3ap-2007*. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4C2 | 8 | RO | FEC negotiated – enable FEC from SEQ | When set to 1, PHY is negotiated to perform FEC. When set to 0, PHY is not negotiated to perform FEC. |
| | 9 | RO | Seq AN Failure | When set to 1, a sequencer Auto Negotiation failure has been detected. When set to 0, an Auto Negotiation failure has not been detected. |
| | 17:12 | RO | KR AN Link Ready[5:0] | Provides a one-hot encoding of an_receive_idle = true and link status for the supported link as described in Clause 73.10.1. The following encodings are defined:<br>• 6'b000000: 1000BASE-KX<br>• 6'b000001: 10GBASE-KX4<br>• 6'b000100: 10GBASE-KR<br>• 6'b001000: 40GBASE-KR4<br>• 6'b010000: 40GBASE-CR4<br>• 6'b100000: 100GBASE-CR10 |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4C3 | 15:0 | RW | User base page low | The Auto Negotiation TX state machine uses these bits if the Auto Negotiation base pages ctrl bit is set. The following bits are defined:<br>• [15]: Next page bit<br>• [14]: ACK which is controlled by the SM<br>• [13]: Remote Fault bit<br>• [12:10]: Pause bits<br>• [9:5]: Echoed nonce which are set by the state machine<br>• [4:0]: Selector<br>Bit 49, the PRBS bit, is generated by the Auto Negotiation TX state machine. |
| | 21:16 | RW | Override AN_TECH[5:0] | AN_TECH value with which to override the current value. The following bits are defined:<br>• Bit-16 = AN_TECH[0]= 1000BASE-KX<br>• Bit-17 = AN_TECH[1] = XAUI<br>• Bit-18 = AN_TECH[2] = 10GBASE-KR<br>• Bit-19 = AN_TECH[3] = 40G<br>• Bit-20 = AN_TECH[4] = CR-4<br>• Bit-21 = AN_TECH[5] = 100G<br>You must set 0x4C0 bit-5 for this to take effect . |
| | 25:24 | RW | Override AN_FEC[1:0] | AN_FEC value with which to override the current value. The following bits are defined:<br>• Bit-24 = AN_ FEC [0] = Capability<br>• Bit-25 = AN_ FEC [1] = Request<br>You must set 0x4C0 bit-5 for this to take effect. |
| | 30:28 | RW | Override AN_PAUSE[2:0] | AN_PAUSE value with which to override the current value. The following bits are defined:<br>• Bit-28 = AN_PAUSE [0] = Pause Ability<br>• Bit-29 = AN_PAUSE [1] = Asymmetric Direction<br>• Bit-30 = AN_PAUSE [2] = Reserved<br>You must set 0x4C0 bit-5 for this to take effect. |
| 0x4C4 | 31:0 | RW | User base page high | The Auto Negotiation TX state machine uses these bits if the Auto Negotiation base pages ctrl bit is set. The following bits are defined:<br>• [29:5]: Correspond to page bits 45:21 which are the technology ability.<br>• [4:0]: Correspond to bits 20:16 which are TX nonce bits.<br>Bit 49, the PRBS bit, is generated by the Auto Negotiation TX state machine. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4C5 | 15:0 | RW | User Next page low | The Auto Negotiation TX state machine uses these bits if the AN Next Page control bit is set. The following bits are defined:<br>• [15]: next page bit<br>• [14]: ACK controlled by the state machine<br>• [13]: Message Page (MP) bit<br>• [12]: ACK2 bit<br>• [11]: Toggle bit<br>For more information, refer to Clause 73.7.7.1 Next Page encodings of *IEEE 802.3ap-2007*. Bit 49, the PRBS bit, is generated by the Auto-Negotiation TX state machine. |
| 0x4C6 | 31:0 | RW | User Next page high | The Auto Negotiation TX state machine uses these bits if the Auto Negotiation next pages ctrl bit is set. Bits [31:0] correspond to page bits [47:16]. Bit 49, the PRBS bit, is generated by the Auto Negotiation TX state machine. |
| 0x4C7 | 15:0 | RO | LP base page low | The AN RX state machine receives these bits from the link partner. The following bits are defined:<br>• [15] Next page bit<br>• [14] ACK which is controlled by the state machine<br>• [13] RF bit<br>• [12:10] Pause bits<br>• [9:5] Echoed Nonce which are set by the state machine<br>• [4:0] Selector |
| 0x4C8 | 31:0 | RO | LP base page high | The AN RX state machine receives these bits from the link partner. The following bits are defined:<br>• [31:30]: Reserved<br>• [29:5]: Correspond to page bits [45:21] which are the technology ability<br>• [4:0]: Correspond to bits [20:16] which are TX Nonce bits |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4C9 | 15:0 | RO | LP Next page low | The AN RX state machine receives these bits from the link partner. The following bits are defined:<br>• [15]: Next page bit<br>• [14]: ACK which is controlled by the state machine<br>• [13]: MP bit<br>• [12] ACK2 bit<br>• [11] Toggle bit<br>For more information, refer to Clause 73.7.7.1 Next Page encodings of IEEE 802.3ap-2007. |
| 0x4CA | 31:0 | RO | LP Next page high | The AN RX state machine receives these bits from the link partner. Bits [31:0] correspond to page bits [47:16] |
| 0x4CB | 24:0 | RO | AN LP ADV Tech_A[24:0] | Received technology ability field bits of Clause 73 Auto Negotiation. The 10GBASE-KR PHY supports A0 and A2. The following protocols are defined:<br>• A0 1000BASE-KX<br>• A1 10GBASE-KX4<br>• A2 10GBASE-KR<br>• A3 40GBASE-KR4<br>• A4 40GBASE-CR4<br>• A5 100GBASE-CR10<br>• A24:6 are reserved<br>For more information, refer to Clause 73.6.4 and AN LP base page ability registers (7.19-7.21) of Clause 45 of *IEEE 802.3ap-2007*. |
| | 26:25 | RO | AN LP ADV FEC_F[1:0] | Received FEC ability bits FEC (F0:F1) is encoded in bits D46:D47 of the base Link Codeword. F0 is FEC ability. F1 is FEC requested. See Clause 73.6.5 of *IEEE 802.3ap-2007* for details. |
| | 27 | RO | AN LP ADV Remote Fault | Received Remote Fault (RF) ability bits. RF is encoded in bit D13 of the base link codeword in Clause 73 AN. For more information, refer to Clause 73.6.7 of *IEEE 802.3ap-2007*. |
| | 30:28 | RO | AN LP ADV Pause Ability_C[2:0] | Received pause ability bits. Pause (C0:C1) is encoded in bits D11:D10 of the base link codeword in Clause 73 AN as follows:<br>• C0 is the same as PAUSE as defined in Annex 28B<br>• C1 is the same as ASM_DIR as defined in Annex 28B<br>• C2 is reserved |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4D0 | 0 | RW | Link Training enable | When 1, enables the 10GBASE-KR start-up protocol. When 0, disables the 10GBASE-KR start-up protocol. The default value is 1. For more information, refer to Clause 72.6.10.3.1 and 10GBASE-KR PMD control register bit (1.150.1) of *IEEE 802.3ap-2007*. |
| | 1 | RW | dis_max_wait_tmr | When set to 1, disables the LT max_wait_timer. Used for characterization mode when setting much longer BER timer values. The default value is 0. |
| | 2 | RW | Reserved | Reserved |
| | 3 | RW | Reserved | Reserved |
| | 7:4 | RW | main_step_cnt [3:0] | Specifies the number of equalization steps for each main tap update. There are about 20 settings for the internal algorithm to test. The valid range is 1-15. The default value is 4'b0001. |
| | 11:8 | RW | prepost_step_cnt [3:0] | Specifies the number of equalization steps for each pre- and post-tap update. From 16-31 steps are possible. The default value is 4'b0001. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4D0 | 14:12 | RW | equal_cnt [2:0] | Adds hysteresis to the error count to avoid local minimums. The following values are defined:<br>• 000 = 0<br>• 001 = 2<br>• 010 = 4<br>• 011 = 8<br>• 100 = 16<br>• 101 = 32<br>• 110 = 64<br>• 111 = 128<br>The default value is 101. |
| | 15 | RW | disable Initialize PMA on max_wait_ timeout | When set to 1, PMA values (VOD, Pre-tap, Post-tap) are not initialized upon entry into the Training_Failure state. This happens when max_ wait_timer_done, which sets training_failure = true (reg 0xD2 bit 3). Used for UNH testing. When set to 0, PMA values are initialized upon entry into Training_Failure state. Refer to Figure 72-5 of *IEEE 802.3ap-2007* for more details. The default value is 0. |
| | 16 | RW | Ovride LP Coef enable | When set to 1, overrides the link partner's equalization coefficients; software changes the update commands sent to the link partner TX equalizer coefficients. When set to 0, uses the Link Training logic to determine the link partner coefficients. Used with 0x4D1 bit-4 and 0x4D4 bits[7:0]. The default value is 0. |
| | 17 | RW | Ovride Local RX Coef enable | When set to 1, overrides the local device equalization coefficients generation protocol. When set, the software changes the local TX equalizer coefficients. When set to 0, uses the update command received from the link partner to determine local device coefficients. Used with 0x4D1 bit-8 and 0x4D4 bits[23:16]. The default value is 0. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4D0 | 18 | RW | VOD Training Enable | Defines whether or not to skip adjustment of the link partner's VOD (main tap) during link training. The following values are defined:<br>• 1 = Exercise VOD (main tap) adjustment during link training<br>• 0 = Skip VOD (main tap) adjustment during link training<br>The default value is 0. |
| | 19 | RW | Bypass DFE | Defines whether or not Decision Feedback Equalization (DFE) is enabled at the end of link training. The following values are defined:<br>• 1 = Bypass continuous adaptive DFE at the end of link training<br>• 0 = Enable continuous adaptive DFE at the end of link training<br>The default value for simulation is 1. The default value for hardware is 0. |
| | 21:20 | RW | dfe_freeze_mode | Defines the behavior of DFE taps at the end of link training<br>• 00 = do not freeze any DFE taps<br>• 01 = Freeze all DFE taps<br>• 10 = reserved<br>• 11 = reserved<br>The default value is 01.<br>**Note:** These bits will be effective only when bit [19] is set to 0. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4D0 | 22 | RW | adp_ctle_vga_mode | Defines whether or not CTLE/VGA adaptation is in adaptive or manual mode. The following values are defined:<br>• 0 = CTLE sweep before start of TX-EQ during link training.<br>• 1 = manual CTLE mode. Link training algorithm sets fixed CTLE value, as specified in bits [28:24]. The default value is 1 for simulation. .<br>The default value is 0 for hardware. |
| | 28:24 | RW | Manual CTLE | Defines the CTLE value used by the link training algorithm when in manual CTLE mode. These bits are only effective when 0x4D0[22] is set to 1.<br>The default value is 1. |
| | 31:29 | RW | Manual VGA | Defines the VGA value used by the link training algorithm when in manual VGA mode. These bits are only effective when 0x4D0[22] is set to 1.<br>The default value is 4 for simulation. The default value is 7 for hardware. |
| 0x4D1 | 0 | RW | Restart Link training | When set to 1, resets the 10GBASE-KR start-up protocol. When set to 0, continues normal operation. This bit self clears. For more information, refer to the state variable mr_restart_training as defined in Clause 72.6.10.3.1 and 10GBASE-KR PMD control register bit (1.150.0) *IEEE 802.3ap-2007*. |
| | 4 | RW | Updated TX Coef new | When set to 1, there are new link partner coefficients available to send. The LT logic starts sending the new values set in 0x4D4 bits[7:0] to the remote device. When set to 0, continues normal operation. This bit self clears. Must enable this override in 0x4D0 bit16. |
| | 8 | RW | Updated RX coef new | When set to 1, new local device coefficients are available. The LT logic changes the local TX equalizer coefficients as specified in 0x4D4 bits[23:16]. When set to 0, continues normal operation. This bit self clears. Must enable the override in 0x4D0 bit17. |
| | 21:20 | RW | Reserved | Reserved |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4D2 | 0 | RO | Link Trained - Receiver status | When set to 1, the receiver is trained and is ready to receive data. When set to 0, receiver training is in progress. For more information, refer to the state variable rx_trained as defined in Clause 72.6.10.3.1 of *IEEE 802.3ap-2007*. |
| | 1 | RO | Link Training Frame lock | When set to 1, the training frame delineation has been detected. When set to 0, the training frame delineation has not been detected. For more information, refer to the state variable frame_lock as defined in Clause 72.6.10.3.1 of *IEEE 802.3ap-2007*. |
| | 2 | RO | Link Training Start-up protocol status | When set to 1, the start-up protocol is in progress. When set to 0, start-up protocol has completed. For more information, refer to the state training as defined in Clause 72.6.10.3.1 of *IEEE 802.3ap-2007*. |
| | 3 | RO | Link Training failure | When set to 1, a training failure has been detected. When set to 0, a training failure has not been detected For more information, refer to the state variable training_failure as defined in Clause 72.6.10.3.1 of IEEE 802.3ap-2007. |
| | 4 | RO | Link Training Error | When set to 1, excessive errors occurred during Link Training. When set to 0, the BER is acceptable. |
| | 5 | RO | Link Training Frame lock Error | When set to 1, indicates a frame lock was lost during Link Training. If the tap settings specified by the fields of 0x4D5 are the same as the initial parameter value, the frame lock error was unrecoverable. |
| | 6 | RO | RXEQ Frame Lock Loss | Frame lock not detected at some point during RXEQ, possibly triggering conditional RXEQ mode. |
| | 7 | RO | CTLE Fine-grained Tuning Error | Could not determine the best CTLE due to maximum BER limit at each step in the Fine-grained Tuning mode. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4D3 | 9:0 | RW | ber_time_frames | Specifies the number of training frames to examine for bit errors on the link for each step of the equalization settings. Used only when ber_time_k_frames is 0.The following values are defined:<br>• A value of 2 is about $10^3$ bytes<br>• A value of 20 is about $10^4$ bytes<br>• A value of 200 is about $10^5$ bytes<br>The default value for simulation is 2'b11. The default value for hardware is 0. |
| | 19:10 | RW | ber_time_k_frames | Specifies the number of thousands of training frames to examine for bit errors on the link for each step of the equalization settings. Set *ber_time_m_frames = 0* for time/bits to match the following values:<br>• A value of 3 is about $10^7$ bits = about 1.3 ms<br>• A value of 25 is about $10^8$ bits = about 11ms<br>• A value of 250 is about $10^9$ bits = about 11 0ms<br>The default value for simulation is 0. The default value for hardware is 0xF. |
| | 29:20 | RW | ber_time_m_frames | Specifies the number of millions of training frames to examine for bit errors on the link for each step of the equalization settings. Set *ber_time_k_frames = 4'd1000 = 0x43E8* for time/bits to match the following values:<br>• A value of 3 is about $10^{10}$ bits = about 1.3 seconds<br>• A value of 25 is about $10^{11}$ bits = about 11 seconds<br>• A value of 250 is about $10^{12}$ bits = about 110 seconds |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4D4 | 5:0 | RO or RW | `LD coefficient update[5:0]` | Reflects the contents of the first 16-bit word of the training frame sent from the local device control channel. Normally, the bits in this register are read-only; however, when you override training by setting the `Ovride Coef enable` control bit, these bits become writeable. The following fields are defined:<br>• [5: 4]: Coefficient (+1) update<br>  • 2'b11: Reserved<br>  • 2'b01: Increment<br>  • 2'b10: Decrement<br>  • 2'b00: Hold<br>• [3:2]: Coefficient (0) update (same encoding as [5:4])<br>• [1:0]: Coefficient (-1) update (same encoding as [5:4])<br>For more information, refer to 10G BASE-KR LD coefficient update register bits (1.154.5:0) in Clause 45.2.1.80.3 of *IEEE 802.3ap-2007*. |
| | 6 | RO or RW | `LD Initialize Coefficients` | When set to 1, requests the link partner coefficients be set to configure the TX equalizer to its INITIALIZE state. When set to 0, continues normal operation. For more information, refer to 10G BASE-KR LD coefficient update register bits (1.154.12) in Clause 45.2.1.80.3 and Clause 72.6.10.2.3.2 of *IEEE 802.3ap-2007*. |
| | 7 | RO or RW | `LD Preset Coefficients` | When set to 1, requests the link partner coefficients be set to a state where equalization is turned off. When set to 0 the link operates normally. For more information, refer to 10G BASE-KR LD coefficient update register bit (1.154.13) in Clause 45.2.1.80.3 and Clause 72.6.10.2.3.2 of *IEEE 802.3ap-2007*. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4D4 | 13:8 | RO | `LD coefficient status[5:0]` | Status report register for the contents of the second, 16-bit word of the training frame most recently sent from the local device control channel. The following fields are defined:<br>• [5:4]: Coefficient (post-tap)<br>  • 2'b11: Maximum<br>  • 2'b01: Minimum<br>  • 2'b10: Updated<br>  • 2'b00: Not updated<br>• [3:2]: Coefficient (0) (same encoding as [5:4])<br>• [1:0]: Coefficient (pre-tap) (same encoding as [5:4])<br>For more information, refer to 10G BASE-KR LD status report register bit (1.155.5:0) in Clause 45.2.1.81 of *IEEE 802.3ap-2007*. |
| | 14 | RO | `Link Training ready - LD Receiver ready` | When set to 1, the local device receiver has determined that training is complete and is prepared to receive data. When set to 0, the local device receiver is requesting that training continue. Values for the receiver ready bit are defined in Clause 72.6.10.2.4.4. For more information, refer to 10G BASE-KR LD status report register bit (1.155.15) in Clause 45.2.1.81 of *IEEE 802.3ap-2007*. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4D4 | 21:16 | RO or RW | `LP coefficient update[5:0]` | Reflects the contents of the first 16-bit word of the training frame most recently received from the control channel.<br>Normally the bits in this register are read only; however, when training is disabled by setting low the KR Training enable control bit, these bits become writeable. The following fields are defined:<br>• [5: 4]: Coefficient (+1) update<br>  • 2'b11: Reserved<br>  • 2'b01: Increment<br>  • 2'b10: Decrement<br>  • 2'b00: Hold<br>• [3:2]: Coefficient (0) update (same encoding as [5:4])<br>• [1:0]: Coefficient (-1) update (same encoding as [5:4])<br>For more information, refer to 10G BASE-KR LP coefficient update register bits (1.152.5:0) in Clause 45.2.1.78.3 of *IEEE 802.3ap-2007*. |
| | 22 | RO or RW | `LP Initialize Coefficients` | When set to 1, the local device transmit equalizer coefficients are set to the INITIALIZE state. When set to 0, normal operation continues. The function and values of the initialize bit are defined in Clause 72.6.10.2.3.2. For more information, refer to 10G BASE-KR LP coefficient update register bits (1.152.12) in Clause 45.2.1.78.3 of *IEEE 802.3ap-2007*. |
| | 23 | RO or RW | `LP Preset Coefficients` | When set to 1, the local device TX coefficients are set to a state where equalization is turned off. Preset coefficients are used. When set to 0, the local device operates normally. The function and values of the preset bit are defined in 72.6.10.2.3.1. The function and values of the initialize bit are defined in Clause 72.6.10.2.3.2. For more information, refer to 10G BASE-KR LP coefficient update register bits (1.152.13) in Clause 45.2.1.78.3 of IEEE 802.3ap-2007. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4D4 | 29:24 | RO | `LP coefficient status[5:0]` | Status report register reflects the contents of the second, 16-bit word of the training frame most recently received from the control channel: The following fields are defined:<br>• [5:4]: Coefficient (+1)<br>  • 2'b11: Maximum<br>  • 2'b01: Minimum<br>  • 2'b10: Updated<br>  • 2'b00: Not updated<br>• [3:2]: Coefficient (0) (same encoding as [5:4])<br>• n [1:0]: Coefficient (-1) (same encoding as [5:4])<br>For more information, refer to 10G BASE-KR LP status report register bits (1.153.5:0) in Clause 45.2.1.79 of *IEEE 802.3ap-2007*. |
| | 30 | RO | `LP Receiver ready` | When set to 1, the link partner receiver has determined that training is complete and is prepared to receive data. When set to 0, the link partner receiver is requesting that training continue.<br>Values for the receiver ready bit are defined in Clause 72.6.10.2.4.4. For more information, refer to 10G BASE-KR LP status report register bits (1.153.15) in Clause 45.2.1.79 of *IEEE 802.3ap-2007*. |
| 0x4D5 | 4:0 | R | `LT` $V_{OD}$ `setting` | Stores the most recent TX $V_{OD}$ setting trained by the link partner's RX based on the LT coefficient update logic driven by Clause 72. It reflects Link Partner commands to fine-tune the TX preemphasis taps. |
| | 13:8 | R | `LT Post-tap setting` | Stores the most recent TX post-tap setting trained by the link partner's RX based on the LT coefficient update logic driven by Clause 72. It reflects Link Partner commands to fine-tune the TX pre-emphasis taps. |
| | 20:16 | R | `LT Pre-tap setting` | Store the most recent TX pre-tap setting trained by the link partner's RX based on the LT coefficient update logic driven by Clause 72. It reflects Link Partner commands to fine-tune the TX pre-emphasis taps. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4D5 | 27:24 | R | RXEQ CTLE Setting | Most recent ctle_rc setting sent to the reconfig bundle during RX equalization. |
| | 29:28 | R | RXEQ CTLE Mode | Most recent ctle_mode setting sent to the reconfig bundle during RX equalization. |
| | 31:30 | R | RXEQ DFE Mode | Most recent dfe_mode setting sent tothe reconfig bundle during RX equalization. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4D6 | 4:0 | RW | LT VODMAX ovrd | Override value for the **VMAXRULE** parameter. When enabled, this value substitutes for the **VMAXRULE** to allow channel-by-channel override of the device settings. This only affects the local device TX output for the channel specified.<br>This value must be greater than the **INITMAINVAL** parameter for proper operation. Note this will also override the **PREMAINVAL** parameter value. |
| | 5 | RW | LT VODMAX ovrd Enable | When set to 1, enables the override value for the **VMAXRULE** parameter stored in the LT VODMAX ovrd register field. |
| | 12:8 | RW | LT VODMin ovrd | Override value for the **VODMINRULE** parameter. When enabled, this value substitutes for the **VMINRULE** to allow channel-by-channel override of the device settings. This override only effects the local device TX output for this channel.<br>The value to be substituted must be less than the **INITMAINVAL** parameter and greater than the **VMINRULE** parameter for proper operation. |
| | 13 | RW | LT VODMin ovrd Enable | When set to 1, enables the override value for the **VODMINRULE** parameter stored in the LT VODMin ovrd register field. |
| | 21:16 | RW | LT VPOST ovrd | Override value for the **VPOSTRULE** parameter. When enabled, this value substitutes for the **VPOSTRULE** to allow channel-by-channel override of the device settings. This override only effects the local device TX output for this channel.<br>The value to be substituted must be greater than the **INITPOSTVAL** parameter for proper operation. |
| | 22 | RW | LT VPOST ovrd Enable | When set to 1, enables the override value for the **VPOSTRULE** parameter stored in the LT VPOST ovrd register field. |
| | 28:24 | RW | LT VPre ovrd | Override value for the **VPRERULE** parameter. When enabled, this value substitutes for the **VPOSTRULE** to allow channel-by-channel override of the device settings. This override only effects the local device TX output for this channel.<br>The value to be substituted must be greater than the **INITPREVAL** parameter for proper operation. |
| | 29 | RW | LT VPre ovrd Enable | When set to 1, enables the override value for the **VPRERULE** parameter stored in the LT VPre ovrd register field. |

| Word Addr | Bit | R/W | Name | Description |
|---|---|---|---|---|
| 0x4D6 to 0x4FF | | | Reserved for 40G KR | Left empty for address compatibility with 40G MAC+PHY KR solution. |

# Differences Between Low Latency 40-100GbE IP Core and 40-100GbE IP Core v15.1

B

2017.12.28

UG-01172 Subscribe Send Feedback

The Low Latency 40-100GbE MegaCore function provides new functionality, and also sacrifices some dynamic configuration options and features to achieve low latency and lower resource utilization than the legacy 40-100GbE MegaCore function. In some cases, parameter options allow you to configure the Low Latency IP core to include or exclude features that are dynamically configurable in the legacy IP core, providing you the flexibility to achieve resource savings by removing features your design does not require.

To compare the Low Latency 40-100GbE MegaCore function to previous releases of the 40-100GbE MegaCore function, refer to the table and to the Altera documentation about the revision history of the legacy 40-100GbE MegaCore function.

### Table B-1: Major Differences Between the Low Latency 40-100GbE IP Core v15.1 and the 40-100GbE IP Core v15.1

Lists the basic differences between the Altera 40- and 100-Gbps Ethernet MAC and PHY (40-100GbE) MegaCore function, a product available through several previous Altera software releases, and the Low Latency 40- and 100-Gbps Ethernet MAC and PHY (Low Latency 40-100GbE, or LL 40-100GbE) MegaCore function in its current release.

| Property | Low Latency 40-100GbE IP Core | 40-100GbE IP Core |
|---|---|---|
| IP core installation | IP core for Arria 10 devices is included in the Altera IP Library (included in the ACDS) installation<br>IP core for Stratix V devices is available for installation and integration into your ACDS installation from the Self-Service Licensing Center for releases 14.1, 15.0, and 15.1 | IP core is included in the Altera IP Library |
| Device support | Supports Stratix V and Arria 10 device families. Altera recommends you target an Arria 10 device or use the 40-100GbE IP core instead. | Supports Stratix IV, Arria V GZ, and Stratix V device families. |
| Core options | MAC & PHY | PHY only, MAC only, or MAC & PHY |

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

| Property | Low Latency 40-100GbE IP Core | 40-100GbE IP Core |
|---|---|---|
| Duplex mode | Full duplex mode | RX-only, TX-only, or full duplex mode |
| MAC client interface | Avalon-ST interface or new, different custom streaming interface | Avalon-ST interface or custom streaming interface |
| CAUI–4 variations | Available for Arria 10 GT devices | Available for Stratix V GT devices |
| 4 × 6.25 variations | Not currently available | Available for all supported devices |
| 40GBASE-KR4 variations | Available for Arria 10 devices | Available for Stratix V devices |
| Statistics counters | RX statistics counters are configurable (turned on or off) in the parameter editor, and TX statistics counters are configurable in the parameter editor, with two separate parameters<br>Software clears all RX or all TX statistics counters at one time with a dedicated configuration register. Reading a statistics register does not affect its value. | RX and TX statistics counters are all available, or no statistics counters are available. Configurable in the parameter editor.<br>Programmable bits in `MAC_CMD_CONFIG` register determine whether reading a statistics register resets it to zero. |
| Statistics counter increment vectors | Available<br>Four additional RX statistics counter increment vectors and two new counters for number of bytes in non-errored RX frames and number of bytes in non-errored TX frames | Available |
| Synchronous Ethernet support | Available | Available |
| Preamble pass-through mode | Linked RX and TX preamble pass-through modes turned on or off in parameter editor. The RX and TX modes cannot be turned on or off individually and the mode cannot be dynamically configured. | RX and TX preamble pass-through modes individually dynamically configurable (programmable). |
| RX filtering | Programmable option to filter pause frames only | Programmable options for many RX filtering options, including destination address filtering |

| Property | Low Latency 40-100GbE IP Core | 40-100GbE IP Core |
|---|---|---|
| Maximum Ethernet frame size | Programmable maximum received frame size controls effect on the statistics increment vectors and statistics counters. | Programmable maximum received frame size controls oversized frame rejection and effect on the statistics increment vectors and statistics counters.Default maximum size is different for cut-through and store-and-forward modes. |
| Link fault signaling | Link fault signaling turned on or off in parameter editor. If link fault signaling is turned off, the relevant signals are not available.<br>If turned on, the IP core has a configurable option for *IEEE 802.3 –2012 Ethernet Standard* Clause 66 support.<br>The response in case of remote or local fault is determined by whether or not you turn on Clause 66 support. | Available. Programmable options to specify response to remote or local fault.<br>*IEEE 802.3 –2012 Ethernet Standard* Clause 66 support is not available. |
| TX FCS (CRC-32) insertion | Configurable in parameter editor. | Programmable in IP core registers. |
| Deficit idle counter (maintenance of minimum average 12-byte IPG) | Configurable in parameter editor. You can specify a minimum average IPG of 8 bytes or 12 bytes. | Always included. |
| More precise IPG control. | Available if you include the deficit idle counter. | Programmable in IP core registers. |
| RX FCS error flag alignment with EOP | Configurable in parameter editor. If alignment parameter is turned on, the RX FCS error flag is asserted in the same clock cycle with the EOP signal. If the parameter is turned off, the FCS error signal might be asserted in a later clock cycle. | If IP core detects an FCS error, RX FCS error flag is asserted in same clock cycle with the EOP signal, unless the IP core is in RX automatic pad removal mode. In RX automatic pad removal mode, the IP core might assert the two signals on different clock cycles. |
| RX automatic pad removal | Not available. | Programmable in IP core registers. |
| IEEE-1588 (PTP) support | Configurable in parameter editor. | Not available. |

| Property | Low Latency 40-100GbE IP Core | 40-100GbE IP Core |
|---|---|---|
| TX source address insertion | Not available. The IP core transmits the source address provided on the TX client interface in the source address field of the Ethernet frame. | Programmable in IP core registers. |
| Pause frame control and processing | You can request transmission of an XOFF or XON pause frame by asserting or deasserting a level input signal. You configure registers to specify the values in the pause frame.<br>For backward compatibility, you can also configure a register to request transmission of a pause frame. However, Altera recommends you make the request using the input signal instead.<br>Supports user-specified retransmission hold-off time, to specify the duration between repeat transmission of XOFF frames.<br>Pause frame address filtering of received pause frames compares the incoming pause frame destination address to the value in an IP core register.<br>Two independent enable register fields control the TX MAC processing of incoming pause frames on the Ethernet link and the response to a user pause request. A third enable register field controls the RX MAC processing of incoming pause frames on the Ethernet link. | In IP core variations without adapters, you can request transmission of an XOFF or XON pause frame by pulsing an edge-triggered input signal. Additional input signals specify the values in the pause frame. In all IP core MAC variations, you can request transmission and provide pause frame values by configuring registers.<br>Does not support retransmission hold-off time. You must control retransmission of XOFF frames in user logic. Asserting the edge-triggered input signal or setting the relevant pause request register field generates a single XOFF frame.<br>Pause frame address filtering of received pause frames compares the incoming pause frame destination address to the address of the IP core.<br>Two independent enable register fields control the IP core processing of incoming unicast and multicast pause frames on the Ethernet link. Does not support register control of IP core processing of incoming user pause requests. To avoid the IP core generating an outgoing pause frame in response to a user request, you must avoid generating user pause requests. |
| Priority-based flow control | Configurable in parameter editor. | Not available. |

| Property | Low Latency 40-100GbE IP Core | 40-100GbE IP Core |
|---|---|---|
| Reset | Single asynchronous reset signal resets the entire IP core. Additional reset signals reset the individual Avalon-MM interfaces: the control and status interface in all IP core variations, and the Arria 10 transceiver reconfiguration interface in IP core variations that target an Arria 10 device. | Five asynchronous reset signals reset individual components of the IP core. User must enforce recommended reset sequence. |
| Clocks | Input clocks for Avalon-MM interfaces and PLL reference clock.<br>PLL reference clock drives internal IP core clocks, and `clk_rxmac` and `clk_txmac` are output clocks.<br>Configurable option to provide TX MAC input clock from an external PLL or to include the PLL in the IP core. Clock signal from the external MAC PLL is a new input clock signal. | Input clocks for Avalon-MM interfaces, PLL reference clock, and RX and TX MAC (`clk_rxmac` and `clk_txmac`). |
| Stratix V transceiver dynamic reconfiguration controller | You must instantiate an external reconfiguration controller. | You must instantiate an external reconfiguration controller. |
| TX error insertion test and debug feature | Available | Not available |
| RX control frame status flags | Available | Not available |

**Related Information**

- **40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide**
  For detailed information about changes to top-level interfaces in different versions of the legacy 40-100GbE IP core, refer to the Document Revision History in the Additional Information chapter.
- **MegaCore IP Library Release Notes and Errata**
  For information about changes in different versions of the legacy 40-100GbE IP core, refer to the Product Revision History in the 40-100GbE MegaCore function chapter. The Low Latency 40-100GbE IP core v14.0 is an extended IP core. Extended IP cores are not included in this document. The Low Latency 40-100GbE IP core is documented in the Altera IP Release Notes starting in version 14.0 Arria 10 Edition.

# Additional Information C

2017.12.28

UG-01172 Subscribe Send Feedback

## Low Latency 40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide Archives

If an IP core version is not listed, the user guide for the previous IP core version applies.

| IP Core Version | User Guide |
|---|---|
| 15.1 | Low Latency 40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide 15.1 |
| 15.0 | Low Latency 40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide 15.0 |
| 14.1 | Low Latency 40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide 14.1 |

## Low Latency 40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide Revision History

**Table C-1: Document Revision History**

| Date | Compatible ACDS Version | Changes |
|---|---|---|
| 2017.12.28 | 16.0 | Added note to clarify that this user guide documents the 16.0 version of the IP core, and that as of 2017.12.28, the 16.0 version of the Stratix V LL 40-100GbE IP core is the most recent Stratix V LL 40-100GbE IP core available in the Self-Service Licensing Center. The current versions of the Arria 10 40GbE and 100GbE IP cores are documented separately in the Low Latency 40-Gbps Ethernet IP Core User Guide and the Low Latency 100-Gbps Ethernet IP Core User Guide, respectively. |

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

| Date | Compatible ACDS Version | Changes |
|---|---|---|
| 2016.05.02 | 16.0 | • Removed the On Die Instrumentation (ODI) support.<br>• Removed the following parameters:<br>  • **Enable ODI (On Die instrumentation) acceleration logic**<br>  • **Enable KR4 Reconfiguration**<br>• Corrected the 10GBASE-KR PHY Register Definitions.<br>• Added the following topics:<br>  • **Creating a SignalTap II Debug File to Match Your Design Hierarchy**<br>  • **Low Latency 40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide Archives**<br>• Removed chapters Arria 10 Low Latency 40-100GbE Quick Start Guide and Low Latency 40-100GbE IP Core Design Example, instead refer to:<br>  • **Low Latency 40G Ethernet Example Design User Guide**<br>  • **Low Latency 100G Ethernet Example Design User Guide**<br>• Corrected mentions of TCP checksum to UDP checksum. Refer to **PTP Transmit Functionality** on page 3-43 and **1588 PTP Interface Signals** on page 3-47.<br>• Fixed assorted typos and formatting issues. |
| 2015.11.12 | 15.1 | • Corrected descriptions of 1588 PTP feature and signals in **PTP Transmit Functionality** on page 3-43, **PTP Timestamp and TOD Formats** on page 3-45, and **1588 PTP Interface Signals** on page 3-47.<br>• Updated address ranges in Low Latency 40-100GbE IP Core Address Map table in **Software Interface: Registers** on page 3-72. |

| Date | Compatible ACDS Version | Changes |
|---|---|---|
| 2015.11.02 | 15.1 | • Updated for new Quartus Prime software v15.1 release.<br>• Added new Quick Start Guide chapter to accelerate familiarity with the IP core and document the new hardware design example and supporting parameter editor tab.<br>• Added new parameters **Enable Altera Debug Master Endpoint** and **Enable ODI acceleration logic** to turn on these features in the Arria 10 Native PHY IP core that specifies the transceiver settings in Arria 10 variations of the LL 40-100GbE IP core. Refer to **IP Core Parameters** on page 2-5.<br>• Updated descriptions of PTP support functionality to incorporate changes to the IP core v15.1.<br>  • Added support for fingerprint passing.<br>  • Added 64-bit timestamp interface option to TOD module and LL 40-100GbE IP core.<br>  • Added new parameters **Enable 96b Time of Day Format**, **Enable 64b Time of Day Format**, and **Timestamp fingerprint width**. Refer to **IP Core Parameters** on page 2-5.<br>  • Removed `TX_PTP_STATUS` register. To specify that the 1588 PTP module should always provide the timestamps in outbound Ethernet frames in V2 format or in V1 format, turn on only one of the new parameters **Enable 96b Time of Day Format** and **Enable 64b Time of Day Format**. If you turn on both of these parameters, you can specify the current timestamp format in one-step mode with the relevant format signal. In two-step mode, you can maintain both formats at once. To specify whether the IP core uses the one-step process or the two-step process, assert the appropriate signals for the desired process for the current packet. Refer to **PTP Transmit Functionality** on page 3-43, **PTP Timestamp and TOD Formats** on page 3-45, **1588 PTP Registers** on page 3-112, **IP Core Parameters** on page 2-5, and **1588 PTP Interface Signals** on page 3-47.<br>  • Added new 1588 PTP registers `TX_PTP_ASYM_DELAY`, `TX_PTP_PMA_LATENCY`, and `RX_PTP_PMA_LATENCY`. Refer to **1588 PTP Registers** on page 3-112.<br>  • Added new signals and renamed all old top-level 1588 PTP block signals. Refer to **1588 PTP Interface Signals** on page 3-47.<br>• Corrected information about link fault signaling signals and registers, in **Link Fault Signaling Interface** on page 3-33 and **Link Fault Signaling Registers** on page 3-79.<br>• Clarified that you must assert the `reset_async` signal for a full ten `clk_status` cycles to ensure correct reset of the IP core. Refer to **Resets** on page 3-61.<br>• Clarified that if you do not turn on **Enable alignment EOP on FCS word**, the delay from EOP to FCS error indication is non-deterministic. Refer to **IP Core Parameters** on page 2-5 |

| Date | Compatible ACDS Version | Changes |
|---|---|---|
| | | • Updated 40GBASE-KR4 registers. Refer to **LL 40GBASE-KR4 Registers** on page 3-81.<br>  • Added default value for 0xB0[12]: `LT Failure Reset`<br>  • Added two fields that were new in the 15.0 Update 1 release: 0xD0[18]: `VOD Training Enable` and 0xD0[19]: `Bypass DFE`<br>  • Added default value for 0xD0[21:20]: `rx_ctle_mode`<br>• Added information about IP core behavior in case of loss of signal. Refer to *Debugging the 40GbE and 100GbE Link*.<br>• Fixed description of `PHY_FRAME_ERROR` register at offset 0x323 to clarify it is not sticky. Refer to **PHY Registers** on page 3-76.<br>• Added figure to help distinguish the TX transceiver PLL from the TX MAC PLL, in **Transceiver PLL Required in Arria 10 Designs** on page 2-20.<br>• Corrected allowed frequency range for `clk_status` in description of **Status clock rate** parameter allowed values, in **IP Core Parameters** on page 2-5. The allowed frequency range remains correct at 100–125 MHz in other locations in the user guide.<br>• Added missing reset values for various register fields.<br>• Fixed assorted typos and formatting issues, including register field width in `IPG_COL_REM` register at offset 0x406. |
| 2015.05.04 | 15.0 | • User guide part number change from UG-01150 to UG-01172.<br>• Updated release-specific information for the software release v15.0.<br>• Added Synchronous Ethernet support for Arria 10 variations. Documented new **Enable SyncE** parameter and new `clk_rx_recover` output signal. Described expected usage in **Clocks** on page 3-58.<br>• Updated handling of received malformed packets, in **LL 40-100GbE IP Core Malformed Packet Handling** on page 3-18, to incorporate these changes in the IP core v15.0:<br>  • The IP core asserts the `l<n>_rx_error[0]` or `rx_error[0]` signal in the case of an unexpected control character that is not an Error character.<br>  • Both the LL 40GbE IP core and the LL 100GbE IP core handle received malformed packets the same way.<br>• Updated the descriptions of `l<n>_rx_error[0]` and `rx_error[0]` from PHY error to malformed packet error.<br>• Added new three-bit `l<n>_rx_status` and `rx_status` signals on the RX client interface. These signals explain the control frames that the IP core passes to the client interface. Refer to **Low Latency 40-100GbE IP Core RX Data Bus** on page 3-21 and **Low Latency 40-100GbE IP Core RX Data Bus Without Adapters (Custom Streaming Interface)** on page 3-25, and to new section **Control Frame Identification** on page 3-20. |

| Date | Compatible ACDS Version | Changes |
|---|---|---|
| | | • Added new TX error insertion feature. User logic can direct the IP core to insert an error in an outgoing Ethernet frame, using the new `l<n>_tx_error` and `tx_error` input signals on the TX client interface. Refer to **Low Latency 40-100GbE IP Core TX Data Bus with Adapters (Avalon-ST Interface)** on page 3-7 and **Low Latency 40-100GbE IP Core TX Data Bus Without Adapters (Custom Streaming Interface)** on page 3-10, and to new section **Error Insertion Test and Debug Feature** on page 3-5.<br>• Updated description of priority-based flow control. Priority-based flow control is now available for both LL 40GbE IP core variations and LL 100GbE IP core variations. Previously it was available only in LL 100GbE variations.<br>• Added new signal `tx_lanes_stable`, in **PHY Status Interface** on page 3-55.<br>• Updated description of the `rx_ctle_mode` 40GBASE-KR4 register and added default value for `Enable Arria 10 Calibration` 40GBASE-KR4 register. Also corrected the bit range of the `LP Coefficients Update` register field. For these changes and other 40GBASE-KR4 register information specific to this IP core, refer to **LL 40GBASE-KR4 Registers** on page 3-81. For changes to the underlying Arria 10 10GBASE-KR PHY registers, refer to the Arria 10 Transceiver PHY User Guide or to the **Arria 10 10GBASE-KR Registers** on page 5-1 appendix.<br>• Corrected the addresses of the `CNTR_TX_STATUS` and `CNTR_RX_STATUS` registers.<br>• Corrected the descriptions of the `PHY_PCS_INDIRECT_ADDR` and `PHY_PCS_INDIRECT_DATA` registers at offsets 0x314 and 0x315.<br>• Added note recommending that designs with multiple LL 40-100GbE IP cores not use the ATX PLL HDL code that is currently provided with the IP core. This code is deprecated. Refer to **Transceiver PLL Required in Arria 10 Designs** on page 2-20.<br>• Added clock information for multiple IP core top-level signals.<br>• Added information about how to check for word lock and alignment marker lock, in *Debugging the 40GbE and 100GbE Link*.<br>• Updated **Differences Between Low Latency 40-100GbE IP Core and 40-100GbE IP Core v15.1** appendix with information about the v15.0 IP core.<br>• Clarified the different paths to the example design and testbench directories and scripts in variations that target different device families.<br>• Added description of the `AM_CNT_BITS` parameter in **Optimizing the Low Latency 40-100GbE IP Core Simulation With the Testbenches** on page 2-30. You might need to modify this parameter in your own simulation environment. |

| Date | Compatible ACDS Version | Changes |
|---|---|---|
| | | • Removed text of optimization parameter display `initial` block from **Optimizing the Low Latency 40-100GbE IP Core Simulation With the Testbenches** on page 2-30. Refer to the testbench for your IP core variation for the correct HDL code for the variation.<br>• Changed default value of 40GBASE-KR4 link training PMA parameter `INITPOSTVAL` from 22 to 13, in **IP Core Parameters** on page 2-5.<br>• Updated descriptions of IP core example project. The IP core now generates an example project that is configurable on a device, for most variations. The older type of example projects, which are not configurable, are also generated. Refer to *Low Latency 40-100GbE IP Core Design Example*.<br>• Fixed assorted minor errors and typos. |
| 2014.12.17 | 14.1 | • Corrected PTP usage figures in Implementing a 1588 System That Includes a LL 40-100GbE IP Core.<br>• Clarified that user-defined extra latency is included in calculation of PTP exit timestamp in both one-step mode and two-step mode, in PTP Transmit Functionality.<br>• Clarified that TOD module is expected to provide the current continuously updating time of day. The output signals of this module must provide the current time of day on every clock cycle, in V2 format.<br>• Moved External Time-of-Day Module for 1588 PTP Variations into the 1588 PTP section of the Functional Description chapter. |
| 2014.12.15 | 14.1 | • Updated release-specific information for the software release v14.1.<br>• Moved licensing information and the description of the OpenCore Plus evaluation feature to Getting Started chapter.<br>• Added option to instantiate the TX MAC PLL outside the IP core. The new PLL generates the TX MAC clock. Added new parameter **Use external TX MAC PLL**. If you turn on this parameter the IP core has an additional input clock signal `clk_txmac_in`.Changes primarily located in new sections External TX MAC PLL in Getting Started chapter and External TX MAC PLL in Functional Description chapter, in IP Core Parameters section, and in Clocks section.<br>• Added support for new 40GBASE-KR4 LL 40GbE IP core variation. Changes located in existing IP Core Parameters section and descriptions of the testbench for these IP core variations in the Low Latency 40-100GbE IP Core Testbenches section. Added new sections Clock Requirements for 40GBASE-KR4 Variations, Low Latency 40GBASE-KR4 IP Core Variations, and LL 40GBASE-KR4 Registers, and a reference appendix Arria 10 10GBASE-KR Registers. |

| Date | Compatible ACDS Version | Changes |
|---|---|---|
| | | • Added new six-bit RX error signal on client interface. On Avalon-ST client interface, `l<n>_rx_error[5:0]` replaces single-bit RX error signal `l<n>_rx_error`. On custom client interface, `rx_error[5:0]` is new signal.<br>• Added information about how the IP core handles malformed packets it receives. Previously the IP core did not terminate an incoming packet if it receives an unexpected control character. Changes are located in the new LL 40-100GbE IP Core Malformed Packet Handling section.<br>• Added two new 64-bit statistics counters `RXOctets_OK` (offset 0x960) and `TxOctetsOK` (offset 0x860) to count the payload bytes (octets) in received and transmitted frames with no FCS errors, undersized, oversized, or payload length errors.<br>• Added four new signals in new octetsOK interface. These signals provide per-frame information about the octet count in the two new statistics counters: rx_inc_octetsOK, rx_inc_octetsOK_valid, tx_inc_octetsOK, tx_inc_octetsOK_valid. Refer to new section OctetOK Count Interface.<br>• Added new `CFG_PLEN_CHECK` register at offset 0x50A, to support bit[4] of the new six-bit RX error signal.<br>• Added new link fault signals `unidirectional_en` and `link_fault_gen_en` that provide status from the `LINK_FAULT_CONFIG` register.<br>• Described new method for handling module-specific signals when the module is not included in your IP core variation. TX MAC input clock, link fault signals, pause signals, and PTP signals are not available in newly generated IP cores that do not include the relevant module. However, for backward compatibility, if you upgrade an IP core variation, link fault signals, pause signals, and PTP signals in the earlier release of the IP core variation remain available in the 14.1 version after upgrade. |

| Date | Compatible ACDS Version | Changes |
|---|---|---|
| | | • Updated PTP module description and signals for current IP core version to support changes to the IP core 1588 PTP functionality. Corrected description of the `tx_in_ptp_overwrite[1:0]` signal. Improved usage description. If you turn on **Enable 1588 PTP**, the PTP module has the following new features and requirements::<br>  • You must instantiate a time-of-day module and connect it to the IP core. Previously, this module was included in the IP core. The change facilitates TOD module sharing among IP cores.<br>  • Added new PTP signals `tod_rxmac_in` and `tod_txmac_in` to receive the timestamps the TOD module generates in the two clock domains.<br>  • Removed TX PTP module TOD calculation registers at 0xB06 through 0xB08. The TOD module now provides the functionality the registers supported in previous versions of the IP core.<br>  • Added support for resetting the TCP checksum to zero if the application does not recalculate it. The new feature adds two new PTP signals `tx_in_zero_tcp` and `tx_in_tcp_offset` with which the application communicates such a request to the IP core.<br>• Clarified that IP core does not identify frames of eight bytes or less as runts but instead as FCS errors.<br>• Clarified that IP core does not generate frames of eight bytes or less.<br>• Added waveform to illustrate register access on the control and status interface, in the Control and Status Interface section<br>• Updated Differences Between Low Latency 40-100GbE IP Core and 40-100GbE IP Core v14.1 appendix with information about the v14.1 IP core.<br>• Corrected misinformation about the IP core variations that support the **Enable TX CRC insertion** parameter. Only IP core variations that do not support flow control provide this parameter.<br>• Corrected order of `TX_PFC_DADDRH` (offset 0x60E) and `TX_PFC_DADDRL` (offset 0x60D) registers.<br>• Corrected order of `TX_PFC_SADDRH` (offset 0x610) and `TX_PFC_SADDRL` (offset 0x60F) registers.<br>• Corrected description of `RX_PAUSE_DADDR` registers at offsets 0x707 and 0x708. These registers are the same for standard flow control and for priority-based flow control.<br>• Corrected installation directory figure for Stratix V variations, in Installation and Licensing for LL 40-100GbE IP Core for Stratix V Devices.<br>• Corrected erroneous indication that the value in `RETRANSMIT_XOFF_HOLDOFF_EN` at offset 0x607 is indexed by the value in the `TX_PAUSE_QNUMBER` register. In fact, the `RETRANSMIT_XOFF_HOLDOFF_EN` register includes one bit for every flow-control priority queue (one bit in case of standard flow control). Change is in Pause Registers section.<br>• Fixed assorted typos and minor errors. |

| Date | Compatible ACDS Version | Changes |
|---|---|---|
| 2014.08.18 | 14.0 and 14.0 Arria 10 Edition | • Updated for new Quartus II IP Catalog, which replaces the MegaWizard Plug-In Manager starting in the Quartus II sofware v14.0. Changes are located primarily in Getting Started chapter. Reordered the chapter to accommodate the new descriptions.<br>• Updated the Installation section to clarify that the LL 40-100GbE IP core v14.0 is available from the Self-Service Licensing Center, and the LL 40-100GbE IP core v14.0 Arria 10 Edition is included in the Quartus II software installation.<br>• Added new, additional allowed value for PHY reference clock frequency: 322.265625 MHz.<br>• Added new parameter option to configure an inter-packet gap of 8.<br>• Added new parameter option to configure the IP core without adapters, exposing a custom streaming client interface that is narrower than the Avalon-ST interface. This option is available in IP cores configured without a 1588 PTP module and without an internal flow control scheme. You must select the custom streaming client interface or the Avalon-ST client interface. Your selection applies to both the RX and TX client interfaces.<br>• Added new parameter option to configure priority-based flow control. This option is available in 100GbE variations with Tx CRC insertion turned on and with an Avalon-ST client interface.<br>  • Adds a new parameter to specify between 1 and 8 priority queues, inclusive.<br>  • Expands the width of the pause signals to the number of priority queues. Each bit refers to the corresponding priority queue.<br>  • Modifies the pause registers to apply to both standard and priority-based flow control. In some cases the register fields do not change and in others the register field widens to one bit per priority queue. In the case of three registers, modifies them drastically so they are essentially different registers in the case of priority-based flow control:<br>  • `TX_XOF_EN` at offset 0x60A has no equivalent register in priority-based flow control. The replacement register at this offset, `TX_PAUSE_QNUMBER`, holds the queue number of the queue to which the current contents of the `RETRANSMIT_XOFF_HOLDOFF_EN`, `RETRANSMIT_XOFF_HOLDOFF_QUANTA`, and `TX_PAUSE_QUANTA` apply.<br>  • `RX_PAUSE_DADDR1` and `RX_PAUSE_DADDR0` are replaced with `RX_PFC_DADDRH` and `RX_PFC_DADDRL`, which divide the 48-bit destination address for matching differently than the standard flow-control registers.<br>  • Adds new pause registers `CFG_RETRANSMIT_EN`, `CFG_RETRANSMIT_QUANTA`, `TX_PFC_DADDRH`, `TX_PFC_DADDRL`, `TX_PFC_SADDRH`, and `TX_PFC_SADDRL`. |

| Date | Compatible ACDS Version | Changes |
|---|---|---|
| | | • Stratix V variations no longer support an internal transceiver reconfiguration controller. User logic must instantiate a transceiver reconfiguration controller.<br>  • Removed registers at offsets 0x350, 0x351, 0x352, and 0x353.<br>  • Added signals `reconfig_from_xcvr`, `reconfig_to_xcvr`, and `reconfig_busy` in Stratix V variations.<br>• Updated Differences Between Low Latency 40-100GbE IP Core and 40-100GbE IP Core v14.1 appendix with new Low Latency 40-100GbE IP core features.<br>• Updated description of example project to clarify the Arria 10 project no longer implements additional input clocks to demonstrate static timing derivation.<br>• Added instructions to generate the testbench and example project in the 14.0 and 14.0 Arria 10 Edition versions of the IP core.<br>• Reorganized the testbench sections of Chapter 2, Getting Started.<br>• Corrected addresses of the following statistics registers:<br>  • `CNTR_TX_CRCERR` registers from 0x804-0x805 to 0x806-0x807.<br>  • `CNTR_TX_FCS` registers from 0x806-0x807 to 0x804-0x805<br>  • `CNTR_RX_CRCERR` registers from 0x904-0x8905 to 0x906-0x907.<br>  • `CNTR_RX_FCS` registers from 0x906-0x907 to 0x904-0x905<br>• Corrected direction of `clk_txmac` and `clk_rxmac` in table in Signals section to Output.<br>• Added HW reset values for PHY registers that have them. All PHY register listings except those for the identifier string registers and the frequency registers now list a HW reset value.<br>• Added basic information to description of HI BER field of `PHY_RXPCS_STATUS` register. |
| 2014.04.28 | 13.1 Update 3<br>13.1 Arria 10 Edition Update 2 | • Corrected presentation of resource utilization numbers for Stratix V device family to clarify size of IP core without 1588 PTP module<br>• Renamed Pause Registers tables to Ethernet Flow Control (Pause Functionality) Registers for consistency<br>• In Low Latency 40-100GbE IP Core Address Map table, clarified that the 1588 PTP registers are only available if you turn on the parameter to include the 1588 PTP module in your IP core instance. The same information was already available in the table for other registers that are only available if you turn on the associated parameter.<br>• Fixed assorted typos (column width in TX MAC Configuration Registers table, reset value widths of `RX_PAUSE_DADDR0` register in RX Ethernet Flow Control Registers table and of `TX_PTP_CLOCK_PERIOD` register in TX 1588 PTP Registers table) |

| Date | Compatible ACDS Version | Changes |
|---|---|---|
| 2014.04.11 | 13.1 Update 3<br>13.1 Arria 10 Edition Update 2 | • Added resource utilization numbers for Stratix V device family.<br>• Corrected supported Stratix V device speed grade information for Low Latency 100GbE IP cores.<br>• Added note to supported Stratix V device speed grade information table clarifying that Quartus II seed sweeping might be required for variations that include a 1588 PTP module to achieve a comfortable timing margin. |
| 2014.03.06 | 13.1 Update 3<br>13.1 Arria 10 Edition Update 2 | • Corrected descriptions of pause enable registers:<br>  • Added description for `TX_XOF_EN` register at offset 0x60A.<br>  • Enhanced description of `cfg_enable` field of `RX_PAUSE_ENABLE` register at offset 0x705 to clarify the different functions of the `TX_XOF_EN` and `RX_PAUSE_ENABLE` registers in enabling the IP core to incoming pause frames.<br>  • Enhanced description of the `TX_PAUSE_EN` register at offset 0x605 to clarify the different functions of the `TX_XOF_EN` and `TX_PAUSE_EN` registers.<br>• In "Differences Between Low Latency 40-100GbE IP Core and 40-100GbE IP Core v13.1", enhanced the description of the differences in pause frame control and processing and in use of maximum frame size register information.<br>• Updated "Congestion and Flow Control Using Pause Frames" with the enable register field information.<br>• Updated "Pause Control Frame Filtering" with the new enable register field information and to clarify that by default, RX pause frame processing is enabled.<br>• Enhanced "Low Latency 40-100GbE Example Project" to clarify that some project aspects are relevant only for Arria 10 devices.<br>• Corrected "Clocks" section to include `tx_serial_clk` input clocks and to list `clk_rxmac` and `clk_txmac` as output clocks. |
| 2014.02.17 | 13.1 Update 3<br>13.1 Arria 10 Edition Update 2 | Initial release. |

# How to Contact Altera

### Table C-2: How to Contact Altera

To locate the most up-to-date information about Altera products, refer to this table. You can also contact your local Altera sales office or sales representative.

| Contact | Contact Method | Address |
|---|---|---|
| Technical support | Website | www.altera.com/support |
| Technical training | Website | www.altera.com/training |
| | Email | custrain@altera.com |
| Product literature | Website | www.altera.com/literature |
| Nontechnical support: general | Email | nacomp@altera.com |
| Nontechnical support: software licensing | Email | authorization@altera.com |

**Related Information**

- **www.altera.com/support**
- **www.altera.com/training**
- **custrain@altera.com**
- **www.altera.com/literature**
- **nacomp@altera.com**
- **authorization@altera.com**

## Typographic Conventions

**Table C-3: Typographic Conventions**

Lists the typographic conventions this document uses.

| Visual Cue | Meaning |
|---|---|
| **Bold Type with Initial Capital Letters** | Indicate command names, dialog box titles, dialog box options, and other GUI labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI. |
| **bold type** | Indicates directory names, project names, disk drive names, file names, file name extensions, software utility names, and GUI labels. For example, \ **qdesigns** directory, **D:** drive, and **chiptrip.gdf** file. |
| *Italic Type with Initial Capital Letters* | Indicate document titles. For example, *Stratix V Design Guidelines*. |

| Visual Cue | Meaning |
|---|---|
| *italic type* | Indicates variables. For example, *n* + 1.<br>Variable names are enclosed in angle brackets (< >). For example, <*file name*> and <*project name*> **.pof** file. |
| Initial Capital Letters | Indicate keyboard keys and menu names. For example, the Delete key and the Options menu. |
| "Subheading Title" | Quotation marks indicate references to sections in a document and titles of Quartus Prime Help topics. For example, "Typographic Conventions." |
| `Courier type` | Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. The suffix `n` denotes an active-low signal. For example, `resetn`.<br>Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`.<br>Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`). |
| 1., 2., 3., and a., b., c., and so on | Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure. |
| • | Bullets indicate a list of items when the sequence of the items is not important. |
| The **Subscribe** button links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents. | |
| The **Feedback** icon allows you to submit feedback to Altera about the document. Methods for collecting feedback vary as appropriate for each document. | |

**Related Information**

**Email Subscription Management Center**

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***