



# Low Latency 40G for ASIC Proto Ethernet Intel® FPGA IP User Guide

Updated for Intel® Quartus® Prime Design Suite: **20.3**

IP Version: **19.1.0**



[Subscribe](#)

[Send Feedback](#)

**UG-20291 | 2020.10.05**

Latest document on the web: [PDF](#) | [HTML](#)



# Contents

---

- 1. About the Low Latency 40G for ASIC Proto Ethernet Intel® FPGA IP..... 4**
  - 1.1. Low Latency 40G for ASIC Proto Ethernet IP Core Supported Features..... 5
  - 1.2. Device Family Support.....6
  - 1.3. Low Latency 40G for ASIC Proto Ethernet IP Core Device Speed Grade Support..... 7
  - 1.4. Resource Utilization..... 7
  - 1.5. Release Information.....9
- 2. Low Latency 40G for ASIC Proto Ethernet IP Core Parameters..... 10**
- 3. Getting Started..... 13**
  - 3.1. Installing and Licensing Intel FPGA IP Cores..... 13
  - 3.2. Specifying the Low Latency 40G for ASIC Proto Ethernet IP Core Parameters and Options.....13
  - 3.3. Simulating the IP Core.....14
  - 3.4. Generated File Structure..... 15
  - 3.5. Integrating Your IP Core in Your Design..... 16
    - 3.5.1. Pin Assignments..... 16
    - 3.5.2. Adding the Transceiver PLL ..... 17
    - 3.5.3. Adding the External TX MAC PLL..... 18
    - 3.5.4. Placement Settings for the Low Latency 40G for ASIC Proto Ethernet Core.....19
  - 3.6. Low Latency 40G for ASIC Proto Ethernet IP Core Testbench.....19
    - 3.6.1. Understanding the Testbench Behavior.....19
  - 3.7. Compiling the Full Design and Programming the FPGA..... 20
- 4. Functional Description..... 21**
  - 4.1. Low Latency 40G for ASIC Proto Ethernet Core Functional Description.....21
    - 4.1.1. Low Latency 40G for ASIC Proto Ethernet Core TX MAC Datapath..... 22
    - 4.1.2. Low Latency 40G for ASIC Proto Ethernet Core RX MAC Datapath..... 23
    - 4.1.3. Link Fault Signaling Interface.....25
    - 4.1.4. Flow Control.....27
  - 4.2. User Interface to Ethernet Transmission..... 29
    - 4.2.1. Order of Transmission.....29
    - 4.2.2. Bit Order For TX and RX Datapaths.....30
- 5. Reset..... 31**
- 6. Interfaces and Signal Descriptions..... 33**
  - 6.1. TX MAC Interface to User Logic.....33
  - 6.2. RX MAC Interface to User Logic.....35
  - 6.3. TX PCS Interface to User Logic..... 37
  - 6.4. RX PCS Interface to User Logic..... 38
  - 6.5. Transceivers..... 39
  - 6.6. Transceiver Reconfiguration Signals..... 39
  - 6.7. Avalon Memory-Mapped Management Interface..... 41
  - 6.8. Miscellaneous Status and Debug Signals..... 41
  - 6.9. Reset Signals..... 42
  - 6.10. Clocks..... 42
  - 6.11. Flow Control Interface..... 44



- 7. Control, Status, and Statistics Register Descriptions.....45**
  - 7.1. PHY Registers..... 45
  - 7.2. TX MAC Registers.....48
  - 7.3. RX MAC Registers..... 49
  - 7.4. Statistics Registers.....50
    - 7.4.1. TX Statistics Registers..... 50
    - 7.4.2. RX Statistics Registers..... 54
- 8. Debugging the Link..... 58**
- 9. Ethernet Toolkit Overview..... 60**
  - 9.1. Features..... 60
- 10. Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP User Guide Archives.....62**
- 11. Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Revision History..... 63**

## 1. About the Low Latency 40G for ASIC Proto Ethernet Intel® FPGA IP

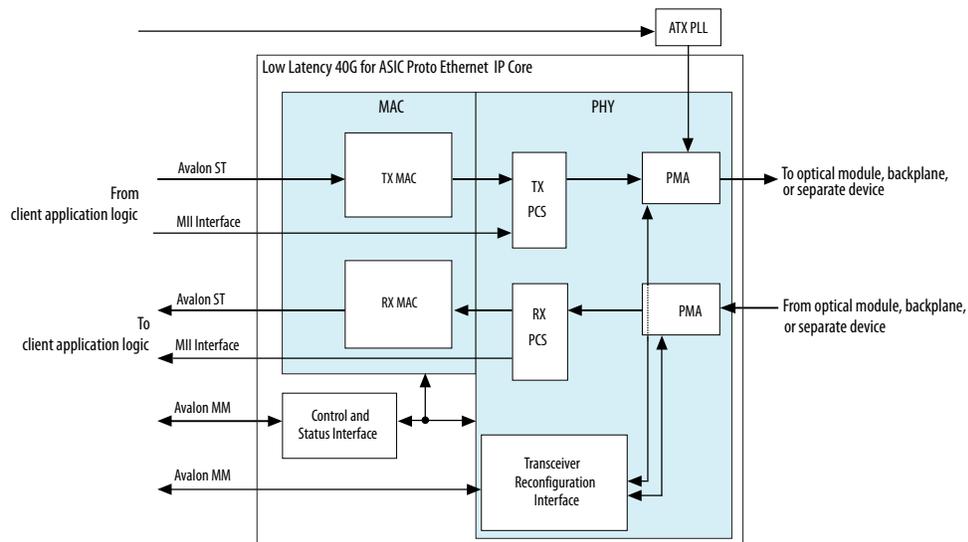
The Low Latency 40G for ASIC Proto Ethernet Intel® FPGA IP core implements the *IEEE 802.3-2010 40G Ethernet Standard*. The IP core includes options to support unidirectional transport as defined in *Clause 66* of the *IEEE 802.3-2012 Ethernet Standard*.

**Table 1. Client Interfaces for IP Core Variations**

IP Core Variation	Client Interface Type	Client Interface Width (Bits)
MAC+PCS	Avalon® streaming interface (Avalon ST)	128
PCS_Only	Media Independent Interface (MII)	128

The IP core provides standard media access control (MAC), physical coding sublayer (PCS), and physical medium attachment (PMA) functions. The PHY comprises the PCS and PMA.

**Figure 1. Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Block Diagram**



### Related Information

- [Introduction to Intel FPGA IP Cores](#)  
 Provides general information about all Intel FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.



- [Creating a Combined Simulator Setup Script](#)  
Create simulation scripts that do not require manual updates for software or IP version upgrades.
- [Managing Intel Quartus Prime Projects](#)  
Guidelines for efficient management and portability of your project and IP files.

## 1.1. Low Latency 40G for ASIC Proto Ethernet IP Core Supported Features

The Low Latency 40G for ASIC Proto Ethernet IP core supports the following features:

- Parameterizable through the IP Catalog available with the Intel Quartus® Prime Pro Edition software.
- Designed to the *IEEE 802.3ba-2010 High Speed Ethernet Standard* available on the IEEE website ([www.ieee.org](http://www.ieee.org)).
- Soft PCS logic that interfaces seamlessly to Intel FPGA 10.3125 gigabits per second (Gbps) serial transceivers.
- Standard XLAUI external interface consisting of FPGA hard serial transceiver lanes operating at 10.3125 Gbps.
- Supports Synchronous Ethernet (SyncE) 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). Interface has data width 128 bits.
- Support for jumbo packets, defined as packets greater than 1500 bytes.
- Receive (RX) CRC removal and pass-through control.
- Optional transmit (TX) CRC generation and insertion.
- RX CRC checking and error reporting.
- RX and TX preamble pass-through option for applications that require proprietary user management information transfer.
- Optional RX strict SFD checking per IEEE specification.
- RX malformed packet checking per IEEE specification.
- TX automatic frame padding to meet the 64-byte minimum Ethernet frame length.
- Received control frame type indication.
- Unidirectional transport as defined in *Clause 66 of the IEEE 802.3-2012 Ethernet Standard*
- Hardware and software reset control.
- MAC provides RX cut-through frame processing, no RX store-and-forward capability.
- Deficit idle counter (DIC) to maintain a 12-byte inter-packet gap (IPG) average.
- Optional fault signaling detects and reports local fault and generates remote fault, with *IEEE 802.3ba-2012 Ethernet Standard Clause 66* support.



- Optional access to Native PHY Debug Master Endpoint (NPDME) for serial link debugging.
- Programmable ready latency of 0 or 3 clock cycles for Avalon-ST TX interface.
- Optional statistics counters.

For a detailed specification of the Ethernet protocol refer to the *IEEE 802.3 Ethernet Standard*.

### Related Information

[IEEE website](#)

The *IEEE 802.3 Ethernet Standard* is available on the IEEE website.

## 1.2. Device Family Support

**Table 2. Intel FPGA IP Core Device Support Levels**

Device Support Level	Definition
<b>Advance</b>	The IP core is available for simulation and compilation for this device family. Timing models include initial engineering estimates of delays based on early post-layout information. The timing models are subject to change as silicon testing improves the correlation between the actual silicon and the timing models. You can use this IP core for system architecture and resource utilization studies, simulation, pinout, system latency assessments, basic timing assessments (pipeline budgeting), and I/O transfer strategy (datapath width, burst depth, I/O standards tradeoffs).
<b>Preliminary</b>	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.
<b>Final</b>	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 3. Low Latency 40G for ASIC Proto Ethernet IP Core Device Family Support**

Shows the level of support offered by the Low Latency 40G for ASIC Proto Ethernet IP core for Intel Stratix® 10 FPGA device family.

Intel Stratix 10 Device Family	Support
1SG10MHN3F74C2LGS1_U1	Preliminary
1SG10MHN3F74C2LGS1_U2	Preliminary
1SG10MHN3F74C2LGS2_U1	Preliminary
1SG10MHN3F74C2LGS2_U2	Preliminary
1SG10MHN3F74C2LG_U1	Preliminary
1SG10MHN3F74C2LG_U2	Preliminary
1SG10MHN3F74E2LG_U1	Preliminary
1SG10MHN3F74E2LG_U2	Preliminary

### Related Information

- [Timing and Power Models](#)  
Reports the default device support levels in the current version of the Quartus Prime Standard Edition software.



- [Timing and Power Models](#)  
 Reports the default device support levels in the current version of the Quartus Prime Pro Edition software.

### 1.3. Low Latency 40G for ASIC Proto Ethernet IP Core Device Speed Grade Support

**Table 4. Slowest Supported Device Speed Grades**

The following table lists the minimum speed grade.

IP Core	Device Family	Supported Speed Grades
Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP	Intel Stratix 10	-2

#### Related Information

- [Quartus Standard to Timing Closure and Optimization](#)  
 Use this link for the Quartus Prime Pro Edition Software.
- [Quartus Standard to Timing Closure and Optimization](#)  
 Use this link for the Quartus Prime Standard Edition Software.

### 1.4. Resource Utilization

Resource utilization changes depending on the parameter settings you specify in the Low Latency 40G for ASIC Proto Ethernet parameter editor. For example, if you turn on statistics counters in the Low Latency 40G for ASIC Proto Ethernet parameter editor, the IP core requires additional resources to implement the additional functionality.

**Table 5. IP Core Options Encoding for Resource Utilization Table**

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

MAC + PCS IP Core Variation	A	B	C	D	E	F
Parameter						
Ready latency	0	0	3	3	3	0
Use external TX MAC PLL	On	On	—	—	—	—
Enable TX CRC insertion	—	On	On	On	On	—
Enable link fault generation	—	—	On	—	—	—
Enable preamble passthrough	—	—	On	—	—	—
Enable MAC stats counters	—	On	On	On	On	—



**Table 6. IP Core FPGA Resource Utilization**

Lists the resources and expected performance for selected variations of the Low Latency 40G for ASIC Proto Ethernet IP core in an Intel Stratix 10 device.

These results were obtained using the Intel Quartus Prime 20.2 software version.

- 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 Prime Fitter Report.

IP Core Variation	ALMs	Dedicated Logic Registers	Memory M20K
A	6,900	16,500	1
B	10,700	24,800	1
C	11,400	26,800	1
D	11,100	25,500	1
E	11,100	26,000	1
F	3,000	7,100	1

### Related Information

[Low Latency 40G for ASIC Proto Ethernet IP Core Parameters](#) on page 10  
Information about the parameters and values in the IP core variations.



## 1.5. Release Information

IP versions are the same as the Intel Quartus Prime Design Suite software versions up to v19.1. From Intel Quartus Prime Design Suite software version 19.2 or later, IP cores have a new IP versioning scheme. If an IP core version is not listed, the user guide for the previous IP core version applies.

The IP versioning scheme (X.Y.Z) number changes from one software version to another. A change in:

- X indicates a major revision of the IP. If you update your Intel Quartus Prime software, you must regenerate the IP.
- Y indicates the IP includes new features. Regenerate your IP to include these new features.
- Z indicates the IP includes minor changes. Regenerate your IP to include these changes.

**Table 7. Low Latency 40G for ASIC Proto Ethernet IP Core Current Release Information**

Item	Description
IP Version	19.1.0
Intel Quartus Prime Version	20.2
Release Date	2020.06.22
Ordering Code	IP-40GEUMACPHY

### Related Information

- [Low Latency 40G for ASIC Proto Ethernet IntelFPGA IP Release Information](#)  
Provides information about the new features and update for each IP release.
- [Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Design Example User Guide](#)  
Provides information about the Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP design examples.

## 2. Low Latency 40G for ASIC Proto Ethernet IP Core Parameters

The Low Latency 40G for ASIC Proto Ethernet parameter editor has an **IP** tab and the **Main** tab.

The Low Latency 40G for ASIC Proto Ethernet parameter editor also includes an **Example Design** tab. For information about that tab, refer to the *Low Latency 40G for ASIC Proto Ethernet IP Design Example User Guide*.

**Figure 2. Example IP Tab in the Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Parameter Editor**

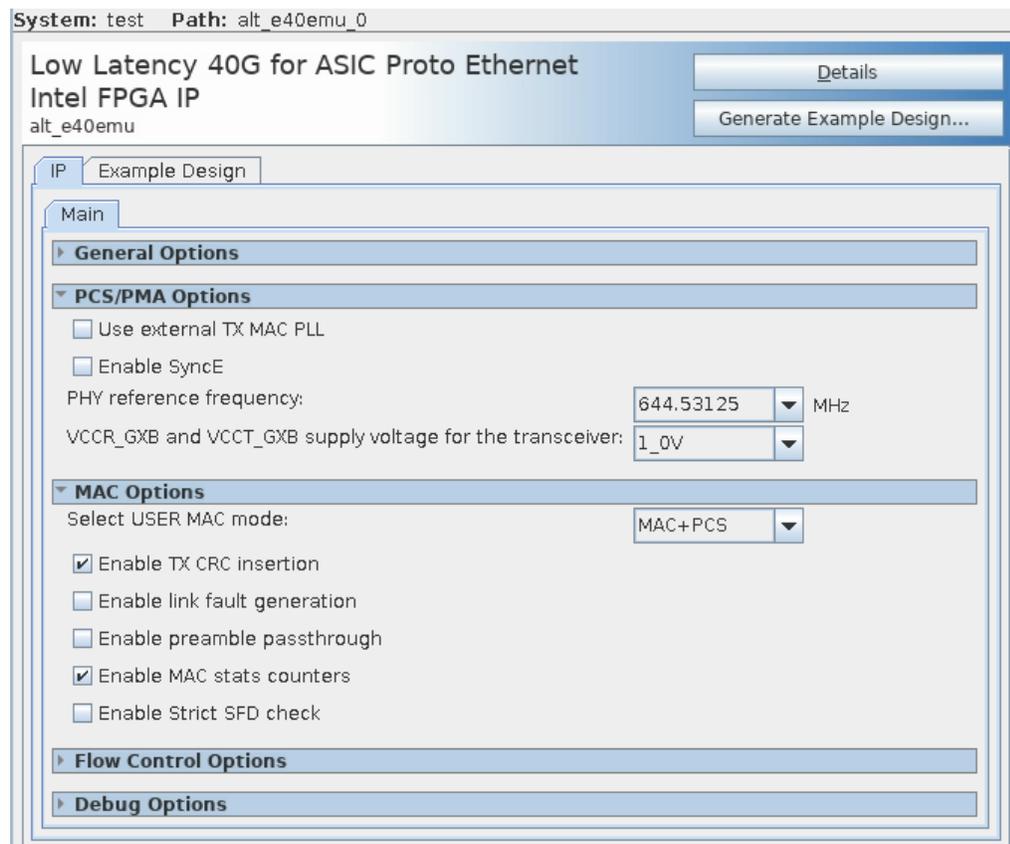




Table 8. Low Latency 40G for ASIC Proto Ethernet IP Core Parameters: Main Tab

Parameter	Range	Default Setting	Description
<b>General</b>			
<b>Target transceiver tile</b>	<b>H-Tile</b>	The tile type of the Intel Quartus Prime project specific target device.	Specifies the transceiver tile on your target device. The Device setting of the Intel Quartus Prime project in which you generate the IP core determines the transceiver tile type.
<b>Protocol speed</b>	<b>40GbE</b>	<b>40GbE</b>	Selects the Ethernet data rate.
<b>Ready latency</b>	<b>0, 3</b>	<b>0</b>	Selects the readyLatency value on the TX client interface. readyLatency is an Avalon streaming interface property that defines the number of clock cycles of delay from when the IP core asserts the <code>l2_tx_ready</code> signal to the clock cycle in which the IP core can accept data on the TX client interface. Refer to the <i>Avalon Interface Specifications</i> . Selecting a latency of 3 eases timing closure at the expense of increased latency for the TX datapath.
<b>PCS/PMA Options</b>			
<b>Use external TX MAC PLL</b>	<b>Enabled, Disabled</b>	<b>Disabled</b>	When enabled, the IP core is configured to expect an input clock to drive the TX MAC. The input clock signal is <code>clk_txmac_in</code> .
<b>Enable SyncE</b>	<b>Enabled, Disabled</b>	<b>Disabled</b>	Exposes the RX recovered clock as an output signal. This feature supports the Synchronous Ethernet standard described in the International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T) G.8261, G.8262, and G.8264 recommendations.
<b>PHY reference frequency</b>	<b>322.265625 MHz, 644.53125 MHz</b>	<b>644.53125 MHz</b>	Sets the expected incoming PHY <code>clk_ref</code> reference frequency. The input clock frequency must match the frequency you specify for this parameter ( $\pm 100\text{ppm}$ ).
<b>VCCR_GXB and VCCT_GXB supply voltage for the transceiver</b>	<b>1_0V, 1_1V</b>	<b>1_0V</b>	Specifies whether the transceiver supply voltage is 1.0 V or 1.1 V. The supply voltage must match the voltage you specify for this parameter.
<b>MAC Options</b>			
<b>Select USER MAC mode</b>	<b>MAC+PCS, PCS_Only</b>	<b>MAC+PCS</b>	Specifies the selection of MAC with the IP. <ul style="list-style-type: none"> <li>MAC+PCS: The generated IP includes the MAC.</li> <li>PCS_Only: The generated IP doesn't contain the MAC.</li> </ul>
<b>Enable TX CRC insertion</b>	<b>Enabled, Disabled</b>	<b>Enabled</b>	When enabled, TX MAC computes and inserts the CRC-32 checksum in the out-going Ethernet frame. When disabled, the TX MAC does not compute a 32-bit FCS in the TX MAC frame. Instead, the client must provide frames with at least 64 bytes, plus the Frame Check Sequence (FCS).
<b>Enable link fault generation</b>	<b>Enabled, Disabled</b>	<b>Disabled</b>	When enabled, the IP core implements link fault signaling as defined in the <i>IEEE 802.3-2012 IEEE Ethernet Standard</i> . The MAC includes a Reconciliation Sublayer (RS) to manage local and remote faults. When enabled, the local RS TX logic can transmit remote fault sequences in case of a local fault and can transmit IDLE control words in case of a remote fault.
<i>continued...</i>			



Parameter	Range	Default Setting	Description
<b>Enable preamble passthrough</b>	<b>Enabled, Disabled</b>	<b>Disabled</b>	When enabled, 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 Start Frame Delimiter (SFD) to the client instead of stripping them out of the Ethernet packet. In TX preamble pass-through mode, the client specifies the preamble and provides the SFD to be sent in the Ethernet frame.
<b>Enable MAC stats counters</b>	<b>Enabled, Disabled</b>	<b>Enabled</b>	When enabled, the IP core includes statistics counters that characterize TX and RX traffic. The statistics module also supports shadow requests that verify counts by taking snapshots of intermediate results.
<b>Enable Strict SFD check</b>	<b>Enabled, Disabled</b>	<b>Disabled</b>	When enabled, the IP core can implement strict SFD checking, depending on register settings.
<b>Flow Control Options</b>			
<b>Enable MAC flow control</b>	<b>Enabled, Disabled</b>	<b>Disabled</b>	When enabled, the IP core implements flow control. When either link partner experiences congestion, the respective transmit control sends pause frames.
<b>Number of queues in priority flow control</b>	1-8	8	Specifies the number of queues used in managing flow control.
<b>Configuration, Debug and Extension Options</b>			
<b>Enable Native PHY Debug Master Endpoint (NPDME)</b>	<b>Enabled, Disabled</b>	<b>Disabled</b>	<p>If enabled, the IP core turns on the following features in the Native PHY IP core that is included in the Low Latency 40G for ASIC Proto Ethernet IP core:</p> <ul style="list-style-type: none"> <li><b>Enable Native PHY Debug Master Endpoint (NPDME)</b></li> <li><b>Enable capability registers</b></li> </ul> <p>If turned off, the IP core is configured without these features.</p> <p>For information about these Intel Stratix 10 features, refer to the <i>Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide</i>.</p>
<b>Enable JTAG to Avalon Master Bridge</b>	<b>Enabled, Disabled</b>	<b>Disabled</b>	<p>If turned on, the IP core includes a JTAG to Avalon memory-mapped interface master bridge connecting internally to status and reconfiguration registers.</p> <p>This allows you to run the Ethernet Link Inspector using the System Console.</p>

**Related Information**

- [Avalon Interface Specifications](#)  
Detailed information about Avalon streaming interfaces and the Avalon streaming interface readyLatency parameter.
- [Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide](#)  
Information about the **NPDME** and **Enable capability registers** parameters of the Intel Stratix 10 H-Tile Transceiver PHY IP core.

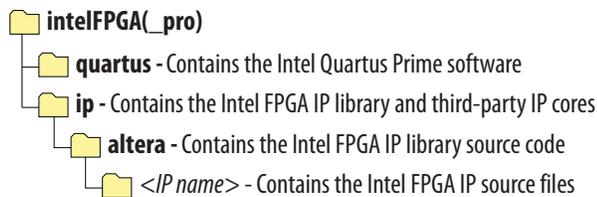
## 3. Getting Started

### 3.1. Installing and Licensing Intel FPGA IP Cores

The Intel Quartus Prime software installation includes the Intel FPGA IP library. This library provides many useful IP cores for your production use without the need for an additional license. Some Intel FPGA IP cores require purchase of a separate license for production use. The Intel FPGA IP Evaluation Mode allows you to evaluate these licensed Intel FPGA IP cores in simulation and hardware, before deciding to purchase a full production IP core license. You only need to purchase a full production license for licensed Intel IP cores after you complete hardware testing and are ready to use the IP in production.

The Intel Quartus Prime software installs IP cores in the following locations by default:

**Figure 3. IP Core Installation Path**



**Table 9. IP Core Installation Locations**

Location	Software	Platform
<drive>:\intelFPGA_pro\quartus\ip\altera	Intel Quartus Prime Pro Edition	Windows*
<drive>:\intelFPGA\quartus\ip\altera	Intel Quartus Prime Standard Edition	Windows
<home directory>:/intelFPGA_pro/quartus/ip/altera	Intel Quartus Prime Pro Edition	Linux*
<home directory>:/intelFPGA/quartus/ip/altera	Intel Quartus Prime Standard Edition	Linux

**Note:** The Intel Quartus Prime software does not support spaces in the installation path.

### 3.2. Specifying the Low Latency 40G for ASIC Proto Ethernet IP Core Parameters and Options

The Low Latency 40G for ASIC Proto Ethernet parameter editor allows you to quickly configure your custom IP variation. Use the following steps to specify IP core options and parameters in the Intel Quartus Prime Pro Edition software.

1. In the Intel Quartus Prime Pro Edition, click **File > New Project Wizard** to create a new Intel Quartus Prime project, or **File > Open Project** to open an existing Intel Quartus Prime project. The wizard prompts you to specify a device.
2. In the IP Catalog (**Tools > 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>.ip`.
4. Click **OK**. The parameter editor appears.
5. On the **IP** tab, specify the parameters for your IP core variation. Refer to [Low Latency 40G for ASIC Proto Ethernet IP Core Parameters](#) on page 10 for information about specific IP core parameters.
6. Optionally, to generate a simulation testbench or compilation and hardware design example, follow the instructions in the *Low Latency 40G for ASIC Proto Ethernet IP Design Example User Guide*.
7. Click **Generate HDL**. The **Generation** dialog box appears.
8. Specify output file generation options, and then click **Generate**. The IP variation files generate according to your specifications.
9. Click **Finish**. The parameter editor adds the top-level `.ip` file to the current project automatically. If you are prompted to manually add the `.ip` file to the project, click **Project > Add/Remove Files in Project** to add the file.
10. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports.

### 3.3. Simulating the IP Core

You can simulate your IP core variation with the functional simulation model and the testbench generated with the IP core. The functional simulation model is a cycle-accurate model that allows for fast functional simulation of your IP core instance using industry-standard Verilog HDL simulators. If your IP core variation does not generate a matching testbench, you can create your own testbench to exercise the IP core functional simulation model.

The functional simulation model and testbench files are generated in project subdirectories. These directories also include scripts to compile and run the example design.

**Note:** Use the simulation models only for simulation and not for synthesis or any other purposes. Using these models for synthesis creates a nonfunctional design.

In the top-level wrapper file for your simulation project, you can set the following RTL parameters to enable simulation optimization. These optimizations significantly decrease the time to reach link initialization.

- `SIM_SHORT_AM`: Shortens the interval between alignment markers to accelerate alignment marker lock. Alignment markers are used when Reed-Solomon FEC is enabled.

In general, parameters are set through the IP core parameter editor and you should not change them manually. The only exceptions are these simulation parameters.



To set the simulation optimization parameters on the PHY blocks, add the following lines to the top-level wrapper file:

```
defparam <dut instance>.SIM_SHORT_AM = 1'b1;
```

**Note:** You can use the example testbench as a guide for setting the simulation parameters in your own simulation environment. These lines are already present in the Intel-provided testbench for the IP core.

**Related Information**

[Simulating Intel FPGA IP Cores](#)

### 3.4. Generated File Structure

The Intel Quartus Prime Pro Edition software generates the following IP core output file structure.

For information about the file structure of the design example, refer to the *Low Latency 40G for ASIC Proto Ethernet IP Design Example User Guide*.

**Table 10. IP Core Generated Files**

File Name	Description
<your_ip>.ip	The Platform Designer (Standard) system or top-level IP variation file. <your_ip> is the name that you give your IP variation.
<your_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. This IP core does not support VHDL. However, the Intel Quartus Prime Pro Edition generates this file.
<your_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.
<your_ip>_generation.rpt	IP or Platform Designer (Standard) generation log file. A summary of the messages during IP generation.
<your_ip>.qgsimc	Lists simulation parameters to support incremental regeneration.
<your_ip>.qgsynthc	Lists synthesis parameters to support incremental regeneration.
<your_ip>.qip	Contains all the required information about the IP component to integrate and compile the IP component in the Intel Quartus Prime Pro Edition software.
<your_ip>.csv	Contains information about the upgrade status of the IP component.
<your_ip>.bsf	A Block Symbol File (.bsf) representation of the IP variation for use in Intel Quartus Prime Block Diagram Files (.bdf).
<your_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.
<your_ip>.ppf	The Pin Planner File (.ppf) stores the port and node assignments for IP components created for use with the Pin Planner.
<your_ip>_bb.v	You can use the Verilog black-box (_bb.v) file as an empty module declaration for use as a black box.
<b>continued...</b>	



File Name	Description
<your_ip>.sip	Contains information required for NativeLink simulation of IP components. You must add the <b>.sip</b> file to your Intel Quartus Prime project.
<your_ip>_inst.v and _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. This IP core does not support VHDL. However, the Intel Quartus Prime Pro Edition generates the _inst.vhd file.
<your_ip>.regmap	If IP contains register information, <b>.regmap</b> file generates. The <b>.regmap</b> file describes the register map information of master and slave interfaces. This enables register display views and user customizable statistics in the System Console.
<your_ip>.svd	Allows hard processor system (HPS) System Debug tools to view the register maps of peripherals connected to HPS within a Platform Designer (Standard) system. During synthesis, the <b>.svd</b> files for slave interfaces visible to System Console masters are stored in the <b>.sof</b> file in the debug section. System Console reads this section, which Platform Designer (Standard) can query for register map information. For system slaves, Platform Designer (Standard) can access the registers by name.
<your_ip>.v	HDL files that instantiate each submodule or child IP core for synthesis or simulation.
mentor/	Contains a ModelSim* script <code>msim_setup.tcl</code> to set up and run a simulation.
aldec/	Contains a Riviera-PRO script <code>rivierapro_setup.tcl</code> to setup and run a simulation. This IP core does not support simulation with the Aldec Riviera-PRO simulator. However, the Intel Quartus Prime Pro Edition generates this directory.
synopsys/vcs/ synopsys/vcsmx/	Contains a shell script <code>vcs_setup.sh</code> to set up and run a VCS* simulation. Contains a shell script <code>vcsmx_setup.sh</code> and <code>synopsys_sim.setup</code> file to set up and run a VCS MX simulation.
cadence/	Contains a shell script <code>ncsim_setup.sh</code> 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, Platform Designer (Standard) generates <code>synth/</code> and <code>sim/</code> sub-directories.

### Related Information

[Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Design Example User Guide](#)  
Provides information about the Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP design examples.

## 3.5. Integrating Your IP Core in Your Design

### 3.5.1. Pin Assignments

When you integrate your Low Latency 40G for ASIC Proto Ethernet core instance in your design, you must make appropriate pin assignments. While compiling the IP core alone, you can create virtual pins to avoid making specific pin assignments for top-level signals. When you are ready to map the design to hardware, you can change to the correct pin assignments.



### Related Information

[Intel Quartus Prime Help](#)

For information about the Intel Quartus Prime software, including virtual pins.

## 3.5.2. Adding the Transceiver PLL

Low Latency 40G for ASIC Proto Ethernet IP cores require an external PLL to drive the TX transceiver serial clock, in order to compile and to function correctly in hardware. In many cases, the same PLL can be shared with other transceivers in your design.

**Figure 4. PLL Configuration Example for MAC+PCS variant**

The TX transceiver PLL is instantiated with an Intel FPGA ATX PLL IP core. The TX transceiver PLL must always be instantiated outside the Low Latency 40G for ASIC Proto Ethernet IP core.

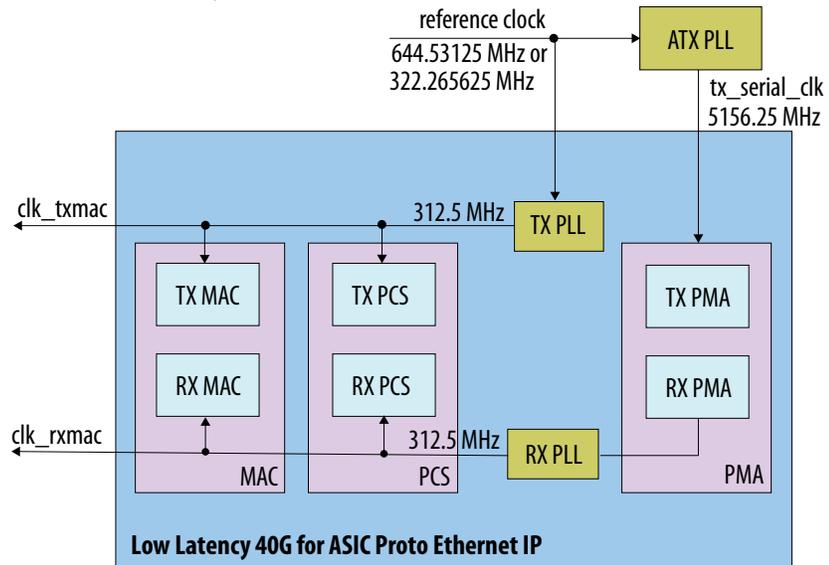
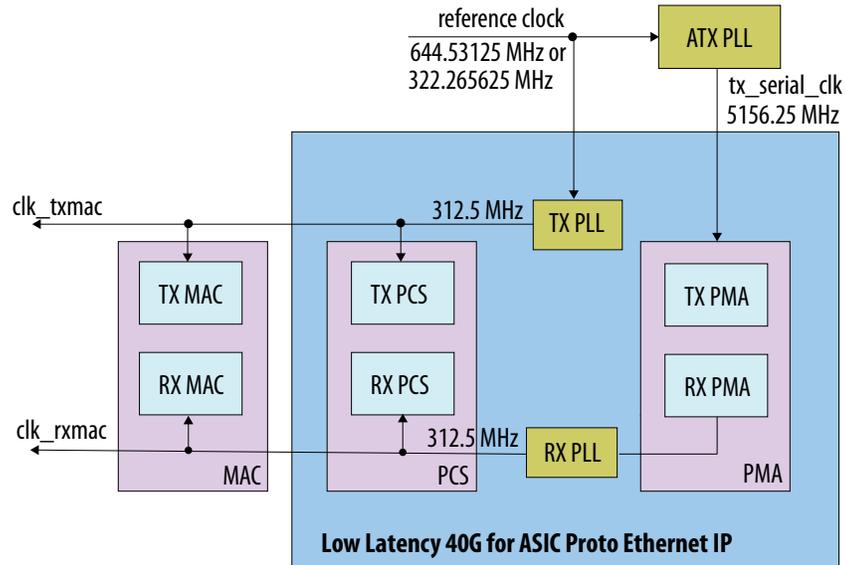


Figure 5. PLL Configuration for PCS only variant



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

- Select **Stratix 10 Transceiver ATX PLL**.
- In the parameter editor, set the following parameter values:
  - **PLL output frequency** to **5156.25 MHz**. 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 Gbps data rate through the transceiver.
  - **PLL integer reference clock frequency** to **644.53125 MHz**.

You must connect the `tx_serial_clk` input pin of the Low Latency 40G for ASIC Proto Ethernet IP core PHY link to the output port of the ATX PLL.

### 3.5.3. Adding the External TX MAC PLL

If you turn on **Use external TX MAC PLL** in the Low Latency 40G for ASIC Proto Ethernet 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. User logic must drive `clk_txmac_in` from a PLL whose input is the PHY reference clock, `clk_ref`.



### 3.5.4. Placement Settings for the Low Latency 40G for ASIC Proto Ethernet Core

The Quartus Prime software provides the options to specify design partitions and Logic Lock (Standard) or Logic Lock 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 design.

#### Related Information

[Intel Quartus Prime Pro Edition User Guide: Design Constraints](#)

Describes incremental compilation, design partitions, and Logic Lock regions.

### 3.6. Low Latency 40G for ASIC Proto Ethernet IP Core Testbench

Intel provides a compilation-only example design and a testbench with most variations of the Low Latency 40G for ASIC Proto Ethernet IP core.

To generate the testbench, 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.

#### Related Information

[Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Design Example User Guide](#)

Provides information about the Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP design examples.

#### 3.6.1. Understanding the Testbench Behavior

The 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 Low Latency 40G for ASIC Proto Ethernet IP core implements virtual lanes as defined in the *IEEE 802.3ba-2012 Ethernet Standard*. The IP core is fixed at four virtual lanes; the four virtual lanes are typically transmitted over four 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.



Both the word lock and the alignment marker lock implement hysteresis as defined in the *IEEE Standard for Ethernet, Section 4*. 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.

### 3.7. Compiling the Full Design and Programming the FPGA

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

*Note:* The Low Latency 40G for ASIC Proto Ethernet core design example synthesis directories include Synopsys Constraint (.sdc) files that you can copy and modify for your own design.

#### Related Information

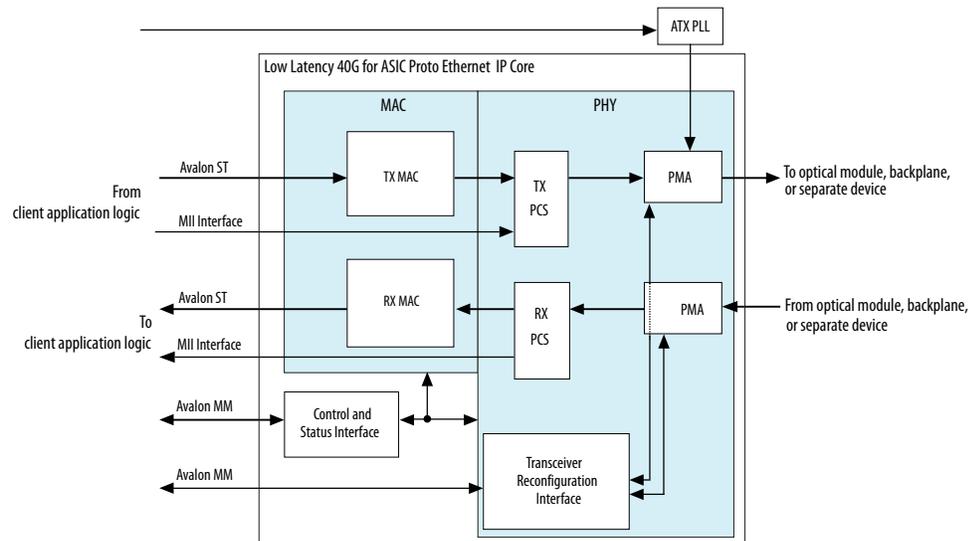
- [Incremental Compilation for Hierarchical and Team-Based Design](#)
- [Programming Intel Devices](#)

## 4. Functional Description

### 4.1. Low Latency 40G for ASIC Proto Ethernet Core Functional Description

The Low Latency 40G for ASIC Proto Ethernet core implements an Ethernet MAC in accordance with the *IEEE 802.3 Ethernet Standard*. The IP core implements an Ethernet PCS and PMA (PHY) that handles the frame encapsulation and flow of data between a client logic and Ethernet network.

**Figure 6. Low Latency 40G for ASIC Proto Ethernet Core Block Diagram**



In the TX direction, the MAC assembles packets and sends them to the PHY. It completes the following tasks:

- Accepts client frames.
- Inserts the inter-packet gap (IPG), preamble, start of frame delimiter (SFD), and padding. The source of the preamble and SFD depends on whether the IP core is in preamble-pass-through mode.
- Adds the CRC bits if enabled.
- Updates statistics counters if enabled.

In the RX direction, the PMA passes frames to the PCS that sends them to the MAC. The MAC completes the following tasks:

- Performs CRC and malformed packet checks.
- Updates statistics counters if enabled.
- Strips out the CRC, preamble, and SFD.
- Passes the remainder of the frame to the client.

In 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, the MAC passes on the CRC bytes to the client and asserts the end-of-packet signal in the same clock cycle as the final CRC byte.

### 4.1.1. Low Latency 40G for ASIC Proto Ethernet Core TX MAC Datapath

The TX MAC module receives the client payload data with the destination and source addresses. It 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. It pads the payload of frames greater than eight bytes to satisfy the minimum Ethernet frame payload of 46 bytes. The TX MAC module inserts IDLE bytes to maintain an average IPG of 12.

#### Figure 7. 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:

- $\langle p \rangle$  = payload size, which is arbitrarily large
- $\langle s \rangle$  = number of padding bytes (0-46)
- $\langle g \rangle$  = number of IPG bytes

#### 4.1.1.1. Frame Padding

When the length of the client frame is less than 64 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 (including CRC).

The IP core filters out all client frames with lengths less than 9 bytes.

#### 4.1.1.2. Preamble Insertion

In the TX datapath the MAC prepends an eight-byte preamble to the client frame. If you turn on **Enable link fault generation**, this MAC module also incorporates the functions of the reconciliation sublayer (RS).

The source of the 7-byte preamble (including a Start byte) and 1-byte SFD depends on whether you turn on **Enable preamble passthrough** in the parameter editor.

Note that a single parameter in the Low Latency 40G for ASIC Proto Ethernet parameter editor turns on both RX and TX preamble passthrough.



### 4.1.1.3. Inter-Packet Gap Generation and Insertion

The TX MAC maintains the minimum inter-packet gap (IPG) between transmitted frames required by the IEEE 802.3 Ethernet standard. The deficit idle counter (DIC) maintains the average IPG of 12 bytes.

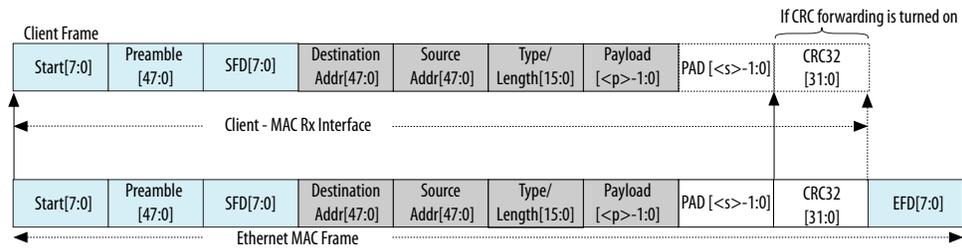
### 4.1.2. Low Latency 40G for ASIC Proto Ethernet Core RX MAC Datapath

The RX MAC receives Ethernet frames and forwards the payload with relevant header bytes to the client after performing some MAC functions on header bytes. The RX MAC processes all incoming valid frames.

**Figure 8. Flow of Client Frame With Preamble Pass-Through Turned On**

This figure uses the following notational conventions:

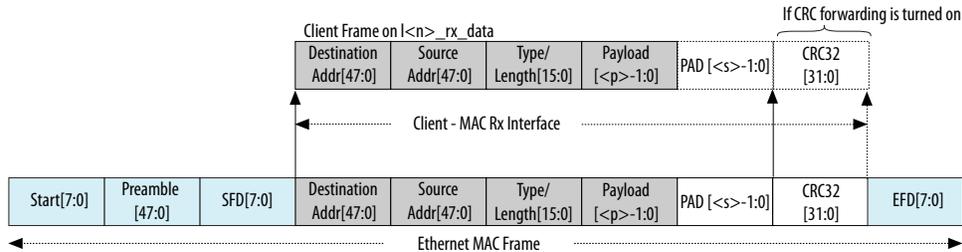
- $\langle p \rangle$  = payload size, which is arbitrarily large.
- $\langle s \rangle$  = number of padding bytes (0–46).



**Figure 9. Flow of Client Frame With Preamble Pass-Through Turned Off**

This figure uses the following notational conventions:

- $\langle p \rangle$  = payload size, which is arbitrarily large.
- $\langle s \rangle$  = number of padding bytes (0–46).



#### 4.1.2.1. IP Core Preamble Processing

If you turn on **Enable preamble passthrough** in the parameter editor, the RX MAC forwards preamble bytes. The TX MAC requires the preamble bytes to be included in the frames at the Avalon Streaming interface.

Note that a single parameter in the Low Latency 40G for ASIC Proto Ethernet parameter editor turns on both RX and TX preamble passthrough.

#### 4.1.2.2. IP Core Strict SFD Checking

The Low Latency 40G for ASIC Proto Ethernet core RX MAC checks all incoming packets for a correct Start byte (0xFB). If you turn on **Enable Strict SFD check** in the Low Latency 40G for ASIC Proto Ethernet parameter editor, you enable the RX MAC to check the incoming preamble and SFD for the following values:

- SFD = 0xD5
- Preamble = 0x555555555555

The RX MAC checks one or both of these values depending on the values in bits [4:3] of the RXMAC\_CONTROL register at offset 0x50A.

**Table 11. Strict SFD Checking Configuration**

Enable Strict SFD check	0x50A[4]: Preamble Check	0x50A[3]: SFD Check	Fields Checked	Behavior if Check Fails
Off	Don't Care	Don't Care	Start byte	IP core does not recognize a malformed Start byte as a Start byte
On	0	0	Start byte	
	0	1	Start byte and SFD	IP Core drops the packet
	1	0	Start byte and preamble	
	1	1	Start byte and preamble and SFD	

#### 4.1.2.3. Length/Type Field Processing

This two-byte header represents either the length of the payload or the type of MAC frame.

- Length/type  $\geq$  0x600—The field represents the frame type. The following frame types are possible:
  - Length/type = 0x8100—VLAN or stacked VLAN tagged frames. The MAC RX continues to check the frame and payload lengths.
  - Length/type = 0x8808—Control frames. The next two bytes are the Opcode field that indicates the type of control frame. For pause frames (Opcode = 0x0001) and PFC frames (Opcode = 0x0101), the MAC RX proceeds with pause frame processing.
  - For other field values, the MAC RX forwards the receive frame to the client.

##### 4.1.2.3.1. Length Checking

The MAC function checks the frame and payload lengths of basic, VLAN tagged, and stacked VLAN tagged frames.



The IP core checks that the frame length is valid—is neither undersized nor oversized. A valid frame length is at least 64 (0x40) bytes and does not exceed the following maximum value for the different frame types:

- Basic frames—The number of bytes specified in the `MAX_RX_SIZE_CONFIG` register.
- VLAN tagged frames—The value specified in the `MAX_RX_SIZE_CONFIG` register plus four bytes.
- Stacked VLAN tagged frames—The value specified in the `MAX_RX_SIZE_CONFIG` register plus eight bytes.

If the length/type field in a basic MAC frame or the client length/type field in a VLAN tagged frame has a value less than 0x600, the IP core also checks the payload length. The IP core keeps track of the payload length as it receives a frame, and checks the length against the relevant frame field. The payload length is valid if it satisfies the following conditions:

The RX MAC does not drop frames with invalid length or invalid payload length. If the frame or payload length is not valid, the MAC function asserts output error bits.

If the length field value is greater than the actual payload length, the IP core asserts . If the length field value is less than the actual payload length, the MAC RX considers the frame to have excessive padding and does not assert .

#### 4.1.2.4. RX CRC Checking and Dynamic Forwarding

By default, the RX MAC strips off the CRC bytes before forwarding the packet to the MAC client. You can configure the core to retain the RX CRC and forward it to the client by updating the `MAC_CRC_CONFIG` register.

#### 4.1.3. Link Fault Signaling Interface

Link fault signaling reflects the health of the link. It operates between the remote Ethernet device Reconciliation Sublayer (RS) and the local Ethernet device RS. The link fault modules communicate status during the interframe period.

For unidirectional fault signaling, the core implements *Clause 66 of the IEEE 802.3-2012 Ethernet Standard*.

##### Local Fault (LF)

If an Ethernet PHY sublayer detects a fault that makes the link unreliable, it notifies the RS of the local fault condition. If unidirectional is not enabled, the core follows *Clause 46*. The RS stops sending MAC data, and continuously generates a remote fault status on the TX datapath. After a local fault is detected, the RX PCS modifies the MII data and control to send local fault sequence ordered sets. Refer to *Link Fault Signaling Based On Configuration and Status* below.

The RX PCS cannot recognize the link fault under the following conditions:

- The RX PCS is not fully aligned.
- The bit error rate (BER) is high.



### Remote Fault (RF)

If unidirectional is not enabled, the core follows *Clause 46*. If the RS receives a remote fault status, the TX datapath stops sending MAC data and continuously generates idle control characters. If the RS stops receiving fault status messages, it returns to normal operation, sending MAC client data. Refer to *Link Fault Signaling Based On Configuration and Status* below.

### Link Status Signals

The MAC RX generates two link fault signals: `local_fault_status` and `remote_fault_status`.

**Note:** These signals are real time status signals that reflect the status of the link regardless of the settings in the link fault configuration register.

This register is generated only if you turn on **Enable link fault generation**. The MAC TX interface uses the link fault status signals for additional link fault signaling.

**Table 12. Link Fault Signaling Based On Configuration and Status**

For more information about the `LINK_FAULT` register, refer to TX MAC Registers.

LINK_FAULT Register (0x405)				Real Time Link Status		Configured TX Behavior		Comment
Bit [0]	Bit [3]	Bit [1]	Bit [2]	LF Received	RF Received	TX Data	TX RF	
1'b0	Don't care	Don't care	Don't care	Don't care	Don't care	On	Off	Disable Link fault signaling on TX. RX still reports link status. TX side Link fault signaling disabled on the link. TX data and idle.
1'b1	1'b1	Don't care	Don't care	Don't care	Don't care	Off	On	Force RF. TX: Stop data. Transmit RF only
1'b1	1'b0	1'b1	1'b1	Don't care	Don't care	On	Off	Unidir: Backwards compatible. TX: Transmit data and idle. No RF.
1'b1	1'b0	1'b1	1'b0	1'b1	1'b0	On	On	Unidir: LF received. TX: Transmit data 1 column IDLE after end of packet and RF
1'b1	1'b0	1'b1	1'b0	1'b0	1'b1	On	Off	Unidir: RF receives TX: Transmit data and idle. No RF.
1'b1	1'b0	1'b1	1'b0	1'b0	1'b0	On	Off	Unidir: No link fault TX: Transmit data and idle. No RF.
1'b1	1'b0	1'b0	Don't care	1'b1	1'b0	Off	On	Bidir: LF received TX: Stop data. Transmit RF only.
1'b1	1'b0	1'b0	Don't care	1'b0	1'b1	Off	Off	Bidir: RF received TX: Stop data. Idle only. No RF.
1'b1	1'b0	1'b0	Don't care	1'b0	1'b0	On	Off	Bidir: No link fault TX: Transmit data and idle. No RF.

### Related Information

[IEEE website](#)

The Ethernet specifications are available on the IEEE website.

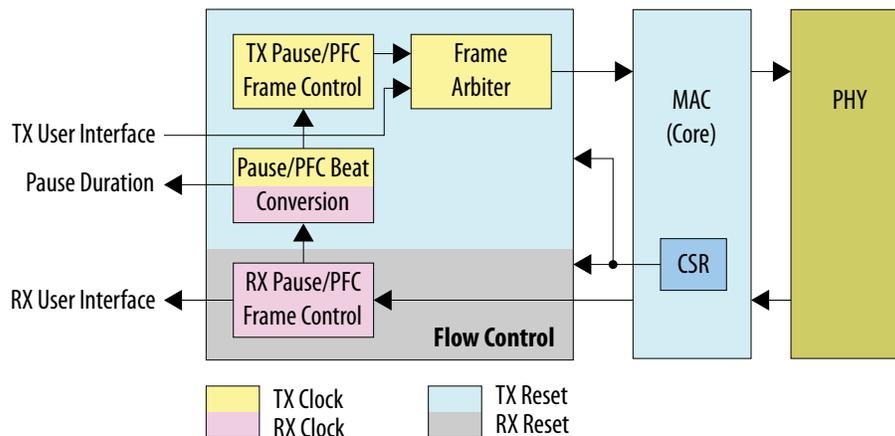
#### 4.1.4. Flow Control

Flow control reduces congestion at the local or remote link partner. When either link partner experiences congestion, the respective transmit control sends pause frames. XOFF Pause frames stop the remote transmitter. XON Pause frames let the remote transmitter resume data transmission. Flow control supports both Pause and Priority Flow Control (PFC) control frames:

- IEEE 802.3 flow control—implements the IEEE 802.3 Annex 31B standard to manage congestion. This flow control is a mechanism to manage congestion at the local or remote partner. When the receiving device experiences congestion, it sends an XOFF pause frame to the emitting device to instruct the emitting device to stop sending data for a duration specified by the congested receiver. Data transmission resumes when the emitting device receives an XON pause frame (pause quanta = zero) or when the timer expires.
- Priority-based flow control (PFC)—implements the IEEE 802.1Qbb standard. PFC manages congestion based on priority levels. It supports up to 8 priority queues. When the receiving device experiences congestion on a priority queue, it sends a PFC frame requesting the emitting device to stop transmission on the priority queue for a duration specified by the congested receiver. When the receiving device is ready to receive transmission on the priority queue again, it sends a PFC frame instructing the emitting device to resume transmission on the priority queue.

**Figure 10. Flow Control Module Conceptual Overview**

The flow control module acts as a buffer between client logic and the TX and RX MAC.



Flow Control includes the following features:

- Pause or PFC frame generation and transmission:
  - Configurable selection of standard or priority-based flow control
  - Programmable 1- or 2-bit XON/XOFF request mode
  - In 2-bit request mode, programmable selection of register or signal-based control
  - Programmable per-queue XOFF frame separation
  - Programmable destination and source addresses in outgoing pause and PFC frames
  - Programmable pause and PFC quanta
- Client versus Pause or PFC frame transmission based on a priority-based arbitration scheme with frame-type indication for external downstream logic
- Stopping the next client frame transmission on the reception of a valid Pause frame
- Stopping the per queue client frame transmission on the reception of a valid PFC frame from the client. Includes per-queue PFC Pause quanta duration indicator
- Pause or PFC frame reception and decode:
  - Programmable destination address for filtering incoming pause and PFC frames
  - Configurable Pause or PFC per-queue enable, directing the IP core to ignore incoming pause frames on disabled queues
  - Per-queue client frame transmission pause duration indicator

#### 4.1.4.1. TX Pause/PFC Flow Control Transmission

An XON/XOFF request triggers the IP core to transmit a Pause or PFC flow control frame on the Ethernet link. You can control XON/XOFF requests using the TX flow control registers or the `pause_insert_tx0` and `pause_insert_tx1` input signals.

You can specify whether the IP core accepts XON/XOFF requests in 1-bit or 2-bit format by updating the TX Flow Control CSR XON/XOFF Request register field. By default, the IP core assumes 1-bit requests.

#### 4.1.4.2. XON/XOFF Pause Frames

The sender transmits a PFC frame with the specified PFC pause quanta value when it receives an XOFF request. If an enabled priority queue is in the XOFF condition, a new PFC frame is transmitted after the minimum time gap. You specify the minimum time gap in the per priority queue TX Flow Control Signal XOFF Request Hold Quanta register. The minimum time gap between two consecutive PFC frames is 1 pause quanta or 512-bit times. PFC frame transmission ends when none of the PFC interfaces of all enabled priority queues is requesting PFC frames.

A transition from XOFF to XON in any enabled priority queue triggers the IP core to transmit a PFC frame with pause quanta of 0 for the associated priority queue. The IP core sends a single XON flow control frame. In the rare case that the XON frame is lost or corrupted, the remote partner should still be able to resume transmission. The remote partner resumes transmission after the duration specified in the previous XOFF flow control frame expires.



In the case of standard flow control, the IP core transmits Pause frames instead of PFC frames. The transmission behavior is identical.

When the IP core is in standard flow control mode and receives a Pause frame, the IP core stops processing TX client data, either immediately or at the next frame boundary. Client data transmission resumes when all of the following conditions are true:

- The time specified by the pause quanta has elapsed and there is no new quanta value
- A valid pause frame with 0 pause duration has been received

A Pause frame has no effect if the associated TX Flow Control Enable register bit is set to disable XON and XOFF flow control.

## 4.2. 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 IP core transmits 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.

### 4.2.1. 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. On the transmit client interface, the IP core expects the client to send the most significant bytes of the frame first, and to send each byte in big-endian format. Similarly, on the receive client interface, the IP core sends the client the most significant bytes of the frame first, and orders each byte in big-endian format.

**Figure 11. Byte Order on the Client Interface Lanes**

Describes the byte order on the Avalon streaming interface. 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 5 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]).



### **4.2.2. Bit Order For TX and RX Datapaths**

The TX bit order matches the placement shown in the PCS lanes as illustrated in *IEEE Standard for Ethernet, Section 4, Figure 49-5*. The RX bit order matches the placement shown in *IEEE Standard for Ethernet, Section 4, Figure 49-6*.

#### **Related Information**

[IEEE website](#)

The *IEEE Standard for Ethernet, Section 4* is available on the IEEE website.

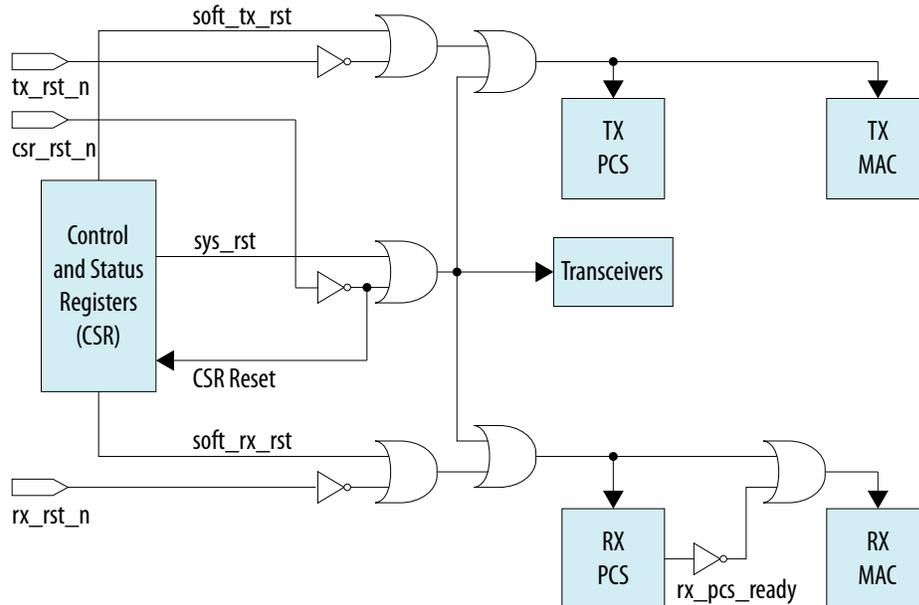
## 5. Reset

Control and Status registers control three parallel soft resets. These soft resets are not self-clearing. Software clears them by writing to the appropriate register. In addition, the IP core has three hard reset signals.

Asserting the external hard reset `csr_rst_n` returns all Control and Status registers to their original values, except the statistics counters. An additional dedicated reset signal resets the transceiver reconfiguration interface.

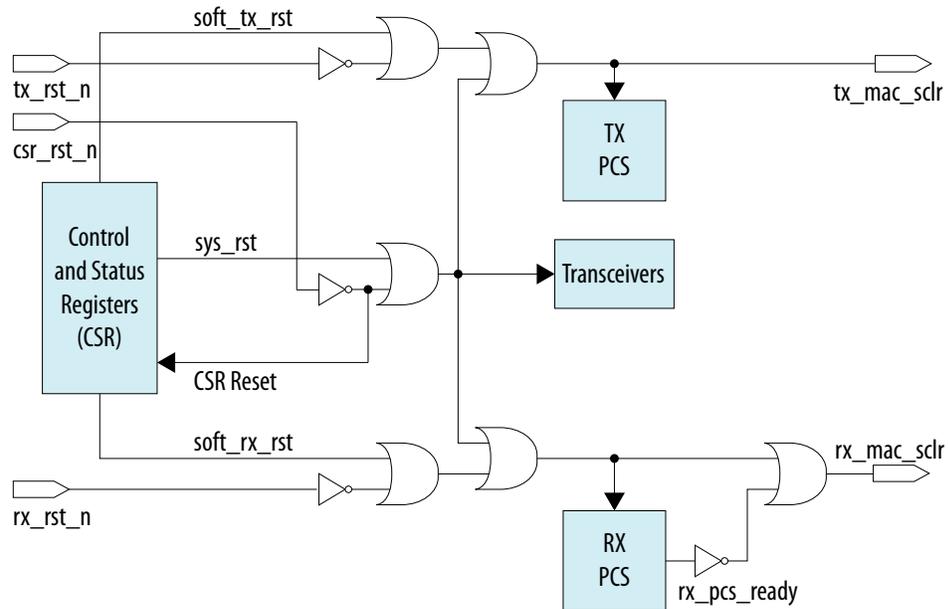
### Figure 12. Conceptual Overview of General IP Core Reset Logic: MAC with PCS variant

The three hard resets are top-level ports. The three soft resets are internal signals which are outputs of the `PHY_CONFIG` register. Software writes the appropriate bit of the `PHY_CONFIG` to assert a soft reset.



**Figure 13. Conceptual Overview of General IP Core Reset Logic: PCS Variant**

Two additional resets are top-level ports in PCS variant.



The general reset signals reset the following functions:

- `soft_tx_rst`, `tx_rst_n`: Resets the IP core in the TX direction. Resets the TX PCS and TX MAC. This reset leads to deassertion of the `tx_lanes_stable` output signal.
- `soft_rx_rst`, `rx_rst_n`: Resets the IP core in the RX direction. Resets the RX PCS and RX MAC. This reset leads to deassertion of the `rx_pcs_ready` output signal.
- `sys_rst`, `csr_rst_n`: Resets the IP core. Resets the TX and RX MACs, PCS, and transceivers.

*Note:* `csr_rst_n` resets the Control and Status registers, except the statistics counters. `sys_rst` does not reset any Control and Status registers.

This reset leads to deassertion of the `tx_lanes_stable` and `rx_pcs_ready` output signals.

- `tx_mac_sclr`: Available in PCS only variant, this reset signal resets the user TX MAC. The user MAC is not part of the Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP.
- `rx_mac_sclr`: Available in PCS only variant, this reset signal resets the user RX MAC. The user MAC is not part of the Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP.

In addition, the synchronous `reconfig_reset` signal resets the IP core transceiver reconfiguration interface, an Avalon memory-mapped interface. Associated clock is the `reconfig_clk`, which clocks the transceiver reconfiguration interface.

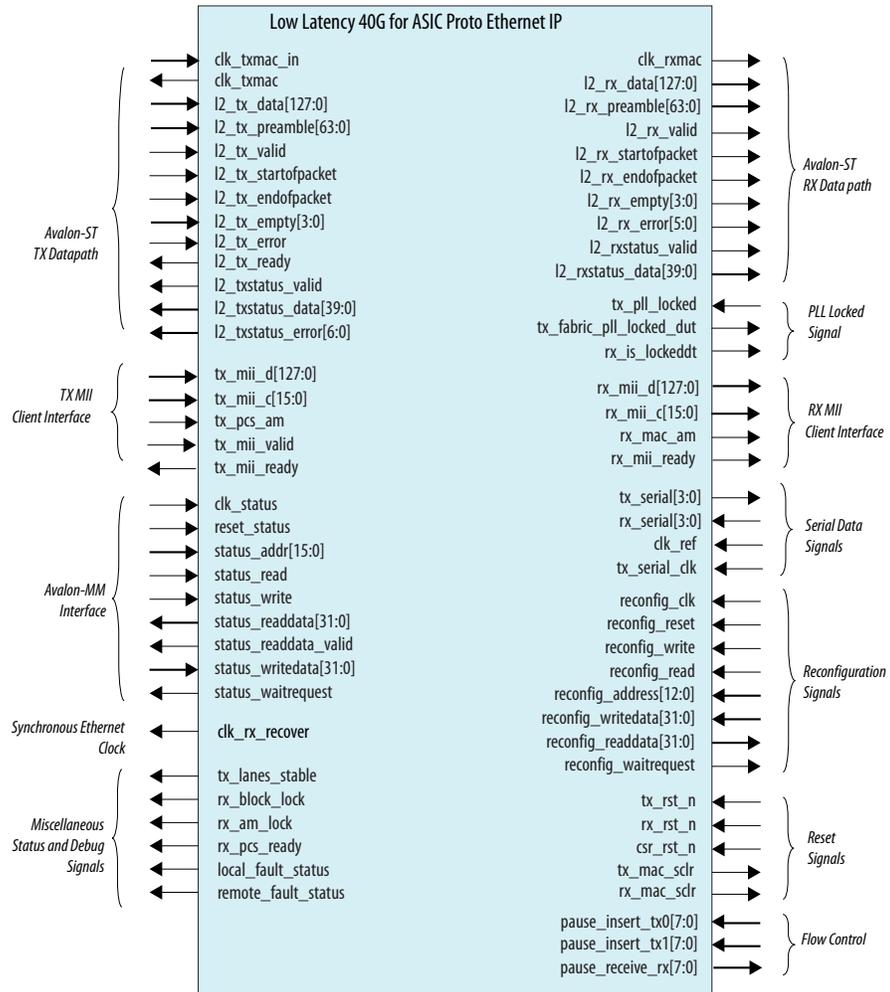
### Related Information

[Reset Signals](#) on page 42



## 6. Interfaces and Signal Descriptions

Figure 14. Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Signals and Interfaces



### 6.1. TX MAC Interface to User Logic

The TX MAC provides an Avalon streaming interface to the FPGA fabric. The datapath comprises 2, 64-bit words. The minimum packet size is nine bytes.



**Table 13. Avalon Streaming TX MAC Interface Signals**

All interface signals are clocked by the `clk_txmac` clock. The value you specify for **Ready latency** in the Low Latency 40G for ASIC Proto Ethernet parameter editor is the Avalon streaming interface readyLatency value on this interface.

Signal	Direction	Description
<code>clk_txmac</code>	Output	The TX clock for the IP core is <code>clk_txmac</code> . The frequency of this clock is 312.5 MHz. If you turn on <b>Use external TX MAC PLL</b> in the Low Latency 40G for ASIC Proto Ethernet parameter editor, the <code>clk_txmac_in</code> input clock drives <code>clk_txmac</code> .
<code>l2_tx_data[127:0]</code>	Input	Data input to MAC. Bit 127 is the MSB and bit 0 is the LSB. Bytes are read in the usual left to right order.
<code>l2_tx_preamble[63:0]</code>	Input	User preamble data. Available when you turn on <b>Enable preamble passthrough</b> in the Low Latency 40G for ASIC Proto Ethernet parameter editor. User logic drives the custom preamble data when <code>l2_tx_startofpacket</code> is asserted.
<code>l2_tx_valid</code>	Input	When asserted, indicates valid data.
<code>l2_tx_startofpacket</code>	Input	When asserted, indicates the first byte of a frame. When <code>l2_tx_startofpacket</code> is asserted, the MSB of <code>l2_tx_data</code> drives the start of packet. Packets that drive <code>l2_tx_startofpacket</code> and <code>l2_tx_endofpacket</code> in the same cycle are ignored.
<code>l2_tx_endofpacket</code>	Input	When asserted, indicates the end of a packet. Packets that drive <code>l2_tx_startofpacket</code> and <code>l2_tx_endofpacket</code> in the same cycle are ignored.
<code>l2_tx_empty[3:0]</code>	Input	Specifies the number of empty bytes when <code>l2_tx_endofpacket</code> is asserted.
<code>l2_tx_ready</code>	Output	When asserted, indicates that the MAC can accept the data. The IP core asserts the <code>l2_tx_ready</code> signal on clock cycle $\langle n \rangle$ to indicate that clock cycle $\langle n + \text{readyLatency} \rangle$ is a ready cycle. The client may only assert <code>l2_tx_valid</code> and transfer data during ready cycles.
<code>l2_tx_error</code>	Input	When asserted in an EOP cycle (while <code>l2_tx_endofpacket</code> is asserted), directs the IP core to insert an error in the packet before sending it on the Ethernet link. <i>Note:</i> This functionality is not available in the Quartus Prime Pro 17.1 Stratix 10 ES Editions software.
<code>l2_txstatus_valid</code>	Output	When asserted, indicates that <code>l2_txstatus_data</code> and <code>l2_txstatus_error[6:0]</code> are driving valid data.
<code>l2_txstatus_data[39:0]</code>	Output	Specifies information about the transmit frame. The following fields are defined: <ul style="list-style-type: none"> <li>[Bit 39]: When asserted, indicates a PFC frame</li> <li>[Bit 38]: When asserted, indicates a unicast frame</li> <li>Bit[37]: When asserted, indicates a multicast frame</li> <li>Bit[36]: When asserted, indicates a broadcast frame</li> <li>Bit[35]: When asserted, indicates a pause frame</li> <li>Bit[34]: When asserted, indicates a control frame</li> <li>Bit[33]: When asserted, indicates a VLAN frame</li> </ul>

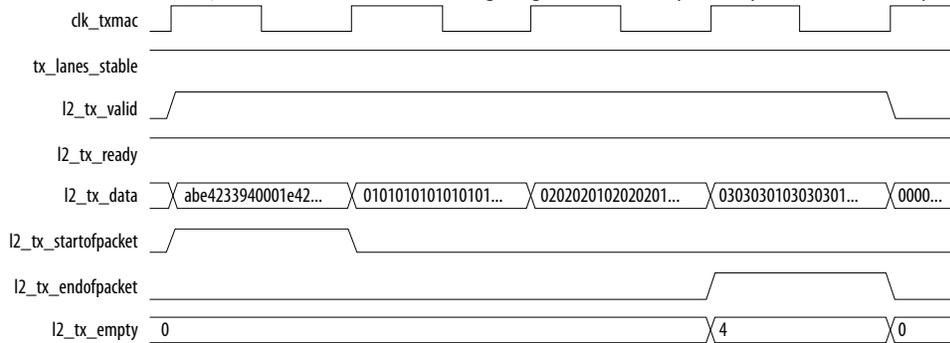
*continued...*



Signal	Direction	Description
		<ul style="list-style-type: none"> <li>Bit[32]: When asserted, indicates a stacked VLAN frame</li> <li>Bits[31:16]: Specifies the frame length from the first byte of the destination address to the last byte of the FCS</li> <li>Bits[15:0]: Specifies the payload length</li> </ul>
l2_txstatus_error[6:0]	Output	Specifies the error type in the transmit frame. The following fields are defined: <ul style="list-style-type: none"> <li>Bits[6:3]: Reserved</li> <li>Bit[2]: Payload length error</li> <li>Bit[1]: Oversized frame</li> <li>Bit[0]: Reserved.</li> </ul>

**Figure 15. Client to MAC Avalon streaming interface**

l2\_tx\_data reception order is highest byte to lowest byte. The first byte of the destination address is on l2\_tx\_data[127:120], 0xab4233... in this timing diagram. The ready latency is 0 in this example.



### Related Information

#### Avalon Interface Specifications

Detailed information about Avalon streaming interface and the Avalon streaming interface readyLatency parameter.

## 6.2. RX MAC Interface to User Logic

The RX MAC provides an Avalon streaming interface to the FPGA fabric. The datapath comprises 2, 64-bit words.

**Table 14. Avalon Streaming RX MAC Interface Signals**

All interface signals are clocked by the `clk_rxmac` clock.

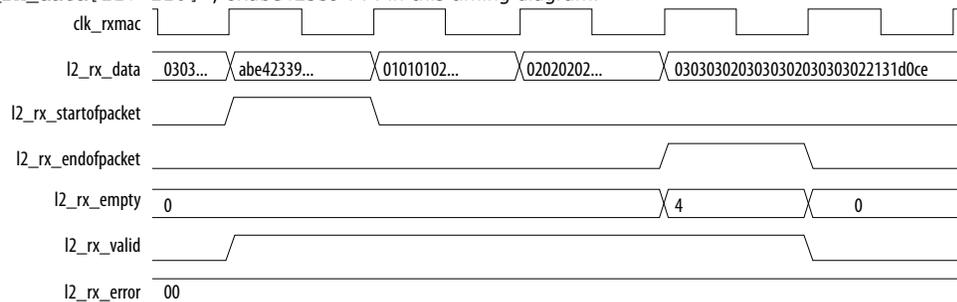
Signal	Direction	Description
clk_rxmac	Output	Clock for the RX MAC. Recovered from the incoming data. This clock is guaranteed stable when <code>rx_pcs_ready</code> is asserted. The frequency of this clock is 312.5 MHz. All RX MAC interface signals are synchronous to <code>clk_rxmac</code> .
l2_rx_data[127:0]	Output	Data output from the MAC. Bit 127 is the MSB and bit 0 is the LSB. Bytes are read in the usual left to right order. The IP core reverses the byte order to meet the requirements of the Ethernet standard.
l2_rx_preamble[63:0]	Output	Received preamble data. Available when you select <code>PREAMBLE_PASS-THROUGH</code> mode. Valid when <code>l2_rx_startofpacket</code> is asserted.

*continued...*

Signal	Direction	Description
l2_rx_valid	Output	When asserted, indicates that l2_rx_data[127:0] is driving data.
l2_rx_startofpacket	Output	When asserted, indicates the first byte of a frame.
l2_rx_endofpacket	Output	When asserted, indicates the last data byte of a frame, before the frame check sequence (FCS). In CRC pass-through mode, it is the last byte of the FCS. The packet can end at any byte position.
l2_rx_empty[3:0]	Output	Specifies the number of empty bytes when l2_rx_endofpacket is asserted. The packet can end at any byte position. The empty bytes are the low-order bytes.
l2_rx_error[5:0]	Output	When asserted in the same cycle as l2_rx_endofpacket, indicates the current packet should be treated as an error packet. The 6 bits of l<n>_rx_error specify the following errors: <ul style="list-style-type: none"> <li>l2_rx_error[5]: Unused.</li> <li>l2_rx_error[4]: Payload length error. The length field has a value less than 1535 (0x600) bytes and the received payload length is less than what is advertised in the length field.</li> <li>l2_rx_error[3]: Oversized frame. The frame size is greater than the value specified in the MAX_RX_SIZE_CONFIG register.</li> <li>l2_rx_error[2]: Undersized frame - The frame size is less than 64 bytes. Frame size = header size + payload size.</li> <li>l2_rx_error[1]: CRC Error. The computed CRC value differs from the received CRC.</li> <li>l2_rx_error[0]: Malformed packet. The packet is terminated with a non-terminate control character. When this bit is asserted, l2_rx_error[1] is also asserted.</li> </ul>
l2_rxstatus_valid	Output	When asserted, indicates that l2_rxstatus_data is driving valid data.
l2_rxstatus_data[39:0]	Output	Specifies information about the received frame. The following fields are defined: <ul style="list-style-type: none"> <li>[Bit 39]: When asserted, indicates a PFC frame</li> <li>[Bit 38]: When asserted, indicates a unicast frame</li> <li>Bit[37]: When asserted, indicates a multicast frame</li> <li>Bit[36]: When asserted, indicates a broadcast frame</li> <li>Bit[35]: When asserted, indicates a pause frame</li> <li>Bit[34]: When asserted, indicates a control frame</li> <li>Bit[33]: When asserted, indicates a VLAN frame</li> <li>Bit[32]: When asserted, indicates a stacked VLAN frame</li> <li>Bits[31:16]: Specifies the frame length from the first byte of the destination address to the last byte of the FCS</li> <li>Bits[15:0]: Specifies the payload length</li> </ul>

**Figure 16. MAC to Client Avalon Streaming Interface**

l2\_rx\_data reception order is highest byte to lowest byte. The first byte of the destination address is on l2\_rx\_data[127:120], 0xabe42339... in this timing diagram.





### 6.3. TX PCS Interface to User Logic

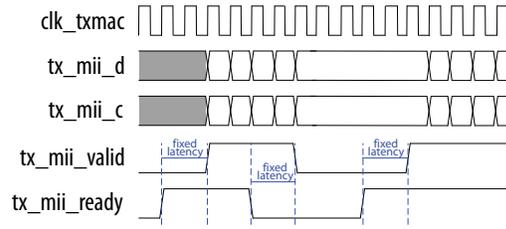
The Low Latency 40G for ASIC Proto Ethernet IP core TX client interface in PCS Only variations employs the Media Independent Interface (MII) protocol.

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

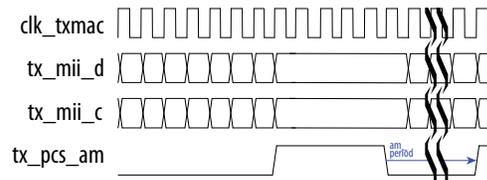
**Table 15. Signals of the MII TX Client Interface**

Signal Name	Description
clk_txmac	The TX clock for the IP core is <code>clk_txmac</code> . The frequency of this clock is 312.5 MHz. If you turn on <b>Use external TX MAC PLL</b> parameter, the <code>clk_txmac_in</code> input clock drives <code>clk_txmac</code> .
tx_mii_d[127:0]	TX MII input data from MAC to PCS. Data must be in MII encoding. The IP core handles two data blocks per clock cycle.
tx_mii_c[15:0]	TX MII control bits. Each bit corresponds to a byte of <code>i_tx_mii_d</code> . <code>i_tx_mii_c[0]</code> corresponds to <code>i_tx_mii_d[7:0]</code> , <code>i_tx_mii_c[1]</code> corresponds to <code>i_tx_mii_d[15:8]</code> , and so on. The IP core handles two data blocks per clock cycle.
tx_mii_valid	Indicates that <code>tx_mii_d</code> is valid. You must assert this signal a fixed number of clock cycles after the IP core raises <code>tx_mii_ready</code> , and must deassert this signal the same number of clock cycles after the IP core deasserts <code>tx_mii_ready</code> . The number must be in the range of 1-10 clock cycles. While you hold the value of this signal at 0, you must freeze the values of both <code>tx_mii_d</code> and <code>tx_mii_c</code> stable.
tx_mii_ready	Indicates the PCS is ready to receive new data.
tx_pcs_am	Alignment marker insertion bit. You must hold this signal asserted for 2 <code>clk_txmac</code> consecutive clock cycles, counting only the valid cycles, to drive the insertion of an alignment marker. A valid cycle is one in which <code>tx_mii_valid</code> has the value of 1. The number of valid clock cycles from deassertion of <code>tx_pcs_am</code> (alignment marker insertion bit signal) to reassertion of <code>tx_pcs_am</code> is the <code>am_period</code> . <ul style="list-style-type: none"> <li>For normal operation of the Ethernet link, you must ensure that the value of <code>am_period</code> is 32766 clock cycles.</li> <li>In simulation you can reduce this value to 510. This change decreases the simulation time to RX PCS alignment. You can set the IP core to expect this interval by setting the <code>sim_mode</code> RTL parameter to <code>Enable</code>.</li> </ul> <p><i>Note:</i> The value for the MAC+PCS variations is different, to ensure the value on the internal signal has the appropriate <code>am_period</code>.</p> <p>For an example that handles this setting for simulation and drives the <code>tx_pcs_am</code> signal appropriately for simulation, refer to the IP core design example for PCS Only variations. For information about how to generate the IP core design example, refer to the <i>Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Design Example User Guide</i>. For information about the <code>sim_mode</code> RTL parameter, refer to the <i>RTL Parameters</i> section of this user guide.</p> <p>While you hold the value of this signal at 1, you must freeze the values of both <code>tx_mii_d</code> and <code>tx_mii_c</code>.</p>

**Figure 17. TX MII Client Interface for Low Latency 40G for ASIC Proto Ethernet IP Core**



**Figure 18. Alignment Marker Insertion on the TX MII Client Interface**



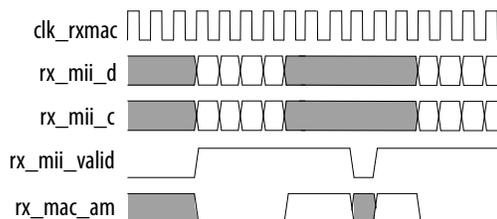
### 6.4. RX PCS Interface to User Logic

The Low Latency 40G for ASIC Proto Ethernet IP core RX client interface in PCS Only variations employs the Media Independent Interface (MII) protocol.

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

**Table 16. Signals of the MII RX Client Interface**

Signal Name	Description
clk_rxmac	The RX MAC clock. This clock is recovered from the incoming data. The frequency of this clock is 312.5 MHz. All RX MAC interface signals are synchronous to clk_rxmac.
rx_mii_d[127:0]	RX MII data. Data is in MII encoding. o_rx_mii_d[7:0] holds the first byte the IP core received on the Ethernet link. o_rx_mii_d[0] holds the first bit the IP core received on the Ethernet link. When o_rx_mii_valid has the value of 0 or o_rx_mac_am has the value of 1, the value on o_rx_mii_d is invalid.
rx_mii_c[15:0]	RX MII control bits. Each bit corresponds to a byte of rx_mii_d. rx_mii_c[0] corresponds to rx_mii_d[7:0], rx_mii_c[1] corresponds to rx_mii_d[15:8], and so on. If the value of a bit is 1, the corresponding data byte is a control byte. If the value of a bit is 0, the corresponding data byte is data. The Start of Packet byte (0xFB), End of Packet byte (0xFD), Idle bytes (0x07), and error byte (0xFE) are control bytes, but the preamble bytes, Start of Frame (SFD) byte (0xD5), CRC bytes, and payload bytes are data bytes. When rx_mii_valid has the value of 0 or rx_mac_am has the value of 1, the value on rx_mii_c is invalid.
rx_mii_valid	Indicates that rx_mii_d, rx_mii_c, and rx_mac_am are valid.
rx_mac_am	Indicates the IP core received a valid alignment marker on the Ethernet link. When rx_mii_valid has the value of 0, the value on rx_mac_am is invalid. The value of rx_mii_valid may fall while the IP core is asserting rx_mac_am.


**Figure 19. RX MII Client Interface for Low Latency 40G for ASIC Proto Ethernet IP Core**


## 6.5. Transceivers

The transceivers require a separately instantiated advanced transmit (ATX) PLL to generate the high speed serial clock. For Low Latency 40G for ASIC Proto Ethernet IP core, you can use the same ATX PLL for all four transceivers. In many cases, the same ATX PLL can serve as input to additional transceivers that have similar input clocking requirements. In comparison to the fractional PLL (fPLL) and clock multiplier unit PLL, the ATX PLL has the best jitter performance and supports the highest frequency operation.

**Table 17. Transceiver Signals**

Signal	Direction	Description
tx_serial[3:0]	Output	TX transceiver data. Each tx_serial bit becomes two physical pins that form a differential pair.
rx_serial[3:0]	Input	RX transceiver data. Each rx_serial bit becomes two physical pins that form a differential pair.
clk_ref	Input	The PLL reference clock. Input to the clock data recovery (CDR) circuitry in the RX PMA. The frequency of this clock is .

## 6.6. Transceiver Reconfiguration Signals

You access the transceiver control and status registers using the transceiver reconfiguration interface. This is an Avalon memory-mapped interface.

The Avalon memory-mapped interface implements a standard memory-mapped protocol. You can connect an Avalon master to this bus to access the registers of the embedded Transceiver PHY IP core.

**Table 18. Reconfiguration Interface Ports with Shared Native PHY Reconfiguration Interface**

All interface signals are clocked by the reconfig\_clk clock.

Port Name	Direction	Description
reconfig_clk	Input	Avalon clock. The clock frequency is 100-161 MHz. <b>All signals transceiver reconfiguration interface signals are synchronous to reconfig_clk.</b>
reconfig_reset	Input	Resets the Avalon memory-mapped interface and all of the registers to which it provides access.
reconfig_write	Input	Write enable signal. Signal is active high.
reconfig_read	Input	Read enable signal. Signal is active high.

*continued...*



Port Name	Direction	Description
reconfig_address[12:0]	Input	Address bus.
reconfig_writedata[31:0]	Input	A 32-bit data write bus. reconfig_address specifies the address.
reconfig_readdata[31:0]	Output	A 32-bit data read bus. Drives read data from the specified address. Signal is valid after reconfig_waitrequest is deasserted.
reconfig_waitrequest	Output	Indicates the Avalon memory-mapped interface is busy. Keep the reconfig_write or reconfig_read asserted until reconfig_waitrequest is deasserted.



## 6.7. Avalon Memory-Mapped Management Interface

You access control and status registers using an Avalon memory-mapped management interface. The interface responds regardless of the link status. It also responds when the IP core is in a reset state driven by any reset signal or soft reset other than the `csr_rst_n` signal. Asserting the `csr_rst_n` signal resets all statistics registers; while this reset is in process, the Avalon memory-mapped management interface does not respond.

**Table 19. Avalon Memory-Mapped Management Interface**

*Note:* All `status_*` signals are synchronous to `clk_status` signal.

Signal	Direction	Description
<code>clk_status</code>	Input	The clock that drives the control and status registers. The frequency of this clock is 100-161 MHz.
<code>status_addr[15:0]</code>	Input	Drives the Avalon memory-mapped register address.
<code>status_read</code>	Input	When asserted, specifies a read request.
<code>status_write</code>	Input	When asserted, specifies a write request.
<code>status_readdata[31:0]</code>	Output	Drives read data. Valid when <code>status_readdata_valid</code> is asserted.
<code>status_readdata_valid</code>	Output	When asserted, indicates that <code>status_read_data[31:0]</code> is valid.
<code>status_writedata[31:0]</code>	Input	Drives the write data. The packet can end at any byte position. The empty bytes are the low-order bytes.
<code>status_waitrequest</code>	Output	Indicates that the control and status interface is not ready to complete the transaction. <code>status_waitrequest</code> is only used for read transactions.

### Related Information

[Typical Read and Write Transfers section in the \*Avalon Interface Specifications\*](#)

Describes typical Avalon memory-mapped read and write transfers with a slave-controlled waitrequest signal.

## 6.8. Miscellaneous Status and Debug Signals

The miscellaneous status and debug signals are asynchronous.

**Table 20. Avalon Memory-Mapped Interface**

Signal	Direction	Description
<code>tx_lanes_stable</code>	Output	Asserted when all TX lanes are stable and ready to transmit data.
<code>rx_block_lock</code>	Output	Asserted when all lanes have identified 66-bit block boundaries in the serial data stream.
<code>rx_am_lock</code>	Output	Asserted when all lanes have identified alignment markers in the data stream.
<code>rx_pcs_ready</code>	Output	Asserted when the RX lanes are fully aligned and ready to receive data.
<code>local_fault_status</code>	Output	Asserted when the RX MAC detects a local fault. This signal is available only if you turn on <b>Enable link fault generation</b> in the parameter editor.

*continued...*



Signal	Direction	Description
remote_fault_status	Output	Asserted when the RX MAC detects a remote fault. This signal is available only if you turn on <b>Enable link fault generation</b> in the parameter editor.
tx_fabric_pll_locked_dut	Output	This signal indicates that all ATX PLL(s) are locked. This signal is available only if you turn on <b>Select USER MAC mode</b> and select <b>PCS_Only</b> option in the parameter editor.
rx_is_lockeddt	Output	This signal indicates that RX PLL synchronized locked status output to reset the user MAC. This signal is available only if you turn on <b>Select USER MAC mode</b> and select <b>PCS_Only</b> option in the parameter editor.

## 6.9. Reset Signals

The IP core has three external hard reset inputs. These resets are asynchronous and are internally synchronized. Assert resets for ten `clk_txmac` clock cycles or until you observe the effect of their specific reset. Asserting the external hard reset `csr_rst_n` returns control and status registers to their original values, with the exception of the statistics counters. `rx_pcs_ready` and `tx_lanes_stable` are asserted when the IP core has exited reset successfully.

**Table 21. Reset Signals**

Signal	Direction	Description
tx_rst_n	Input	Active low hard reset signal. Resets the TX interface, including the TX PCS and TX MAC. This reset leads to the deassertion of the <code>tx_lanes_stable</code> output signal.
rx_rst_n	Input	Active low hard reset signal. Resets the RX interface, including the RX PCS and RX MAC. This reset leads to the deassertion of the <code>rx_pcs_ready</code> output signal.
csr_rst_n	Input	Active low hard global reset. Resets the full IP core. Resets the TX MAC, RX MAC, TX PCS, RX PCS, transceivers, and control and status registers, except the statistics counters. This reset leads to the deassertion of the <code>tx_lanes_stable</code> and <code>rx_pcs_ready</code> output signals.
tx_mac_sclr	Output	Soft reset in PCS variations. Resets the TX interface, including the TX PCS.
rx_mac_sclr	Output	Soft reset in PCS variations. Resets the RX interface, including the RX PCS.

### Related Information

[Reset](#) on page 31

## 6.10. Clocks

You must set the transceiver reference clock (`clk_ref`) frequency to a value that the IP core supports. The Low Latency 40G for ASIC Proto Ethernet IP core supports a `clk_ref` frequency of 644.53125 MHz  $\pm$ 100 ppm or 322.265625 MHz  $\pm$ 100 ppm. The  $\pm$ 100ppm value is required for any clock source providing the transceiver reference clock.

SyncE 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.



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 Low Latency 40G for ASIC Proto Ethernet IP core performs the filtering.

**Table 22. Clock Inputs**

Describes the input clocks that you must provide.

Signal Name	Description
clk_ref	The input clock <code>clk_ref</code> is the reference clock for the transceiver RX CDR PLL. This clock must have a frequency of 644.53125 MHz or 322.265625 MHz with a $\pm 100$ ppm accuracy per the <i>IEEE 802.3ba-2010 Ethernet Standard</i> . In addition, <code>clk_ref</code> must meet the jitter specification of the <i>IEEE 802.3ba-2010 Ethernet Standard</i> . 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.
clk_txmac_in	If you turn on <b>Use external TX MAC PLL</b> in the Low Latency 40G for ASIC Proto Ethernet 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 <code>clk_txmac</code> . The required TX MAC clock frequency is 312.5 MHz. User logic must drive <code>clk_txmac_in</code> from a PLL whose input is the PHY reference clock, <code>clk_ref</code> .
tx_serial_clk	This input clock is part of the external PLL interface. The IP core fans out the clock to target each of the four transceiver PHY links. You must drive this clock from a single TX transceiver PLL that you configure separately from the Low Latency 40G for ASIC Proto Ethernet IP core. The required frequency is 5156.25 MHz.
clk_status	Clocks the control and status interface. The clock quality and pin chosen are not critical. <code>clk_status</code> is expected to be a 100–161 MHz clock.
reconfig_clk	Clocks the transceiver reconfiguration interface. The clock quality and pin chosen are not critical. <code>reconfig_clk</code> is expected to be a 100–161 MHz clock.

**Table 23. 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 <code>clk_txmac</code> . The TX MAC clock frequency is 312.5 MHz. If you turn on <b>Use external TX MAC PLL</b> in the Low Latency 40G for ASIC Proto Ethernet parameter editor, the <code>clk_txmac_in</code> input clock drives <code>clk_txmac</code> .
clk_rxmac	The RX clock for the IP core is <code>clk_rxmac</code> . The RX MAC clock frequency is 312.5 MHz. This clock is only reliable when <code>rx_pcs_ready</code> has the value of 1. The IP core generates <code>clk_rxmac</code> 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 <b>Enable SyncE</b> in the Low Latency 40G for ASIC Proto Ethernet parameter editor. The RX recovered clock frequency is 156.25 MHz during normal operation. The expected usage is that you drive the TX transceiver PLL reference clock with a filtered version of <code>clk_rx_recover</code> , 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.

## 6.11. Flow Control Interface

Flow Control Interface signals become available when you turn on **Enable MAC Flow Control** in the parameter editor.

**Table 24. Flow Control Signals**

Signal Name	Direction	Description
<p>pause_insert_tx0[(FCQN-1):0]</p> <p>pause_insert_tx1[(FCQN-1):0]</p>	Input	<p>This signal is available if you specify pause on PFC.</p> <p>The signal indicates to the MAC whether XON or XOFF Pause or PFC flow control frame should be sent.</p> <ul style="list-style-type: none"> <li>FCQN = 1 for Pause</li> <li>FCQN = 1 ~ 8 for PFC</li> </ul> <p>The request for XON/XOFF flow control frame transmission can be done in either 1 or 2-bit request mode.</p> <p><b>1-bit mode request model:</b></p> <p>The IP core ignores pause_insert_tx1[(FCQN-1):0].</p> <p>The following encoding is defined:</p> <ul style="list-style-type: none"> <li>0: No request</li> <li>0 to 1: Generate XOFF request</li> <li>1: Continue to generate XOFF request</li> <li>1 to 0: Generate XON request</li> </ul> <p><b>2-bit mode request model:</b></p> <p>The higher-order bit is in pause_insert_tx1[(FCQN-1):0] and the lower-order bit is in pause_insert_tx0[(FCQN-1):0].</p> <p>The XON/XOFF request is a level-based request.</p> <p>The following encoding is defined:</p> <ul style="list-style-type: none"> <li>00: No further XON/XOFF request. In an XON/XOFF flow control frame is in progress, it is sent.</li> <li>01: Generate XON flow control frame and continuously sends them thereafter</li> <li>10: Generate XOFF flow control frame and continuously sends them thereafter</li> <li>11: Invalid request</li> </ul>
<p>pause_receive_rx[(FCQN-1):0]</p>	Output	<p>Each pause_receive_rx[(FCQN-1):0] bit indicates the corresponding queue is being paused.</p> <p>This is a level-based signal.</p>

## 7. Control, Status, and Statistics Register Descriptions

This section provides information about the memory-mapped registers. You access these registers using the IP core Avalon memory-mapped 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 result. Write operations to Reserved registers have no effect. Accesses to registers that do not exist in your IP core variation, or to register bits that are not defined in your IP core variation, have an unspecified result. You should consider these registers and register bits Reserved. Although you can only access registers in 32-bit read and write operations, you should not attempt to write or ascribe meaning to values in undefined register bits.

**Table 25. Register Base Addresses**

Word Offset	Register Type
0x300-0x3FF	PHY registers
0x400-0x4FF	TX MAC registers
0x500-0x5FF	RX MAC registers
0x600-0x6FF	TX Flow Control registers
0x700-0x7FF	RX Flow Control registers
0x800-0x8FF	Statistics Counter registers - TX direction
0x900-0x9FF	Statistics Counter registers - RX direction

### 7.1. PHY Registers

**Table 26. PHY Registers**

The global hard reset `csr_rst_n` resets all of these registers. The TX reset `tx_rst_n` and RX reset `rx_rst_n` signals do not reset these registers.

Addr	Name	Description	Reset	Access
0x300	REVID	IP core PHY module revision ID.	0x0627 2016	RO
0x301	SCRATCH	Scratch register available for testing.	0x0000 0000	RW
0x302	PHY_NAME_0	First characters of IP core variation identifier string, "0040". The "00" is unprintable.	0x0000 3430	RO
<i>continued...</i>				

(1) X means "Don't Care".

(2) Register value convert in decimal.

Intel Corporation. All rights reserved. Agilix, Altera, Arria, Cyclone, Enpirion, Intel, the Intel logo, 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.



Addr	Name	Description	Reset	Access
0x303	PHY_NAME_1	Next characters of IP core variation identifier string, "00GE". The "00" is unprintable.	0x0000 4745	RO
0x304	PHY_NAME_2	Final characters of IP core variation identifier string, "0pcs". The "0" is unprintable.	0x0070 6373	RO
0x310	PHY_CONFIG	PHY configuration registers. The following bit fields are defined: <ul style="list-style-type: none"> <li>• Bit[0]: : <code>sys_rst</code>. Full system reset (except registers). Set this bit to initiate the internal reset sequence.</li> <li>• Bit[1]: <code>soft_txp_rst</code>. TX soft reset. Resets TX PCS and TX MAC.</li> <li>• Bit[2]: <code>soft_rxp_rst</code>. RX soft reset. Resets RX PCS and RX MAC.</li> <li>• Bits[31:3]: Reserved.</li> </ul>	29'hX_3'b0 <sup>(1)</sup>	RW
0x312	WORD_LOCK	When asserted, indicates that the virtual channel has identified 66 bit block boundaries in the serial data stream.	28'hX4'b0 <sup>(1)</sup>	RO
0x313	EIO_SLOOP	Serial PMA loopback. Setting a bit puts the corresponding transceiver in serial loopback mode. In serial loopback mode, the TX lane loops back to the RX lane on an internal loopback path.	28'hX4'b0 <sup>(1)</sup>	RW
0x314	EIO_FLAG_SEL	Supports indirect addressing of individual FIFO flags in the 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 <code>EIO_FLAGS</code> register. The value in the <code>EIO_FLAG_SEL</code> register directs the IP core to make available the following FIFO flag: <ul style="list-style-type: none"> <li>• 3'b000: TX FIFO full</li> <li>• 3'b001: TX FIFO empty</li> <li>• 3b010: TX FIFO partially full</li> <li>• 3'b011: TX FIFO partially empty</li> <li>• 3b100: RX FIFO full</li> <li>• 3b101: RX FIFO empty</li> <li>• 3b110: RX FIFO partially full</li> <li>• 3b111: RX FIFO partially empty</li> </ul>	29'hX3'b0 <sup>(1)</sup>	RW
0x315	EIO_FLAGS	PCS indirect data. To read a FIFO flag, set the value in the <code>EIO_FLAG_SEL</code> register to indicate the flag you want to read. After you specify the flag in the <code>EIO_FLAG_SEL</code> register, each bit [n] in the <code>EIO_FLAGS</code> register has the value of that FIFO flag for the transceiver channel for lane [n].	28'hX4'b0 <sup>(1)</sup>	RO
0x321	EIO_FREQ_LOCK	Each asserted bit indicates that the corresponding lane RX clock data recovery (CDR) phase-locked loop (PLL) is locked.	28'hX4'b0 <sup>(1)</sup>	RO
0x322	PHY_CLK	The following encodings are defined:	29'hX3'b00 <sup>(1)</sup>	RO

*continued...*

(1) X means "Don't Care".

(2) Register value convert in decimal.

## 7. Control, Status, and Statistics Register Descriptions

UG-20291 | 2020.10.05



Addr	Name	Description	Reset	Access
		<ul style="list-style-type: none"> <li>Bit[0]: Indicates if the TX PCS is ready</li> <li>Bit [1]: Indicates if the TX MAC PLL is locked.</li> <li>Bit[2]: Indicates if the RX PLL is locked.</li> </ul>		
0x323	FRM_ERR	If the IP core loses word lock, it clears this register.	28'hX_4'b0 <sup>(1)</sup>	RO
0x324	SCLR_FRM_ERR	Synchronous clear for FRM_ERR register. Write 1'b1 to this register to clear the FRM_ERR register and bit [1] of the LANE_DESKEWED register. A single bit clears all sticky framing errors. This bit does not auto-clear. Write a 1'b0 to continue logging frame errors.	0x0	RW
0x325	EIO_RX_SOFT_PURGE_S	Set bit [0] to clear the RX FIFO for all four physical lanes. <ul style="list-style-type: none"> <li>Bit[11]: If set to 1, disables the bitslip request from PCS to PMA.</li> <li>Bit[12]: If set to 1, disables the reset request from PCS to PMA. To use the transceiver toolkit, this bit must be set to 1.</li> </ul>	0x0000	RO
0x326	RX_PCS_FULLY_ALIGNED_S	Indicates the RX PCS is fully aligned and ready to accept traffic. <ul style="list-style-type: none"> <li>Bit[0]: RX PCS fully aligned status.</li> <li>Bit[1]: RX PCS bit error rate status. A bit value of 1 indicates a bit error rate higher than 10<sup>-4</sup> or there are at least 16 errors within 50 us. This bit value is only valid when the link fault generation is enabled.</li> </ul>	31'hX1'b0 <sup>(1)</sup>	RO
0x329	LANE_DESKEWED	The following encodings are defined: <ul style="list-style-type: none"> <li>Bit [0]: Indicates all lanes are deskewed.</li> <li>Bit [1]: When asserted indicates a change in lanes deskewed status. To clear this sticky bit, write 1'b1 to the corresponding bit of the SCLR_FRM_ERR register. This is a latched signal.</li> </ul>	30'hX2'b00 <sup>(1)</sup>	RO
0x330	PCS_VLANE	The following encodings are defined: <ul style="list-style-type: none"> <li>Bit[1:0]: Virtual index for physical lane 0.</li> <li>Bit[3:2]: Virtual index for physical lane 1.</li> <li>Bit[5:4]: Virtual index for physical lane 2.</li> <li>Bit[7:6]: Virtual index for physical lane 3.</li> </ul>	24'bX8'b0 <sup>(1)</sup>	RO
0x341	KHZ_RX	The register indicates the value of RX clock (clk_rxmac) frequency. Apply the following definition for the frequency value: [(Register value <sup>(2)</sup> * clk_status)/10] KHZ	0x0000 0000	RO
0x342	KHZ_TX	The register indicates the value of TX clock (clk_txmac) frequency. Apply the following definition for the frequency value: [(Register value <sup>(2)</sup> * clk_status)/10] KHZ	0x0000 0000	RO

(1) X means "Don't Care".

(2) Register value convert in decimal.



## 7.2. TX MAC Registers

Table 27. TX MAC Registers

Addr	Name	Description	Reset	Access
0x400	TXMAC_REVID	TX MAC revision ID for 40GbE TX MAC CSRs.	0x0627 2016	RO
0x401	TXMAC_SCRATCH	Scratch register available for testing.	0x0000 0000	RW
0x402	TXMAC_NAME_0	First 4 characters of module variation identifier string, "40gMACTxCSR".	0x3430 674D	RO
0x403	TXMAC_NAME_1	Next 4 characters of IP core variation identifier string, "ACTx".	0x4143 5278	RO
0x404	TXMAC_NAME_2	Final 4 characters of IP core variation identifier string, "0CSR". The "0" is unprintable.	0x0043 5352	RO
0x405	LINK_FAULT	<p>Link Fault Configuration Register. The following bits are defined:</p> <ul style="list-style-type: none"> <li>Force Remote Fault bit[3]: When link fault generation is enabled, stops data transmission and forces transmission of a remote fault.</li> <li>Disable Remote Fault bit[2]: When both link fault reporting and unidirectional transport are enabled, the core transmits data and does not transmit remote faults (RF). This bit takes effect when the value of this register is 28'hX4'b0111.</li> <li>Unidir Enable bit[1]: When asserted, the core includes Clause 66 support for the remote link fault reporting on the Ethernet link.</li> <li>Link Fault Reporting Enable bit[0]: The following encodings are defined: <ul style="list-style-type: none"> <li>1'b1: The PCS generates the proper fault sequence on Ethernet link, when conditions are met.</li> <li>1'b0: The PCS does not generate the fault sequence.</li> </ul> </li> </ul>	28'hX_4'b0001 <sup>(3)</sup>	RW
0x406	IPG_COL_REM	<p>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.</p> <p>Bits [31:8] of this register are Reserved.</p>	0xXXXX 0004 <sup>(3)</sup>	RW
0x407	MAX_TX_SIZE_CONFIG	<p>Specifies the maximum TX frame length. Frames that are longer are considered oversized. They are transmitted, but also increment the CNTR_TX_OVERSIZE register. Bits [31:16] of this register are Reserved.</p>	0xXXXX 2580 <sup>(3)</sup>	RW
0x40A	TX_MAC_CONTROL	<p>TX MAC Control Register. A single bit is defined:</p> <ul style="list-style-type: none"> <li>Bit[1]: VLAN detection disabled. This bit is deasserted by default, implying VLAN detection is enabled.</li> </ul>	30'hX2'b0X <sup>(3)</sup>	RW

<sup>(3)</sup> X means "Don't Care".



## 7.3. RX MAC Registers

**Table 28. RX MAC Registers**

Addr	Name	Description	Reset	Access
0x500	RXMAC_REVID	RX MAC revision ID for Low Latency 40G for ASIC Proto Ethernet IP core.	0x0627 2016	RO
0x501	RXMAC_SCRATCH	Scratch register available for testing.	0x0000 0000	RW
0x502	RXMAC_NAME_0	First 4 characters of IP core variation identifier string, "40gMACRxCSR".	0x3430 674D	RO
0x503	RXMAC_NAME_1	Next 4 characters of IP core variation identifier string, "ACRx".	0x4143 5278	RO
0x504	RXMAC_NAME_2	Final 4 characters of IP core variation identifier string, "0CSR". The "0" is unprintable.	0x0043 5352	RO
0x506	MAX_RX_SIZE_CONFIG	Specifies the maximum frame length available. The MAC asserts when the length of the received frame exceeds the value of this register.  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 statistics registers, the IP core increments the 64-bit CNTR_RX_OVERSIZE counter.	0xXXXX 2580 <sup>(4)</sup>	RW
0x507	MAC_CRC_CONFIG	The RX CRC forwarding configuration register. The following encodings are defined: <ul style="list-style-type: none"> <li>1'b0: Remove RX CRC, do not forward it to the RX client interface</li> <li>1'b1: Retain RX CRC, forward it to the RX client interface</li> </ul> In either case, the IP core checks the incoming RX CRC and flags errors.	31'hX1'b0 <sup>(4)</sup>	RW
0x508	LINK_FAULT	Link Fault Status Register. <ul style="list-style-type: none"> <li>Bit[0]: Local fault status bit.</li> <li>Bit[1]: Remote fault status bit.</li> </ul> For regular (non-unidirectional) Link Fault, implements <i>IEEE 802.3 BA Ethernet Clause 81.3.4</i> . For unidirectional Link Fault, implements <i>IEEE 802.3 Ethernet Clause 66</i> .	30'hX2'b00 <sup>(4)</sup>	RO
0x50A	RX_MAC_CONTROL	RX MAC Control Register. The following bits are defined: <ul style="list-style-type: none"> <li>Bit[4]: Preamble check. Strict SFD checking option to compare each packet preamble to 0x555555555555. This field is available only if you turn on <b>Enable Strict SFD check</b>.</li> <li>Bit[3]: SFD check. Strict SFD checking option to compare each SFD byte to 0x5D. This field is available only if you turn on <b>Enable Strict SFD check</b>.</li> <li>Bit [1]: VLAN detection disabled. This bit is deasserted by default implying VLAN detection is enabled.</li> </ul>	30'h0_2'b0X <sup>(4)</sup>	RW

(4) X means "Don't Care".



## 7.4. Statistics Registers

The Low Latency 40G for ASIC Proto Ethernet statistics registers count 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 statistics counters module is a synthesis option. The statistics registers are counters that are implemented inside the CSR. When you turn on the **Enable MAC stats counters** parameter in the Low Latency 40G for ASIC Proto Ethernet parameter editor, the counters are implemented in the CSR. When you turn off the **Enable MAC stats counters** parameter in the Low Latency 40G for ASIC Proto Ethernet parameter editor, the counters are not implemented in the CSR, and read access to the counters returns undefined results.

After system power-up, the statistics counters have random values. You must clear these registers and confirm the system is stable before using their values. You can clear the registers with the `csr_rst_n` input signal, or with the configuration registers at offsets 0x845 and 0x945.

The configuration register at offset 0x845 allows you to clear all of the TX statistics counters. The configuration register at offset 0x945 allows you to clear all of the RX statistics counters. 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.

Reading the value of a statistics register does not affect its value.

To ensure that the counters you read are consistent, you should issue a shadow request to create a snapshot of all of the TX or RX statistics registers, by setting bit [2] of the configuration register at offset 0x845 or 0x945, respectively. Until you reset this bit, the counters continue to increment but the readable values remain constant. You can read bit [1] of the status register at offset 0x846 or 0x946, respectively, to confirm your shadow request has been granted or released.

### 7.4.1. TX Statistics Registers

**Table 29. Transmit Side Statistics Registers**

Address	Name-	Description	Access
0x800	CNTR_TX_FRAGMENT_S_LO	Number of transmitted frames less than 64 bytes and reporting a CRC error (lower 32 bits).	RO
0x801	CNTR_TX_FRAGMENT_S_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
<i>continued...</i>			

## 7. Control, Status, and Statistics Register Descriptions

UG-20291 | 2020.10.05



Address	Name	Description	Access
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
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
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

*continued...*



Address	Name-	Description	Access
0x81A	CNTR_TX_128to255 B_LO	Number of transmitted frames between 128–255 bytes (lower 32 bits).	RO
0x81B	CNTR_TX_128to255 B_HI	Number of transmitted frames between 128–255 bytes (upper 32 bits).	RO
0x81C	CNTR_TX_256to511 B_LO	Number of transmitted frames between 256–511 bytes (lower 32 bits).	RO
0x81D	CNTR_TX_256to511 B_HI	Number of transmitted frames between 256–511 bytes (upper 32 bits).	RO
0x81E	CNTR_TX_512to102 3B_LO	Number of transmitted frames between 512–1023 bytes (lower 32 bits).	RO
0x81F	CNTR_TX_512to102 3B_HI	Number of transmitted frames between 512–1023 bytes (upper 32 bits).	RO
0x820	CNTR_TX_1024to15 18B_LO	Number of transmitted frames between 1024–1518 bytes (lower 32 bits).	RO
0x821	CNTR_TX_1024to15 18B_HI	Number of transmitted frames between 1024–1518 bytes (upper 32 bits).	RO
0x822	CNTR_TX_1519toMA XB_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_1519toMA XB_HI	Number of transmitted frames of size 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_DA TA_OK_LO	Number of valid multicast frames transmitted, excluding control frames (lower 32 bits).	RO
0x827	CNTR_TX_MCAST_DA TA_OK_HI	Number of valid multicast frames transmitted, excluding control frames (upper 32 bits).	RO
0x828	CNTR_TX_BCAST_DA TA_OK_LO	Number of valid broadcast frames transmitted, excluding control frames (lower 32 bits).	RO
0x829	CNTR_TX_BCAST_DA TA_OK_HI	Number of valid broadcast frames transmitted, excluding control frames (upper 32 bits).	RO
0x82A	CNTR_TX_UCAST_DA TA_OK_LO	Number of valid unicast frames transmitted, excluding control frames (lower 32 bits).	RO
0x82B	CNTR_TX_UCAST_DA TA_OK_HI	Number of valid unicast frames transmitted, excluding control frames (upper 32 bits).	RO
0x82C	CNTR_TX_MCAST_CT RL_LO	Number of valid multicast frames transmitted, excluding data frames (lower 32 bits).	RO
0x82D	CNTR_TX_MCAST_CT RL_HI	Number of valid multicast frames transmitted, excluding data frames (upper 32 bits).	RO
0x82E	CNTR_TX_BCAST_CT RL_LO	Number of valid broadcast frames transmitted, excluding data frames (lower 32 bits).	RO

*continued...*

## 7. Control, Status, and Statistics Register Descriptions

UG-20291 | 2020.10.05



Address	Name-	Description	Access
0x82F	CNTR_TX_BCAST_CTL_HI	Number of valid broadcast frames transmitted, excluding data frames (upper 32 bits).	RO
0x830	CNTR_TX_UCAST_CTL_LO	Number of valid unicast frames transmitted, excluding data frames (lower 32 bits).	RO
0x831	CNTR_TX_UCAST_CTL_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. Therefore, this counter does not increment in normal operating conditions.	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. Therefore, this counter does not increment in normal operating conditions.	RO
0x836–0x844	Reserved		
0x845	CNTR_TX_CONFIG	<p>Bits[2:0]: Configuration of TX statistics counters:</p> <ul style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>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.</li> </ul> <p>Bits[31:3] are Reserved.</p>	RW
0x846	CNTR_TX_STATUS	<ul style="list-style-type: none"> <li>Bit[1]: Indicates that the TX statistics registers are paused (while CNTR_TX_CONFIG[2] is asserted).</li> <li>Bit[0]: Indicates the presence of at least one parity error in the TX statistics counters.</li> </ul> <p>Bits[31:2] are Reserved.</p>	RO
0x847–0x85F	Reserved		
0x860	TxPayloadOctetsOK_LO	Number of transmitted payload bytes in frames with no FCS, undersized, oversized, or payload length errors. If VLAN detection is turned off for the TX MAC (bit[1] of the TX_MAC_CONTROL register at offset 0x40A has the value of 1), the IP core counts the VLAN header bytes (4 bytes for VLAN and 8 bytes for stacked VLAN) as payload bytes. This register is compliant with the requirements for a OctetsTransmittedOK in section 5.2.2.1.8 of the <i>IEEE Standard 802.3-2008</i> .	RO
0x861	TxPayloadOctetsOK_HI		RO
0x862	TxFrameOctetsOK_LO	Number of transmitted bytes in frames with no FCS, undersized, oversized, or payload length errors. This register is compliant with the requirements for ifOutOctets in RFC3635 (Managed Objects for Ethernet-like Interface Types) and TX etherStatsOctets in RFC2819(Remote Network Monitoring Management Information Base (RMON)).	RO
0x863	TxFrameOctetsOK_HI		RO



### 7.4.2. RX Statistics Registers

**Table 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
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

*continued...*

## 7. Control, Status, and Statistics Register Descriptions

UG-20291 | 2020.10.05



Address	Name	Description	Access
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
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
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

*continued...*



Address	Name	Description	Access
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 received pause frames, with or without error (lower 32 bits)	RO
0x933	CNTR_RX_PAUSE_HI	Number of received pause frames, with or without error (upper 32 bits)	RO
0x934	CNTR_RX_RUNT_LO	Number of received runt packets (lower 32 bits) A runt 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) A runt 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
0x936–0x944	Reserved		
0x945	CNTR_RX_CONFIG	Bits[2:0]: Configuration of RX statistics counters: <ul style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>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.</li> </ul> Bits[31:3] are Reserved.	RW
0x946	CNTR_RX_STATUS	<ul style="list-style-type: none"> <li>Bit[1]: Indicates that the RX statistics registers are paused (while CNTR_RX_CONFIG[2] is asserted).</li> <li>Bit[0]: Indicates the presence of at least one parity error in the RX statistics counters.</li> </ul> Bits [31:2] are Reserved.	RO

continued...

## 7. Control, Status, and Statistics Register Descriptions

UG-20291 | 2020.10.05



Address	Name	Description	Access
0x947–0x95F	Reserved		
0x960	RxPayloadOctetsOK_LO	Number of received payload bytes in frames with no FCS, undersized, oversized, or payload length errors. If VLAN detection is turned off for the RX MAC (bit [1] of the RXMAC_CONTROL register at offset 0x50A has the value of 1), the IP core counts the VLAN header bytes (4 bytes for VLAN and 8 bytes for stacked VLAN) as payload bytes. This register is compliant with the requirements for aOctetsReceivedOK in section 5.2.2.1.14 of the <i>IEEE Standard 802.3-2008</i> .	RO
0x961	RxPayloadOctetsOK_HI		RO
0x962	RxFrameOctetsOK_LO	Number of received bytes in frames with no FCS, undersized, oversized, or payload length errors. This register is compliant with the requirements for ifInOctets in RFC3635 (Managed Objects for Ethernet-like Interface Types) and RX etherStatsOctets in RFC2819 (Remote Network Monitoring Management Information Base (RMON)).	RO
0x963	RxFrameOctetsOK_HI		RO

## 8. Debugging the Link

---

The following steps should help you identify and resolve common problems that occur when bringing up a Low Latency 40G for ASIC Proto Ethernet 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 the IP core is 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 (`KHZ_TX`, `KHZ_RX` PHY registers) 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 in 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 the bit that corresponds to any lane, indicates a clock recovery problem. Clock recovery difficulties are typically caused by the following problems:
  - Bit errors
  - 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 the `EIO_FLAGS` register. During normal operation, the TX and RX FIFOs should be nominally filled. Observing 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 transceiver analog parameters to improve the bit error rate. IDLE traffic is representative for analog purposes.



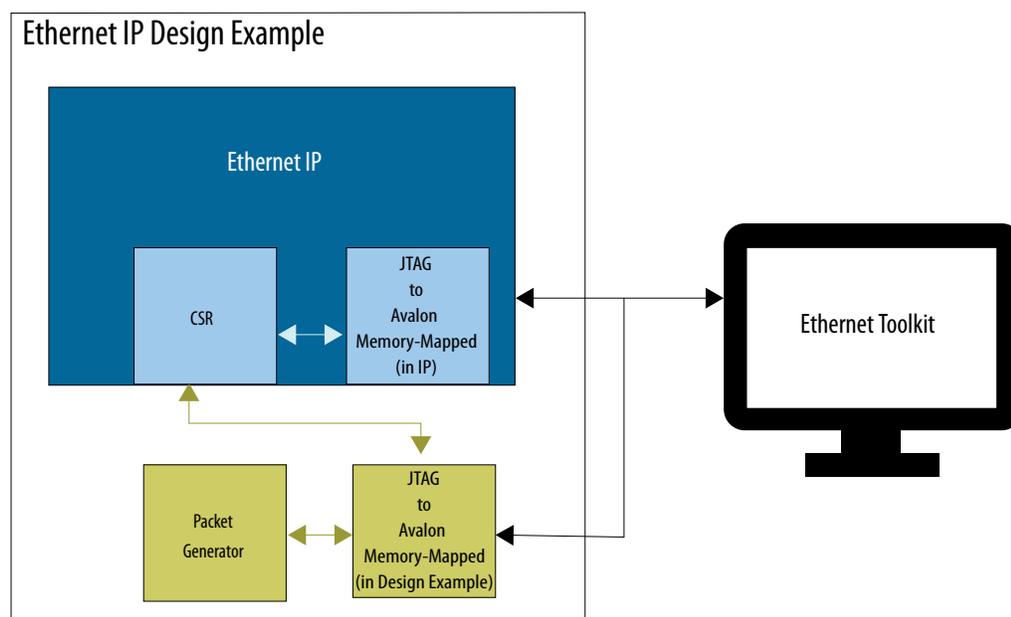
In addition, your IP core can experience loss of signal on the Ethernet link after it is established. In this case, the TX functionality is unaffected, but the RX functionality is disrupted. The following symptoms indicate a loss of signal on the Ethernet link:

- 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 triggers the RX digital reset process.

## 9. Ethernet Toolkit Overview

The Ethernet Toolkit is a TCL based debugging tool that allows you to interact with an Ethernet Intel FPGA IP in real time.

**Figure 20. Block Diagram of the Ethernet Toolkit**



You can use the Ethernet Toolkit with hardware design that has standalone Ethernet IP. You can also use the Ethernet Toolkit with an Intel Quartus Prime generated Ethernet IP design example.

### 9.1. Features

The Ethernet Toolkit offers the following features when used with hardware design that has standalone Ethernet IP as well as with an Intel Quartus Prime generated Ethernet IP design example:

- Verifies the status of the Ethernet link.
- Reads and writes to status and configuration registers of the IP.
- Displays the values of TX/RX status and statistics registers.
- Ability to assert and deassert IP resets.
- Verifies the IP's error correction capability.



The Ethernet Toolkit also offers some additional features when used with an Intel Quartus Prime generated Ethernet IP design example:

- Provides access to the example design packet generator.
- Execute testing procedures to verify the functionality of Ethernet IPs.
- Enable and disable MAC loopback.
- Set source and destination MAC addresses.

**Related Information**

[Ethernet Toolkit User Guide](#)



## 10. Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP User Guide Archives

---

IP versions are the same as the Intel Quartus Prime Design Suite software versions up to v19.1. From Intel Quartus Prime Design Suite software version 19.2 or later, IP cores have a new IP versioning scheme.

If an IP core version is not listed, the user guide for the previous IP core version applies.

Intel Quartus Prime Version	IP Core Version	User Guide
20.2	19.1.0	<a href="#">Low Latency 40G for ASIC Proto Intel FPGA IP Ethernet Intel FPGA IP User Guide</a>



## 11. Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Revision History

Date	Intel Quartus Prime Version	IP Version	Changes
2020.10.05	20.3	19.1.0	<ul style="list-style-type: none"> <li>• Revised <b>Select USER MAC mode</b> parameter options: <ul style="list-style-type: none"> <li>– Updated from PHY+MAC+PCS to PCS+MAC.</li> <li>– Updated from PHY+PCS to PCS_Only.</li> </ul> </li> <li>• Revised the <i>Flow Control Signals</i> table to update the following signal description: <ul style="list-style-type: none"> <li>– pause_insert_tx0[(FCQN-1):0]</li> <li>– pause_insert_tx1[(FCQN-1):0]</li> <li>– pause_insert_rx[(FCQN-1):0]</li> </ul> </li> <li>• Added new section: <i>Ethernet Toolkit Overview</i>.</li> </ul>
2020.07.07	20.2	19.1.0	Initial release.

Intel Corporation. All rights reserved. Agilix, Altera, Arria, Cyclone, Enpirion, Intel, the Intel logo, 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  
9001:2015  
Registered