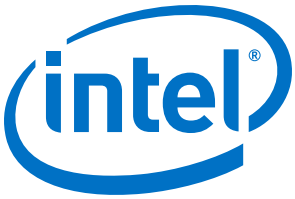


Low Latency E-Tile 40G Ethernet Intel® FPGA IP User Guide

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

IP Version: **21.0.0**

Subscribe
Send Feedback

UG-20272 | 2020.10.05
Latest document on the web: **PDF** | **HTML**

Contents

1. About the Low Latency E-Tile 40G Ethernet Intel FPGA IP

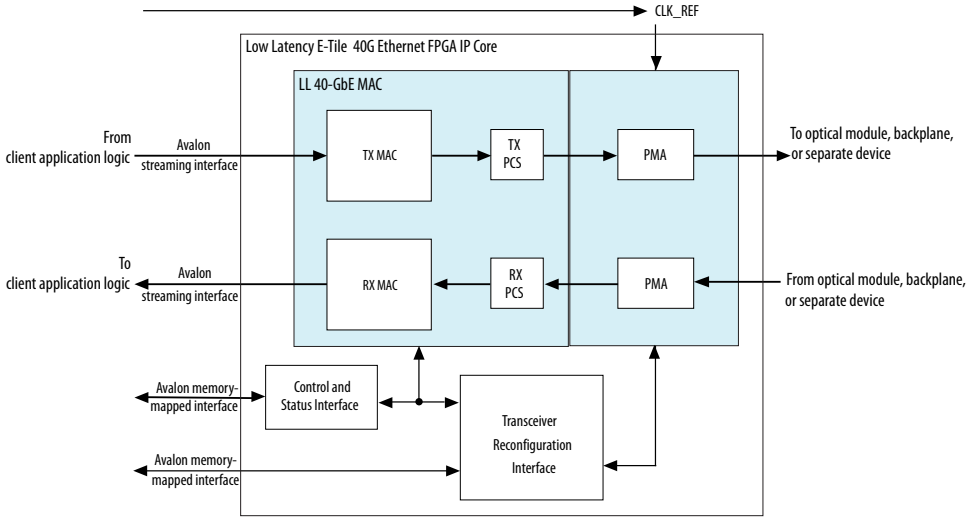
The Low Latency E-Tile 40-Gbps Ethernet (LL E-Tile 40GbE) IP core is used in multiple variants of the Intel® Stratix® 10 and Intel Agilex™ device families. The IP core implements the *IEEE 802.3-2010 40G Ethernet Standard* and includes options to support unidirectional transport as defined in *Clause 66* of the *IEEE 802.3-2012 Ethernet Standard*.

The MAC client side interface for the Low Latency E-Tile 40G Ethernet IP core is a 128-bit Avalon® streaming interface and a 32-bit Avalon memory-mapped interface control path. The network interfaces are standard XLAUI interfaces.

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

Figure 1. Low Latency E-Tile 40G Ethernet Block Diagram

Main blocks, internal connections, and external block requirements.



1.1. Low Latency E-Tile 40G Ethernet IP Core Supported Features

The Low Latency E-Tile 40G Ethernet IP core supports the following features:

- Parameterizable through the IP Catalog available with the Intel Quartus® Prime software.
- Designed to the *IEEE 802.3ba-2010 High Speed Ethernet Standard* available on the IEEE website (www.ieee.org).
- Soft PCS logic that interfaces seamlessly to 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 management interface to access the IP core control and status registers.
- Avalon streaming 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 streaming transmitter interface.
- Optional statistics counters.

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

1.2. Low Latency E-Tile 40G Ethernet Core Device Family and Speed Grade Support

1.2.1. Device Family Support

Table 1. Intel FPGA IP Core Device Support Levels

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

Table 2. Low Latency E-Tile 40G Ethernet IP Core Device Family Support

Shows the level of support offered by the Low Latency E-Tile 40G Ethernet IP core for each Intel FPGA device family.

Device Family	Support
Intel Stratix 10	Preliminary
Intel Agilex	Preliminary
Other device families	No support

1.2.2. Low Latency E-Tile 40G Ethernet IP Core Device Speed Grade Support

Table 3. Slowest Supported Device Speed Grades

IP Core	Device Family	Supported Speed Grades
Low Latency E-Tile 40G Ethernet	Intel Stratix 10	–1
		–2
		–3
	Intel Agilex	–1
		–2
		–3

1.3. Resource Utilization

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

Table 4. IP Core Variation Encoding for Resource Utilization Table

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

IP Core Variation / Parameter	A	B	C	E	F
Ready latency	0	0	3	3	3
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
Enable Strict SFD check	—	—	On	—	On

Table 5. IP Core FPGA Resource Utilization in Intel Stratix 10 device

Lists the resources and expected performance for selected variations of the Low Latency E-Tile 40G Ethernet IP core in a Intel Stratix 10 device.

These results were obtained using the Intel Quartus Prime 20.1 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 Intel Quartus Prime Fitter Report.

IP Core Variation	ALMs	Dedicated Logic Registers	Memory M20K
A	7,584	18,644	1
B	11,480	27,379	1
C	12,086.5	28,676	1
E	11,455.3	26,941	1
F	11,712	28,104	1

Table 6. IP Core FPGA Resource Utilization in Intel Agilex device

Lists the resources and expected performance for selected variations of the Low Latency E-Tile 40G Ethernet IP core in a Intel Agilex device.

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

IP Core Variation	ALMs	Dedicated Logic Registers	Memory M20K
A	7,379.9	15,958	1
B	11,207	24,538	1
C	11,912.4	26,210	1
E	11,393.9	25,237	1
F	11,468.7	25,454	1

Related Information

Low Latency E-Tile 40G Ethernet IP Core Parameters on page 9
Information about the parameters and values in the IP core variations.

1.4. Release Information

Table 7. Release information for the *Low Latency E-Tile 40G Ethernet Intel FPGA* IP

Item	Description
IP Version	21.0.0
Intel Quartus Prime Pro Edition	20.3
Release Date	2020.09.28
Ordering Code	IP–40GETILEMAC

Related Information

- Low Latency E-Tile 40G Ethernet Release Notes
  The IP Release Notes list IP changes in a particular release.
- Low Latency E-Tile 40G Ethernet Design Example User Guide
  The design example user guide for the Low Latency E-Tile 40G Ethernet IP.

2. Low Latency E-Tile 40G Ethernet IP Core Parameters

The Low Latency E-Tile 40G Ethernet parameter editor has an **IP** tab and the **Main** tab.

The Low Latency E-Tile 40G Ethernet parameter editor also includes an **Example Design** tab. For information about that tab, refer to the *Low Latency E-Tile 40G Ethernet Design Example User Guide*.

Table 8. Low Latency E-Tile 40G Ethernet IP Core Parameters: Main Tab

Parameter	Range	Default Setting	Description
General			
Target transceiver tile	**E-Tile**	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.
Protocol speed	**40GbE**	**40GbE**	Selects the Ethernet data rate.
Ready latency	**0**, **3**	**0**	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 `l2_tx_ready` signal to the clock cycle in which the IP core can accept data on the TX client interface. Refer to the *Avalon Interface Specifications*. Selecting a latency of 3 eases timing closure at the expense of increased latency for the TX datapath.
PCS/PMA Options			
Enable SyncE	**Enabled**, **Disabled**	**Disabled**	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.
PHY reference frequency	**156.25 MHz**	**156.25 MHz**	Sets the expected incoming PHY `clk_ref` reference frequency. The input clock frequency must match the frequency you specify for this parameter (±100ppm).
MAC Options			
Enable TX CRC insertion	**Enabled**, **Disabled**	**Enabled**	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).
Enable link fault generation	**Enabled**, **Disabled**	**Disabled**	When enabled, the IP core implements link fault signaling as defined in the *IEEE 802.3-2012 IEEE Ethernet Standard*. The MAC includes a Reconciliation

continued...

Parameter	Range	Default Setting	Description
			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.
Enable preamble passthrough	**Enabled**, **Disabled**	**Disabled**	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.
Enable MAC stats counters	**Enabled**, **Disabled**	**Enabled**	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.
Enable Strict SFD check	**Enabled**, **Disabled**	**Disabled**	When enabled, the IP core can implement strict SFD checking, depending on register settings.
Flow Control Options			
Enable MAC flow control	**Enabled**, **Disabled**	**Disabled**	When enabled, the IP core implements flow control. When either link partner experiences congestion, the respective transmit control sends pause frames.
Number of queues in priority flow control	1-8	8	Specifies the number of queues used in managing flow control.
Configuration, Debug and Extension Options			
Enable Native PHY Debug Master Endpoint (NPDME)	**Enabled**, **Disabled**	**Disabled**	If this parameter is turned on, the E-Tile Transceiver Native PHY IP includes an embedded Native PHY Debug Master Endpoint that connects internally to the Avalon memory-mapped slave interface for dynamic reconfiguration. The Native PHY Debug Master Endpoint can access the transceiver's reconfiguration space. It can perform certain tests and debug functions via JTAG using the System Console.
Enable JTAG to Avalon Master Bridge	**Enabled**, **Disabled**	**Disabled**	If turned on, the IP core includes a JTAG to Avalon memory-mapped interface master bridge connecting internally to status and reconfiguration registers.

Related Information

- Low Latency E-Tile 40G Ethernet Design Example User Guide
  Information about the parameters on the **Example Design** tab.
- Avalon Interface Specifications
  Detailed information about Avalon streaming interfaces and the Avalon streaming interface readyLatency parameter.
- E-Tile Transceiver PHY User Guide
  Information about the **NPDME** and **Enable capability registers** parameters of the E-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 2. IP Core Installation Path

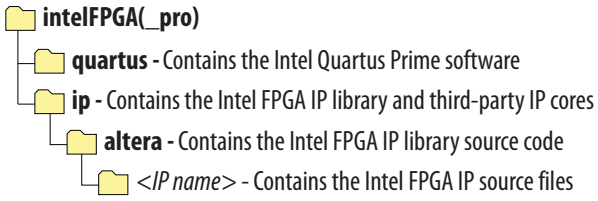


Table 9. IP Core Installation Locations

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 E-Tile 40G Ethernet IP Core Parameters and Options

The Low Latency E-Tile 40G 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 E-Tile 40G Ethernet IP Core Parameters on page 9 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 E-Tile 40G Ethernet 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.

Related Information

Low Latency E-Tile 40G Ethernet Design Example User Guide

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
- Low Latency E-Tile 40G Ethernet Design Example User Guide
  Describes the Low Latency E-Tile 40G Ethernet testbench and design example and how to simulate the testbench.

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 E-Tile 40G Ethernet Design Example User Guide*.

Figure 3. IP Core Generated Files

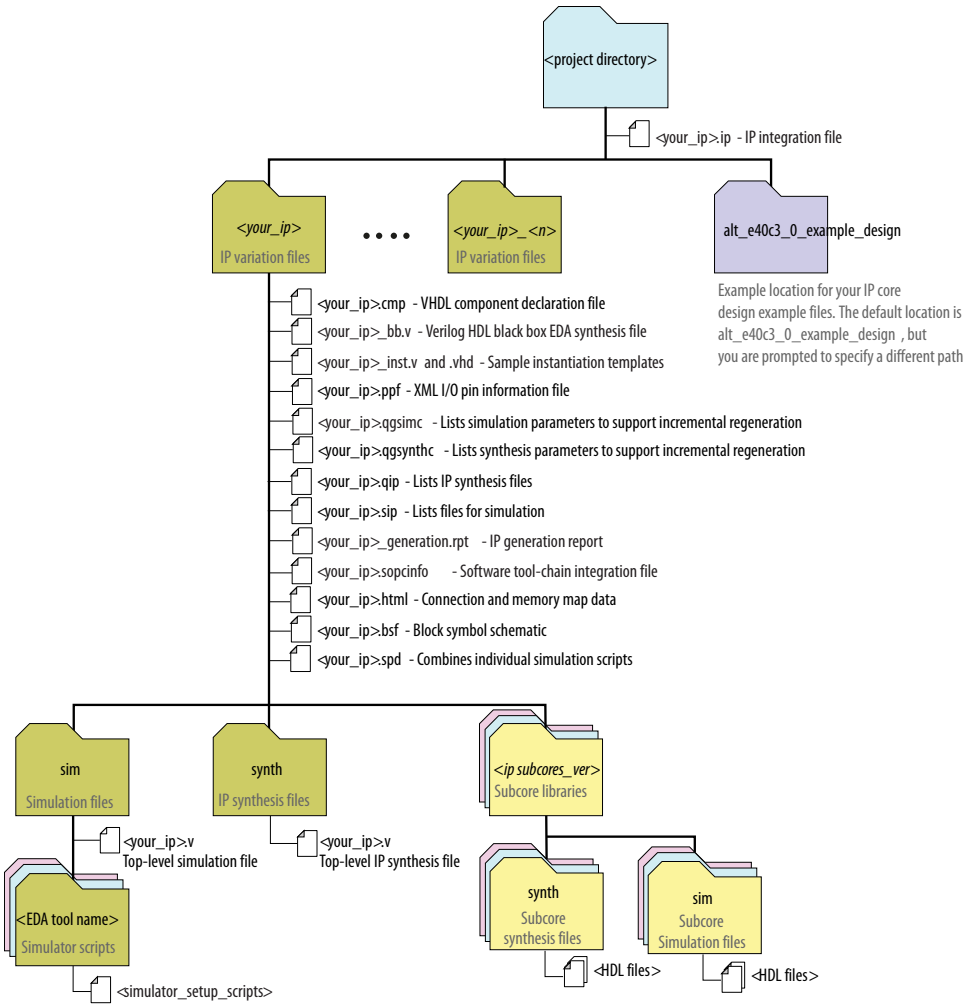


Table 10. IP Core Generated Files

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.

continued...

File Name	Description
<*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.
<*your_ip*>.sip	Contains information required for NativeLink simulation of IP components. You must add the **.sip** 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, **.regmap** file generates. The **.regmap** 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 **.svd** files for slave interfaces visible to System Console masters are stored in the **.sof** file in the debug section. System Console reads this section, which 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 `msim_setup.tcl` to set up and run a simulation.
aldec/	Contains a Riviera-PRO script `rivierapro_setup.tcl` to setup and run a simulation. 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 `vcs_setup.sh` to set up and run a VCS* simulation. Contains a shell script `vcsmx_setup.sh` and `synopsys_ sim.setup` file to set up and run a VCS MX simulation.
cadence/	Contains a shell script `ncsim_setup.sh` and other setup files to set up and run an NCSIM simulation.
submodules/	Contains HDL files for the IP core submodule.
<*child IP cores*>/	For each generated child IP core directory, Platform Designer (Standard) generates `synth/` and `sim/` sub-directories.

Related Information

Low Latency E-Tile 40G Ethernet Design Example User Guide
Information about the Low Latency E-Tile 40G Ethernet design example file structure.

3.5. Integrating Your IP Core in Your Design

3.5.1. Pin Assignments

When you integrate your Low Latency E-Tile 40G 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. Ethernet Adaptation Flow

Refer to *Loading a PMA Configuration* and *PMA Registers 0x200 to 0x203 Usage* sections in the *E-tile Transceiver PHY User Guide* for more details on the adaptation flow and how to get started.

This adaptation flow assumes a valid Ethernet traffic.

1. Assert `tx_rst_n` and `rx_rst_n` signals.
2. Trigger PMA analog.
3. Reload PMA settings and call PMA attribute sequencer on all lanes.
4. Apply control status registers (CSR) reset.
5. Deassert the `rx_rst_n` signal.
6. If using a PMA configuration, load the PMA configuration using control status registers (CSR). This is loaded to the registers using PMA registers 0x200 to 0x203(1).
   a. Write 0x40143 = 0x80.
   b. Read 0x40144[0] until it changes to 1.
7. Enable internal serial loopback(2) and run initial adaptation. Verify that the initial adaptation status is complete using interrupt code 0x0126 and data 0x0B00.
8. Enable mission mode and disable internal serial loopback (skip this step if using internal serial loopback)(2).
9. Wait for valid data traffic on RX and then proceed to the next step.
10. Run initial adaptation. Verify that the initial adaptation status is complete using interrupt code 0x0126 and data 0x0B00 (skip this step if using internal serial loopback).
11. Run continuous adaptation(3).

---

(1) Refer to *Loading a PMA Configuration* and *PMA Registers 0x200 to 0x203 Usage*.

(2) For how to enable and disable internal serial loopback, refer to *0x0008: Internal Serial Loopback and Reverse Parallel Loopback Control*.

(3) During the continuous adaptation, the link partner must keep sending the data. If link goes down, the entire sequence must be repeated.

12. Deassert the `rx_rst_n` signal.
13. Optional: Verify that the link status signal `rx_aligned` transitions high.
14. Send packets.

3.5.3. Placement Settings for the Low Latency E-Tile 40G 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 E-Tile 40G Ethernet IP Core Testbench

Intel provides a compilation-only example design and a testbench with most variations of the Low Latency E-Tile 40G 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.

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 E-Tile 40G 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 E-Tile 40G 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
- Low Latency E-Tile 40G Ethernet Design Example User Guide

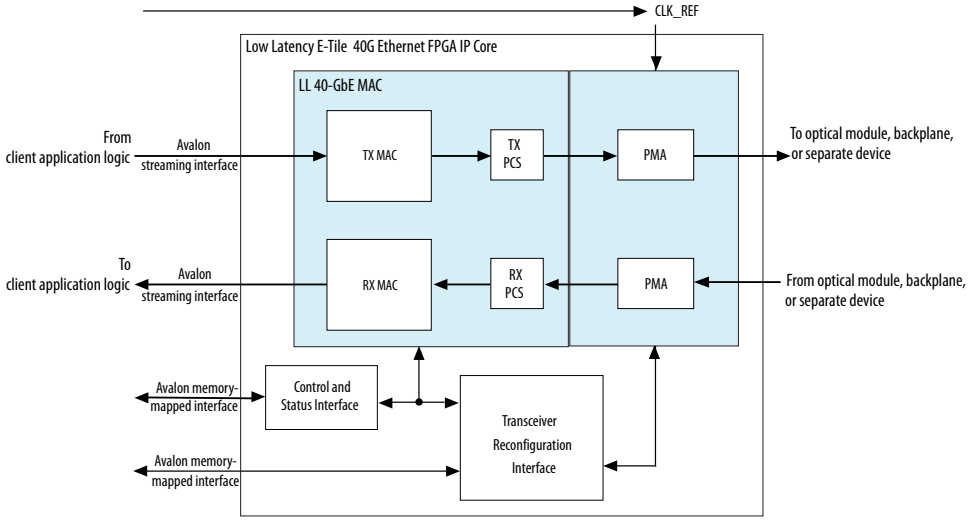
4. Functional Description

4.1. Low Latency E-Tile 40G Ethernet Core Functional Description

The Low Latency E-Tile 40G 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 4. Low Latency E-Tile 40G Ethernet Core Block Diagram

Main blocks, internal connections, and external block requirements.



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 E-Tile 40G 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 5. Typical Client Frame at the Transmit Interface

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

- <*p*> = payload size, which is arbitrarily large
- <*s*> = number of padding bytes (0–46)
- <*g*> = 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.If `l2_tx_startofpacket` and `l2_tx_endofpacket` are asserted in the same cycle, the IP filters out the frame regardless of preamble pass-through mode.

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 E-Tile 40G 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.

4.1.1.4. Frame Check Sequence (CRC-32) Insertion

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

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

CRC bits are transmitted with MSB (X32) first.

You can configure your IP core TX MAC to implement TX CRC insertion or not, by turning **Enable TX CRC insertion** on or off in the IP core parameter editor. By default, the CRC insertion feature is enabled.

In CRC pass-through mode, the `l2_tx_endofpacket` and `l2_rx_endofpacket` along with `l2_tx_empty` and `l2_rx_empty` point to the last byte of Frame Check Sequence (FCS).

Note: In TX CRC pass-through mode, you must provide frames with at least 64 bytes. If **Flow control mode** is enabled, the IP core generates and inserts the CRC for flow control packets.

If you have not selected the CRC pass-through mode, the `l2_tx_endofpacket` and `l2_rx_endofpacket` along with `l2_tx_empty` and `l2_rx_empty` point to the last byte before the first FCS.

Note: The TX CRC insertion requires to set **Flow control mode** to 1.

4.1.2. Low Latency E-Tile 40G 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 6. Flow of Client Frame With Preamble Pass-Through Turned On

This figure uses the following notational conventions:

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

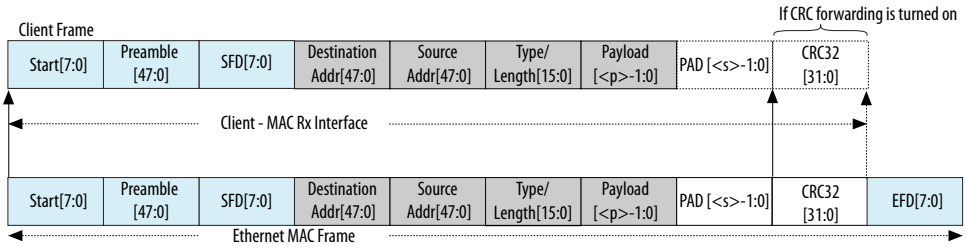
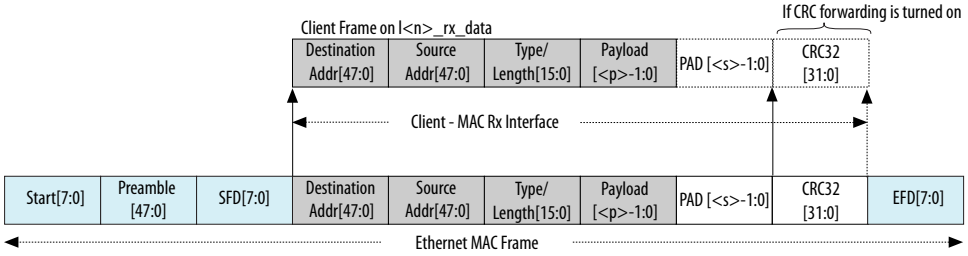


Figure 7. Flow of Client Frame With Preamble Pass-Through Turned Off

This figure uses the following notational conventions:

- <*p*> = payload size, which is arbitrarily large.
- <*s*> = 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 E-Tile 40G Ethernet parameter editor turns on both RX and TX preamble passthrough.

4.1.2.2. IP Core Strict SFD Checking

The Low Latency E-Tile 40G 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 E-Tile 40G 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 <= 0x5DC (1500) —The field represents the payload length of a basic Ethernet frame. The MAC TX/RX continues to check the frame and payload lengths.
- 0x5DC (1500) < Length/type < 0x600 (1536) — The frames with payloads size in this range are not standard basic Ethernet frames, nor they are legal control packets. The payload length is not checked for this kind of packets.
- Length/type >= 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.

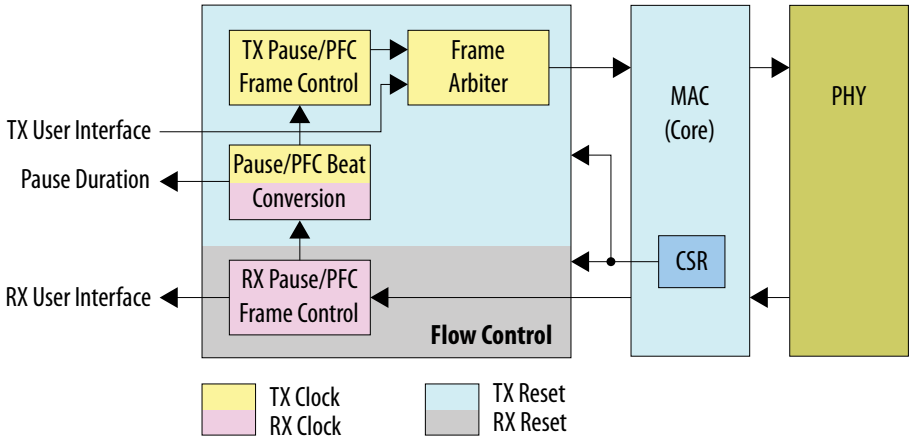
4.1.4. Flow Control

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 8. 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.

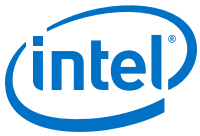
Figure 9. Byte Order on the Client Interface Lanes

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.

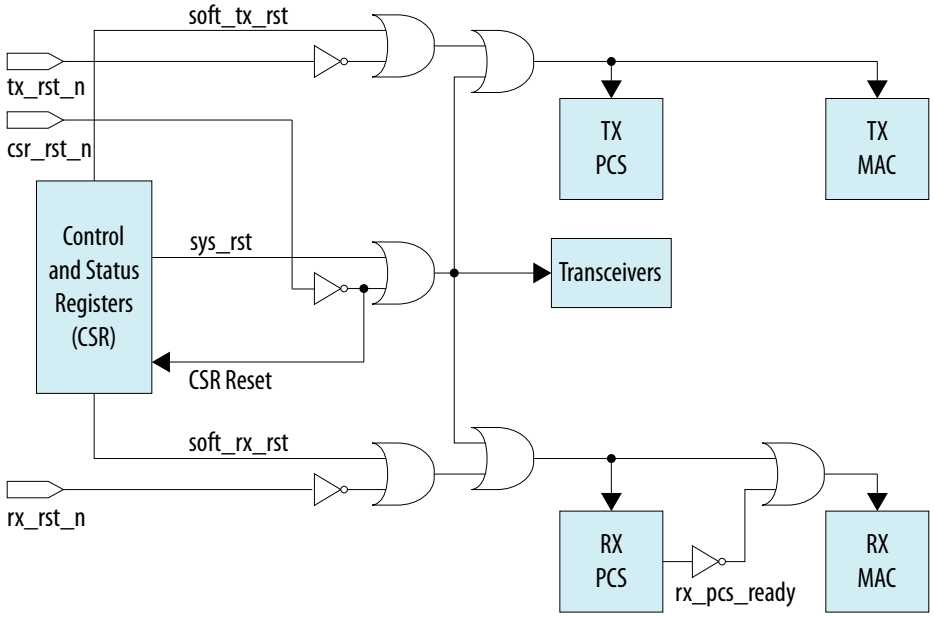
5. Reset

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 10. Conceptual Overview of General IP Core Reset Logic

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.



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.

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.

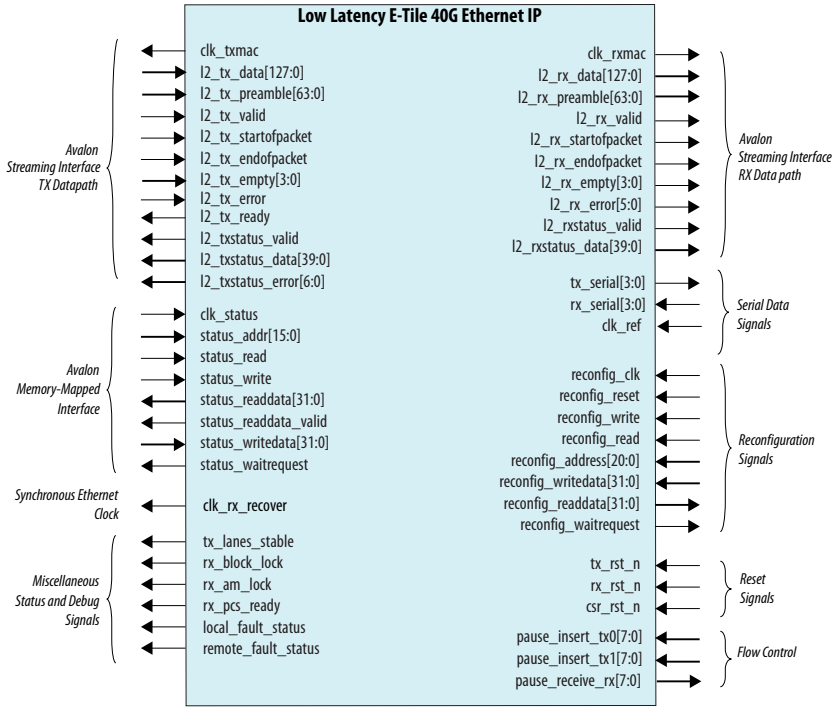
The CSR register read/write needs to wait at least 2 clock cycles after `csr_rst_n` assertion.

Related Information

Reset Signals on page 38

6. Interfaces and Signal Descriptions

Figure 11. Low Latency E-Tile 40G 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 E-Tile 40G Ethernet parameter editor is the Avalon streaming interface readyLatency value on this interface.

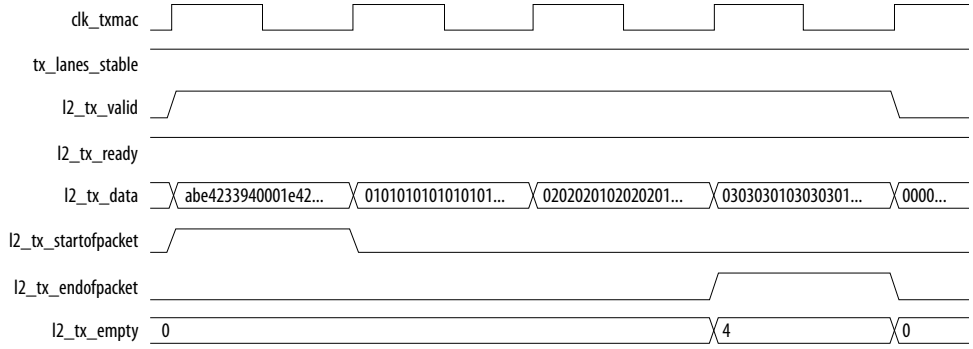
Signal	Direction	Description
`clk_txmac`	Output	The TX clock for the IP core is `clk_txmac`. The frequency of this clock is 312.5 MHz. If you turn on **Use external TX MAC PLL** in the Low Latency E-Tile 40G Ethernet parameter editor, the `clk_txmac_in` input clock drives `clk_txmac`.
`l2_tx_data[127:0]`	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.

continued...

Signal	Direction	Description
l2_tx_preamble[63:0]	Input	User preamble data. Available when you turn on **Enable preamble passthrough** in the Low Latency E-Tile 40G Ethernet parameter editor. User logic drives the custom preamble data when l2_tx_startofpacket is asserted.
l2_tx_valid	Input	When asserted, indicates valid data.
l2_tx_startofpacket	Input	When asserted, indicates the first byte of a frame. When l2_tx_startofpacket is asserted, the MSB of l2_tx_data drives the start of packet. Packets that drive l2_tx_startofpacket and l2_tx_endofpacket in the same cycle are ignored.
l2_tx_endofpacket	Input	When asserted, indicates the end of a packet. Packets that drive l2_tx_startofpacket and l2_tx_endofpacket in the same cycle are ignored.
l2_tx_empty[3:0]	Input	Specifies the number of empty bytes when l2_tx_endofpacket is asserted.
l2_tx_ready	Output	When asserted, indicates that the MAC can accept the data. The IP core asserts the l2_tx_ready signal on clock cycle <*n*> to indicate that clock cycle <*n* + readyLatency> is a ready cycle. The client may only assert l2_tx_valid and transfer data during ready cycles.
l2_tx_error	Input	When asserted in an EOP cycle (while l2_tx_endofpacket is asserted), directs the IP core to insert an error in the packet before sending it on the Ethernet link. *Note:* This functionality is not available in the Quartus Prime Pro 17.1 Stratix 10 ES Editions software.
l2_txstatus_valid	Output	When asserted, indicates that l2_txstatus_data and l2_txstatus_error[6:0] are driving valid data.
l2_txstatus_data[39:0]	Output	Specifies information about the transmit frame. The following fields are defined: • [Bit 39]: When asserted, indicates a PFC frame • [Bit 38]: When asserted, indicates a unicast frame • Bit[37]: When asserted, indicates a multicast frame • Bit[36]: When asserted, indicates a broadcast frame • Bit[35]: When asserted, indicates a pause frame • Bit[34]: When asserted, indicates a control frame • Bit[33]: When asserted, indicates a VLAN frame • Bit[32]: When asserted, indicates a stacked VLAN frame • Bits[31:16]: Specifies the frame length from the first byte of the destination address to the last bye of the FCS • Bits[15:0]: Specifies the payload length
l2_txstatus_error[6:0]	Output	Specifies the error type in the transmit frame. The following fields are defined: • Bits[6:3]: Reserved • Bit[2]: Payload length error • Bit[1]: Oversized frame • Bit[0]: Reserved.

Figure 12. 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]` , 0xabe4233 . . . 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.

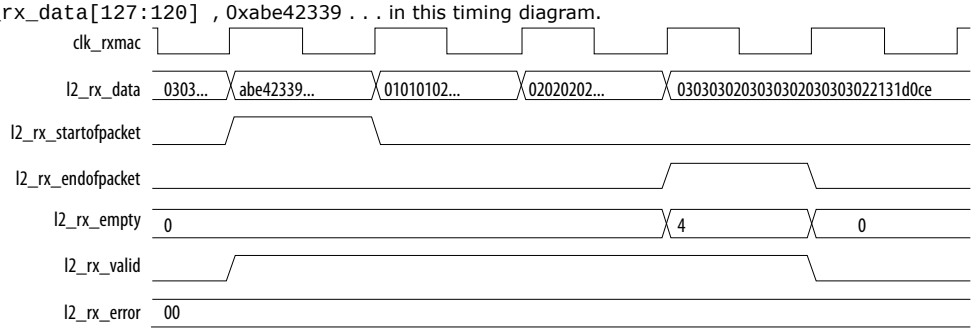
Signal	Direction	Description
`clk_rxmac`	Output	Clock for the RX MAC. Recovered from the incoming data. This clock is guaranteed stable when `rx_pcs_ready` is asserted. The frequency of this clock is 312.5 MHz. All RX MAC interface signals are synchronous to `clk_rxmac`.
`l2_rx_data[127:0]`	Output	Data 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 PREAMBLE PASS-THROUGH mode. Valid when `l2_rx_startofpacket` is asserted.
`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.

continued...

Signal	Direction	Description
`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: • `l2_rx_error[5]`: Unused. • `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. • `l2_rx_error[3]`: Oversized frame. The frame size is greater than the value specified in the `MAX_RX_SIZE_CONFIG` register. • `l2_rx_error[2]`: Undersized frame – The frame size is less than 64 bytes. Frame size = header size + payload size. • `l2_rx_error[1]`: CRC Error. The computed CRC value differs from the received CRC. • `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.
`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: • [Bit 39]: When asserted, indicates a PFC frame • [Bit 38]: When asserted, indicates a unicast frame • Bit[37]: When asserted, indicates a multicast frame • Bit[36]: When asserted, indicates a broadcast frame • Bit[35]: When asserted, indicates a pause frame • Bit[34]: When asserted, indicates a control frame • Bit[33]: When asserted, indicates a VLAN frame • Bit[32]: When asserted, indicates a stacked VLAN frame • Bits[31:16]: Specifies the frame length from the first byte of the destination address to the last bye of the FCS • Bits[15:0]: Specifies the payload length

Figure 13. 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. Transceivers

The transceivers require a separately instantiated advanced transmit (ATX) PLL to generate the high speed serial clock. For Low Latency E-Tile 40G 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 15. 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 156.25 MHz.

6.4. 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 16. 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. **All signals transceiver reconfiguration interface signals are synchronous to reconfig_clk**.
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.
reconfig_address[20:0]	Input	PMA reconfiguration interface address bus. For more information, refer to the *E-Tile Transceiver PHY User Guide*.
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.

Related Information

E-Tile Transceiver PHY User Guide
Provides more information about the transceiver reconfiguration interface, including timing diagrams for reads and writes.

6.5. 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 17. Avalon Memory-Mapped Management Interface

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

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

6.6. Miscellaneous Status and Debug Signals

The miscellaneous status and debug signals are asynchronous.

Table 18. Avalon Memory-Mapped Interface

Signal	Direction	Description
`tx_lanes_stable`	Output	Asserted when all TX lanes are stable and ready to transmit data.
`rx_block_lock`	Output	Asserted when all lanes have identified 66-bit block boundaries in the serial data stream.
`rx_am_lock`	Output	Asserted when all lanes have identified alignment markers in the data stream.

continued...

Signal	Direction	Description
rx_pcs_ready	Output	Asserted when the RX lanes are fully aligned and ready to receive data.
local_fault_status	Output	Asserted when the RX MAC detects a local fault. This signal is available only if you turn on **Enable link fault generation** in the parameter editor.
remote_fault_status	Output	Asserted when the RX MAC detects a remote fault. This signal is available only if you turn on **Enable link fault generation** in the parameter editor.

6.7. Reset Signals

The IP core has three external hard reset inputs. These resets are asynchronous and are internally synchronized. Assert resets for ten 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. rx_pcs_ready and tx_lanes_stable are asserted when the IP core has exited reset successfully.

Table 19. 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 tx_lanes_stable 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 rx_pcs_ready 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 registers. This reset leads to the deassertion of the tx_lanes_stable and rx_pcs_ready output signals.

Related Information

Reset on page 30

6.8. Clocks

You must set the transceiver reference clock (clk_ref) frequency to a value that the IP core supports. The Low Latency E-Tile 40G Ethernet IP core supports a clk_ref frequency of 156.25 MHz ±100 ppm. The ±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 E-Tile 40G Ethernet IP core performs the filtering.

Table 20. Clock Inputs

Describes the input clocks that you must provide.

Signal Name	Description
`clk_ref`	The input clock `clk_ref` is the reference clock for the transceiver RX CDR PLL. This clock must have a frequency of 156.25 MHz with a ±100 ppm accuracy per the *IEEE 802.3ba-2010 Ethernet Standard.* In addition, `clk_ref` must meet the jitter specification of the *IEEE 802.3ba-2010 Ethernet Standard*. 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_status`	Clocks the control and status interface. The clock quality and pin chosen are not critical. `clk_status` 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. `reconfig_clk` is expected to be a 100–161 MHz clock.

Table 21. Clock Outputs

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

Signal Name	Description
`clk_txmac`	The TX clock for the IP core is `clk_txmac`. The TX MAC clock frequency is 312.5 MHz.
`clk_rxmac`	The RX clock for the IP core is `clk_rxmac`. The RX MAC clock frequency is 312.5 MHz. This clock is only reliable when `rx_pcs_ready` has the value of 1. The IP core generates `clk_rxmac` from a recovered clock that relies on the presence of incoming RX data.
`clk_rx_recover`	RX recovered clock. This clock is available only if you turn on **Enable SyncE** in the Low Latency E-Tile 40G Ethernet parameter editor. The RX recovered clock frequency is 156.25 MHz during normal operation.

6.9. Flow Control Interface

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

Table 22. Flow Control Signals

Signal Name	Direction	Description
`pause_insert_tx0[(FCQN-1):0]` `pause_insert_tx1[(FCQN-1):0]`	Input	This signal is available if you specify pause on PFC. The signal indicates to the MAC whether XON or XOFF Pause or PFC flow control frame should be sent. • FCQN = 1 for Pause • FCQN = 1 ~ 8 for PFC The request for XON/XOFF flow control frame transmission can be done in either 1 or 2-bit request mode. **1-bit mode request model:** The IP core ignores `pause_insert_tx1[(FCQN-1):0]`. The following encoding is defined: • 0: No request • 0 to 1: Generate XOFF request • 1: Continue to generate XOFF request • 1 to 0: Generate XON request

continued...

Signal Name	Direction	Description
		2-bit mode request model: 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]`. The XON/XOFF request is a level-based request. The following encoding is defined: • 00: No further XON/XOFF request. In an XON/XOFF flow control frame is in progress, it is sent. • 01: Generate XON flow control frame and continuously sends them thereafter • 10: Generate XOFF flow control frame and continuously sends them thereafter • 11: Invalid request
`pause_receive_rx[(FCQN-1):0]`	Output	Each `pause_receive_rx[(FCQN-1):0]` bit indicates the corresponding queue is being paused. This is a level-based signal.

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 23. 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 24. 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

continued...

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

(5) Register value convert in decimal.

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: • Bit[0]: : sys_rst. Full system reset (except registers). Set this bit to initiate the internal reset sequence. • Bit[1]: soft_txp_rst. TX soft reset. Resets TX PCS and TX MAC. • Bit[2]: soft_rxp_rst. RX soft reset. Resets RX PCS and RX MAC. • Bits[31:3]: Reserved.	29'hX_3'b0 (4)	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 (4)	RO
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 EIO_FLAGS register. The value in the EIO_FLAG_SEL register directs the IP core to make available the following FIFO flag: • 3'b000: TX FIFO full • 3'b001: TX FIFO empty • 3b010: TX FIFO partially full • 3'b011: TX FIFO partially empty • 3b100: RX FIFO full • 3b101: RX FIFO empty • 3b110: RX FIFO partially full • 3b111: RX FIFO partially empty	29'hX3'b0 (4)	RW
0x315	EIO_FLAGS	PCS indirect data. To read a FIFO flag, set the value in the EIO_FLAG_SEL register to indicate the flag you want to read. After you specify the flag in the EIO_FLAG_SEL register, each bit [n] in the EIO_FLAGS register has the value of that FIFO flag for the transceiver channel for lane [n].	28'hX4'b0 (4)	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 (4)	RO
0x322	PHY_CLK	The following encodings are defined: • Bit[0]: Indicates if the TX PCS is ready	29'hX3'b00 (4)	RO
0x323	FRM_ERR	Each asserted bit indicates that the corresponding virtual lane has a frame error. You can read this register to determine if the IP core sustains a low number of frame	28'hX_4'b0 (4)	RO

continued...

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

(5) Register value convert in decimal.

Addr	Name	Description	Reset	Access
		errors, below the threshold to lose word lock. These bits are sticky, unless the IP core loses word lock. Write 1'b1 to the SCLR_FRM_ERR register to clear. If the IP core loses word lock, it clears this register.		
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 both physical lanes.	0x0000	RW
0x326	RX_PCS_FULLY_ALIGNED_S	Indicates the RX PCS is fully aligned and ready to accept traffic. • Bit[0]: RX PCS fully aligned status.	31'hX1'b0 (4)	RO
0x329	LANE_DESKEWED	The following encodings are defined: • Bit [0]: Indicates all lanes are deskewed. • 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.	30'hX2'b00 (4)	RO
0x331	PHY_RX_DELAY	Test and debug feature to test deskew. Allows you to insert a programmable skew on one physical channel. • Bit[0]: Allows you to you to shift 33 bits for incoming stream in physical channel 0, effectively inserting sew on two of the virtual lanes. • Bit[5:1]: Allows you to insert a 1-32 word delay on the two virtual channels arriving on physical lane 0. To use bits[5;1], you must set an RX PCS dynamic parameter that is not exposed at the top level (DYNAMIC_SKEW).	26'hX_6'b0	RW
0x341	KHZ_RX	The register indicates the value of RX clock (clk_rxmac) frequency. Apply the following definition for the frequency value: [(Register value (5) * 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 (5) * clk_status)/10] KHZ	0x0000 0000	RO

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

(5) Register value convert in decimal.

7.2. TX MAC Registers

Table 25. 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	Link Fault Configuration Register. The following bits are defined: • `Force Remote Fault` bit[3]: When link fault generation is enabled, stops data transmission and forces transmission of a remote fault. • `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. • `Unidir Enable` bit[1]: When asserted, the core includes Clause 66 support for the remote link fault reporting on the Ethernet link. • `Link Fault Reporting Enable` bit[0]: The following encodings are defined: — 1'b1: The PCS generates the proper fault sequence on Ethernet link, when conditions are met. — 1'b0: The PCS does not generate the fault sequence.	28'hX_4'b0001 (6)	RW
0x407	MAX_TX_SIZE_CONFIG	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.	0xXXXX 2580 (6)	RW
0x40A	TX_MAC_CONTROL	TX MAC Control Register. A single bit is defined: • Bit[1]: VLAN detection disabled. This bit is deasserted by default, implying VLAN detection is enabled.	30'hX2'b0X (6)	RW

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

7.3. RX MAC Registers

Table 26. RX MAC Registers

Addr	Name	Description	Reset	Access
0x500	RXMAC_REVID	RX MAC revision ID for Low Latency E-Tile 40G EthernetIP 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.	0xXXXX 2580 (7)	RW
0x507	MAC_CRC_CONFIG	The RX CRC forwarding configuration register. The following encodings are defined: • 1'b0: Remove RX CRC, do not forward it to the RX client interface • 1'b1: Retain RX CRC, forward it to the RX client interface In either case, the IP core checks the incoming RX CRC and flags errors.	31'hX1'b0 (7)	RW
0x508	LINK_FAULT	Link Fault Status Register. For unidirectional Link Fault, implements *IEEE 802.3 Ethernet Clause 66.*	30'hX2'b00 (7)	RO
0x50A	RX_MAC_CONTROL	RX MAC Control Register. • Bit [1]: VLAN detection disabled. This bit is deasserted by default implying VLAN detection is enabled.	30'h0_2'b0X (7)	RW

7.4. Pause/PFC Flow Control Registers

Some of the registers in this table cannot be updated during normal operation. To ensure correct operation, perform a soft reset by writing Bit[0] of the PHY_CONFIG (0x310) after updating registers that cannot be changed dynamically.

Table 27. TX Flow Control Registers

Addr	Bit	Name	Description	Reset	Access
0x600	31:0	TX Flow Control Revision ID	Specifies the revision ID, "100GFCTx CSR"	0x0809_2 0017	RO
0x601	31:0	TX Flow Control Scratch Pad	Scratch register for testing.	0	RW

continued...

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

Addr	Bit	Name	Description	Reset	Access
0x602	31:0	TX Flow Control IP Core Variant 0	Specifies first 4 characters of IP core variation identifier ASCII string, "100G".	0x3130_3047	RO
0x603	31:0	TX Flow Control IP Core Variant 1	Next 4 characters of IP core variation identifier ASCII string, "FCTx".	0x4643_5478	RO
0x604	31:0	TX Flow Control IP Core Variant 2	Final 4 characters of IP core variation identifier ASCII string, "xCSR".	0x0043_5352	RO
0x605	7:0	TX Flow Control Enable One bit per queue	Enables the IP core to generate XON and XOFF Pause/PFC flow control frames to the remote partner. The following encodings are defined: • 1'b0: XON or XOFF Pause/PFC flow control is disabled. • 1'b1: XON or XOFF Pause/PFC flow control is enabled. You can change this field dynamically.	0xFF	RW
	31:8	Reserved	Reserved	0	RO
0x606	7:0	TX Flow Control CSR XON/XOFF Request 0	XON/XOF flow control frame request bit 0. Interpretation depends on whether the IP core is in 1-bit FC request mode or in 2-bit FC request mode. This register affects a flow control queue only if the corresponding bit of the TX Flow Control Enable register has the value of 1. In 2-bit mode, in addition, this register is active for a specific flow control queue only if the corresponding bit in the TX 2-bit Flow Control Request Mode register field (bits 7:0 of the register at offset 0x641) specifies that the flow control logic accepts input from this register. The following encodings are defined for 1-bit mode. The IP core reads the 1-bit mode value in TX Flow Control CSR XON/XOFF Request 0. • 0 = No request • 0 to 1 = Generate XOFF request • 1 = Continue to generate XOFF request • 1 to 0 = Generate XON request The following encodings are defined for 2-bit mode. The IP core reads the 2-bit mode value in {TX Flow Control CSR XON/XOFF Request 1, TX Flow Control CSR XON/XOFF Request 1}. • 00 = No request • 01 = XON request • 10 = XOFF request • 11 = Invalid You can modify the value of this field dynamically.	0	RW
	15:8	Reserved	Reserved	0	RO

continued...

Addr	Bit	Name	Description	Reset	Access
	23:16	TX Flow Control CSR XON/XOFF Request 1	In conjunction with Flow Control XON/XOFF Request 0 specifies a 2-bit request for XON/XOFF flow control frame transmission. This bit is the upper bit of the 2-bit control field. You can change the value of this field dynamically.	0	RW
	31:24	Reserved	Reserved	0	RO
0x60A	0	TX Pause Enable 1-bit	Determines whether receiving a valid Pause frame stops TX user data transmission. 1'b0: Transmission is not stopped 1'b1: Transmission stops You cannot change the value of this field dynamically.	0	RW
	31:1	Reserved	Reserved	0	RO
0x60D	31:0	TX Flow Control Destination Address Lower	Specifies the 48-bit Destination Address of the flow control frame. Contains the 32 LSB of the address field. You cannot modify the value of this field dynamically.	0xC2000001	RW
0x60E	15:0	TX Flow Control Destination Address Upper	Specifies the 48-bit Destination Address of flow control frame. Contains the 16 MSB of the address field. You cannot modify the value of this field dynamically.	0x0180	RW
	31:16	Reserved	Reserved	0	RO
0x60F	31:0	TX Flow Control Source Address Lower	Specifies the 48-bit Source Address of flow control frame. Contains the 32 LSB of the address field.	0xCBFC5ADD	RW
0x610	15:0	TX Flow Control Source Address Upper	Specifies the 48-bit Source Address of flow control frame. Contains the 16 MSB of the address field. You cannot modify the value of this field dynamically.	0xE100	RW
	31:16	Reserved	Reserved	0	RO
0x620, 0x621, …, 0x620+ (FCQN-1 where FCQN = 0 to 7)	15:0	TX Flow Control Quanta 16-bit per FCQN	Specifies the pause quanta of Pause/PFC flow control frames to be sent to remote partner. You cannot modify the value of this field dynamically.	0xFFFF	RW
	31:16	Reserved	Reserved	0	RO
0x628, 0x629, …, 0x628+ (FCQN-1 where FCQN = 0 to 7)	15:0	TX Flow Control Signal XOFF Request Hold Quanta 16-bit per FCQN	Specifies the separation between 2 consecutive XOFF flow control frames. You cannot modify the value of this field dynamically.	0xFFFF	RW
	31:16	Reserved	Reserved	0	RO
0x640	0	TX Flow Control Select 1-bit	Specifies whether the TX hardware generates Pause or PFC frames. Affects only PFC Queue 0.	1	RW

continued...

Addr	Bit	Name	Description	Reset	Access
			Usage example: You can synthesize a single PFC queue and use it for both Pause and PFC purpose. 1'b0: Pause 1'b1: PFC You cannot modify the value of this field dynamically.		
	31:1	Reserved.	Reserved.	0	RO
0x641	(FCQN-1): 0	TX 2-bit Flow Control Request Mode 1-bit per FCQN	Determines whether the TX Flow Control CSR XON/XOFF Request register or the pause_insert_tx0 and pause_insert_tx1 signals control XON/XOFF mode in 2-bit control mode. 1'b0: The pause_insert_tx0 and pause_insert_tx1 signals control requests 1'b1: The TX Flow Control CSR XON/XOFF Request register fields control requests You cannot modify the value of this field dynamically.	0	RW
	16	TX Flow Control Request Mode 1 bit for all FCQN	Determines whether the IP core is in TX flow control 1-bit mode or 2-bit mode. 1'b0: Use 1-bit mode to make TX flow control requests 1'b1: Use 2-bit mode to make TX flow control requests	0	RW
	31:17	Reserved	Reserved	0	RO

Table 28. RX Flow Control Registers

Addr	Bit	Name	Description	Reset	Access
0x700	31:0	RX Flow Control Revision ID	Specifies the revision ID, "100GFCTx CSR"	0x0809_2017	RO
0x701	31:0	RX Flow Control Scratch Pad	Provides a register for debug.	0	RW
0x702	31:0	RX Flow Control IP Core Variant 0	First 4 characters of IP core variation identifier ASCII string, Specifies first 4 characters of IP core variation identifier ASCII string, "100G".	0x3130_3047	RO
0x703	31:0	RX Flow Control IP Core Variant 1	Next 4 characters of IP core variation identifier ASCII string, "FCRx".	0x4643_5278	RO
0x704	31:0	RX Flow Control IP Core Variant 2	Final 4 characters of IP core variation identifier ASCII string, "0CSR". The "0" is unprintable.	0x0043_5352	RO
0x705	7:0	RX PFC Enable 1 bit per queue	Determines whether receiving a valid PFC frame causes the PFC duration user interface to indicate a valid pause quanta duration to the user logic. 1'b0: Disable 1'b1: Enable	0xFF	RW

continued...

Addr	Bit	Name	Description	Reset	Access
			You cannot modify the value of this field dynamically.		
	31:8	Reserved	Reserved	0	RO
0x707	31:0	`RX Flow Control Destination Address Lower`	Specifies the 48-bit `Destination Address` of the flow control frame. Contains the 32 LSB of the address field. The flow control frame is sent with the destination address matching the address specified in this register or the multicast address. If the address is not a match, the Flow Control block does not respond to the incoming frame; the IP core just passes it through. You cannot modify the value of this field dynamically.	0xC2000001	RW
0x708	15:0	`RX Flow Control Destination Address Upper`	Specifies the 48-bit `Destination Address` of flow control frame. Contains the 16 MSB of the address field. The flow control frame is sent with the destination address matching the address specified in this register or the multicast address. If the address is not a match, the Flow Control block does not respond to the incoming frame; the IP core just passes it through. You cannot modify the value of this field dynamically.	0x0180	RW
	31:16	Reserved	Reserved	0	RO

8. Debugging the Link

The following steps should help you identify and resolve common problems that occur when bringing up a Low Latency E-Tile 40G 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.

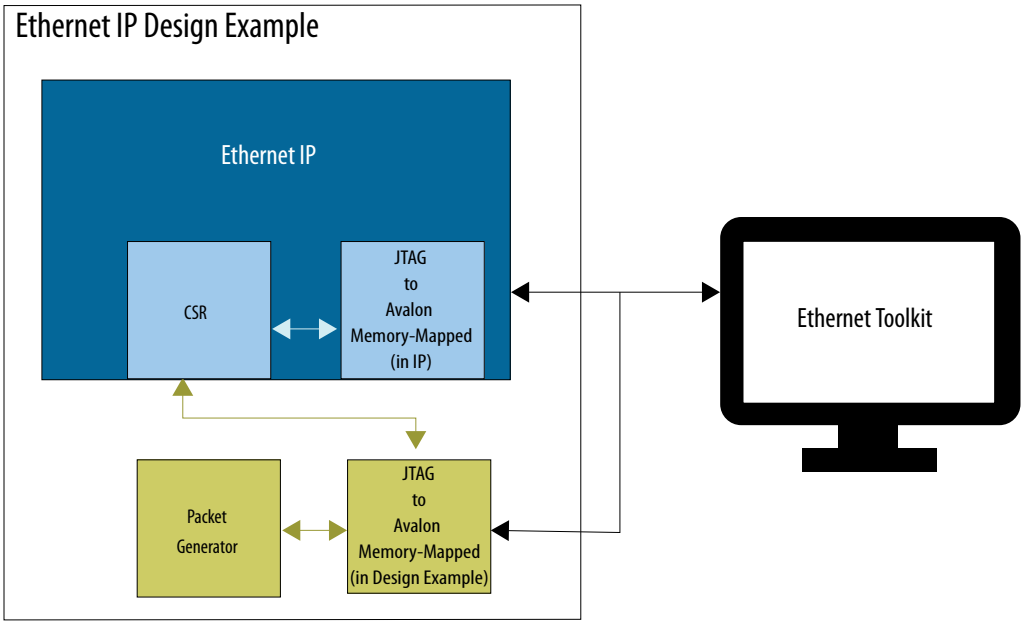
In addition, your IP core can experience loss of signal on the Ethernet link after it is established. In this case, the TX functionality is unaffected, but the RX functionality is disrupted. The following symptoms indicate a loss of signal on the Ethernet link:

- 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 14. 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 IPs 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 E-Tile 40G Ethernet Intel FPGA IP User Guide Archives

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	20.0.0	Low Latency E-Tile 40G Ethernet Intel FPGA IP User Guide
20.1	19.1.0	Low Latency E-Tile 40G Ethernet Intel FPGA IP User Guide

11. Comparison Between Low Latency E-Tile 40G Ethernet Core and Low Latency 40GbE IP Core

Table 29. Comparison Between the Low Latency E-Tile 40G Ethernet Core and the Low Latency 40G Ethernet Intel FPGA IP Core

The comparison is defined relative to these versions:

- Low Latency E-Tile 40G Ethernet IP core available with the Intel Quartus Prime Pro Edition software release v20.1.
- Low Latency 40G Ethernet Intel FPGA IP core available with the Quartus Prime Pro 17.1 Stratix 10 ES Editions software.

This table does not list individual signal name and register differences.

Property	Low Latency E-Tile 40G Ethernet IP Core	Low Latency 40G Ethernet Intel FPGA IP Core
Transceiver tile support	E-tile	L-tile, H-tile
Device support	Supports Intel Stratix 10 and Intel Agilex device family.	Supports Intel Stratix 10 device family.
Reset	Provides three asynchronous hard reset signals (general, receiver only, and transmitter only) and three soft reset register bits.	Provides three asynchronous hard reset signals (general, receiver only, and transmitter only) and three soft reset register bits.
Client interface width	Avalon streaming interface 128-bit data bus	Avalon streaming interface 128-bit data bus
Avalon streaming transmitter interface readyLatency	Avalon streaming transmitter interface readyLatency configurable at 0 or 3 (parameter).	Avalon streaming transmitter interface readyLatency configurable at 0 or 3 (parameter).
Preamble passthrough	Available as a configuration option (parameter). When preamble passthrough is turned on, you must provide the preamble on a separate bus, `l2_tx_preamble[63:0]`, and the IP core provides the RX preamble on a separate bus, `l2_rx_preamble[63:0]`.	Available as a configuration option (parameter). When preamble passthrough is turned on, you must provide the preamble on a separate bus, `l2_tx_preamble[63:0]`, and the IP core provides the RX preamble on a separate bus, `l2_rx_preamble[63:0]`.
Interface to transceiver TX PLL	Not required.	You must instantiate a single TX PLL IP core to connect to the single `tx_serial_clk` input pin of the Low Latency 40G Ethernet Intel FPGA IP core.
Statistics counters	Available as a configuration option (parameter).	Available as a configuration option (parameter).
Statistics counter increment vectors	`l2_txstatus_data`, `l2_txstatus_error`, and `l2_rxstatus_data` signals available on client interface, whether or not statistics registers are enabled.	`l2_txstatus_data`, `l2_txstatus_error`, and `l2_rxstatus_data` signals available on client interface, whether or not statistics registers are enabled.
40GBASE-KR4	Not supported.	Available as a configuration option. Configurable support for 40GBASE-KR4 or 40GBASE-CR4. Implements the *IEEE Backplane Ethernet Standard 802.3-2012*.

continued...

Property	Low Latency E-Tile 40G Ethernet IP Core	Low Latency 40G Ethernet Intel FPGA IP Core
Flow control	Available as a configuration option (parameter).	Available as a configuration option (parameter).
1588 PTP support	Not supported.	Not supported.
Enable alignment of EOP on FCS word	Always turned on.	Always turned on.
Minimum average interpacket gap	Value is 12 bytes.	Value is 12 bytes.

Related Information

Low Latency 40-Gbps Ethernet IP Core User Guide

12. Document Revision History

Document Version	Intel Quartus Prime Version	IP Version	Changes
2020.10.05	20.3	21.0.0	• Corrected the width of the reconfiguration address from `reconfig_address[21:0]` to `reconfig_address[20:0]`. • Revised the *Flow Control Signals* table to update the following signal description: — `pause_insert_tx0[(FCQN-1):0]` — `pause_insert_tx1[(FCQN-1):0]` — `pause_insert_rx[(FCQN-1):0]` • Added new section: *Ethernet Toolkit Overview*.
2020.06.22	20.2	20.0.0	• Added support for the Intel Agilex device family. • Removed **VCCR_GXB and VCCT_GXB supply voltage for the transceiver** parameter from the *Low Latency E-Tile 40G Ethernet IP Core Parameters: Main Tab* section. This parameter is not utilized in the Low Latency E-Tile 40G Ethernet Intel FPGA IP. • Updated `reconfig_address[21:0]` description in the *Transceiver Reconfiguration Signals* section.
2020.04.20	20.1	19.1.0	Initial release.