1G/10G/25G Switching Ethernet Subsystem v2.0

LogiCORE IP Product Guide

Vivado Design Suite

PG292 April 4, 2018

XILINX
ALL PROGRAMMABLE™

Table of Contents

Introduction

The Xilinx® 1/10/25G Ethernet dynamically switching MAC and PCS/PMA Subsystem provides a flexible solution for connection to transmit and receive data interfaces using AXI4-Stream interfaces.

Features

- Designed to the Ethernet requirements for 1/10 Gb/s operation specified by IEEE 802.3 Clause 49 or Clause 36
- Runtime switchable Ethernet MAC and PCS/PMA functions for 1/10/25 Gb/s operation
- Supports only the GTHE3/GTYE3 and GTHE4/GTYE4 transceiver supported devices.
- AXI-Lite interface for control bus
- Simple packet-oriented user interface
- Comprehensive statistics gathering
- Status signals for all major functional indicators
- Delivered with a top-level wrapper including functional transceiver wrapper, IP netlist, sample test scripts, and Vivado® Design Suite tools compile scripts
- PCS sublayer operating at 25.78125 Gb/s or 10.3125 Gb/s or 1.25 Gb/s
- Optional IEEE 1588 two-step hardware timestamping
- Single channel (lane) support

Facts Table	
Core Specifics	
Supported Device Family(1)	Zynq® UltraScale+™ MPSoC UltraScale and UltraScale+ Families
Supported User Interfaces	AXI4-Stream and AXI Lite for all variants, XGMII and GMII for PCS-only variants
Resources	See Performance and Resource Utilization web page.
Provided with Core	
Design Files	Encrypted register transfer level (RTL)
Example Design	Verilog
Test Bench	Verilog
Constraints File	Xilinx Design Constraints (XDC)
Simulation Model	Verilog
Supported S/W Driver	NA
Tested Design Flows(2)	
Design Entry	Vivado Design Suite
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide.
Synthesis	Vivado Synthesis
Support	
Provided by Xilinx at the Xilinx Support web page	

Notes:

1. For a complete list of supported devices, see the Vivado IP catalog.
2. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Overview

This document details the features of the Ethernet 1/10/25G dynamically switching PCS/PMA and MAC Core. 10G/25G Ethernet subsystem is defined by the 25G Ethernet Consortium[Ref 2]. 10G PCS functionality is defined by *IEEE Standard 802.3, 2015, Clause 49, Physical Coding Sublayer (PCS) for 64B/66B, type 10GBASE-R* [Ref 1]. 1G PCS functionality is defined in Clause 36. For 25G operation, clock frequencies are increased to provide a serial interface operating at 25.78125 Gb/s to leverage the latest high-speed serial transceivers. The low latency design is optimized for UltraScale+™ architecture devices.

Feature Summary

See Table 1-1 for compatibility of options with the different variants of the LogiCORE™ IP core.

1G/10G/25G Supported Features

- Complete MAC and PCS functions
- 10G BASE-R mode based on IEEE 802.3 Clause 49 or 1000BASE-X mode based on IEEE 802.3 Clause 36.
- 32-bit/64-bit AXI4-Stream user interface
- AXI4-Lite control and status interface
- Statistics and diagnostics
- Custom preamble and adjustable Inter Frame Gap

Table 1-1: **Feature Compatibility Matrix**

Variant	User Interface	MAC	PCS	Pause Processing	Auto-Negotiation and Link Training	Clause 74 FEC	Clause 108 RS-FEC	IEEE 1588 Hardware Time Stamp
10G MAC with PCS[(1)]	32-bit AXI4-Stream	✓	✓					✓
25G MAC/PCS	64-bit AXI4-Stream	✓	✓					
10G PCS only	XGMII and GMII		✓					

Notes:
1. Only 2-step timestamping is supported.

Applications

IEEE Std 802.3 enables several different Ethernet speeds for Local Area Network (LAN) applications. IP delivers capability to switch between 10GBASE-R and 1000BASE-X PHY.

Licensing and Ordering

License Checkers

If the IP requires a license key, the key must be verified. The Vivado® design tools have several license checkpoints for gating licensed IP through the flow. If the license check succeeds, the IP can continue generation. Otherwise, generation halts with error. License checkpoints are enforced by the following tools:

- Vivado Synthesis
- Vivado Implementation
- write_bitstream (Tcl Console command)

IMPORTANT: *IP license level is ignored at checkpoints. The test confirms a valid license exists. It does not check IP license level.*

License Type

10G/25G Ethernet PCS/PMA (10G/25G BASE-R)

This 1G/10G/25G Ethernet Subsystem module is provided at no additional cost with the Xilinx Vivado Design Suite under the terms of the Xilinx End User License. To use the Subsystem, a 25G Ethernet MAC/PCS license must be purchased governed under the terms of the Xilinx Core License Agreement. For information about pricing and availability of other Xilinx IP modules and tools, contact your local Xilinx sales representative.

For more information, visit the 1G/10G/25G Switching Ethernet Subsystem page.

Ordering Information

To purchase any of these IP cores, contact your local Xilinx Sales Representative referencing the appropriate part number(s) in Table 1-2.

Table 1-2: **Ordering Information**

Description	Part Number	License Key
10G/25G Ethernet MAC + BASE-R (64-bit)	EF-DI-25GEMAC-PROJ(1)	xxv_eth_mac_pcs
10E MAC + PCS/PMA (32-bit)	EF-DI-25GEMAC-SITE(1)	
Standalone 10GE MAC (64-bit)		x_eth_mac

Notes:

1. Includes access to legacy 10 Gigabit Ethernet Media Access Controller - 10GEMAC for 7 series and UltraScale devices (key name: ten_gig_eth_mac).

Product Specification

Figure 2-1 show the block diagram of the 32-bit 1G/10G MAC and PCS/PMA Switching Ethernet Subsystem.

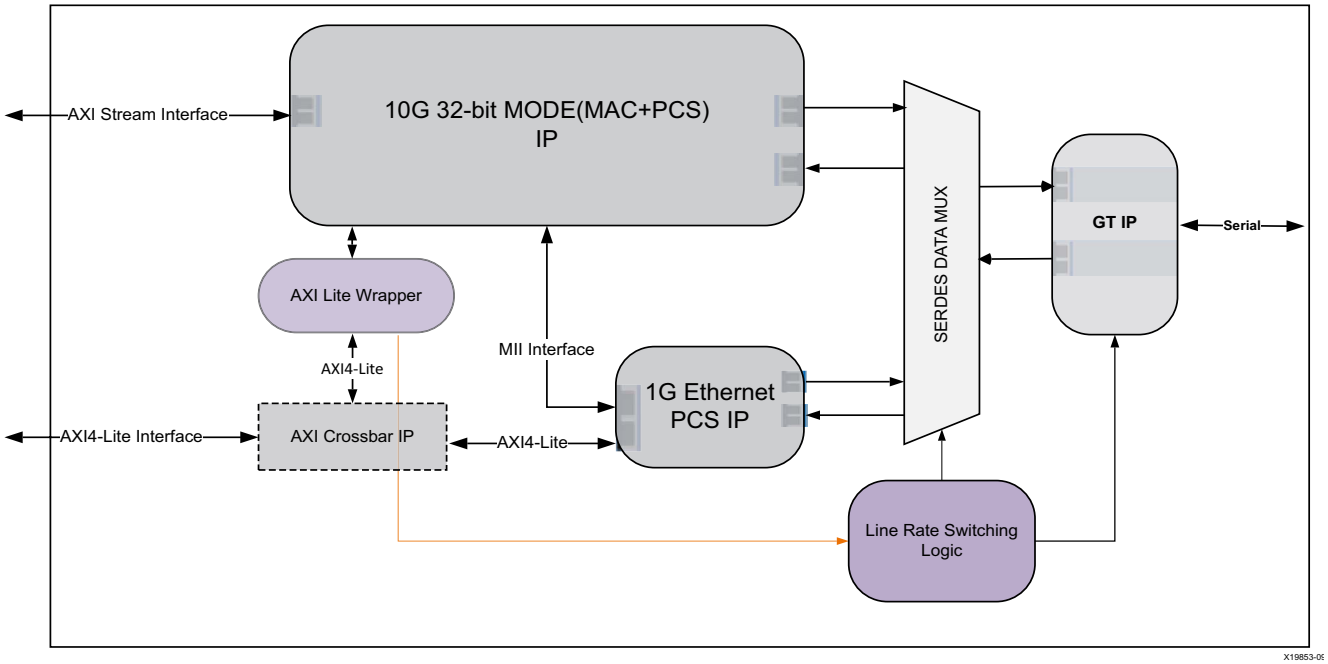

Figure 2-1: **Ethernet 1/10/25G dynamically switching 32-bit MAC and PCS/PMA IP Block Diagram**

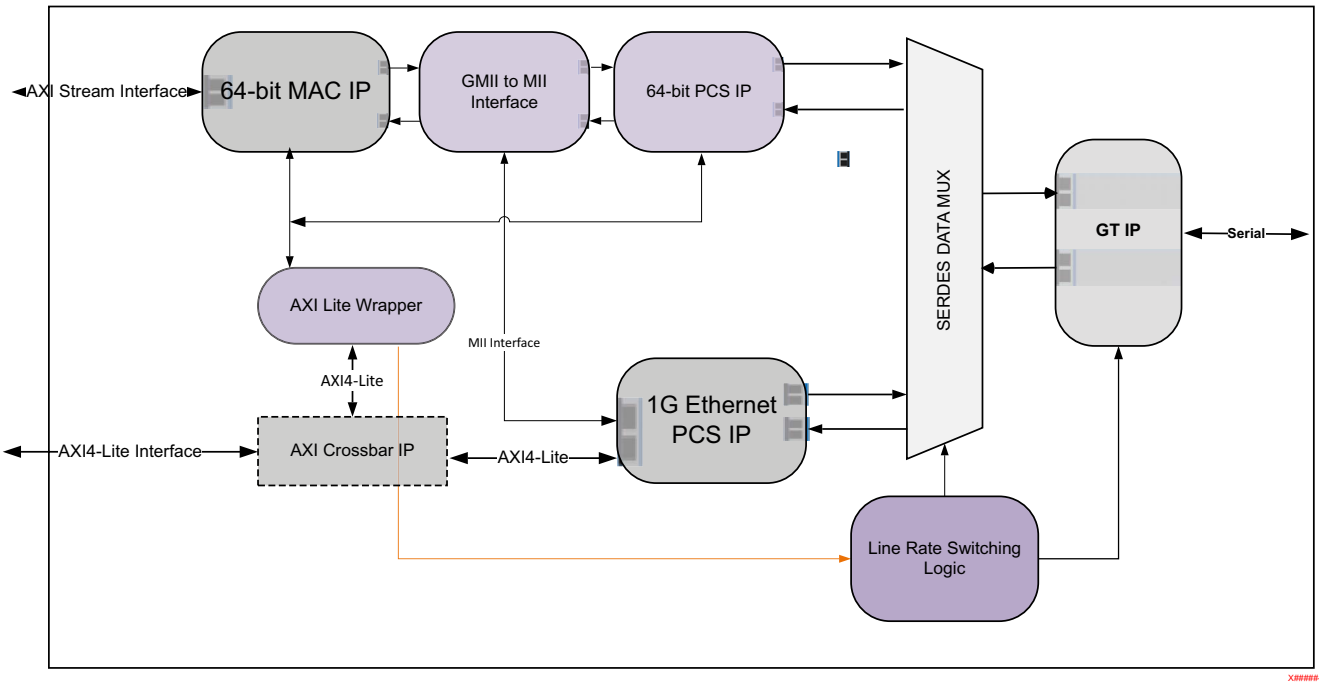

Figure 2-2: **Ethernet 1/10/25G Dynamically Switching 64-bit MAC and PCS/PMA IP Block Diagram**

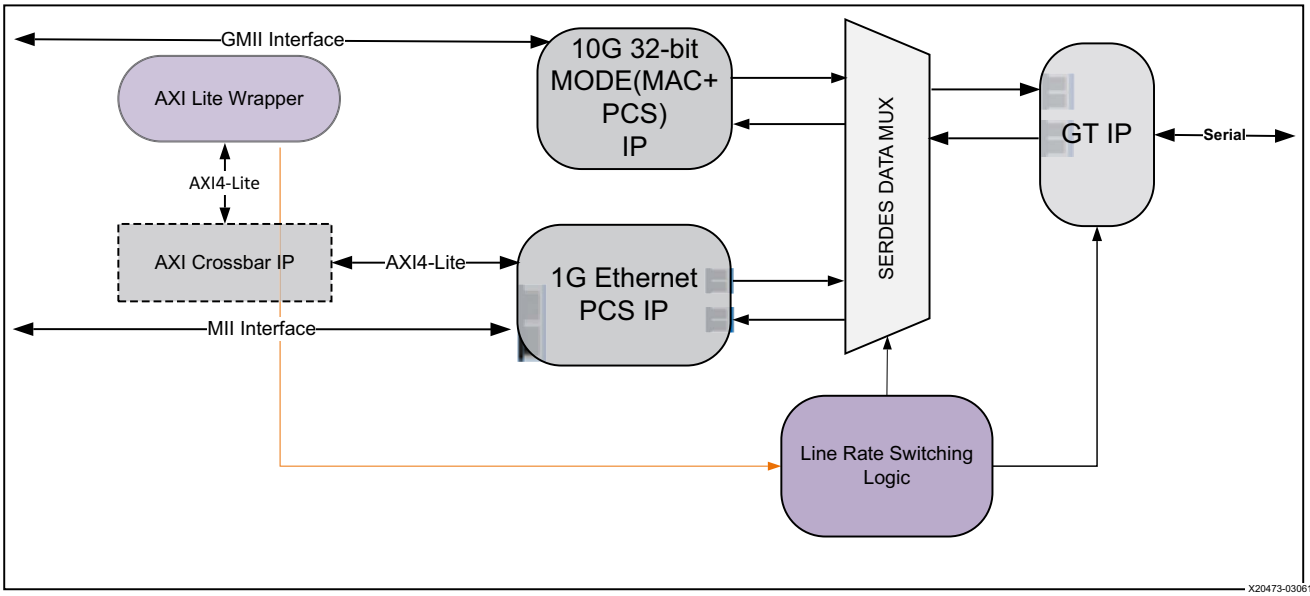

Figure 2-3: **Ethernet 1/10G Dynamically Switching 32-bit PCS/PMA IP Block Diagram**

Standards

The 1G/10G/25G Switching Ethernet Subsystem is designed to the standard specified in the *IEEE Std 802.3* [Ref 1].

Performance and Resource Utilization

For full details about performance and resource utilization, See the Performance and Resource Utilization web page.

Latency

Table 2-1 provides the measured latency information for the 1G/10G/25G Switching Ethernet Subsystem.

Table 2-1: **Latency**

Core	Core Configuration	Latency (ns)	User Bus Width (bits)	Core Clock Frequency (MHz)
32-bit MAC+PCS/PMA	10G mode TX latency	28.8	32	312.5
	1G mode TX latency	632	32	125
	10G mode RX latency	25.6	32	312.5
	1G mode RX latency	304	32	125
64-bit MAC+PCS/PMA	25G mode TX latency	77	64	390.625
	10G mode TX latency	192	64	156.25
	1G mode TX latency	864	64	125
	25G mode RX latency	113	64	390.625
	10G mode RX latency	281	64	156.25
	1G mode RX latency	568	64	125
32-bit PCS/PMA	10G mode TX latency	19.2	32	312.5
	1G mode TX latency	80	8	125
	10G mode RX latency	19.2	32	312.5
	1G mode RX latency	88	8	125

Port Descriptions – MAC+PCS Variant

The following tables list the ports for the 1G/10G/25G Switching Ethernet Subsystem. These signals are usually found at the `wrapper.v` hierarchy. These ports are applicable for both the 64-bit integrated MAC+PCS for 25 Gb/s and 10 Gb/s line rates and the low-latency 32-bit integrated MAC + PCS for 10 Gb/s line rate. When the AXI register interface is included, some of these ports are accessed through the registers instead of the broadside bus.

Transceiver Interface

Table 2-2 shows the transceiver I/O ports for the 1G/10G/25G Ethernet subsystem. See Clocking in Chapter 3 for details regarding each clock domain.

Table 2-2: **Transceiver I/O**

Name	Direction	Description	Clock Domain
gt_tx_reset	Input	Reset for the gigabit transceiver (GT) TX.	Asynch
gt_rx_reset	Input	GT RX reset.	Asynch
ctl_gt_reset_all	Input	Active-High asynchronous reset for the transceiver startup Finite State Machine (FSM). Note that this signal also initiates the reset sequence for the entire 1G/10G/25G Ethernet subsystem.	Asynch
refclk_n0	Input	Differential reference clock input for the SerDes, negative phase.	Refer to Clocking.
refclk_p0	Input	Differential reference clock input for the SerDes, positive phase.	Refer to Clocking.
rx_serdes_data_n0	Input	Serial data from the line; negative phase of the differential signal	Refer to Clocking.
rx_serdes_data_p0	Input	Serial data from the line; positive phase of the differential signal	Refer to Clocking.
tx_serdes_data_n0	Output	Serial data to the line; negative phase of the differential signal.	Refer to Clocking.
tx_serdes_data_p0	Output	Serial data to the line; positive phase of the differential signal.	Refer to Clocking.
tx_serdes_clkout	Output	When present, same as tx_clk_out.	Refer to Clocking.

AXI4-Stream Interface

The 1G/10G/25G Switching Ethernet Subsystem IP core provide 32-bit and 64-bit options of AXI4-Stream interface. For 1G / 10G switching IP, the 32-bit and 64-bit interfaces are provided. For 1G/10G/25G switching IP, only 64-bit interface is provided.

AXI4-Stream Clocks and Resets

Table 2-3: **AXI4-Stream Interface–Clock/Reset Signals**

Name	Direction	Description	Clock Domain
rx_clk_out	Output	Receive AXI4-Stream clock. All signals between the 1G/10G/25G High Speed Ethernet Subsystem and the user-side logic are synchronized to the positive edge of this signal. This clock is 125 MHz for 1G configuration, 165.25 / 312.5 MHz for 10G core configuration and 390.625 MHz for 25G configuration. When the RX FIFO is included, the RX AXI4-Stream clock is an input and should be equal to or greater than tx_clk_out.	Refer to Clocking.
tx_clk_out	Output	Transmit AXI4-Stream clock. All signals between the 1G/10G/25G High Speed Ethernet Subsystem and the user-side logic are synchronized to the positive edge of this signal. This clock is 125 MHz for 1G configuration, 165.25 / 312.5 MHz for 10G core configuration and 390.625 MHz for 25G configuration.	Refer to Clocking.
rx_reset	Input	Reset for the RX circuits. This signal is active-High (1 = reset) and must be held High until clk is stable.The core handles synchronizing the rx_reset input to the appropriate clock domains within the core.	Asynch
tx_reset	Input	Reset for the TX circuits. This signal is active-High (1 = reset) and must be held High until clk is stable. The core handles synchronizing the tx_reset input to the appropriate clock domains within the core.	Asynch
rx_core_clk	Input	The rx_core_clk signal is used to clock the receive AXI4-Stream interface. It is an input when the FIFO is included and is not an input port when in low latency mode with FIFO not included; instead it is driven internally by rx_clk_out. Clock Domain: rx_core_clk	rx_core_clk

Transmit AXI4-Stream Interface

Table 2-4 shows the AXI4-Stream transmit interface signals.

Table 2-4: **AXI4-Stream Transmit Interface Signals**

Signal	Direction	Description
tx_axis_tdata[63 or 31:0]	In	AXI4-Stream data. 32-bit and 64-bit interfaces are available. Bus width depends on the selection of 64-bit or 32-bit interfaces.
tx_axis_tkeep[7:0 or 3:0]	In	AXI4-Stream Data Control. Bus width depends on selection of 64-bit or 32-bit interfaces.
tx_axis_tvalid	In	AXI4-Stream Data Valid input
tx_axis_tuser	In	AXI4-Stream User Sideband interface. Equivalent to the tx_errin signal. 1 indicates a bad packet 0 indicates a good packet

Table 2-4: **AXI4-Stream Transmit Interface Signals *(Cont'd)***

Signal	Direction	Description
tx_axis_tlast	In	AXI4-Stream signal indicating End of Ethernet Packet.
tx_axis_tready	Out	AXI4-Stream acknowledge signal to indicate to start the Data transfer.

Data Lane Mapping

For transmit data `tx_axis_tdata`, the port is logically divided into lane 0 to lane 3 for the 32-bit interface (See Table 2-5) or lane 0 to lane 7 for the 64-bit interface (See Table 2-6) with the corresponding bit of the `tx_axis_tkeep` word signifying valid data on `tx_axis_tdata`.

Table 2-5: **tx_axis_tdata Lanes – 32 Bits**

Lane/ tx_axis_tkeep	tx_axis_tdata[31:0] Bits
0	7:0
1	15:8
2	23:16
3	31:24

Table 2-6: **tx_axis_tdata Lanes – 64 Bits**

Lane/ tx_axis_tkeep	tx_axis_tdata[63:0] Bits
0	7:0
1	15:8
2	23:16
3	31:24
4	39:32
5	47:40
6	55:48
7	63:56

Normal Transmission

The timing of a normal frame transfer is shown in Figure 2-4. When the client wants to transmit a frame, it asserts the `tx_axis_tvalid` and places the data and control in `tx_axis_tdata` and `tx_axis_tkeep` in the same clock cycle. When this data is accepted by the core, indicated by `tx_axis_tready` being asserted, the client must provide the next cycle of data. If `tx_axis_tready` is not asserted by the core, the client must hold the current valid data value until it is. The end of packet is indicated to the core by `tx_axis_tlast` asserted for 1 cycle. The bits of `tx_axis_tkeep` are set appropriately to indicate the number of valid bytes in the final data transfer. `tx_axis_tuser` is also asserted to indicate a bad packet.

After `tx_axis_tlast` is deasserted, any data and control is deemed invalid until `tx_axis_tvalid` is next asserted.

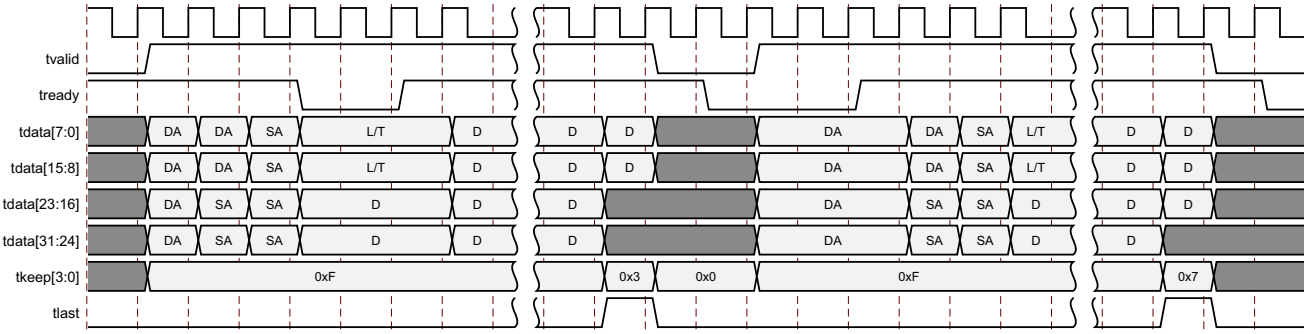

Figure 2-4: **Normal Frame Transfer – 32 Bits**

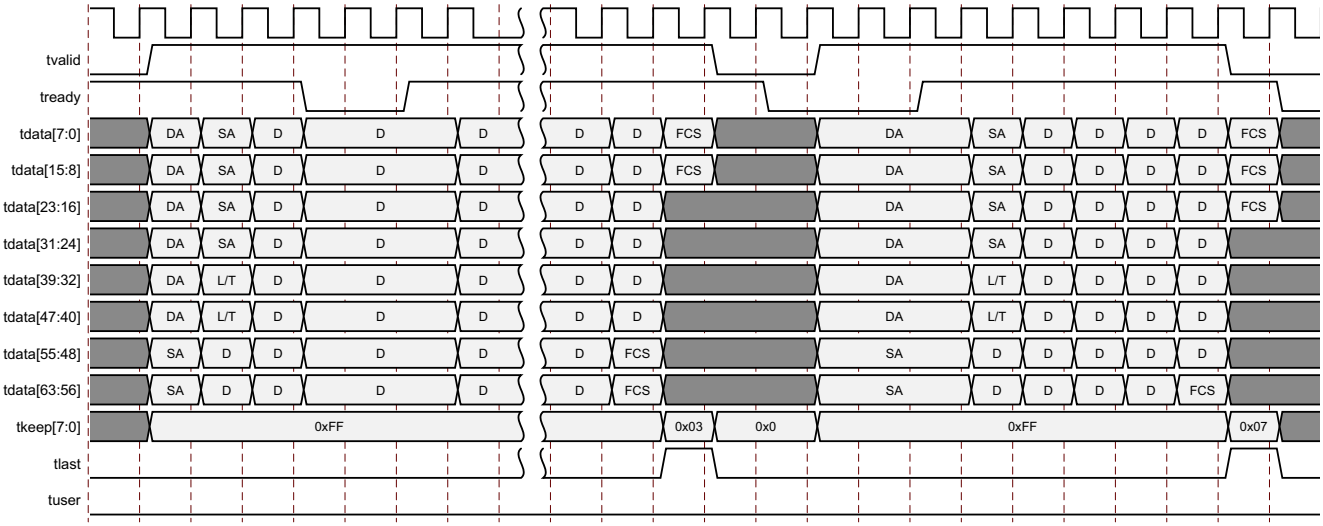

Figure 2-5: **Normal Frame Transfer - 64 Bits**

Back-to-Back Continuous Transfers

Continuous data transfer on the transmit AXI4-Stream interface is possible, as the signal `tx_axis_tvalid` can remain continuously High, with packet boundaries defined solely by `tx_axis_tlast` asserted for the end of the Ethernet packet. However, the core can deassert the `tx_axis_tready` acknowledgment signal to throttle the client data rate as required. See Figure 2-6 and Figure 2-7. The client data logic can update the AXI4-Stream interface with valid data while the core deasserts the `tx_axis_tready` acknowledgment signal. However, after valid is asserted and new data has been placed on the AXI4-Stream, it should remain there until the core asserts `tx_axis_tready` signal.

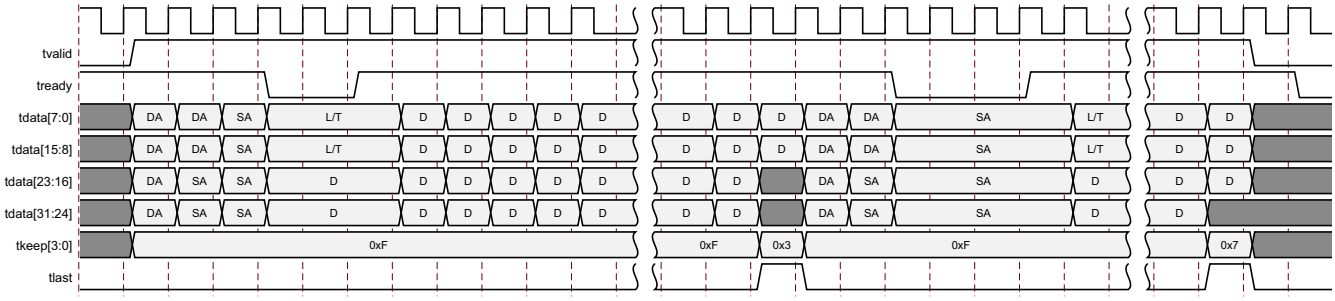

Figure 2-6: **Back-to-Back Continuous Transfer on Transmit Client Interface—32-bit**

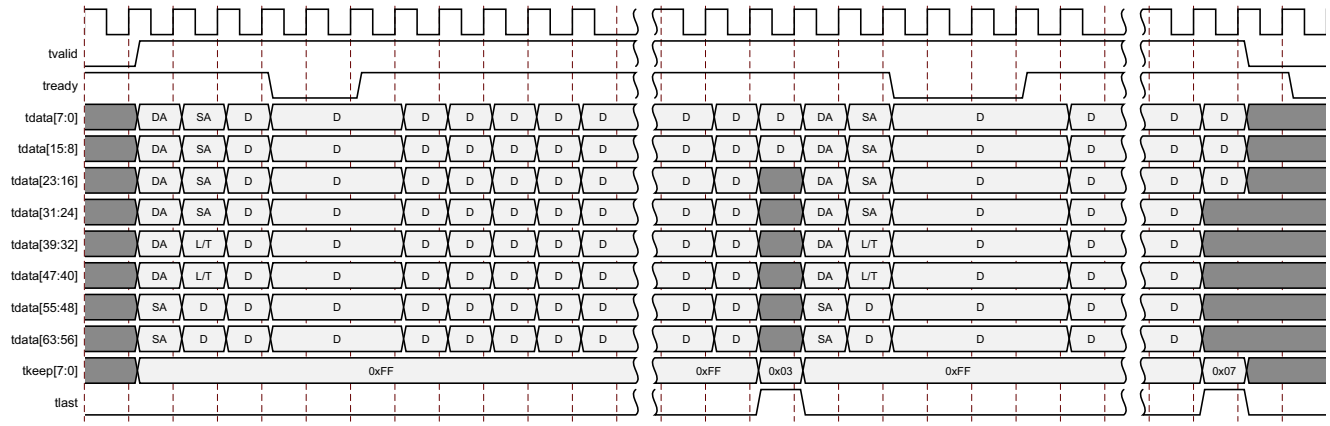

Figure 2-7: **Continuous Transfer on transmit Client Interface -64 bit**

Aborting a Transmission

The aborted transfer of a packet on the client interface is called an underrun. This can happen if a FIFO in the AXI Transmit client interface empties before a frame is completed.

This is indicated to the core in one of two ways.

- An explicit error in which a frame transfer is aborted by asserting `tx_axis_tuser` High while `tx_axis_tlast` is High.
- An implicit underrun, in which a frame transfer is aborted by deasserting `tx_axis_tvalid` without asserting `tx_axis_tlast`.

When either of the two scenarios occurs during a frame transmission, the core inserts error codes into the data stream to flag the current frame as an errored frame. It remains the responsibility of the client to re-queue the aborted frame for transmission, if necessary.

Receive AXI4-Stream Interface

Table 2-7 shows the AXI4-Stream receive interface signals.

Table 2-7: **AXI4-Stream Receive Interface Signals**

Signal	Direction	Description
rx_axis_tdata[63 or 31:0]	Out	AXI4-Stream Data to upper layer. Bus width depends on 64-bit or 32-bit selection.
rx_axis_tkeep[7 or 3:0]	Out	AXI4-Stream Data Control to upper layer. Bus width depends on 64-bit or 32-bit selection.
rx_axis_tvalid	Out	AXI4-Stream Data Valid
rx_axis_tuser	Out	AXI4-Stream User Sideband interface. 1 indicates a bad packet has been received. 0 indicates a good packet has been received.
rx_axis_tlast	Out	AXI4-Stream signal indicating an end of packet.

Data Lane Mapping

For receive data `rx_axis_tdata`, the port is logically divided into lane 0 to lane 3 for the 32-bit interface (See Table 2-8) or lane 0 to lane 7 for the 64-bit interface (see Table 2-9) with the corresponding bit of the `rx_axis_tkeep` word signifying valid data on `rx_axis_tdata`.

Table 2-8: **rx_axis_tdata Lanes - 32 bits**

Lane/rx_axis_tkeep	rx_axis_tdata[31:0] bits
0	7:0
1	15:8
2	23:16
3	31:24

Table 2-9: **rx_axis_tkeep Lanes - 64 bits**

Lane/ rx_axis_tkeep	rx_axis_tdata Bits
0	7:0
1	15:8
2	23:16
3	31:24
4	39:32
5	47:40
6	55:48
7	63:56

Normal Frame Reception

The client must be prepared to accept data at any time; there is no buffering within the core to allow for latency in the receive client. When frame reception begins, data is transferred on consecutive clock cycles to the receive client.

During frame reception, `rx_axis_tvalid` is asserted to indicate that valid frame data is being transferred to the client on `rx_axis_tdata`. All bytes are always valid throughout the frame, as indicated by all `rx_axis_tkeep` bits being set to 1, except during the final transfer of the frame when `rx_axis_tlast` is asserted. During this final transfer of data for a frame, `rx_axis_tkeep` bits indicate the final valid bytes of the frame using the mapping from above. The valid bytes of the final transfer always lead out from `rx_axis_tdata[7:0]` (`rx_axis_tkeep[0]`) because Ethernet frame data is continuous and is received least significant byte first.

The `rx_axis_tlast` is asserted and `rx_axis_tuser` is deasserted, along with the final bytes of the transfer, only after all frame checks are completed. This is after the frame check sequence (FCS) field has been received. The core keeps the `rx_axis_tuser` signal deasserted to indicate that the frame was successfully received and that the frame should be analyzed by the client. This is also the end of packet signaled by `rx_axis_tlast` asserted for one cycle.

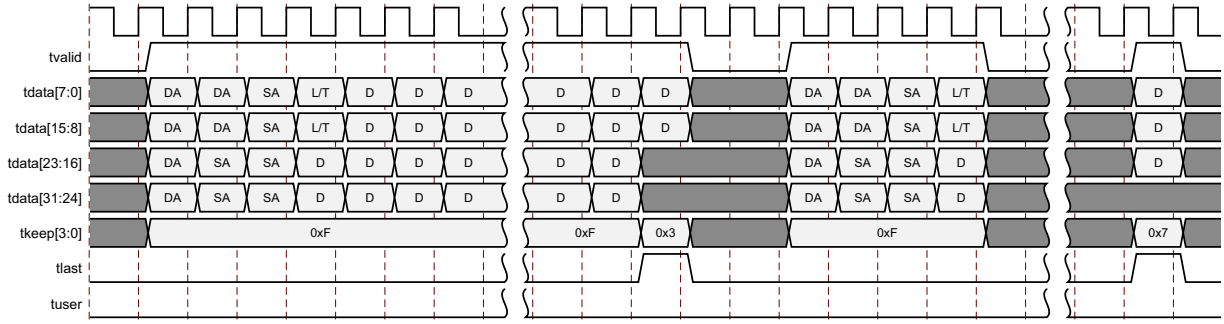

Figure 2-8: **Normal Frame Reception – 32 Bits**

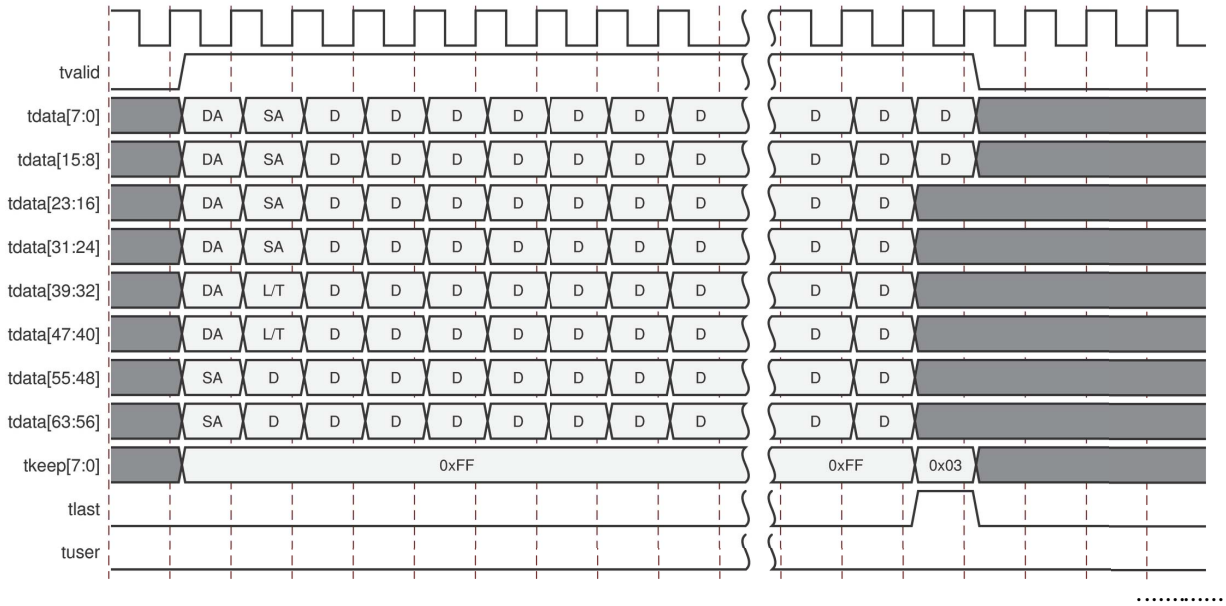

Figure 2-9: **Normal Frame Reception -64 Bits**

Frame Reception with Errors

The case of an unsuccessful frame reception (for example, a runt frame or a frame with an incorrect FCS). In this case, the bad frame is received and the signal `rx_axis_tuser` is asserted to the client at the end of the frame. It is then the responsibility of the client to drop the data already transferred for this frame.

The following conditions cause the assertion of `rx_axis_tlast` along with `rx_axis_tuser` = 1 signifying a bad_frame:

- FCS errors occur.
- Packets are shorter than 64 bytes (undersize or fragment frames).
- Frames of length greater than the maximum transmission unit (MTU) Size programmed are received.
- Any control frame that is received is not exactly the minimum frame length.
- The XGMII data stream contains error codes.

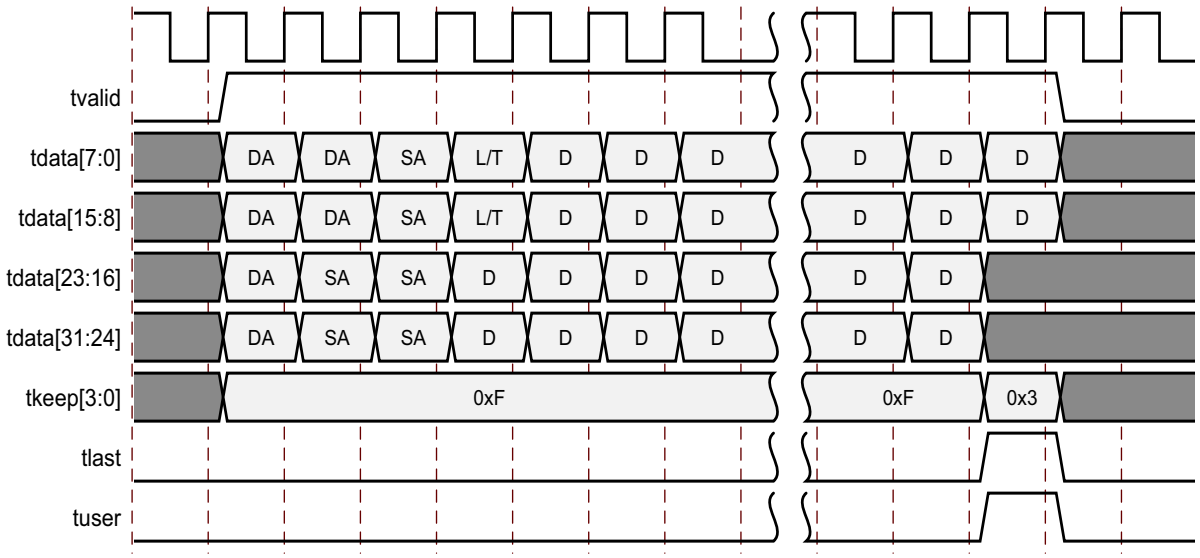

Figure 2-10: **Frame Reception with Errors – 32 Bits**

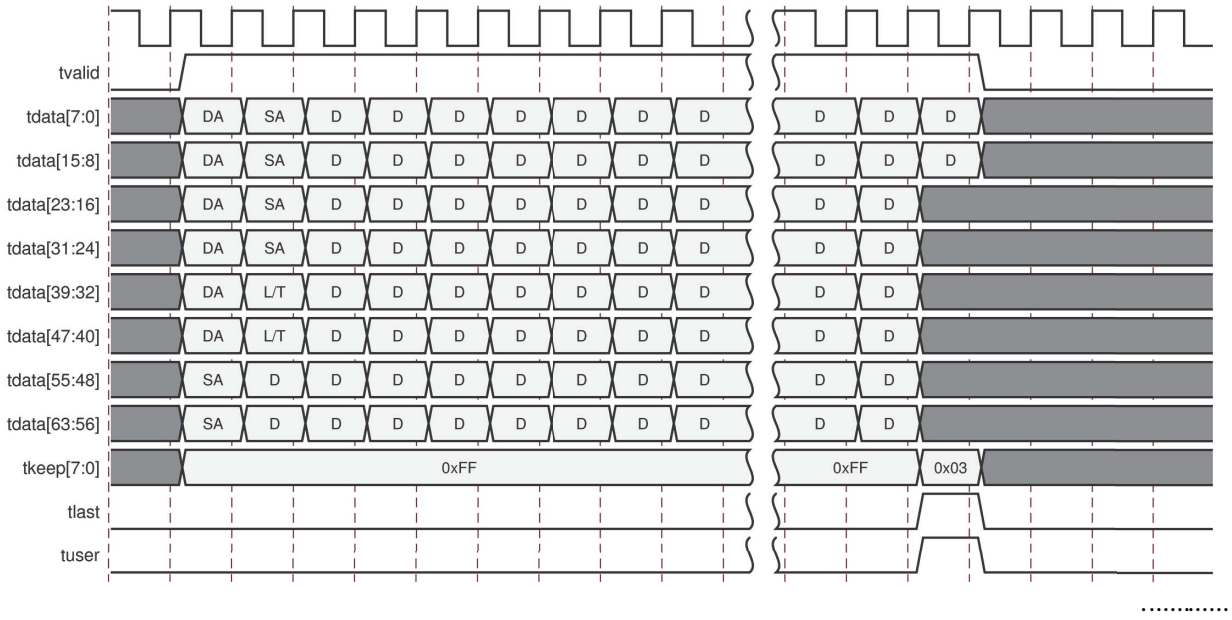

Figure 2-11: **Frame Reception with Errors – 64 Bits**

AXI4-Stream Control and Status Ports

Table 2-10: **AXI4-Stream Interface–TX Path Control/Status Signals**

Name	Direction	Description	Clock Domain
ctl_tx_custom_preamble_enable[(1)]	Input	When asserted, this signals enables the use of tx_preamblein as a custom preamble instead of inserting a standard preamble. ***Note:*** When the core is switched to 1G, this should be always 0.	tx_clk_out
tx_preamblein [55:0][(1)]	Input	This is the custom preamble which is a separate input port rather than being in-line with the data. It should be valid during the start of packet. ***Note:*** When the core is switched to 1G, this should be always 0.	tx_clk_out
ctl_tx_ipg_value[3:0][(1)]	Input	This signal can be optionally present. The ctl_tx_ipg_value defines the target average minimum Inter Packet Gap (IPG, in bytes) inserted between AXI4-Stream packets. Valid values are 8 to 12. The `ctl_tx_ipg_value` can be programmed to a value in the 0 to 7 range, but in that case, it is interpreted as 8 (the minimum valid value). ***Note:*** When the core is switched to 1G, this should be always 12.	tx_clk_out
ctl_tx_enable	Input	TX Enable. This signal is used to enable the transmission of data when it is sampled as a 1. When sampled as a 0, only idles are transmitted by the core. This input should not be set to 1 until the receiver it is sending data to (that is, the receiver in the other device) is fully synchronized and ready to receive data (that is, the other device is not sending a remote fault condition). Otherwise, loss of data can occur. If this signal is set to 0 while a packet is being transmitted, the current packet transmission is completed and then the core stops transmitting any more packets.	tx_clk_out
ctl_tx_send_rfi[(2)]	Input	Transmit Remote Fault Indication (RFI) code word. If this input is sampled as a 1, the TX path only transmits Remote Fault code words. This input should be set to 1 until the RX path is fully synchronized and is ready to accept data from the link partner.	tx_clk_out
ctl_tx_send_lfi[(2)]	Input	Transmit Local Fault Indication (LFI) code word. Takes precedence over Remote Fault Indication (RFI).	tx_clk_out

Table 2-10: **AXI4-Stream Interface–TX Path Control/Status Signals *(Cont'd)***

Name	Direction	Description	Clock Domain
ctl_tx_send_idle	Input	Transmit Idle code words. If this input is sampled as a 1, the TX path only transmits Idle code words. This input should be set to 1 when the partner device is sending RFI code words.	tx_clk_out
ctl_tx_fcs_ins_enable	Input	Enable FCS insertion by the TX core. If this bit is set to 0, the core does not add FCS to packet. If this bit is set to 1, the core calculates and adds the FCS to the packet. This input cannot be changed dynamically between packets.	tx_clk_out
ctl_tx_ignore_fcs	Input	Enable FCS error checking at the AXI4-Stream interface by the TX core. This input only has effect when ctl_tx_fcs_ins_enable is Low. If this input is Low and a packet with bad FCS is being transmitted, it is not binned as good. If this input is High, a packet with bad FCS is binned as good. The error is flagged on the signals stat_tx_bad_fcs and stomped_fcs, and the packet is transmitted as it was received. ***Note:*** Statistics are reported as if there was no FCS error.	tx_clk_out
stat_tx_local_fault(2)	Output	A value of 1 indicates the receive decoder state machine is in the TX_INIT state. This output is level sensitive.	tx_clk_out

Notes:

1. This signal is valid for 64-bit core configurations only.
2. This signal is not valid in 1G mode.

Table 2-11: **AXI4-Stream Interface–RX Path Control/Status Signals**

Name	Direction	Description	Clock Domain
rx_preambleout [55:0][2]	Output	This is the preamble. It is now a separate output instead of in-line with data as was done with previous releases. ***Note:*** When the core is switched to 1G, this should be always 0.	rx_core_clk
ctl_rx_enable	Input	RX Enable. For normal operation, this input must be set to 1. When this input is set the to 0, after the RX completes the reception of the current packet (if any), it stops receiving packets by keeping the PCS from decoding incoming data. In this mode, there are no statistics reported and the AXI4-Stream interface is idle.	rx_clk_out
ctl_rx_check_preamble[2]	Input	When asserted, this input causes the MAC to check the preamble of the received frame.	rx_clk_out
ctl_rx_check_sfd[1]	Input	When asserted, this input causes the MAC to check the Start of Frame Delimiter of the received frame.	rx_clk_out
ctl_rx_force_resync[1]	Input	RX force resynchronization input. This signal is used to force the RX path to reset and re-synchronize. A value of 1 forces the reset operation. A value of 0 allows normal operation. Note that this input should normally be Low and should only be pulsed (1 cycle minimum pulse).	rx_clk_out
ctl_rx_delete_fcs	Input	Enable FCS removal by the RX core. If this bit is set to 0, the core does not remove the FCS of the incoming packet. If this bit is set to 1, the core deletes the FCS to the received packet. Note that FCS is not deleted for packets that are less than or equal to 8 bytes long. This input should only be changed while the corresponding reset input is asserted.	rx_clk_out
ctl_rx_ignore_fcs	Input	Enable FCS error checking at the AXI4-Stream interface by the RX core. If this bit is set to 0, a packet received with an FCS error is sent with the rx_axis_tuser pin asserted during the last transfer (rx_axis_tlast sampled 1). If this bit is set to 1, the core does not flag an FCS error at the AXI4-Stream interface. ***Note:*** The statistics are reported as if the packet is good. The signal stat_rx_bad_fcs, however, reports the error.	rx_clk_out

Table 2-11: **AXI4-Stream Interface–RX Path Control/Status Signals *(Cont'd)***

Name	Direction	Description	Clock Domain
ctl_rx_max_packet_len[14:0]	Input	Any packet longer than this value is considered to be oversized. If a packet has a size greater than this value, the packet is truncated to this value and the rx_axis_tuser signal is asserted along with the rx_axis_tlast signal. Packets less than 4 bytes are dropped. ctl_rx_max_packet_len[14] is reserved and must be set to 0.	rx_clk_out
ctl_rx_min_packet_len[7:0]	Input	Any packet shorter than this value is considered to be undersized. If a packet has a size less than this value, the rx_axis_tuser signal is asserted during the rx_axis_tlast asserted cycle. Packets less than 4 bytes are dropped.	rx_clk_out
stat_rx_framing_err[1:0](1)	Output	The RX sync header bits framing error is a bus that indicates how many sync header errors were received. The value of the bus is only valid when stat_rx_framing_err_valid is a 1. The values can be updated at any time and are intended to be used as increment values for sync header error counters.	rx_clk_out
stat_rx_framing_err_valid(1)	Output	Valid indicator for stat_rx_framing_err. When sampled as a 1, the value on stat_rx_framing_err is valid.	rx_clk_out
stat_rx_local_fault(1)	Output	This output is High when stat_rx_internal_local_fault or stat_rx_received_local_fault is asserted. This output is level sensitive.	rx_clk_out
stat_rx_status	Output	Indicates current status of the link.	rx_clk_out
stat_rx_block_lock(1)	Output	Block lock status. A value of 1 indicates that block lock is achieved as defined in Clause 49.2.14 and MDIO register 3.32.0 This output is level sensitive.	rx_clk_out
stat_rx_remote_fault(1)	Output	Remote fault indication status. If this bit is sampled as a 1, it indicates a remote fault condition was detected. If this bit is sampled as a 0, remote fault condition does not exist. This output is level sensitive.	rx_clk_out
stat_rx_bad_fcs[1:0]	Output	Bad FCS indicator. The value on this bus indicates packets received with a bad FCS, but not a stomped FCS during a cycle. A stomped FCS is defined as the bitwise inverse of the expected good FCS. This output is pulsed for one clock cycle to indicate an error condition. Note that pulses can occur in back to back cycles.	rx_clk_out

Table 2-11: **AXI4-Stream Interface–RX Path Control/Status Signals *(Cont'd)***

Name	Direction	Description	Clock Domain
stat_rx_stomped_fcs[1:0]	Output	Stomped FCS indicator. The value on this bus indicates the packets received with a stomped FCS. A stomped FCS is defined as the bitwise inverse of the expected good FCS. This output is pulsed for one clock cycle to indicate the stomped condition. Note that pulses can occur in back to back cycles.	rx_clk_out
stat_rx_truncated	Output	Packet truncation indicator. A value of 1 indicates that the current packet in flight is truncated due to its length exceeding ctl_rx_max_packet_len[14:0]. This output is pulsed for one clock cycle to indicate the truncated condition. Note that pulses can occur in back to back cycles.	rx_clk_out
stat_rx_internal_local_fault(1)	Output	High when an internal local fault is generated due to any one of the following: test pattern generation or high bit error rate. Note that this signal remains High as long as the fault condition persists.	rx_clk_out
stat_rx_received_local_fault(1)	Output	High when enough local fault words are received from the link partner to trigger a fault condition as specified by the IEEE fault state machine. Remains High as long as the fault condition persists.	rx_clk_out
stat_rx_hi_ber(1)	Output	High Bit Error Rate (BER) indicator. When set to 1, the BER is too high as defined by IEEE Std. 802.3. Corresponds to MDIO register bit 3.32.1 as defined in Clause 49.2.14. This output is level sensitive.	rx_clk_out
ctl_rx_custom_preamble_enable(2)	Input	When asserted, this signal causes the side band of a packet presented on the AXI4-Stream to be the preamble as it appears on the line. ***Note:*** When the core is switched to 1G, this should be always 0.	rx_clk_out

Notes:

1. This signal is not valid in 1G mode.
2. This signal is valid for 64-bit core configurations only.

Miscellaneous Status/Control Signals

Table 2-12 shows the miscellaneous status and control I/O signals.

Table 2-12: **Miscellaneous Status/Control Ports**

Name	Direction	Description	Clock Domain
dclk	Input	Dynamic Reconfiguration Port (DRP) clock input. The required frequency is set by providing the value in the **GT DRP Clock** field in the Vivado IDE GT Selection and Configuration tab. This must be a free running input clock.	Refer to Clocking.
stat_rx_valid_ctrl_code(1)	Output	Indicates that a PCS block with a valid control code was received.	rx_clk_out
ctl_local_loopback	Input	Loopback enable. A value of 1 enables loopback as defined in Clause 49. Corresponds to management data input/output (MDIO) register bit 3.0.14 as defined in Clause 45. This input should only be changed while the corresponding reset input is asserted.	Asynch
stat_rx_got_signal_os(1)	Output	Signal OS indication. If this bit is sampled as a 1, it indicates that a Signal OS word was received. Note that Signal OS should not be received in an Ethernet network.	rx_clk_out
ctl_rx_process_lfi(1)	Input	When this input is set to 1, the RX core expects and processes LF control codes coming in from the transceiver. When set to 0, the RX core ignores LF control codes coming in from the transceiver.	rx_clk_out
ctl_rx_test_pattern(1)	Input	Test pattern checking enable for the RX core. A value of 1 enables test mode as defined in Clause 49. Corresponds to MDIO register bit 3.42.2 as defined in Clause 45. Checks for scrambled idle pattern.	rx_clk_out
ctl_tx_test_pattern(1)	Input	Test pattern generation enable for the TX core. A value of 1 enables test mode as defined in Clause 49. Corresponds to MDIO register bit 3.42.3 as defined in Clause 45. Generates a scrambled idle pattern.	tx_clk_out
stat_rx_test_pattern_mismatch(1)	Output	Test pattern mismatch increment. A non zero value in any cycle indicates how many mismatches occurred for the test pattern in the RX core. This output is only active when ctl_rx_test_pattern is set to a 1. This output can be used to generate MDIO register as defined in Clause 45. This output is pulsed for one clock cycle.	rx_clk_out

Table 2-12: **Miscellaneous Status/Control Ports *(Cont'd)***

Name	Direction	Description	Clock Domain
ctl_rx_data_pattern_select[(1)]	Input	Corresponds to MDIO register bit 3.42.0 as defined in Clause 45.	rx_clk_out
ctl_rx_test_pattern_enable[(1)]	Input	Test pattern enable for the RX core. A value of 1 enables test mode. Corresponds to MDIO register bit 3.42.2 as defined in Clause 45. Takes second precedence.	rx_clk_out
ctl_tx_data_pattern_select[(1)]	Input	Corresponds to MDIO register bit 3.42.0 as defined in Clause 45.	tx_clk_out
ctl_tx_test_pattern_enable[(1)]	Input	Test pattern generation enable for the TX core. A value of 1 enables test mode. Corresponds to MDIO register bit 3.42.3 as defined in Clause 45. Takes second precedence.	tx_clk_out
ctl_tx_test_pattern_seed_a[57:0][(1)]	Input	Corresponds to MDIO registers 3.34 through to 3.37 as defined in Clause 45.	tx_clk_out
ctl_tx_test_pattern_seed_b[57:0][(1)]	Input	Corresponds to MDIO registers 3.38 through to 3.41 as defined in Clause 45.	tx_clk_out
ctl_tx_test_pattern_select[(1)]	Input	Corresponds to MDIO register bit 3.42.1 as defined in Clause 45.	tx_clk_out
gig_ethernet_pcs_pma_status_vector_0 [15:0][(1)]	Output	See Status Vector Table in *1G/2.5G Ethernet PCS/PMA or SGMII LogiCORE IP Product Guide* (PG047)[Ref 3]	

Notes:
1. This signal is not valid in 1G mode.

Statistics Interface Ports

Table 2-13 and Table 2-14 show the Statistics interface I/O ports.

Table 2-13: **Statistics Interface - RX Path**

Name	Direction	Description	Clock Domain
stat_rx_total_bytes[3:0]	Output	Increment for the total number of bytes received.	rx_clk_out
stat_rx_total_packets[1:0]	Output	Increment for the total number of packets received.	rx_clk_out
stat_rx_total_good_bytes[13:0]	Output	Increment for the total number of good bytes received. This value is only non-zero when a packet is received completely and contains no errors.	rx_clk_out

Table 2-13: **Statistics Interface - RX Path *(Cont'd)***

Name	Direction	Description	Clock Domain
stat_rx_total_good_packets	Output	Increment for the total number of good packets received. This value is only non-zero when a packet is received completely and contains no errors.	rx_clk_out
stat_rx_packet_bad_fcs	Output	Increment for packets between 64 and ctl_rx_max_packet_len bytes that have Frame Check Sequence (FCS) errors.	rx_clk_out
stat_rx_packet_64_bytes	Output	Increment for good and bad packets received that contain 64 bytes.	rx_clk_out
stat_rx_packet_65_127_bytes	Output	Increment for good and bad packets received that contain 65 to 127 bytes.	rx_clk_out
stat_rx_packet_128_255_bytes	Output	Increment for good and bad packets received that contain 128 to 255 bytes.	rx_clk_out
stat_rx_packet_256_511_bytes	Output	Increment for good and bad packets received that contain 256 to 511 bytes.	rx_clk_out
stat_rx_packet_512_1023_bytes	Output	Increment for good and bad packets received that contain 512 to 1,023 bytes.	rx_clk_out
stat_rx_packet_1024_1518_bytes	Output	Increment for good and bad packets received that contain 1,024 to 1,518 bytes.	rx_clk_out
stat_rx_packet_1519_1522_bytes	Output	Increment for good and bad packets received that contain 1519 to 1522 bytes.	rx_clk_out
stat_rx_packet_1523_1548_bytes	Output	Increment for good and bad packets received that contain 1,523 to 1,548 bytes.	rx_clk_out
stat_rx_packet_1549_2047_bytes	Output	Increment for good and bad packets received that contain 1,549 to 2,047 bytes.	rx_clk_out
stat_rx_packet_2048_4095_bytes	Output	Increment for good and bad packets received that contain 2,048 to 4,095 bytes.	rx_clk_out
stat_rx_packet_4096_8191_bytes	Output	Increment for good and bad packets received that contain 4,096 to 8,191 bytes.	rx_clk_out
stat_rx_packet_8192_9215_bytes	Output	Increment for good and bad packets received that contain 8,192 to 9,215 bytes.	rx_clk_out
stat_rx_packet_small	Output	Increment for all packets that are less than 64 bytes long. Packets that are less than 4 bytes are dropped.	rx_clk_out
stat_rx_packet_large	Output	Increment for all packets that are more than 9,215 bytes long.	rx_clk_out
stat_rx_oversize	Output	Increment for packets longer than ctl_rx_max_packet_len with good FCS.	rx_clk_out
stat_rx_toolong	Output	Increment for packets longer than ctl_rx_max_packet_len with good and bad FCS.	rx_clk_out

Table 2-13: **Statistics Interface - RX Path *(Cont'd)***

Name	Direction	Description	Clock Domain
stat_rx_undersize	Output	Increment for packets shorter than ctl_rx_min_packet_len with good FCS.	rx_clk_out
stat_rx_fragment	Output	Increment for packets shorter than ctl_rx_min_packet_len with bad FCS.	rx_clk_out
stat_rx_jabber	Output	Increment for packets longer than ctl_rx_max_packet_len with bad FCS.	rx_clk_out
stat_rx_bad_code(1)	Output	Increment for 64B/66B code violations. This signal indicates that the RX PCS receive state machine is in the RX_E state as specified by IEEE Std. 802.3. This output can be used to generate MDIO register as defined in Clause 45.	rx_clk_out
stat_rx_bad_sfd(1)	Output	Increment bad SFD. This signal indicates if the Ethernet packet received was preceded by a valid SFD. A value of 1 indicates that an invalid SFD was received.	rx_clk_out
stat_rx_bad_preamble(1)	Output	Increment bad preamble. This signal indicates if the Ethernet packet received was preceded by a valid preamble. A value of 1 indicates that an invalid preamble was received.	rx_clk_out

Notes:
1. This signal is not valid in 1G mode.

Table 2-14: **Statistics Interface - TX Path**

Name	Direction	Description	Clock Domain
stat_tx_total_bytes[3:0]	Output	Increment for the total number of bytes transmitted. The signal width for stat_tx_total_bytes will be [2:0] when the 32-bit AXI4-Stream option is selected.	tx_clk_out
stat_tx_total_packets	Output	Increment for the total number of packets transmitted.	tx_clk_out
stat_tx_total_good_bytes[13:0]	Output	Increment for the total number of good bytes transmitted. This value is only non-zero when a packet is transmitted completely and contains no errors.	tx_clk_out
stat_tx_total_good_packets	Output	Increment for the total number of good packets transmitted.	tx_clk_out
stat_tx_bad_fcs	Output	Increment for packets greater than 64 bytes that have FCS errors.	tx_clk_out

Table 2-14: **Statistics Interface - TX Path *(Cont'd)***

Name	Direction	Description	Clock Domain
stat_tx_packet_64_bytes	Output	Increment for good and bad packets transmitted that contain 64 bytes.	tx_clk_out
stat_tx_packet_65_127_bytes	Output	Increment for good and bad packets transmitted that contain 65 to 127 bytes.	tx_clk_out
stat_tx_packet_128_255_bytes	Output	Increment for good and bad packets transmitted that contain 128 to 255 bytes.	tx_clk_out
stat_tx_packet_256_511_bytes	Output	Increment for good and bad packets transmitted that contain 256 to 511 bytes.	tx_clk_out
stat_tx_packet_512_1023_bytes	Output	Increment for good and bad packets transmitted that contain 512 to 1,023 bytes.	tx_clk_out
stat_tx_packet_1024_1518_bytes	Output	Increment for good and bad packets transmitted that contain 1,024 to 1,518 bytes.	tx_clk_out
stat_tx_packet_1519_1522_bytes	Output	Increment for good and bad packets transmitted that contain 1,519 to 1,522 bytes.	tx_clk_out
stat_tx_packet_1523_1548_bytes	Output	Increment for good and bad packets transmitted that contain 1,523 to 1,548 bytes.	tx_clk_out
stat_tx_packet_1549_2047_bytes	Output	Increment for good and bad packets transmitted that contain 1,549 to 2,047 bytes.	tx_clk_out
stat_tx_packet_2048_4095_bytes	Output	Increment for good and bad packets transmitted that contain 2,048 to 4,095 bytes.	tx_clk_out
stat_tx_packet_4096_8191_bytes	Output	Increment for good and bad packets transmitted that contain 4,096 to 8,191 bytes.	tx_clk_out
stat_tx_packet_8192_9215_bytes	Output	Increment for good and bad packets transmitted that contain 8,192 to 9,215 bytes.	tx_clk_out
stat_tx_packet_small	Output	Increment for all packets that are less than 64 bytes long.	tx_clk_out
stat_tx_packet_large	Output	Increment for all packets that are more than 9,215 bytes long.	tx_clk_out
stat_tx_frame_error	Output	Increment for packets with tx_axis_tuser set to indicate an End of Packet (EOP) abort.	tx_clk_out

XGMII/GMII Interface Ports

Table 2-15 shows the XGMII/GMII Interface ports.

Table 2-15: **XGMII/GMII Interface Ports**

Name	Direction	Description	Clock Domain
rx_mii_d[31:0]	Output	Receive XGMII Data bus.	rx_mii_clk
rx_mii_c[3:0]	Output	Receive XGMII Control bus.	rx_mii_clk
rx_mii_clk	Input	Receive XGMII Clock input.	See Clocking for more information.
tx_mii_d[31:0]	Input	Transmit XGMII Data bus.	rx_mii_clk
tx_mii_c[3:0]	Input	Transmit XGMII Control bus.	rx_mii_clk
tx_mii_clk	Input	Transmit XGMII Clock input.	See Clocking for more information.
gmii_rxd[7:0]	Output	Receive GMII Data bus.	rx_core_clk
gmii_rx_dv	Output	Receive GMII Control signal.	rx_core_clk
gmii_rx_er	Output	Receive GMII error signal.	rx_core_clk
gmii_txd[7:0]	Output	Transmit GMII Data bus.	tx_out_clk
gmii_tx_en	Output	Transmit GMII enable signal.	tx_out_clk
gmii_rx_er	Output	Transmit GMII error signal.	tx_out_clk

Register Space

The 1G/10G/25G Switching Ethernet Subsystem is configured with AXI4-Lite registers to access the configuration and status signals. For more information, see *1G/2.5G Ethernet PCS/PMA or SGMII LogiCORE IP Product Guide* (PG047)[Ref 3]and *10G/25G High Speed Ethernet Subsystem Product Guide* (PG210)[Ref 4]. AXI Crossbar module is initiated within the IP to access the AXI4-Lite interface control and statistics registers for Gigabit Ethernet PCS-PMA and 10G MAC+PCS subsystem. This module is provided with a single user AXI-Lite interface.

The AXI Crossbar IP is configured with 1-master and 2-slave interfaces. Both Gigabit Ethernet PCS-PMA and 10G/25G MAC+PCS subsystem control and status registers can be accessed with AXI4-Lite interface through AXI Crossbar. Refer to the AXI Interconnect v2.1 LogiCORE IP Product Guide (PG059) for AXI Crossbar soft IP functionality.

The following are the configured base address locations for Gigabit Ethernet PCS-PMA and 10G/25G MAC+PCS subsystem control and status registers in the AXI Crossbar soft IP:

- **0x0000_0000 to 0x0000_0FFF**: Address locations for 10G/25G MAC+PCS subsystem
- **0x0000_1000 to 0x0000_1FFF**: Address locations for Gigabit Ethernet PCS-PMA

AXI4-Lite Ports

Table 2-16 describes the port list for the AXI processor interface.

Table 2-16: **AXI Ports**

Signal	Direction	Description
s_axi_aclk	In	AXI4-Lite clock. Range between 10 MHz and 300 MHz
s_axi_aresetn	In	Asynchronous active-Low reset
s_axi_awaddr[31:0]	In	Write address bus
s_axi_awvalid	In	Write address valid
s_axi_awready	Out	Write address acknowledge
s_axi_wdata[31:0]	In	Write data bus
s_axi_wstrb[3:0]	In	Strobe signal for the data bus byte lane
s_axi_wvalid	Out	Write data valid
s_axi_wready	Out	Write data acknowledge
s_axi_bresp[1:0]	Out	Write transaction response
s_axi_bvalid	Out	Write response valid
s_axi_bready	In	Write response acknowledge
s_axi_araddr[31:0]	In	Read address bus
s_axi_arvalid	In	Read address valid
s_axi_arready	Out	Read address acknowledge
s_axi_rdata[31:0]	Out	Read data output
s_axi_rresp[1:0]	Out	Read data response
s_axi_rvalid	Out	Read data/response valid
s_axi_rready	In	Read data acknowledge
pm_tick	In	Top level signal to read statistics counters; requires MODE_REG[30] (`tick_reg_mode_sel`) be set to 0.

Additional information for the operation of the AXI4 bus is found in *ARM AMBA AXI Protocol v2.0 Specification (ARM IHI 0022C)*[Ref 15].

As noted previously, the top level signal `pm_tick` can be used to read statistics counters instead of the configuration register TICK_REG. In this case, configuration register MODE_REG bit 30 (`tick_reg_mode_sel`) should be set to 0. If `tick_reg_mode_sel` is set to 1, `tick_reg` is used to read the statistics counters.

Configuration and Status Register Map

Configuration Register Map for 1G/10G/25G Ethernet Subsystem

The configuration space provides software with the ability to configure the IP core for various use cases. Certain features are optional and the assigned register might not exist in a particular variant, in which case the applicable registers are considered RESERVED.

In order for the programmed configurations to take effect, it is necessary to issue `s_axi_aresetn`, which is active-Low.

Table 2-17: **Configuration Register Map**

Hex Address	Register Name/Link to Description	Notes
0x0000	GT_RESET_REG: 0000	
0x0004	RESET_REG: 0004	
0x0008	MODE_REG: 0008	
0x000C	CONFIGURATION_TX_REG1: 000C	
0x0014	CONFIGURATION_RX_REG1: 0014	
0x0018	CONFIGURATION_RX_MTU: 0018	Only in MAC+PCS variant and MAC-only variants
0x0020	TICK_REG: 0020	
0x0024	CONFIGURATION_REVISION_REG: 0024	
0x0028	CONFIGURATION_TX_TEST_PAT_SEED_A_LSB: 0028	Only in MAC+PCS and PCS-only variants
0x002C	CONFIGURATION_TX_TEST_PAT_SEED_A_MSB: 002C	Only in MAC+PCS and PCS-only variants
0x0030	CONFIGURATION_TX_TEST_PAT_SEED_B_LSB: 0030	Only in MAC+PCS and PCS-only variants
0x0034	CONFIGURATION_TX_TEST_PAT_SEED_B_MSB: 0034	Only in MAC+PCS and PCS-only variants
0x0190	CONFIGURATION_1588_REG	Only in MAC+PCS and PCS-only variants
0x0194	TX_CONFIGURATION_1588_REG	Only in MAC+PCS and PCS-only variants
0x0198	RX_CONFIGURATION_1588_REG	Only in MAC+PCS and PCS-only variants

Status Register Map for 1G/10G/25G Ethernet Subsystem

The status registers provide an indication of the health of the system. These registers are Read-Only and a read operation clears the register.

Status registers are cleared according to the following conditions.

- Applying `s_axi_aresetn` clears both TX and RX status registers
- When a particular status register is read (clear on read)
- Applying `rx_reset` clears the RX status registers only
- Applying `tx_reset` clears the TX status registers only

Table 2-18: **Status Register Map**

Hex Address	Register Name/Link to Description	Notes
0x0180	STAT_CORE_SPEED_REG	
0x0400	STAT_TX_STATUS_REG1: 0400	
0x0404	STAT_RX_STATUS_REG1: 0404	
0x0408	STAT_STATUS_REG1: 0408	Only in MAC+PCS and PCS-only variants
0x040C	STAT_RX_BLOCK_LOCK_REG: 040C	Only in MAC+PCS and PCS-only variants
0x0494	STAT_RX_VALID_CTRL_CODE: 0494	Only in MAC+PCS and PCS-only variants

Configuration and Status Register Map for 1G Ethernet PCS/PMA

Table 2-19: **Registers for 1G Ethernet PCS/PMA**

Register Address	Register Name
0x0000	Register 0: Control Register(1)
0x0004	Register 1: Status Register
0x0008	Register 2: PHY Identifier
0x000C	Register 3: PHY Identifier
0x003C	Register 15: Extended Status
0x004C	1588 Control: Vendor Specific Register 19
0x0050	RX PHY Fixed Latency: Vendor Specific Register 20
0x0054	RX PHY Variable Latency: Vendor Specific Register 21

Notes:
1. Power down and Loopback feature is not supported.

Register Descriptions

This section contains descriptions of the configuration registers. In the cases where the features described in the bit fields are not present in the IP core, the bit field is assumed to be RESERVED. The below descriptions covers only the registers that have been modified with respect to default operation of the respective IPs in applicable mode. For further information on register interface definitions (not defined in this guide), see *10G/25G High Speed Ethernet Subsystem Product Guide* (PG210)[Ref 4]or *1G/2.5G Ethernet PCS/PMA or SGMII LogiCORE IP Product Guide* (PG047)[Ref 3].

Configuration Registers for 1G/10G/25G Subsystem

See *10G/25G High Speed Ethernet Subsystem Product Guide* (PG210) [Ref 3] for information on Configuration Registers for 1G/10G/25G Subsystem.

CONFIGURATION_TX_REG1: 000C

Table 2-20: **CONFIGURATION_TX_REG1: 000C**

Bits	Default	Type	Signal
0	1	RW	ctl_tx_enable(1)
1	1	RW	ctl_tx_fcs_ins_enable(1)
2	0	RW	ctl_tx_ignore_fcs(1)
3	0	RW	ctl_tx_send_lfi(1)
4	0	RW	ctl_tx_send_rfi(1)
5	0	RW	ctl_tx_send_idle(1)
7:6	2	RW	ctl_core_speed_sel(3)
13:10	12	RW	ctl_tx_ipg_value(1)
14	0	RW	ctl_tx_test_pattern
15	0	RW	ctl_tx_test_pattern_enable
16	0	RW	ctl_tx_test_pattern_select
17	0	RW	ctl_tx_data_pattern_select
18	0	RW	ctl_tx_custom_preamble_enable(1)
23	0	RW	ctl_tx_prbs31_test_pattern_enable(2)

Notes:

1. Only in MAC+PCS variant
2. Only in PCS variant
3. 2'b10: This is the default configuration. You have to write this to configure core in 10G mode

   2'b01: You have to write this to configure core in 1G mode

   others: Reserved

STAT_CORE_SPEED_REG: 0180

Table 2-21: **STAT_CORE_SPEED_REG: 0180**

Bits	Default	Type	Signal
1:0	2	RO	stat_core_speed(1)

Notes:
1. Each mode indicate different configurations.
- 2'b10: Indicates that the core is configured in 10G mode
- 2'b01: Indicates that the core is configured in 1G mode
- others: Reserved

CONFIGURATION_1588_REG: 0x0190

Table 2-22: **CONFIGURATION_1588_REG: 0x0190**

Bits	Default	Type	Signal
0	1	RW	ctl_tx_lat_adj_enb
1	1	RW	ctl_rx_lat_adj_enb ***Note:*** For 1G mode, this register is **Reserved.**
2	0	RW	ctl_ptp_transpclk_mode
3	0	RW	ctl_tx_timestamp_adj_enb
4	1	RW	ctl_rx_timestamp_adj_enb ***Note:*** For 1G mode, this register is **Reserved.**

CONFIGURATION_1588_REG: 0x0194

Table 2-23: **CONFIGURATION_1588_REG: 0x0194**

Bits	Default	Type	Signal
31:0	0	RW	ctl_tx_latency

CONFIGURATION_1588_REG: 0x0198

Table 2-24: **CONFIGURATION_1588_REG: 0x0198**

Bits	Default	Type	Signal
31:0	0	RW	ctl_rx_latency ***Note:*** For 1G mode, this register is **Reserved**.

Status Registers for 1G/10G/25G Subsystem

See *1G/2.5G Ethernet PCS/PMA or SGMII LogiCORE IP Product Guide* (PG047)[Ref 4] for information on Status Registers for 1G/10G/25G Subsystem.

Configuration and Status Registers for 1G/2.5G Ethernet PCS-PMA

AXI-Lite support has been added in 1G Ethernet PCS-PMS IP to enable you in programming the control and status registers. The addresses of the registers are word aligned for axi-lite accesses. Strobing for data while accessing these registers is disabled.

Any read/write operations to given addresses will lead to read/write operations to corresponding 16 bit register as defined in *1G/2.5G Ethernet PCS/PMA or SGMII LogiCORE IP Product Guide* (PG047)[Ref 4].

The following vendor-specific registers have be added to the MDIO PCS Address space when configured for 1000BASE-X operation. See the *1G/2.5G Ethernet PCS/PMA or SGMII LogiCORE IP Product Guide* (PG047) [Ref 4]

For registers 0x0000-0x003C, see1000BASE-X or 2500BASE-X Standard Without Optional Auto-Negotiation Table in *1G/2.5G Ethernet PCS/PMA or SGMII LogiCORE IP Product Guide* (PG047)[Ref 4].

Table 2-25: **1588 Control: Vendor Specific Register 19**

Bits	Default Value	Access	Description
15:4	N/A	RO	Reserved
3	1	RW	Timestamp correction enable. When 1, the RX timestamp is adjusted to compensate for enabled PHY fixed and variable latencies. When 0, no adjustment is made to the timestamp.
2	1	RW	Fixed RX PHY latency correction enable. When 1, the RX timestamp is adjusted to compensate for fixed PHY latency by using the correction value specified in Table 2-24. When 0, no adjustment is made to compensate for fixed known latencies.
1	0	RO	Reserved
0	1	RW	Variable RX transceiver latency correction enable. When 1, the RX timestamp is adjusted to compensate for measurable variable transceiver latency (for 1000BASE-X this is the barrel shift position of the serial-to-parallel converter in the GTX transceiver PMA). This only varies when the subsystem is initialized following a power-on, reset, or recovery from loss of synchronization; it then remains constant for normal operation. When 0, no adjustment is made to compensate for measurable variable known latencies.

Table 2-26: **RX PHY Fixed Latency: Vendor Specific Register 20**

Bits	Default Value	Access	Description
15:0	0xC8	RW	RX 1000BASE-X Fixed Delay in ns. This value is initialized to the known RX latency from the serial wire input into the FPGA, through the transceiver fixed latency components prior to the timestamping position.

Table 2-27: **RX PHY Variable Latency: Vendor Specific Register 21**

Bits	Default Value	Access	Description
15:0	N/A	RO	RX 1000BASE-X variable RX Delay in UI. This value is measured within the subsystem following RX synchronization (for 1000BASE-X this is the barrel shift position of the serial-to-parallel converted in the transceiver PMA). This only varies when the subsystem is initialized following a power-on, reset, or recovery from loss of synchronization; it then remains constant for normal operation.

See *1G/2.5G Ethernet PCS/PMA or SGMII LogiCORE IP Product Guide* (PG047)[Ref 4] for information on Configuration Registers for 1G/2.5G Ethernet PCS-PMA.

Statistics Counters

The statistics counters provide histograms of the classification of traffic and error counts. These counters can be read either by a 1 on `pm_tick` or by writing a 1 to `tick_reg`, depending on the value of MODE_REG[30] (`tick_reg_mode_sel`). `pm_tick` will be used when MODE_REG[30] = 0 and `tick_reg` will be used when MODE_REG[30] = 1 (1 = default).

The counters employ an internal accumulator. A write to the `tick_reg` register causes the accumulated counts to be pushed to the readable STAT_*_MSB/LSB registers and simultaneously clear the accumulators. The STAT_*_MSB/LSB registers can then be read. In this way all values stored in the statistics counters represent a snap-shot over the same time interval.

The STAT_CYCLE_COUNT_MSB/LSB register contains a count of the number of RX core clock cycles between `tick_reg` writes. This allows for easy time-interval based statistics.

Statistic counter registers are cleared according to the following conditions

- Applying `s_axi_aresetn` clears both TX and RX statistics counter registers
- Applying PM Tick clears both TX and RX statistics counter registers
- Applying `rx_reset` clears the RX statistics counter registers only
- Applying `tx_reset` clears the TX statistics counter registers only

Implementation of statistics counters for 1G and 10G modes of operation is common to both 10G/25G Ethernet subsystem and 1G/2.5G Ethernet PCS/PMA. The current values of statistics shows the value pertaining to the current mode of operation. Whenever there is a change in line rate, the system resets statistics counters.

Table 2-28: **Statistics Counters**

Hex Address	Register Name/Link to Description	Notes
0x0500	STATUS_CYCLE_COUNT_LSB: 0500	
0x0504	STATUS_CYCLE_COUNT_MSB: 0504	
0x0648	STAT_RX_FRAMING_ERR_LSB: 0648	Only in MAC+PCS and PCS-only variants
0x064C	STAT_RX_FRAMING_ERR_MSB: 064C	Only in MAC+PCS and PCS-only variants
0x0660	STAT_RX_BAD_CODE_LSB: 0660	
0x0664	STAT_RX_BAD_CODE_MSB: 0664	
0x06A0	STAT_TX_FRAME_ERROR_LSB: 06A0	Only in MAC+ PCS and MAC-only variants
0x06A4	STAT_TX_FRAME_ERROR_MSB: 06A4	Only in MAC+ PCS and MAC-only variants
0x0700	STAT_TX_TOTAL_PACKETS_LSB: 0700	Only in MAC+ PCS and MAC-only variants
0x0704	STAT_TX_TOTAL_PACKETS_MSB: 0704	Only in MAC+ PCS and MAC-only variants
0x0708	STAT_TX_TOTAL_GOOD_PACKETS_LSB: 0708	Only in MAC+ PCS and MAC-only variants
0x070C	STAT_TX_TOTAL_GOOD_PACKETS_MSB: 070C	Only in MAC+ PCS and MAC-only variants
0x0710	STAT_TX_TOTAL_BYTES_LSB: 0710	Only in MAC+ PCS and MAC-only variants
0x0714	STAT_TX_TOTAL_BYTES_MSB: 0714	Only in MAC+ PCS and MAC-only variants
0x0718	STAT_TX_TOTAL_GOOD_BYTES_LSB: 0718	Only in MAC+ PCS and MAC-only variants
0x071C	STAT_TX_TOTAL_GOOD_BYTES_MSB: 071C	Only in MAC+ PCS and MAC-only variants
0x0720	STAT_TX_PACKET_64_BYTES_LSB: 0720	Only in MAC+ PCS and MAC-only variants
0x0724	STAT_TX_PACKET_64_BYTES_MSB: 0724	Only in MAC+ PCS and MAC-only variants
0x0728	STAT_TX_PACKET_65_127_BYTES_LSB: 0728	Only in MAC+ PCS and MAC-only variants
0x072C	STAT_TX_PACKET_65_127_BYTES_MSB: 072C	Only in MAC+ PCS and MAC-only variants

Table 2-28: **Statistics Counters *(Cont'd)***

Hex Address	Register Name/Link to Description	Notes
0x0730	STAT_TX_PACKET_128_255_BYTES_LSB: 0730	Only in MAC+ PCS and MAC-only variants
0x0734	STAT_TX_PACKET_128_255_BYTES_MSB: 0734	Only in MAC+ PCS and MAC-only variants
0x0738	STAT_TX_PACKET_256_511_BYTES_LSB: 0738	Only in MAC+ PCS and MAC-only variants
0x073C	STAT_TX_PACKET_256_511_BYTES_MSB: 073C	Only in MAC+ PCS and MAC-only variants
0x0740	STAT_TX_PACKET_512_1023_BYTES_LSB: 0740	Only in MAC+ PCS and MAC-only variants
0x0744	STAT_TX_PACKET_512_1023_BYTES_MSB: 0744	Only in MAC+ PCS and MAC-only variants
0x0748	STAT_TX_PACKET_1024_1518_BYTES_LSB: 0748	Only in MAC+ PCS and MAC-only variants
0x074C	STAT_TX_PACKET_1024_1518_BYTES_MSB: 074C	Only in MAC+ PCS and MAC-only variants
0x0750	STAT_TX_PACKET_1519_1522_BYTES_LSB: 0750	Only in MAC+ PCS and MAC-only variants
0x0754	STAT_TX_PACKET_1519_1522_BYTES_MSB: 0754	Only in MAC+ PCS and MAC-only variants
0x0758	STAT_TX_PACKET_1523_1548_BYTES_LSB: 0758	Only in MAC+ PCS and MAC-only variants
0x075C	STAT_TX_PACKET_1523_1548_BYTES_MSB: 075C	Only in MAC+ PCS and MAC-only variants
0x0760	STAT_TX_PACKET_1549_2047_BYTES_LSB: 0760	Only in MAC+ PCS and MAC-only variants
0x0764	STAT_TX_PACKET_1549_2047_BYTES_MSB: 0764	Only in MAC+ PCS and MAC-only variants
0x0768	STAT_TX_PACKET_2048_4095_BYTES_LSB: 0768	Only in MAC+ PCS and MAC-only variants
0x076C	STAT_TX_PACKET_2048_4095_BYTES_MSB: 076C	Only in MAC+ PCS and MAC-only variants
0x0770	STAT_TX_PACKET_4096_8191_BYTES_LSB: 0770	Only in MAC+ PCS and MAC-only variants
0x0774	STAT_TX_PACKET_4096_8191_BYTES_MSB: 0774	Only in MAC+ PCS and MAC-only variants
0x0778	STAT_TX_PACKET_8192_9215_BYTES_LSB: 0778	Only in MAC+ PCS and MAC-only variants
0x077C	STAT_TX_PACKET_8192_9215_BYTES_MSB: 077C	Only in MAC+ PCS and MAC-only variants
0x0780	STAT_TX_PACKET_LARGE_LSB: 0780	Only in MAC+ PCS and MAC-only variants

Table 2-28: **Statistics Counters *(Cont'd)***

Hex Address	Register Name/Link to Description	Notes
0x0784	STAT_TX_PACKET_LARGE_MSB: 0784	Only in MAC+ PCS and MAC-only variants
0x0788	STAT_TX_PACKET_SMALL_LSB: 0788	Only in MAC+ PCS and MAC-only variants
0x078C	STAT_TX_PACKET_SMALL_MSB: 078C	Only in MAC+ PCS and MAC-only variants
0x07B8	STAT_TX_BAD_FCS_LSB: 07B8	Only in MAC+ PCS and MAC-only variants
0x07BC	STAT_TX_BAD_FCS_MSB: 07BC	Only in MAC+ PCS and MAC-only variants
0x0808	STAT_RX_TOTAL_PACKETS_LSB: 0808	Only in MAC+ PCS and MAC-only variants
0x080C	STAT_RX_TOTAL_PACKETS_MSB: 080C	Only in MAC+ PCS and MAC-only variants
0x0810	STAT_RX_TOTAL_GOOD_PACKETS_LSB: 0810	Only in MAC+ PCS and MAC-only variants
0x0814	STAT_RX_TOTAL_GOOD_PACKETS_MSB: 0814	Only in MAC+ PCS and MAC-only variants
0x0818	STAT_RX_TOTAL_BYTES_LSB: 0818	Only in MAC+ PCS and MAC-only variants
0x081C	STAT_RX_TOTAL_BYTES_MSB: 081C	Only in MAC+ PCS and MAC-only variants
0x0820	STAT_RX_TOTAL_GOOD_BYTES_LSB: 0820	Only in MAC+ PCS and MAC-only variants
0x0824	STAT_RX_TOTAL_GOOD_BYTES_MSB: 0824	Only in MAC+ PCS and MAC-only variants
0x0828	STAT_RX_PACKET_64_BYTES_LSB: 0828	Only in MAC+ PCS and MAC-only variants
0x082C	STAT_RX_PACKET_64_BYTES_MSB: 082C	Only in MAC+ PCS and MAC-only variants
0x0830	STAT_RX_PACKET_65_127_BYTES_LSB: 0830	Only in MAC+ PCS and MAC-only variants
0x0834	STAT_RX_PACKET_65_127_BYTES_MSB: 0834	Only in MAC+ PCS and MAC-only variants
0x0838	STAT_RX_PACKET_128_255_BYTES_LSB: 0838	Only in MAC+ PCS and MAC-only variants
0x083C	STAT_RX_PACKET_128_255_BYTES_MSB: 083C	Only in MAC+ PCS and MAC-only variants
0x0840	STAT_RX_PACKET_256_511_BYTES_LSB: 0840	Only in MAC+ PCS and MAC-only variants
0x0844	STAT_RX_PACKET_256_511_BYTES_MSB: 0844	Only in MAC+ PCS and MAC-only variants

Table 2-28: **Statistics Counters *(Cont'd)***

Hex Address	Register Name/Link to Description	Notes
0x0848	STAT_RX_PACKET_512_1023_BYTES_LSB: 0848	Only in MAC+ PCS and MAC-only variants
0x084C	STAT_RX_PACKET_512_1023_BYTES_MSB: 084C	Only in MAC+ PCS and MAC-only variants
0x0850	STAT_RX_PACKET_1024_1518_BYTES_LSB: 0850	Only in MAC+ PCS and MAC-only variants
0x0854	STAT_RX_PACKET_1024_1518_BYTES_MSB: 0854	Only in MAC+ PCS and MAC-only variants
0x0858	STAT_RX_PACKET_1519_1522_BYTES_LSB: 0858	Only in MAC+ PCS and MAC-only variants
0x085C	STAT_RX_PACKET_1519_1522_BYTES_MSB: 085C	Only in MAC+ PCS and MAC-only variants
0x0860	STAT_RX_PACKET_1523_1548_BYTES_LSB: 0860	Only in MAC+ PCS and MAC-only variants
0x0864	STAT_RX_PACKET_1523_1548_BYTES_MSB: 0864	Only in MAC+ PCS and MAC-only variants
0x0868	STAT_RX_PACKET_1549_2047_BYTES_LSB: 0868	Only in MAC+ PCS and MAC-only variants
0x086C	STAT_RX_PACKET_1549_2047_BYTES_MSB: 086C	Only in MAC+ PCS and MAC-only variants
0x0870	STAT_RX_PACKET_2048_4095_BYTES_LSB: 0870	Only in MAC+ PCS and MAC-only variants
0x0874	STAT_RX_PACKET_2048_4095_BYTES_MSB: 0874	Only in MAC+ PCS and MAC-only variants
0x0878	STAT_RX_PACKET_4096_8191_BYTES_LSB: 0878	Only in MAC+ PCS and MAC-only variants
0x087C	STAT_RX_PACKET_4096_8191_BYTES_MSB: 087C	Only in MAC+ PCS and MAC-only variants
0x0880	STAT_RX_PACKET_8192_9215_BYTES_LSB: 0880	Only in MAC+ PCS and MAC-only variants
0x0884	STAT_RX_PACKET_8192_9215_BYTES_MSB: 0884	Only in MAC+ PCS and MAC-only variants
0x0888	STAT_RX_PACKET_LARGE_LSB: 0888	Only in MAC+ PCS and MAC-only variants
0x088C	STAT_RX_PACKET_LARGE_MSB: 088C	Only in MAC+ PCS and MAC-only variants
0x0890	STAT_RX_PACKET_SMALL_LSB: 0890	Only in MAC+ PCS and MAC-only variants
0x0894	STAT_RX_PACKET_SMALL_MSB: 0894	Only in MAC+ PCS and MAC-only variants
0x0898	STAT_RX_UNDERSIZE_LSB: 0898	Only in MAC+ PCS and MAC-only variants

Table 2-28: **Statistics Counters *(Cont'd)***

Hex Address	Register Name/Link to Description	Notes
0x089C	STAT_RX_UNDERSIZE_MSB: 089C	Only in MAC+ PCS and MAC-only variants
0x08A0	STAT_RX_FRAGMENT_LSB: 08A0	Only in MAC+ PCS and MAC-only variants
0x08A4	STAT_RX_FRAGMENT_MSB: 08A4	Only in MAC+ PCS and MAC-only variants
0x08A8	STAT_RX_OVERSIZE_LSB: 08A8	Only in MAC+ PCS and MAC-only variants
0x08AC	STAT_RX_OVERSIZE_MSB: 08AC	Only in MAC+ PCS and MAC-only variants
0x08B0	STAT_RX_TOOLONG_LSB: 08B0	Only in MAC+ PCS and MAC-only variants
0x08B4	STAT_RX_TOOLONG_MSB: 08B4	Only in MAC+ PCS and MAC-only variants
0x08B8	STAT_RX_JABBER_LSB: 08B8	Only in MAC+ PCS and MAC-only variants
0x08BC	STAT_RX_JABBER_MSB: 08BC	Only in MAC+ PCS and MAC-only variants
0x08C0	STAT_RX_BAD_FCS_LSB: 08C0	Only in MAC+ PCS and MAC-only variants
0x08C4	STAT_RX_BAD_FCS_MSB: 08C4	Only in MAC+ PCS and MAC-only variants
0x08C8	STAT_RX_PACKET_BAD_FCS_LSB: 08C8	Only in MAC+ PCS and MAC-only variants
0x08CC	STAT_RX_PACKET_BAD_FCS_MSB: 08CC	Only in MAC+ PCS and MAC-only variants
0x08D0	STAT_RX_STOMPED_FCS_LSB: 08D0	Only in MAC+ PCS and MAC-only variants
0x08D4	STAT_RX_STOMPED_FCS_MSB: 08D4	Only in MAC+ PCS and MAC-only variants
0x0910	STAT_RX_TRUNCATED_LSB: 0910	Only in MAC+ PCS and MAC-only variants
0x0914	STAT_RX_TRUNCATED_MSB: 0914	Only in MAC+ PCS and MAC-only variants
0x0918	STAT_RX_TEST_PATTERN_MISMATCH_LSB: 0918	Only in MAC+PCS variant
0x091C	STAT_RX_TEST_PATTERN_MISMATCH_MSB: 091C	Only in MAC+PCS variant

For description of statistics counters, see Statistics Counters Tables in *10G/25G High Speed Ethernet Subsystem Product Guide* (PG210)[Ref 3] for the bit assignments for the statistics counters.

Designing with the Subsystem

This chapter includes guidelines and additional information to facilitate designing with the subsystem.

Clocking

This section describes the clocking for all the 1G configurations at the component support wrapper layer.

32Bit 1/10/25G Ethernet MAC with PCS/PMA Clocking

The clocking architecture for the 32Bit 1/10/25G Ethernet MAC with PCS/PMA clocking is illustrated in Figure 3-1. Low latency is achieved by omitting the RX FIFOs, which results in different clocking arrangement. There are two clock domains in the datapath, as illustrated by the dashed lines in Figure 3-1.

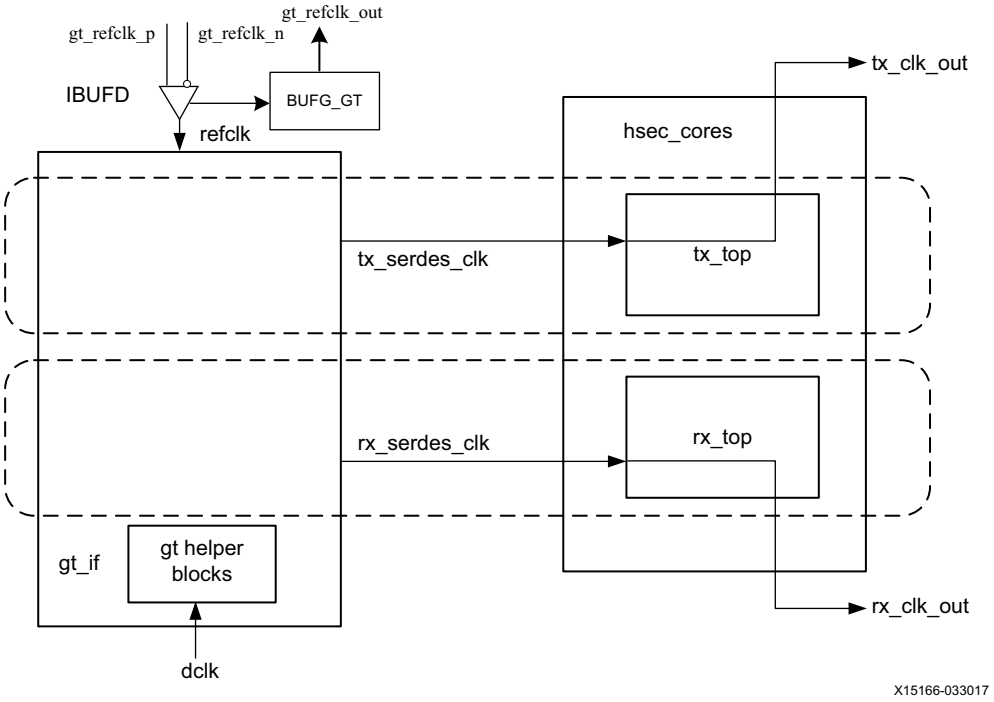

Figure 3-1: **32Bit 1/10/25G Ethernet MAC with PCS/PMA Clocking**

refclk_p0, refclk_n0, tx_serdes_refclk

The `refclk` differential pair is required to be an input to the FPGA. The example design includes a buffer to convert this clock to a single-ended signal `refclk`, which is used as the reference clock for the GT block. The `tx_serdes_refclk` is directly derived from `refclk`. Note that `refclk` must be chosen so that the `tx_serdes_refclk` meets the requirements of 802.3, which is within 100 ppm of 390.625 MHz for 25G, and 156.25 MHz for 10G.

tx_clk_out

This clock is used for clocking data into the TX AXI4-Stream Interface and it is also the reference clock for the TX control and status signals. It is the same frequency as `tx_serdes_refclk`. Because there is no TX FIFO, you must respond immediately to the `tx_axis_tready` signal.

rx_clk_out

The `rx_clk_out` output signal is presented as a reference for the RX control and status signals processed by the RX core. It is the same frequency as the `rx_serdes_clk`. Because there is no RX FIFO, this is also the clock which drives the RX AXI4-Stream Interface. In this arrangement, `rx_clk_out` and `tx_clk_out` are different frequencies and have no defined phase relationship to each other.

dclk

The `dclk` signal must be a convenient stable clock. It is used as a reference frequency for the GT helper blocks which initiate the GT itself. In the example design, a typical value is 75 MHz, which is readily derived from the 300 MHz clock available on the VCU107 evaluation board. Note that the actual frequency must be known to the GT helper blocks for proper operation.

PCS/PMA Only Clocking

The clocking architecture for the 10G/25G PCS is illustrated below. There are three clock domains in the datapath, as illustrated by the dashed lines in Figure 3-2.

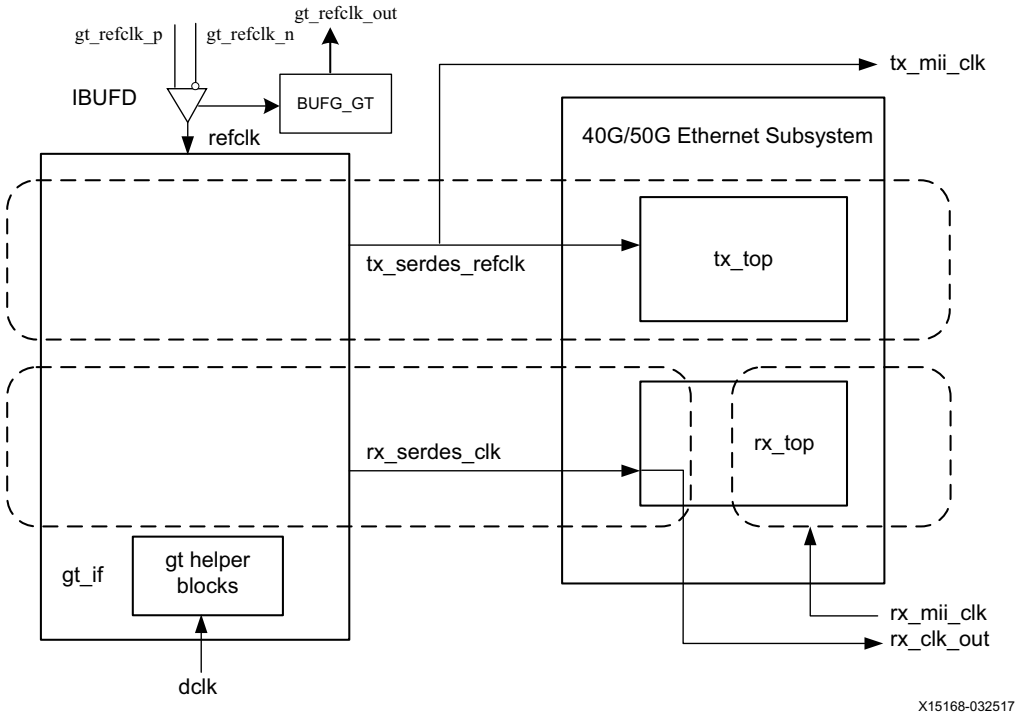

Figure 3-2: **PCS/PMA Clocking**

refclk_p0, refclk_n0, tx_serdes_refclk

The `refclk` differential pair is required to be an input to the FPGA. The example design includes a buffer to convert this clock to a single-ended signal `refclk`, which is used as the reference clock for the GT block. The `tx_serdes_refclk` is directly derived from `refclk`. Note that `refclk` must be chosen so that the `tx_mii_clk` meets the requirements of 802.3, which is within 100 ppm of 390.625 MHz for 25G and 156.25 MHz for 10G.

tx_mii_clk

The tx_mii_clk is an output which is the same as the tx_serdes_refclk. The entire TX path is driven by this clock. You must synchronize the TX path mii bus to this clock output. All TX control and status signals are referenced to this clock.

rx_serdes_clk

The rx_serdes_clk is derived from the incoming data stream within the GT block. The incoming data stream is processed by the RX core in this clock domain.

rx_clk_out

The rx_clk_out output signal is presented as a reference for the RX control and status signals processed by the RX core. It is the same frequency as the rx_serdes_clk.

rx_mii_clk

The rx_mii_clk input is required to be synchronized to the RX XGMII/25GMII data bus. This clock and the RX XGMII/25GMII bus must be within 100 ppm of the required frequency, which is 390.625 MHz for 25G and 156. 25 MHz for 10G.

dclk

The dclk signal must be a convenient stable clock. It is used as a reference frequency for the GT helper blocks which initiate the GT itself. In the example design, a typical value is 75 MHz, which is readily derived from the 300 MHz clock available on the VCU107 evaluation board. Note that the actual frequency must be known to the GT helper blocks for proper operation.

Resets

Figure 3-3 shows the reset structure for the 10G/25G Ethernet MAC with PCS/PMA as implemented at the component support wrapper layer. Clocks are not shown for clarity.

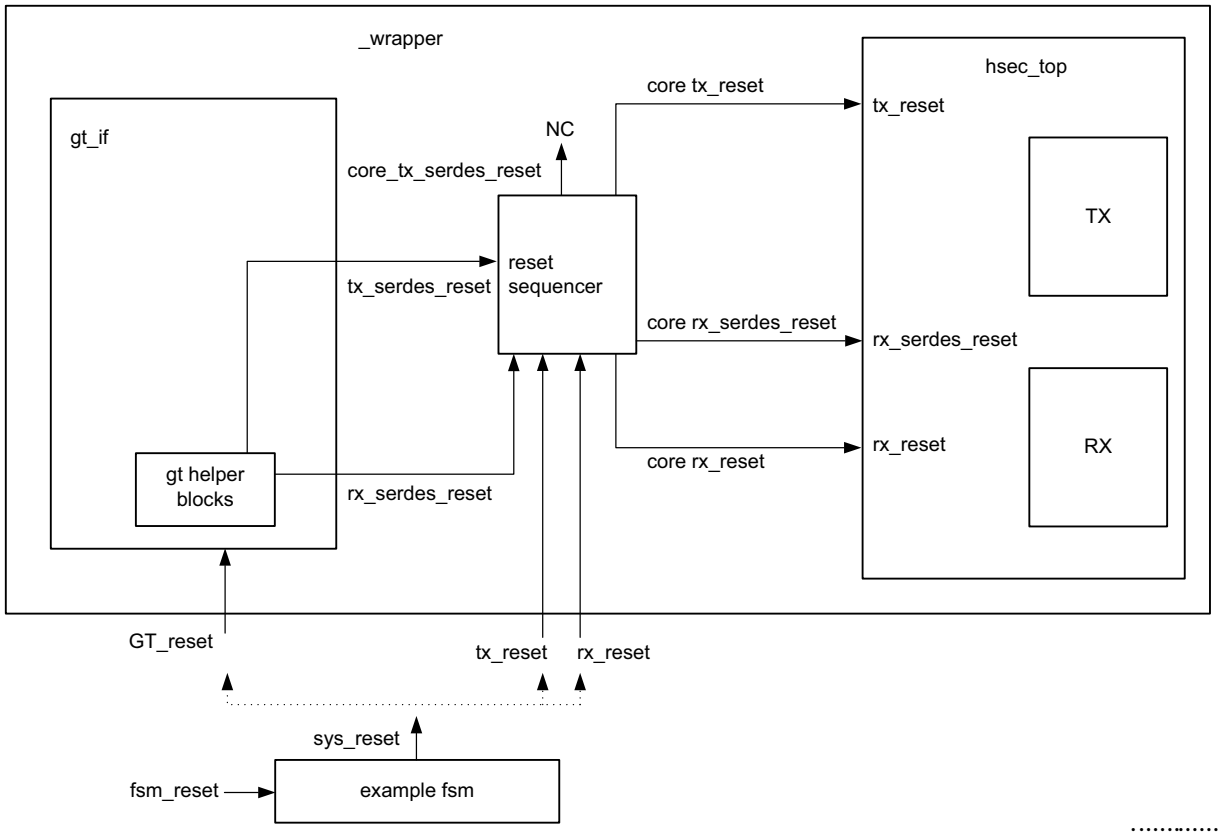

Figure 3-3: **Reset Structure**

Component Support Layer Resets

In the example design, a single reset is used to reset the entire wrapper layer. Using the external stimulus `fsm_reset`, the example_fsm block issues the signal `sys_reset` which is connected to the three `_wrapper` resets. The example design demonstrates that all three wrapper resets can be released simultaneously and correct operation follows.

Wrapper Resets

The _wrapper layer of the hierarchy is assumed to be what you instantiate in your own design. There are three resets to be handled as follows:

- GT_reset
- tx_reset
- rx_reset

Timing of the reset signals is handled by the reset_sequencer block.

GT_reset

The `GT_reset` is the asynchronous active-High reset input to the GT. Internal resets of the GT are handled by the GT helper blocks.

tx_reset

The `tx_reset` is the asynchronous active-High reset for the TX path logic of the 10G/25G Ethernet IP core. While it is connected to the GT reset in the example design, this reset can be asserted at any time to reset the TX path independently without disturbing the RX path.

rx_reset

The `rx_reset` is the asynchronous active-High reset for the RX path logic of the 10G/25G Ethernet IP core. While it is connected to the GT reset in the example design, this reset can be asserted at any time to reset the RX path independently without disturbing the TX path.

LogiCORE Example Design Clocking and Resets

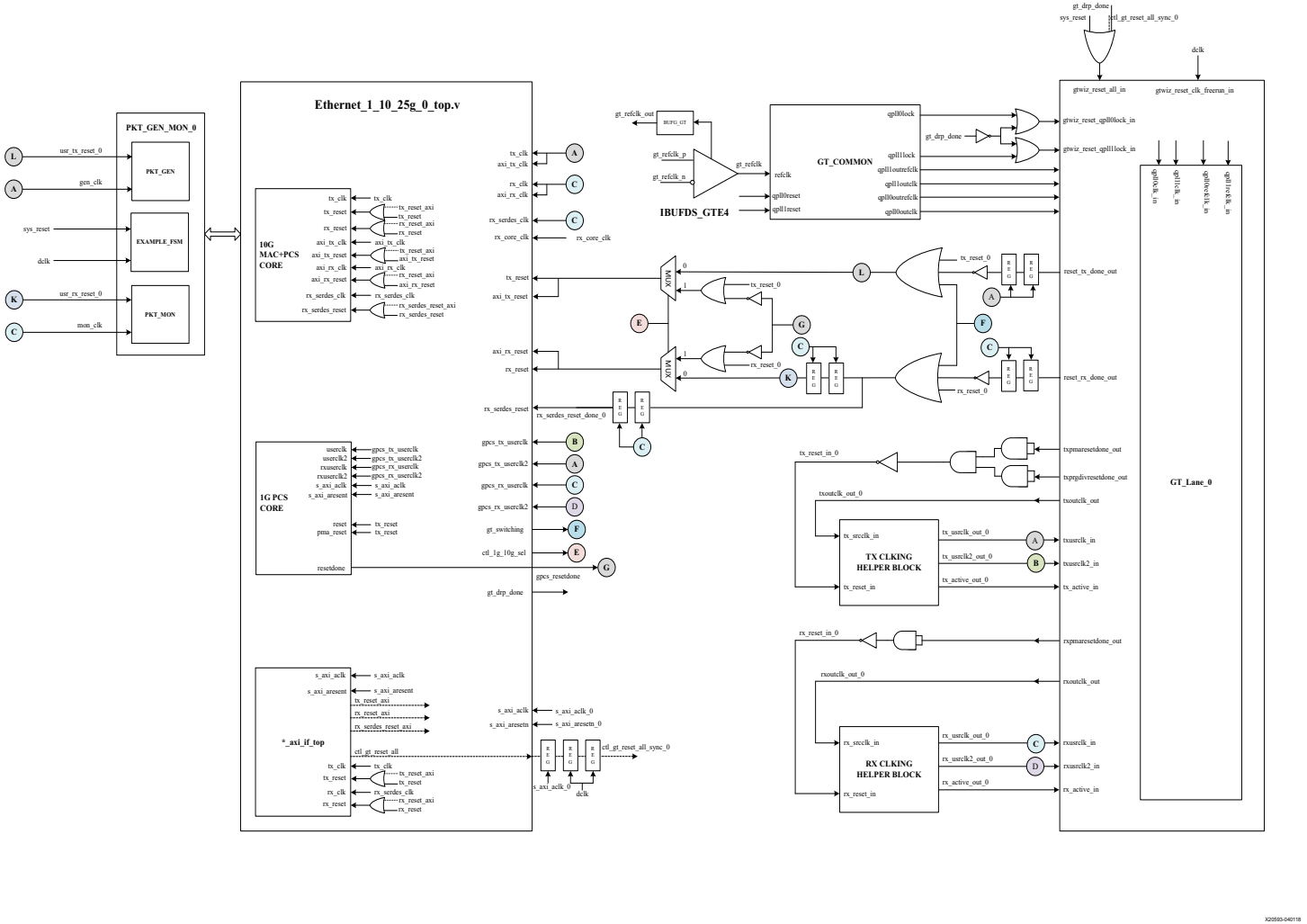

Figure 3-4: **Detailed diagram of 32-bit MAC+PCS/PMA Single Core**

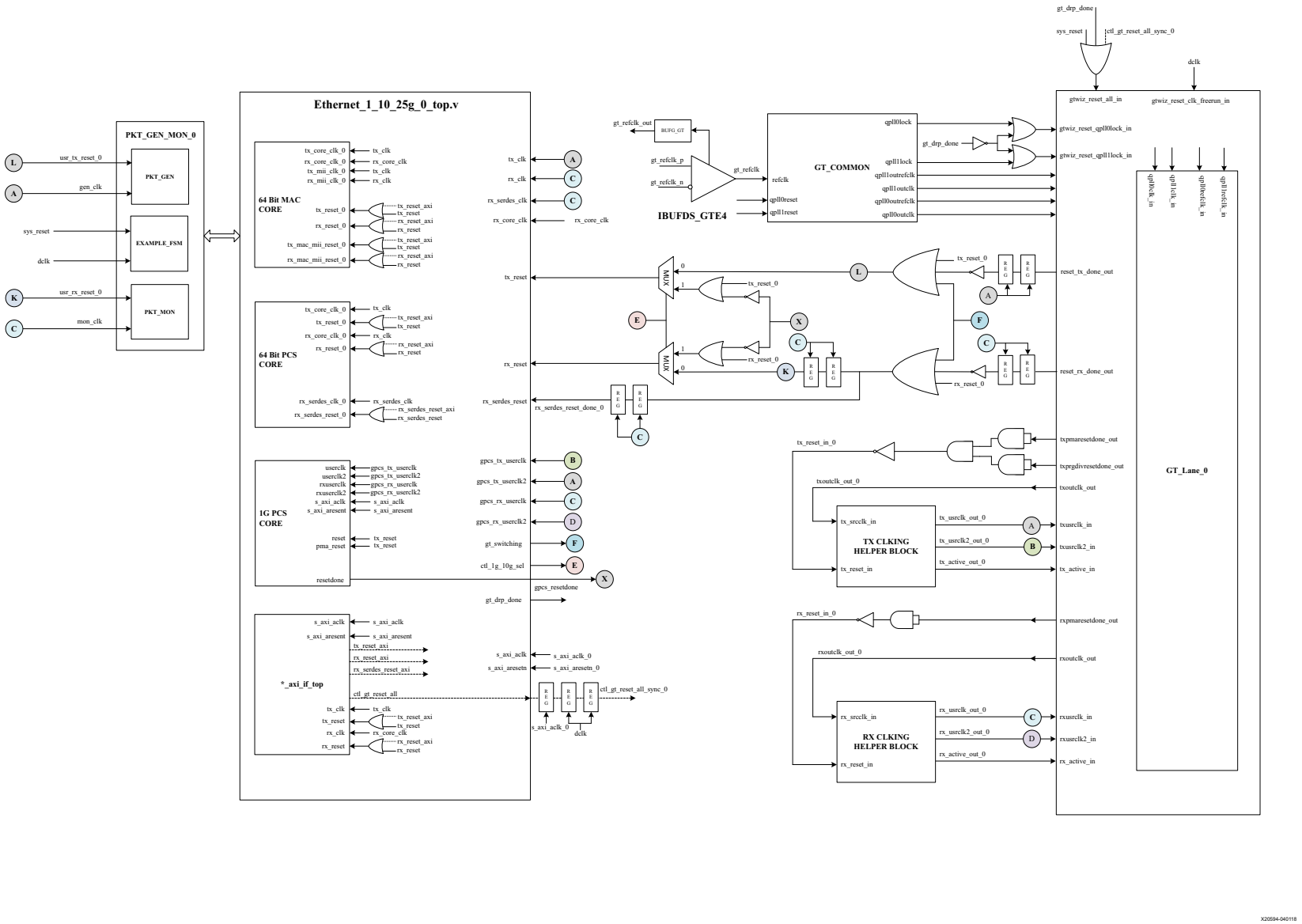

Figure 3-5: **Detailed diagram of 64-bit MAC+PCS/PMA Single Core**

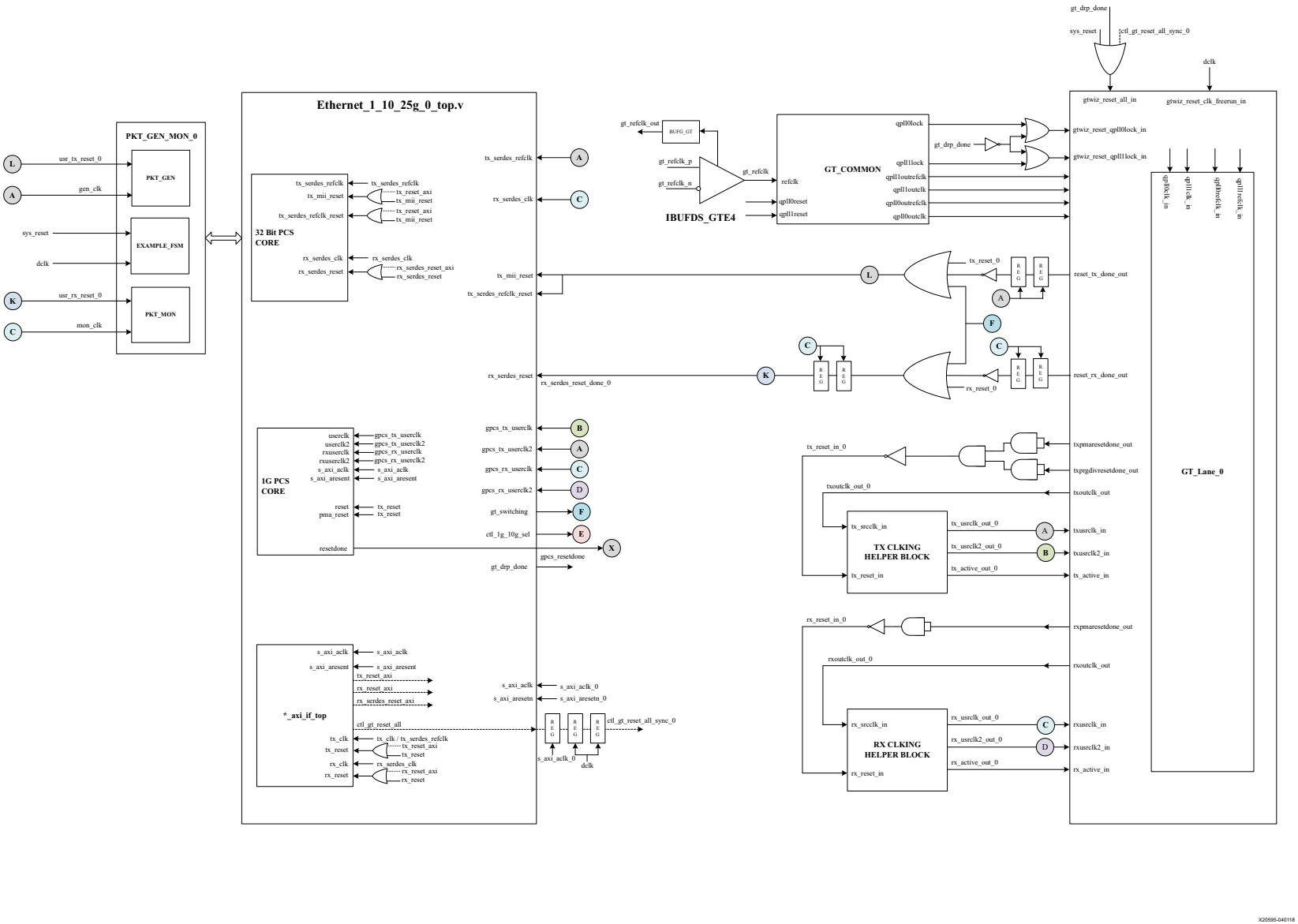

Figure 3-6: **Detailed diagram of 32-bit PCS only Single Core**

Support for IEEE Standard 1588v2

Overview

This section details the packet timestamping function of the Ethernet_1_10_25G subsystem when the MAC layer is included. The timestamping option must be specified at the time of generating the subsystem from the IP catalog or ordering the IP Core asynchronously. This feature provides support only for two-step IEEE 1588v2 functionality.

Ethernet frames are timestamped at both ingress and egress. The option can be used for implementing all kinds of IEEE 1588v2 clocks: Ordinary, Transparent, and Boundary. It can also be used for the generic timestamping of packets at the ingress and egress ports of a system. While this feature can be used for a variety of packet timestamping applications, the rest of this section assumes that you are also implementing the IEEE 1588v2 Precision Time Protocol (PTP).

IEEE 1588v2 defines a protocol for performing timing synchronization across a network. A 1588 network has a single master clock timing reference, usually selected through a best master clock algorithm. Periodically, this master samples its system timer reference counter, and transmits this sampled time value across the network using defined packet formats. This timer should be sampled (a timestamp) when the start of a 1588 timing packet is transmitted. Therefore, to achieve high synchronization accuracy over the network, accurate timestamps are required. If this sampled timer value (the timestamp) is placed into the packet that triggered the timestamp, this is known as one-step operation. Alternatively, the timestamp value can be placed into a follow up packet; this is known as two-step operation.

Other timing slave devices on the network receive these timing reference packets from the network timing master and attempt to synchronize their own local timer references to it. This mechanism relies on these Ethernet ports also taking timestamps (samples of their own local timer) when the 1588 timing packets are received. Further explanation of the operation of 1588 is out of scope of this document. This document now describes the 1588 hardware timestamping features of the subsystem.

The 1588 timer provided to the subsystem and the consequential timestamping taken from it are available in one of two formats which are selected during subsystem generation.

- Time-of-Day (ToD) format: IEEE 1588-2008 format consisting of an unsigned 48-bit second field and a 32-bit nanosecond field.
- Correction Field format: IEEE 1588-2008 numerical format consisting of a 64-bit signed field representing nanoseconds multiplied by 2^{16} (see IEEE 1588 clause 13.3.2.7). This timer should count from 0 through the full range up to 2^{64} -1 before wrapping around.

Egress

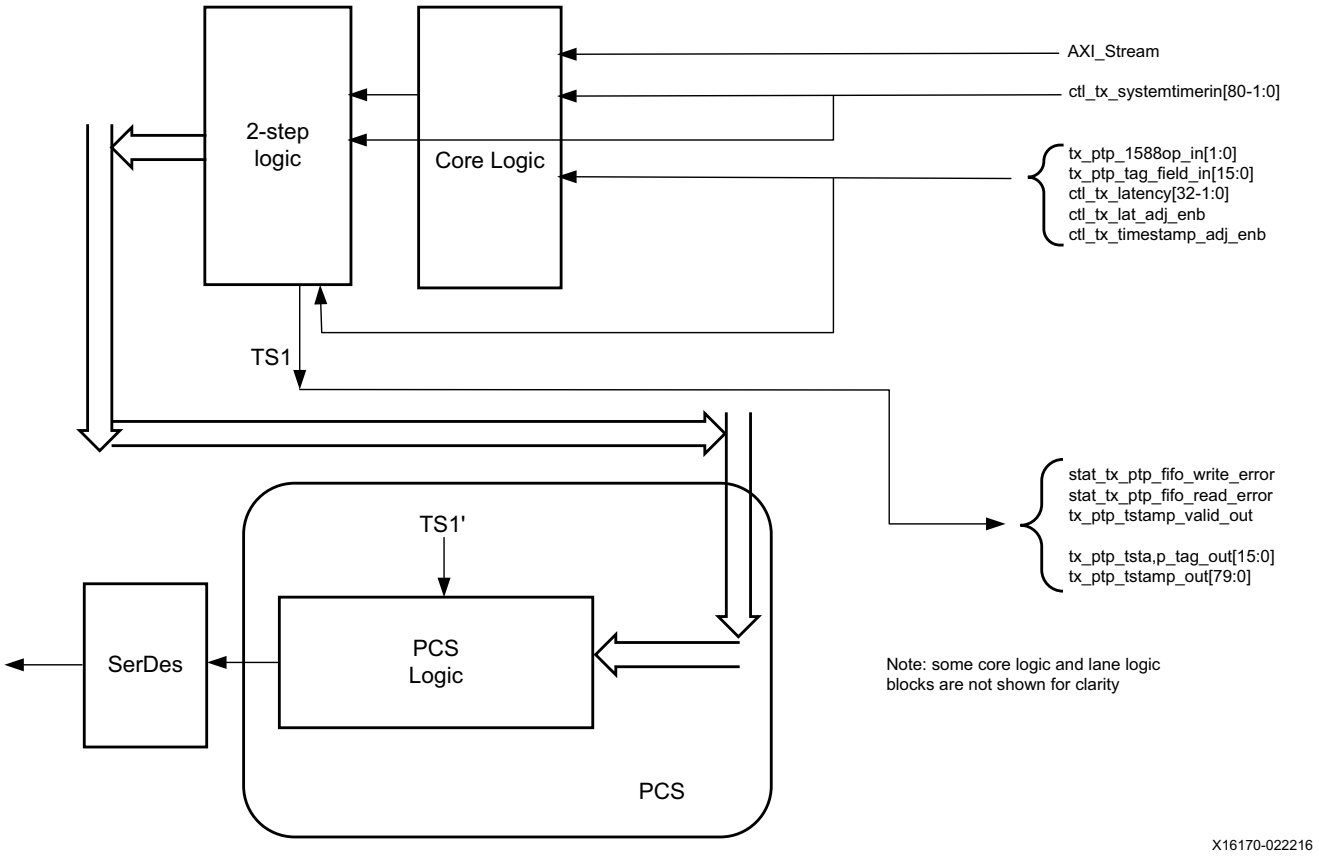

Figure 3-7: **Egress**

The TS references are defined as follows:

- TS1: The output timestamp signal when a 2-step operation is selected.
- TS1': The plane to which both timestamps are corrected.

TS1 always has a correction applied so that it is referenced to the TS1' plane.

Note: For 10G 1588, registers defined at address 0x0190, 0x0194 and 0x0198 must be programmed. For 1G 1588, registers defined at address 0x0190, 0x0194, 0x004C, 0x0050 and 0x0054 must be programmed.

If using the ToD format, the full captured 80-bit ToD timestamp is returned to the client logic using the additional ports defined in Table 3-1.

If using the Correction Field format, the full captured 64-bit timestamp is returned to the client logic using the additional ports defined in Table 3-1 (with the upper bits of data set to zero as defined in the table).

For 2-step transmit operation, all Precision Time Protocol (PTP) frame types are supported.

Frame-by-Frame Timestamping Operation

The operational mode of the egress timestamping function is determined by the settings on the 1588 command port. The information contained within the command port indicates one of the following:

- No operation: the frame is not a PTP frame and no timestamp action should be taken.
- Two-step operation is required and a tag value (user-sequence ID) is provided as part of the command field; the frame should be timestamped, and the timestamp made available to the client logic, along with the provided tag value for the frame. The additional MAC transmitter ports provide this function.

Ingress

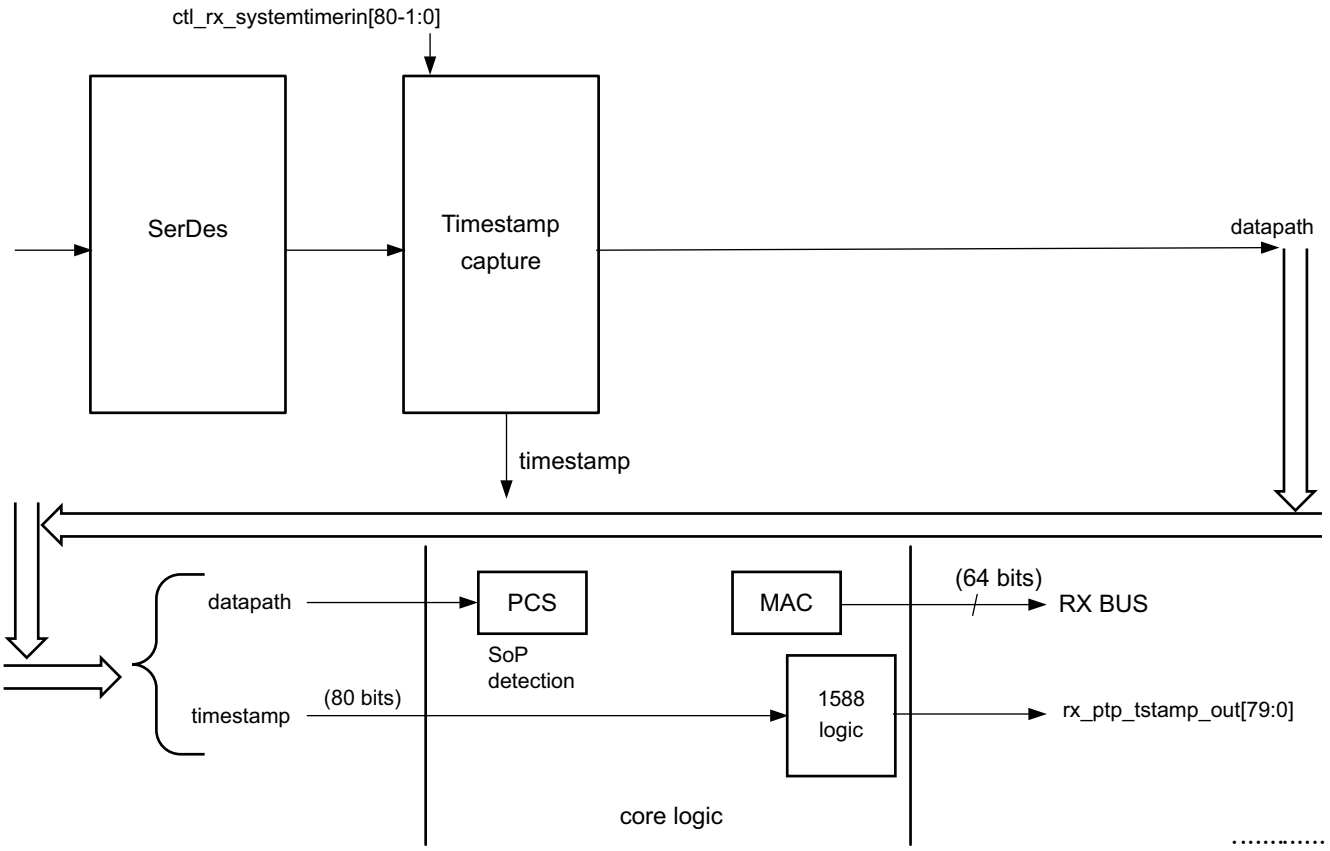

Figure 3-8: **Ingress**

The ingress logic does not parse the ingress packets to search for 1588 (PTP) frames. Instead, it takes a timestamp for every received frame and outputs this value to the user logic. The feature is always enabled, but the timestamp output can be ignored if you do not require this function.

Timestamps are filtered after the PCS decoder to retain only those timestamps corresponding to an Start Of Packet (SOP). These 80-bit timestamps are output on the system side. The timestamp is valid during the SoP cycle and when ena_out = 1.

Port Descriptions

The following table details the additional signals present when the packet timestamping feature is included.

Table 3-1: **1588v2 Port List and Descriptions**

Name	Direction	Description	Clock Domain
COMMON			
systemtimerin	Input	Common System timer input. In TOD mode, the 32 LSBs carry nsec and the 48 MSBs carry seconds. In transparent clock mode, bit 63 is expected to be zero, bits 62:16 carry nanoseconds, and bits 15:0 carry fractional nanoseconds. Refer to IEEE 1588v2 for the representational definitions.	
IEEE 1588 Interface – TX Path			
tx_ptp_tstamp_valid_out	Output	This bit indicates that a valid timestamp is being presented on the TX system interface.	tx_clk_out
tx_ptp_tstamp_tag_out[15:0]	Output	Tag output corresponding to tx_ptp_tag_field_in[15:0]	tx_clk_out
tx_ptp_tstamp_out[80-1:0]	Output	Timestamp for the transmitted packet SOP corresponding to the time at which it passed the capture plane. Time format same as timer input.	tx_clk_out
tx_ptp_1588op_in[1:0]	Input	The signal should be valid on the first cycle of the packet. For PCS cores, the first cycle corresponds with the first data word of the packet. 2'b00 – No operation: no timestamp will be taken and the frame will not be modified. 2'b01 – Reserved. 2'b10 – 2-step: a timestamp should be taken and returned to the client using the additional ports of 2-step operation. The frame itself will not be modified. 2'b11 – Reserved: act as No operation.	tx_clk_out

Table 3-1: **1588v2 Port List and Descriptions *(Cont'd)***

Name	Direction	Description	Clock Domain
tx_ptp_tag_field_in[15:0]	Input	The usage of this field is dependent on the 1588 operation. The signal should be valid on the first cycle of the packet. • For No operation, this field will be ignored. • For 1-step and 2-step this field is a tag field. This tag value will be returned to the client with the timestamp for the current frame using the additional ports of 2-step operation. This tag value can be used by software to ensure that the timestamp can be matched with the PTP frame that it sent for transmission.	tx_clk_out
stat_tx_ptp_fifo_write_error	Output	Transmit PTP FIFO write error. A value of 1 on this status indicates that an error occurred during the PTP Tag write. A TX Path reset is required to clear the error.	tx_clk_out
stat_tx_ptp_fifo_read_error	Output	Transmit PTP FIFO read error. A value of 1 on this status indicates that an error occurred during the PTP Tag read. A TX Path reset is required to clear the error.	tx_clk_out
ctl_tx_latency	Input	This is the static latency of the TX path of the core including the GT. The MSB 16 bits indicate the delay in ns and the LSB 16 bits indicate sub ns values. The latency is in binary Q16.16 format.	tx_clk_out
ctl_tx_lat_adj_enb	Input	When this signal is enabled, the delay computation on the TX path takes into account the value provided by the ctl_tx_latency_0 register.	tx_clk_out
ctl_tx_timestamp_adj_enb	Input	When this signal is enabled, the delay computation on the TX path takes into account the value got from GT DRP read for latency of TX gearbox FIFO. Since the design does not use TX gearbox FIFO, this signal need not be updated.	tx_clk_out
IEEE 1588 Interface – RX Path			
ctl_rx_systemtimerin[80-1:0]	Input	System timer input for the RX. Same time format as the TX. This input must be in the same clock domain as the RX SerDes.	rx_serdes_clk
rx_ptp_tstamp_out[80-1:0]	Output	Timestamp for the received packet SOP corresponding to the time at which it passed the capture plane. Note that this signal will be valid on the first cycle of the packet.	rx_clk_out
rx_ptp_tstamp_valid_out	Output	This bit indicates that a valid timestamp is being presented on the rx	rx_serdes_clk

Table 3-1: **1588v2 Port List and Descriptions *(Cont'd)***

Name	Direction	Description	Clock Domain
ctl_rx_latency	Input	This is the static latency of the RX path of the core including the GT. The MSB 16 bits indicate the delay in ns and the LSB 16 bits indicate sub ns values. The latency is in binary Q16.16 format. ***Note:*** This signal is not valid in 1G mode.	rx_clk_out
ctl_rx_lat_adj_enb	Input	When this signal is enabled, the delay computation on the RX path takes into account the value provided by the ctl_rx_latency_0 register. ***Note:*** This signal is not valid in 1G mode.	rx_clk_out
ctl_rx_timestamp_adj_enb	Input	When this signal is enabled, the delay computation on the RX path takes into account the value obtained from the GT DRP read for latency of the RX gearbox FIFO. ***Note:*** This signal is not valid in 1G mode.	rx_clk_out

IEEE 1588v2 PTP Functional Description

The IEEE 1588v2 feature of the 10G/25G High Speed Ethernet subsystem provides accurate timestamping of Ethernet frames at the hardware level for both the ingress and egress directions.

Timestamps are captured according to the input clock source above. However, it is required that this time source be in the same clock domain as the SerDes. This might require re-timing by an external circuit provided by the user.

All ingress frames receive a timestamp. It is up to you to interpret the received frames and determine whether a particular frame contains PTP information (by means of its Ethertype) and if the timestamp needs to be retained or discarded.

Egress frames are timestamped if they are tagged as PTP frames. The timestamps of egress frames are matched to their user-supplied tags.

Timestamps for incoming frames are presented at the user interface during the same clock cycle as the start of packet. You can then append the timestamp to the packet as required.

By definition, a timestamp is captured coincident with the passing of the SOP through the capture plane within the 10G/25G High Speed Ethernet subsystem. This is illustrated in the following schematic diagrams:

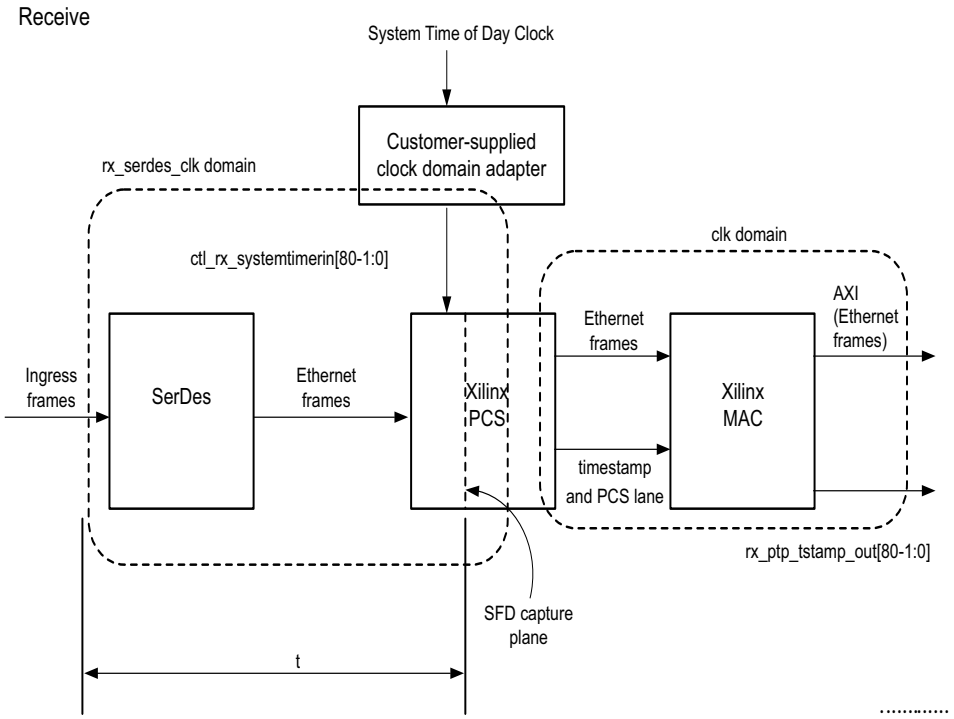

Figure 3-9: **Receive**

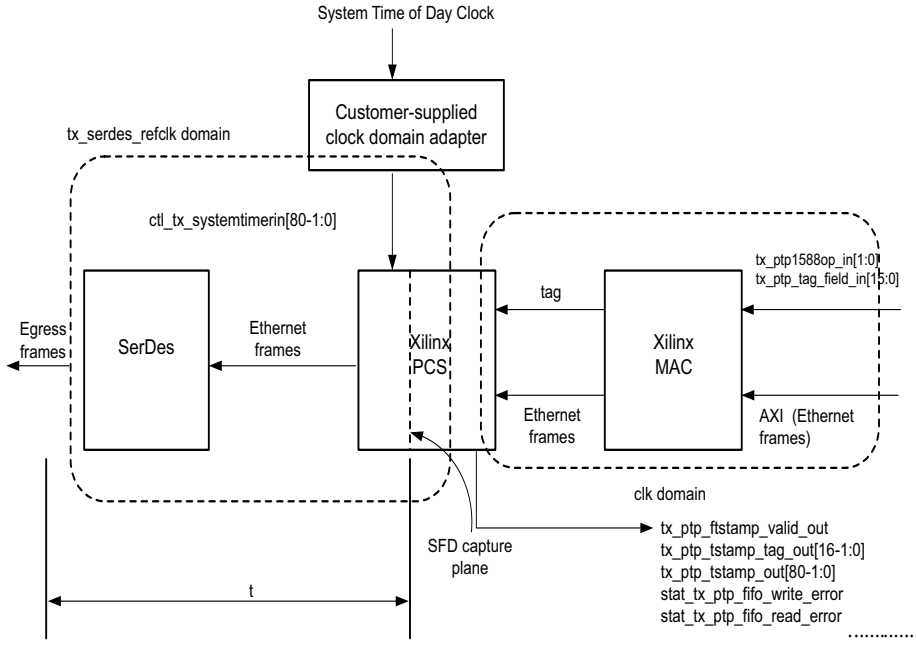

Figure 3-10: **Transmit**

Performance

In a typical application, the difference between the ingress and egress capture times is important for determining absolute time. The PTP algorithm can use asymmetry information to improve accuracy.

The 1588v2 feature requires that all clock frequencies be known in order to make internal calculations. The clock frequencies should be specified at the time the PTP IP core is ordered in order for the timestamp correction to work properly.

In a typical application, the PTP algorithm (or servo, not part of this IP) will remove jitter over the course of time (many packet samples). It is advantageous for the jitter to be as small as possible in order to minimize the convergence time as well as minimizing slave clock drift.

Status/Control Interface

The Status/Control interface allows you to set up the 10G/25G Ethernet core configuration and to monitor its status. This sections describes in more detail some of the Status and Control signals.

stat_rx_framing_err and stat_rx_framing_err_valid

These signals are used to keep track of sync header errors. This set of buses is used to keep track of sync header errors. The `stat_rx_framing_err` output indicates how many sync header errors were received and it is qualified (that is, the value is only valid) when the corresponding `stat_rx_framing_err_valid` is sampled as a 1.

stat_rx_block_lock

This bit indicates that the interface has achieved sync header lock as defined by IEEE Std. 802.3. A value of 1 indicates block lock is achieved.

stat_rx_local_fault

This output is High when `stat_rx_internal_local_fault` or `stat_rx_received_local_fault` is asserted. This is output is level sensitive.

RX Error Status

The core provides status signals to identify 64b/66b words and sequences violations and CRC32 checking failures.

All signals are synchronous with the rising-edge of `clk` and a detailed description of each signal follows.

stat_rx_bad_fcs[1:0]

When this signal is positive, it indicates that the error detection logic has identified mismatches between the expected and received value of CRC32 in the received packet.

When a CRC32 error is detected, the received packet is marked as containing an error and is sent with `rx_errout` asserted during the last transfer (the cycle with `rx_eopout` asserted), unless `ctl_rx_ignore_fcs` is asserted. This signal is asserted for one clock period for each CRC32 error detected.

stat_rx_bad_code

This signal indicates how many cycles the RX PCS receive state machine is in the RX_E state as defined by IEEE Std. 802.3.

Design Flow Steps

This chapter describes customizing and generating the core, constraining the core, and the simulation, synthesis and implementation steps that are specific to this core. More detailed information about the standard Vivado® design flows and the Vivado IP integrator can be found in the following Vivado Design Suite user guides:

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 3]
- *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6]
- *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 7]
- *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 8]

Customizing and Generating the Core

This section includes information about using Xilinx tools to customize and generate the core in the Vivado Design Suite.

If you are customizing and generating the core in the Vivado IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 3] for detailed information. IP integrator might auto-compute certain configuration values when validating or generating the design. To check whether the values do change, see the description of the parameter in this chapter. To view the parameter value, run the `validate_bd_design` command in the Tcl console.

You can customize the IP for use in your design by specifying values for the various parameters associated with the core using the following steps:

1. Select the IP from the IP catalog.
2. Double-click the selected IP or select the Customize IP command from the toolbar or right-click menu.

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6] and the *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 7].

Note: Figures in this chapter are illustrations of the Vivado IDE. The layout depicted here might vary from the current version.

Configuration Tab

The Configuration tab provides the basic core configuration options. Default values are pre-populated in all tabs.

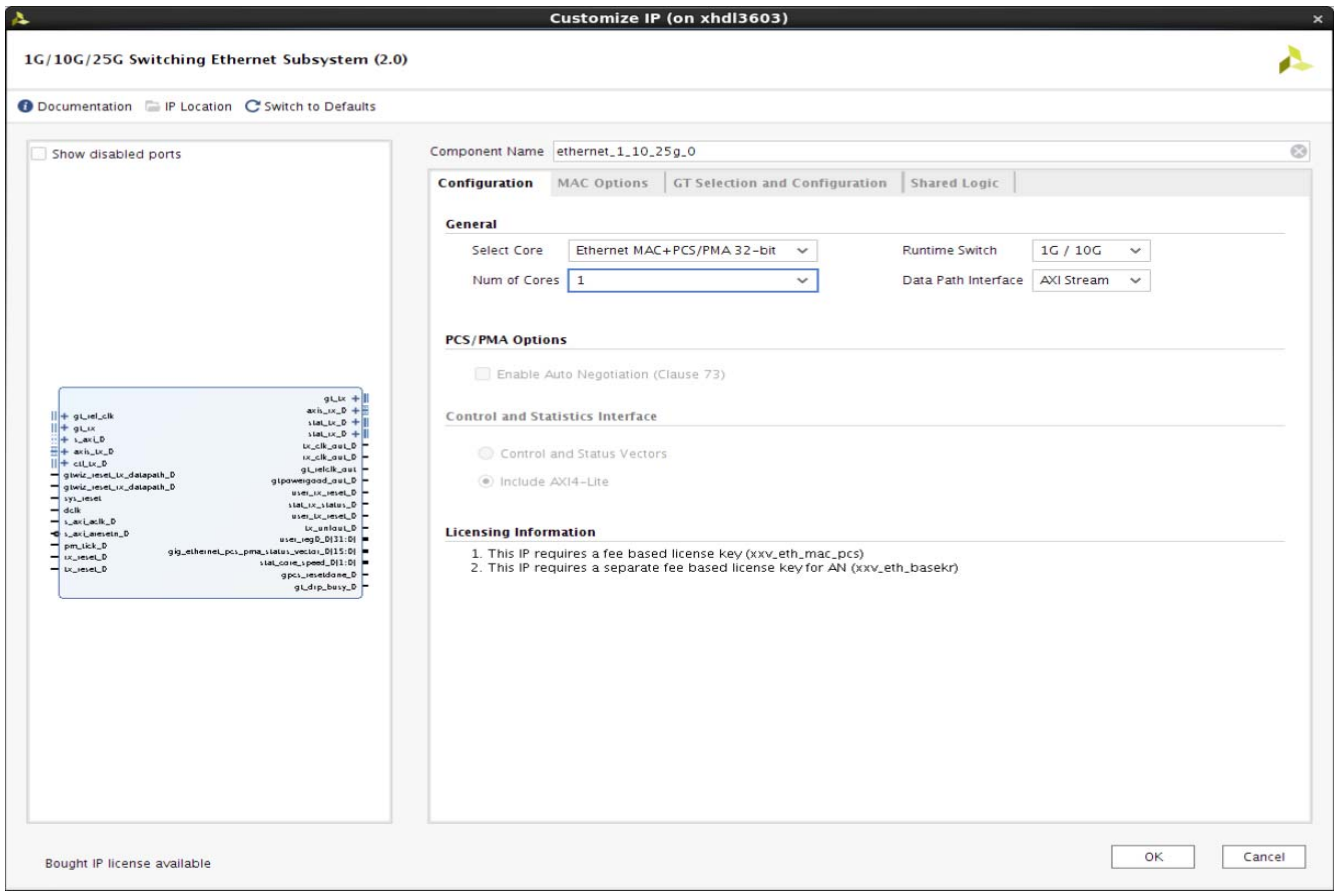

Figure 4-1: **Configuration Tab**

Table 4-1shows the Configuration options of 1G/10G/25G Switching Ethernet Subsystem.

Table 4-1: **Configuration Options**

Option	Values	Default
General		
Select Core	Ethernet MAC+PCS/PMA 32-bit Ethernet MAC+PCS/PMA 64-bit and Ethernet PCS/PMA 32-bit	Ethernet MAC+PCS/PMA 32-bit
Runtime Switch	1G/10G/25G	1G/10G

Table 4-1: **Configuration Options**

Option	Values	Default
Num of Cores	1 2 3 4	1
Data Path Interface	AXI Stream(1) MII(2)	AXI Stream
Auto Negotiation/Link Training Logic		
Enable Auto Negotiation (Clause 73)	0,1	0
Control and Statistics Interface		
Control and Statistics interface	Control and Status Vectors Include AXI4-Lite	Include AXI4-Lite

Notes:

1. The AXI Stream interface is visible and is the only option for the Ethernet MAC+PCS/PMA.
2. The MII interface is visible and is only option for the Ethernet PCS/PMA.

MAC Options Tab

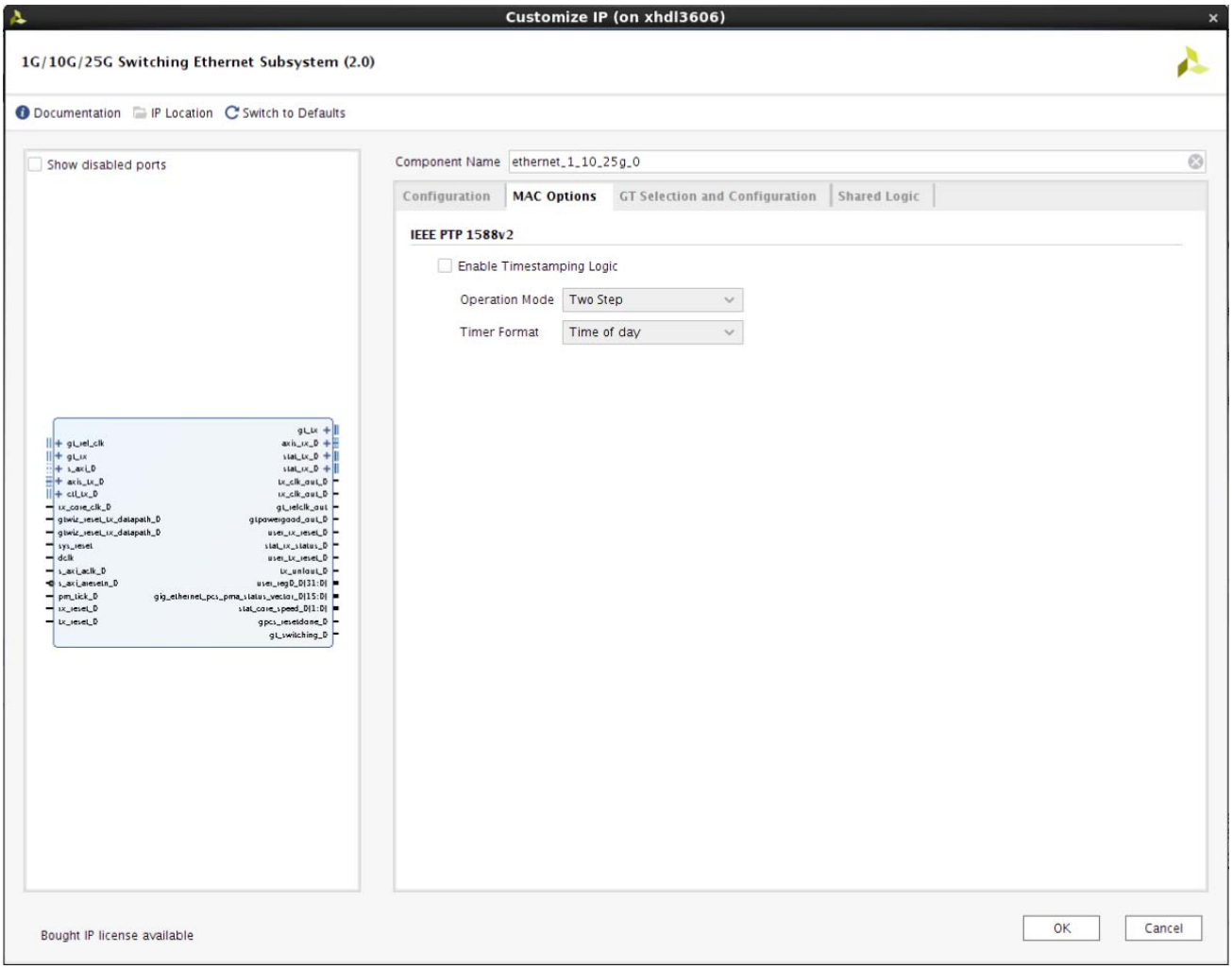

Figure 4-2: **MAC Options Tab**

Table 4-2shows the MAC options of 1G/10G/25G Switching Ethernet Subsystem.

Table 4-2: **MAC Options**

Option	Values	Default
IEEE PTP 1588v2		
Enable Timestamping Logic	Checked, Unchecked	Unchecked
Operation Mode	One Step Two Step	Two Step
Timer Format	Time of the day Correction Field Format	Time of the day

GT Selection and Configuration Tab

The GT Selection and Configuration tab enables you to configure the serial transceiver features of the core.

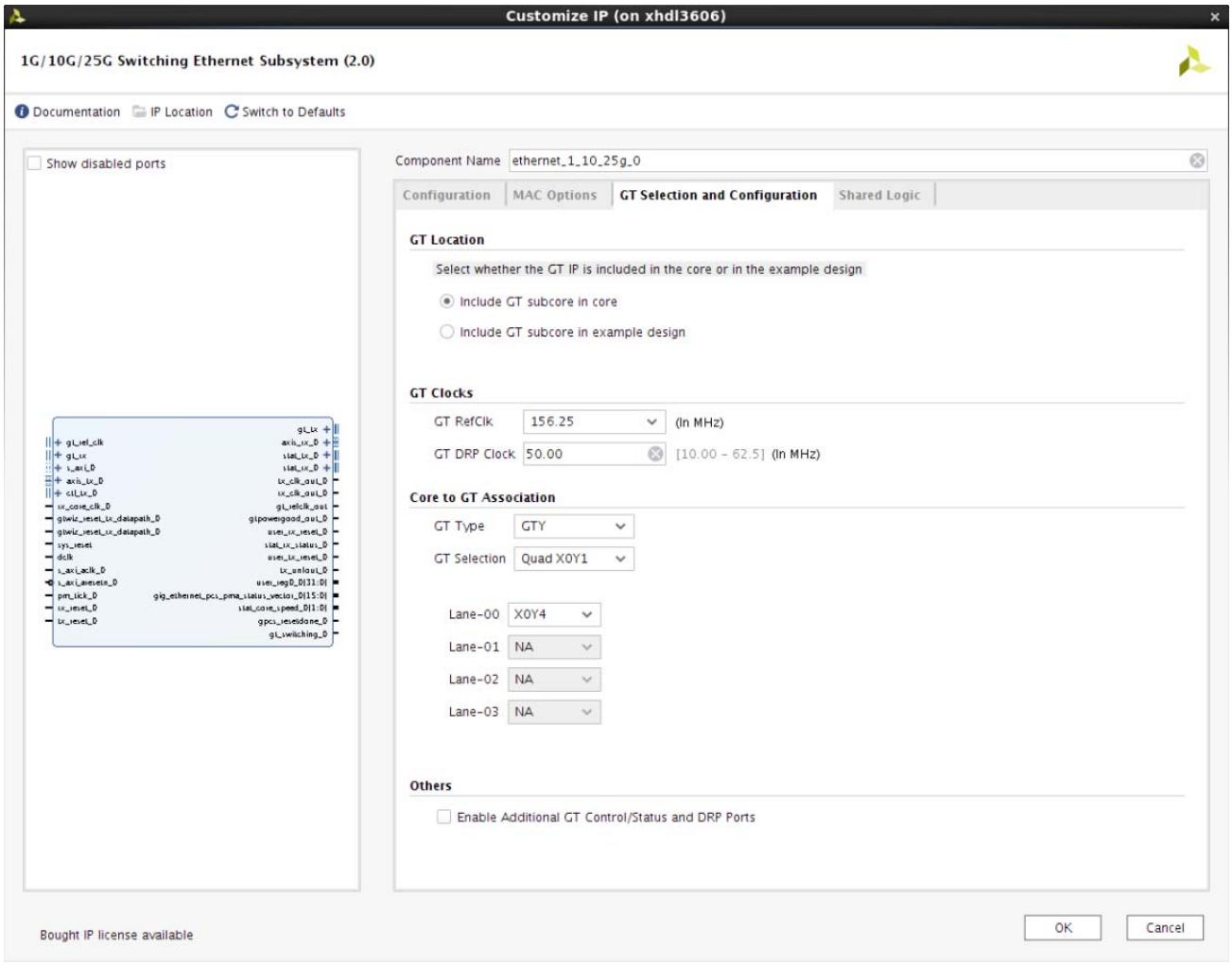

Figure 4-3: **GT Selection and Configuration Tab**

Table 4-3shows the GT Selection and Configuration options of 1G/10G/25G Switching Ethernet Subsystem.

Table 4-3: **GT Clocks Options**

Option	Values	Default
GT Location		
Select whether the GT IP is included in the core or in the example design	Include GT subcore in core Include GT subcore in example design	Include GT subcore in core
GT Clocks		
GT RefClk (In MHz)(1)	156.25	156.25

Table 4-3: **GT Clocks Options**

Option	Values	Default
GT DRP Clock (In MHz)	10 – 62.5	50.00
Core to GT Association		
GT Type	GTY GTH	GTH
GT Selection	Options based on device/package Quad groups. For example: Quad X0Y1 Quad X0Y2 Quad X0Y3 ...	Quad X0Y1
Lane-00 to Lane-03	Auto filled based on device/package. For example, if Num of Core = 4, and GT Selection = Quad X0Y1, four lanes are: X0Y4 X0Y5 X0Y6 X0Y7	
Others		
Enable Additional GT Control/Status and DRP Ports	Checked, Unchecked	Unchecked

Notes:

1. This list provides a list of the popular frequencies used. See Vivado IDE in the latest version of the tools for a complete list of supported clock frequencies.

Shared Logic Tab

The Shared Logic tab enables you to use shared logic in either the core or the example design.

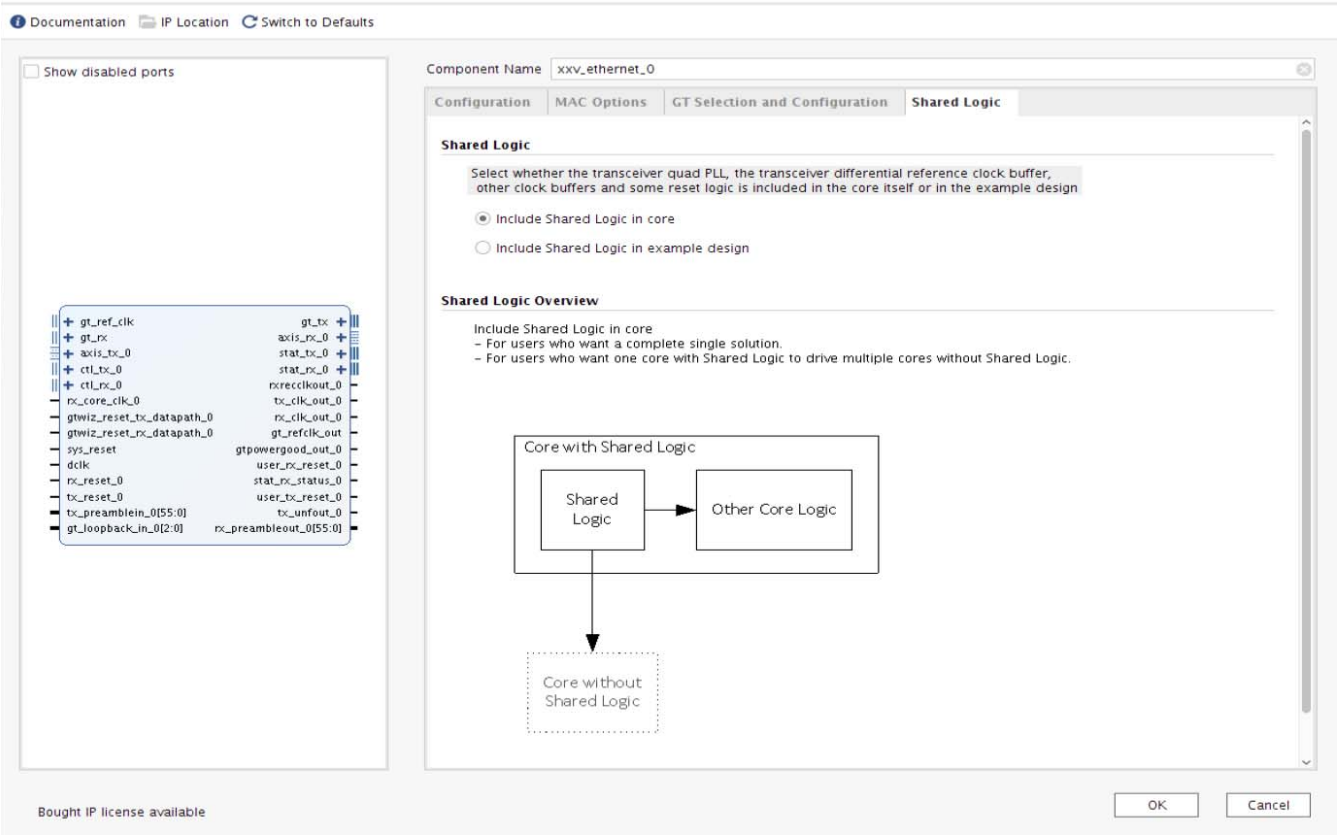

Figure 4-4: **Shared Logic Options**

Table 4-4shows the Shared Logic options of 1G/10G/25G Switching Ethernet Subsystem.

Table 4-4: **Shared Logic Options**

Options	Default
Include Shared Logic in core Include Shared Logic in example design	Include Shared Logic in core

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6].

Constraining the Core

This section contains information about constraining the core in the Vivado Design Suite.

Required Constraints

This section is not applicable for this core.

Device, Package, and Speed Grade Selections

This section is not applicable for this core.

Clock Frequencies

This section is not applicable for this core.

Clock Management

This section is not applicable for this core.

Clock Placement

This section is not applicable for this core.

Banking

This section is not applicable for this core.

Transceiver Placement

This section is not applicable for this core.

I/O Standard and Placement

This section is not applicable for this core.

Simulation

For comprehensive information about Vivado simulation components, as well as information about using supported third-party tools, see the *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 8].

Simulation Speed Up

The example design contains wait timers. A `define SIM_SPEED_UP is available to improve simulation time by speeding up these wait times.

VCS

Use the vlogan option: `+define+SIM_SPEED_UP`

ModelSim

Use the vlog option: `+define+SIM_SPEED_UP`

IES

Use the ncvlog option: `+define+SIM_SPEED_UP`

Vivado Simulator

Use the xvlog option: `-d SIM_SPEED_UP`

Synthesis and Implementation

For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6].

Example Design

This chapter contains information about the example design provided in the Vivado® Design Suite when using the Vivado Integrated Design Environment (IDE).

Overview

Figure 5-1 shows the instantiation of various modules and their hierarchy for a single core configuration of ethernet_1_10_25g_0 example design for 32bit MAC + PCS/PMA core. Clocking helper blocks are used to generate the required clock frequency for the core.

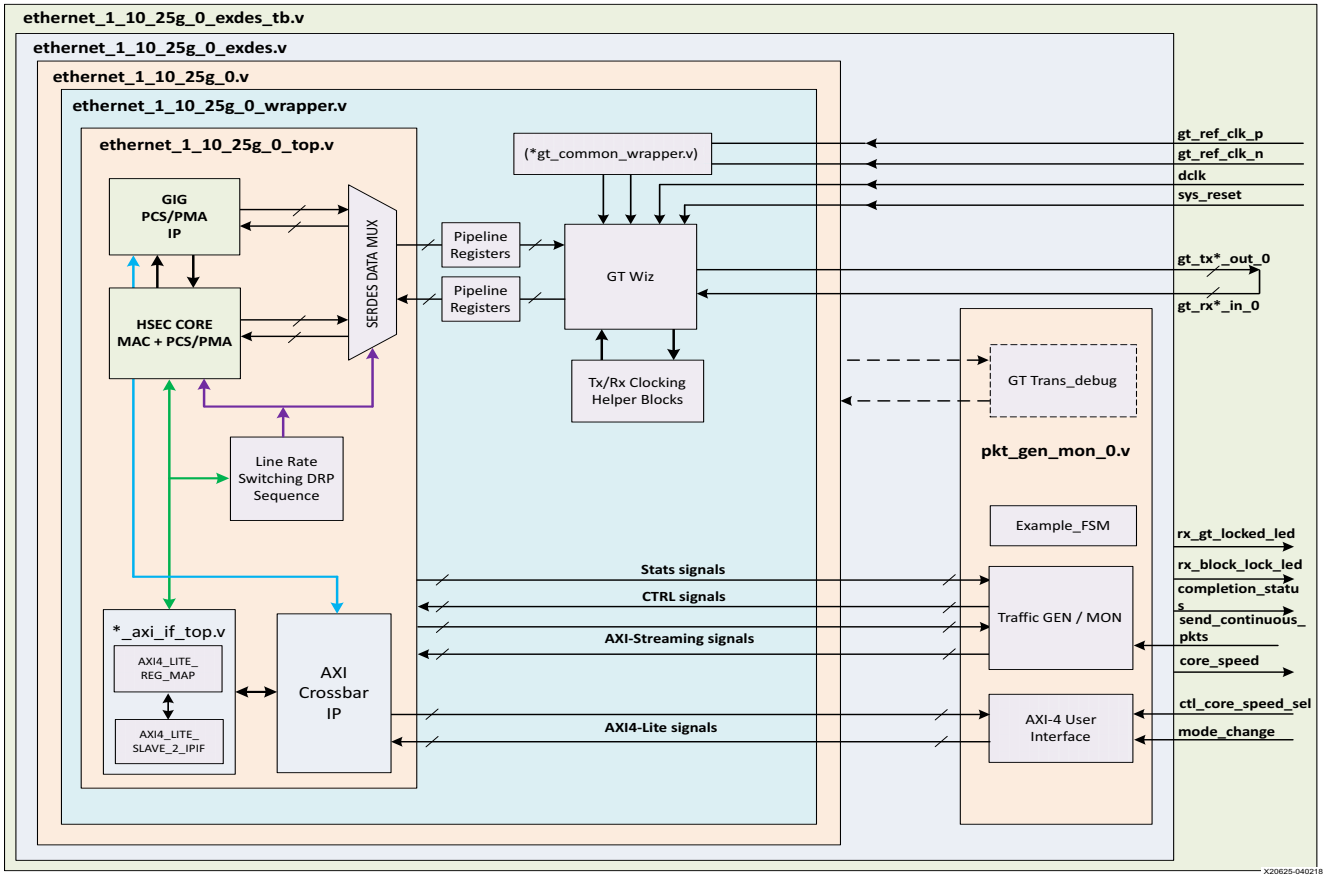

Figure 5-1: **32bit MAC + PCS/PMA Single Core Example Design Hierarchy**

Figure 5-2 shows the instantiation of various modules and their hierarchy for a single core configuration of ethernet_1_10_25g_0 example design for 64bit MAC + PCS/PMA core configuration.

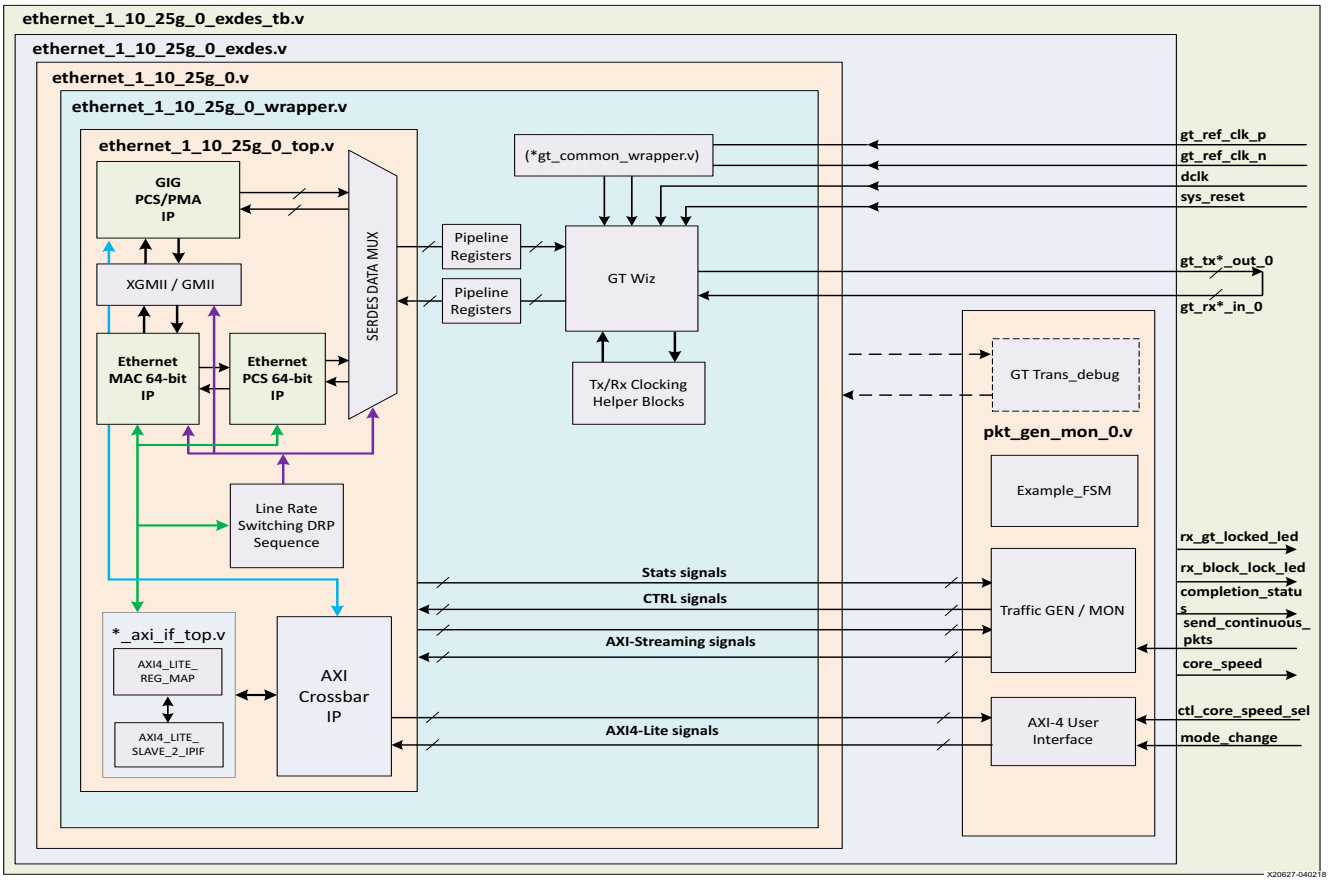

Figure 5-2: **64bit MAC + PCS/PMA Single Core Example Design Hierarchy**

Figure 5-3 shows the instantiation of various modules and their hierarchy for a single core configuration of ethernet_1_10_25g_0 example design for 32bit PCS/PMA core.

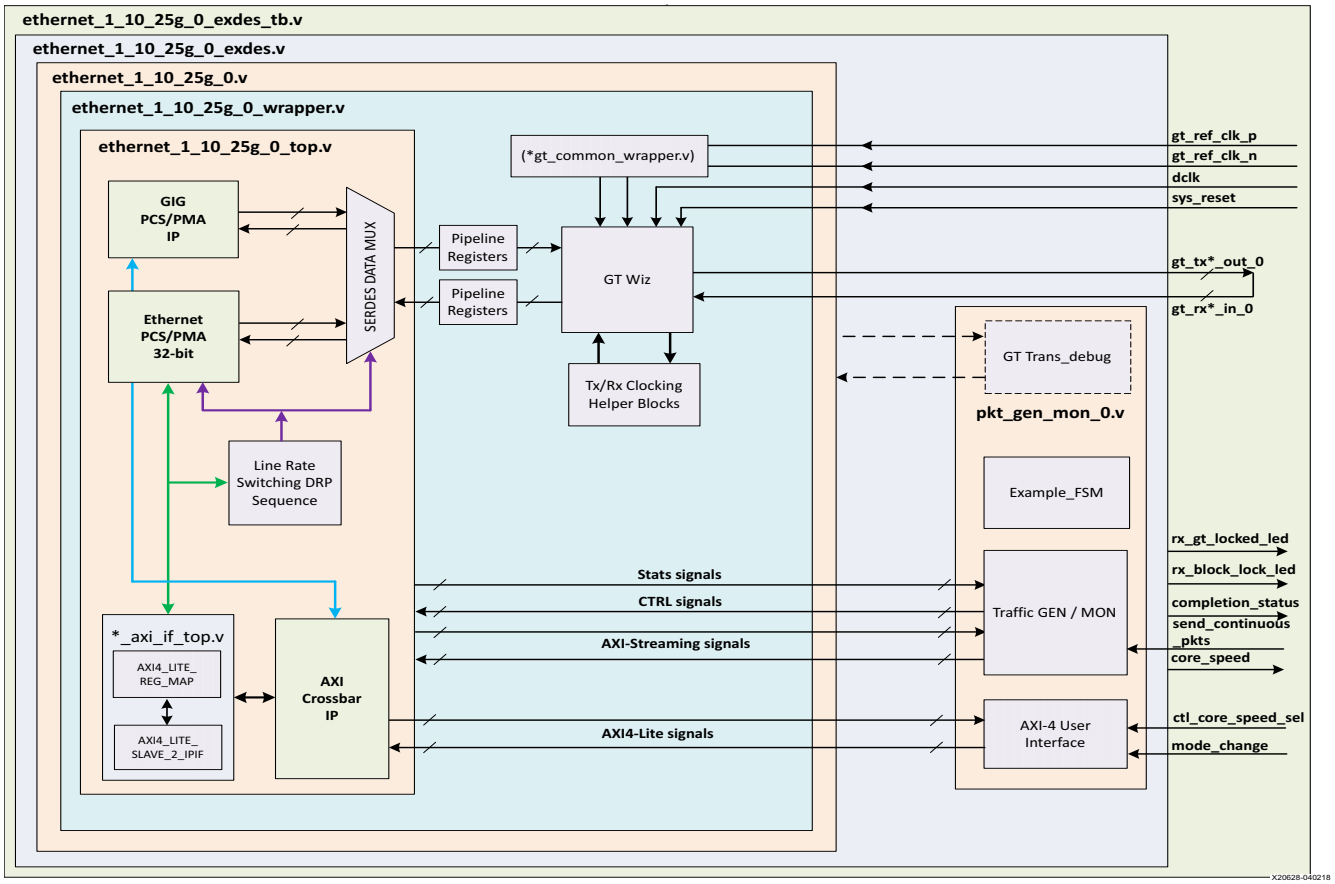

Figure 5-3: **32bit PCS only Single Core Example Design Hierarchy**

Following are the user interfaces available for different configurations.

- MAC/PCS configuration:
  - AXI4-Stream for datapath interface
  - AXI4-Lite for control and statistics interface
- PCS only configuration:
  - XGMII interface
  - GMII interface
  - AXI4-Lite for control and statistics interface

The `ethernet_1_10_25g_0` module is used to generate the data packets for sanity testing. The packet generation and checking is controlled by a FSM module.

The optional modules are described as follows:

- **TX / RX Pipeline register**: The TX Pipeline register double synchronizes the data from the core to the GT with respect to the `tx_clk`. The RX pipeline register double synchronizes the data from the GT to the core with respect to the `rx_serdes_clk`.

Figure 5-4 shows the instantiation of various modules and their hierarchy for the multiple core configuration of ethernet_1_10_25g_0 example design for 32bit MAC + PCS/PMA core.

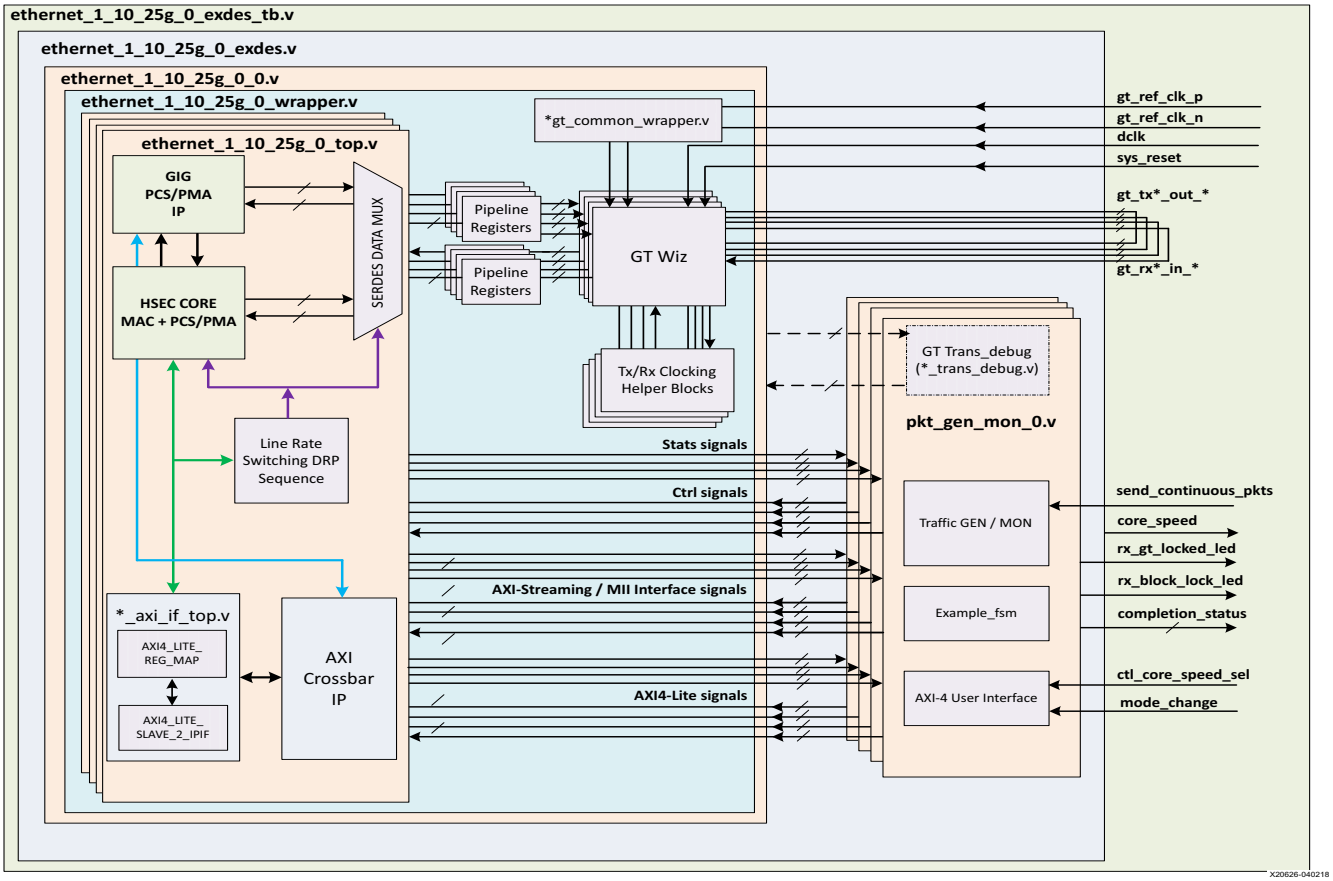

Figure 5-4: **32bit MAC + PCS/PMA Multi-core Example Design Hierarchy**

Figure 5-5 shows the instantiation of various modules and their hierarchy for the multiple core configuration of ethernet_1_10_25g_0 example design for 64bit MAC + PCS/PMA core.

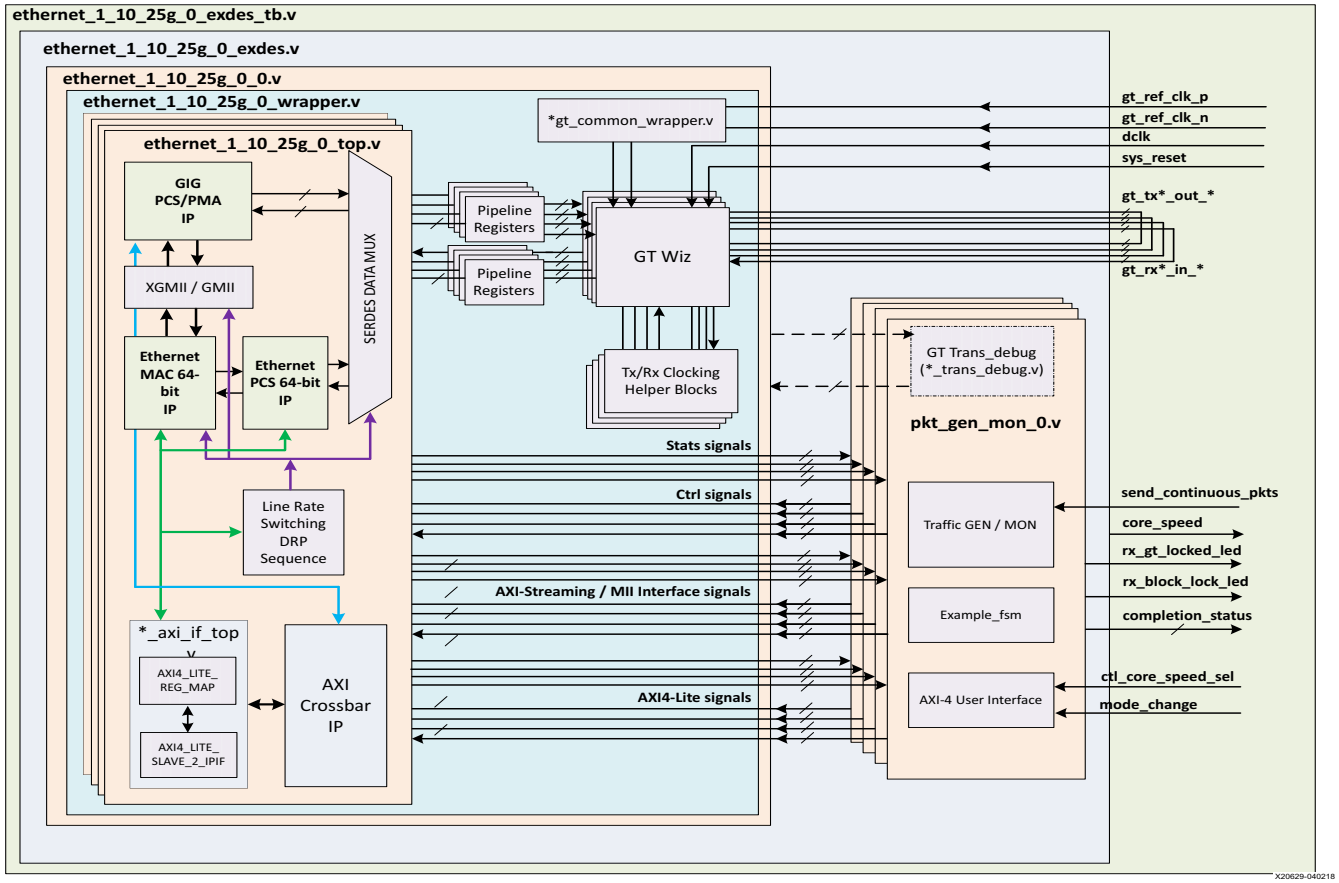

Figure 5-5: **64bit MAC + PCS/PMA Multi-core Example Design Hierarchy**

Figure 5-6 shows the instantiation of various modules and their hierarchy for the multiple core configuration of ethernet_1_10_25g_0 example design for 32bit PCS/PMA core.

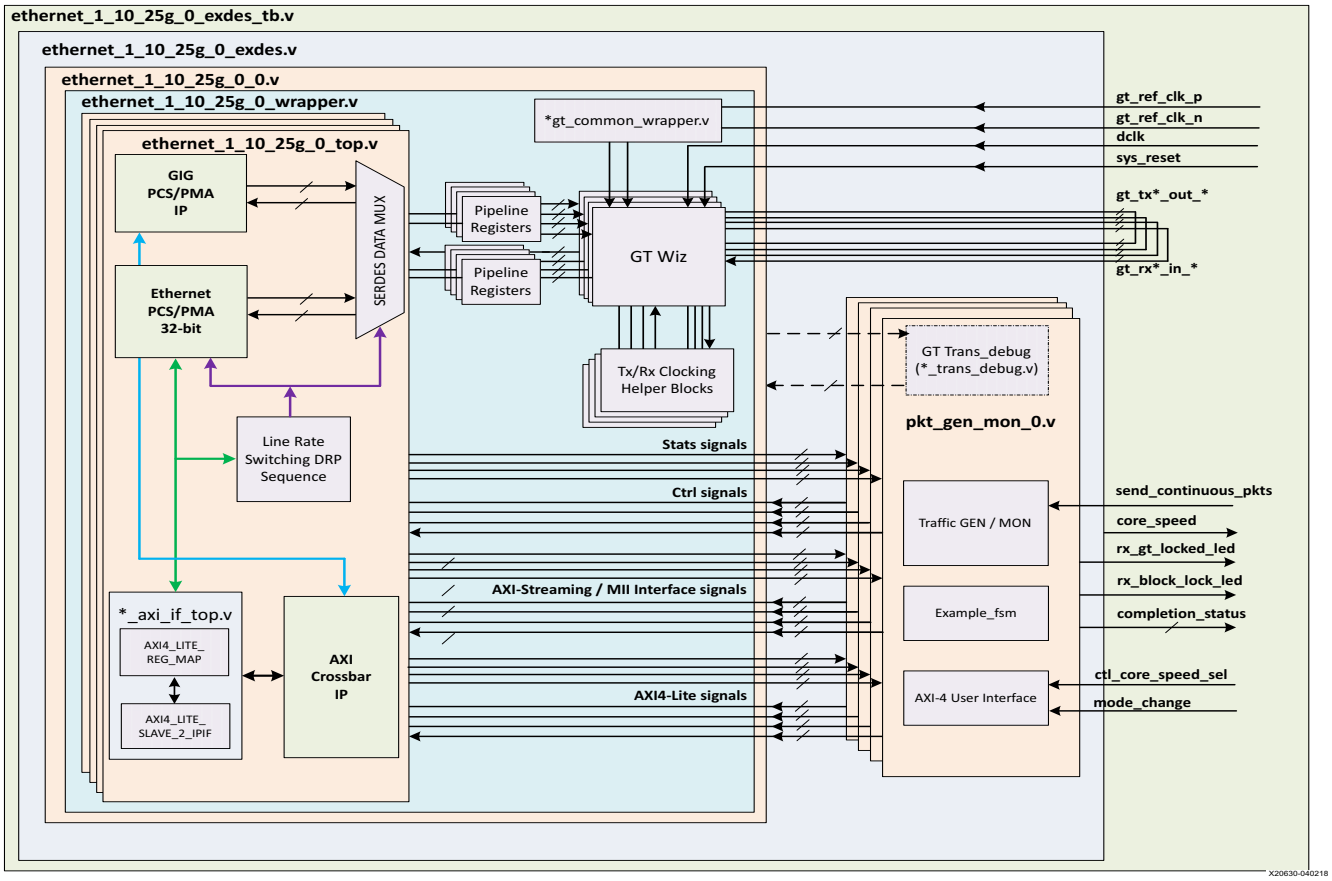

Figure 5-6: **32bit PCS/PMA only Multi-core Example Design Hierarchy**

Example Design Hierarchy (GT in Example Design)

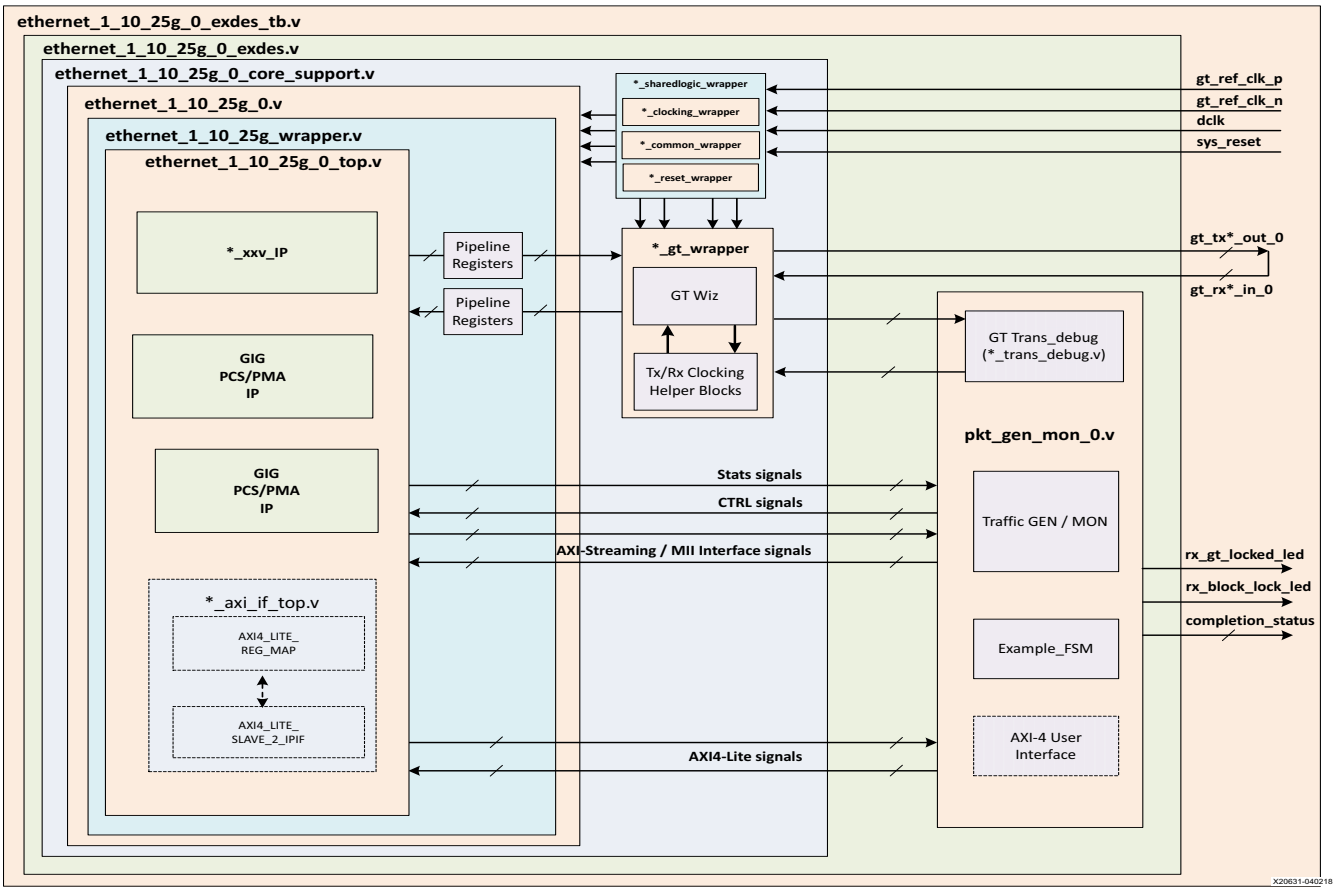

Figure 5-7: **Single Core with GT in Example Design Hierarchy**

Figure 5-7 shows the instantiation of various modules and their hierarchy for a single core configuration of the ethernet_1_10_25g_0 example design when the GT (serial transceiver) is outside the IP Core, that is, in the example design. This hierarchical example design is delivered when you select the **Include GT subcore in example design** option from the GT Selection and Configuration tab.

The `ethernet_1_10_25g_0_core_support.v` is present in the hierarchy when you select the **Include GT subcore in example design** option from the GT Selection and Configuration tab or the **Include Shared Logic in example design** option from the Shared Logic tab. This instantiates the `ethernet_1_10_25g_0_sharedlogic_wrapper.v` module and the `ethernet_1_10_25g_0.v` module for the **Include Shared Logic in example design** option. The `ethernet_1_10_25g_0_gt_wrapper.v` module will be present when you select the **GT subcore in example design** option.

The user interface available for MAC/PCS configuration and PCS configuration configurations is the same as mentioned in the Overview.

The `ethernet_1_10_25g_0.v` module instantiates the necessary the sync registers/ re-timing pipeline registers for the synchronization of data between the core and the GT.

The `ethernet_1_10_25g_0_pkt_gen_mon` module is used to generate the data packets for sanity testing. The packet generation and checking is controlled by a Finite State Machine (FSM) module.

Description of optional modules are as follows:

- ethernet_1_10_25g_0_sharedlogic_wrapper

  This module is present in the example design when you select the **Include GT subcore in example design** option from the GT Selection and Configuration tab or **Include Shared Logic** in the Example Design from the Shared Logic tab. This module brings all modules that can be shared between multiple IP cores and designs outside the IP core.

- ethernet_1_10_25g_0_gt_wrapper

  This module is present in the example design when you select the **Include GT subcore in example** design option from the GT Selection and Configuration tab. This module is having instantiations of the GT along with various helper blocks. The clocking helper blocks are used to generate the required clock frequency for the Core.

Figure 5-8 shows the instantiation of various modules and their hierarchy for the multiple core configuration of the ethernet_1_10_25g_0 example design when the GT is in the example design.

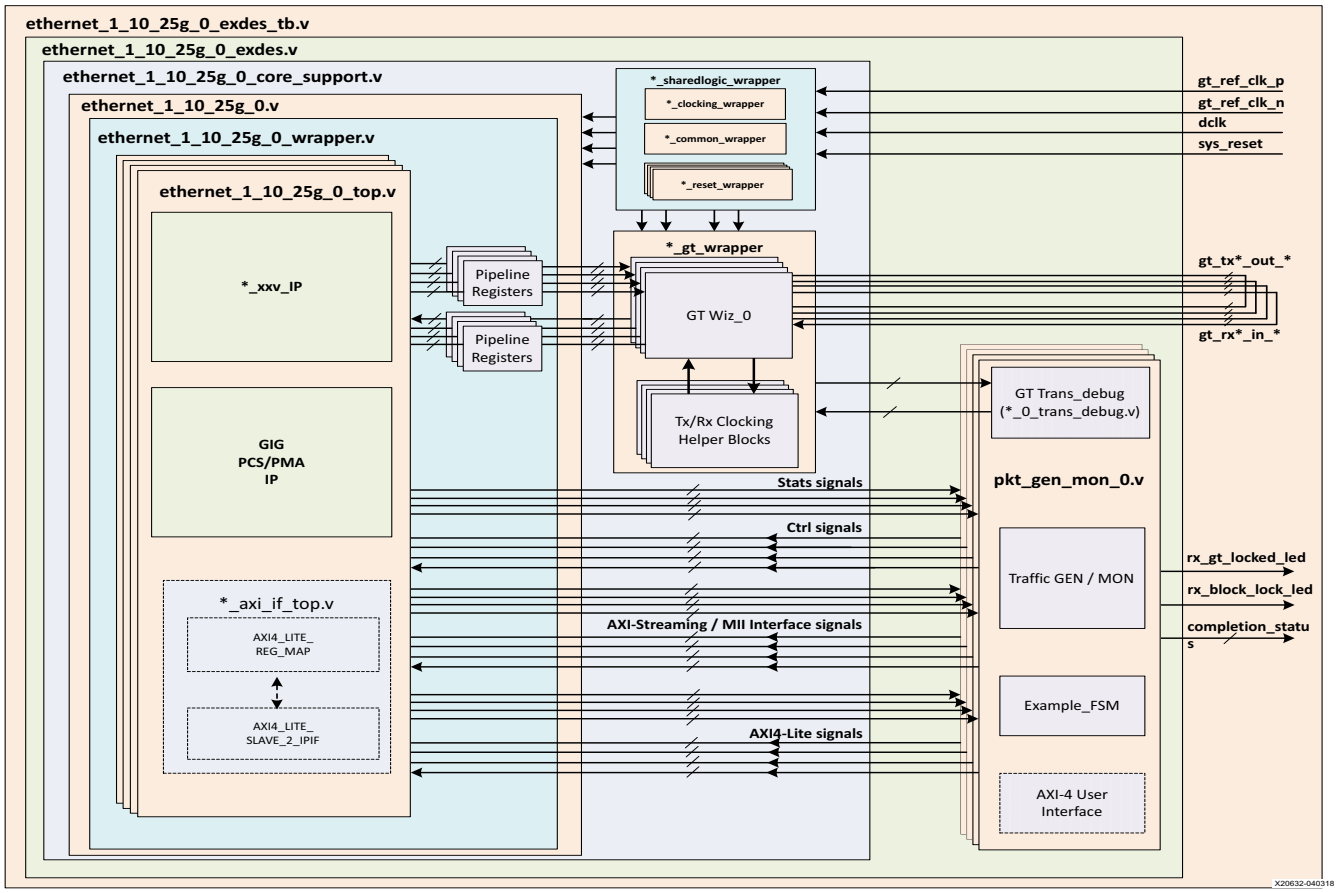

Figure 5-8: **Multiple Core with GT in Example Design Hierarchy**

User Interface

General purpose I/Os (GPIOs) are provided to control the example design. The user input and user output ports are described in Table 5-1.

Table 5-1: **User Input and User Output Ports**

Name	Size	Direction	Description
sys_reset	1	Input	Reset for ethernet_1_10_25g core.
gt_ref_clk_p	1	Input	Differential input clk to GT. This clock frequency should be equal to the GT RefClk frequency mentioned in the Vivado IDE GT Selection and Configuration tab.
gt_ref_clk_n	1	Input	Differential input clk to GT. This clock frequency should be equal to the GT RefClk frequency mentioned in the Vivado IDE GT Selection and Configuration tab.
dclk	1	Input	Stable/free running input clk to GT. This clock frequency should be equal to the GT DRP clock frequency mentioned in the Vivado IDE GT Selection and Configuration tab.
rx_gt_locked_led_0	1	Output	Indicates that GT has been locked.
rx_block_lock_led_0	1	Output	Indicates RX block lock has been achieved.
restart_tx_rx_0	1	Input	This signal is used to restart the packet generation and reception for the data sanity test when the packet generator and the packet monitor are in idle state.

Table 5-1: **User Input and User Output Ports *(Cont'd)***

Name	Size	Direction	Description
completion_status	5	Output	This signal represents the test status/result. • 5'd0 Test did not run. • 5'd1 PASSED 25GE/10GE CORE TEST SUCCESSFULLY COMPLETED • 5'd2 No block lock on any lanes. • 5'd3 Not all lanes achieved block lock. • 5'd4 Some lanes lost block lock after achieving block lock. • 5'd5 No lane sync on any lanes. • 5'd6 Not all lanes achieved sync. • 5'd7 Some lanes lost sync after achieving sync. • 5'd8 No alignment status or rx_status was achieved. • 5'd9 Loss of alignment status or rx_status after both were achieved. • 5'd10 TX timed out. • 5'd11 No TX data was sent. • 5'd12 Number of packets received did not equal the number of packets sent. • 5'd13 Total number of bytes received did not equal the total number of bytes sent. • 5'd14 A protocol error was detected. • 5'd15 Bit errors were detected in the received packets. • 5'd31 Test is stuck in reset.
mode_change_*	1	Input	This is used to switch the core speed.
core_speed_*	2	Output	This output signal indicates the speed with which the core is working: 2'b00 = 25G 2'b01 = 1G 2'b10 = 10G 2'b11 = Reserved
send_continuous_pkts_*	1	Input	Use this port to send continuous packets for board validation. • 1'b0 - Sends fixed 20 packets for simulation • 1'b1- send continuous packets for board.
ctl_core_speed_sel	2	Input	This signal is used to set the operating speed of the core. 2'b00 = 25G 2'b01 = 1G 2'b10 = 10G 2'b11 = Reserved

Core xci Top Level Port List

The top level port list for the core xci with all features enabled is listed below:

In the following table an asterisk (*) represents CORE number, having value 0 to 3.

Example: port_name_*

- port_name_0: for first CORE
- port_name_1: for second CORE (will be present when user selects number of cores >=2)
- port_name_2: for third CORE (will be present when user selects number of cores >=3)
- port_name_3: for fourth CORE (will be present when user selects number of cores =4)

Table 5-2: **Core xci Top Level Port List**

Name	Size	Direction	Description
Common Clock/Reset Signals			
sys_reset	1	Input	Reset for core.
dclk	1	Input	Stable input clk to GT.
gt_refclk_p	1	Input	Differential input clk to GT. ***Note:*** This port is available when the **Include GT subcore in core** option is selected in the GT Selection and Configuration tab and the **Include Shared Logic in core** option is selected in the Shared Logic tab.
gt_refclk_n	1	Input	Differential input clk to GT. ***Note:*** This port is available when the **Include GT subcore in core** option is selected in the GT Selection and Configuration tab and the **Include Shared Logic in core** option is selected in the Shared Logic tab.
tx_clk_out_*	1	Output	TX user clock output from GT. ***Note:*** This port is available when the **Select Core** is Ethernet MAC+PCS/PMA 32/64-bit.
rx_clk_out_*	1	Output	RX user clock output from GT.
tx_mii_clk_*	1	Output	TX mii clock output from GT. ***Note:*** This port is available when Select Core is Ethernet PCS 32-bit.
tx_stats_clk_out_*	1	Output	312.5 MHz / 125 MHz out put clock to be used for the tx statistic.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
rx_stats_clk_out_*	1	Output	312.5 MHz / 125 MHz out put clock to be used for the rx statistics.
tx_reset_*	1	Input	TX reset input to the core.
user_tx_reset_*	1	Output	TX reset output for the user logic.
rx_reset_*	1	Input	RX reset input to the core.
user_rx_reset_*	1	Output	RX reset output for the user logic.
gtpowergood_out_*	1	Output	See the *UltraScale Architecture GTH Transceivers User Guide* (UG576) [Ref 13] for the port description.
Common Transceiver Interface ports			
gt_loopback_in_*	3	Input	GT loopback input signal. Refer to the GT user guide. ***Note:*** This port is available when the **Include GT subcore in core** option is selected in the GT Selection and Configuration tab.
gt_txp_out	1	Output	Differential serial GT TX output ***Note:*** This port is available when the **Include GT subcore in core** option and Board support is selected in the GT Selection and Configuration tab.
gt_txn_out	1	Output	Differential serial GT TX output. ***Note:*** This port is available when the **Include GT subcore in core** option and Board support is selected in the GT Selection and Configuration tab.
gt_rxn_in	1	Input	Differential serial GT RX input. ***Note:*** This port is available when the **Include GT subcore in core** option and Board support is selected in the GT Selection and Configuration tab.
gt_rxp_in	1	Input	Differential serial GT RX input. ***Note:*** This port is available when the **Include GT subcore in core** option and Board support is selected
gt_rxp_in_0	1	Input	Differential serial GT RX input for lane 0. ***Note:*** This port is available when the **Include GT subcore in core** option is selected in the GT Selection and Configuration tab.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
gt_rxn_in_0	1	Input	Differential serial GT RX input for lane 0. ***Note:*** This port is available when the **Include GT subcore in core** option is selected in the GT Selection and Configuration tab.
gt_txp_out_0	1	Output	Differential serial GT TX output for lane 0. ***Note:*** This port is available when the **Include GT subcore in core** option is selected in the GT Selection and Configuration tab.
gt_txn_out_0	1	Output	Differential serial GT TX output for lane 0. ***Note:*** This port is available when the **Include GT subcore in core** option is selected in the GT Selection and Configuration tab.
Transceiver control and status debug ports Ports under this section will be available when the **Include GT subcore in core** option is selected in the GT Selection and Configuration tab or **Enable Additional GT Control/Status and DRP Ports** is selected from the GT Selection and Configuration tab.			
gt_dmonitorout_*	16	Output	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_eyescandataerror_*	1	Output	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_eyescanreset_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_eyescantrigger_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_pcsrsvdin_*	16	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
gt_rxbufreset_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description. ***Note:*** Port width: 2-bit for 50G single core and 4-bits for 40G.
gt_rxbufstatus_*	3	Output	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxcdrhold_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxcommadeten_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxdfeagchold_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxdfelpmreset_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxlatclk_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxlpmen_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxpcsreset_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
gt_rxpmareset_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxpolarity_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxprbscntreset_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxprbserr_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxprbssel_*	4	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxrate_*	3	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxslide_in_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_rxstartofseq_*	2	Output	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_txbufstatus_*	2	Output	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
gt_txdiffctrl_*	5	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_txinhibit_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_txlatclk_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_txmaincursor_*	7	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_txpcsreset_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_txpmareset_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_txpolarity_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_txpostcursor_*	5	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_txprbsforceerr_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
gt_txprbssel_*	4	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_txprecursor_*	5	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gtwiz_reset_tx_datapath_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gtwiz_reset_rx_datapath_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_ch_drpclk_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_ch_drpdo_*	16	Output	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_ch_drprdy_*	1	Output	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_ch_drpen_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_ch_drpwe_*	1	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
gt_ch_drpaddr_*	10	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.
gt_ch_drpdi_*	16	Input	See *UltraScale Architecture GTH Transceivers User Guide* (UG576)[Ref 13]/ *UltraScale Architecture GTY Transceivers User Guide* (UG578)[Ref 14] for the port description.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
AXI4-Lite Interface Ports Ports under this section will be available when the **Include AXI4-Lite** is selected from the Configuration tab.			
s_axi_aclk_*	1	Input	AXI clock signal
s_axi_aresetn_*	1	Input	AXI reset signal
pm_tick_*	1	Input	PM tick user input
s_axi_awaddr_*	32	Input	AXI write address
s_axi_awvalid_*	1	Input	AXI write address valid
s_axi_awready_*	1	Output	AXI write address ready
s_axi_wdata_*	32	Input	AXI write data
s_axi_wstrb_*	4	Input	AXI write strobe. This signal indicates which byte lanes hold valid data.
s_axi_wvalid_*	1	Input	AXI write data valid. This signal indicates that valid write data and strobes are available.
s_axi_wready_*	1	Output	AXI write data ready
s_axi_bresp_*	2	Output	AXI write response. This signal indicates the status of the write transaction. 'b00 = OKAY 'b01 = EXOKAY 'b10 = SLVERR 'b11 = DECERR
s_axi_bvalid_*	1	Output	AXI write response valid. This signal indicates that the channel is signaling a valid write response.
s_axi_bready_*	1	Input	AXI write response ready.
s_axi_araddr_*	32	Input	AXI read address
s_axi_arvalid_*	1	Input	AXI read address valid
s_axi_arready_*	1	Output	AXI read address ready
s_axi_rdata_*	32	Output	AXI read data issued by slave

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
s_axi_rresp_*	2	Output	AXI read response. This signal indicates the status of the read transfer. 'b00 = OKAY 'b01 = EXOKAY 'b10 = SLVERR 'b11 = DECERR
s_axi_rvalid_*	1	Output	AXI read data valid
s_axi_rready_*	1	Input	AXI read ready. This signal indicates the user/master can accept the read data and response information.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
AXI4-Stream User Interface Signals Ports under this section will be available when **Ethernet MAC+PCS/PMA-32 bit** is selected from the Configuration tab.			
tx_unfout_*	1	Output	Underflow signal for TX data path from core. If tx_unfout_* is sampled as 1, a violation has occurred meaning the current packet is corrupted. Error control blocks are transmitted as long as the underflow condition persists. It is up to the user logic to ensure a complete packet is input to the core without under-running the TX data path interface.
tx_axis_tready_*	1	Output	TX path ready signal from core.
tx_axis_tvalid_*	1	Input	Transmit AXI4-Stream Data valid.
tx_axis_tdata_*	32	Input	Transmit AXI4-Stream Data bus.
tx_axis_tlast_*	1	Input	Transmit AXI4-Stream tlast.
tx_axis_tkeep_*	8/4	Input	Transmit AXI4-Stream tkeep.
tx_axis_tuser_*	1	Input	Transmit AXI4-Stream tuser.
tx_preamblein_*	56	Input	Transmit AXI4-Stream preamble.
rx_axis_tvalid_*	1	Output	Receive AXI4-Stream Data valid.
rx_axis_tdata_*	32	Output	Receive AXI4-Stream Data bus.
rx_axis_tlast_*	1	Input	Receive AXI4-Stream tlast.
rx_axis_tkeep_*	8/4	Input	Receive AXI4-Stream tkeep.
rx_axis_tuser_*	1	Input	Receive AXI4-Stream tuser.
rx_preamblein_*	56	Input	Receive AXI4-Stream preamble.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
TX Path Control / Status / Statistics Signals			
ctl_tx_send_rfi_*	1	Input	Transmit Remote Fault Indication (RFI) code word. If this input is sampled as a 1, the TX path only transmits Remote Fault code words. This input should be set to 1 until the RX path is fully aligned and is ready to accept data from the link partner.
ctl_tx_send_lfi_*	1	Input	Transmit Local Fault Indication (LFI) code word. Takes precedence over RFI.
ctl_tx_send_idle_*	1	Input	Transmit Idle code words. If this input is sampled as a 1, the TX path only transmits Idle code words. This input should be set to 1 when the partner device is sending Remote Fault Indication (RFI) code words.
stat_tx_local_fault_*	1	Output	A value of 1 indicates the receive decoder state machine is in the TX_INIT state. This output is level sensitive.
stat_tx_total_bytes_*	5	Output	Increment for the total number of bytes transmitted.
stat_tx_total_packets_*	1	Output	Increment for the total number of packets transmitted.
stat_tx_total_good_bytes_*	14	Output	Increment for the total number of good bytes transmitted. This value is only non-zero when a packet is transmitted completely and contains no errors.
stat_tx_total_good_packets_*	1	Output	Increment for the total number of good packets transmitted.
stat_tx_bad_fcs_*	1	Output	Increment for packets greater than 64 bytes that have FCS errors.
stat_tx_packet_64_bytes_*	1	Output	Increment for good and bad packets transmitted that contain 64 bytes.
stat_tx_packet_65_127_bytes_*	1	Output	Increment for good and bad packets transmitted that contain 65 to 127 bytes.
stat_tx_packet_128_255_bytes_*	1	Output	Increment for good and bad packets transmitted that contain 128 to 255 bytes.
stat_tx_packet_256_511_bytes_*	1	Output	Increment for good and bad packets transmitted that contain 256 to 511 bytes.
stat_tx_packet_512_1023_bytes_*	1	Output	Increment for good and bad packets transmitted that contain 512 to 1,023 bytes.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
stat_tx_packet_1024_1518_bytes_*	1	Output	Increment for good and bad packets transmitted that contain 1,024 to 1,518 bytes.
stat_tx_packet_1519_1522_bytes_*	1	Output	Increment for good and bad packets transmitted that contain 1,519 to 1,522 bytes.
stat_tx_packet_1523_1548_bytes_*	1	Output	Increment for good and bad packets transmitted that contain 1,523 to 1,548 bytes.
stat_tx_packet_1549_2047_bytes_*	1	Output	Increment for good and bad packets transmitted that contain 1,549 to 2,047 bytes.
stat_tx_packet_2048_4095_bytes_*	1	Output	Increment for good and bad packets transmitted that contain 2,048 to 4,095 bytes.
stat_tx_packet_4096_8191_bytes_*	1	Output	Increment for good and bad packets transmitted that contain 4,096 to 8,191 bytes.
stat_tx_packet_8192_9215_bytes_*	1	Output	Increment for good and bad packets transmitted that contain 8,192 to 9,215 bytes.
stat_tx_packet_small_*	1	Output	Increment for all packets that are less than 64 bytes long. Packets that are less than 64 bytes are not transmitted.
stat_tx_packet_large_*	1	Output	Increment for all packets that are more than 9,215 bytes long.
stat_tx_frame_error_*	1	Output	Increment for packets with tx_errin set to indicate an EOP abort.
gig_ethernet_pcs_pma_status_vector_*	16	Output	See Status Vector Table in *1G/2.5G Ethernet PCS/PMA or SGMII LogiCORE IP Product Guide* (PG047)[Ref 3]
stat_core_speed_*	2	Output	Indicates the operating core speed • 2'b10: Core configured in 10G mode • 2'b01: Core configured in 1G mode
gpcs_resetdone_*	1	Output	Indicates the 1G core is out-of reset
gt_drp_busy_*	1	Output	Indicates the GT DRP operation are in progress.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
RX Path Control / Status / Statistics Signals			
ctl_rx_data_pattern_select_*			Corresponds to MDIO register bit 3.42.0 as defined in Clause 45. ***Note:*** This port is available when **Include AXI4-Lite** is not selected in the Configuration tab and **Select Core** is **Ethernet MAC+PCS/ PMA-32 bit** and the **Include FIFO Logic** is disabled.
ctl_rx_test_pattern_enable_*			Test pattern enable for the RX core. A value of 1 enables test mode. Corresponds to MDIO register bit 3.42.2 as defined in Clause 45. Takes second precedence. ***Note:*** This port is available when **Include AXI4-Lite** is not selected in the Configuration tab and **Select Core** is **Ethernet MAC+PCS/ PMA-32-bit** and the **Include FIFO Logic** is disabled.
stat_rx_block_lock_*	1	Output	Block lock status for each PCS lane. A value of 1 indicates that the corresponding lane has achieved block lock as defined in Clause 82. Corresponds to MDIO register bit 3.50.7:0 and 3.51.11:0 as defined in Clause 82.3. This output is level sensitive.
stat_rx_framing_err_valid_*	1	Output	Valid indicator for stat_rx_framing_err. When 1 stat_rx_framing_err_0 is valid.
stat_rx_framing_err_*	1	Output	RX sync header bits framing error. Each PCS Lane has a four-bit bus that indicates how many sync header errors were received for that PCS Lane. The value of the bus is only valid when the corresponding stat_rx_framing_err_valid is a 1. The values on these buses can be updated at any time and are intended to be used as increment values for sync header error counters.
stat_rx_hi_ber_*	1	Output	High Bit Error Rate (BER) indicator. When set to 1, the BER is too high as defined by IEEE Std 802.3-2015. Corresponds to MDIO register bit 3.32.1 as defined in Clause 82.3. This output is level sensitive.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
stat_rx_bad_code_*	1	Output	Increment for 64B/66B code violations. This signal indicates that the RX PCS receive state machine is in the RX_E state as specified by the IEEE Std 802.3-2015. This output can be used to generate MDIO register 3.33:7:0 as defined in Clause 82.3.
stat_rx_error_valid_*	1	Output	Indicates when stat_rx_error is valid.
stat_rx_total_packets_*	2	Output	Increment for the total number of packets received.
stat_rx_total_good_packets_*	1	Output	Increment for the total number of good packets received. This value is only non-zero when a packet is received completely and contains no errors.
stat_rx_total_bytes_*	6	Output	Increment for the total number of bytes received.
stat_rx_total_good_bytes_*	14	Output	Increment for the total number of good bytes received. This value is only non-zero when a packet is received completely and contains no errors.
stat_rx_packet_small_*	2	Output	Increment for all packets that are less than 64 bytes long. Packets that are less than 4 bytes are dropped.
stat_rx_jabber_*	1	Output	Increment for packets longer than ctl_rx_max_packet_len with bad FCS.
stat_rx_packet_large_*	1	Output	Increment for all packets that are more than 9,215 bytes long.
stat_rx_oversize_*	1	Output	Increment for packets longer than ctl_rx_max_packet_len with good FCS.
stat_rx_undersize_*	2	Output	Increment for packets shorter than stat_rx_min_packet_len with good FCS.
stat_rx_toolong_*	1	Output	Increment for packets longer than ctl_rx_max_packet_len with good and bad FCS.
stat_rx_fragment_*	2	Output	Increment for packets shorter than stat_rx_min_packet_len with bad FCS.
stat_rx_packet_64_bytes_*	1	Output	Increment for good and bad packets received that contain 64 bytes.
stat_rx_packet_65_127_bytes_*	1	Output	Increment for good and bad packets received that contain 65 to 127 bytes.
stat_rx_packet_128_255_bytes_*	1	Output	Increment for good and bad packets received that contain 128 to 255 bytes.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
stat_rx_packet_256_511_bytes_*	1	Output	Increment for good and bad packets received that contain 256 to 511 bytes.
stat_rx_packet_512_1023_bytes_*	1	Output	Increment for good and bad packets received that contain 512 to 1,023 bytes.
stat_rx_packet_1024_1518_bytes_*	1	Output	Increment for good and bad packets received that contain 1,024 to 1,518 bytes.
stat_rx_packet_1519_1522_bytes_*	1	Output	Increment for good and bad packets received that contain 1,519 to 1,522 bytes.
stat_rx_packet_1523_1548_bytes_*	1	Output	Increment for good and bad packets received that contain 1,523 to 1,548 bytes.
stat_rx_packet_1549_2047_bytes_*	1	Output	Increment for good and bad packets received that contain 1,549 to 2,047 bytes.
stat_rx_packet_2048_4095_bytes_*	1	Output	Increment for good and bad packets received that contain 2,048 to 4,095 bytes.
stat_rx_packet_4096_8191_bytes_*	1	Output	Increment for good and bad packets received that contain 4,096 to 8,191 bytes.
stat_rx_packet_8192_9215_bytes_*	1	Output	Increment for good and bad packets received that contain 8,192 to 9,215 bytes.
stat_rx_bad_fcs_*	2	Output	Bad FCS indicator. The value on this bus indicates packets received with a bad FCS, but not a stomped FCS. A stomped FCS is defined as the bitwise inverse of the expected good FCS. This output is pulsed for one clock cycle to indicate an error condition. Pulses can occur in back-to-back cycles.
stat_rx_packet_bad_fcs_*	1	Output	Increment for packets between 64 and ctl_rx_max_packet_len bytes that have FCS errors.
stat_rx_stomped_fcs_*	2	Output	Stomped FCS indicator. The value on this bus indicates packets were received with a stomped FCS. A stomped FCS is defined as the bitwise inverse of the expected good FCS. This output is pulsed for one clock cycle to indicate the stomped condition. Pulses can occur in back-to-back cycles.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
stat_rx_bad_preamble_*	1	Output	Increment bad preamble. This signal indicates if the Ethernet packet received was preceded by a valid preamble. A value of 1 indicates that an invalid preamble was received.
stat_rx_bad_sfd_*	1	Output	Increment bad SFD. This signal indicates if the Ethernet packet received was preceded by a valid SFD. A value of 1 indicates that an invalid SFD was received.
stat_rx_got_signal_os_*	1	Output	Signal OS indication. If this bit is sampled as a 1, it indicates that a Signal OS word was received. ***Note:*** Signal OS should not be received in an Ethernet network.
stat_rx_test_pattern_mismatch_*	2	Output	Test pattern mismatch increment. A nonzero value in any cycle indicates how many mismatches occurred for the test pattern in the RX core. This output is only active when ctl_rx_test_pattern is set to a 1. This output can be used to generate MDIO register 3.43.15:0 as defined in Clause 82.3. This output is pulsed for one clock cycle.
stat_rx_truncated_*	1	Output	Packet truncation indicator. A value of 1 indicates that the current packet in flight is truncated due to its length exceeding ctl_rx_max_packet_len[14:0]. This output is pulsed for one clock cycle to indicate the truncated condition. Pulses can occur in back-to-back cycles.
stat_rx_local_fault_*	1	Output	This output is High when stat_rx_internal_local_fault or stat_rx_received_local_fault is asserted. This output is level sensitive.
stat_rx_remote_fault_*	1	Output	Remote fault indication status. If this bit is sampled as a 1, it indicates a remote fault condition was detected. If this bit is sampled as a 0, a remote fault condition does not exist. This output is level sensitive.
stat_rx_internal_local_fault_*	1	Output	This signal goes High when an internal local fault is generated due to any one of the following: test pattern generation, bad lane alignment, or high bit error rate. This signal remains High as long as the fault condition persists.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
stat_rx_received_local_fault_*	1	Output	This signal goes High when enough local fault words are received from the link partner to trigger a fault condition as specified by the IEEE fault state machine. This signal remains High as long as the fault condition persists.
stat_rx_valid_ctrl_code_*	1	Output	Indicates that a PCS block with a valid control code was received.
stat_rx_status_*	1	Output	Indicates the link status.
IEEE 1588 TX/RX Interface Control / Status / Statistics Signals Ports under this section will be available when **Enable_Time_Stamping** is selected from the MAC Options tab.			
ctl_tx_systemtimerin_*	80	Input	System timer input for the TX. In normal clock mode, the time format is according to the IEEE 1588 format, with 48 bits for seconds and 32 bits for nanoseconds. In transparent clock mode, bit 63 is expected to be zero, bits 62:16 carry nanoseconds, and bits 15:0 carry fractional nanoseconds. Refer to IEEE 1588v2 for the representational definitions. This input must be in the TX clock domain.
ctl_rx_systemtimerin_*	80	Input	System timer input for the RX. In normal clock mode, the time format is according to the IEEE 1588 format, with 48 bits for seconds and 32 bits for nanoseconds. In transparent clock mode, bit 63 is expected to be zero, bits 62:16 carry nanoseconds, and bits 15:0 carry fractional nanoseconds. Refer to IEEE 1588v2 for the representational definitions. This input must be in the same clock domain as the lane 0 RX SerDes.
stat_tx_ptp_fifo_read_error_*	1	Output	Transmit PTP FIFO write error. A value of 1 on this status indicates that an error occurred during the PTP Tag write. A TX Path reset is required to clear the error.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
stat_tx_ptp_fifo_write_error_*	1	Output	Transmit PTP FIFO read error. A value of 1 on this status indicates that an error occurred during the PTP Tag read. A TX Path reset is required to clear the error.
tx_ptp_1588op_in_*	2	Input	2'b00 – "No operation": no timestamp will be taken and the frame will not be modified. 2'b01 – "1-step": a timestamp should be taken and inserted into the frame. 2'b10 – "2-step": a timestamp should be taken and returned to the client using the additional ports of 2-step operation. The frame itself will not be modified. 2'b11 – Reserved: act as "No operation".
tx_ptp_tag_field_in_*	16	Input	The usage of this field is dependent on the 1588 operation • For "No operation", this field will be ignored. • For "1-step" and "2-step" this field is a tag field. This tag value will be returned to the client with the timestamp for the current frame using the additional ports of 2-step operation. This tag value can be used by software to ensure that the timestamp can be matched with the PTP frame that it sent for transmission.
tx_ptp_tstamp_valid_out_*	1	Output	This bit indicates that a valid timestamp is being presented on the tx.
tx_ptp_tstamp_tag_out_*	16	Output	Tag output corresponding to tx_ptp_tag_field_in[15:0]
tx_ptp_tstamp_out_*	80	Output	Time stamp for the transmitted packet SOP corresponding to the time at which it passed the capture plane. The representation of the bits contained in this bus is the same as the timer input.

Table 5-2: **Core xci Top Level Port List *(Cont'd)***

Name	Size	Direction	Description
rx_ptp_tstamp_valid_out_*	1	Output	This bit indicates that a valid timestamp is being presented on the rx. ***Note:*** This will be present only when core is Ethernet MAC+PCS/PMA-32/64-bit.
rx_ptp_tstamp_out_*	80	Output	Time stamp for the received packet SOP corresponding to the time at which it passed the capture plane. Note that this signal will be valid starting at the same clock cycle during which the SOP is asserted for one of the segments. The representation of the bits contained in this bus is the same as the timer input.

Duplex Mode of Operation

In this mode of operation, both the transmitter and receiver of the core are active and loopback is provided at the GT output interface, that is, output is fed back as input. Packet generation and monitor modules are active in this mode. The generator module is responsible for generating the desired number of packets and transmit to the core using the available data interface. The monitor module checks the packets from the receiver.

Figure 5-9 shows the duplex mode of operation.

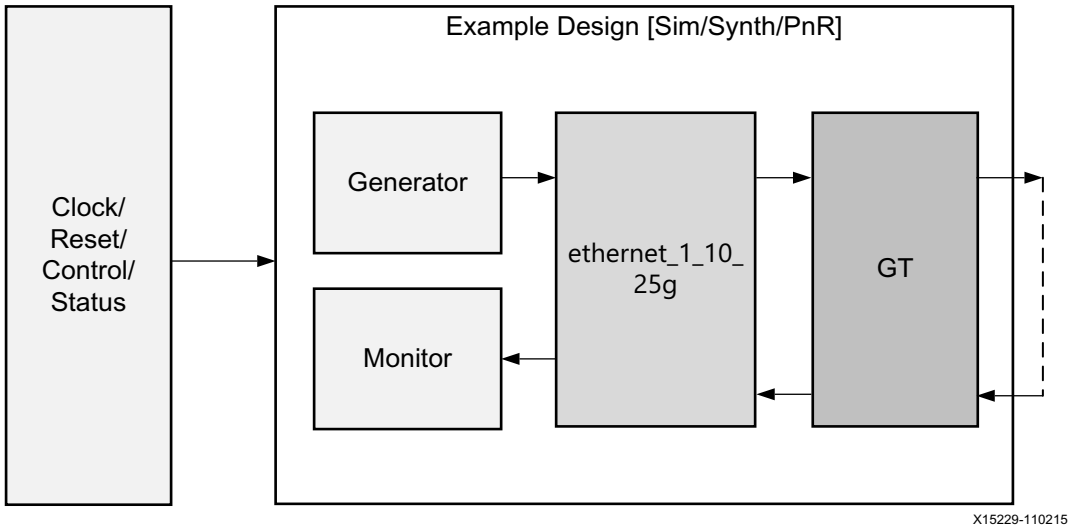

Figure 5-9: **Duplex Mode of Operation**

AXI4-Lite Interface Implementation

In order to instantiate the AXI4-Lite interface to access the control and status registers of the ethernet_1_10_25g core, enable the **Include AXI4-Lite** check box in the Configuration Tab of the Vivado IDE. This option enables the `ethernet_1_10_25g_axi_if_top` module (which contains `ethernet_1_10_25g_pif_registers` with the `ethernet_1_10_25g_slave_2_ipif` module). You can access the AXI4-Lite interface logic registers (control, status and statistics) from the `ethernet_1_10_25g_pkt_gen_mon` module.

This mode enables the following features:

- You can configure all the control (CTL) ports of the core through the AXI4-Lite interface. This operation is performed by writing to a set of address locations with the required data to the register map interface.
- You can access all the status and statistics registers from the core through the AXI4-Lite interface. This operation is performed by reading the address locations for the status and statistics registers through register map.

AXI4 Interface User Logic

The following sections provide the AXI4-Lite interface state machine control and ports.

User State Machine

The read and write through the AXI4-Lite slave module interface is controlled by a state machine as shown below:

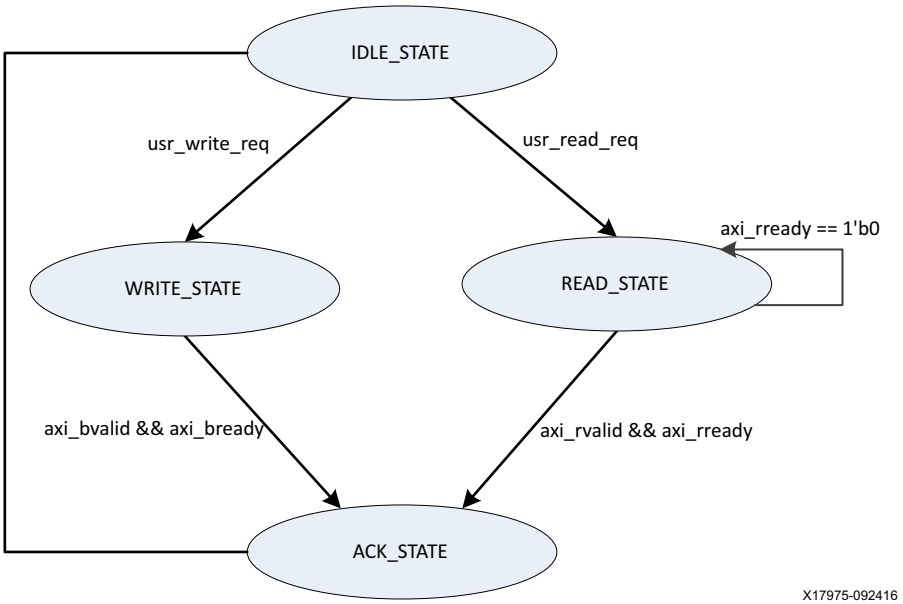

Figure 5-10: **User State Machine for AXI4-Lite Interface**

A functional description of each state is described as below:

- **IDLE_STATE**: By default the FSM will be in IDLE_STATE. When the `user_read_req` signal becomes High, then it moves to the READ_STATE else if the `user_write_req` signal is High, it moves to WRITE_STATE else it remains in IDLE_STATE.
- **WRITE_STATE**: You provide `S_AXI_AWVALID`, `S_AXI_AWADDR`, `S_AXI_WVALID`, `S_AXI_WDATA` and `S_AXI_WSTRB` in this state to write to the register map through AXI. When `S_AXI_BVALID` and `S_AXI_BREADY` from the AXI slave are High then it moves to ACK_STATE. If any write operation happens in any illegal addresses, the `S_AXI_BRESP[1:0]` indicates `2'b10` that asserts the write error signal.
- **READ_STATE**: You provide `S_AXI_ARVALID` and `S_AXI_ARADDR` in this state to read from the register map through AXI. When `S_AXI_RVALID` and `S_AXI_RREADY` are High then it moves to ACK_STATE. If any read operation happens from any illegal addresses, the `S_AXI_RRESP[1:0]` indicates `2'b10` that asserts the read error signal.
- **ACK_STATE**: The state moves to IDLE_STATE.

AXI User Interface Ports

Table 5-3: **AXI User Interface Ports**

Name	Size	Direction	Description
S_AXI_ACLK	1	Input	AXI clock signal
S_AXI_ARESETN	1	Input	AXI active-Low synchronous reset

Table 5-3: **AXI User Interface Ports *(Cont'd)***

Name	Size	Direction	Description
S_AXI_PM_TICK	1	Input	PM tick user input
S_AXI_AWADDR	32	Input	AXI write address
S_AXI_AWVALID	1	Input	AXI write address valid
S_AXI_AWREADY	1	Output	AXI write address ready
S_AXI_WDATA	32	Input	AXI write data
S_AXI_WSTRB	4	Input	AXI write strobe. This signal indicates which byte lanes hold valid data.
S_AXI_WVALID	1	Input	AXI write data valid. This signal indicates that valid write data and strobes are available.
S_AXI_WREADY	1	Output	AXI write data ready
S_AXI_BRESP	2	Output	AXI write response. This signal indicates the status of the write transaction. 'b00 = OKAY 'b01 = EXOKAY 'b10 = SLVERR 'b11 = DECERR
S_AXI_BVALID	1	Output	AXI write response valid. This signal indicates that the channel is signaling a valid write response.
S_AXI_BREADY	1	Input	AXI write response ready.
S_AXI_ARADDR	32	Input	AXI read address
S_AXI_ARVALID	1	Input	AXI read address valid
S_AXI_ARREADY	1	Output	AXI read address ready
S_AXI_RDATA	32	Output	AXI read data issued by slave
S_AXI_RRESP	2	Output	AXI read response. This signal indicates the status of the read transfer. 'b00 = OKAY 'b01 = EXOKAY 'b10 = SLVERR 'b11 = DECERR
S_AXI_RVALID	1	Output	AXI read data valid
S_AXI_RREADY	1	Input	AXI read ready. This signal indicates the user/master can accept the read data and response information.

Valid Write Transactions

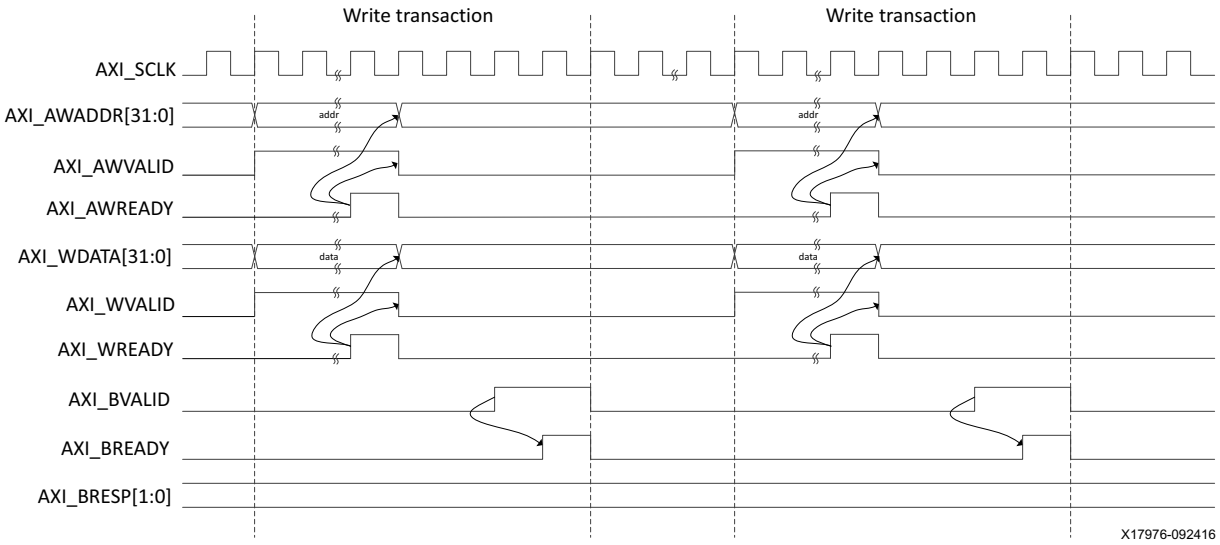

Figure 5-11: **AXI4-Lite User Side Write Transaction**

Invalid Write Transactions

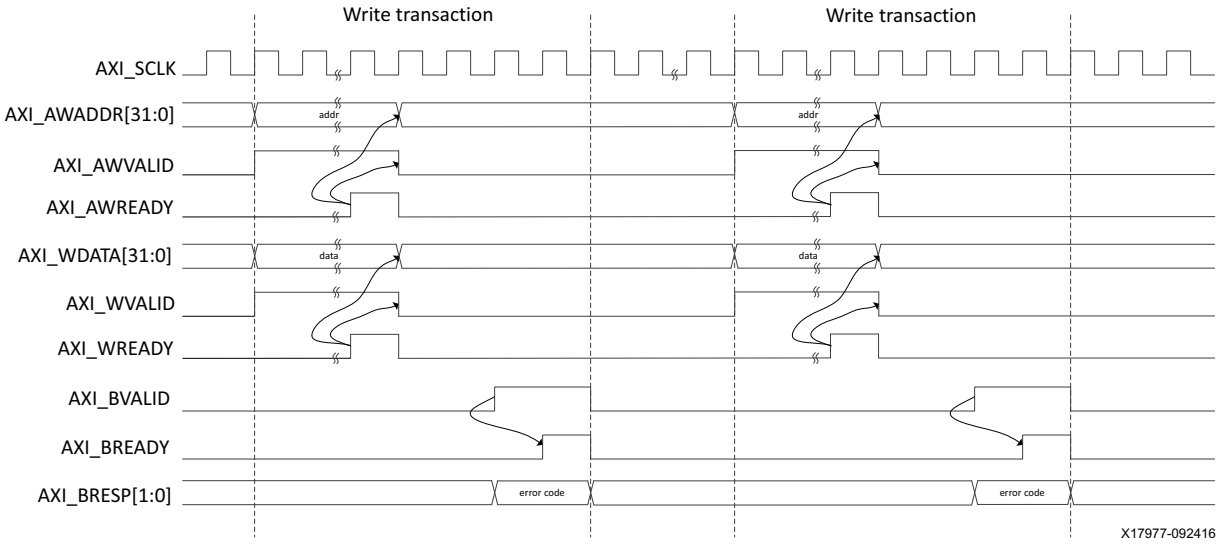

Figure 5-12: **AXI4-Lite User Side Write Transaction with Invalid Write Address**

Valid Read Transactions

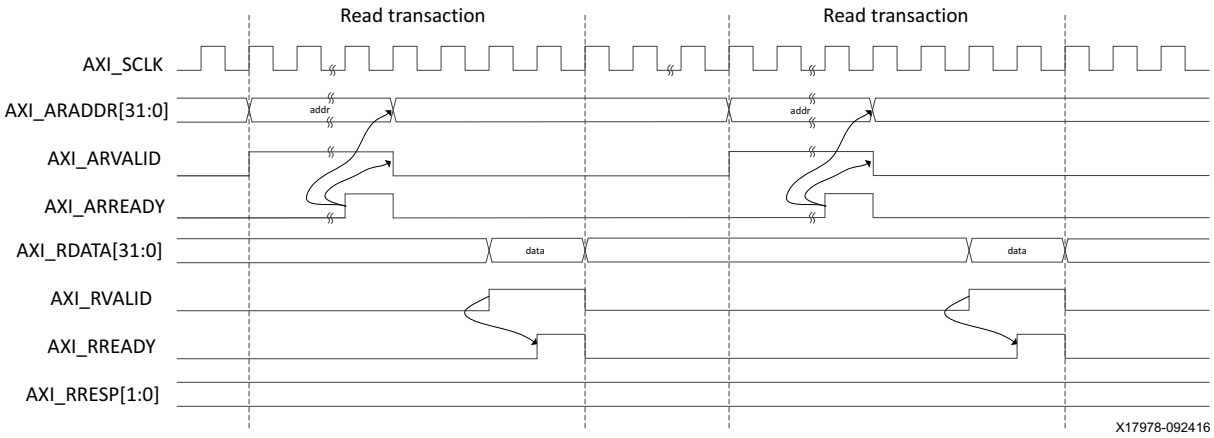

Figure 5-13: **AXI4-Lite User Side Read Transaction**

Invalid Read Transactions

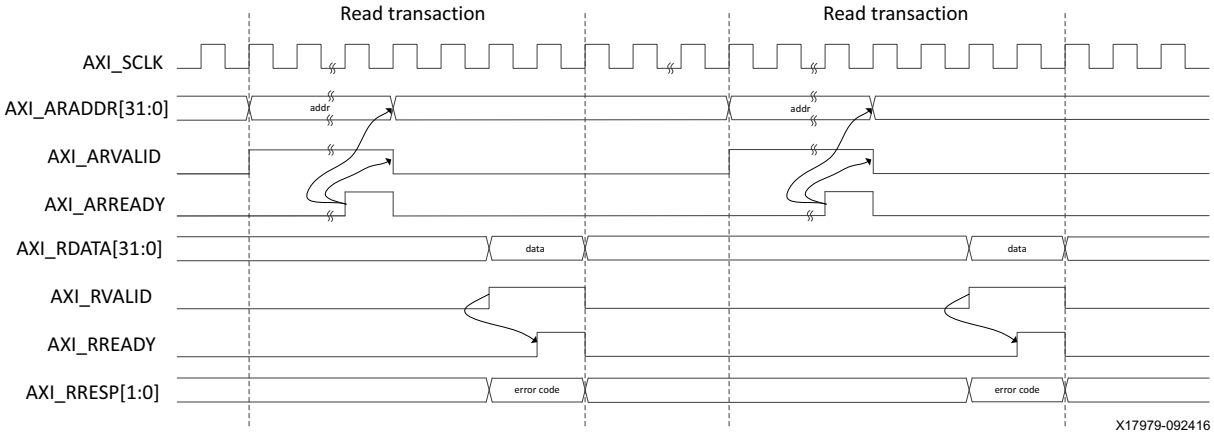

Figure 5-14: **AXI4-Lite User Side Read Transaction with Invalid Read Address**

Batch Mode Test Bench

Each batch mode release of the 1G/10G/25G Ethernet subsystem includes a demonstration test bench that performs a loopback test on the complete subsystem. For your convenience, scripts are provided to launch the test bench from several industry-standard simulators. The test program exercises the datapath to check that the transmitted frames are received correctly. Register Transfer Level (RTL) simulation models for the subsystem are included. You must provide the correct path for the transceiver simulation model according to the latest simulation environment settings in your version of the Vivado® Design Suite.

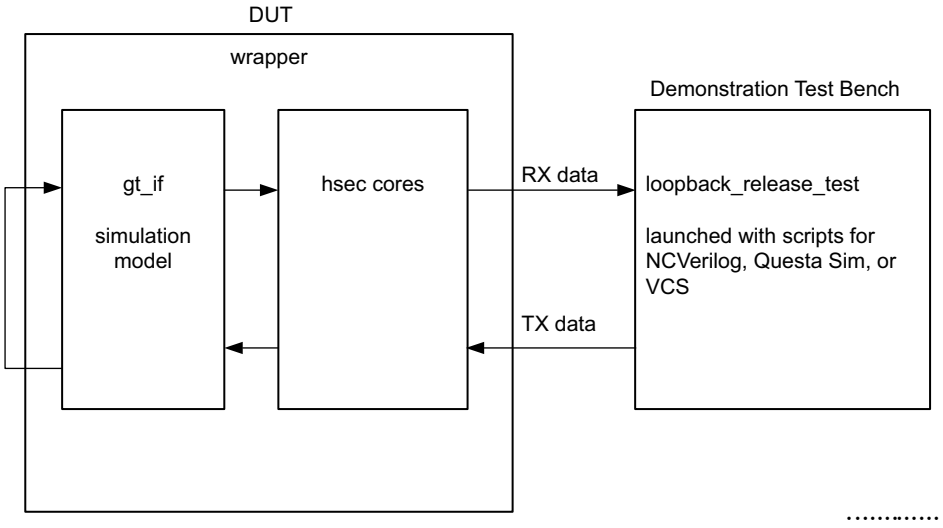

Figure 6-1: **Test Bench**

Debugging

This appendix includes details about resources available on the Xilinx Support website and debugging tools.

TIP: *If the IP generation halts with an error, there might be a license issue. See License Checkers in Chapter 1 for more details.*

Finding Help on Xilinx.com

To help in the design and debug process when using the 1G/10G/25G Ethernet subsystem, the Xilinx Support web page contains key resources such as product documentation, release notes, answer records, information about known issues, and links for obtaining further product support.

Documentation

This product guide is the main document associated with the 1G/10G/25G Ethernet subsystem. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page or by using the Xilinx® Documentation Navigator.

Download the Xilinx Documentation Navigator from the Downloads page. For more information about this tool and the features available, open the online help after installation.

Solution Centers

See the Xilinx Solution Centers for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

Refer to the Xilinx Ethernet IP Solution Center.

Answer Records

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product. Answer Records are created and maintained daily ensuring that users have access to the most accurate information available.

Answer Records for this subsystem can be located by using the Search Support box on the main Xilinx support web page. To maximize your search results, use proper keywords such as

- Product name
- Tool message(s)
- Summary of the issue encountered

A filter search is available after results are returned to further target the results.

Technical Support

Xilinx provides technical support in the Xilinx Support web page for this product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support if you do any of the following:

- Implement the solution in devices that are not defined in the documentation.
- Customize the solution beyond that allowed in the product documentation.
- Change any section of the design labeled DO NOT MODIFY.

To contact Xilinx Technical Support, go to the Xilinx Support web page.

Debug Tools

There are many tools available to address 1G/10G/25G Ethernet subsystem design issues. It is important to know which tools are useful for debugging various situations.

Vivado Design Suite Debug Feature

The Vivado® Design Suite debug feature inserts logic analyzer and virtual I/O cores directly into your design. The debug feature also allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature in the Vivado IDE is used for logic debugging and validation of a design running in Xilinx devices.

The Vivado logic analyzer is used with the logic debug IP cores, including:

- ILA 2.0 (and later versions)
- VIO 2.0 (and later versions)

See the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 9].

Reference Boards

Various Xilinx development boards support the 1G/10G/25G Ethernet subsystem. These boards can be used to prototype designs and establish that the core can communicate with the system.

- UltraScale™ FPGA evaluation boards
  - ZCU102

Simulation Debug

Simulator License Availability

If the simulator does not launch, you might not have a valid license. Ensure that the license is up to date. It is also possible that your organization has a license available for one of the other simulators, so try all the provided scripts.

Slow Simulation

Simulations can appear to run slowly under some circumstances. If a simulation is unacceptably slow, the following suggestions might improve the run-time performance.

1. Use a faster computer with more memory.
2. Make use of a Platform Load Sharing Facility (LSF) if available in your organization.
3. Bypass the Xilinx transceiver (this might require that the customer create their own test bench.
4. Send fewer packets.
5. If using the example design, see Simulation Speed Up in Chapter 4 to speed up wait timers in the example design.

Simulation Fails Before Completion

If the sample simulation fails or hangs before successfully completing, it is possible that a timeout has occurred. Ensure that the simulator timeouts are long enough to accommodate the waiting periods in the simulation, for example during the lane alignment phase.

Simulation Completes But Fails

If the sample simulation completes with a failure, contact Xilinx technical support. Each release is tested prior to shipment and normally completes successfully. Consult the sample simulation log file for the expected behavior.

The simulation debug flow for Questa® SIM is illustrated in Figure A-1. A similar approach can be used with other simulators.

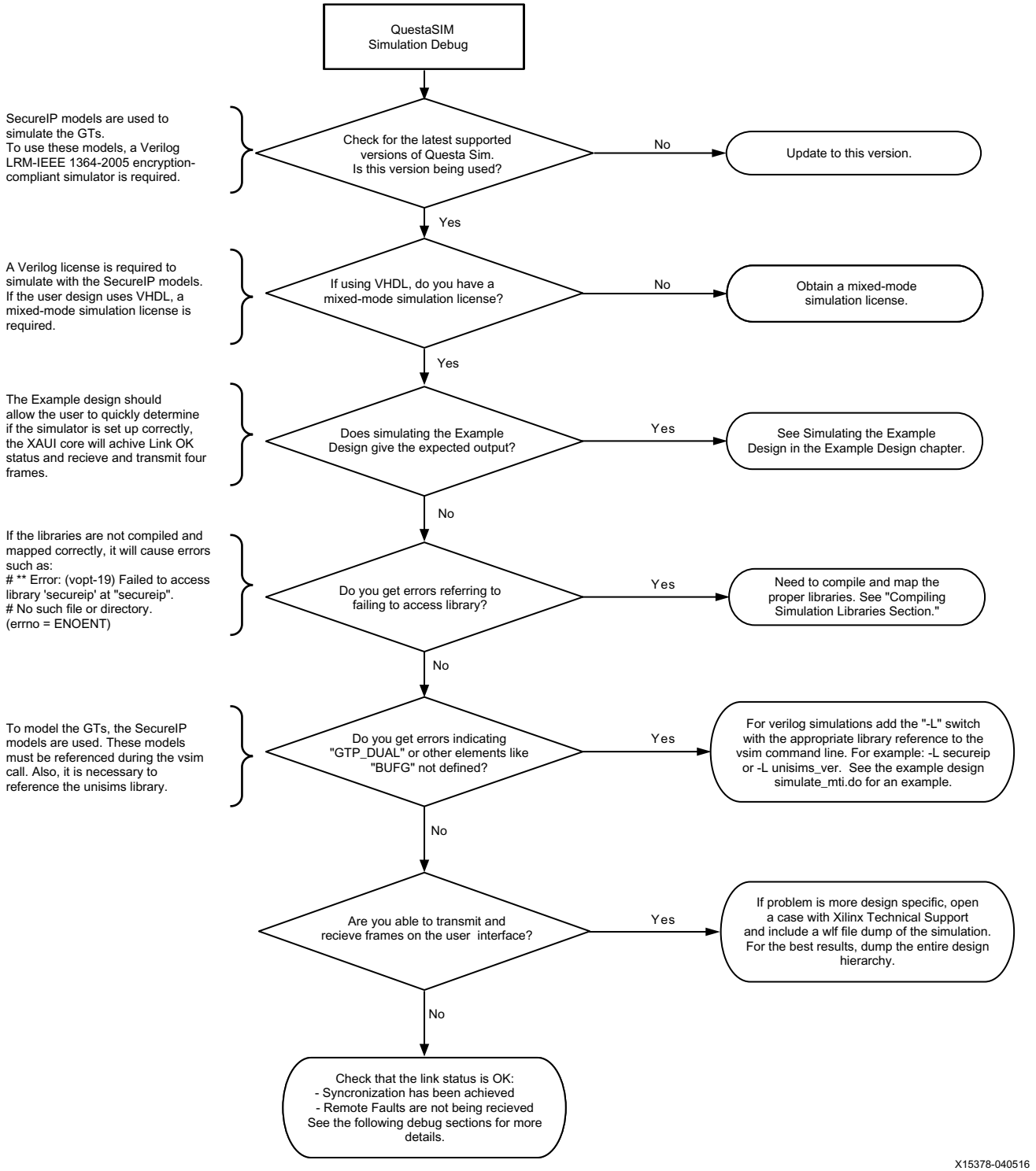

Figure A-1: **Mentor Graphics Questa Advanced Simulator Simulation Debug Flow**

Hardware Debug

Hardware issues can range from link bring-up to problems seen after hours of testing. This section provides debug steps for common issues. The Vivado debug feature is a valuable resource to use in hardware debug. The signal names mentioned in the following individual sections can be probed using the debug feature for debugging the specific problems.

General Checks

Ensure that all the timing constraints for the core were properly incorporated from the example design and that all constraints were met during implementation.

- Does it work in post-place and route timing simulation? If problems are seen in hardware but not in timing simulation, this could indicate a PCB issue. Ensure that all clock sources are active and clean.
- If using mixed-mode clock managers (MMCMs) in the design, ensure that all MMCMs have obtained lock by monitoring the LOCKED port.
- If your outputs go to 0, check your licensing.

Timing

Ensure that the timing is met according to the Vivado tools before attempting to implement the IP in hardware.

Transceiver Specific Checks

- Ensure that the polarities of the txn/txp and rxn/rxp lines are not reversed. If they are, these can be fixed by using the `TXPOLARITY` and `RXPOLARITY` ports of the transceiver.
- Check that the transceiver is not being held in reset or still being initialized. The RESETDONE outputs from the transceiver indicate when the transceiver is ready.
- Place the transceiver into parallel or serial near-end loopback.
- If correct operation is seen in the transceiver serial loopback, but not when loopback is performed through an optical cable, it might indicate a faulty optical module.
- If the core exhibits correct operation in the transceiver parallel loopback but not in serial loopback, this might indicate a transceiver issue.
- A mild form of bit error rate might be solved by adjusting the transmitter Pre-Emphasis and Differential Swing Control attributes of the transceiver.

Ethernet Specific Checks

A number of issues can commonly occur during the first hardware test of an 1G/10G/25G Ethernet subsystem. These should be checked as indicated below.

It is assumed that the 1G/10G/25G Ethernet subsystem has already passed all simulation testing which is being implemented in hardware. This is a pre-requisite for any kind of hardware debug.

The usual sequence of debugging is to proceed in the following sequence:

1. Clean up signal integrity.
2. Ensure that the SerDes achieves clock data recovery (CDR) lock.
3. Check that the 1G/10G/25G Ethernet subsystem has achieved word sync.
4. Proceed to Interface and Protocol debug.

Signal Integrity

When bringing up a board for the first time and the 1G/10G/25G Ethernet subsystem does not seem to be achieving word sync, the most likely problem is related to signal integrity. Signal integrity issues must be addressed before any other debugging can take place.

Signal integrity should be debugged independently from the 1G/10G/25G Ethernet subsystem. The following procedures should be carried out. (Note that it assumed that the PCB itself has been designed and manufactured in accordance with the required trace impedances and trace lengths, including the requirements for skew set out in the IEEE 802.3 specification.)

- Transceiver Settings
- Checking For Noise
- Bit Error Rate Testing

If assistance is required for transceiver and signal integrity debugging, contact Xilinx technical support.

N/P Swapping

If the positive and negative signals of a differential pair are swapped, then data cannot be correctly received on that lane. You should verify that the link has the correct polarity of each differential pair.

Clocking and Resets

Refer to the Clocking and Resets in Chapter 3 for these requirements.

Ensure that the clock frequencies for both the 1G/10G/25G Ethernet subsystem as well as the Xilinx Transceiver reference clock match the configuration requested when the subsystem was ordered. The core clock has a minimum frequency associated with it. The maximum core clock frequency is determined by timing constraints. The minimum core clock frequency is derived from the required Ethernet bandwidth plus the margin reserved for clock tolerance, wander and jitter.

The first thing to verify during debugging is to ensure that resets remain asserted until the clock is stable. It must be frequency-stable as well as free from glitches before the 1G/10G/25G Ethernet subsystem is taken out of reset. This applies to both the SerDes clock as well as the core clock.

If any subsequent instability is detected in a clock, the 1G/10G/25G Ethernet subsystem must be reset. One example of such instability is a loss of CDR lock. The user logic should determine all external conditions which would require a reset (e.g. clock glitches, loss of CDR lock, power supply glitches, etc.).

The GT requires a GTRXRESET after the serial data becomes valid to ensure correct CDR lock to the data. This is required after powering on, resetting or reconnecting the link partner. At the core level to avoid interruption on the TX side of the link, the reset can be triggered using `gtwiz_reset_rx_datapath`. If available, signal detect or inversion of loss of signal from the optics can be used to trigger the reset. If signal detect or loss of signal are not available, timeout logic can be added to monitor if alignment has not completed and issue the `gtwiz_reset_rx_datapath` reset.

Configuration changes cannot be made unless the subsystem is reset. An example of a configuration change would be setting a different maximum packet length. Check the description for the particular signal on the port list to determine if this requirement applies to the parameter that is being changed.

RX Debug

Consult the port list section for a description of the diagnostic signals which are available to debug the RX.

stat_rx_block_lock

This signal indicates that the receiver has detected and locked to the word boundaries as defined by a 01 or 10 control or data header. This is the first step to ensure that the 10/25G Ethernet IP is functioning normally.

CAUTION! *Under some conditions of no signal input, the SerDes receiver exhibits a steady pattern of alternating 1010101.... This can cause erroneous block lock, but still indicates that the receiver has detected the pattern.*

stat_rx_bad_fcs

A bad FCS indicates a bit error in the received packet. An FCS error could be due to any number of causes of packet corruption such as noise on the line.

stat_rx_local_fault

A local fault indication can be locally generated or received. Some causes of a local fault are:

- block lock not complete
- high bit error rate
- overflow or underflow

Loopback Check

If the Ethernet packets are being transmitted properly according to 802.3, there should not be RX errors. However, the signal integrity of the received signals must be verified first.

To aid in debug, a local loopback can be performed with the signal `ctl_local_loopback`. This connects the TX SerDes to the RX SerDes, effectively bypassing potential signal integrity problems. The transceiver is placed into "PMA loopback", which is fully described in the transceiver product guide. In this way, the received data can be checked against the transmitted packets to verify that the logic is operating properly

Protocol Interface Debug

To achieve error-free data transfers with the 1G/10G/25G Ethernet subsystem, the 802.3 specification should be followed. Note that signal integrity should always be ensured before proceeding to the protocol debug.

Diagnostic Signals

There are many error indicators available to check for protocol violations. Carefully read the description of each one to see if it is useful for a particular debugging problem.

The following is a suggested debug sequence:

1. Ensure that Word sync has been achieved.
2. Make sure there are no descrambler state errors.
3. Eliminate CRC32 errors, if any.
4. Make sure the protocol is being followed correctly.
5. Ensure that there are no overflow or underflow conditions when packets are sent.

Statistics Counters

After error-free communication has been achieved, the statistics indicators can be monitored to ensure that traffic characteristics meet expectations. Note that some signals are strobes only, which means that the counters are not part of the subsystem. This is done so that the counter size can be customized. Counters are optionally available with the AXI interface.

Appendix B

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

Documentation Navigator and Design Hubs

Xilinx Documentation Navigator provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open the Xilinx Documentation Navigator (DocNav):

- From the Vivado® IDE, select **Help > Documentation and Tutorials**.
- On Windows, select **Start > All Programs > Xilinx Design Tools > DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In the Xilinx Documentation Navigator, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

Note: For more information on Documentation Navigator, see the Documentation Navigator page on the Xilinx website.

References

These documents provide supplemental material useful with this product guide:

1. IEEE Standard 802.3-2015
2. *25G and 50G Ethernet Consortium Schedule 3 version 1.6 (August 18, 2015*

   (http://25gethernet.org/)
3. *1G/2.5G Ethernet PCS/PMA or SGMII LogiCORE IP Product Guide* (PG047)
4. *10G/25G High Speed Ethernet Subsystem Product Guide* (PG210)
5. *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
6. *Vivado Design Suite User Guide: Designing with IP* (UG896)
7. *Vivado Design Suite User Guide: Getting Started* (UG910)
8. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
9. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
10. *Vivado Design Suite User Guide - Implementation* (UG904)
11. *Vivado Design Suite AXI Reference Guide* (UG1037)
12. IEEE Standard 1588-2008, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems" (standards.ieee.org/findstds/standard/1588-2008.html)
13. *UltraScale Architecture GTH Transceivers User Guide* (UG576)
14. *UltraScale Architecture GTY Transceivers User Guide* (UG578)
15. *ARM AMBA AXI Protocol v2.0 Specification (ARM IHI 0022C)*

    (http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ihi0022c/index.html)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/04/2018	2.0	Initial Xilinx release.
12/20/2017	1.0	EA: Xilinx Confidential Draft. Approved for external release under NDA only.

Please Read: Important Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

© Copyright 2017-18 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.