

2.5G Ethernet MAC + PHY IP

IP Version: v1.0.0

User Guide

FPGA-IPUG-02293-1.2

October 2025



Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor's inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice's inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.



Contents

| Contents | 3 |
|--|----|
| Abbreviations in This Document | 7 |
| 1. Introduction | 8 |
| 1.1. Overview of the IP | 8 |
| 1.2. Quick Facts | 8 |
| 1.3. Features | |
| 1.4. Licensing and Ordering Information | g |
| 1.4.1. Ordering Part Number | g |
| 1.5. IP Validation Summary | g |
| 1.6. Minimum Device Requirements | |
| 1.7. Naming Conventions | 10 |
| 1.7.1. Nomenclature | 10 |
| 1.7.2. Signal Names | |
| 2. Functional Description | |
| 2.1. MAC + PHY | |
| 2.1.1. IP Architecture Overview | |
| 2.1.2. Management Block | |
| 2.2. MAC | |
| 2.2.1. IP Architecture Overview | |
| 2.2.2. Block Diagram | |
| 2.2.3. Ethernet Data Format | |
| 2.2.4. Receive MAC | |
| 2.2.5. Transmit MAC | |
| 2.2.6. Receive AXI4-Stream Interface | |
| 2.2.7. Transmit AXI4-Stream Interface | |
| 2.2.8. MAC Management Block | |
| 2.3. PHY (CertusPro-NX Devices) | |
| 2.3.1. IP Architecture Overview | |
| 2.3.2. Lane Merging | |
| 2.3.3. Reference Clocks | |
| 2.3.4. Transmit State Machine | |
| 2.3.5. Receive State Machine | |
| 2.3.6. Auto-Negotiation State Machine | |
| 2.3.7. PHY Management Block | |
| 2.3.8. PCS Loopback | |
| 2.4. Clocking | |
| 2.4.1. Clocking | |
| 2.5.1. MAC Reset Sequence | |
| 2.5.2. PHY Reset Sequence (CertusPro-NX Devices) | |
| IP Parameter Description | |
| 3.1. MAC + PHY | |
| 3.2. MAC Only | |
| 3.3. PHY Only | |
| 4. Signal Description | |
| 4.1. MAC + PHY Signals (CertusPro-NX Devices) | |
| 4.2. MAC Only Signals | |
| 4.3. PHY Only Signals (CertusPro-NX Devices) | |
| 5. Register Description | |
| 5.1. MAC + PHY Registers (CertusPro-NX Devices) | |
| 5.1.1. Register Address Mapping | |
| 5.2. MAC Registers | |
| 0 | |



| 5.2.1. Configuration Registers for MAC | 40 |
|---|----|
| 5.2.2. Statistics Counters | 44 |
| 5.3. PHY Registers (CertusPro-NX Devices) | 49 |
| 5.3.1. MMD1 Device Identifier 0 | 49 |
| 5.3.2. MMD1 Device Identifier 1 | 50 |
| 5.3.3. MMD1 Devices in Package 0 | 50 |
| 5.3.4. MMD1 Devices in Package 1 | |
| 5.3.5. MMD1 Status Register | |
| 5.3.6. MMD1 Package Identifier 0 | |
| 5.3.7. MMD1 Package Identifier 1 | |
| 6. Example Design | |
| 6.1. Example Design Supported Configuration | |
| 6.2. Overview of the Example Design and Features | |
| 6.3. Example Design Components | |
| 6.3.1. 2.5G Ethernet MAC+PHY IP | |
| 6.3.2. PLL | |
| 6.3.3. OSC | |
| 6.3.4. APB | |
| 6.3.5. Traffic Gen | |
| 6.3.6. XTS-FMC Daughtercard | |
| 6.4. Simulating the Example Design | |
| 6.4.1. Example Design Testbench | |
| 6.4.2. 2.5G Ethernet CertusPro-NX Hardware Example Design Simulation Flow | |
| 6.5. Hardware Testing | |
| 6.5.1. Using 2.5G Ethernet Example Design | |
| 6.5.2. Hardware Setup for CertusPro-NX Evaluation Board with 2.5G Ethernet Example Design | |
| 7. Designing with the IP | |
| 7. Designing with the IP | |
| 7.1. Generating and instantiating the iP | |
| 7.1.1. Generated rifes and rife structure | |
| 7.3. Timing Constraints | |
| ~ | |
| 7.4. Specifying the Strategy | |
| 7.5. Running Functional Simulation | |
| 7.5.1. Simulation Results | |
| 8. Known Issues | |
| 8.1. Minimum IPG Limitation | |
| 8.1.1. Devices Affected | |
| 8.1.2. Designs Affected | |
| 8.1.3. Minimum IPG Limitation | |
| 8.1.4. Planned Fix | |
| Appendix A. Resource Utilization | |
| Appendix B. Guide to Close Timing | |
| References | |
| Technical Support Assistance | |
| Revision History | 82 |
| | |
| Figures | |
| Figure 2.1. 2.5G Ethernet MAC + PHY IP Core Block Diagram | 11 |
| Figure 2.2. 2.5G MAC Only Block Diagram | 12 |
| Figure 2.3. Untagged Ethernet Frame Format | 12 |
| Figure 2.4. Tagged Ethernet Frame Format | |
| Figure 2.5. Pause Frame Format | 13 |



| Figure 2.6. AXI4-Stream RX Adapter Interface Diagram | |
|--|----|
| Figure 2.7. Normal Frame Reception | 15 |
| Figure 2.8. Back-to-back Normal Frame Reception | |
| Figure 2.9. Frame Reception with In-Band FCS Passing | |
| Figure 2.10. AXI4-Stream TX Adapter Interface Diagram | 16 |
| Figure 2.11. Default Normal Frame Transmission | |
| Figure 2.12. Transmission with In-Band FCS Passing | 17 |
| Figure 2.13. 2.5G PHY Only Architecture Diagram | 18 |
| Figure 2.14. CertusPro-NX 2.5G Ethernet PHY Only Top-Level Block Diagram | |
| Figure 2.15. CertusPro-NX 2.5G Ethernet GUI Option | 20 |
| Figure 2.16. Reference Clock Block Diagram | 21 |
| Figure 2.17. Typical Transmit Timing Diagram | 22 |
| Figure 2.18. Typical Receive Timing Diagram | 22 |
| Figure 2.19. Typical Auto-Negotiation Timing Diagram for MAC-Side Entity | 22 |
| Figure 2.20. 8b10b PCS Loopback Diagram | 23 |
| Figure 2.21. 2.5G MAC + PHY IP Core Clock Domain Block Diagram | 24 |
| Figure 2.22. Sequence to Configure RX MAC In-Band FCS Passing | 25 |
| Figure 2.23. Power-up Sequence for CertusPro-NX Devices | 25 |
| Figure 6.1. 2.5G MAC + PHY Hardware Example Design Block Diagram | |
| Figure 6.2. 2.5G Ethernet Example Design Flowchart | |
| Figure 6.3. Simulation Wizard Interface | 55 |
| Figure 6.4. Overall Simulation Results for 2.5G Ethernet MAC+PHY | |
| Figure 6.5. Simulation Results for 2.5G Link Status | |
| Figure 6.6. Simulation Results for Data Transmission | |
| Figure 6.7. Simulation Transcript | |
| Figure 6.8. CertusPro-NX Evaluation Board | |
| Figure 6.9. XTS-FMC Board | |
| Figure 6.10. Project Creation | |
| Figure 6.11. Project Name and Location | |
| Figure 6.12. Select a Device | |
| Figure 6.13. Project Synthesis Tool | |
| Figure 6.14. IP Catalog | |
| Figure 6.15. IP Component | |
| Figure 6.16. IP Component Configuration – MAC+PHY | |
| Figure 6.17. Importing Versa Files into a Project | |
| Figure 6.18. Steps to Set the top.sv File | |
| Figure 6.19. Steps to Set the to top.v File | |
| Figure 6.20. Steps to Add the cpnx_eval_2p5g.pdc File | |
| Figure 6.21. Generate Bitstream | |
| Figure 6.22. Bitstream Completion | |
| Figure 6.23. Programmer | |
| Figure 6.24. Select Bitstream | |
| Figure 6.25. Programming Bitstream | |
| Figure 6.26. Connectivity Block for CertusPro-NX and XTS-FMC Loopback | |
| Figure 6.27. Actual Board Setup | |
| Figure 6.28. Reveal Signal During Transmission | |
| Figure 7.1. Module/IP Block Wizard | |
| Figure 7.2. IP Configuration | |
| Figure 7.3. Check Generated Result | |
| Figure 7.4. Timing Constraint File (.pdc) for the MAC+PHY Option | |
| Figure 7.5. Timing Constraint File (.pdc) for the PHY Only Option | |
| Figure 7.6. Simulation Wizard | |
| Figure 7.7. Add and Reorder Source | |
| Figure 7.8. Summary Window | |
| · · · · · · · · · · · · · · · · · · · | |



| Figure 7.9. Simulation Waveform | 75 |
|---|----|
| Figure 7.10. Simulation Result | |
| Figure B.1. Multiple Iterations of Place and Route | |
| | |
| | |
| | |
| | |
| Tables | |
| Table 1.1. Summary of the 2.5G Ethernet IP Core | 8 |
| Table 1.2. Ordering Part Number | |
| Table 1.3. 2.5G Ethernet IP Support Readiness | |
| Table 1.4. Minimum Device Requirements for 2.5G Ethernet MAC + PHY IP Core | |
| Table 2.1. Reference Clock MUX Tree Control Signals | |
| Table 3.1. MAC + PHY Attributes | |
| Table 3.2. MAC Only Attributes | |
| Table 3.3. PHY Only Attributes | 28 |
| Table 4.1. Signal Description—MAC + PHY | 30 |
| Table 4.2. Signal Description—MAC Only | 34 |
| Table 4.3. Signal Description—PHY Only (CertusPro-NX Devices) | 36 |
| Table 5.1. Register Address Mapping | 40 |
| Table 5.2. Access Types for MAC Only | 40 |
| Table 5.3. Configuration Registers for MAC | 40 |
| Table 5.4. [0x000] MODE Register | |
| Table 5.5. [0x004] TX_CTL Register | |
| Table 5.6. [0x008] RX_CTL Register | |
| Table 5.7. [0x00C] MAX_PKT_LNGTH Register | |
| Table 5.8. [0x014] MAC_ADDR_0 Register | |
| Table 5.9. [0x018] MAC_ADDR_1 Register | |
| Table 5.10. [0x01C] TX_RX_STS Register | |
| Table 5.11. [0x020] VLAN_TAG Register | |
| Table 5.12. [0x024] MC_TABLE_0 Register | |
| Table 5.13. [0x028] MC_TABLE_1 Register | |
| Table 5.14. [0x02C] PAUSE_OPCODE Register | |
| Table 5.15. [0x030] MAC_CTL Register | |
| Table 5.16. [0x034] PAUSE_TM Register | |
| Table 5.17. Summary of Statistics Counters | |
| Table 5.19. Access Type Definition | |
| Table 5.20. MMD1 Device Identifier 0 | |
| Table 5.21. MMD1 Device Identifier 1 | |
| Table 5.22. MMD1 Devices in Package 0 | |
| Table 5.23. MMD1 Devices in Package 1 | |
| Table 5.24. MMD1 Status Register | |
| Table 5.25. MMD1 Package Identifier 0 | |
| Table 5.26. MMD1 Package Identifier 1 | |
| Table 6.1. 2.5G Ethernet MAC + PHY IP Configuration Supported by the Example Design | |
| Table 6.2. Generated File List | |
| Table 6.3. DIP Switch for CertusPro-NX Evaluation Board | |
| Table 6.4. Pushbuttons for CertusPro-NX Evaluation Board | |
| Table 7.1. Generated File List | |
| Table A.1. Resource Utilization for LFCPNX-100 | |
| | |



Abbreviations in This Document

A list of abbreviations used in this document.

| Abbreviations | Definition |
|-------------------|---|
| APB | Advanced Peripheral Bus |
| AXI4-Lite | Advanced eXtensible Interface 4 Lite |
| AXI4-Stream/AXI4S | Advanced eXtensible Interface 4 Stream |
| CRC | Cyclic Redundancy Check |
| DA | Destination Address |
| DIC | Deficit Idle Count |
| EBR | Embedded Block RAM |
| FCS | Frame Check Sequence |
| FIFO | First-In First-Out |
| GMII | Gigabit Media Independent Interface |
| GPLL | Generic Phase-Locked Loop |
| IEEE | Institute of Electrical and Electronics Engineers |
| IFG | Inter-Frame Gap |
| LUT | Look-Up Table |
| L/T | Length/Type |
| MAC | Media Access Controller |
| MDIO | Management Data Input/Output |
| MMD | MDIO Manageable Device |
| MTU | Maximum Transmission Unit |
| PCS | Physical Coding Sublayer |
| PHY | Physical Layer Device |
| PMA | Physical Medium Attachment |
| SA | Source Address |
| SFD | Start Frame Delimiter |
| SOF | Start of Frame |



1. Introduction

1.1. Overview of the IP

The Lattice 2.5G Ethernet MAC + PHY IP core implements the Media Access Controller (MAC) and state machine functions for the Physical Coding Sub-layer (PCS) described in the IEEE 802.3 (1000BASE-X) specification. Note that the IEEE specification describes a PCS that operates at 1 Gbps. Therefore, this 2.5G PCS state machine does not conform to the IEEE standard specification. The two major differences are data rate (2.5 Gbps instead of 1 Gbps) and 16-bits GMII data bus width.

This 2.5G Ethernet MAC + PHY IP core supports the ability to transmit and receive data between a host processor and an Ethernet network. The main function of the Ethernet MAC is to ensure that the Media Access rules specified in the IEEE 802.3 standard are met while transmitting a frame of data over Ethernet. On the receiving side, Ethernet MAC extracts the different components of a frame and transfers them to higher applications through the FIFO interface. The Lattice 2.5G MAC + PHY IP cores are 100% compatible and can be used to create a full PHY/MAC Ethernet datapath that operates at 2.5 Gbps.

This document describes the 2.5G Ethernet MAC + PHY IP core operation and provides instructions for generating the core through the Lattice Radiant™ software. This document also provides guidance on instantiating, synthesizing, and simulating the core.

1.2. Quick Facts

Table 1.1. Summary of the 2.5G Ethernet IP Core

| able 1.1. Julillary of the 2.50 Effective Core | | | | |
|--|--------------------------|---|--|--|
| | Supported Devices | CertusPro™-NX | | |
| | | MAC only | | |
| IP Requirements | IP Option | PHY only | | |
| ir Requirements | | MAC + PHY | | |
| | IP Changes | For a list of changes to the IP, refer to the 2.5G Ethernet MAC + PHY IP Release Notes (FPGA-RN-02083). | | |
| Danassana Hailination | Supported User Interface | AXI4-Stream or APB. | | |
| Resource Utilization | Resources | See Appendix A. Resource Utilization. | | |
| | Lattice Implementation | IP core v1.0.0—Lattice Radiant™ software 2025.1 or later. | | |
| Design Tool Support | Synthesis | Synopsys® Synplify Pro® software, O-2018.09LR-SP1. | | |
| Design 1001 support | Simulation | For the list of supported simulators, see the Lattice Radiant Software User Guide on the Lattice Radiant Software web page. | | |

1.3. Features

The 2.5G MAC + PHY Ethernet IP core offers the following key features:

MAC

- Compliant to IEEE 802.3-2005 standard
- Full-duplex operation
- Independent TX and RX Maximum Transmission Unit (MTU) frame length
- Programmable promiscuous (transparent) mode
- Supports deficit idle count (DIC)
- Advanced Peripheral Bus (APB) interface for register access
- Transmit and receive statistics vector
- Multicast address filtering
- Supports the following frame features:
 - Full-duplex control using PAUSE frames
 - VLAN tagged frames



- Automatic padding of short frames
- Multicast and Broadcast frames
- Optional FCS transmission and reception
- Jumbo frames of 9,600 bytes

PHY

- Implements the transmit and receive functions described in Clause 36 of IEEE 802.3
- 16-bits GMII interface operating at 156.25 MHz (2.5 Gbps)
- Parallel signal interface for control and status management
- Supports standard 2.5 Gbps Ethernet link layer data rate
- 8b/10b encoding and decoding support
- Auto negotiation with link configuration functions described in Clause 37 of IEEE 802.3

Other Features

Supports driver codes for the Propel™ Builder software

1.4. **Licensing and Ordering Information**

An IP specific license string is required to enable full use of the 2.5G Ethernet MAC + PHY IP core in a complete, top-level design. The IP can be fully evaluated through functional simulation and implementation (synthesis, map, place and route) without an IP license string. This IP supports Lattice's IP hardware evaluation capabilities. You can create versions of the IP to operate in hardware for a limited time (approximately four hours) without requiring an IP license string. A license string is required to enable timing simulation and to generate a bitstream file that does not include the hardware evaluation timeout limitation.

For more information about pricing and availability of the 2.5G Ethernet MAC + PHY IP core, contact your local Lattice Sales Office.

1.4.1. Ordering Part Number

Table 1.2. Ordering Part Number

| Device Family | Part Number | | | |
|---------------|--|--------------------|--|--|
| | Single Seat Annual Single Seat Perpetual | | | |
| CertusPro-NX | ETHER-2PT5-CPNX-US | ETHER-2PT5-CPNX-UT | | |

1.5. **IP Validation Summary**

The following table provides IP support information on the 2.5G Ethernet MAC + PHY IP core.

Table 1.3. 2.5G Ethernet IP Support Readiness

| Device Family | Mode | Radiant Timing Model | Hardware Validated |
|----------------------|-----------|----------------------|--------------------|
| CertusPro-NX | MAC + PHY | Final | Yes |



1.6. Minimum Device Requirements

The minimum device requirements for the 2.5G Ethernet MAC + PHY IP are as follows:

Table 1.4. Minimum Device Requirements for 2.5G Ethernet MAC + PHY IP Core

| Devices | Speed Grades |
|---------------|----------------|
| LFCPNX-50/100 | Speed grade 7* |

^{*}Note: Speed grade 7 is only supported for the default mode. Other modes require speed grade 8.

1.7. Naming Conventions

1.7.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL.

1.7.2. Signal Names

- _n are active low signals (asserted when value is logic 0)
- _*i* are input signals
- _o are output signals
- _io are bi-directional input/output signals



2. Functional Description

2.1. MAC + PHY

2.1.1. IP Architecture Overview

The 2.5G Ethernet MAC + PHY IP core is a fully synchronous machine composed of Transmit and Receive sections that operate independently to support full duplex operation between host processor and Ethernet network. With the MAC + PHY option selected, the IP consists of a MAC and a PHY that is connected through the 16-bit GMII internally. This IP option is supported by CertusPro-NX devices.

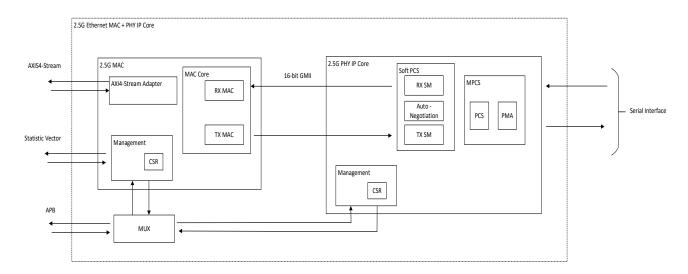


Figure 2.1. 2.5G Ethernet MAC + PHY IP Core Block Diagram

The 2.5G Ethernet MAC + PHY IP core includes the following layers:

- Physical layer
- MAC layer

2.1.2. Management Block

The Management block can be accessed through the APB interface when the MAC + PHY option is selected.

For more details about the management blocks in MAC and PHY respectively, refer to the MAC Management Block and PHY Management Block sections. For protocol timing details, refer to AMBA 3 APB Protocol version 1.0 Specification.

2.2. MAC

2.2.1. IP Architecture Overview

The 2.5G MAC Only IP option core transmits and receives data between a client application and an Ethernet network. The main function of the Ethernet MAC is to ensure that the Media Access rules specified in the IEEE 802.3 standard are met while transmitting and receiving Ethernet frames. Figure 2.3, Figure 2.4, and Figure 2.5 show some of the frame formats of data transmitted and received on the Ethernet network that the 2.5G MAC Only IP option core supports.

On the receiving side, the Ethernet MAC extracts the different components of a frame and transfers them to higher applications through the client FIFO interface.



The data received from the GMII interface is first buffered until sufficient data is available to be processed by the Receive MAC (RX MAC). The Preamble and the Start-of-Frame Delimiter (SFD) information are then extracted from the incoming frame to determine the start of a valid frame. The Receive MAC checks the address of the received packet and validates whether the frame can be received before transferring it into the FIFO. Only valid frames are transferred into the FIFO (runts and fragments are discarded). The RX MAC also provides a statistical vector on a per packet basis that can be used by the application. The 2.5G MAC Only IP option core always calculates CRC to check whether the frame was received error-free.

On the transmit side, the TX MAC is responsible for controlling access to the physical medium. The TX MAC reads data from an external client TX FIFO, formats this data into an Ethernet packet, and passes it to the GMII module.

The TX MAC reads data from the TX Client FIFO when the client indicates a packet is available, and the TX MAC is in its appropriate state. The TX MAC pre-fixes the Preamble and the SFD information to the data and appends the Frame Check Sequence at the end of the data.

2.2.2. Block Diagram

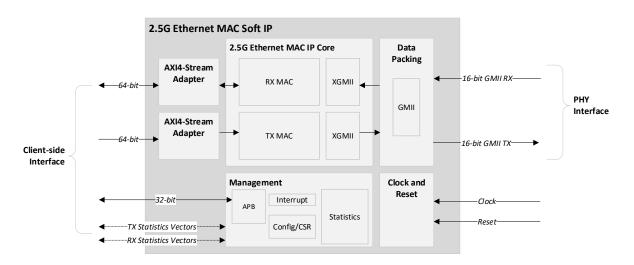


Figure 2.2. 2.5G MAC Only Block Diagram

2.2.3. Ethernet Data Format

The following figure shows an untagged Ethernet frame format.

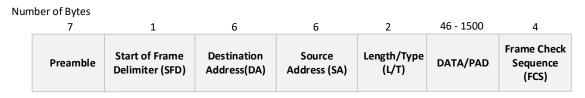


Figure 2.3. Untagged Ethernet Frame Format

The following figure shows a tagged Ethernet frame format.

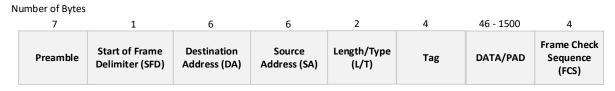


Figure 2.4. Tagged Ethernet Frame Format



The following figure shows Ethernet Control Pause Frame Format

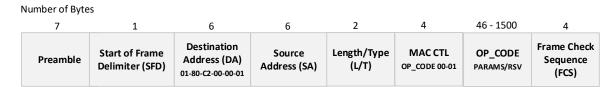


Figure 2.5. Pause Frame Format

A Tagged frame includes a 4-byte VLAN Tag field, which is located between the Source Address field and the Length/Type field. The VLAN Tag field includes the VLAN Identifier and other control information needed when operating with Virtual Bridged LANs as described in IEEE 802.1Q.

The MAC is responsible for constructing a valid frame from the data received from the client before transmitting it. On the receiver path, it receives frames from the network through the 16-bit GMII interface and passes the parameters of the frame to the client through the AXI4-Stream interface.

Note that the 2.5G MAC GMII interface uses a 16-bit data interface. Although this interface conforms to the timing specified in IEEE 802.3 Clause 35, the data bus width and clock frequency do not conform to the IEEE standard. To overcome this issue, Lattice provides a 2.5G PHY IP OPTION core with a 16-bit datapath and 156.25 MHz clock that is directly compatible with the 2.5G MAC GMII interface as shown in Figure 2.1.

2.2.4. Receive MAC

The Receive MAC receives the incoming frames and transfers them to the client via the AXI4-Stream interface. In the process, it performs the following operations:

- Checks the frame for a valid start of frame (SOF) and SFD symbol.
- Determines whether the frame should be received by analyzing the DA.
- Determines the type of the frame by analyzing the length/type field.
- Checks for any errors in the frame by recalculating the CRC and comparing it with the expected value.

The RX MAC operation is determined by programming the MODE and RX CTL registers.

You can specify whether the FCS field should be transferred on to the AXI4-Stream interface by programming the rx_pass_fcs bit of RX_CTL register. If the FCS field is to be stripped off the frame, any padding bytes within the frame are stripped off as well.

Once a valid SOF is detected, the DA field of the incoming frame is analyzed. If the DA field is a unicast address, it is compared with the programmed MAC address. Unless the prms bit of RX_CTL register was set, the incoming frame is discarded if the DA field and the programmed MAC address (MAC_ADDR_{0,1} registers) do not match. If the frame had a multicast address and if receive_all_mc signal is not asserted, all such frames are dropped (except PAUSE frames). If the frame had a multicast address and if receive_all_mc signal is asserted, the multicast frames are subject to the following address filtering rules:

For all frames with multicast address, the CRC of the destination address is computed and the mid-six bits of the least significant byte of the CRC is chosen as the address to a hash table. The 64-bit hash table is programmed in the MC_TABLE_{0,1} registers. The MAC implements an eight-row table with eight bits in each row. The lower three bits of the selected CRC are used to select one of these eight rows and the next three bits are used to select one of the bits in the selected table. The incoming multicast frame is accepted if the bit selected from the hash table is set to one. It is discarded if the bit selected is zero.

If the incoming frame had a broadcast address, it is accepted if either the prms or the receive_bc bit of RX_CTL register is set. A broadcast frame is discarded if none of these bits are set.



2.2.5. Transmit MAC

The Transmit MAC controls access to the physical medium. Its main functions are as follows:

- Data padding for short frames when FCS generation is enabled.
- Generation of a pause frame when the tx_pausreq bit of MAC_CTL register is asserted. The bit tx_fc_en of TX_CTL register should be set to 1 to enable this feature.
- To stop frame transmission when a pause frame is received by the Receive MAC.
- Implement link fault signaling logic and transmit appropriate sequences based on the remote link status of the TX_RX_STS register.

The TX MAC operation is determined by programming the MODE and TX_CTL registers.

By default, the Transmit MAC is configured to generate the FCS pattern for the frame to be transmitted. However, this can be prevented by setting tx_pass_fcs bit of the TX_CTL register. This feature is useful if the frames being presented for transmission already contain the FCS field. When FCS field generation by the MAC is disabled, ensure that short frames are properly padded before the FCS is generated. If the MAC receives a frame to transmit that is shorter than 64 bytes when FCS generation is disabled, the frame is sent as is and a statistic vector for the condition is generated.

The DA, SA, L/T, and data fields are derived from higher applications through the AXI4-Stream interface and then encapsulated into an un-tagged Ethernet frame. The frame encapsulation consists of adding the preamble bits, the SFD bits, and the CRC check sum to the end of the frame. If transmit_short bit of TX_CTL register is not set, all short frames are padded.

The frame is not sent over the network until the network has been idle for a minimum of Inter-Packet Gap (IPG). The TX MAC requires a continuous stream of data for the entire frame. There cannot be any bubbles of no data transfer within a frame. After the TX MAC is done transmitting a frame, it waits for more frames from the AXI4-Stream interface. During this time, it goes to an idle state that can be detected by reading the TX_RX_STS register. Because the MODE register can be written at any time, the TX MAC can be disabled while it is actively transmitting a frame. In such cases, the MAC completely transmits the current frame and then return to the idle state. The control registers should be programmed only after the MAC has returned to the IDLE state.

There are two different methods for transmitting a pause frame. In the first method, the application layer forms a pause frame and submits it for transmission via the AXI4-Stream interface. In the other method, the application layer signals the TX MAC directly to transmit a pause frame. This is accomplished by asserting tx_pausreq bit of MAC_CTL register. In this case, the TX MAC completes the transmission of the current packet and then transmit a pause frame with the pause time value supplied through the tx_paustim bits of the PAUSE_TMR register.

2.2.6. Receive AXI4-Stream Interface

The receive client-side interface supports the AXI4-Stream interface.

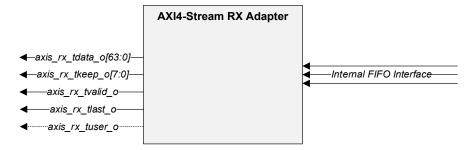


Figure 2.6. AXI4-Stream RX Adapter Interface Diagram

2.2.6.1. Default Normal Frame

The following figure shows the timing diagram of a default normal frame at the Receive AXI4-Stream interface:

- T0: MAC asserts axis_rx_tvalid_o to indicate start of frame. The valid packet includes from the DA field up to the
 end of the data field.
- T1: MAC asserts axis_rx_tlast_o to indicate last packet transfer.
- T2: MAC deasserts axis_rx_tvalid_o when there is no next packet transfer.



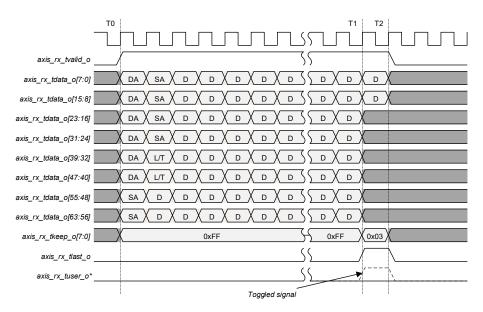


Figure 2.7. Normal Frame Reception

2.2.6.2. Back-to-back Normal Frame

The following figure shows the timing diagram of a back-to-back frame at the Receive AXI4-Stream interface:

- T0: MAC asserts axis_rx_tvalid_o to indicate start of frame. The valid packet starts from the DA field up to the end of the data field.
- T1: MAC asserts axis rx tlast o to indicate last packet transfer.

T2: MAC still asserts the axis_rx_tvalid_o to indicate another packet transfer.

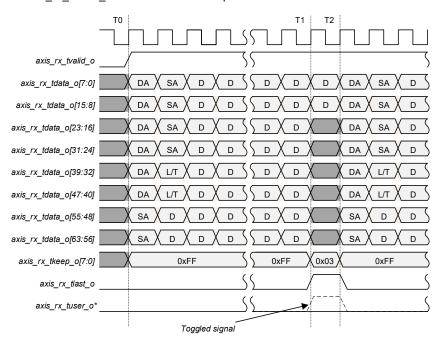


Figure 2.8. Back-to-back Normal Frame Reception

^{*} Note: Under normal circumstances, the axis_rx_tuser_o signal must not be asserted when there is no error. However, if an error occurs and the axis_rx_tuser_o signal is toggled, you must check on the status signal—rx_statvec_o signal.



* Note: Under normal circumstances, the axis_rx_tuser_o signal must not be asserted when there is no error. However, if an error occurs and the axis_rx_tuser_o signal is toggled, you must check on the status signal—rx_statvec_o signal.

2.2.6.3. In-Band FCS Passing

The following figure shows the timing diagram of a frame with in-band FCS. The FCS field is included inside the valid frame.

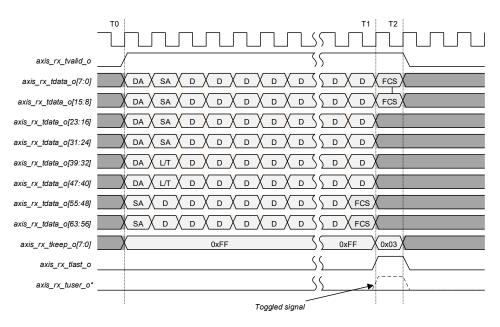


Figure 2.9. Frame Reception with In-Band FCS Passing

* Note: Under normal circumstances, the axis_rx_tuser_o signal must not be asserted when there is no error. However, if an error occurs and the axis_rx_tuser_o signal is toggled, you must check on the status signal—rx_statvec_o signal.

2.2.7. Transmit AXI4-Stream Interface

The transmit client-side interface supports the AXI4-Stream interface.

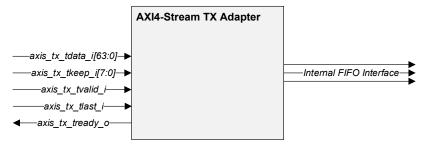


Figure 2.10. AXI4-Stream TX Adapter Interface Diagram

2.2.7.1. Default Normal Frame

The following figure shows the timing diagram of a default normal frame at the Transmit AXI4-Stream interface:

- T0: Client asserts axis_tx_tvalid_i to indicate the start of packet transfer. It must hold the data and axis_tx_tvalid_i until axis tx tready o asserts.
- T1: Client continues to send data once it sees axis_tx_tready_o asserted.
- T2: Client asserts axis_tx_tlast_i to indicate last packet transfer.



T3: Client deasserts axis_tx_tvalid_i when there is no packet transfer. Transmit MAC also deasserts axis_tx_tready_o once it sees the last packet transfer (axis_tx_tlast_i = 1).

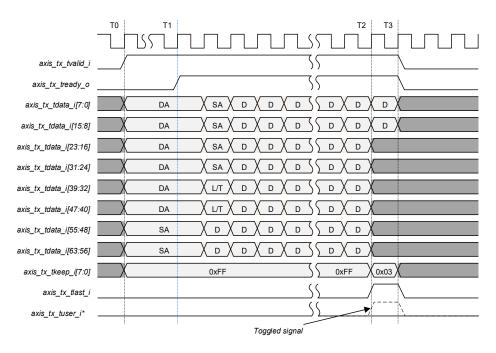


Figure 2.11. Default Normal Frame Transmission

* Note: You can inject the error by asserting the axis_tx_tuser_i signal during EOP. Otherwise, the axis_tx_tuser_i signal must drive to low.

2.2.7.2. In-Band FCS Passing

The following figure shows the timing diagram of a frame when In-Band FCS passing is enabled.

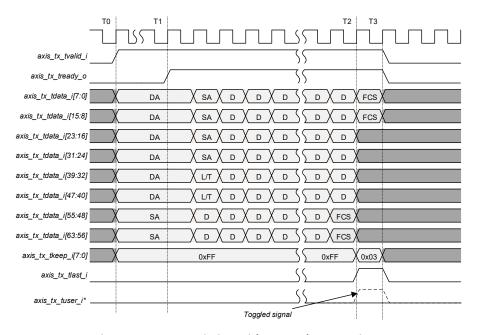


Figure 2.12. Transmission with In-Band FCS Passing



* Note: You can inject the error by asserting the axis_tx_tuser_i signal during EOP. Otherwise, the axis_tx_tuser_i signal must drive to low.

2.2.8. MAC Management Block

The MAC Management block is accessed through the APB interface. This block is responsible for the following:

- Configuration of the core
- Access to interrupt block
- Access to statistics counter

The various events that occur during the reception of a frame are also logged into the statistics vector signals (rx_statvec_o) and the TX_RX_STS register. At the end of reception, the rx_staten_o signal is asserted to qualify the rx_statvec_o signal. A vector is not generated for all those frames that are discarded (no address match or frame length is less than 64 bytes) or ignored (you assert the ignore_pkt of MAC_CTL register). For every frame transmitted, a statistics vector signals (tx_statvec_o) is generated, including all the statistical information collected in the process of transmitting the frame. Data on tx_statvec_o is qualified by assertion of the tx_staten_o signal. These statistics can also be accessed in the statistics counter when the *Statistics Counter Register* attribute is enabled.

For the timing details of the AMBA 3 APB protocol, refer to the AMBA 3 APB Protocol version 1.0 Specification.

2.3. PHY (CertusPro-NX Devices)

2.3.1. IP Architecture Overview

The CertusPro-NX 2.5G Ethernet PHY IP core provides the 16-bit GMII interface to the MAC and follows the IEEE802.3 standard. It supports 16-bit of data and 2-bit of control signals for both transmit and receive path.

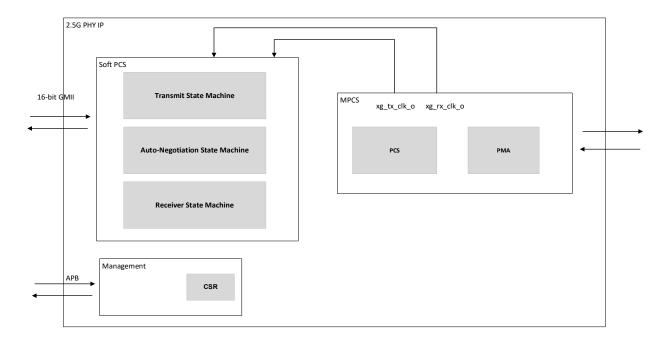


Figure 2.13. 2.5G PHY Only Architecture Diagram

This IP core instantiates the MPCS Module foundation IP configured as single lane 8b/10b Encoder/Decoder and supports APB and MDIO access to MPCS and MMD registers. For more information on MPCS Module foundation IP, refer to the MPCS Module User Guide (FPGA-IPUG-02118).



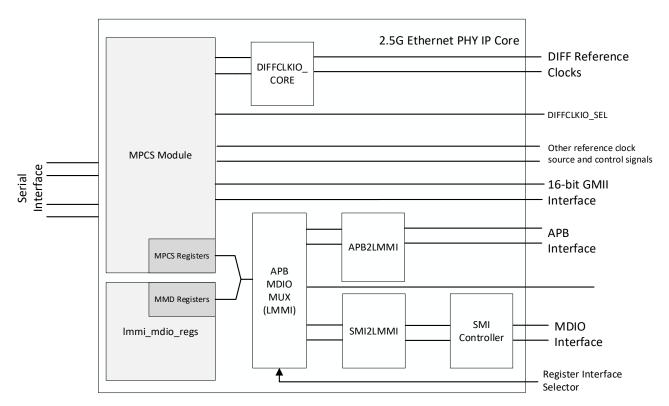


Figure 2.14. CertusPro-NX 2.5G Ethernet PHY Only Top-Level Block Diagram

2.3.2. Lane Merging

To set the LANE ID configuration of the IP according to the number of quad you want to use, follow these steps:

- 1. In the **Select IP Option** field, select **PHY only** or **MAC + PHY**.
- 2. In the **PCS Lane ID** field, set the lane ID configuration of the IP as shown in the figure below. By default, the ID is set to **AUTO**. For more details, refer to the PHY (CertusPro-NX Devices) section on the supported lane.



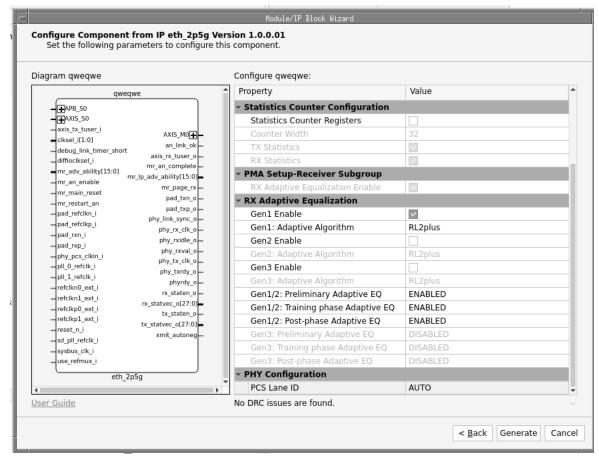


Figure 2.15. CertusPro-NX 2.5G Ethernet GUI Option

2.3.3. Reference Clocks

The CertusPro-NX 2.5G Ethernet PHY IP core provides other sources of reference clocks and can be used for dynamic switching. The pll_0/1_refclk_i must only be connected to PLL and must not be directly connected to the fabric. To use this feature, you must set the reference clock source control signals according to the following table.



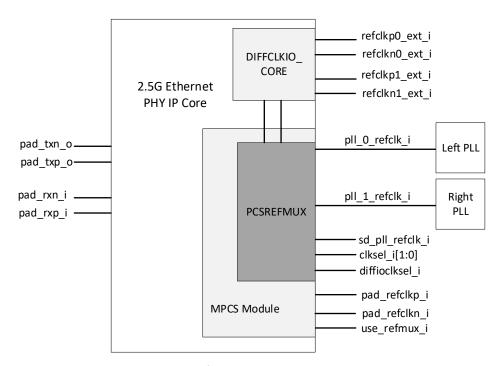


Figure 2.16. Reference Clock Block Diagram

Table 2.1. Reference Clock MUX Tree Control Signals

| Reference Clock Control Signal | Usage | |
|--------------------------------|--|--|
| use_refmux_i | 1'b1 – Use reference clock output from Clock MUX Tree (PCSREFMUX). | |
| | 1'b0 – Use dedicated reference clocks (pad_refclkp/n_i). | |
| diffioclksel_i | 1'b1 – Use refclkp/n1_ext_i as reference clocks. | |
| | 1'b0 – Use refclkp/n0_ext_i as reference clocks. | |
| clksel_i | 2'b00 – Use pll_0_refclk_i as reference clocks. | |
| | 2'b01 – Use pll_1_refclk_i as reference clocks. | |
| | 2'b10 – Use reference clock based on diffioclksel_i. | |
| | 2'b11 – Use sd_pll_refclk_i as reference clock. | |

2.3.4. Transmit State Machine

The transmit state machine implements the transmit functions described in clause 36 of the IEEE 802.3 specification. The state machine's main purpose is to convert GMII data frames into code groups. A typical timing diagram for this circuit block is shown in Figure 2.17. Note that the state machine in this IP core does not fully implement the conversion to 10-bit code groups as specified in the IEEE 802.3 specification. Instead, partial conversion to 8-bit code groups is performed. A separate encoder in the Lattice SERDES completes the full conversion to 10-bit code groups.



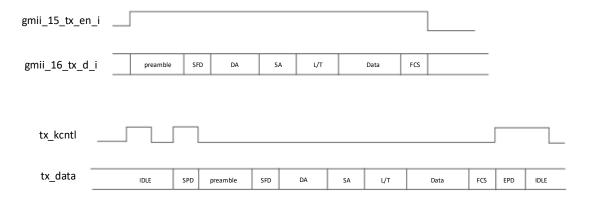


Figure 2.17. Typical Transmit Timing Diagram

2.3.5. Receive State Machine

The receive state machine implements the receive functions described in clause 36 of the IEEE 802.3 specification. The state machine's main purpose is to convert code groups into GMII data frames. A typical timing diagram for this circuit block is shown in Figure 2.18. Note that the state machine in this IP core does not fully implement the conversion from 10-bit code groups as specified in the IEEE 802.3 specification. Instead, partial conversion from 8- bit code groups is performed. A separate decoder in the Lattice SERDES performs 10-bit to 8-bit code group conversions.

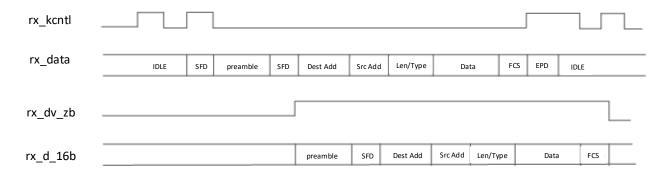


Figure 2.18. Typical Receive Timing Diagram

2.3.6. Auto-Negotiation State Machine

The auto-negotiation state machine implements the link configuration functions described in clause 37 of IEEE 802.3 specification. Refer to the IEEE specification for a detailed description of auto-negotiation operation. From a high-level view, the primary auto-negotiation functions are checking the physical link for proper operation and passing link configuration information between Ethernet PHYs sitting on both sides of the link.

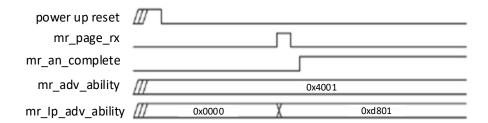


Figure 2.19. Typical Auto-Negotiation Timing Diagram for MAC-Side Entity

© 2025 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.

All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

FPGA-IPUG-02293-1.2



2.3.7. PHY Management Block

The PHY Management block is accessed through the APB interface. This block is responsible for register access of the PCS registers. For details of the PCS register access, refer to the Register Description section.

2.3.8. PCS Loopback

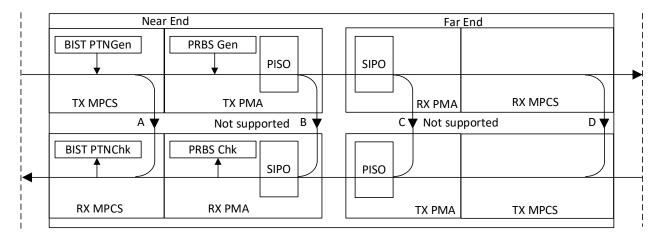


Figure 2.20. 8b10b PCS Loopback Diagram

The following lists the various types of PCS loopback:

- Loopback A This loopback can be enabled when Loopback = 8b/10b PCS Near-End Parallel Loopback Mode.
- Loopback B Not supported.
- Loopback C Not supported.
- Loopback D This loopback can be enabled when Loopback = 8b/10b PCS Far-End Parallel Loopback Mode.

2.4. Clocking

2.4.1. Clocking

2.4.1.1. Clocking Overview

The reference clock frequency for pad_refclk[p,n]_i is 156.25 MHz. Once the 2.5G PHY achieved the synchronization, the xg_rx/tx_clk_o will output a 156.25 MHz clock frequency and supply them to the MAC core for IP_OPTION = MAC + PHY, MAC ONLY, and PHY ONLY. The sysbus_clk_i is used as the register clock source.



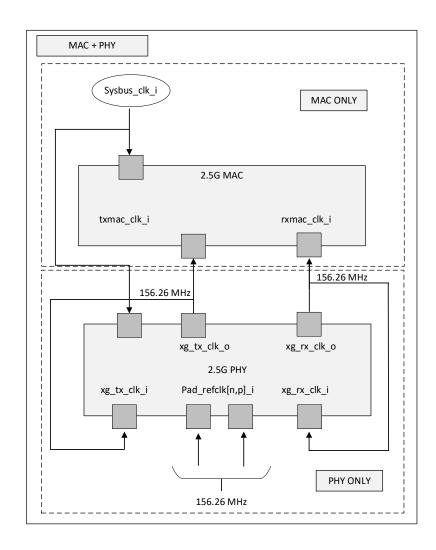


Figure 2.21. 2.5G MAC + PHY IP Core Clock Domain Block Diagram

2.5. Reset

2.5.1. MAC Reset Sequence

An asynchronous reset pin (active low) as system reset is used for resetting the 2.5G Ethernet MAC + PHY IP core. Internal reset logic is implemented to guarantee synchronous de-assertion all throughout the different clock domain among soft logic blocks. The minimum assertion of reset is five clock cycles of the slowest clock*.

2.5.1.1. Power-Up Sequence

The following lists the sequence after power-up of the chip or system:

- 1. T1: Assert reset pin for five clock cycles of the slowest clock* of the system.
- 2. T2: Wait for the 2.5G PHY to be in the ready state.
 For more information on the sequence, see the PHY (CertusPro-NX Devices) section.
- 3. Send settings through the APB interface to configure the system. For a list of MAC registers, refer to the Configuration Registers for MAC section.
- 4. T3: The MAC is ready to receive transfers.
- * Note: The slowest clock refers to the sysbus clk i with frequency that can be as low as 20 MHz.



The following figure shows a sample sequence when the RX MAC is configured to pass the FCS field to the client.

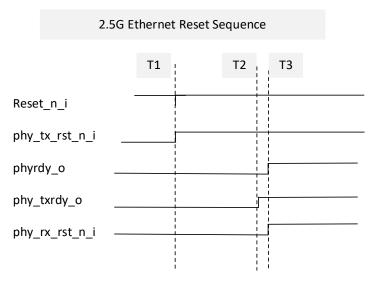


Figure 2.22. Sequence to Configure RX MAC In-Band FCS Passing

2.5.2. PHY Reset Sequence (CertusPro-NX Devices)

2.5.2.1. Reset

An asynchronous reset pin (active low) as system reset is used for resetting the 2.5G Ethernet PHY IP core. Internal reset logic is implemented to guarantee synchronous de-assertion all throughout the different clock domain among soft logic blocks. The minimum assertion of reset is five clock cycles of the slowest clock*.

2.5.2.2. Power-Up Sequence

The following lists the sequence after power-up of the chip or system:

- 1. T1: Assert reset n i pin for five clock cycles of the slowest clock* of the system.
- 2. T2: Wait for the 2.5G PHY (Example: phy_link_sync_o, phyrdy_o, and phy_txrdy_o) to be in the ready state. If phy_link_sync_o, phyrdy_o and phy_txrdy_o signals is deasserted, then input reset pin such as reset_n_i, phy_tx_rst_n_i, and phy_rx_rst_n_i must be deasserted.
- 3. Send settings through the APB interface to configure the system.
- 4. T3: The IP is ready to transmit and receive packets.

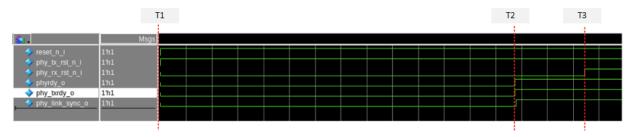


Figure 2.23. Power-up Sequence for CertusPro-NX Devices



3. IP Parameter Description

The configurable attributes of the 2.5G Ethernet IP core are shown in the following tables. You can configure the IP core by setting the attributes accordingly in the IP Catalog's Module/IP wizard of the Lattice Radiant software.

3.1. MAC + PHY

The following table lists the 2.5G Ethernet IP core configurable attributes for the *MAC + PHY* option. The values set for attributes with corresponding registers serve as the maximum values and cannot be set higher during dynamic reconfiguration. Select IP Option—*MAC + PHY* option.

Table 3.1. MAC + PHY Attributes

| Attribute | Selectable Values | Default | Description | Dependency on Other Attributes | |
|--|--|-----------|--|--|--|
| General Configuration | | 1 | | | |
| Select IP Option | MAC + PHYPHY OnlyMAC Only | MAC + PHY | IP option. | _ | |
| Host Interface | Not editable | АРВ | Set the register interface. | _ | |
| Loopback Mode | DisabledNear PMA LoopbackFabric Loopback | Disabled | PCS loopback. For more information, refer to the PCS Loopback section. | _ | |
| MAC Configuration | | | | | |
| Multicast Address Filtering | Enabled Disabled | Disabled | Enables or disables address filtering for multicast frames. | Enabled when Select IP Option == MAC Only or MAC + PHY | |
| TX Pause Frame Generation via Ports | EnabledDisabled | Disabled | Enables or disables TX pause frame generation via ports. | Enabled when Select IP Option == MAC Only or MAC + PHY | |
| Statistics Counter Configurati | on | 1 | · · · | <u>I</u> | |
| Statistic Counter Registers | EnabledDisabled | Disabled | Enables or disables statistics counter registers | Enabled when Select IP Option == MAC Only or MAC + PHY | |
| Counter Width | • 32 • 64 | 32 | Statistics counters register size. | Enabled when Select IP Option == MAC Only or MAC + PHY | |
| TX Statistics | EnabledDisabled | Disabled | TX statistics. | Enabled when Select IP Option == MAC Only or MAC + PHY | |
| RX Statistics | EnabledDisabled | Disabled | RX statistics. | Enabled when Select IP Option == MAC Only or MAC + PHY | |
| PMA Setup-Receiver Subgroup (default values are recommended) | | | | | |
| RX Adaptive Equalization Enable | Not editable | Enabled | RX adaptive EQ capability | _ | |
| RX Adaptive Equalization | | | | | |
| Setting1: Enable | CheckedUnchecked | Checked | _ | _ | |
| Setting1: Adaptive Algorithm | SS_LMS | RL2plus | _ | Editable if Setting1: | |



| Attribute | Selectable Values | Default | Description | Dependency on Other Attributes |
|---|--|-----------|--------------------|--|
| | RL2plus | | | Enable == Checked |
| Setting2: Enable | CheckedUnchecked | Unchecked | _ | _ |
| Setting2: Adaptive Algorithm | SS_LMSRL2plus | RL2plus | _ | Editable if Setting2: Enable == Checked |
| Setting3: Enable | CheckedUnchecked | Unchecked | _ | _ |
| Setting3: Adaptive Algorithm | SS_LMS RL2plus | RL2plus | _ | Editable if Setting3: Enable == Checked |
| Setting1/2: Preliminary Adaptive EQ | EnabledDisabled | Enabled | _ | Editable if Setting1/2: Enable == Checked |
| Setting1/2: Training Phase Adaptive EQ | EnabledDisabled | Enabled | _ | Editable if Setting1/2: Enable == Checked |
| Setting1/2: Post-Phase Adaptive EQ | EnabledDisabled | Enabled | _ | Editable if Setting1/2: Enable == Checked |
| Setting3: Preliminary Adaptive EQ | EnabledDisabled | Disabled | _ | Editable if Setting3: Enable == Checked |
| Setting3: Training Phase Adaptive EQ | EnabledDisabled | Disabled | _ | Editable if Setting3: Enable == Checked |
| Setting3: Post-Phase Adaptive EQ | EnabledDisabled | Disabled | _ | Editable if Setting3: Enable == Checked |
| PHY Configuration | | | 1 | |
| PCS Lane ID | For CertusPro-NX¹ devices: • AUTO • 0 • 1 • 2 • 3 • 4 • 5 | AUTO | Specifies the Lane | Enabled when <i>Select IP Option == PHY Only</i> or <i>MAC + PHY</i> |
| | 67 | | | |

Note:

- 1. CertusPro-NX devices supported lanes:
 - LFCPNX-100 supported lanes: Lane 0, Lane 1, Lane 2, Lane 3, Lane 4, Lane 5, Lane 6, and Lane 7.
 - LFCPNX-50 supported lanes: Lane 0, Lane 1, Lane 2, and Lane 3 only.

3.2. MAC Only

The following table lists the 2.5G Ethernet IP core configurable attributes for the *MAC Only* option. The values set for attributes with corresponding registers serve as the maximum values and cannot set higher than these values during dynamic reconfiguration. Select IP Option—*MAC Only* option.

Table 3.2. MAC Only Attributes

| Table Old William Comy / tetributes | | | | | | | | | | |
|-------------------------------------|-------------------|----------|-------------|-----------------------------------|--|--|--|--|--|--|
| Attribute | Selectable Values | Default | Description | Dependency on Other Attributes | | | | | | |
| General Configuration | | | | | | | | | | |
| Select IP Option | • MAC + PHY | MAC Only | IP option. | _ | | | | | | |

© 2025 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.

All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.



| Attribute | Selectable Values | Default | Description | Dependency on Other Attributes |
|-------------------------------------|---|----------|---|--|
| | PHY OnlyMAC Only | | | |
| Host Interface | Not editable | АРВ | Set the register interface. | _ |
| MAC Configuration | | | | |
| Multicast Address Filtering | EnabledDisabled | Disabled | Enables or disables address filtering for Multicast frames. | Enabled when Select IP Option == MAC Only or MAC + PHY |
| TX Pause Frame Generation via Ports | EnabledDisabled | Disabled | Enables or disables TX pause frame generation via ports. | Enabled when Select IP Option == MAC Only or MAC + PHY |
| Statistics Counter Configurati | on | · | | |
| Statistics Counter Registers | EnabledDisabled | Disabled | Enables or disables statistics counter registers. | Enabled when Select IP Option == MAC Only or MAC + PHY |
| Counter Width | • 32 • 64 | 32 | Statistics counters register size. | Enabled when Select IP Option == MAC Only or MAC + PHY |
| TX Statistics | EnabledDisabled | Disabled | TX statistics. | Enabled when Select IP Option == MAC Only or MAC + PHY |
| RX Statistics | EnabledDisabled | Disabled | RX statistics. | Enabled when Select IP Option == MAC Only or MAC + PHY |

3.3. **PHY Only**

The following table lists the 2.5G Ethernet IP core configurable attributes for the PHY Only option in CertusPro-NX devices. Select IP Option—PHY Only option.

| Attribute | Selectable Values | Default | Description | Dependency on Other Attributes |
|------------------------------------|--|----------|--|--|
| General | <u>. </u> | | | |
| Select IP Option | MAC OnlyPHY OnlyMAC + PHY | MAC Only | IP option. | _ |
| Register Interface | APB MDIO | АРВ | Specifies the preferred interface for register access. | _ |
| Loopback Mode | DisabledNear PMA LoopbackFabric Loopback | Disabled | PCS loopback. For more information, refer to the PCS Loopback section. | _ |
| PMA Setup-Receiver Subgr | oup (default values are recor | nmended) | | |
| RX Adaptive Equalization Enable | Not editable | Enabled | RX adaptive EQ capability | _ |
| RX Adaptive Equalization | <u>. </u> | | | |
| Setting1: Enable | Checked Unchecked | Checked | _ | _ |
| Setting1: Adaptive Algorithm | SS_LMS RL2plus | RL2plus | _ | Editable if Setting1: Enable == Checked |



| Attribute | Selectable Values | Default | Description | Dependency on Other Attributes |
|---|---|-----------|-----------------------|--|
| Setting2: Enable | CheckedUnchecked | Unchecked | _ | - |
| Setting2: Adaptive Algorithm | SS_LMSRL2plus | RL2plus | _ | Editable if Setting2: Enable == Checked |
| Setting3: Enable | CheckedUnchecked | Unchecked | _ | _ |
| Setting3: Adaptive Algorithm | SS_LMSRL2plus | RL2plus | _ | Editable if Setting3: Enable == Checked |
| Setting1/2: Preliminary Adaptive EQ | EnabledDisabled | Enabled | _ | Editable if Setting1/2: Enable == Checked |
| Setting1/2: Training Phase Adaptive EQ | EnabledDisabled | Enabled | _ | Editable if Setting1/2: Enable == Checked |
| Setting1/2: Post-Phase Adaptive EQ | EnabledDisabled | Enabled | _ | Editable if Setting1/2: Enable == Checked |
| Setting3: Preliminary Adaptive EQ | EnabledDisabled | Enabled | _ | Editable if Setting3: Enable == "Checked" |
| Setting3: Training Phase Adaptive EQ | EnabledDisabled | Enabled | _ | Editable if Setting3: Enable == Checked |
| Setting3: Post-Phase Adaptive EQ | EnabledDisabled | Enabled | _ | Editable if Setting3: Enable == Checked |
| PHY Configuration | | | | |
| PCS Lane ID | For CertusPro-NX¹ devices: | AUTO | Specifies the Lane ID | Enabled when <i>Select IP</i> Option == PHY Only or MAC + PHY |

Note:

- 1. CertusPro-NX devices supported lanes:
 - LFCPNX-100 supported lanes: Lane 0, Lane 1, Lane 2, Lane 3, Lane 4, Lane 5, Lane 6, and Lane 7.
 - LFCPNX-50 supported lanes: Lane 0, Lane 1, Lane 2, and Lane 3 only.



4. Signal Description

4.1. MAC + PHY Signals (CertusPro-NX Devices)

Table 4.1. Signal Description—MAC + PHY

| Table 4.1. Signal D Port Name | Clock Domain | 1/0 | Width | Description |
|--------------------------------|-------------------|--------|-------|--|
| Clock and Reset | ı | | | |
| reset_n_i | Asynchronous | In | 1 | Active-low asynchronous reset. |
| phy_tx_clk_o | _ | Out | 1 | TX clock output from PHY (156.25 MHz). |
| phy_rx_clk_o | _ | Out | 1 | RX clock output from PHY (156.25 MHz). |
| xg_tx_gclk_o | _ | Out | 2 | TX clock forwarded to global clock distribution (156.25 MHz). |
| xg_rx_gclk_o | _ | Out | 1 | RX clock forwarded to global clock distribution (156.25 MHz). |
| | _ | | | Clock for the Management module (APB interface). |
| sysbus_clk_i | | In | 1 | It is recommended to use between 20 MHz to 156.25 MHz clock for GMII interface. |
| Other Reference C | lock Source | | | |
| refclkp0_ext_i | _ | In | 1 | 156.25 MHz reference clock from external I/O pad0 (+). |
| refclkn0_ext_i | _ | In | 1 | 156.25 MHz reference clock from external I/O pad0 (+). |
| refclkp1_ext_i | _ | In | 1 | 156.25 MHz reference clock from external I/O pad1 (+). |
| refclkn1_ext_i | _ | In | 1 | 156.25 MHz reference clock from external I/O pad1 (+). |
| pll_0_refclk_i | _ | In | 1 | 156.25 MHz generated reference clock from the Left PLL. |
| pll_1_refclk_i | _ | In | 1 | 156.25 MHz generated reference clock from the Left PLL. |
| sd_pll_refclk_i | _ | In | 1 | 156.25 MHz reference clock from fabric. |
| Reference Clock M | UX Tree Control S | ignals | | |
| use_refmux_i | _ | In | 1 | Refer to the Reference Clocks section. |
| diffioclksel_i | _ | In | 1 | Refer to the Reference Clocks section. |
| clksel_i | _ | In | 2 | Refer to the Reference Clocks section. |
| Clock and Reset | | , | | |
| phy_pcs_clkin_i | _ | In | 1 | This clock also drives the PMA epcs_clkin_i clock port with 156.25 MHz. |
| Serial I/O | | ı | | |
| pad_refclkn_i | _ | In | 1 | 156.25 MHz SERDES PLL reference clock signal (-). |
| pad_refclkp_i | _ | In | 1 | 156.25 MHz SERDES PLL reference clock signal (+). |
| pad_rxn_i | _ | In | 1 | RX- differential signal. |
| pad_rxp_i | _ | In | 1 | RX+ differential signal. |
| pad_txn_o | _ | Out | 1 | TX- differential signal. |
| pad_txp_o | _ | Out | 1 | TX+ differential signal. |
| Client-Side Interfac | ce | | | |
| AXI4-Stream Recei | ve Interface | | | |
| axis_rx_tdata_o | phy_rx_clk_o | Out | 64 | AXI4-Stream data from PHY to the client. |
| | phy_rx_clk_o | | | AXI4-Stream control that indicates which data is valid on axis_rx_tdata_o[]. |
| axis_rx_tkeep_o | | Out | 8 | axis_rx_tkeep_o[0]: axis_rx_tdata_o [7:0] axis_rx_tkeep_o[1]: axis_rx_tdata_o [15:8] axis_rx_tkeep_o[2]: axis_rx_tdata_o [23:16] axis_rx_tkeep_o[3]: axis_rx_tdata_o [31:24] axis_rx_tkeep_o[4]: axis_rx_tdata_o [39:32] axis_rx_tkeep_o[5]: axis_rx_tdata_o [47:40] axis_rx_tkeep_o[6]: axis_rx_tdata_o [55:48] |
| | | | | axis_rx_tkeep_o[7]: axis_rx_tdata_o [63:56] |



| Port Name | Clock Domain | 1/0 | Width | Description |
|-------------------------|---------------|-----|-------|---|
| axis_rx_tvalid_o | phy_rx_clk_o | Out | 1 | AXI4-Stream data valid. |
| axis_rx_tuser_o | phy_rx_clk_o | Out | 1 | AXI4-Stream user signal used to indicate that the frame had a length error, termination error, or a cyclic redundancy check (CRC) error. This signal is qualified with the axis_rx_tlast_o signal. |
| axis_rx_tlast_o | phy_rx_clk_o | Out | 1 | AXI4-Stream signal indicating the end of a packet. |
| AXI4-Stream Trans | mit Interface | | | |
| axis_tx_tready_o | phy_tx_clk_o | Out | 1 | AXI4-Stream signal indicating that the core can accept data transfer. |
| axis_tx_tdata_i | phy_tx_clk_o | In | 64 | AXI4-Stream data from the client. |
| | phy_tx_clk_o | In | | AXI4-Stream control that indicates which data is valid on axis_tx_tdata_i []. • axis_tx_tkeep_i[0]: axis_tx_tdata_i[7:0] |
| axis_tx_tkeep_i | | | 8 | axis_tx_tkeep_i[1]: axis_tx_tdata_i[15:8] axis_tx_tkeep_i[2]: axis_tx_tdata_i[23:16] axis_tx_tkeep_i[3]: axis_tx_tdata_i[31:24] axis_tx_tkeep_i[4]: axis_tx_tdata_i[39:32] axis_tx_tkeep_i[5]: axis_tx_tdata_i[47:40] |
| | | | | axis_tx_tkeep_i[6]: axis_tx_tdata_i[55:48] i |
| | | | | axis_tx_tkeep_i[7]: axis_tx_tdata_i[63:56] |
| axis_tx_tvalid_i | phy_tx_clk_o | In | 1 | AXI4-Stream data valid. |
| axis_tx_tuser_i | phy_tx_clk_o | In | 1 | AXI4-Stream user signal to indicate an error in the frame. This signal is qualified with the axis_tx_tlast_i signal. |
| axis_tx_tlast_i | phy_tx_clk_o | In | 1 | AXI4-Stream signal indicating the end of a packet. |
| Auto-Negotiation | | | | |
| mr_adv_ability | phy_tx_clk_o | In | 16 | Advertised Ability—Configuration status transmitted by PCS during autonegotiation process. |
| mr_an_enable | phy_tx_clk_o | In | 1 | Auto-Negotiation Enable - Active-high signal that enables auto-negotiation state machine to function. |
| mr_main_reset | phy_tx_clk_o | In | 1 | Main Reset—Active-high signal that forces all PCS state machines to reset. |
| mr_restart_an | phy_tx_clk_o | In | 1 | Auto-Negotiation Restart—Active-high signal that forces auto-negotiation process to restart. |
| mr_an_complete | phy_tx_clk_o | Out | 1 | Auto-Negotiation Complete—Active-high signal that indicates that the auto-negotiation process is completed. |
| mr_lp_adv_abilit y | phy_tx_clk_o | Out | 16 | Link Partner Advertised Ability—Configuration status received from partner PCS entity during the auto-negotiation process. The bit definitions are the same as described above for the mr_adv_ability port |
| mr_page_rx | phy_tx_clk_o | Out | 1 | Auto-Negotiation Page Received—Active-high signal that asserts while the auto-negotia-tion state machine is in the <i>Complete_Acknowledge</i> state. |
| xmit_autoneg | phy_tx_clk_o | Out | 1 | Auto-negotiation XMIT Status —This signal should be tied to the "xmit_chn" pin of the SERDES. When the signal is high, it forces the Rx data path of the SERDES to periodically insert idle-code groups into the SERDES Rx data stream. This ensures that the SERDES CTC has opportunities to rate-adapt for ppm offsets. The 2.5G Ethernet PCS IP core drives this signal high when the auto-negotiation process is running. At all other times, the IP core drives this signal low. |
| force_isolate | phy_tx_clk_o | In | 1 | Force PCS Isolate —Active-high signal that isolates the PCS. When asserted, the RX direction forces the GMII port to all zeros, regardless of the condition of the incoming 3.125 Gbps serial data stream. In the TX direction, the condition of the incoming GMII port is ignored. The TX PCS behaves as though the GMII TX input port was forced to all zeros. Note, however, that the isolate function does not produce any electrical isolation — such as tri-stating of the GMII RX outputs of the IP core. When the signal is deasserted (low), the PCS isolation functions are deactivated. The use of this signal is optional. If you choose not to use the isolate |



| force_unidir | phy_tx_clk_o phy_tx_clk_o | In In | 1 | function, then this signal must be tied low. Force PCS Loopback—Active-high signal that activates the PCS loopback function. When asserted, the 8-bit code-group output of the transmit state machine is looped back to the 8- bit code-group input of the receive state machine. When deasserted, the loopback function is deactivated. The use of this signal is optional. If you chooses not to use the loopback function, then this signal must be tied low. Force PCS Unidirectional Mode—Active-high signal that activates the PCS unidirectional mode. When asserted, the transmit state machine path between the TX GMII input and the TX 8-bit code-group output will remain operational, regardless of what happens on the RX datapath. (Normally RX loss of sync, invalid code-group reception, auto-negotiation restarts can force the transmit state machine to temporarily ignore inputs from the TX GMII port). When deasserted, the unidirectional mode is deactivated. The use of this signal is optional. If you choose not to use the unidirectional |
|----------------------------|----------------------------|-------|----|---|
| force_unidir | phy_tx_clk_o | In | | function. When asserted, the 8-bit code-group output of the transmit state machine is looped back to the 8- bit code-group input of the receive state machine. When deasserted, the loopback function is deactivated. The use of this signal is optional. If you chooses not to use the loopback function, then this signal must be tied low. Force PCS Unidirectional Mode—Active-high signal that activates the PCS unidirectional mode. When asserted, the transmit state machine path between the TX GMII input and the TX 8-bit code-group output will remain operational, regardless of what happens on the RX datapath. (Normally RX loss of sync, invalid code-group reception, auto-negotiation restarts can force the transmit state machine to temporarily ignore inputs from the TX GMII port). When deasserted, the unidirectional mode is deactivated. The use of this signal is optional. If you choose not to use the unidirectional |
| | | | 1 | unidirectional mode. When asserted, the transmit state machine path between the TX GMII input and the TX 8-bit code-group output will remain operational, regardless of what happens on the RX datapath. (Normally RX loss of sync, invalid code-group reception, auto-negotiation restarts can force the transmit state machine to temporarily ignore inputs from the TX GMII port). When deasserted, the unidirectional mode is deactivated. The use of this signal is optional. If you choose not to use the unidirectional |
| an_link_ok | phy_tx_clk_o | Out | | function, then this signal must be tied low. |
| | | - | 1 | Auto-Negotiation Link Status OK—Active-high signal that indicates that the link is OK. The signal is driven by the auto-negotiation state machine. When auto-negotiation is enabled, the signal asserts when the state machine is in the LINK_OK state. If auto-negotiation is disabled, the signal asserts when the state machine is in the AN_DISABLE_LINK_OK state. See IEEE 802.3 figure 37-6. |
| Non-Standard RX/TX | Signals | | | |
| debug_link_timer _short | phy_tx_clk_i | In | 1 | Debug Link Timer Mode—Active-high signal that forces the autonegotiation link timer to run much faster than normal. This mode is provided for debug purposes. For example, allowing simulations to run through the auto-negotiation process much faster than normal. |
| phy_rxval_o | phy_rx_clk_o | Out | 1 | This signal is used to signal receiving valid data. |
| phy_txrdy_o phy_txrdy_o | phy_pcs_clkin _i | Out | 1 | PHY ready: This signal is asserted when the PHY has completed the calibration sequence for each specific lane. This signal is driven by phy_pcs_clkin_i. |
| phyrdy_o | phy_tx_clk_o | Out | 1 | When asserted, this signal tells you that the PHY is ready to transmit. |
| phy_link_sync_o | phy_rx_clk_o | Out | 1 | Link Synchronization output port. 1'b1 – link synchronization is acquired. 1'b0 – loss of link synchronization. |
| phy_rxidle_o | phy_rx_clk_o | Out | 1 | This port is used to signal the Electrical Idle condition detected by the PMA control logic. |
| APB Interface ¹ | | | | |
| apb_psel_i | sysbus_clk_i | In | 1 | Select signal. Indicates that the completer device is selected, and a data transfer is required. |
| apb_paddr_i | sysbus_clk_i | In | 32 | Address signal. |
| | sysbus_clk_i | In | 32 | Write data signal. |
| | sysbus_clk_i | In | 1 | Direction signal. Write = 1, Read = 0. |
| apb_penable_i | sysbus_clk_i | In | 1 | Enable signal. Indicates the second and subsequent cycles of an APB transfer. |
| apb_pready_o | sysbus_clk_i | Out | 1 | Ready signal. Indicates transfer completion. The completer uses this signal to extend an APB transfer. |
| apb_prdata_o | sysbus_clk_i | Out | 32 | Read data signal. |
| apb_pslverr_o | sysbus_clk_i | Out | 1 | Transfer success or failure signal. Tie to low. |
| Statistics Vector Inter | rface | | | |
| tx_statvec_o | phy_tx_clk_o | Out | 28 | Contains information on the frame transmitted. This bus is qualified by the tx_staten_o signal. tx_statvec_o[13:0]: Frame byte count. |



| Port Name | Clock Domain | 1/0 | Width | Description |
|-----------------------------|--------------|-----|-------|---|
| | | | | tx_statvec_o[14]: Transmit is OK. |
| | | | | tx_statvec_o[15]: MAC control inserted by MAC. |
| | | | | tx_statvec_o[16]: MAC control inserted by client. |
| | | | | tx_statvec_o[17]: Jumbo frame. |
| | | | | tx_statvec_o[18]: Tagged frame. |
| | | | | tx_statvec_o[19]: Broadcast address. |
| | | | | tx_statvec_o[20]: Multicast address. |
| | | | | tx_statvec_o[21]: Underrun error. |
| | | | | tx_statvec_o[22]: CRC error. |
| | | | | tx_statvec_o[23]: Length check error. |
| | | | | tx_statvec_o[24]: Terminate error. |
| | | | | tx_statvec_o[25]: Long frame error. |
| | | | | tx_statvec_o[26]: PTP1588 frame. |
| | | | | tx_statvec_o[27]: Reserved for future use. |
| ty states - | phy_tx_clk_o | 0 | 4 | When asserted, it indicates that the contents of the tx_statvec_o bus are |
| tx_staten_o | | Out | 1 | valid. This signal is asserted for three phy_tx_clk_o periods. |
| | phy_rx_clk_o | | | Contains information on the frame received. This bus is qualified by the |
| | | | | rx_staten_o signal. |
| | | | | rx_statvec_o[13:0]: Frame byte count. |
| | | | | rx_statvec_o[14]: Frame dropped. |
| | | | | rx_statvec_o[15]: Broadcast frame received. |
| | | | | rx_statvec_o[16]: Multicast frame received. |
| | | | | rx_statvec_o[17]: CRC error. |
| | | | | rx_statvec_o[18]: VLAN tag detected. |
| rx statvec o | | Out | 28 | rx_statvec_o[19]: Pause frame. |
| TX_3tatvec_0 | | Out | 20 | rx_statvec_o[20]: Length check error. |
| | | | | rx_statvec_o[21]: Frame is too long. |
| | | | | rx_statvec_o[22]: MAC address mismatch. |
| | | | | rx_statvec_o[23]: Unsupported opcode. Only the opcode for pause frame is supported. |
| | | | | rx_statvec_o[24]: Minimum IPG violated. |
| | | | | rx_statvec_o[25]: Receive packet ignored. |
| | | | | rx_statvec_o[26]: PTP1588 frame received. |
| | | | | rx_statvec_o[27]: Reserved for future use. |
| | phy_rx_clk_o | _ | | When asserted, indicates that the contents of the rx_statvec_o bus are |
| rx_staten_o | | Out | 1 | valid. This signal is asserted for three phy_rx_clk_o periods. |
| TX Pause Frame ² | | | | |
| tx_pausreq_i | phy_tx_clk_o | In | 1 | Transmit pause frame. |
| | | | | 1 = send request, 0 = do not send request. This is a positive edge-triggered |
| | | | | bit. |
| | | | | The tx_fc_en bit of TX_CTL register should be set to 1 to enable this feature. |
| | | | | When this port is enabled, the tx_pausreq bit of MAC_CTL register is not accessible. |
| | | | | Subsequent pause request will not be serviced if the pause frame of the first request is not transmitted yet. |
| tx_paustm_i | phy_tx_clk_o | In | 16 | Pause Quanta time. |
| · | ,,oo | | | This is equivalent to the tx_paustim bit of PAUSE_TM register. |
| | L | L | l | |

Notes:

- Only APB interface is supported for IP_OPTION = MAC + PHY.
- 2. Available when TX Pause Frame Generation via Ports is enabled.



4.2. MAC Only Signals

Table 4.2. Signal Description—MAC Only

| Port Name | Clock Domain | 1/0 | Width | Description |
|-----------------------------|-----------------|-----|-------|---|
| Clock and Reset | | | | |
| reset_n_i | Asynchronous | In | 1 | Active-low asynchronous reset. |
| | | | 4 | Clock for Receive AXI4-Stream and 16-bit GMII. |
| rxmac_clk_i | | ln | 1 | 156.25 MHz clock for 16-bit interface. |
| txmac_clk_i | _ | In | 1 | Clock for Transmit AXI4-Stream and GMII datapath. |
| txiiiac_cik_i | | "" | - | 156.25 MHz clock for 16-bit interface. |
| sysbus_clk_i | _ | In | 1 | Clock for Management module (APB interface). It is recommended to use between 20 MHz to 156.25 MHz clock for 16-GMII interface. |
| PHY Interface | | | | |
| GMII Interface ² | | | | |
| gmii_16_rx_d_i | rxmac_clk_i | In | 16 | 16-bit GMII RX data. |
| gmii_16_rx_dv _i | rxmac_clk_i | In | 2 | Indicates the GMII RX data is valid when asserted. |
| gmii_16_rx_err _i | rxmac_clk_i | In | 2 | Indicates the GMII RX data contains error. |
| gmii_16_tx_d_ o | txmac_clk_i | Out | 16 | 16-bit GMII TX data. |
| gmii_16_tx_en _o | txmac_clk_i | Out | 2 | Indicates the GMII TX data is valid when asserted. |
| gmii_16_tx_err _o | txmac_clk_i | Out | 2 | Indicates the GMII TX data contains error. |
| Client-Side Inter | face | | I. | |
| AXI4-Stream Rec | eive Interface | | | |
| axis_rx_tdata_ o | rxmac_clk_i | Out | 64 | AXI4-Stream data from PHY to client. |
| axis_rx_tkeep_ o | rxmac_clk_i | Out | 8 | AXI4-Stream control that indicates which data is valid on axis_rx_tdata_o[]. axis_rx_tkeep_o[0]: axis_rx_tdata_o [7:0] axis_rx_tkeep_o[1]: axis_rx_tdata_o [15:8] axis_rx_tkeep_o[2]: axis_rx_tdata_o [23:16] axis_rx_tkeep_o[3]: axis_rx_tdata_o [31:24] axis_rx_tkeep_o[4]: axis_rx_tdata_o [39:32] axis_rx_tkeep_o[5]: axis_rx_tdata_o [47:40] axis_rx_tkeep_o[6]: axis_rx_tdata_o [55:48] axis_rx_tkeep_o[7]: axis_rx_tdata_o [63:56] |
| axis_rx_tvalid_ o | rxmac_clk_i | Out | 1 | AXI4-Stream data valid. |
| axis_rx_tuser_ o | rxmac_clk_i | Out | 1 | AXI4-Stream user signal used to indicate that the frame had a length error, termination error, or a CRC error. This signal is qualified with the axis_rx_tlast_o signal. |
| axis_rx_tlast_o | rxmac_clk_i | Out | 1 | AXI4-Stream signal indicating the end of a packet. |
| AXI4-Stream Tra | nsmit Interface | | | |
| axis_tx_tready _o | txmac_clk_i | Out | 1 | AXI4-Stream signal indicating that the core can accept data transfer. |
| axis_tx_tdata_i | txmac_clk_i | In | 64 | AXI4-Stream data from client. |
| axis_tx_tkeep_ i | txmac_clk_i | In | 8 | AXI4-Stream control that indicates which data is valid on axis_tx_tdata_i []. axis_tx_tkeep_i[0]: axis_tx_tdata_i[7:0] axis_tx_tkeep_i[1]: axis_tx_tdata_i[15:8] |



| Port Name | Clock Domain | 1/0 | Width | Description |
|----------------------------|--------------|-----|-------|--|
| | | | | axis_tx_tkeep_i[2]: axis_tx_tdata_i[23:16] |
| | | | | axis_tx_tkeep_i[3]: axis_tx_tdata_i[31:24] |
| | | | | axis_tx_tkeep_i[4]: axis_tx_tdata_i[39:32] |
| | | | | axis_tx_tkeep_i[5]: axis_tx_tdata_i[47:40] |
| | | | | axis_tx_tkeep_i[6]: axis_tx_tdata_i[55:48] |
| | | | | axis_tx_tkeep_i[7]: axis_tx_tdata_i[63:56] |
| axis_tx_tvalid_i | txmac_clk_i | In | 1 | AXI4-Stream data valid. |
| axis_tx_tuser_i | txmac_clk_i | In | 1 | AXI4-Stream user signal used to indicate an error in the frame. This signal is qualified with the axis_tx_tlast_i signal. |
| axis_tx_tlast_i | txmac_clk_i | In | 1 | AXI4-Stream signal indicating the end of a packet. |
| APB Interface ⁵ | | L | | |
| apb_psel_i | sysbus_clk_i | In | 1 | Select signal. Indicates that the completer device is selected and a data transfer is required. |
| apb_paddr_i | sysbus_clk_i | In | 32 | Address signal. |
| apb_pwdata_i | sysbus_clk_i | In | 32 | Write data signal. |
| apb_pwrite_i | sysbus clk i | In | 1 | Direction signal. Write = 1, Read = 0. |
| apb_penable_i | sysbus_clk_i | In | 1 | Enable signal. Indicates the second and subsequent cycles of an APB transfer. |
| apb_pready_o | sysbus_clk_i | Out | 1 | Ready signal. Indicates transfer completion. The completer uses this signal to extend an APB transfer. |
| apb_prdata_o | sysbus_clk_i | Out | 32 | Read data signal. |
| apb_pslverr_o | sysbus_clk_i | Out | 1 | Transfer success or failure signal. Tie to low. |
| Statistics Vector | | 1 | | · · · · · · · · · · · · · · · · · · · |
| | txmac_clk_i | | | Contains information on the frame transmitted. This bus is qualified by the |
| tx_statvec_o | | Out | 28 | tx_statvec_o[13:0]: Frame byte count tx_statvec_o[14]: Transmit is OK tx_statvec_o[15]: MAC control inserted by MAC tx_statvec_o[16]: MAC control inserted by Client tx_statvec_o[17]: Jumbo frame tx_statvec_o[18]: Tagged frame tx_statvec_o[19]: Broadcast address tx_statvec_o[20]: Multicast address tx_statvec_o[21]: Underrun error tx_statvec_o[22]: CRC error tx_statvec_o[23]: Length check error tx_statvec_o[24]: Terminate error tx_statvec_o[25]: Long Frame error tx_statvec_o[26]: PTP1588 frame tx_statvec_o[27]: Reserved for future use |
| tx_staten_o | txmac_clk_i | Out | 1 | When asserted, it indicates that the contents of the tx_statvec_o bus are valid. This signal is asserted for three txmac_clk_i periods. |
| rx_statvec_o | rxmac_clk_i | Out | 28 | Contains information on the frame received. This bus is qualified by the rx_staten_o signal. rx_statvec_o[13:0]: Frame byte count rx_statvec_o[14]: Frame dropped rx_statvec_o[15]: Broadcast frame received rx_statvec_o[16]: Multicast frame received rx_statvec_o[17]: CRC error rx_statvec_o[18]: VLAN Tag detected rx_statvec_o[19]: PAUSE frame |



| Port Name | Clock Domain | 1/0 | Width | Description |
|----------------|--------------|-----|-------|---|
| | | | | rx_statvec_o[20]: Length check error rx_statvec_o[21]: Frame is too long rx_statvec_o[22]: MAC Address mismatch rx_statvec_o[23]: Unsupported opcode. Only the opcode for PAUSE frame is supported. rx_statvec_o[24]: Minimum IPG violated rx_statvec_o[25]: Receive packet ignored rx_statvec_o[26]: PTP1588 frame received rx statvec_o[27]: Reserved for future use |
| rx_staten_o | rxmac_clk_i | Out | 1 | When asserted, indicates that the contents of the rx_statvec_o bus are valid. This signal is asserted for three rxmac_clk_i periods. |
| pcsrdy_i | rxmac_clk_i | In | 1 | This is an active-high level status signal from external PHY. Asserted when the PHY link status is ready. In cases where the local fault/remote fault is sending an earlier signal such as during the reset state, then this signal is used by MAC to determine the PHY link status when out of reset. As a result, the RX_STAT_LINK_OK counter is increased. |
| TX Pause Frame | 5 | • | | |
| tx_pausreq_i | txmac_clk_i | In | 1 | Transmit PAUSE frame. 1 = send request, 0 = do not send request. This is a positive edge-triggered bit. The tx_fc_en bit of TX_CTL register should be set to 1 to enable this feature. When this port is enabled, the tx_pausreq bit of MAC_CTL register is not accessible. Subsequent pause request will not be serviced if the pause frame of the first request is not transmitted yet. |
| tx_paustm_i | txmac_clk_i | In | 16 | Pause Quanta time. This is equivalent to the tx_paustim bit of the PAUSE_TM register. |

Notes:

- 1. Available when APB is selected as Host Interface.
- 2. Available when TX Pause Frame Generation via Ports is enabled.

4.3. PHY Only Signals (CertusPro-NX Devices)

Table 4.3. Signal Description—PHY Only (CertusPro-NX Devices)

| Port Name | Clock Domain | 1/0 | Width | Description | | | | | |
|------------------------------|-----------------|-----|-------|--|--|--|--|--|--|
| Serial I/O | Serial I/O | | | | | | | | |
| pad_refclkn_i | _ | In | 1 | 156.25 MHz SERDES PLL reference clock signal (-). | | | | | |
| pad_refclkp_i | _ | In | 1 | 156.25 MHz SERDES PLL reference clock signal (+). | | | | | |
| pad_rxn_i | pad_refclkn_i | In | 1 | RX- differential signal. | | | | | |
| pad_rxp_i | pad_refclkp_i | In | 1 | RX+ differential signal. | | | | | |
| pad_txn_o | pad_refclkn_i | Out | 1 | TX- differential signal. | | | | | |
| pad_txp_o | pad_refclkp_i | Out | 1 | TX+ differential signal. | | | | | |
| Other Reference Clock Source | | | | | | | | | |
| refclkn0_ext_i | refclkn0_ext_i | In | 1 | 156.25 MHz reference clock from external I/O pad0 (-). | | | | | |
| refclkp0_ext_i | refclkp0_ext_i | In | 1 | 156.25 MHz reference clock from external I/O pad0 (+) | | | | | |
| refclkn1_ext_i | refclkn1_ext_i | In | 1 | 156.25 MHz reference clock from external I/O pad1 (-). | | | | | |
| refclkp1_ext_i | refclkp1_ext_i | In | 1 | 156.25 MHz reference clock from external I/O pad1 (+). | | | | | |
| pll_0_refclk_i | pll_0_refclk_i | In | 1 | 156.25 MHz reference clock from Left PLL. | | | | | |
| pll_1_refclk_i | pll_1_refclk_i | In | 1 | 156.25 MHz reference clock from Right PLL. | | | | | |
| sd_pll_refclk_i | sd_pll_refclk_i | In | 1 | 156.25 MHz reference clock from fabric. | | | | | |



| Port Name | Clock Domain | I/O | Width | Description |
|--------------------------------|----------------------|-----|-------|---|
| Reference Clock MUX | Tree Control Signals | • | | |
| use_refmux_i | _ | In | 1 | Refer to the Reference Clocks section. |
| diffioclksel_i | _ | In | 1 | Refer to the Reference Clocks section. |
| clksel_i | _ | In | 2 | Refer to the Reference Clocks section. |
| Clock and Reset | | | | |
| phy_tx_clk_i | _ | In | 1 | TX clock input – 156.25 MHz clock for transmit datapath. |
| phy_tx_rst_n_i | Asynchronous | In | 1 | TX active low reset. Asynchronous assert, synchronous (phy_tx_clk_i) deassert. |
| phy_rx_clk_i | _ | In | 1 | RX clock input – 156.25 MHz clock for receive datapath. |
| phy_rx_rst_n_i | Asynchronous | In | 1 | RX active low reset. Asynchronous assert, synchronous (phy_rx_clk_i) deassert. |
| phy_pcs_clkin_i | _ | In | 1 | Clock input to PMA with 156.25 MHz. |
| phy_tx_clk_o | _ | Out | 1 | TX clock output – 156.25 MHz. |
| phy_rx_clk_o | _ | Out | 1 | RX clock output – 156.25 MHz. |
| 16-bit GMII | <u> </u> | | | |
| gmii_16_tx_en_i gmii_16_tx_d_i | phy_tx_clk_i | In | 16 | 2-bit active-high signal that asserts when incoming GMII data is valid. bit[0] is the valid signal for gmii_16_tx_d_i [7:0] bit[1] is the valid signal for gmii_16_tx_d_i [15:8] gmii_16_tx_en_i [1] is associated with the upper byte of the GMII data (bits 15:8). gmii_16_tx_en_i [0] is associated with the lower byte of the GMII data (bits 7:0). Note that when an Ethernet frame is being transmitted, most of the time both transmit enable bits are asserted simultaneously. However, at the beginning and end of the frame, it is possible for only one of the two GMII data bytes to be valid. A GMII data frame can begin and end on any byte (upper or lower). |
| | | 1 | 2 | |
| gmii_16_tx_err_i | phy_tx_clk_i | In | 2 | Transmit Error—2-bit, active-high signal that denotes transmission errors or carrier extension events on the GMII Tx data port. gmii_16_tx_err_i [1] is associated with the upper byte of the GMII data. gmii_16_tx_err_i [0] is associated with the lower byte of the GMII data. |
| gmii_16_rx_dv_o | phy_rx_clk_i | Out | 2 | Receive Data Valid—2-bit, active-high signal that asserts when outgoing GMII data is valid. gmii_16_rx_dv_o [1] is associated with the upper byte of the GMII data (bits 15:8). gmii_16_rx_dv_o [0] is associated with the lower byte of the GMII data (bits 7:0). Note that when an Ethernet frame is being received, most of the time both receive valid bits are asserted simultaneously. However, at the beginning and end of the frame, it is possible for only one of the two GMII data bytes to be valid. A GMII data frame can begin and end on any byte (upper or lower). |
| gmii_16_rx_d_o | phy_rx_clk_i | Out | 16 | 16-bit RX data signal. |
| gmii_16_rx_err_o | phy_rx_clk_i | Out | 2 | Receive Error—2-bit, active-high signal that denotes transmission errors or carrier extension events on the GMII Rx data port. gmii_16_rx_err_o [1] is associated with the upper byte of the GMII data. |



| Port Name | Clock Domain | 1/0 | Width | Description |
|-------------------|--------------|-----|-------|---|
| | | | | gmii_16_rx_err_o [0] is associated with the lower byte of the GMII data. |
| Auto-Negotiation | · | | | |
| mr_adv_ability | phy_tx_clk_i | In | 16 | Advertised Ability—Configuration status transmitted by PCS during auto-negotiation process. |
| mr_an_enable | phy_tx_clk_i | In | 1 | Auto-Negotiation Enable—Active-high signal that enables autonegotiation state machine to function. |
| mr_main_reset | phy_tx_clk_i | In | 1 | Main Reset—Active-high signal that forces all PCS state machines to reset. |
| mr_restart_an | phy_tx_clk_i | In | 1 | Auto-Negotiation Restart—Active-high signal that forces auto- negotiation process to restart. |
| mr_an_complete | phy_tx_clk_i | Out | 1 | Auto-Negotiation Complete—Active-high signal that indicates that the auto-negotiation process is completed. |
| mr_lp_adv_ability | phy_tx_clk_i | Out | 16 | Link Partner Advertised Ability Configuration status received from partner PCS entity during the auto-negotiation process. The bit definitions are the same as described above for the mr_adv_ability port. |
| mr_page_rx | phy_tx_clk_i | Out | 1 | Auto-Negotiation Page Received—Active-high signal that asserts while the auto-negotiation state machine is in the <i>Complete_Acknowledge</i> state. |
| xmit_autoneg | phy_tx_clk_i | Out | 1 | Auto-negotiation XMIT Status – This signal should be tied to the "xmit_chn" pin of the SerDes. When the signal is high, it forces the Rx data path of the SerDes to periodically insert idlecode groups into the SerDes RX data stream. This ensures that SerDes CTC has opportunities to rate-adapt for ppm offsets. The 2.5G Ethernet PCS IP core drives this signal high when the auto-negotiation process is running. At all other times, the IP core drives this signal low. |
| force_isolate | phy_tx_clk_i | In | 1 | Force PCS Isolate – Active-high signal that isolates the PCS. When asserted, the RX direction forces the GMII port to all zeros, regardless of the condition of the incoming 3.125 Gbps serial data stream. In the TX direction, the condition of the incoming GMII port is ignored. The TX PCS behaves as though the GMII TX input port was forced to all zeros. Note that the isolate function does not produce any electrical isolation – such as tri-stating of the GMII RX outputs of the IP core. When the signal is deasserted (low), the PCS isolation functions are deactivated. The use of this signal is optional. If you choose not to use the isolate function, then this signal must be tied low. |
| force_loopback | phy_tx_clk_i | In | 1 | Force PCS Loopback — Active-high signal that activates the PCS loopback function. When asserted, the 8-bit code-group output of the transmit state machine is looped back to the 8- bit codegroup input of the receive state machine. When deasserted, the loopback function is deactivated. The use of this signal is optional. If you choose not to use the loopback function, then this signal must be tied low. |
| force_unidir | phy_tx_clk_i | In | 1 | Force PCS Unidirectional Mode – Active-high signal that activates the PCS unidirectional mode. When asserted, the transmit state machine path between the TX GMII input and the TX 8-bit code-group output will remain operational, regardless of what happens on the RX datapath. (Normally RX loss of sync, invalid code-group reception, auto-negotiation restarts can force the transmit state machine to temporarily ignore inputs from the TX GMII port). When deasserted, the unidirectional mode is deactivated. The use of this signal is optional. If you choose not to use the unidirectional function, |



| Port Name | Clock Domain | 1/0 | Width | Description |
|----------------------------|---------------------|--------|-------|---|
| | | | | then this signal must be tied low. |
| an_link_ok | phy_tx_clk_i | Out | 1 | Auto-Negotiation Link Status OK — Active-high signal that indicates that the link is OK. The signal is driven by the autonegotiation state machine. When auto-negotiation is enabled, the signal asserts when the state machine is in the LINK_OK state. If auto-negotiation is disabled, the signal asserts when the state machine is in the AN_DISABLE_LINK_OK state. See IEEE 802.3 figure 37-6. |
| Non-Standard RX/TX Sign | als | | | |
| debug_link_timer_short | phy_tx_clk_i | In | 1 | Debug Link Timer Mode – Active-high signal that forces the auto-negotiation link timer to run much faster than normal. This mode is provided for debug purposes. For example, allowing simulations to run through the auto-negotiation process much faster than normal. |
| phy_rxval_o | phy_rx_clk_o | Out | 1 | This signal is used to signal receiving valid data. |
| phy_txrdy_o | phy_pcs_clkin_i | Out | 1 | PHY ready: This signal is asserted when the PHY has completed the calibration sequence for each specific lane. This signal is driven by phy_pcs_clkin_i. |
| phyrdy_o | phy_tx_clk_o | Out | 1 | When asserted, this signal tells you that the PHY is ready to transmit while using mpcs_txval_i. |
| phy_link_sync_o | phy_rx_clk_o | Out | 1 | Link Synchronization output port. 1'b1 – link synchronization is acquired. 1'b0 – loss of link synchronization. |
| phy_rxidle_o | phy_rx_clk_o | Out | 1 | This port is used to signal the Electrical Idle condition detected by the PMA control logic. |
| APB Interface ¹ | | | • | |
| apb_pclk_i | sysbus_clk_i | In | 1 | Clock for Management module (APB interface). It is recommended to use between 20 MHz to 156.25 MHz clock. |
| apb_preset_n_i | sysbus_clk_i | In | 1 | Active low reset. Asynchronous assert, synchronous (apb_pclk_i) deassert. |
| apb_psel_i | sysbus_clk_i | In | 1 | Completer select. |
| apb_penable_i | sysbus_clk_i | In | 1 | Enable. This signal indicates the second and subsequent cycles of an APB transfer. |
| apb_paddr_i | sysbus_clk_i | In | 16 | Address. |
| apb_pwdata_i | sysbus_clk_i | In | 16 | Write Data. |
| apb_pwrite_i | sysbus_clk_i | In | 1 | Indicates write or read access. 0 – Read. 1 – Write. |
| apb_prdata_o | sysbus_clk_i | Out | 16 | Read Data. |
| apb_pready_o | sysbus_clk_i | Out | 1 | Ready. The completer uses this signal to extend an APB transfer. |
| Management Data Input/ | Output ¹ | | | |
| mdc_i | mdc_i | In | 1 | MDIO clock signal (2.5 MHz maximum frequency). |
| md_resetn_i | _ | In | 1 | MDIO active low reset signal. |
| mdio_io | mdc_i | In/Out | 1 | MDIO bidirectional data signal. |

Note:

1. Only one of the two interfaces is available as selected by the *Register Interface*.



5. Register Description

5.1. MAC + PHY Registers (CertusPro-NX Devices)

The registers available in the Ethernet MAC + PHY option include all registers from the MAC and PHY.

For register descriptions in the 2.5G MAC and PHY, refer to the MAC Registers and PHY Registers (CertusPro-NX Devices) sections.

Note: With the MAC + PHY selected as the IP option, only the APB interface is available for access to both PCS and MAC registers. MMD registers are not accessible with this IP option.

5.1.1. Register Address Mapping

The following table lists the address decoding for the MAC and PHY blocks. The base address must be adjusted accordingly to access each individual registers in the respective blocks.

Table 5.1. Register Address Mapping

| Block | APB Base Address |
|-------|------------------|
| MAC | 0x0000 – 0x3FFF |
| PHY | 0x5000 onwards |

5.2. MAC Registers

All registers are accessed through the APB interface when the IP is configured as MAC Only.

Table 5.2. Access Types for MAC Only

| Access Type | Behavior on Read Access | Behavior on Write Access |
|-------------|-------------------------|---|
| RO | Returns register value. | Ignores write access. |
| WO | Returns 0. | Updates register value. |
| RW | Returns register value. | Updates register value. |
| RW1C | Returns register value. | Writing 1'b1 on register bit clears the bit to 1'b0. Writing 1'b0 on register bit is ignored. |
| Reserved | Returns 0. | Ignores write access. |

5.2.1. Configuration Registers for MAC

The following table lists the 2.5G MAC configuration registers.

Table 5.3. Configuration Registers for MAC

| Offset | Register Name | Access | Description |
|--------|---------------|--------|---------------------------------------|
| 0x000 | MODE | RW | Mode of Operation Register. |
| 0x004 | TX_CTL | RW | Transmit MAC Control Register. |
| 0x008 | RX_CTL | RW | Receive MAC Control Register. |
| 0x00C | MAX_PKT_LNGTH | RW | Maximum Packet Size Register. |
| 0x014 | MAC_ADDR_0 | RW | MAC Address Register Word 0. |
| 0x018 | MAC_ADDR_1 | RW | MAC Address Register Word 1. |
| 0x01C | TX_RX_STS | RO | Transmit and Receive Status Register. |
| 0x020 | VLAN_TAG | RO | VLAN Tag Register. |
| 0x024 | MC_TABLE_0 | RW | Multicast Tables Register Word 0. |
| 0x028 | MC_TABLE_1 | RW | Multicast Tables Register Word 1. |
| 0x02C | PAUSE_OPCODE | RW | Pause Opcode . |
| 0x030 | MAC_CTL | RW | MAC Control Register. |

© 2025 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.

All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

FPGA-IPUG-02293-1.2



| Offset | Register Name | Access | Description |
|--------|---------------|--------|---------------------|
| 0x034 | PAUSE_TM | RW | Pause Time Register |

5.2.1.1. [0x000] MODE Register

This register enables the operation of the MAC. It can be written at any time.

Table 5.4. [0x000] MODE Register

| Bit | Name | Description | Default |
|--------|----------|--|---------|
| [31:2] | Reserved | Reserved bits. | 0 |
| [1] | tx_en | TX MAC Enable When set to 1, the TX MAC is enabled to transmit frames. | 0 |
| [0] | rx_en | RX MAC Enable When set to 1, the RX MAC is enabled to receive frames. | 0 |

5.2.1.2. [0x004] TX_CTL Register

This register must be overwritten only when the TX MAC is disabled. Writing this register while TX MAC is active results in unpredictable behavior.

Table 5.5. [0x004] TX_CTL Register

| Bit | Name | Description | Default |
|--------|----------------|---|---------|
| [31:4] | Reserved | Reserved bits. | 0 |
| [3] | transmit_short | Transmit Short Frames. When set to 1, the TX MAC transmits frames shorter than 64 bytes without adding padding bytes. When set to 0, TX MAC adds padding bytes when frames are shorter than 64 bytes before transmitted to the PHY. | 0 |
| [2] | Reserved | Reserved bits. | 0 |
| [1] | tx_fc_en | Flow-control Enable. When set to 1, this enables the flow control functionality of the TX MAC. This bit must be set for the TX MAC to transmit a pause frame. | 0 |
| [0] | tx_pass_fcs | In-band FCS Enable. When set to 1, the FCS field generation is disabled in the TX MAC, and the client is responsible to generate the appropriate FCS field. | 0 |

5.2.1.3. [0x008] RX_CTL Register

This register must be overwritten only when the RX MAC is disabled. Writing this register while RX MAC is active results in unpredictable behavior.

Table 5.6. [0x008] RX_CTL Register

| Bit | Name | Description | Default |
|--------|----------------|--|---------|
| [15:7] | Reserved | Reserved bits. | 0 |
| [6] | drop_mac_ctrl | Drop MAC Control Frames. When set to 1, all MAC control frames are not passed on to the client interface. | 0 |
| [5] | receive_short | Receive Short Frames. When set to 1, it enables the RX MAC to receive frames shorter than 64 bytes. | 0* |
| [4] | receive_bc | Receive Broadcast Frames. When set to 1, it enables the RX MAC to receive broadcast frames. | 0 |
| [3] | receive_all_mc | Receive Multicast Frames. When set to 1, the multicast frames are received per the filtering rules for such frames. When set to 0, no multicast (except pause frames) frames are received. | 0 |
| [2] | rx_pause_en | Receive Pause Frames. | 0 |

© 2025 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.

All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.



| Bit | Name | Description | Default |
|-----|-------------|---|---------|
| | | When set to 1, the RX MAC indicates the pause frame reception to the TX MAC. Pause frames are received and transferred to the AXI4-Stream interface only when the drop_mac_ctrl bit is not set. | |
| [1] | rx_pass_fcs | In-band FCS Passing. When set to 1, the FCS and any of the padding bytes are passed to the AXI4-Stream interface. When set to 0, the MAC strips off the FCS and any padding bytes before transferring it to the AXI4-Stream interface. | 0 |
| [0] | prms | Promiscuous Mode. When set to 1, all filtering schemes are abandoned, and the RX MAC receives frames with any address. | 0 |

^{*} Note: If the L/T value is less than 46 bytes, it detects as short frame, and it will not be dropped. If the L/T value is less than 46 bytes but the payload is more than the defined value, then the extra payload will be treated as the padded byte.

5.2.1.4. [0x00C] MAX_PKT_LNGTH Register

This register must be overwritten only when the MAC is disabled. All frames longer than the value (number of bytes) in this register is tagged as long frames. Writing this register while the MAC is active results in unpredictable behavior.

Table 5.7. [0x00C] MAX_PKT_LNGTH Register

| Bit | Name | Description | Default |
|---------|-------------|---|---------|
| [31:14] | Reserved | Reserved bits. | 0 |
| [13:0] | max_pkt_len | Maximum Frame Length. Used only for statistical purposes, all frames longer than the value given here are marked as long. This value does not affect the frame's reception. | 1518 |

5.2.1.5. [0x014] MAC_ADDR_0 and [0x018] MAC_ADDR_1 Register

The MAC address is stored in the registers in hexadecimal form. For example, to set the MAC address to: AC-DE-48-00-00-80 would require writing 0x48_00_00_80 to address 0x014 (MAC_ADDR_0). 0xAC_DE to address 0x018 (MAC ADDR 1).

Table 5.8. [0x014] MAC_ADDR_0 Register

| Bit | Name | Description | Default |
|--------|------------|---|---------|
| [31:0] | | First four bytes of the MAC address. | 0 |
| [51.0] | mac_addr_0 | Ethernet address assigned to the port that is supported by the MAC. | U |

Table 5.9. [0x018] MAC_ADDR_1 Register

| Bit | Name | Description | |
|---------|------------|---|---|
| [31:16] | Reserved | Reserved bits. | |
| [15:0] | mac_addr_1 | Last two bytes of the MAC address. | 0 |
| [15:0] | | Ethernet address assigned to the port that is supported by the MAC. | 0 |

5.2.1.6. [0x01C] TX_RX_STS Register

Table 5.10. [0x01C] TX_RX_STS Register

| Bit | Name | Description | Default |
|--------|--------------|---|---------|
| [31:5] | Reserved | Reserved bits. | 0 |
| [4] | link_ok | Link is OK. When set to 1, this indicates that no fault symbols were received on the link. | 0 |
| [3] | remote_fault | Remote fault. When set to 1, this indicates that remote fault symbols were received on the link. | 0 |

© 2025 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal. All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.



| Bit | Name | Description | Default |
|-----|-------------|---|---------|
| [2] | local_fault | Local fault. When set to 1, this indicates that local fault symbols were received on the link. | 0 |
| [1] | rx_idle | Receive MAC idle. When set to 1, this indicates that the RX MAC is inactive. | |
| [0] | tx_idle | Transmit MAC idle. When set to 1, this indicates that the TX MAC is inactive. | 0 |

5.2.1.7. [0x020] VLAN_TAG Register

This register has the VLAN tag field of the most recent tagged frame that was received.

Table 5.11. [0x020] VLAN_TAG Register

| Bit | Name | Description | Default |
|---------|----------|----------------|---------|
| [31:16] | Reserved | Reserved bits. | 0 |
| [15:0] | vlan_tag | VLAN tag ID. | 0 |

5.2.1.8. [0x024] MC_TABLE_0 and [0x028] MC_TABLE_1 Register

When the core is programmed to receive multicast frames, a filtering scheme is used to decide whether the frame should be received or not. This 64-bit matrix forms the hash table that is used to filter out the incoming multicast frames. For details, refer to the Receive MAC section.

Table 5.12. [0x024] MC_TABLE_0 Register

| Bit | Name | Description | Default |
|-------------------|------------|-----------------------------------|---------|
| [21:0] | me table 0 | Multicast Table Word 0. | 0 |
| [31:0] mc_table_0 | | First 4-bytes of the 64-bit hash. | U |

Table 5.13. [0x028] MC_TABLE_1 Register

| Bit | Name | Description | Default | |
|-------------------|------------|----------------------------------|---------|--|
| [31:0] | me table 1 | Multicast Table Word 1. | 0 | |
| [31:0] mc_table_1 | | Last 4-bytes of the 64-bit hash. | U | |

5.2.1.9. [0x02C] PAUSE_OPCODE Register

This register contains the pause opcode. This is compared against the opcode in the received pause frame. This value is also included in any pause frame transmitted by the MAC.

Table 5.14. [0x02C] PAUSE_OPCODE Register

| Bit | Name | Description | Default |
|---------|--------------|----------------|----------|
| [31:16] | Reserved | Reserved bits. | 0 |
| [15:0] | pause_opcode | Pause opcode. | 16'h0001 |

5.2.1.10. [0x030] MAC_CTL Register

Table 5.15. [0x030] MAC_CTL Register

| Table 5:15. [0x050] MAC_CTE Register | | | |
|--------------------------------------|--|----------------|---------|
| Bit | Name | Description | Default |
| [31:5] | Reserved | Reserved bits. | |
| [4] | ignore_pkt Ignore packet. When set to 1, the RX MAC ignores or drops incoming packets. | | 0 |
| [3:1] | Reserved | Reserved bits. | |



| Bit | Name | Description | Default |
|-----|------------|---|---------|
| | tx_pausreq | Transmit pause frame. | |
| | | 1 = send request, 0 = do not send request. This is a positive | |
| [0] | | edge-triggered bit. | 0 |
| | | The tx_fc_en bit of TX_CTL register should be set to 1 to enable this | |
| | | feature. | |

5.2.1.11. [0x034] PAUSE_TM Register

This register has the pause time for a flow control packet sourced by the 2.5G MAC transmitter.

Table 5.16. [0x034] PAUSE_TM Register

| Bit | Name | Description | Default |
|---------|------------|-----------------|---------|
| [31:16] | Reserved | Reserved bits. | _ |
| [15:0] | tx_paustim | Pause duration. | 0 |

5.2.2. Statistics Counters

These statistic counters are wraparound counters and can only be reset when the system reset is asserted. The default value of these counters is 0.

The register name with "_0" refers to the least significant word of the counter and "_1" refers to the most significant word.

Table 5.17. Summary of Statistics Counters

| Offset | Register Name | Access | Description |
|--------|---------------------|--------|--|
| 0x044 | TX_STAT_PKT_LNGTH_0 | RO | Transmit Packet Length Statistics Counter. Indicates the total number of octets transmitted in a |
| 0x048 | TX_STAT_PKT_LNGTH_1 | RO | particular frame. tx_statvec_o[13:0] is used to implement this counter. |
| 0x04C | TX_STAT_ERR_0 | RO | Transmit TX Error Statistics Counter. Counts the total number of PHY terminated packet. The |
| 0x050 | TX_STAT_ERR_1 | RO | tx_statvec_o[24] is used to implement this counter. |
| 0x054 | TX_STAT_UNDER_RUN_0 | RO | Transmit Underrun Error Statistics Counter. Counts the total number of underrun packets |
| 0x058 | TX_STAT_UNDER_RUN_1 | RO | transmitted. tx_statvec_o[21] is used to implement this counter. |
| 0x05C | TX_STAT_CRC_ERR_0 | RO | Transmit CRC Error Statistics Counter. Counts the total number of packets transmitted with |
| 0x060 | TX_STAT_CRC_ERR_1 | RO | CRC error. tx_statvec_o[22] is used to implement this counter. |
| 0x064 | TX_STAT_LNGTH_ERR_0 | RO | Transmit Length Error Statistics Counter. Counts the total number of packets transmitted with |
| 0x068 | TX_STAT_LNGTH_ERR_1 | RO | length of the packet and length in the Length/Type field mismatch. tx_statvec_o[23] is used to implement this counter. |
| 0x06C | TX_STAT_LNG_PKT_0 | RO | Transmit Long packet Statistics Counter. Counts the total number of packets transmitted with |
| 0x070 | TX_STAT_LNG_PKT_1 | RO | length of the packet longer than the max_frm_size. tx_statvec_o[25] is used to implement this counter. |
| 0x074 | TX_STAT_MULTCST_0 | RO | Transmit Multicast Packet Statistics Counter. Counts the total number of multicast packets transmitted. |
| 0x078 | TX_STAT_MULTCST_1 | RO | tx_statvec_o[20] is used to implement this counter. |

FPGA-IPUG-02293-1.2



| Offset | Register Name | Access | Description |
|--------|-------------------------|--------|---|
| 0x07C | TX_STAT_BRDCST_0 | RO | Transmit Broadcast Packet Statistics Counter. Counts the total number of broadcast packets transmitted. |
| 0x080 | TX_STAT_BRDCST_1 | RO | tx_statvec_o[19] is used to implement this counter. |
| 0x084 | TX_STAT_CNT_0 | RO | Transmit Control Packet Statistics Counter. Counts the total number of control packets (pause frame) |
| 0x088 | TX_STAT_CNT_1 | RO | transmitted by the AXI4-Stream interface. tx_statvec_o[16] is used to implement this counter. |
| 0x08C | TX_STAT_JMBO_0 | RO | Transmit Jumbo Packet Statistics Counter. Counts the total number of jumbo packets transmitted. |
| 0x090 | TX_STAT_JMBO_1 | RO | tx_statvec_o[17] is used to implement this counter. |
| 0x094 | TX_STAT_PAUSE_0 | RO | Transmit Pause Packet Statistics Counter. Counts the total number of pause packets inserted by |
| 0x098 | TX_STAT_PAUSE_1 | RO | the MAC core (enabled through MAC_CTL register). tx_statvec_o[15] is used to implement this counter. |
| 0x09C | TX_STAT_VLN_TG_0 | RO | Transmit VLAN Tag Statistics Counter. Counts the total number of tagged packets transmitted. |
| 0x0A0 | TX_STAT_VLN_TG_1 | RO | tx_statvec_o[18] is used to implement this counter. |
| 0x0A4 | TX_STAT_PKT_OK_0 | RO | Transmit Packet OK Statistics Counter. Counts the total number of packets transmitted without |
| 0x0A8 | TX_STAT_PKT_OK_1 | RO | any errors. tx_statvec_o[14] is used to implement this counter. |
| 0x0AC | TX_STAT_PKT_64_0 | RO | Transmit Packet 64 Statistics Counter. |
| 0x0B0 | TX_STAT_PKT_64_1 | RO | Counts the total number of packets transmitted with length equal to 64. |
| 0x0B4 | TX_STAT_PKT_65_127_0 | RO | Transmit Packet 65 - 127 Statistics Counter. Counts the |
| 0x0B8 | TX_STAT_PKT_65_127_1 | RO | total number of packets transmitted with lengths between 65 and 127. |
| 0x0BC | TX_STAT_PKT_128_255_0 | RO | Transmit Packet 128-255 Statistics Counter. Counts the |
| 0x0C0 | TX_STAT_PKT_128_255_1 | RO | total number of packets transmitted with lengths between 128 and 255. |
| 0x0C4 | TX_STAT_PKT_256_511_0 | RO | Transmit Packet 256-511 Statistics Counter. Counts the |
| 0x0C8 | TX_STAT_PKT_256_511_1 | RO | total number of packets transmitted with lengths between 256 and 511. |
| 0x0CC | TX_STAT_PKT_512_1023_0 | RO | Transmit Packet 512-1023 Statistics Counter. Counts the |
| 0x0D0 | TX_STAT_PKT_512_1023_1 | RO | total number of packets transmitted with lengths between 512 and 1,023. |
| 0x0D4 | TX_STAT_PKT_1024_1518_0 | RO | Transmit Packet 1024-1518 Statistics Counter. Counts |
| 0x0D8 | TX_STAT_PKT_1024_1518_1 | RO | the total number of packets transmitted with lengths between 1,024 and 1,518. |
| 0x0DC | TX_STAT_PKT_1518_0 | RO | Transmit Packet 1518 Statistics Counter. |
| 0x0E0 | TX_STAT_PKT_1518_1 | RO | Counts the total number of packets transmitted with length greater than 1,518. |
| | i . | | 1 |



| Offset | Register Name | Access | Description |
|--------|--------------------------|--------|---|
| 0x0E4 | TX_STAT_FRM_ERR_0 | RO | Transmit Frame Error Statistics Counter. Counts the total number of packets transmitted with |
| 0x0E8 | TX_STAT_FRM_ERR_1 | RO | error. tx_statvec_o[14] is used to implement this counter. |
| 0x0EC | TX_STAT_PKT_1519_2047_0 | RO | Transmit Packet 1519-2047 Statistics Counter. Counts |
| 0x0F0 | TX_STAT_PKT_1519_2047_1 | RO | the total number of packets transmitted with lengths between 1,024 and 2,047. |
| 0x0F4 | TX_STAT_PKT_2048_4095_0 | RO | Transmit Packet 2048-4095 Statistics Counter. Counts the total number of packets transmitted with lengths |
| 0x0F8 | TX_STAT_PKT_2048_4095_1 | RO | between 2,048 and 4,095. |
| 0x0FC | TX_STAT_PKT_4096_9216_0 | RO | Transmit Packet 4096-9216 Statistics Counter. Counts |
| 0x100 | TX_STAT_PKT_4096_9216_1 | RO | the total number of packets transmitted with lengths between 4,096 and 9,216. |
| 0x104 | TX_STAT_PKT_9217_16383_0 | RO | Transmit Packet 9217-16383 Statistics Counter. Counts |
| 0x108 | TX_STAT_PKT_9217_16383_1 | RO | the total number of packets transmitted with lengths between 9,217 and 16,383. |
| 0x10C | RX_STAT_PKT_LNGTH_0 | RO | Receive Packet Length Statistics Counter. Indicated the |
| 0x110 | RX_STAT_PKT_LNGTH_1 | RO | length of the packet received. rx_statvec_o[13:0] is used to implement this counter. |
| 0x114 | RX_STAT_VLN_TG_0 | RO | Receive VLAN Tag Statistics Counter. |
| 0x118 | RX_STAT_VLN_TG_1 | RO | Counts the total number of tagged packets received. rx_statvec_o[18] is used to implement this counter. |
| 0x11C | RX_STAT_PAUSE_0 | RO | Receive Pause Packet Statistics Counter. |
| 0x120 | RX_STAT_PAUSE_1 | RO | Counts the total number of pause packets received. rx_statvec_o[19] is used to implement this counter. |
| 0x124 | RX_STAT_FLT_0 | RO | Receive Filtered Packet Statistics Counter. |
| 0x128 | RX_STAT_FLT_1 | RO | rx_statvec_o[22] is used to implement this counter. |
| 0x12C | RX_STAT_UNSP_OPCODE_0 | RO | Receive Unsupported Opcode Statistics Counter. Counts the number of packets received with unsupported |
| 0x130 | RX_STAT_UNSP_OPCODE_1 | RO | Opcode. rx_statvec_o[23] is used to implement this counter. |
| 0x134 | RX_STAT_BRDCST_0 | RO | Receive Broadcast Packet Statistics Counter. Counts the number of packets received that were directed to the |
| 0x138 | RX_STAT_BRDCST_1 | RO | broadcast address. This does not include multicast packets. rx_statvec_o[15] is used to implement this counter. |
| 0x13C | RX_STAT_MULTCST_0 | RO | Receive Multicast Packet Statistics Counter. Counts the number of packets received that were directed to the |
| 0x140 | RX_STAT_MULTCST_1 | RO | multicast address. This does not include broadcast packets. rx_statvec_o[16] is used to implement this counter. |
| 0x144 | RX_STAT_LNGTH_ERR_0 | RO | Receive Length Error Statistics Counter. Counts the total number of packets received with |



| Offset | Register Name | Access | Description |
|--------|-------------------------|--------|--|
| 0x148 | RX_STAT_LNGTH_ERR_1 | RO | length of the packet and length in the Length/Type field mismatch. rx_statvec_o[20] is used to implement this counter. |
| 0x14C | RX_STAT_LNG_PKT_0 | RO | Receive Long Packet Statistics Counter. Counts the number of packets received longer than the |
| 0x150 | RX_STAT_LNG_PKT_1 | RO | max_pkt_len. rx_statvec_o[21] is used to implement this counter. |
| 0x154 | RX_STAT_CRC_ERR_0 | RO | Receive CRC Error Statistics Counter. |
| 0x158 | RX_STAT_CRC_ERR_1 | RO | Counts the number of packets received with CRC error. rx_statvec_o[17] is used to implement this counter. |
| 0x15C | RX_STAT_PKT_DISCARD_0 | RO | Receive Packet Discard Statistics Counter. Counts the number of packets discarded at the receive |
| 0x160 | RX_STAT_PKT_DISCARD_1 | RO | end. rx_statvec_o[14] is used to implement this counter. |
| 0x164 | RX_STAT_PKT_IGNORE_0 | RO | Receive Packet Ignored Statistics Counter. Counts the number of packets ignored when you request using the |
| 0x168 | RX_STAT_PKT_IGNORE_1 | RO | ignore_pkt. rx_statvec_o[25] is used to implement this counter. |
| 0x16C | RX_STAT_PKT_FRAGMENTS_0 | RO | Receive Packet Fragments Statistics Counter. Counts the number of packets received with less than 64 octets in |
| 0x170 | RX_STAT_PKT_FRAGMENTS_1 | RO | length and has either an FCS error or an alignment error. rx_statvec_o[13:6] along with rx_statvec[17] are used to implement this counter. |
| 0x174 | RX_STAT_PKT_JABBERS_0 | RO | Receive Packet Jabbers Statistics Counter. Counts the number of packets received with length longer than |
| 0x178 | RX_STAT_PKT_JABBERS_1 | RO | 1,518 octets and has either an FCS error or an alignment error. rx_statvec_o[13:0] along with rx_statvec_o[17] are used to implement this counter. |
| 0x17C | RX_STAT_PKT_64_0 | RO | Receive Packet 64 Statistics Counter. Counts the number of packets received that were 64 octets in |
| 0x180 | RX_STAT_PKT_64_1 | RO | length (including bad packets). rx_statvec_o[13:0] is used to implement this counter. |
| 0x184 | RX_STAT_PKT_65_127_0 | RO | Receive Packet 65-127 Statistics Counter. Counts the number of packets received that were between 65-127 |
| 0x188 | RX_STAT_PKT_65_127_1 | RO | octets in length (including bad packets). |
| 0x18C | RX_STAT_PKT_128_255_0 | RO | Receive Packet 128-255 Statistics Counter. Counts the number of packets received that were between 128- |
| 0x190 | RX_STAT_PKT_128_255_1 | RO | 255 octets in length (including bad packets). |
| 0x194 | RX_STAT_PKT_256_511_0 | RO | Receive Packet 256-511 Statistics Counter. Counts the number of packets received that were between 256- |
| 0x198 | RX_STAT_PKT_256_511_1 | RO | 511 octets in length (including bad packets). |
| 0x19C | RX_STAT_PKT_512_1023_0 | RO | Receive Packet 512-1023 Statistics Counter. Counts the number of packets received that were between 512- |
| 0x1A0 | RX_STAT_PKT_512_1023_1 | RO | 1,023 octets in length (including bad packets). |
| 0x1A4 | RX_STAT_PKT_1024_1518_0 | RO | Receive Packet 1024-1518 Statistics Counter. Counts |
| 0x1A8 | RX_STAT_PKT_1024_1518_1 | RO | the number of packets received that were between 1,024-1,518 octets in length (including bad packets). |



| Offset | Register Name | Access | Description |
|--------|-----------------------------|--------|--|
| 0x1AC | RX_STAT_PKT_UNDERSIZE_0 | RO | Receive Packet Undersize Statistics Counter. Counts the number of packets received that were less than 64 |
| 0x1B0 | RX_STAT_PKT_UNDERSIZE_1 | RO | octets long and were otherwise well formed. |
| 0x1B4 | RX_STAT_PKT_UNICAST_0 | RO | Receive Packet Unicast Statistics Counter. Counts the number of good packets received that were |
| 0x1B8 | RX_STAT_PKT_UNICAST_1 | RO | directed to a single address. |
| 0x1BC | RX_STAT_PKT_RCVD_0 | RO | Packets Received Statistics Counter. Counts the number of packets received (including bad packet, broadcast, and multicast packets). |
| 0x1C0 | RX_STAT_PKT_RCVD_1 | RO | rx_statvec[15] and rx_statvec_o[16] are used to implement this counter. |
| 0x1C4 | RX_STAT_PKT_64_GOOD_CRC_0 | RO | Receive Packet 64 with Good CRC Statistics Counter. Counts the number of packets received with length less |
| 0x1C8 | RX_STAT_PKT_64_GOOD_CRC_1 | RO | than 64 and with a good CRC. rx_statvec_o[13:6] and rx_statvec_o[17] are used to implement this counter. |
| 0x1CC | RX_STAT_PKT_1518_GOOD_CRC_0 | RO | Receive Packet 1518 with Good CRC Statistics Counter. Counts the number of packets received with length |
| 0x1D0 | RX_STAT_PKT_1518_GOOD_CRC_1 | RO | more than 1,518 and with a good CRC. rx_statvec_o[13:0] and rx_statvec_o[17] are used to implement this counter. |
| 0x1D4 | RX_STAT_PKT_1519_2047_0 | RO | Receive Packet 1519-2047 Statistics Counter. Counts the number of packets received that were between |
| 0x1D8 | RX_STAT_PKT_1519_2047_1 | RO | 1,519 - 2,047 octets in length (including bad packets). |
| 0x1DC | RX_STAT_PKT_2048_4095_0 | RO | Receive Packet 2048-4095 Statistics Counter. Counts the number of packets received that were between |
| 0x1E0 | RX_STAT_PKT_2048_4095_1 | RO | 2,048 – 4,095 octets in length (including bad packets). |
| 0x1E4 | RX_STAT_PKT_4096_9216_0 | RO | Receive Packet 4096-9216 Statistics Counter. Counts the number of packets received that were between |
| 0x1E8 | RX_STAT_PKT_4096_9216_1 | RO | 4,096 – 9,216 octets in length (including bad packets). |
| 0x1EC | RX_STAT_PKT_9217_16383_0 | RO | Receive Packet 9217-16383 Statistics Counter. Counts |
| 0x1F0 | RX_STAT_PKT_9217_16383_1 | RO | the number of packets received that were between 9,217 – 16,383 octets in length (including bad packets). |
| 0X204 | TX_STAT_PTP1588_PKT_0 | RO | Transmit PTP1588 Statistics Counter. Counts the |
| 0x208 | TX_STAT_PTP1588_PKT_1 | RO | number of PTP1588 packets transmitted. The tx_statvec_o[26] is used to implement this counter. |
| 0x20C | RX_STAT_PTP1588_PKT_0 | RO | Receive PTP1588 Statistics Counter. Counts the number |
| 0x210 | RX_STAT_PTP1588_PKT_1 | RO | of PTP1588 packets received. The rx_statvec_o[26] is used to implement this counter. |
| 0x21C | RX_STAT_LINK_OK | RO | Receive link_ok Statistics Counter. Counts the number of link_ok bit received. |
| | | | Note: If you wish to use the RX link_ok statistics counter, you cannot reset the MAC signals, such as xg_rxval_o and xg_rx_blk_lock_o signals when the RX PHY is down. |



| Offset | Register Name | Access | Description |
|--------|--------------------------|--------|---|
| 0x224 | TX_STAT_PKT_LNGTH_ACCU_0 | RO | Transmit Accumulation Byte Statistic Counter. Count and accumulate the total packet length that will receive |
| 0x228 | TX_STAT_PKT_LNGTH_ACCU_1 | RO | from AXIS interface and transmits to GMII interface. The counter will rollover to 0 if full. |
| 0x22C | RX_STAT_PKT_LNGTH_ACCU_0 | RO | Receive Accumulation Byte Statistic Counter. Count and accumulate the total packet length that is received from |
| 0x230 | RX_STAT_PKT_LNGTH_ACCU_1 | RO | GMII interface and transmitted to AXIS interface. The counter will not increase if the packet received is discarded by MAC. The counter will rollover to 0 if full. |

5.3. **PHY Registers (CertusPro-NX Devices)**

This section provides detailed descriptions of 2.5G Ethernet PCS data registers.

The following register address map specifies the available IP core registers.

For register information, refer to the CertusPro-NX SerDes/PCS User Guide (FPGA-TN-02245).

Table 5.18. Register Address Map for PHY (CertusPro-NX Devices)

| APB Offset | MDIO Register Address | Register Name | Bit Width | Description | |
|---------------------------------|--------------------------|---------------------------|--------------|--|--|
| Vendor Specific MMD 1 Registers | | | | | |
| 0x7804 | 0x0002 | MMD1 Device Identifier 0 | 16 | Refer to IEEE 802.3 Clause 45.2. | |
| 0x7805 | 0x0003 | MMD1 Device Identifier 1 | 16 | Refer to IEEE 802.3 Clause 45.2. | |
| 0x780A | 0x0005 | MMD1 Devices in Package 0 | 16 | Refer to IEEE 802.3 Clause 45.2. | |
| 0x780B | 0x0006 | MMD1 Devices in Package 1 | 16 | Refer to IEEE 802.3 Clause 45.2. | |
| 0x7810 | 0x0008 | MMD1 Status Register | 16 | Refer to IEEE 802.3 Clause 45.2. | |
| 0x781C | 0x000E | MMD1 Package Identifier 0 | 16 | Refer to IEEE 802.3 Clause 45.2. | |
| 0x781D | 0x000F | MMD1 Package Identifier 1 | 16 | Refer to IEEE 802.3 Clause 45.2. | |
| 0x7A00 – 0x7AFF | 0x0000 – 0x00FF | PMA Registers | 16 | Refer to the Appendix A section of the CertusPro-NX SerDes/PCS User Guide (FPGA-TN-02245). | |
| 0x7B00 – 0x7BEA | 0x0100 - 0x01EA | MPCS Module Registers | 16 | Refer to the Multi-Protocol PCS Module User Guide (FPGA-IPUG-02118). | |

The behavior of registers to write and read access is defined by its access type, which is defined in the following table.

Table 5.19. Access Type Definition

| able 51257 (toests 1 ype 2 climator) | | | | |
|---|-------------------------|--------------------------|--|--|
| Access Type | Behavior on Read Access | Behavior on Write Access | | |
| RO | Returns register value | Ignores write access | | |
| RW | Returns register value | Updates register value | | |
| RSVD (Applicable to MMD registers only) | Ignore when read | Ignores write access | | |

5.3.1. MMD1 Device Identifier 0

mmd1_dv_id_0—Provides the upper 16-bit value, which constitutes a unique identifier for a particular type of vendorspecific device. For more information regarding this register, see IEEE 802.3 Clause 45.2 Table 45-1.

Table 5.20. MMD1 Device Identifier 0

| Field | Name | Access | Width | Reset | |
|--------|--------------|--------|-------|-------|--|
| [15:0] | mmd1_dv_id_0 | RO | 16 | 16'h0 | |



5.3.2. MMD1 Device Identifier 1

mmd1_dv_id_1—Provides the lower 16-bit value, which constitutes a unique identifier for a particular type of vendor-specific device. For more information regarding this register, see IEEE 802.3 Clause 45.2 Table 45-1.

Table 5.21. MMD1 Device Identifier 1

| Field | Name | Access | Width | Reset |
|--------|--------------|--------|-------|-------|
| [15:0] | mmd1_dv_id_1 | RO | 16 | 16'h3 |

5.3.3. MMD1 Devices in Package 0

mmd1_dv_pkg_0—Provides the lower 16-bit value, each bit read as one indicates which MMDs are instantiated within the same package as the MMD being accessed. PMA and PCS are bit 1 and bit 3 of this register, respectively; hence, the reset value is 16'h000A. For more details regarding this register, see IEEE 802.3 Clause 45.2 Table 45-2.

Table 5.22. MMD1 Devices in Package 0

| Field | Name | Access | Width | Reset |
|--------|---------------|--------|-------|-------|
| [15:0] | mmd1_dv_pkg_0 | RO | 16 | 16'hA |

5.3.4. MMD1 Devices in Package 1

mmd1_dv_pkg_1—Provides the upper 16-bit value, each bit read as one indicates which MMDs are instantiated within the same package as the MMD being accessed. For more details regarding this register, see IEEE 802.3 Clause 45.2 Table 45-2.

Table 5.23. MMD1 Devices in Package 1

| Field | Name | Access | Width | Reset |
|--------|---------------|--------|-------|-------|
| [15:0] | mmd1_dv_pkg_1 | RO | 16 | 16'h0 |

5.3.5. MMD1 Status Register

mmd1 stat reg—Specifies if the device is present and responding at this register address or not.

- 2'b10 Device responding at this address.
- 2'b11 No device responding at this address.
- 2'b01 No device responding at this address.
- 2'b00 No device responding at this address.

Table 5.24. MMD1 Status Register

| Field | Name | Access | Width | Reset |
|---------|---------------|--------|-------|-------|
| [15:14] | mmd1_stat_reg | RO | 2 | 2'h2 |
| [13:0] | reserved | RSVD | 14 | 14'h0 |

5.3.6. MMD1 Package Identifier 0

mmd1_pkg_id_0—Provides the upper 16-bit value, which constitutes a unique identifier for a particular type of package instantiated within. The package identifier may be the same as the device identifier. For more information regarding this register, see IEEE 802.3 Clause 45.2 Table 45-1.

Table 5.25. MMD1 Package Identifier 0

| Field | Name | Access | Width | Reset |
|--------|---------------|--------|-------|-------|
| [15:0] | mmd1_pkg_id_0 | RO | 16 | 16'h0 |

© 2025 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.

All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.



5.3.7. MMD1 Package Identifier 1

mmd1_pkg_id_1— Provides the lower 16-bit value, which constitutes a unique identifier for a particular type of package instantiated within. The package identifier may be the same as the device identifier. For more information regarding this register, see IEEE 802.3 Clause 45.2 Table 45-1.

Table 5.26. MMD1 Package Identifier 1

| Field | Name | Access | Width | Reset |
|--------|---------------|--------|-------|-------|
| [15:0] | mmd1_pkg_id_1 | RO | 16 | 16'h3 |



Example Design

The 2.5G Ethernet IP example design allows you to compile, simulate, and test the IP on the following Lattice evaluation board:

CertusPro-NX Evaluation Board

For more information, refer to the CertusPro-NX Evaluation Board User Guide (FPGA-EB-02046).

6.1. **Example Design Supported Configuration**

This section describes how to use the 2.5G Ethernet MAC + PHY IP with the following example design: CertusPro-NX Evaluation Board with SMA cable serial loopback setup.

Note: In the table below, ✓ refers to a checked option in the 2.5G Ethernet MAC + PHY IP example design.

Table 6.1. 2.5G Ethernet MAC + PHY IP Configuration Supported by the Example Design

| IP GUI Parameter | IP Configuration |
|--|------------------|
| Select IP Option | MAC + PHY |
| Loopback Mode | Disabled |
| MAC Configuration: Multicast Address Filtering TX Pause Frame Generation via Ports | |
| Statistics Counter Configuration: • Statistics Counter Registers | |
| PMA Setup-Receiver Subgroup | |
| RX Adaptive Equalization: Gen1 Enable | ✓ |
| Gen1: Adaptive AlgorithmGen2: Enable | RL2plus — |
| Gen3: Enable | |
| Gen1/2: Preliminary Adaptive EQGen1/2: Training phase Adaptive EQ | Enabled Enabled |
| Gen1/2: Post-phase Adaptive EQ | Enabled |
| PHY Configuration: PCS Lane ID | Auto |

Overview of the Example Design and Features

This section describes how to use the 2.5G Ethernet MAC + PHY IP example design on CertusPro-NX Evaluation Board.

The key features of the example design include:

- 1. APB module that configure MAC to enable transmit and receive datapath.
- 2. Packet generator on transmit datapath.
- Packet checker on receive datapath to check loopback packets or traffic.



6.3. Example Design Components

The following figure shows the block diagram of the example design. The example design features the 2.5G Ethernet IP along with all the components. FMC daughtercard is connected to the CertusPro-NX Evaluation Board for external reference clock and serial data loopback. The external reference clock used in this example design is sourced from a clock generator (for example, CG635 clock generator) that supplies 156.25 MHz LVDS clock.

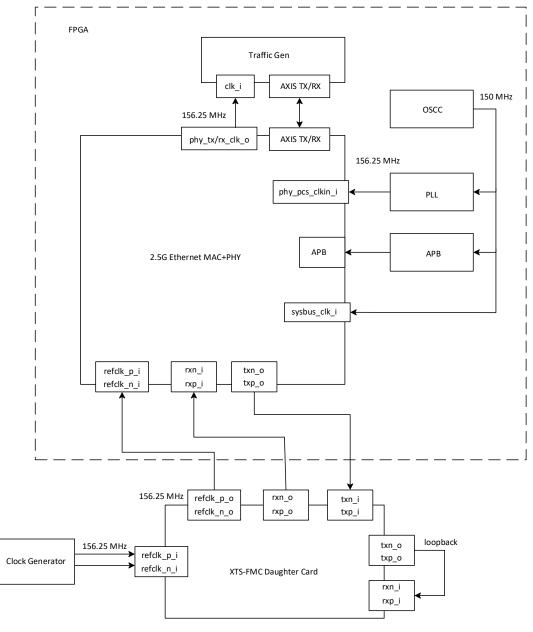


Figure 6.1. 2.5G MAC + PHY Hardware Example Design Block Diagram

The 2.5G Ethernet example design includes the following blocks:

- 2.5G Ethernet MAC+PHY IP
- PLL
- OSCC
- APB
- Traffic_gen
- XTS-FMC daughter card



6.3.1. 2.5G Ethernet MAC+PHY IP

The 2.5G Ethernet IP core contains all the necessary logic, interfacing, and clocking infrastructure ready to integrate with any system efficiently. It supports the ability to transmit and receive data between standard interfaces, such as APB and AXI4-Stream.

6.3.2. PLL

The phase-locked loop (PLL) module is capable of frequency synthesis and clock phase management including clock injection delay cancellation. It has flexibility of input and feedback source selections, multiple output selections, and independent phase-shifting features. The PLL is used to supply 156.25 MHz clock to phy_psc_clkin_i from 2.5G Ethernet IP, by sourcing the clock from OSC.

6.3.3. OSC

The oscillator (OSC) module is designed to produce two clock signals that drive the FPGA clock tree for user-specific applications. The trimmed low-frequency oscillator and trimmed high-frequency oscillator are also used by the IP sub-system. The OSC is used to supply clock to PLL, APB, and sysbus clk i from 2.5G Ethernet IP.

6.3.4. APB

The Advanced Peripheral Bus is the standard interface that the 2.5G Ethernet IP core uses to interface with the host. The APB is used to configure 2.5G Ethernet IP registers, TX enable and RX enable.

6.3.5. Traffic Gen

The Traffic Generator block module is an optional block that is used to generate continuous fixed length frame of size 1500, with pseudo random data for transmission using AXI4-Stream Transmit and Receive interface. This module has data checker included, which performs checking on the data received against the data transmitted.

6.3.6. XTS-FMC Daughtercard

The XTS-FMC daughtercard is designed to convert FPGA transceiver channels to SMA connectors through an FPGA Mezzanine Card interface. For more information about the XTS-FMC Board, refer to the XTS-FMC Board web page.

6.4. Simulating the Example Design

6.4.1. Example Design Testbench

The 2.5G Ethernet MAC + PHY example design comes with a testbench for simulating the versa top file, eval/versa top/lfcpnx 2p5g macphy/tb top.v.

6.4.2. 2.5G Ethernet CertusPro-NX Hardware Example Design Simulation Flow

The following flow chart shows the simulation process from initializing 2.5G Ethernet IP and starting the traffic generators to showcasing the IP transmission and receiver performance.



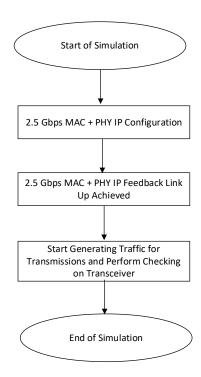


Figure 6.2. 2.5G Ethernet Example Design Flowchart

To simulate the testbench, follow these steps:

1. Launch the Simulation Wizard from the Radiant software and follow the onscreen instructions. Select *tb_top* as the simulation top module.

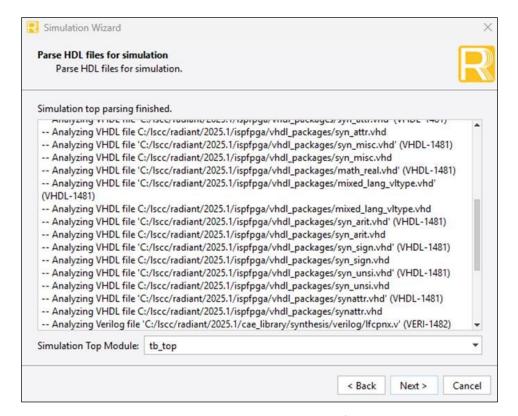


Figure 6.3. Simulation Wizard Interface



2. After you have launched the QuestaSim software, load the macro file, *tb_top.do* to load the preset signal for the simulation. You may need to restart the simulation after loading the preset signal.

The following figure shows the simulation results of the 2.5G Ethernet MAC + PHY IP. The *compareFail* signal from the checker remains low throughout the test, showing all the data transmitted and received have passed the data checker test.

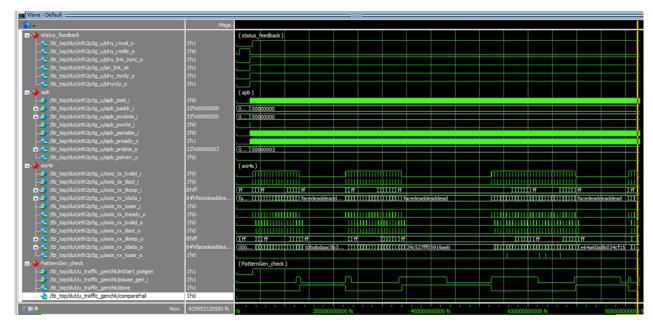


Figure 6.4. Overall Simulation Results for 2.5G Ethernet MAC+PHY

A successful test run displays the following behavior:

phy_link_sync_o and phy_rxval_o signals are asserted from 2.5G MAC+PHY IP, indicating that the IP is ready for transmission.

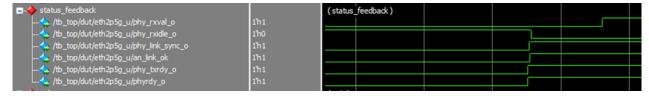


Figure 6.5. Simulation Results for 2.5G Link Status

3. Start pattern generator by asserting *inStart_patgen*, and AXI-S TX and RX are populated by data. If there is any data that fails the checker test, *compareFail* is asserted. Assertion of *pause_gen_i* is used to demonstrate pause and resume of data transmission.

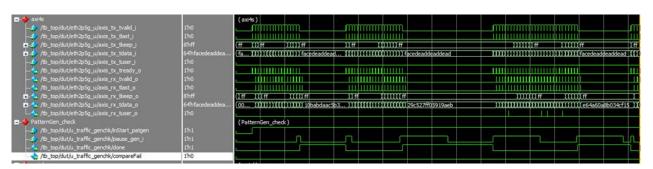


Figure 6.6. Simulation Results for Data Transmission



4. At the end of the transmission, the transcript displays *End of Simulation* along with the result of the checker test, indicating either PASS or FAIL.

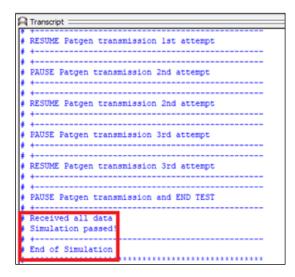


Figure 6.7. Simulation Transcript

6.5. Hardware Testing

The 2.5G Ethernet MAC + PHY IP example design demonstrates serial loopback functionality using the CertusPro-NX Evaluation Board connected with an FMC daughtercard of the XTS-FMC Board by Terasic. In this example design, a 156.25 MHz LVDS clock rate is supplied from an external clock generator to the CertusPro-NX Evaluation Board through the XTS-FMC Board.

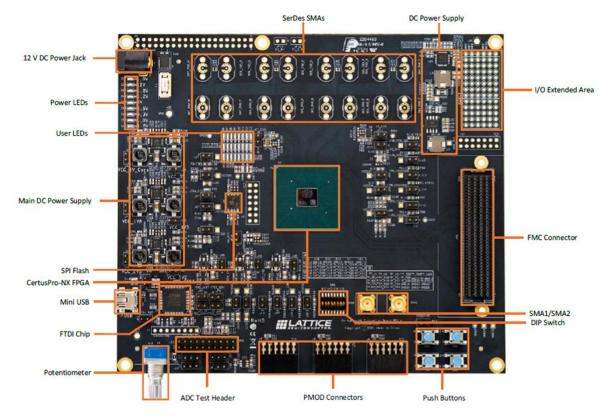


Figure 6.8. CertusPro-NX Evaluation Board



For more information on the CertusPro-NX Evaluation Board, refer to the CertusPro-NX Evaluation Board User Guide (FPGA-EB-02046).

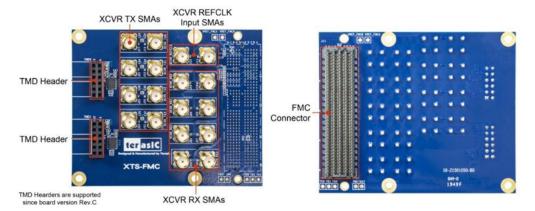


Figure 6.9. XTS-FMC Board

For more information about the XTS-FMC Board, refer to the XTS-FMC Board web page.

6.5.1. Using 2.5G Ethernet Example Design

6.5.1.1. Creating New Radiant Project

To create a new project, follow these steps:

1. In the Lattice Radiant software, go to File -> New -> Project... or click on the New Project icon.

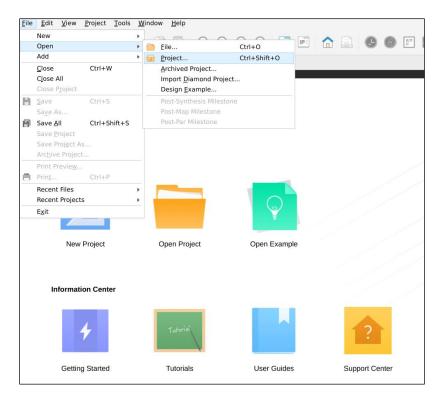


Figure 6.10. Project Creation



2. Specify Project Name, Location, Implementation Name, and click Next.

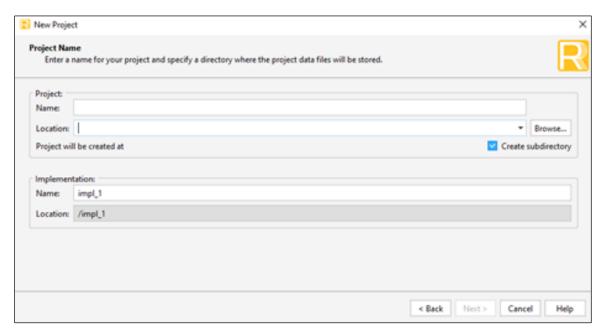


Figure 6.11. Project Name and Location

3. Specify a device for the project by selecting a device family, an operating condition, the package, performance grade, and part number.

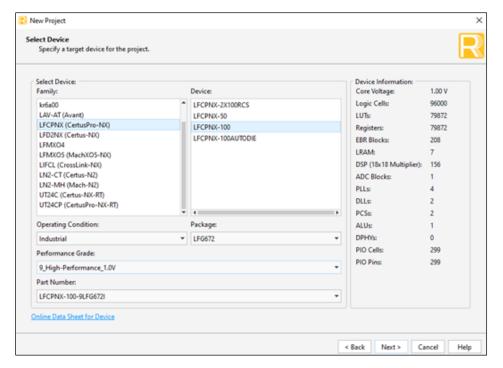


Figure 6.12. Select a Device



4. Select a synthesis tool for the project.

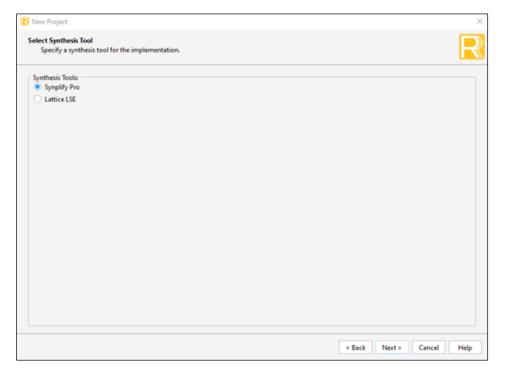


Figure 6.13. Project Synthesis Tool

5. View and verify the project information for the project creation.

6.5.1.2. IP Installation and Generation

To install and generate the IP, follow these steps:

- 1. Go to IP Catalog -> IP on Server to download the IP.
- 2. After you download and install the IP from server, go to IP on Local -> Generate... to launch the IP generation wizard.

FPGA-IPUG-02293-1.2



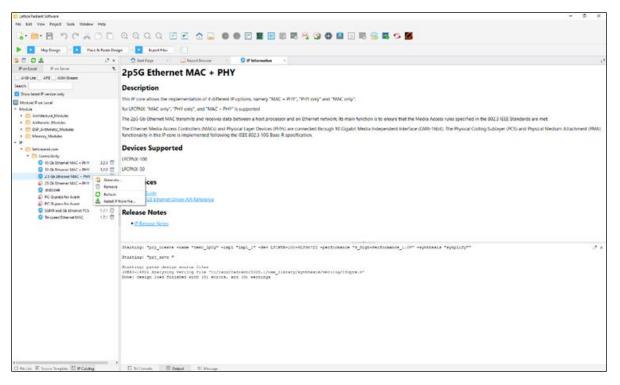


Figure 6.14. IP Catalog

3. Enter the component name and project directory for IP generation, and click Next.

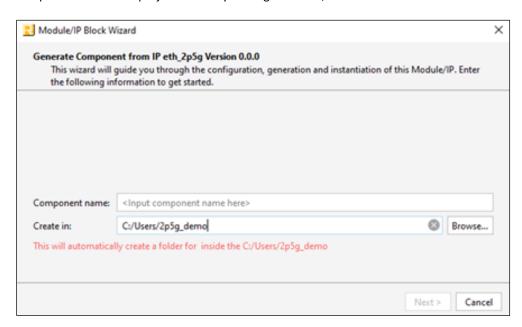


Figure 6.15. IP Component



4. Configure the IP interface, and click **Generate**.

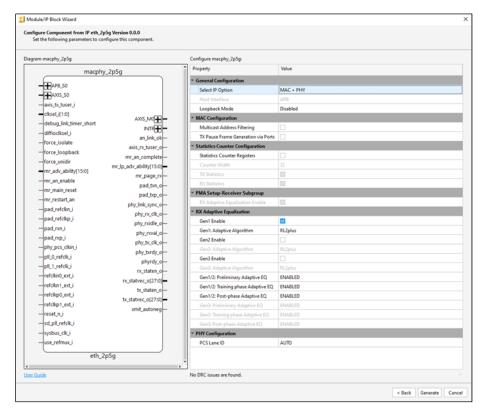


Figure 6.16. IP Component Configuration - MAC+PHY

5. Verify the generated IP, and click **Finish**.



Figure 6.17. Importing Versa Files into a Project



The following table lists the files that are generated.

Table 6.2. Generated File List

| Files | Description |
|----------------------------|---|
| top.v | Hardware example design top file. |
| tb_top.v | Simulation top file for hardware example design. |
| pll_1.v | PLL module from foundation IP. |
| OSCC.v | OSC module from foundation IP. |
| pcs_apb_w.v; pcs_apb_w_r.v | APB module for register configuration. |
| traffic_genchk.v | Traffic generator module for generator random traffic for transmission, and perform checking in loopback. |
| debounce.v | Debounce module for hardware mechanical input. |
| cpnx_eval_2p5g.pdc | Post synthesize constraint file for CertusPro-NX evaluation board. |
| tb_top.do | Script to group simulation signals. |

6.5.1.3. Add Example Design Files into Project

To add an existing example design into a project, follow these steps:

1. Go to **Add-> Existing File**...and select existing example design from *IP_NAME/eval/versa_top/lfcpnx_2p5g_macphy*.

For CertusPro-NX 2.5G

Top-level file to include in project: top.v

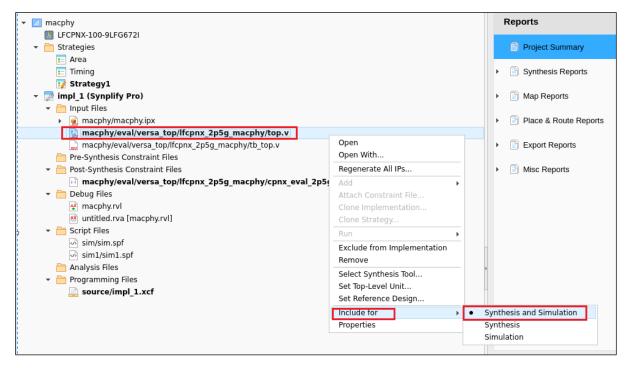


Figure 6.18. Steps to Set the top.sv File



Simulation for top-level file to include: tb_top.v

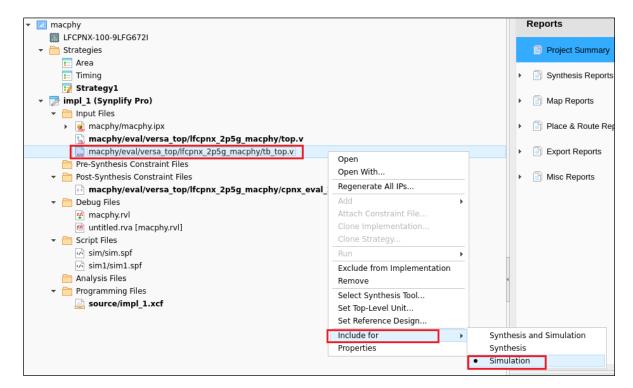


Figure 6.19. Steps to Set the tb_top.v File

Post-synthesis constraint file: cpnx_eval_2p5g.pdc

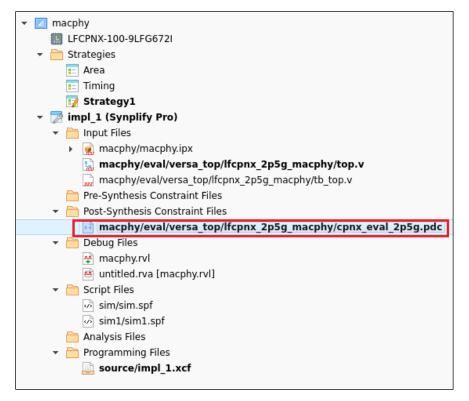


Figure 6.20. Steps to Add the cpnx_eval_2p5g.pdc File



2. Click **Run All** to compile the bitstream file from the design.



Figure 6.21. Generate Bitstream

3. After the Bitstream is compiled successfully, the run button turns green with a checked sign.



Figure 6.22. Bitstream Completion

6.5.1.4. Program Bitstream

To program the bitstream, follow these steps:

1. Go to **Tools -> Programmer** to launch the Radiant Programmer.

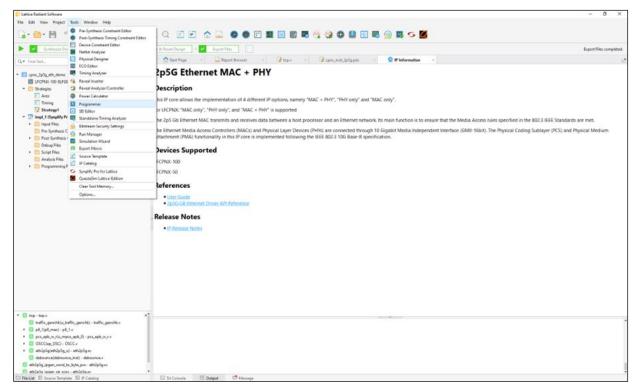


Figure 6.23. Programmer

2. Verify the setting under **Cable Setup**, the device family detected from the programmer, and the correct bitstream is selected in the **File Name** column.

FPGA-IPUG-02293-1.2



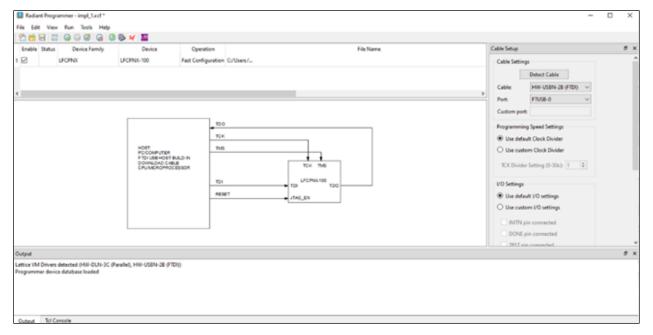


Figure 6.24. Select Bitstream

3. Select **Run -> Program Device** to program bitstream into the target device.

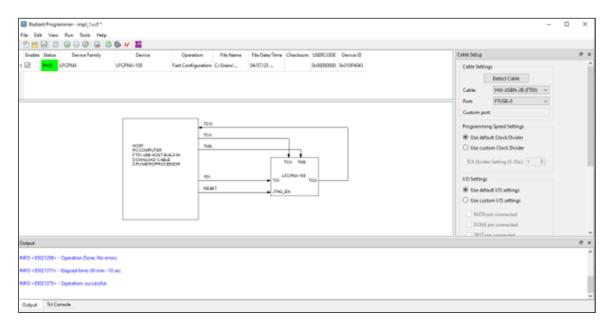


Figure 6.25. Programming Bitstream



6.5.2. Hardware Setup for CertusPro-NX Evaluation Board with 2.5G Ethernet Example Design

Figure 6.26 shows the complete hardware setup for the example design. The CertusPro-NX Evaluation Board is powered up with 12 V power supply. The FMC daughtercard is connected to the CertusPro-NX Evaluation Board via SMA cables with TX and RX loopback setup. A 156.25 MHz LVDS reference clock is supplied from an external clock generator.

6.5.2.1. DIP Switch Definition

The following table lists the programmed DIP switch for the CertusPro-NX Evaluation Board. After programming the bitstream into the device, make sure the DIP switch is configured to the right position before triggering RESET via a pushbutton.

Table 6.3. DIP Switch for CertusPro-NX Evaluation Board

| DIP Switch Signal Name | DIP_SW1. |
|------------------------|---|
| Default State | LOW. |
| Description | Traffic generator will be started when state transition from LOW to HIGH. |

For more information on DIP_SW1 (LOW) Traffic generator trigger, refer to signal naming in the CertusPro-NX Evaluation Board User Guide (FPGA-EB-02046).

6.5.2.2. Pushbutton

The following lists the programmed pushbutton for the CertusPro-NX Evaluation Board.

Table 6.4. Pushbuttons for CertusPro-NX Evaluation Board

| DIP Switch Signal Name | SW1 | SW5 |
|------------------------|--------------|--|
| Default State | HIGH | HIGH |
| Description | Reset button | Pause or resume the traffic transmission |

To perform RESET after programming the bitstream, follow these steps:

- 1. Confirm that the traffic generator (DIP_SW1) is LOW before triggering RESET.
- 2. After link-up is completed successfully, toggle DIP_SW1 from LOW to HIGH to start the traffic generator.
- 3. Check onboard LED for device status.
- 4. You may pause the traffic transmission using SW5 if CONTINUOUS_TRAFFIC is enabled (configuration is in *top.v* file).

6.5.2.3. LED Definition

LED_1 to LED_23 are programmed to light OFF in default state. Each LED will light up corresponding to the status of the example design. Refer to the following table for the description of each segment.

| Signal Name | Description |
|-----------------------|------------------------------------|
| LED_8 (Yellow color) | Reset asserted. |
| LED_9 (Yellow color) | PLL lock achieved. |
| LED_10 (Yellow color) | PHY link synchronization achieved. |
| LED_11 (Yellow color) | TX channel starts transmitting. |
| LED_12 (Yellow color) | RX channel starts receiving. |
| LED_13 (Yellow color) | Transmission completed or paused. |
| LED_14 (Yellow color) | FAIL data checker test. |

For more information, refer to signal naming in the CertusPro-NX Evaluation Board User Guide (FPGA-EB-02046).

FPGA-IPUG-02293-1.2



6.5.2.4. XTS-FMC Daughtercard Connection

Connect the XTS-FMC daughtercard to the CertusPro-NX Evaluation Board via the FMC connector. The XTS-FMC daughtercard connects J17 and J19 of TX SMA to J16 and J18 of RX SMA in loopback manner. The clock generator supplies XCVR REFCLK to J12 and J13 of input SMA.

To configure the *refclk* to source from external using DIFFCLKIO_CORE and PCSREFMUX, refer to the Reference Clocks section.

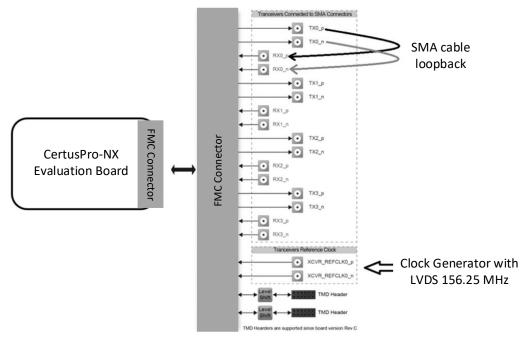


Figure 6.26. Connectivity Block for CertusPro-NX and XTS-FMC Loopback

For the naming of the SMA ports on the XTS-FMC Board, refer to the XTS-FMC Board web page.

6.5.2.5. Hardware Validation Result

The USB-Mini B connects to the host, enabling the Reveal tool to monitor hardware signals, such as TX, RX, and status indicators.



Figure 6.27. Actual Board Setup



The following figure shows continuous transmission from the pattern generator to populate the TX and RX in a loopback setup of the CertusPro-NX Evaluation Board. Each frame is validated in the checker. If there is any mismatch, the *compfail_o* signal is asserted. These signals must be added to the Reveal tool manually. For more information, refer to the Reveal User Guide for Radiant Software.

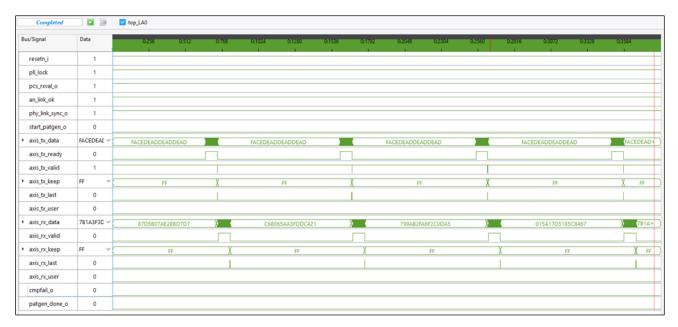


Figure 6.28. Reveal Signal During Transmission



7. Designing with the IP

This section provides information on how to generate the IP core using the Lattice Radiant software and how to run simulation and synthesis. For more details on the Lattice Radiant software, refer to the Lattice Radiant Software User Guide on the Lattice Radiant Software web page.

7.1. Generating and Instantiating the IP

You can use the Radiant software to generate IP modules and integrate them into the device architecture. The steps below describe how to generate the 2.5G Ethernet MAC + PHY IP in the Radiant software.

To generate the 2.5G Ethernet MAC + PHY IP, follow these steps:

- 1. Create a new Lattice Radiant software project or open an existing project.
- 2. In the **IP Catalog** tab, double-click 2.5G Ethernet MAC + PHY under **IP, Connectivity** category. The **Module/IP Block Wizard** opens as shown in the following figure. Enter values in the **Component name** and the **Create in** fields and click **Next**.

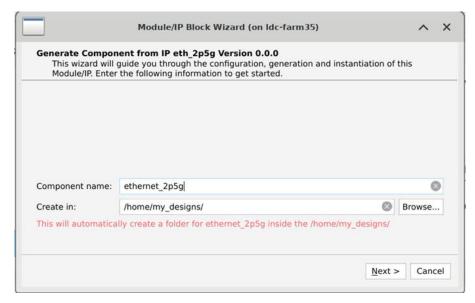


Figure 7.1. Module/IP Block Wizard

3. In the next **Module/IP Block Wizard** window, customize the selected 2.5G Ethernet MAC + PHY IP using drop-down lists and check boxes. The following figure shows an example configuration of the 2.5G Ethernet MAC + PHY IP. For details on the configuration options, refer to the IP Parameter Description section.



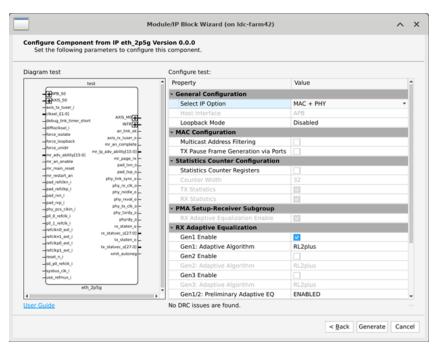


Figure 7.2. IP Configuration

4. Click **Generate**. The **Check Generating Result** dialog box opens, showing design block messages and results as shown in Figure 7.3.

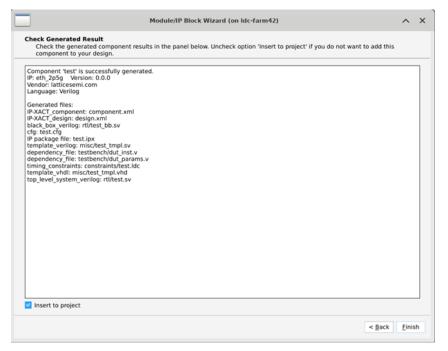


Figure 7.3. Check Generated Result

5. Click **Finish**. All the generated files are placed under the directory paths in the **Create in** and the **Component name** fields shown in Figure 7.1.



7.1.1. Generated Files and File Structure

The generated 2.5G Ethernet MAC + PHY module package includes the closed box (<Component name>_bb.v) and instance templates (<Component name>_tmpl.v/vhd) that can be used to instantiate the core in a top-level design. An example RTL top-level reference source file (<Component name>.v) that can be used as an instantiation template for the module is also provided. You may also use this top-level reference as the starting template for the top-level for their complete design. The generated files are listed in the table below.

Table 7.1. Generated File List

| Attribute | Description |
|---|---|
| <component name="">.ipx</component> | This file contains the information on the files associated to the generated IP. |
| <component name="">.cfg</component> | This file contains the parameter values used in IP configuration. |
| component.xml | Contains the ipxact: component information of the IP. |
| design.xml | Documents the configuration parameters of the IP in IP-XACT 2014 format. |
| rtl/ <component name="">.v</component> | This file provides an example RTL top file that instantiates the module. |
| rtl/ <component name="">_bb.v</component> | This file provides the synthesis closed box. |
| misc/ <component name="">_tmpl.v misc /<component name="">_tmpl.vhd</component></component> | These files provide instance templates for the module. |

7.2. Design Implementation

Completing your design includes additional steps to specify analog properties, pin assignments, and timing and physical constraints. You can add and edit the constraints using the Device Constraint Editor or by manually creating a PDC file. Post-Synthesis constraint files (.pdc) contain both timing and non-timing constraint .pdc source files for storing logical timing or physical constraints. Constraints that are added using the Device Constraint Editor are saved to the active .pdc file. The active post-synthesis design constraint file is then used as input for post-synthesis processes.

Refer to the relevant sections in the Lattice Radiant Software User Guide for more information on how to create or edit constraints and how to use the Device Constraint Editor.

7.3. Timing Constraints

The timing constraints are based on the clock frequency used. The timing constraints for the IP are defined in relevant constraint files. The example below shows the IP timing constraints generated for the two IP options in 2.5G Ethernet MAC + PHY IP. To ensure successful timing closure, you must include the recommended constraints in the .pdc file. You may refer to the constraint.ldc file for the contents in the figures below and Appendix B. For more information on timing constraints, refer to the Lattice Radiant Timing Constraints Methodology Application Note (FPGA-AN-02059).

FPGA-IPUG-02293-1.2



```
create_clock -name {pad_refclkn_i} -period 6.4 [get_ports pad_refclkn_i]
create_clock -name {pad_refclkp_i} -period 6.4 [get_ports pad_refclkp_i]
#create clock -name {refclkn0 ext i} -period 6.4 [get ports refclkn0 ext i]
#create_clock -name {refclkn1_ext_i} -period 6.4 [get_ports refclkn1_ext_i]
#create_clock -name {sd_pll_refclk_i} -period 6.4 [get_ports sd_pll_refclk_i]
create_clock -name {phy_tx_clk_o} -period 6.4 -waveform {0.000 3.200}
[get_pins lscc_2p5_gbe_top_inst.genblkl.lscc_2p5_gbephy_inst.u_mpcs_ten_gbe_wrap.
u_mpcs_twopointfive_gbe.lscc_mpcs_top_inst.PCSX1_inst.
u PCSX1 pcs merge pcs merge pcs merge pcs merge pcs merge pcsx4 merge.PCSX4 inst/CHO PIPE PCS TXCLKOUT]
create_clock -name {phy_rx_clk_o} -period 6.4 -waveform {0.000 3.200}
[get_pins lscc_2p5_gbe_top_inst.genblk1.lscc_2p5_gbephy_inst.u_mpcs_ten_gbe_wrap.
u_mpcs_twopointfive_gbe.lscc_mpcs_top_inst.PCSX1_inst.
u_PCSXI_pcs_merge_pcs_merge_pcs_merge_pcs_merge_pcs_merge_pcsx4_merge.PCSX4_inst/CHO_RXOUTCLK]
create_clock -name {sysbus_clk_i} -period 6.4 [get_ports sysbus_clk_i]
create_clock -name {phy_pcs_clkin_i} -period 6.4 -waveform {0.000 3.200} [get_ports phy_pcs_clkin_i]
create_clock -name {sysbus_clk} -period 50 [get_ports sysbus_clk_i]
set_max_delay -from [get_pins {lscc_2p5_gbe_top_inst/genblkl.lscc_ten_gbe_mac_inst/
u_ten_gbemac_core/rx_ram/u_mem0/mem_main/NON_MIX.ADDR_ROUTE[0].DATA_ROUTE[1].no_init.
u_mem0/LIFCL_MEM.pdp16k.PDP16K_MODE_inst/DO35}] -to [get_pins {lscc_2p5_gbe_top_inst/
genblkl.lscc_ten_gbe_mac_inst/u_ten_gbemac_core/rmac_buf/frames_present_reg[*].ff_inst/DF}]
-datapath only 5.76
set_clock_groups -group [get_clocks pad_refclkn_i] -group [get_clocks pad_refclkp_i]
-group [get_clocks sysbus_clk] -group [get_clocks phy_tx_clk_o] -group [get_clocks phy_rx_clk_o]
-group [get_clocks phy_pcs_clkin_i] -asynchronous
#set_clock_groups -group [get_clocks pad_refclkn_i] -group [get_clocks pad_refclkp_i] -group [get_clocks
set_false_path -from [get_ports reset_n_i]
set false path -from [get ports phy tx rst n i]
set_false_path -from [get_ports phy_rx_rst_n_i]
```

Figure 7.4. Timing Constraint File (.pdc) for the MAC+PHY Option

```
create_clock -name {pad_refclkn_i} -period 6.4 [get_ports pad_refclkn_i]
create_clock -name {pad_refclkp_i} -period 6.4 [get_ports pad_refclkp_i]
#create_clock -name {refclkn0_ext_i} -period 6.4 [get_ports refclkn0_ext_i]
#create_clock -name {refclkn1_ext_i} -period 6.4 [get_ports refclkn1_ext_i]
#create_clock -name {refclkn1_ext_i} -period 6.4 [get_ports refclkn1_ext_i]
#create_clock -name {sd_pll_refclk_i} -period 6.4 [get_ports sd_pll_refclk_i]

create_clock -name {phy_rx_clk_i} -period 6.4 -waveform {0.000 3.200} [get_ports phy_rx_clk_i]
create_clock -name {phy_tx_clk_i} -period 6.4 -waveform {0.000 3.200} [get_ports phy_rx_clk_i]
create_clock -name {phy_pcs_clkin_i} -period 6.4 -waveform {0.000 3.200} [get_ports phy_pcs_clkin_i]
create_clock -name {sysbus_clk} -period 50 [get_ports sysbus_clk_i]

set_clock_groups -group [get_clocks {pad_refclkn_i pad_refclkp_i}] -group [get_clocks sysbus_clk]
-group [get_clocks phy_tx_clk_i] -group [get_clocks phy_rx_clk_i] -group [get_clocks phy_pcs_clkin_i]
-asynchronous

set_false_path -from [get_ports reset_n_i]
set_false_path -from [get_ports phy_tx_rst_n_i]
set_false_path -from [get_ports phy_rx_rst_n_i]
```

Figure 7.5. Timing Constraint File (.pdc) for the PHY Only Option

7.4. Specifying the Strategy

The Radiant software provides two predefined strategies: Area and Timing. It also enables you to create customized strategies. For details on how to create a new strategy, refer to the *Strategies* section of the *Lattice Radiant Software User Guide* on the *Lattice Radiant Software* web page.



7.5. Running Functional Simulation

You can run functional simulation after the IP is generated.

To run functional simulation, follow these steps:

1. Click the button located on the **Toolbar** to initiate the **Simulation Wizard** shown in the figure below.

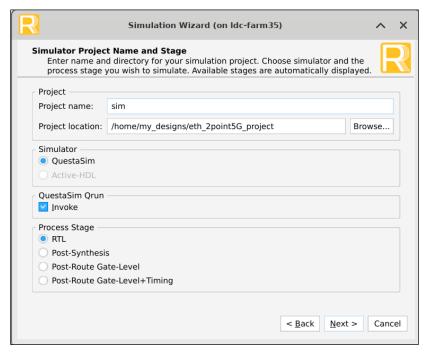


Figure 7.6. Simulation Wizard

2. Click Next to open the Add and Reorder Source window as shown in the figure below.

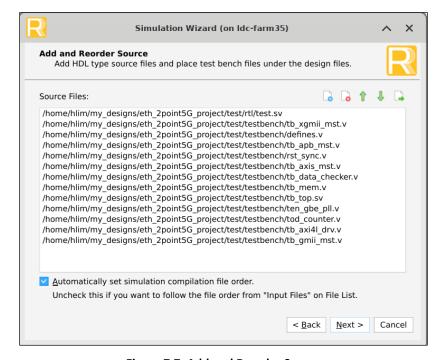


Figure 7.7. Add and Reorder Source



3. Click **Next**. The **Summary** window is shown.

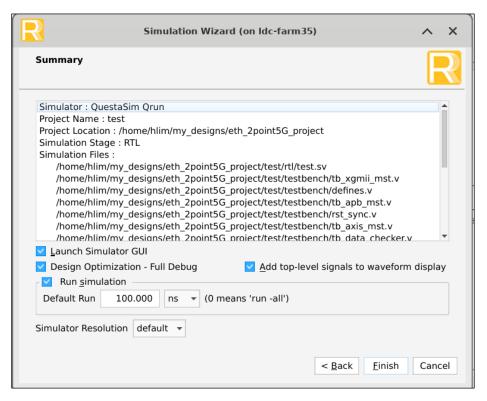


Figure 7.8. Summary Window

4. Click **Finish** to run the simulation.

Note: It is necessary to follow the procedure above until it is fully automated in the Lattice Radiant Software Suite. The waveform below shows an example of the simulation result.

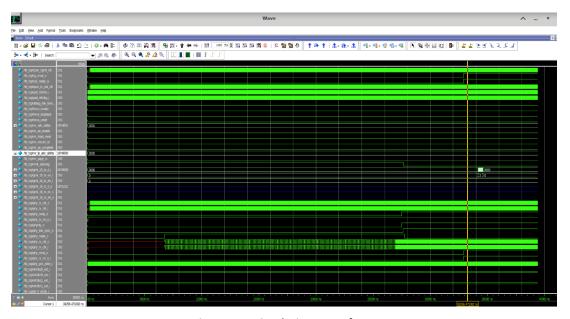


Figure 7.9. Simulation Waveform



7.5.1. Simulation Results

The following figure shows the details of the simulations results.

```
Transcript
         View Bookmarks Window Help
 ■ - 🚅 🗑 🐃 ቆ | ¾ 🖦 🖺 👲 🗅 | ⊘ - 🚜 🖺
 Starting APB Register Access
# [33978000000] [TEST]: SIMULATION PASSED
              Data matched: exp = 5b172db6, obs = 5b172db6

Data matched: exp = a3071a46, obs = a3071a46
 Data matched: exp = 7bd261f7, obs = 7bd261f7
Data matched: exp = da6ebab4, obs = da6ebab4
 Data matched: exp = 147cd928, obs = 147cd928
Data matched: exp = e3c530c7, obs = e3c530c7
Data matched: exp = 8477e408, obs = 8477e408
 Data matched: exp = 847/e408, obs = 847/e408
Data matched: exp = fea7a6fd, obs = fea7a6fd
Data matched: exp = 8837901c, obs = 8837901c
Data matched: exp = ed3408da, obs = ed3408da
Data matched: exp = 334ea766, obs = 334ea766
Data matched: exp = b9f50473, obs = b9f50473
 Data matched: exp = 2f3ab35e, obs = 2f3ab35e
Data matched: exp = 6a8e05d5, obs = 6a8e05d5
 Data matched: exp = dcf000b9, obs = dcf000b9
Data matched: exp = 4b273796, obs = 4b273796
# Break in Module tb top at /home/hlim/my designs/eth 2point5G project/test/testbench/tb top.sv line 592
```

Figure 7.10. Simulation Result



8. Known Issues

8.1. Minimum IPG Limitation

8.1.1. Devices Affected

CertusPro-NX devices (LFCPNX-50/100).

8.1.2. Designs Affected

This issue affects only designs using the *MAC only* and *MAC + PHY* IP option with 16-bit GMII interface on CertusPro-NX devices (LFCPNX-50/100). The *PHY* IP option is not affected.

8.1.3. Minimum IPG Limitation

The 2.5G TX MAC is designed to generate an IDLE packet when no data is being transmitted. Ideally, the minimum Inter-Packet Gap (IPG) must be 12 bytes on average. However, the minimum achievable IPG is approximately 26 bytes

8.1.4. Planned Fix

Lattice plans to resolve this issue in a future release.



Appendix A. Resource Utilization

The following tables show the resource utilization of the 2.5G Ethernet MAC + PHY IP core.

Table A.1. Resource Utilization for LFCPNX-100

| Configuration | Registers | LUTs | EBRs | Target Device | Synthesis Tools |
|-----------------|-------------|--------------|----------|---------------|-----------------|
| PHY Only | 946 (1%) | 1,309 (2%) | 0 | LFCPNX-100 | Synplify Pro |
| Default | | | | | |
| MAC + PHY | 5,561 (7%) | 6,788 (8%) | 82 (39%) | LFCPNX-100 | Synplify Pro |
| default | | | | | |
| MAC + PHY | 9,439 (12%) | 10,583 (13%) | 82 (39%) | LFCPNX-100 | Synplify Pro |
| Full featured | | | | | |
| Stat counter 32 | | | | | |
| MAC Only | 4,603 (6%) | 5,622 (7%) | 82 (39%) | LFCPNX-100 | Synplify Pro |
| default | | | | | |
| MAC Only | 8,496 (11%) | 9,473 (12%) | 82 (39%) | LFCPNX-100 | Synplify Pro |
| Full featured | | | | | |
| Stat counter 32 | | | | | |

^{*}Note: The utilization of MAC Only and MAC+PHY with full featured stat counter 64 will be provided for the next IP user guide release due to software issue.



Appendix B. Guide to Close Timing

For further improvement in timing closure, you can apply the set_max_delay constraint to prioritize specific timing paths. For example, the following constraint is used:

```
set_max_delay -from [get_pins {lscc_2p5_gbe_top_inst/genblk1.lscc_ten_gbe_mac_inst/
u_ten_gbemac_core/rx_ram/u_mem0/mem_main/NON_MIX.ADDR_ROUTE[0].DATA_ROUTE[1].no_init.
u_mem0/LIFCL_MEM.pdp16k.PDP16K_MODE_inst/DO35}] -to [get_pins {lscc_2p5_gbe_top_inst/
genblk1.lscc_ten_gbe_mac_inst/u_ten_gbemac_core/rmac_buf/frames_present_reg[*].ff_inst/DF}]
-datapath_only 5.76
```

This command sets a maximum data path delay of 5.76 ns between the specified source and destination pins. The -datapath_only option ensures that only the datapath delay is considered, excluding clock skew and other clock path delays. By tightening the timing constraint from the original 6.4 ns to 5.76 ns, the tool is instructed to give higher priority to this path during place and route process. This can lead to improved timing closure and better overall performance, especially for critical paths.

To further increase the likelihood of meeting timing constraints, you are encouraged to perform multiple iterations of synthesis and place-and-route. Each iteration allows the tool to explore different optimization strategies and placement solutions, which can improve the timing results incrementally. This iterative approach is especially effective when combined with targeted constraints like set_max_delay, as it helps the tool converge on an optimal solution over time. You can set the number of iterations in the place and route session under Strategy1 as shown in the figure below.

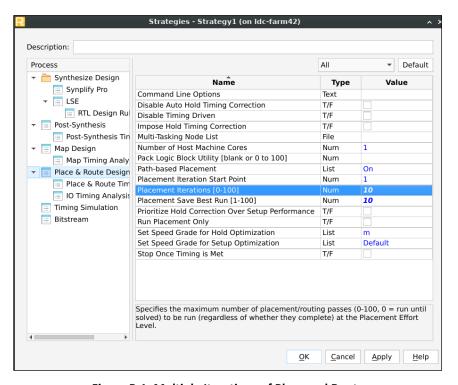


Figure B.1. Multiple Iterations of Place and Route



References

- 2.5G Ethernet MAC + PHY IP Release Notes (FPGA-RN-02083)
- 2.5G, 10G, and 25G Ethernet Driver API Reference (FPGA-TN-02375)
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- 2.5G Ethernet MAC + PHY IP web page
- CertusPro-NX web page
- Lattice Radiant Software web page
- Lattice Solutions IP Cores web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans



Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport. For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.



Revision History

Revision 1.2, IP v1.0.0, October 2025

| Section | Change Summary |
|---------|---|
| All | Removed all content and references related to MachXO5™-NX device support. |

Revision 1.1, IP v1.0.0, July 2025

| Section | Change Summary |
|--------------|-------------------------------|
| Introduction | Updated the Features section. |

Revision 1.0, IP v1.0.0, June 2025

| Section | Change Summary |
|---------|------------------|
| All | Initial release. |



www.latticesemi.com