



10 Gb Ethernet MAC and PCS

Reference Design

FPGA-RD-02248-1.2

March 2024

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

Acronyms in This Document	5
1. Introduction	6
2. Features	6
3. Functional Description	6
4. Waveforms	8
5. Implementation	9
6. Reference Design Simulation	10
6.1. Customization for Reference Design	13
6.1.1. Number of Packets and Packet Length	13
6.1.2. Loopback Mode Configuration	14
7. Reference Design Implementation On Board	17
7.1. Requirements	17
7.2. CertusPro-NX Versa Board	17
7.3. Board-to-Board Testing	18
7.3.1. Design Setup	18
7.3.2. Bitstream Programming	21
7.4. Waveforms Captured Using the Reveal Tool	23
7.4.1. Reveal Inserter	23
7.4.2. Reveal Analyzer	25
Technical Support Assistance	28
Revision History	29

Figures

Figure 3.1. Reference Design Basic Diagram	7
Figure 4.1. Timing Waveforms for MAC TX and RX	8
Figure 6.1. Simulation Wizard	10
Figure 6.2. Adding and Reordering Source	10
Figure 6.3. Summary	11
Figure 6.4. Execute Macro in ModelSim Lattice FPGA Edition	11
Figure 6.5. eth_sim.do in reference design directory	12
Figure 6.6. Simulation Waveform	12
Figure 6.7. Unload the reference design and re-invoke eth_sim.do to re-run the simulation	13
Figure 6.8. Modify Packet Contents in packet_gen_check.v	13
Figure 6.9. NUM_PKT Parameter in Testbench file tb_top.v	13
Figure 6.10. 10 Gb Ethernet PCS IP with Loopback Mode = Near PMA Loopback	14
Figure 6.11. EXT_CONNECTION_EN Parameter for Internal Loopback in Testbench File tb_top.v	15
Figure 6.12. 10 Gb Ethernet PCS IP with Loopback Mode = Disabled	15
Figure 6.12. EXT_CONNECTION_EN Parameter for external interface connection in Testbench File tb_top.v	16
Figure 7.1. LEDs on the CertusPro-NX Versa Board	17
Figure 7.2. PCS IP Loopback Mode	19
Figure 7.3. Change the EXT_CONNECTION_EN Parameter to 1 in Top File top.v	19
Figure 7.4. CertusPro-NX Versa Board-to-Board Connection	20
Figure 7.5. Optic Fiber Cables Connection to Boards	20
Figure 7.6. Standalone Radiant Programmer	21
Figure 7.7. Radiant Programmer Window	21
Figure 7.8. Radiant Programmer: Configure the Settings	22
Figure 7.9. Program Device Toolbar Icon	22
Figure 7.10. Message on Successful Programming	22
Figure 7.11. Toggle Switch 5 in SW1 to Initiate Pattern Generation for CertusPro-NX Versa Board	22

Figure 7.12. Reveal Inserter File in the File List	23
Figure 7.13. Add Logic Analyzer into the Debug File	23
Figure 7.14. Trace Signal Setup	24
Figure 7.15. Trigger Signal Setup	24
Figure 7.16. Create a New Reveal Analyzer File.....	25
Figure 7.17. LA Trigger Tab	26
Figure 7.18. CertusPro-NX Versa Board Disabled Loopback Mode Overview	26
Figure 7.19. MAC axis_rx_tdata_o packets.....	27
Figure 7.20. MAC axis_tx_data_i packets	27

Tables

Table 3.1. Signal Descriptions	7
Table 5.1. Performance and Resource Utilization	9
Table 7.1. Board LED Descriptions	18

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
GPLL	General Purpose PLL
I/O	Input/Output
LED	Light Emitting Diode
LSE	Lattice Synthesis Engine
MAC	Media Access Controller
PCS	Physical Coding Layer
PLL	Phase-Locked Loop
RTL	Register Transfer Language
RX	Receiver
SFP	Small Form-Factor Pluggable
TX	Transmitter

1. Introduction

The Lattice Semiconductor 10 Gb Ethernet MAC IP core supports the ability to transmit and receive data between a host processor and an Ethernet network. The main function of the 10 Gb Ethernet MAC is to ensure that the media access rules specified in the IEEE 802.3 standards are met while transmitting a frame of data over the Ethernet. On the receive side, the Ethernet MAC extracts the different components of a frame and transfers them to higher applications through an AXI4-Stream interface.

The 10 Gb Ethernet PCS IP core provides XGMII interface to MAC and follows IEEE 802.3 10GBASE-R standard. It supports 64-bit data and 8-bit control signals for both transmit and receive paths.

This reference design demonstrates an example of a 10GBASE-R application using a 10 Gb Ethernet MAC IP core with a 10 Gb Ethernet PCS IP core in loopback mode. A simple Ethernet packet generator is included to generate Ethernet packets to be transmitted on the transmitter (TX) and compare them with the received packets from the receiver (RX).

2. Features

The following lists the features for the 10 Gb Ethernet MAC IP core and the 10 Gb Ethernet PCS IP core:

- Ethernet packet generator to generate and compare packets
- Supports internal loopback (Near PMA Loopback) and external loopback mode

For more information on the features of the 10 Gb Ethernet MAC IP core, refer to the [10 Gb Ethernet MAC IP Core User Guide \(FPGA-IPUG-02162\)](#).

For more information on the features of the 10 Gb Ethernet PCS IP core, refer to the [10 Gb Ethernet PCS IP Core User Guide \(FPGA-IPUG-02163\)](#).

3. Functional Description

This reference design instantiates and connects a 10 Gb Ethernet MAC IP core to a 10 Gb Ethernet PCS IP core through an XGMII interface. The Ethernet PCS IP is configured to take in a clock source of 161.1328 MHz from dedicated reference clocks (pad_refclkp/n_i). The TX clock output from the Ethernet PCS is fed to a GPLL to generate 156.25 MHz clock for the data path. Note that the Fractional-N Divider option of the GPLL must be disabled. The serial interface is loop backed externally.

A packet generator is included in the design to generate Ethernet packets to be sent to the Ethernet MAC IP core through the AXI4-Stream interface on the TX path. Another packet generator will generate the same packets to compare with the Ethernet packets received from the Ethernet MAC IP core RX path. [Figure 3.1](#) shows the basic block diagram and connections of the reference design.

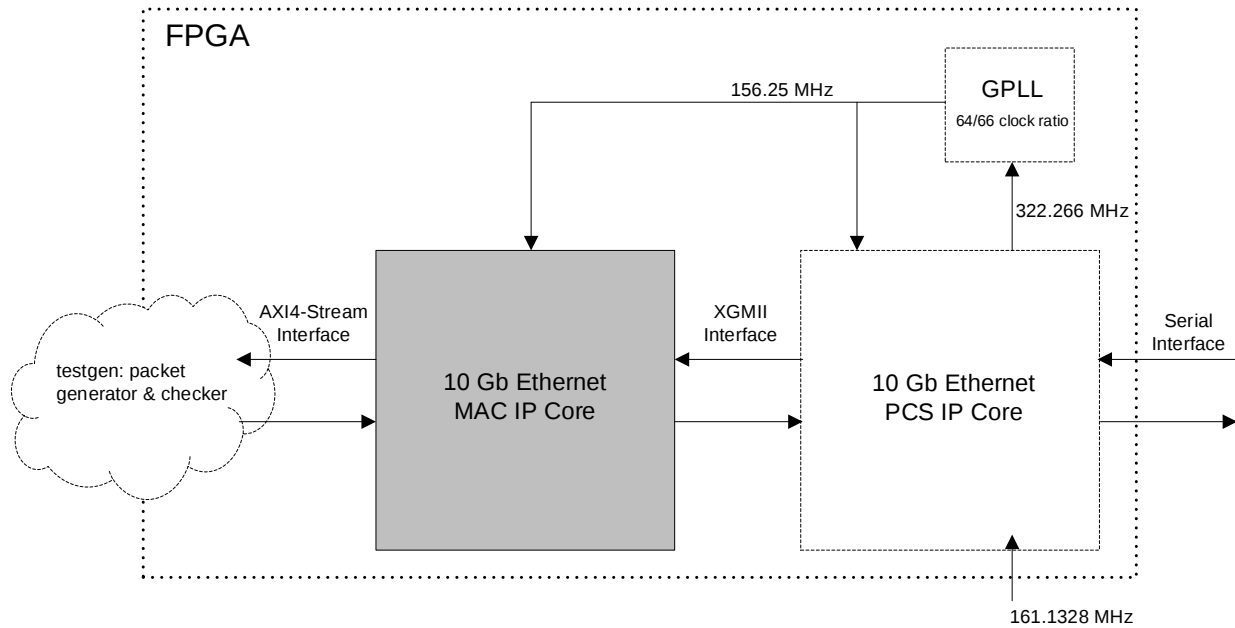


Figure 3.1. Reference Design Basic Diagram

Table 3.1 lists the top-level signal descriptions.

Table 3.1. Signal Descriptions

Signal Name	I/O Type	I/O Width	Function/Connection Description
pad_refclk_i	I	1	Dedicated 161.1328 MHz refclk input for PCS.
pad_refclkp_i	I	1	Dedicated 161.1328 MHz refclk input for PCS.
pad_rxn_i	I	1	RX- differential signal.
pad_rxp_i	I	1	RX- differential signal.
pad_txn_o	O	1	TX- differential signal .
pad_txp_o	O	1	TX- differential signal .
led[7:0]	O	8	LED outputs: Refer to Table 7.1.
rstn	I	1	Active low reset for the reference design.
pll_lock	O	1	GPLL locked signal.
pcs_rxval	O	1	PCS RX valid – To indicate that the PCS is sending out a valid XGMII data.
patgen_done_o	O	1	To indicate that NUM_PKT of packets have been generated and transmitted. Applicable when parameter CONTINUOUS_TRAFFIC=0.
cmpFail_o	O	1	To indicate a data mismatch in the received packet(s).
sfp_tx_disable_o	O	1	To drive the TX Disable (pin 3) in the SFP+ module.
start_patgen_i	I	1	To start generating packets.

4. Waveforms

The design has been verified through RTL function simulation. [Figure 4.1](#) shows the timing waveform of the MAC TX and RX. For detailed timing waveforms, download the design and run the RTL function simulation.

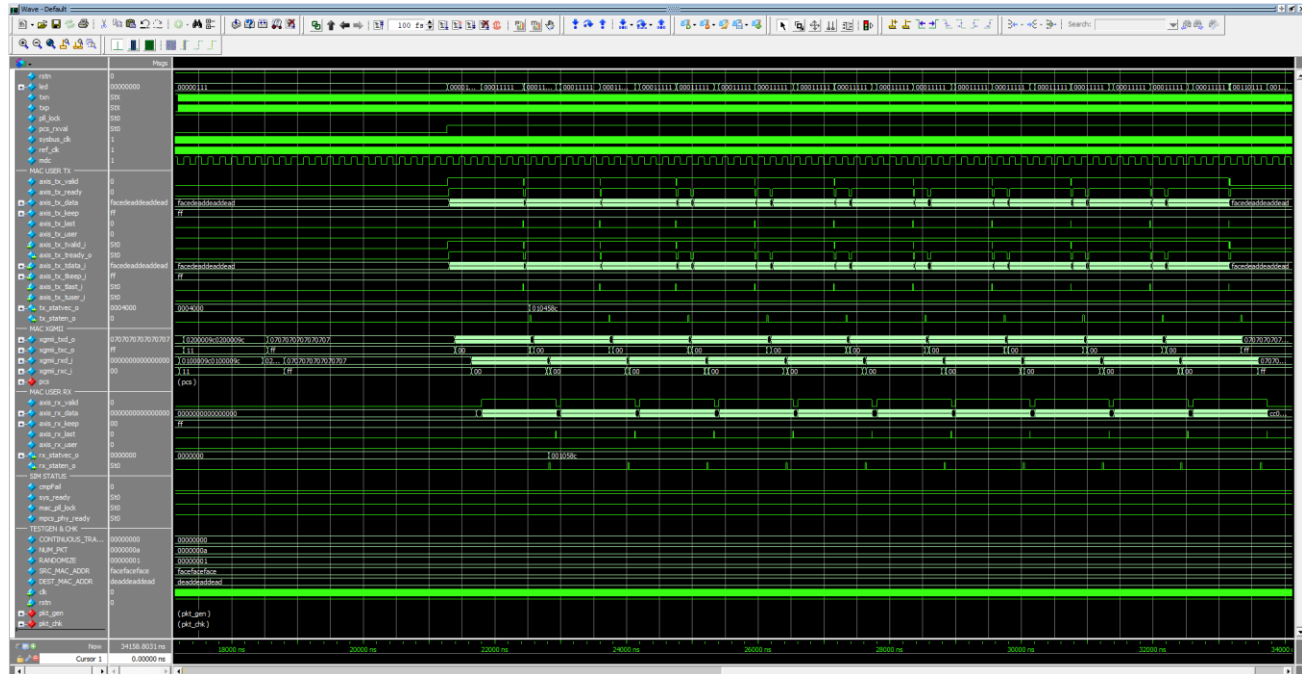


Figure 4.1. Timing Waveforms for MAC TX and RX

5. Implementation

This design is implemented in Verilog. When using this design in a different device, density, speed or grade, performance and utilization may vary. Default settings are used during the fitting of the design.

Table 5.1. Performance and Resource Utilization

Device Family	Language	Speed Grade	Utilization (LUTs)	f _{MAX} (MHz)	I/O	Architecture Resources
CertusPro™-NX ¹	Verilog	-8	4690	>156.25	22	1 PLL, 1 PCS

Notes:

1. Performance and utilization characteristics are generated using LFCPNX-100-8LFG672C, with the Lattice Radiant™ 3.1 design software with Lattice Synthesis Engine (LSE).
2. The Maximum Clock Frequency is obtained by running the timing analysis with the Lattice design software. Timing simulation should be run after any changes are made and the reference design is merged with the overall design.

6. Reference Design Simulation

To run the reference design simulation, follow these steps:

1. Unzip the reference design zip file into a directory.
2. Open reference design project file (eth.rdf) from the Lattice Radiant software.
3. Click the  button located on the **Toolbar** to initiate the **Simulation Wizard**.

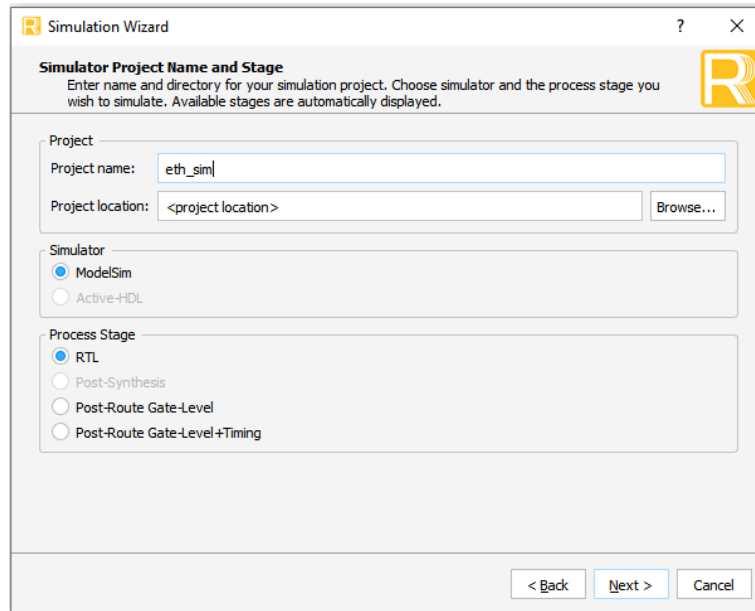


Figure 6.1. Simulation Wizard

4. Click **Next** to open the **Add and Reorder Source** window.

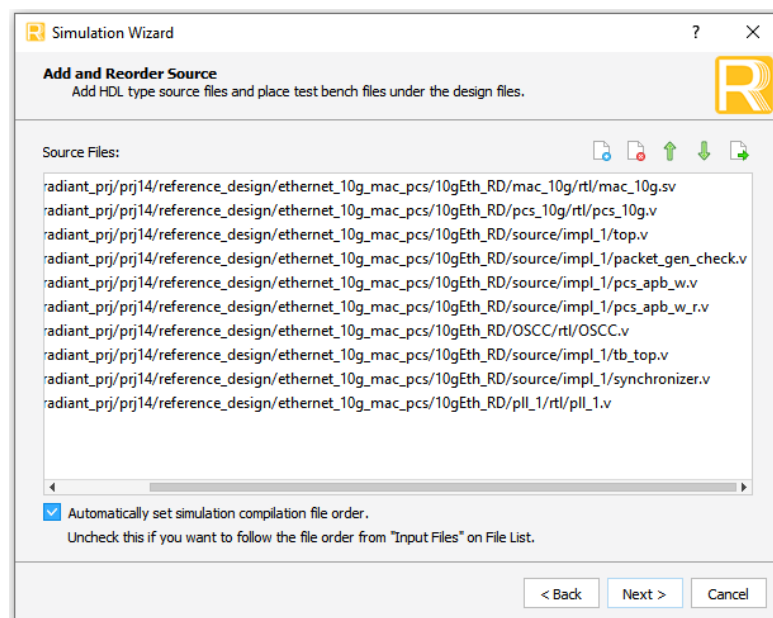


Figure 6.2. Adding and Reordering Source

5. Click **Next**. The **Summary** window is shown. Uncheck **Add top-level signals to waveform display** check box. The **Run simulation** check box will be greyed out when you do that. Click **Finish**.

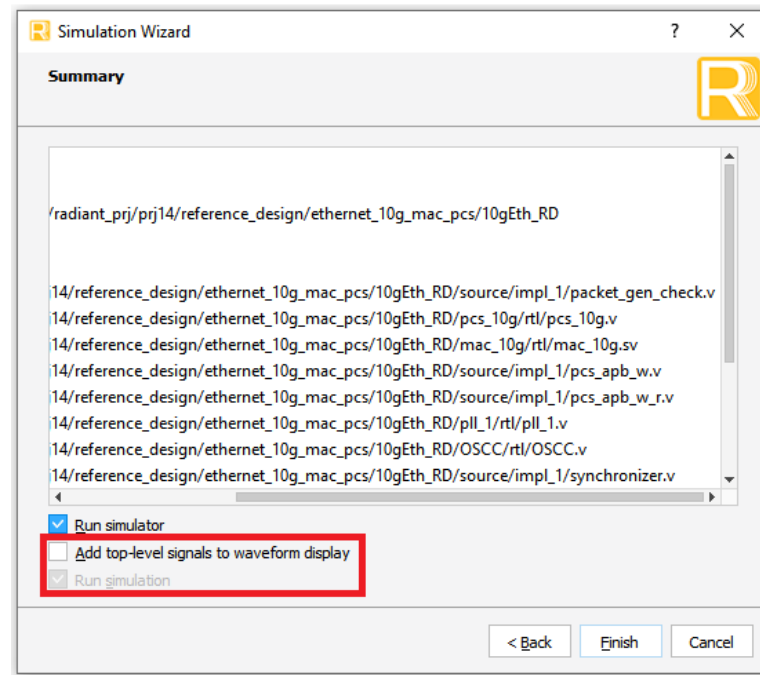


Figure 6.3. Summary

6. Wait until the ModelSim Lattice FPGA Edition has finished loading.
7. In the ModelSim Lattice FPGA Edition, go to **Tools -> Tcl -> Execute Macro...**

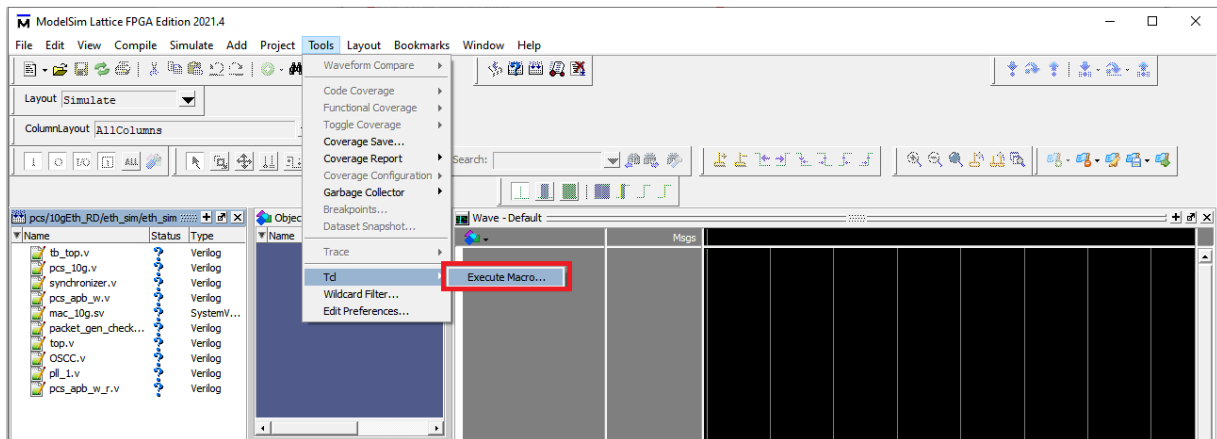


Figure 6.4. Execute Macro in ModelSim Lattice FPGA Edition

8. Go to the reference design directory and select the **eth_sim.do** file. Click **Open** to run the .do file.

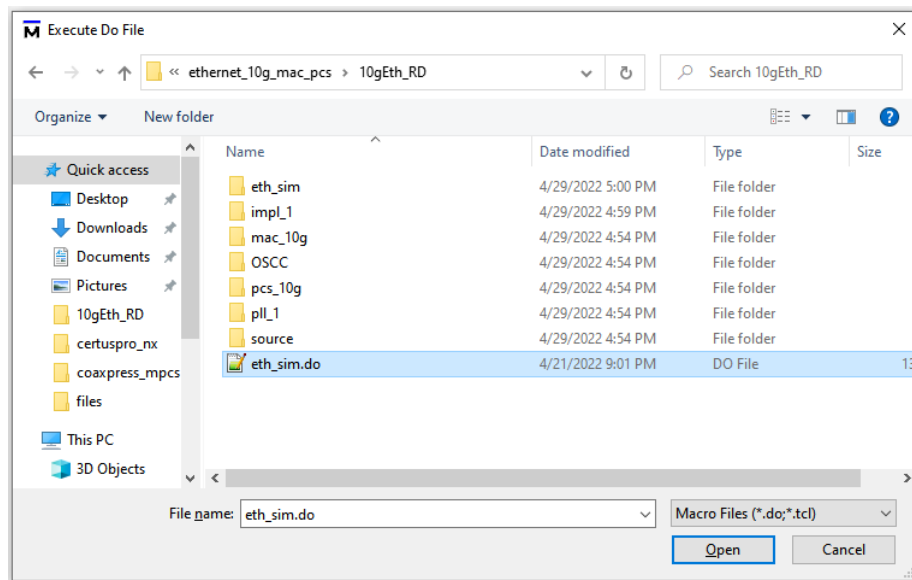


Figure 6.5. eth_sim.do in reference design directory

9. Wait for the simulation to complete to see the waveforms.

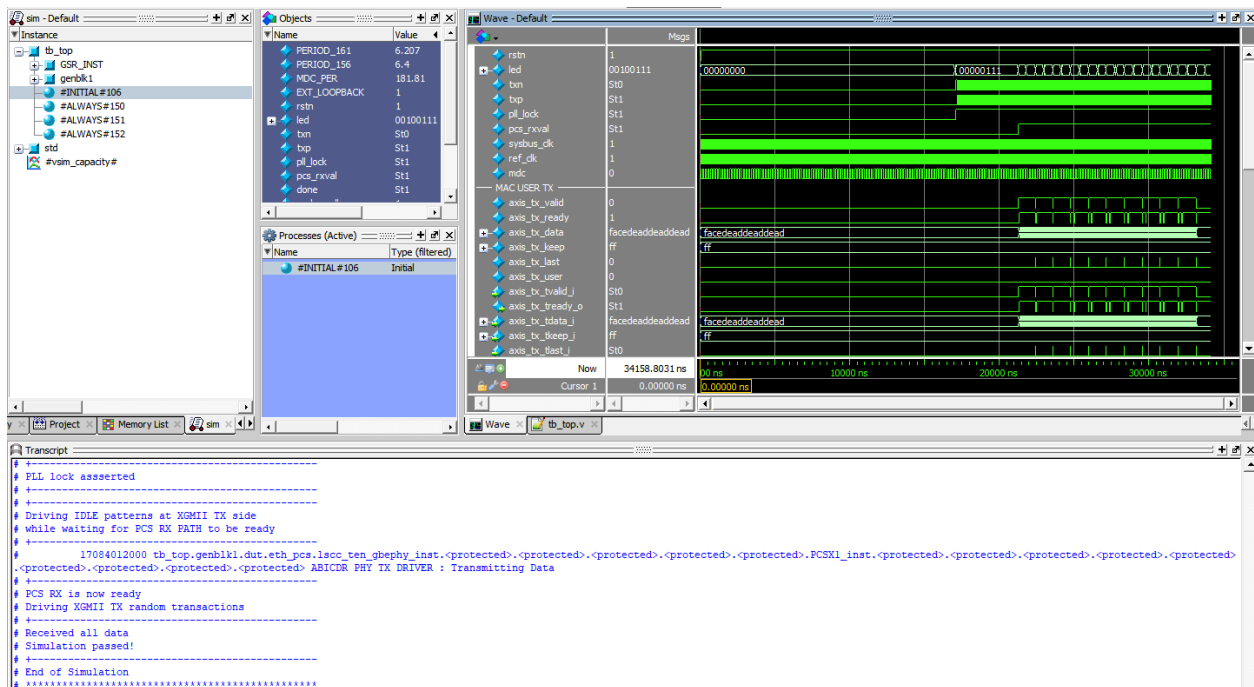
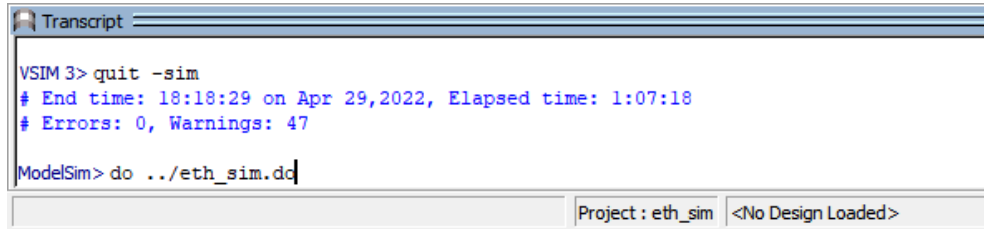


Figure 6.6. Simulation Waveform

10. Type **quit -sim** in the ModelSim console to terminate the simulation. To re-run the simulation, invoke the **eth_sim.do** file in the console.



```

Transcript
VSIM 3> quit -sim
# End time: 18:18:29 on Apr 29, 2022, Elapsed time: 1:07:18
# Errors: 0, Warnings: 47
ModelSim> do ../eth_sim.do
Project : eth_sim <No Design Loaded>

```

Figure 6.7. Unload the reference design and re-invoke eth_sim.do to re-run the simulation

6.1. Customization for Reference Design

6.1.1. Number of Packets and Packet Length

To change the **USR_DATA_BYTE** parameter in packet_gen_check.v for generating different length of packets, refer to [Figure 6.8](#).

To specify the number of packets **NUM_PKT** parameter to be generated in the **tb_top.v** file, refer to [Figure 6.9](#). The **NUM_PKT** is default to 10.



```

// 02/22/2022 ctc 1.0 Original
//
module packet_gen_check (
    clk, rstn,
    inReady, outData, outValid, outKeep, outUser, outLast,
    outReady, inData, inValid, inKeep, inUser, inLast,
    done,
    compareFail, inStart_patgen
);
    parameter CONTINUOUS_TRAFFIC = 0; // From top level: Generate packets continuously
    parameter NUM_PKT = 100; // From top level: Number of packets to generate when CONTINUOUS_TRAFFIC = 0

    parameter RANDOMIZE = 1; // 1 - Generate some junk data for user data bytes. 0 - user data bytes counting
    parameter USR_DATA_BYTE = 16'h05_5C; // Number of user data bytes

    parameter SRC_MAC_ADDR = 48'h00D0_BD00_1234;
    parameter DEST_MAC_ADDR = 48'h00D0_BD00_5678;
    parameter ETHERTYPE = 16'h0800;

    //UDP Header
    parameter UDP_SRC_PORT = 16'hC000;
    parameter UDP_DST_PORT = 16'hD0D8;
    parameter UDP_LENGTH = USR_DATA_BYTE + 16'h0008;

```

Figure 6.8. Modify Packet Contents in packet_gen_check.v



```

module tb_top:
    parameter PERIOD_161 = 5.207;
    parameter PERIOD_156 = 6.4;
    parameter MDC_PER = 181.81; // 5.5 Mhz
    parameter EXT_CONNECTION_EN = 1'b1; // 1 - select external interface connection. 0 - internal loopback - near FPGA loopback.
    // Note: PCS need to be generated with the corresponding setting.

    parameter CONTINUOUS_TRAFFIC = 0;
    parameter NUM_PKT = 10; // From top level: Number of packets to generate when CONTINUOUS_TRAFFIC = 0

```

Figure 6.9. NUM_PKT Parameter in Testbench file tb_top.v

6.1.2. Loopback Mode Configuration

In the reference design, the following two modes are in the PCS IP module for you to select depending on the test case:

- Near PMA Loopback mode
- Disabled Loopback mode

There are two settings that need to be configured for each mode before compiling the design.

6.1.2.1. Internal Loopback (near PMA Loopback) Mode

To configure the settings in this mode, follow these steps:

1. Make sure the 10 Gb Ethernet PCS IP instance (**pcs_10g.ipx**) is configured and generated for internal loopback (**Loopback Mode: near PMA Loopback**).

Module/IP Block Wizard

Configure Component from IP *ten_gbe_pcs* Version 1.3.0
Set the following parameters to configure this component.

Diagram pcs_10g

The diagram shows the **pcs_10g** component with various input and output ports. Inputs include **APB_S0**, **clk_sel_i[1:0]**, **diffioclk_sel_i**, **pad_refclk_n_i**, **pad_refclk_p_i**, **pad_rxn_i**, **pad_rxp_i**, **pll_0_refclk_i**, **pll_1_refclk_i**, **refclk_n0_ext_i**, **refclk_n1_ext_i**, **refclk_p0_ext_i**, **refclk_p1_ext_i**, **sd_pll_refclk_i**, **use_refmux_i**, **xg_pcs_clk_in_i**, **xg_rx_clk_i**, **xg_rx_rst_n_i**, **xg_tx_clk_i**, **xg_tx_rst_n_i**, **xg_bxc_i [7:0]**, **xg_bxd_i [63:0]**, and **xg_bval_i**. Outputs include **pad_txn_o**, **pad_txp_o**, **xg_rx_blk_lock_o**, **xg_rx_clk_o**, **xg_rx_hi_ber_o**, **xg_rxc_o [7:0]**, **xg_rxd_o [63:0]**, **xg_rxval_o**, **xg_tx_clk_o**, and **xg_txrdy_o**.

Configure IP

Property	Value
General	
Register Interface	APB
PCS Lane ID	6
RX Adaptive Equalization	
Setting1 Enable	☒
Setting1: Adaptive Algorithm	RL2plus
Setting2 Enable	☐
Setting2: Adaptive Algorithm	RL2plus
Setting3 Enable	☐
Setting3: Adaptive Algorithm	RL2plus
Setting1/2: Preliminary Adaptive EQ	ENABLED
Setting1/2: Training phase Adaptive EQ	ENABLED
Setting1/2: Post-phase Adaptive EQ	ENABLED
Setting3: Preliminary Adaptive EQ	DISABLED
Setting3: Training phase Adaptive EQ	DISABLED
Setting3: Post-phase Adaptive EQ	DISABLED
Test Feature	
Loopback Mode	Near PMA Loopback

No DRC issues are found.

Generate **Cancel**

Figure 6.10. 10 Gb Ethernet PCS IP with Loopback Mode = Near PMA Loopback

2. In the testbench file (**tb_top.v**), set the **EXT_CONNECTION_EN** parameter to 1'b0.

```

> module tb_top;

    parameter PERIOD_161 = 6.207;
    parameter PERIOD_156 = 6.4;
    parameter MDC_PER = 181.81; // 5.5 Mhz
    parameter EXT_CONNECTION_EN = 1'b0; // 1 - select external interface connection. 0 - internal loopback - near
    // Note: PCS need to be generated with the corresponding setting.

    parameter CONTINUOUS_TRAFFIC = 0;
    parameter NUM_PKT = 10; // From top level: Number of packets to generate when CONTINUOUS_TRAFFIC =

```

Figure 6.11. EXT_CONNECTION_EN Parameter for Internal Loopback in Testbench File tb_top.v

3. Re-run the simulation.

6.1.2.2. Disabled Loopback Mode

To configure the settings in this mode, follow these steps:

1. Make sure the 10 Gb Ethernet PCS IP instance (**pcs_10g.ipx**) is configured and generated for the external loopback (**Loopback Mode: Disabled**).

Module/IP Block Wizard

Configure Component from IP ten_gbe_pcs Version 1.3.0
Set the following parameters to configure this component.

Diagram pcs_10g

Configure IP

Property	Value
General	
Register Interface	APB
PCS Lane ID	6
RX Adaptive Equalization	
Setting1 Enable	☒
Setting1: Adaptive Algorithm	RL2plus
Setting2 Enable	☐
Setting2: Adaptive Algorithm	RL2plus
Setting3 Enable	☐
Setting3: Adaptive Algorithm	RL2plus
Setting1/2: Preliminary Adaptive EQ	ENABLED
Setting1/2: Training phase Adaptive EQ	ENABLED
Setting1/2: Post-phase Adaptive EQ	ENABLED
Setting3: Preliminary Adaptive EQ	DISABLED
Setting3: Training phase Adaptive EQ	DISABLED
Setting3: Post-phase Adaptive EQ	DISABLED
Test Feature	
Loopback Mode	Disabled

No DRC issues are found.

Generate **Cancel**

Figure 6.12. 10 Gb Ethernet PCS IP with Loopback Mode = Disabled

2. In testbench file (**tb_top.v**), set the EXT_CONNECTION_EN parameter to 1'b1.

```
module tb_top;

    parameter PERIOD_161      = 6.207;
    parameter PERIOD_156      = 6.4;
    parameter MDC_PER          = 181.81; // 5.5 Mhz
    parameter EXT_CONNECTION_EN = 1'b1; // 1 - select external interface connection. 0 - internal loopback - near
    // Note: PCS need to be generated with the corresponding setting.

    parameter CONTINUOUS_TRAFFIC = 0;
    parameter NUM_PKT            = 10; // From top level: Number of packets to generate when CONTINUOUS_TRAFFIC =
```

Figure 6.13. EXT_CONNECTION_EN Parameter for external interface connection in Testbench File tb_top.v

3. Re-run the simulation.
4. Select disabled loopback mode for board testing with SFP+ module. The setup is shown in section [7 Reference Design Implementation On Board](#).

7. Reference Design Implementation On Board

To use the external SFP module on board, select disabled loopback mode and configure the settings for the external path.

7.1. Requirements

The following lists the requirements for the reference design implementation on board:

- Two CertusPro™-NX Versa Boards
- At least 1 USB cable for programming
- Two sets of power supply
- Optic fiber cables
- Two sets of 10Gb small form-factor pluggable (SFP) module

7.2. CertusPro-NX Versa Board

For more information on CertusPro-NX Versa Board and schematics, refer to the [CertusPro-NX Versa Board](#) web page.

Caution: The CertusPro-NX Versa Board contains ESD-sensitive components. ESD safe practices should be followed while handling and using the development board.

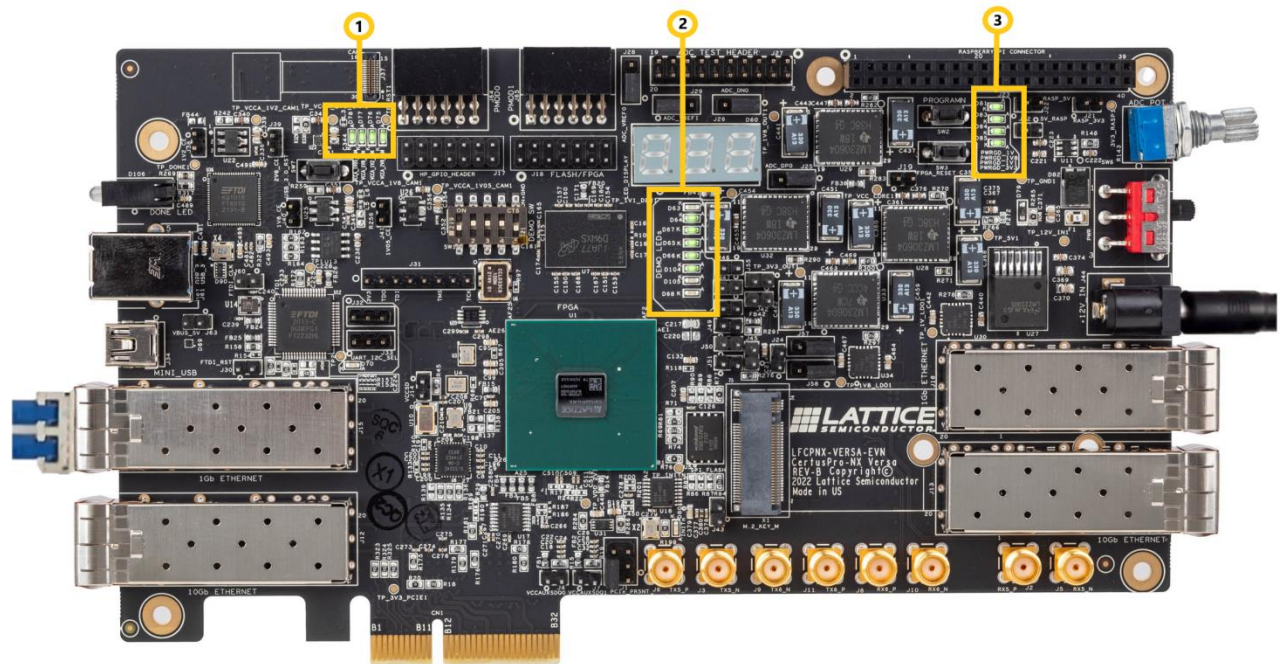


Figure 7.1. LEDs on the CertusPro-NX Versa Board

Table 7.1. Board LED Descriptions

Sl. No.	LEDs	Signal Name	Color	Purpose
1	D81	PWRGD 1V	Green	Lights up if voltage is present.
	D83	PWRGD 1V8	Green	Lights up if voltage is present.
	D84	PWRGD 1V1	Green	Lights up if voltage is present.
	D85	PWRGD 3V3	Green	Lights up if voltage is present.
2	D63	MAC PLL Lock	Green	Lights up when GPLL locked signal.
	D64	MPCS PHY Ready	Green	Lights up when PCS Tx ready signal. When high, indicates that XGMII PCS is ready to accept user data.
	D67	Sys Ready	Green	Lights up when PLL lock and mpcs_phy_ready.
	D65	AXI TX Valid	Green	Lights up when MAC TX AXI4-Stream interface bit for data valid. To show transmit in progress.
	D66	AXI RX Valid	Green	Lights up when MAC RX AXI4-Stream interface bit for data valid. To show receive in progress.
	D104	Pattern Generator Done	Green	Lights up indicates that all the packets specified by NUM_PKT parameter have been generated by the pattern generator.
	D105	Compare Fail	Green	Lights up when packet compare status failed. 1'b1 = mismatch. 1'b0 = match.
	D68	MPCS RX Hi BER	Green	Lights up when PCS high bit error rate (BER) indicator.
3	D78	1.05 V	Green	Lights up if voltage is present.
	D76	1.2 V	Green	Lights up if voltage is present.
	D77	1.8 V	Green	Lights up if voltage is present.
	D75	2.8 V	Green	Lights up if voltage is present.

Note:

- These eight demo LEDs are mapped to corresponding signals in the design file top_module.v. The LEDs are only meaningful after bit file programming.

7.3. Board-to-Board Testing

This section provides the details of the reference design hardware testing on board and guides you through the process of setting up IP configuration and uploading the .bit file.

7.3.1. Design Setup

In the PCS IP configuration, choose **Disabled** in the loopback mode category and change the **EXT_CONNECTION_EN** to bit 1 in the top.v file. After completing these two settings, compile the reference design project in the Lattice Radiant™ software to produce the .bit file to download into the board.

Module/IP Block Wizard

Configure Component from IP ten_gbe_pcs Version 1.3.0
Set the following parameters to configure this component.

Diagram pcs_10g

ten_gbe_pcs

Configure IP

Property	Value
General	
Register Interface	APB
PCS Lane ID	6
RX Adaptive Equalization	
Setting1 Enable	☒
Setting1: Adaptive Algorithm	RL2plus
Setting2 Enable	☐
Setting2: Adaptive Algorithm	RL2plus
Setting3 Enable	☐
Setting3: Adaptive Algorithm	RL2plus
Setting1/2: Preliminary Adaptive EQ	ENABLED
Setting1/2: Training phase Adaptive EQ	ENABLED
Setting1/2: Post-phase Adaptive EQ	ENABLED
Setting3: Preliminary Adaptive EQ	DISABLED
Setting3: Training phase Adaptive EQ	DISABLED
Setting3: Post-phase Adaptive EQ	DISABLED
Test Feature	
Loopback Mode	Disabled

No DRC issues are found.

Generate **Cancel**

Figure 7.2. PCS IP Loopback Mode

```

module top (
    /* pads */
    pad_refclk_i, pad_refclkp_i, pad_rxn_i, pad_rxp_i, pad_txn_o, pad_txp_o,
    led, pll_lock, pcs_rxval, patgen_done_o, start_patgen_i,
    rstn, sfp_tx_disable_o, cmpfail_o
);

parameter CONTINUOUS_TRAFFIC = 1'b1; // Generate packets continuously
parameter NUM_PKT = 10; // Number of packets to generate when CONTINUOUS_TRAFFIC = 0
parameter EXT_CONNECTION_EN = 1'b1; // 1 - external interface connection. 0 - internal loopback - near FMA loopback

```

Figure 7.3. Change the EXT_CONNECTION_EN Parameter to 1 in Top File top.v

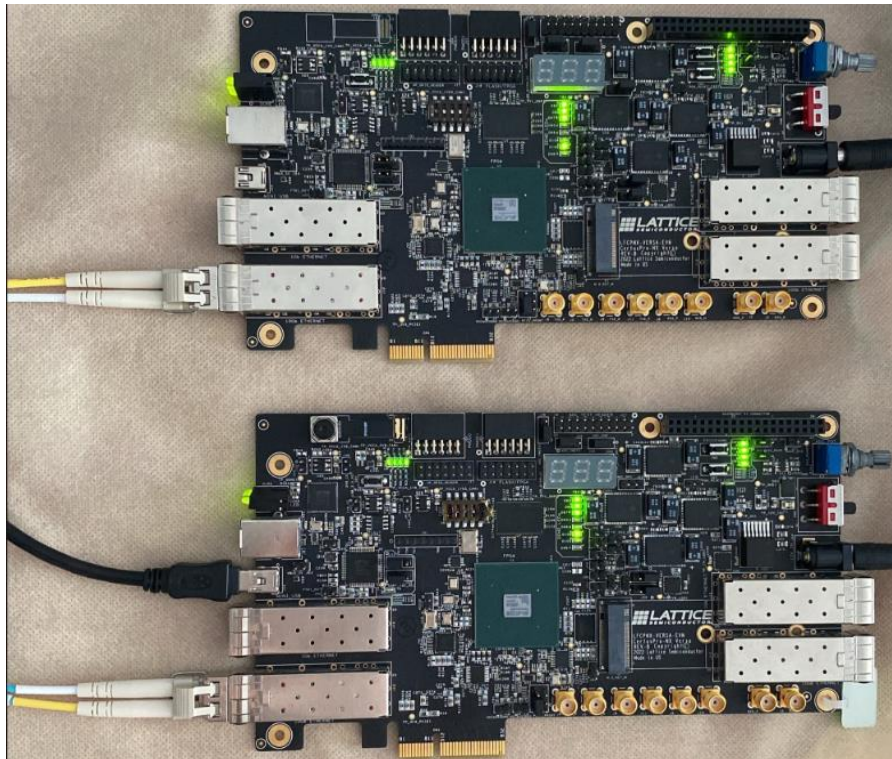


Figure 7.4. CertusPro-NX Versa Board-to-Board Connection

There are two flows of data when connecting board to board. In [Figure 7.5](#), Board A TX1 sends data out from the SFP TX1 through the optic fiber cable to Board B SFP RX2 and data is received by Board B RX2, vice versa when Board B sends data out to Board A.

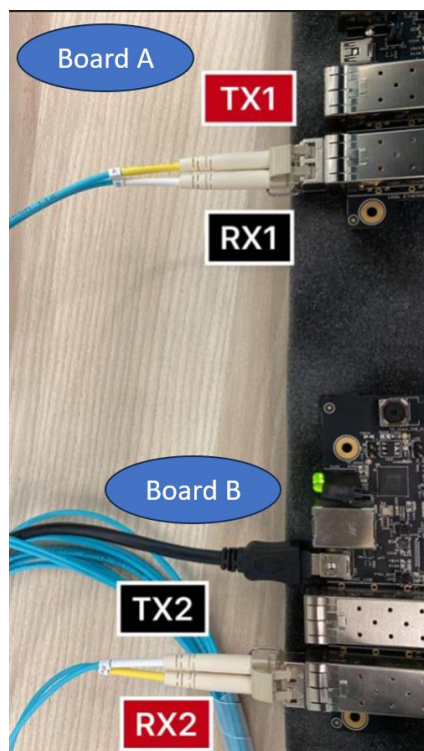


Figure 7.5. Optic Fiber Cables Connection to Boards

7.3.2. Bitstream Programming

Make sure the board is connected to the PC with the Mini USB Type A cable as shown in [Figure 7.4](#). After you have programmed Board A, repeat the same for Board B so that both boards have the same program.

To run the reference design simulation, follow these steps:

1. Click on the Radiant Programmer icon  if the project has already opened inside the Radiant software. Otherwise, open the standalone Radiant Programmer. Select the Project Name, Project Location, and click **OK**.

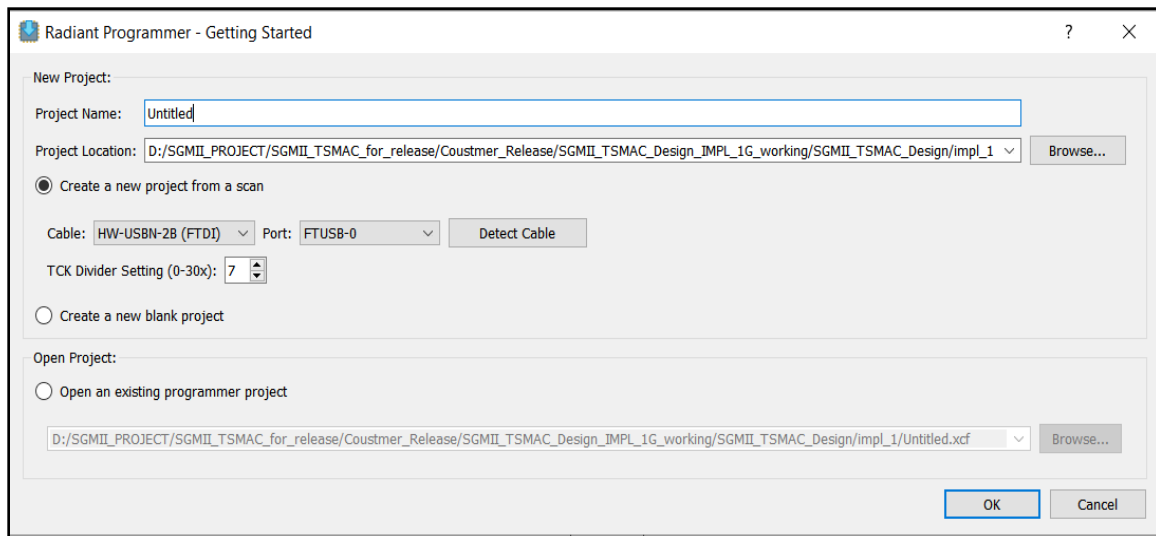


Figure 7.6. Standalone Radiant Programmer

The Radiant Programmer window is shown in [Figure 7.7](#).

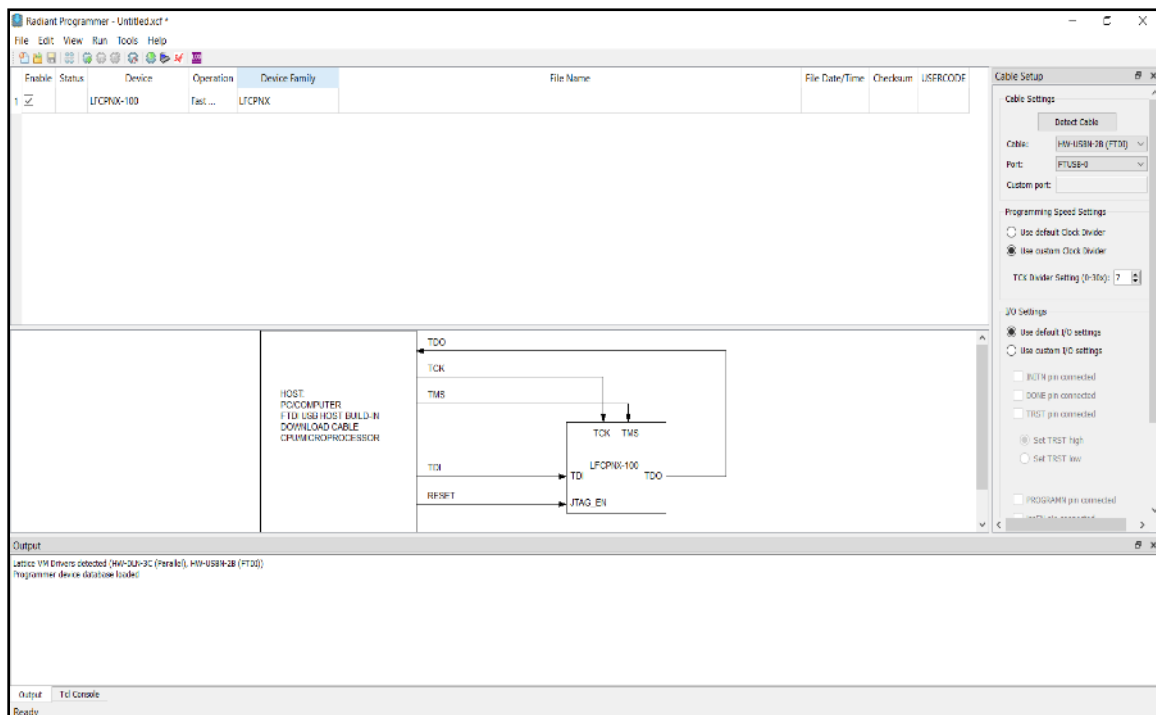


Figure 7.7. Radiant Programmer Window

- Configure the Device Family, Device, and the bit file fields.

	Enable	Status	Device Family	Device	Operation	File Name	File Date/Time	Checksum	USERCODE
1	☒	PASS	LFCPNX	LFCPNX-100	...m_Verify	C:/Users/prince/Downloads/...	04/29/22 ...	0x6FF0	

Figure 7.8. Radiant Programmer: Configure the Settings

- Click on the program device toolbar icon as shown in Figure 7.9.

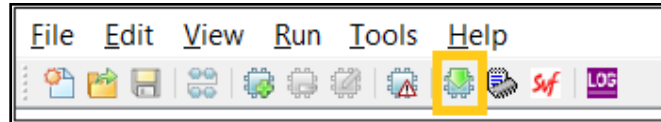


Figure 7.9. Program Device Toolbar Icon

- Check on the output window, where the status of the program device should be *Operation: successful*, as shown in Figure 7.10.

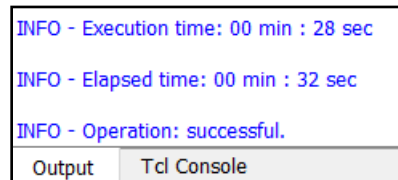


Figure 7.10. Message on Successful Programming

- After programming, restart the board and check the status of the done LED. It indicates that the .bit file has booted successfully into the FPGA.
- Connect the optic fiber cables (TX-RX) as shown in Figure 7.5.
- Upon board-to-board connection, press the reset switch (SW3) and the MAC PLL Lock (D63), MPC5 PHY Ready (D64) and Sys Ready (D67) will light up the green LED.
- Toggle switch 5 in SW1 to start the pattern generator (highlighted in Figure 7.11.)

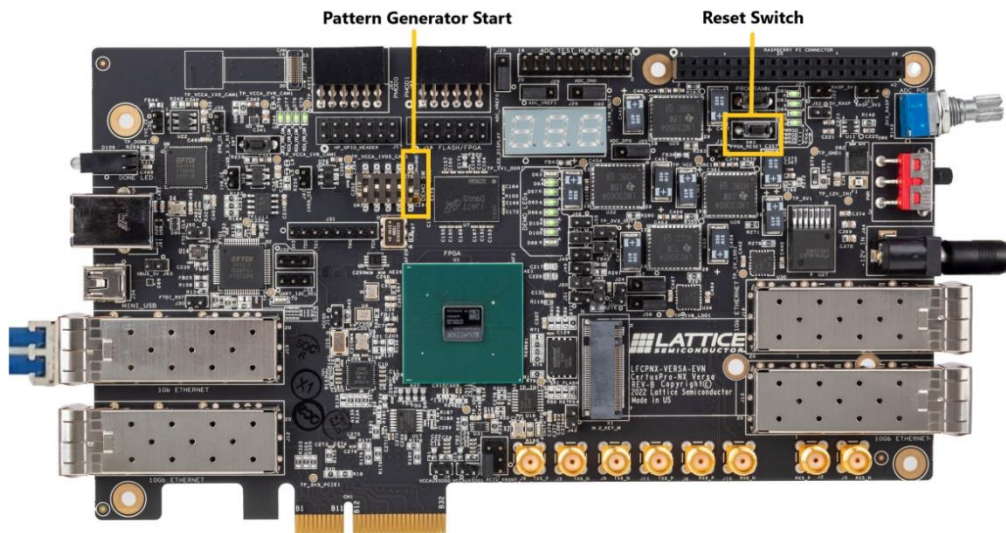


Figure 7.11. Toggle Switch 5 in SW1 to Initiate Pattern Generation for CertusPro-NX Versa Board

7.4. Waveforms Captured Using the Reveal Tool

You can monitor the value of the signals in real-time mode by inserting signals to capture in the Reveal Inserter and observe the signals in the Reveal Analyzer.

7.4.1. Reveal Inserter

The Reveal Inserter enables you to select which design signals to use for debug tracing or triggering, then generate a core based on these signals and their use. After generating the required core, it generates a modified design with the necessary debug connections and links it to the signals. After the design has been modified for debug, it is mapped, placed, and routed with the normal design flow in the Radiant software.

The Reveal Inserter handles the task of inserting the debug logic into the design. Before using the Reveal Analyzer, you must use the Reveal Inserter to allow debug access.

To add the Reveal Inserter into the design, follow these steps:

1. Click on the Reveal Inserter icon  in the toolbar to create a new Reveal Inserter project or open an existing Reveal Inserter project with pre-added signals (eth.rvl) .rvl, which is shown in the file list. If you open an existing file, you can skip this step and proceed to step 5.

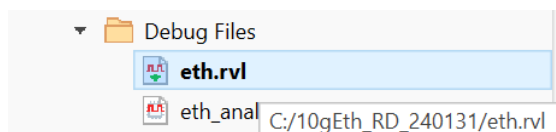


Figure 7.12. Reveal Inserter File in the File List

2. Click on the **Add core** on the top left of the Reveal Inserter file and select **Add Logic Analyzer**.

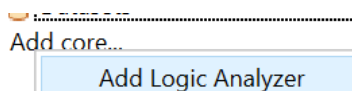


Figure 7.13. Add Logic Analyzer into the Debug File

3. In the debug file, set up the trace signals by dragging the signals into the Trace Signal Setup Tab, as shown in [Figure 7.14](#). Remember to put in the Sample Clock based on the clock driving the trace signals. However, in designs with multiple clock regions, it may be necessary to sample different clock regions at the same time. For those types of designs, it is recommended that you create multiple cores, one for each clock region where the clock is used as the sample clock for the core.

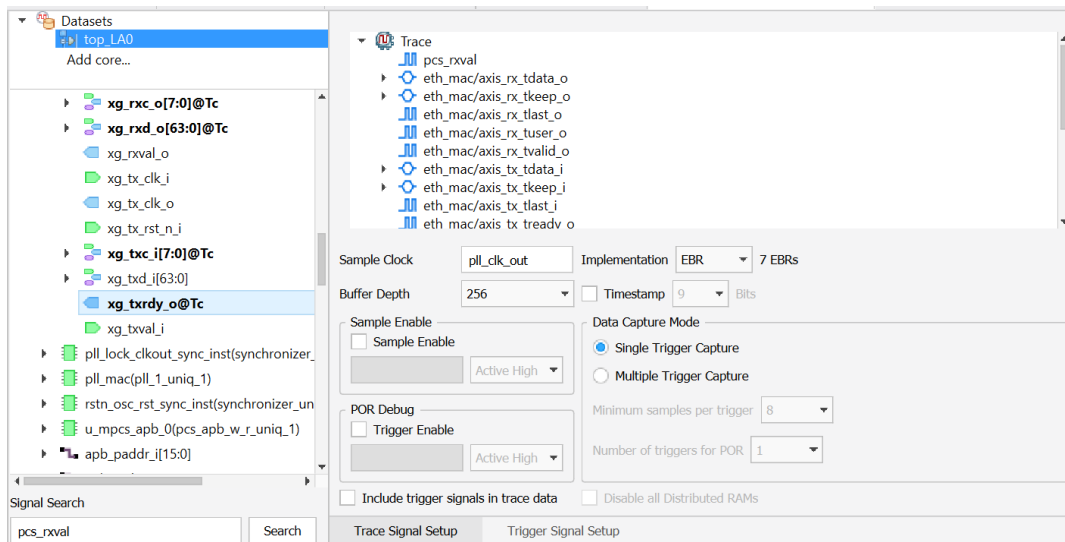


Figure 7.14. Trace Signal Setup

- After adding the signals, set up the trigger signals in the **Trigger Signal Setup Tab** as shown in Figure 7.15. Click on the top left icon with a magnifying glass  to check if the design rule is passing and click on the  icon to insert the debug logic. This process generates and synthesizes the necessary debug logic. The generated .rvl is automatically imported into the Radiant software.

It builds a version of the design that contains the necessary connections for the debug logic to the signals that you are tracing or triggering.

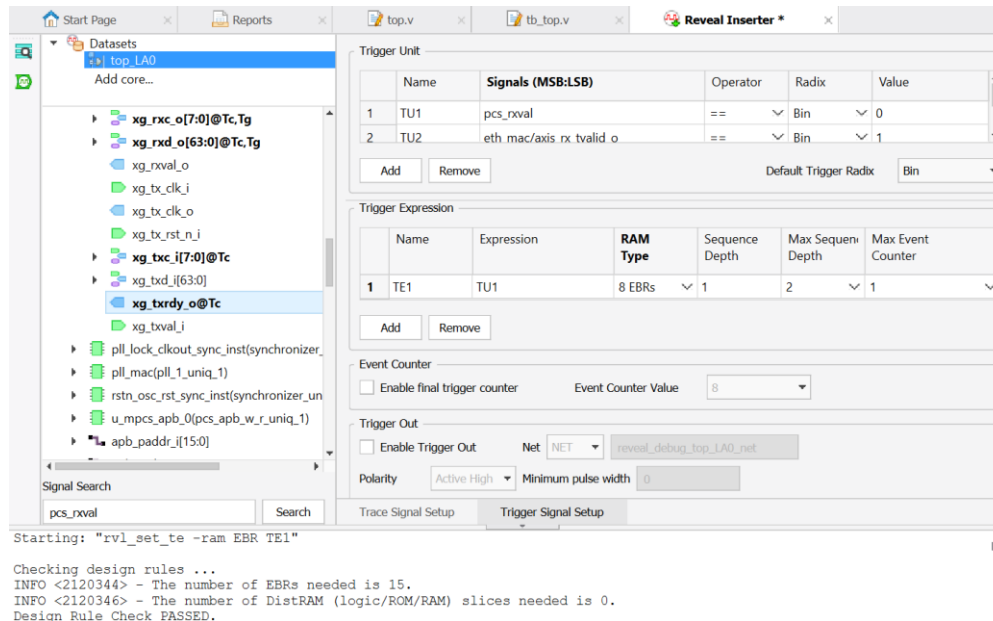


Figure 7.15. Trigger Signal Setup

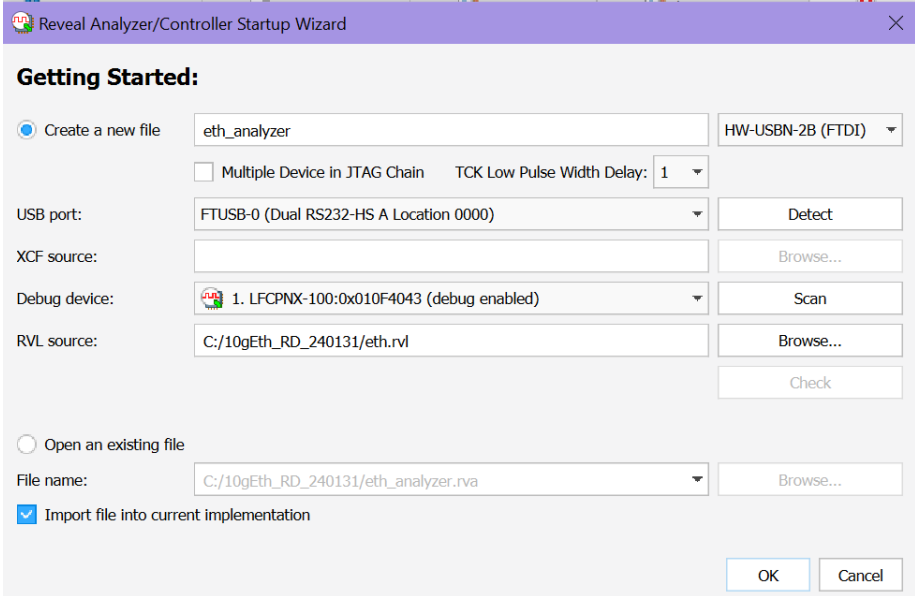
- Resynthesize, map, place and route the design.
- Generate the bitstream data to perform logic analysis with the Reveal Analyzer.
- Set up the cable connection with the Programmer.
- Download the design into the device.
- Start the Reveal Analyzer and perform logic analysis with it.

7.4.2. Reveal Analyzer

The Reveal Analyzer enables you to configure trigger settings and extract information from a programmed device through the JTAG ports. It interfaces directly to the Reveal cores in the design. You can set up triggers, select capture modes, and run or stop the triggers. The Reveal Analyzer displays the data captured on the silicon according to the settings that has been specified in the Reveal Inserter.

To launch the Reveal Analyzer, follow these steps:

1. Click on the Reveal Analyzer icon  in the toolbar and create a new Reveal Analyzer file or open an existing file (eth_analyzer.rva).
2. Select **Create a new file** option. Configure the new file settings as shown in [Figure 7.16](#).
3. Click **Detect** to detect the micro-USB that has been connected to the board after downloading the bitstream that has been programmed with the Reveal Inserter.
4. Click **Scan** to detect the debug device.
5. Click **OK** to proceed.



Reveal Analyzer/Controller Startup Wizard

Getting Started:

☒ Create a new file eth_analyzer HW-USBN-2B (FTDI) ▼

☐ Multiple Device in JTAG Chain TCK Low Pulse Width Delay: 1 ▼

USB port: FTUSB-0 (Dual RS232-HS A Location 0000) Detect

XCF source: Browse...

Debug device: 1. LFCPNX-100:0x010F4043 (debug enabled) Scan

RVL source: C:/10gEth_RD_240131/eth.rvl Browse...

Check

☐ Open an existing file

File name: C:/10gEth_RD_240131/eth_analyzer.rva Browse...

☒ Import file into current implementation

OK Cancel

Figure 7.16. Create a New Reveal Analyzer File

6. Configure the trigger settings in the **LA Trigger** tab for each core in each device as shown in [Figure 7.17](#) to perform logic analysis of the design.



- The eth_mac/axis_rx_tvalid_o signal is asserted to indicate the starting of the frame.
- The eth_mac/axis_rx_tlast_o is asserted to indicate the last packet transfer.
- The eth_mac/axis_rx_tvalid_o is deasserted to indicate the end of active packets.



© 2022-2024 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.

Bus/Signal	Data	0-15	0-16	0-17	0-18	0-19	0-20	0-21
pcs_nval	1							
eth_mac/axis_rx_tvalid_o	1							
eth_mac/axis_rx_tdata_o	EDF16DFF	D000785600BDD0000 AA4500083412008D 110B0000CC007805 00B05C1C00A00000 6405D8D000C0D81C 000A000000003412						
eth_mac/axis_rx_tkeep_o	FF	FF						
eth_mac/axis_rx_tlast_o	0							
eth_mac/axis_rx_tuser_o	0							
cmpfail_o	0							

Figure 7.19. MAC axis_rx_tdata_o packets

eth_mac/axis_tx_tdata_i	D000785E	D000785600BDD0000 AA4500083412008D 110B0000CC007805 00B05C1C00A00000 6405D8D000C0D81C 000A000000003412						
eth_mac/axis_tx_tkeep_i	FF	FF						
eth_mac/axis_tx_tlast_i	0							
eth_mac/axis_tx_tvalid_i	1							
eth_mac/axis_tx_tuser_i	0							

Figure 7.20. MAC axis_tx_data_i packets

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, March 2024

Section	Change Summary
Acronyms in This Document	Added the following acronyms: - LSE - SFP - TX - RX
Features	Re-organized the content in this section.
Functional Description	Updated the description for the <i>led[7:0]</i> signal in Table 3.1. Signal Descriptions .
Reference Design Simulation	- Removed step 11 and 12. - Added section 6.1 Customization for Reference Design.
Reference Design Implementation On Board	Added this section.

Revision 1.1, May 2023

Section	Change Summary
All	Minor adjustments in formatting across the document.
Acronyms in This Document	Added this section.
Functional Description	Updated Tx clock output statement to specify GPLL used to generate 156.25 MHz clock must not enable Fractional-N Divider.
Technical Support Assistance	Added reference to the Lattice Answer Database on the Lattice website.

Revision 1.0, July 2022

Section	Change Summary
All	Initial release. Verified in hardware using CertusPro-NX 10GbE Mini Board.

Revision 0.80, Feb 2022

Section	Change Summary
All	Preliminary revision.



www.latticesemi.com