Low Latency Ethernet 10G MAC Intel Cyclone 10 GX FPGA IP Design Example User Guide

Updated for Intel® Quartus® Prime Design Suite: **18.1**

Subscribe

Send Feedback

UG-20162 | 2018.10.03

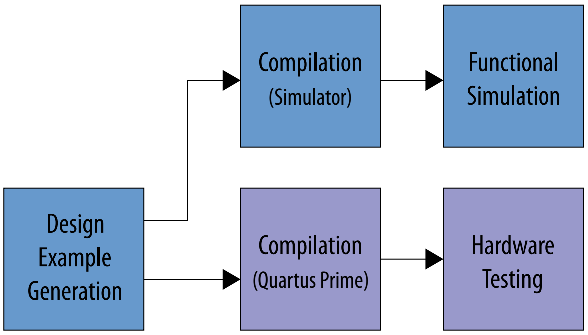
Latest document on the web: PDF | HTML

Contents

1. Quick Start Guide

The Intel® FPGA Low Latency 10G Ethernet (LL 10GbE) MAC Intel FPGA IP core for Intel Cyclone® 10 GX devices provides the capability of generating design examples for selected configurations.

Figure 1. Development Stages for the Design Example



Related Information

- 10GBASE-R Ethernet Design Example for Intel Cyclone 10 GX Devices on page 12
  Provides details on the 10GBASE-R design example.
- 10M/100M/1G/2.5G/5G/10G (USXGMII) Ethernet Design Example for Intel Cyclone 10 GX Devices on page 19
  Provides details on the 10M/100M/1G/2.5G/5G/10G (USXGMII) Ethernet design example.

1.1. Directory Structure

Figure 2. Directory Structure for the Design Example

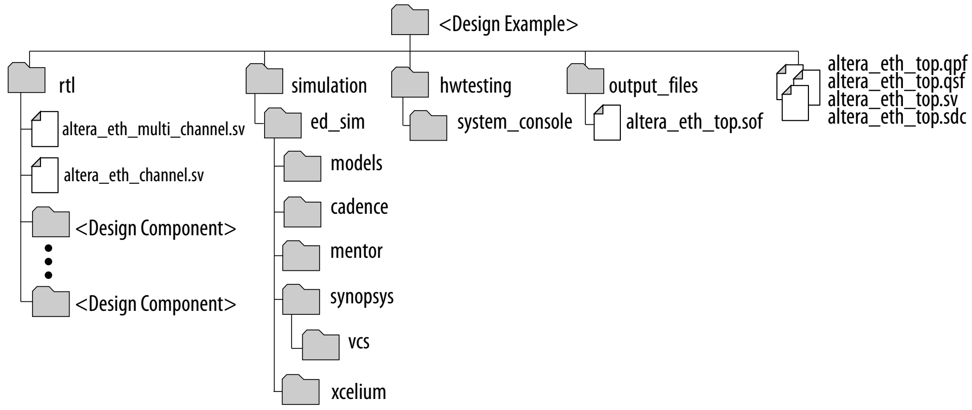


Table 1. Directory and File Description

Directory/File	Description
`altera_eth_top.qpf`	Intel Quartus® Prime project file.
`altera_eth_top.qsf`	Intel Quartus Prime settings file.
`altera_eth_top.sv`	Design example top-level HDL.
`altera_eth_top.sdc`	Synopsys Design Constraints (SDC) file.
`rtl`	The folder that contains the design example synthesizable components.
`rtl/altera_eth_10g_mac_base_r.sv` `rtl/altera_10g_mac_base_r_wrap.v`	Design example DUT top-level files for 10GBASE-R Ethernet design example.
`rtl/altera_mge_multi_channel.sv` `rtl/altera_mge_channel.v`	Design example DUT top-level files for 10M/100M/1G/2.5G/5G/10G (USXGMII) Ethernet design example.
`rtl/<Design Component>`	The folder for each synthesizable component including Platform Designer generated IPs, such as LL 10GbE MAC, PHY, and FIFO.
`simulation/ed_sim/models`	The folder that contains the testbench files.

`simulation/ed_sim/cadence`
`simulation/ed_sim/mentor`
`simulation/ed_sim/synopsys/vcs`
`simulation/ed_sim/xcelium`
	The folder that contains the simulation script. It also serves as a working area for the simulator.

`hwtesting/system_console`	The folder that contains system console scripts for hardware testing.
`output_files`	The folder that contains Intel Quartus Prime output files including Intel Quartus Prime compilation reports and design programing file (`.sof` file).

1.2. Generating the Design

1.2.1. Procedure

You can generate the design example from the IP Parameter Editor.

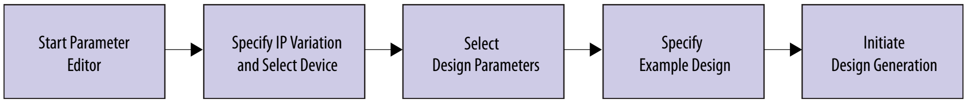
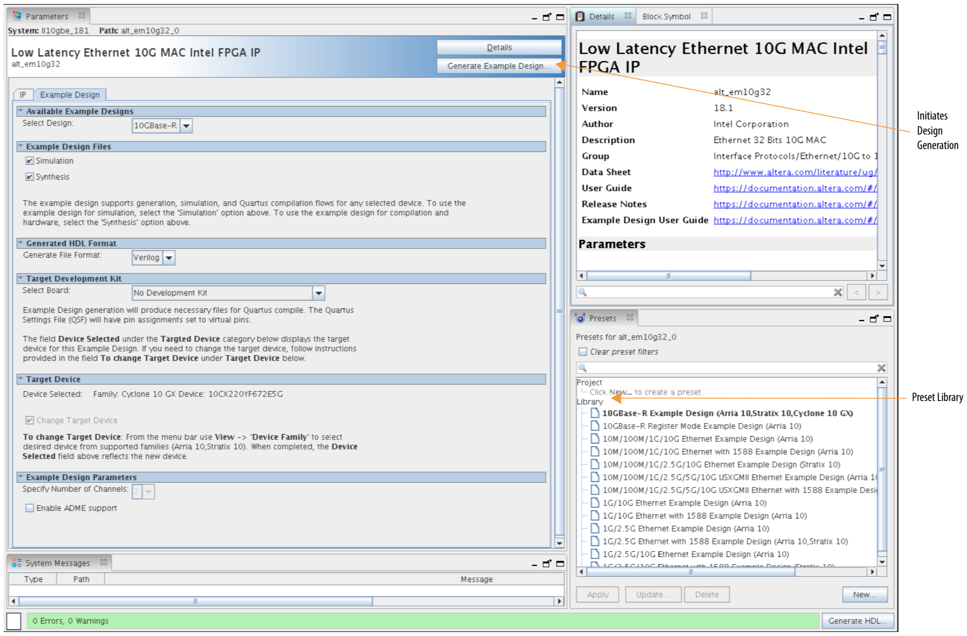


Figure 3. Example Design Tab



System: ll10gbe_181 **Path:** alt_em10g32_0

Low Latency Ethernet 10G MAC Intel FPGA IP
alt_em10g32

IP | Example Design

Available Example Designs

Select Design: 10GBase-R

Example Design Files

☑ Simulation

☑ Synthesis

The example design supports generation, simulation, and Quartus compilation flows for any selected device. To use the example design for simulation, select the 'Simulation' option above. To use the example design for compilation and hardware, select the 'Synthesis' option above.

Generated HDL Format

Generate File Format: Verilog

Target Development Kit

Select Board: No Development Kit

Example Design generation will produce necessary files for Quartus compile. The Quartus Settings File (QSF) will have pin assignments set to virtual pins.

The field **Device Selected** under the **Targted Device** category below displays the target device for this Example Design. If you need to change the target device, follow instructions provided in the field **To change Target Device** under **Target Device** below.

Target Device

Device Selected: Family: Cyclone 10 GX Device: 10CX220YF672E5G

☑ Change Target Device

To change Target Device: From the menu bar use **View** -> '**Device Family**' to select desired device from supported families (Arria 10,Stratix 10). When completed, the **Device Selected** field above reflects the new device.

Example Design Parameters

Specify Number of Channels: 2

☐ Enable ADME support

Type	Path	Message

0 Errors, 0 Warnings

Low Latency Ethernet 10G MAC Intel FPGA IP

Name	alt_em10g32
Version	18.1
Author	Intel Corporation
Description	Ethernet 32 Bits 10G MAC
Group	Interface Protocols/Ethernet/10G to 1
Data Sheet	http://www.altera.com/literature/ug/
User Guide	https://documentation.altera.com/#/
Release Notes	https://documentation.altera.com/#/
Example Design User Guide	https://documentation.altera.com/#/

Parameters

Presets for alt_em10g32_0

☐ *Clear preset filters*

Project
Click New... to create a preset.
Library
10GBase-R Example Design (Arria 10,Stratix 10,Cyclone 10 GX)
10GBase-R Register Mode Example Design (Arria 10)
10M/100M/1G/10G Ethernet Example Design (Arria 10)
10M/100M/1G/10G Ethernet with 1588 Example Design (Arria 10)
10M/100M/1G/2.5G/10G Ethernet Example Design (Stratix 10)
10M/100M/1G/2.5G/5G/10G USXGMII Ethernet Example Design (Arria 1
10M/100M/1G/2.5G/5G/10G USXGMII Ethernet with 1588 Example Desi
1G/10G Ethernet Example Design (Arria 10)
1G/10G Ethernet with 1588 Example Design (Arria 10)
1G/2.5G Ethernet Example Design (Arria 10)
1G/2.5G Ethernet with 1588 Example Design (Arria 10,Stratix 10)
1G/2.5G/10G Ethernet Example Design (Arria 10)

1. Select **Tools ➤ IP Catalog** to open the IP Catalog and select **Low Latency Ethernet 10G MAC Intel FPGA IP**.
   The IP parameter editor appears.
2. Specify a top-level name and the folder for your custom IP variation, and the target device. Click **OK**.
3. To generate a design example, select a design example preset from the **Presets** library and click **Apply**. When you select a design, the system automatically populates the IP parameters for the design.

   The Parameter Editor automatically sets the parameters required to generate the design example. Do not change the preset parameters in the **IP** tab.
4. Specify the parameters in the **Example Design** tab.
5. Click the **Generate Example Design** button.

The software generates all design files in sub-directories. You require these files to run simulation, compilation, and hardware testing.

Related Information

Directory Structure on page 5
Provides more information about the generated design example directories and files.

1.2.2. Design Example Parameters

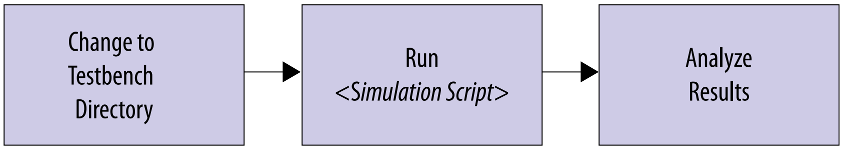
Table 2. Parameters in the Example Design Tab

Parameter	Description
Select Design	Available example designs for the IP parameter settings. When you select an example design from the Preset library, this field shows the selected design.
Example Design Files for Simulation or Synthesis	The files to generate for the different development phase. • Simulation—generates the necessary files for simulating the example design. • Synthesis—generates the synthesis files. Use these files to compile the design in the Intel Quartus Prime Pro Edition software for hardware testing and perform static timing analysis.
Generate File Format	The format of the RTL files for simulation—Verilog or VHDL.
Select Board	Supported hardware for design implementation. When you select an Intel FPGA development board, the *Target Device* is the one that matches the device on the Development Kit. If this menu is not available, there is no supported board for the options that you select. **Intel Cyclone 10 GX FPGA Development Kit**: This option allows you to test the design example on the selected Intel FPGA IP development kit. This option automatically selects the *Target Device* to match the device on the Intel FPGA IP development kit. If your board revision has a different device grade, you can change the target device. **Custom Development Kit:** This option allows you to test the design example on a third party development kit with Intel FPGA IP device, a custom designed board with Intel FPGA IP device, or a standard Intel FPGA IP development kit not available for selection. You can also select a custom device for the custom development kit. **No Development Kit**: This option excludes the hardware aspects for the design example.
Change Target Device	Select this parameter to display and select all devices for the Intel FPGA IP development kit.
Specify Number of Channels	The number of Ethernet channels. For Intel Cyclone 10 GX devices, the default number of channels is 2 and this parameter is not selectable.
Enable ADME support	Turn on this option to enable Transceiver ADME feature. *Note:* This option is only available from Intel Quartus Prime Pro Edition version 17.1 onwards

1.3. Compiling and Simulating the Design

1.3.1. Procedure

You can compile and simulate the design by running a simulation script from the command prompt.



1. At the command prompt, change the working directory to <*Example Design*>\simulation\ed_sim\<*Simulator*>.
2. Run the simulation script for the simulator of your choice.

Simulator	Working Directory	Command
ModelSim*	<*Example Design*>/simulation/ed_sim/mentor	`vsim -c -do tb_run.tcl`
VCS*	<*Example Design*>/simulation/ed_sim/synopsys/vcs	`sh tb_run.sh`
NCSim	<*Example Design*>/simulation/ed_sim/cadence	`sh tb_run.sh`
Xcelium*	<*Example Design*>/simulation/ed_sim/xcelium	`sh tb_run.sh`

A successful simulation ends with the following message:

```
Simulation passed.
```

After successful completion, you can analyze the results.

1.3.2. Testbench

Figure 4. Block Diagram of the Testbench

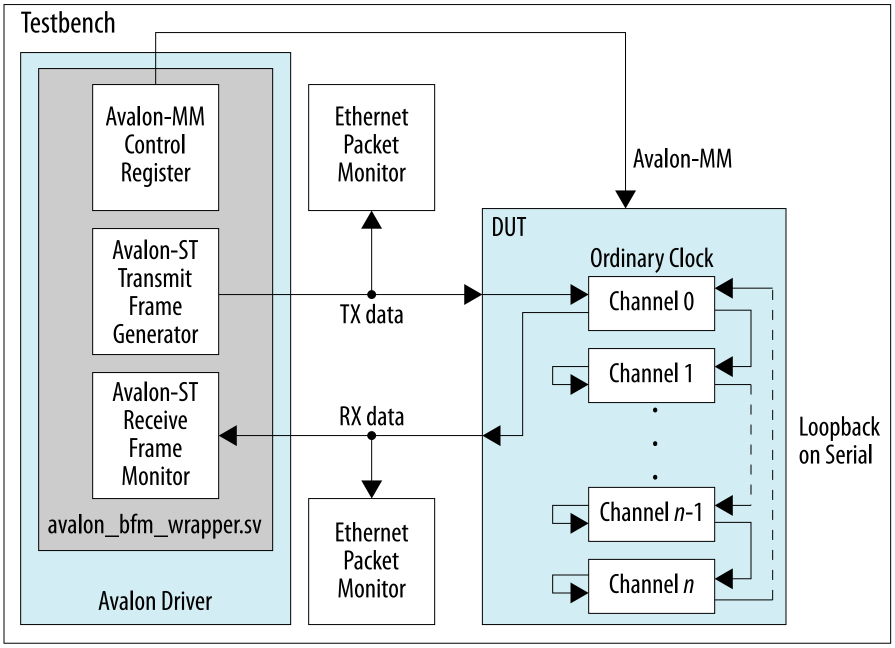


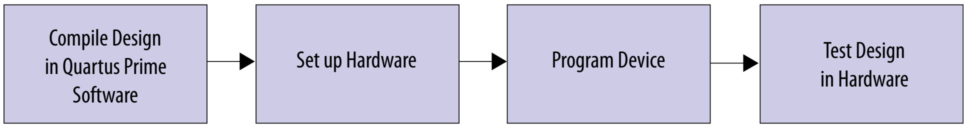
Table 3. Testbench Components

Component	Description
Device under test (DUT)	The design example.
Avalon driver	Consists of Avalon-ST master bus functional models (BFMs). This driver forms the TX and RX paths. The driver also provides access to the Avalon-MM interface of the DUT.
Ethernet packet monitors	Monitor TX and RX datapaths, and display the frames in the simulator console.

1.4. Compiling and Testing the Design in Hardware

1.4.1. Procedure

You can compile and test the design in the supported Intel FPGA development kit.



1. Launch the Intel Quartus Prime software and open the design example project file. Select **Processing ➤ Start Compilation** to compile the design example.

   The timing constraints for the design example and the design components are automatically loaded during compilation.
2. Connect the development board to the host computer.
3. Launch the Clock Controller application, which is part of the development kit, and set new frequencies for the design example.

   Note: For the frequencies to set, refer to the *Hardware Testing* section in the respective design example chapter.
4. In the Intel Quartus Prime software, select **Tools ➤ Programmer** to configure the FPGA on the development board using the generated `.sof` file.
5. Reset the system by pressing the `PB0` push button.
6. In the Intel Quartus Prime software, select **Tools ➤ System Debugging Tools ➤ System Console** to launch the system console.
7. Change the working directory to `<`*`Example Design`*`>\hwtesting\system_console`.
8. Initialize the design command list by running this command: `source main.tcl`.

   Note: For a design example that does not provide the `main.tcl` file, refer to the *Hardware Testing* section in the respective design example chapter.

You can now run any of the predefined hardware tests from the System Console.

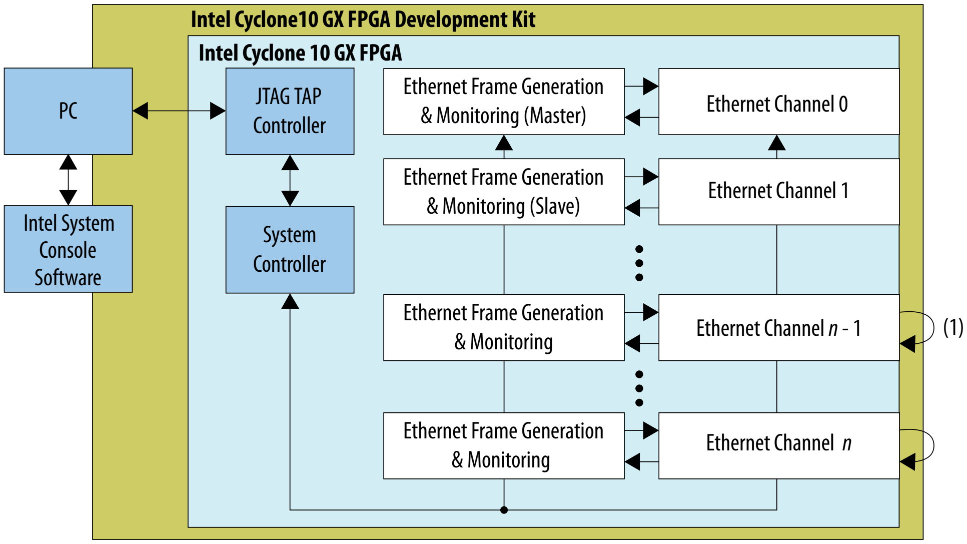
Observe the test results displayed.

Related Information

Intel Cyclone 10 GX FPGA Development Kit Webpage

1.4.2. Hardware Setup

Figure 5. Block Diagram of the Hardware Setup


Intel Cyclone10 GX FPGA Development Kit
Intel Cyclone 10 GX FPGA
PC
Intel System Console Software
JTAG TAP Controller
System Controller
Ethernet Frame Generation & Monitoring (Master)
Ethernet Frame Generation & Monitoring (Slave)
Ethernet Frame Generation & Monitoring
Ethernet Frame Generation & Monitoring
Ethernet Channel 0
Ethernet Channel 1
Ethernet Channel *n* - 1
Ethernet Channel *n*
(1)

(1) Use this type of loopback to test features other than IEEE 1588v2.

2. 10GBASE-R Ethernet Design Example for Intel Cyclone 10 GX Devices

The 10GBASE-R Ethernet design example demonstrates an Ethernet solution for Intel Cyclone 10 GX devices using the LL 10GbE MAC Intel FPGA IP core, the native PHY IP core, and a FPGA Mezzanine Card (FMC) module.

Generate the design example from the **Example Design** tab of the LL 10GbE Intel FPGA IP parameter editor.

2.1. Features

- Supports single Ethernet channels operating at 10G using Intel Cyclone 10 GX Native PHY.
- Packet monitoring on the TX and RX datapaths.

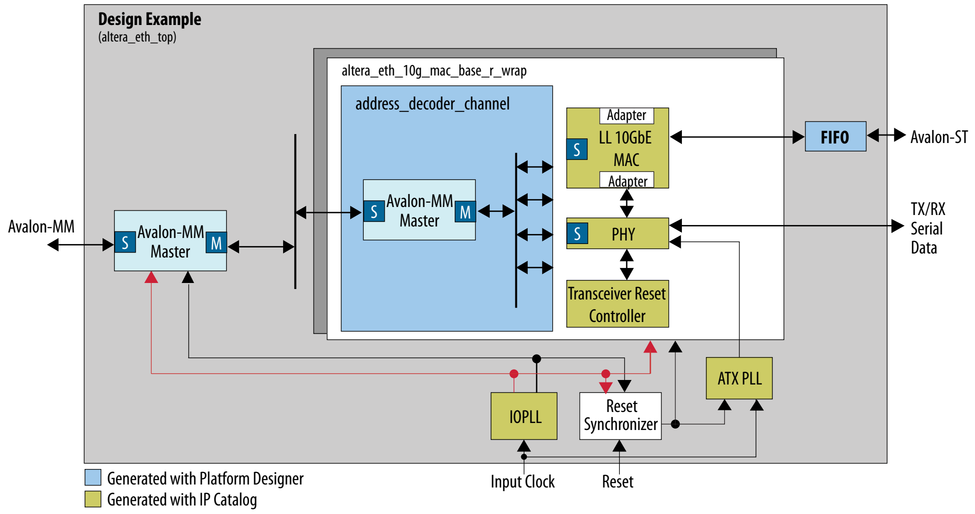
2.2. Hardware and Software Requirements

Intel uses the following hardware and software to test the design example in a Linux system:

- Intel Quartus Prime Pro Edition software
- ModelSim-AE, ModelSim-SE, NCSim (Verilog only), VCS, and Xcelium simulators
- For hardware testing:
  - Intel Cyclone 10 GX FPGA Development Kit (10CX220YF780E5G)
  - FMC loopback card

2.3. Functional Description

Figure 6. Block Diagram—10GBASE-R Ethernet Design Example



2.3.1. Design Components

Table 4. Design Components

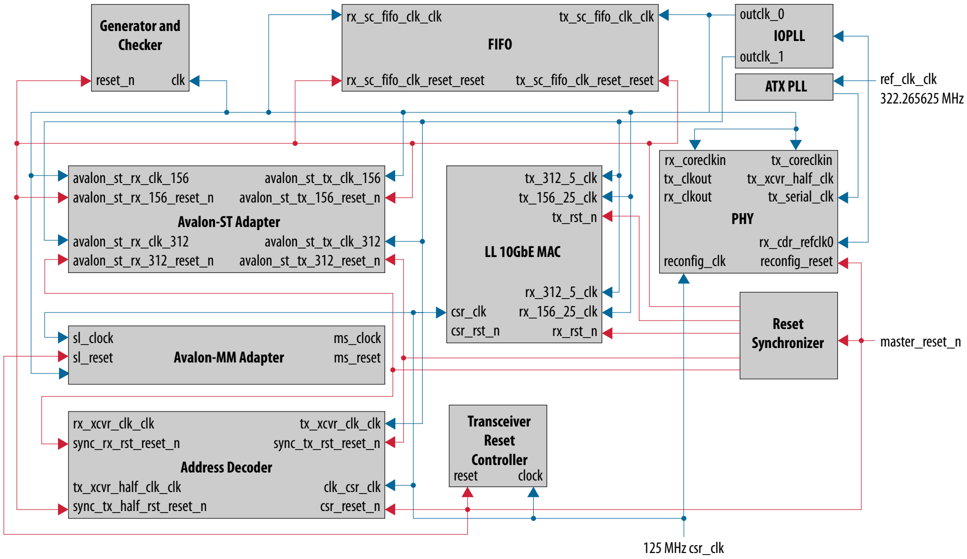
Component	Description
LL 10GbE MAC	The Low Latency Ethernet 10G MAC Intel FPGA IP core with the following configuration: • **Speed**: 10G • **Datapath options**: TX & RX • **Enable ECC on memory blocks**: Not selected • **Enable 10GBASE-R register mode**: Not selected • **Enable supplementary address**: Selected • **Enable statistics collection**: Selected • **Statistics counters**: Memory-based • **TX and RX datapath Reset/Default To Enable**: Selected • **Use legacy XGMII Interface**: Selected. • **Use legacy Avalon Memory-Mapped Interface**: Not Selected • **Use legacy Avalon Streaming Interface**: Not selected
PHY	• The Transceiver Native PHY Intel Arria 10/Intel Cyclone 10 FPGA IP configured for the 10GBASE-R protocol. • The preset sets the PHY's **TX FIFO MODE** to Phase Compensation and **RX FIFO MODE** to 10GBASE-R.
Transceiver Reset Controller	The Transceiver PHY Reset Controller Intel FPGA IP core. Resets the transceiver.
Address decoder	Decodes the addresses of the components.
Reset synchronizer	Synchronizes the reset of all design components.
IOPLL	Generates 312.5 MHz and 156.25 MHz clocks tothe MAC IP core, reset synchronizer, ethernet traffic controller, address decoder, and FIFO.
ATX PLL	Generates a TX serial clock for the Intel Cyclone 10 GX 10G transceiver.
FIFO	• Avalon® Streaming (Avalon-ST) single-clock FIFO. • Buffers the RX and TX data between the MAC IP core and the client.

Related Information

Low Latency Ethernet 10G MAC Intel FPGA IP User Guide
For more information on the MAC parameters.

2.3.2. Clocking and Reset Scheme

Figure 7. Clocking and Reset Scheme for 10GBASE-R Design Example



2.4. Simulation

The simulation test case demonstrates how the MAC and PHY configuration is changed at 10-Gbps throughput. The test case is for a single Ethernet channel.

At the end of the simulation, the simulator generates the statistics of TX and RX packets in the **Transcript** window.

In the **Wave** window, the roundtrip latency for the serial loopback is indicated by the measurement cursors that show the time taken to transmit the first data from the Avalon-ST TX interface to be available at the Avalon-ST RX interface.

Related Information

Compiling and Simulating the Design on page 8
Provides information on the procedure and testbench.

2.5. Hardware Testing

Follow the procedure at the provided link to test the design example in the selected hardware.

In the Clock Controller application, which is part of the development kit, set the following frequencies:

- Y2—322.265625 MHz
- U10, OUT1—125 MHz

Related Information

- Compiling and Testing the Design in Hardware on page 10
  More information on the procedure and hardware setup.
- Clock Controller
  More information on using the Clock Controller application.

2.5.1. Test Cases

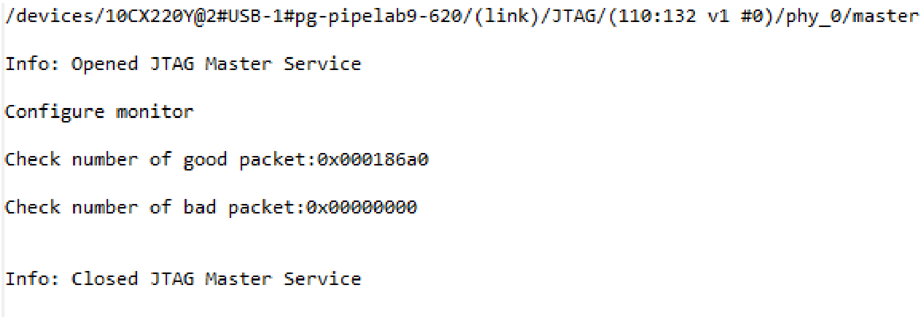
You can run any of the following tests from the System Console.

Table 5. Hardware Test Cases

Test Case	Command	Description
FMC loopback	`source gen_conf.tcl`	The generator generates and sends about 100 000 packets. Wait 30 seconds for it to complete its tasks.
	`source monitor_conf.tcl`	`source show_stats.tcl`
	`source show_stats.tcl`	This script displays the values of the statistics counters.
Avalon-ST loopback	`source loopback_conf.tcl`	This command enables the Avalon-ST loopback. This test is used with an external tester such as Spirent tester.

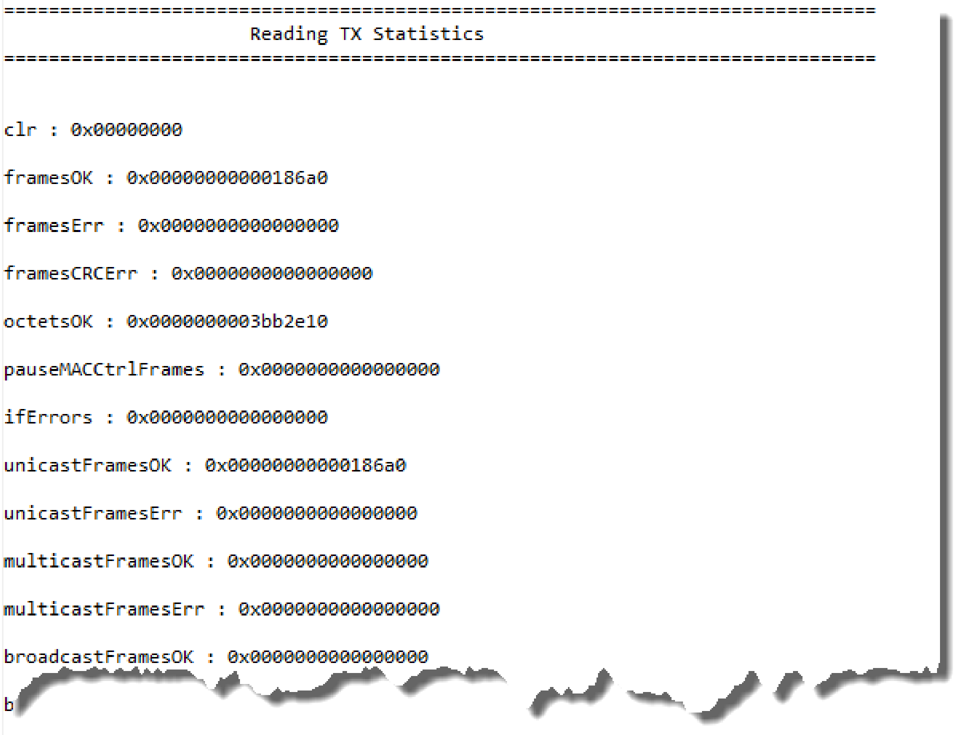
After the test is completed, observe the output displayed in the System Console.

Figure 8. Sample Test Output—Ethernet Packet Monitor



```
/devices/10CX220Y@2#USB-1#pg-pipelab9-620/(link)/JTAG/(110:132 v1 #0)/phy_0/master

Info: Opened JTAG Master Service

Configure monitor

Check number of good packet:0x000186a0

Check number of bad packet:0x00000000


Info: Closed JTAG Master Service
```

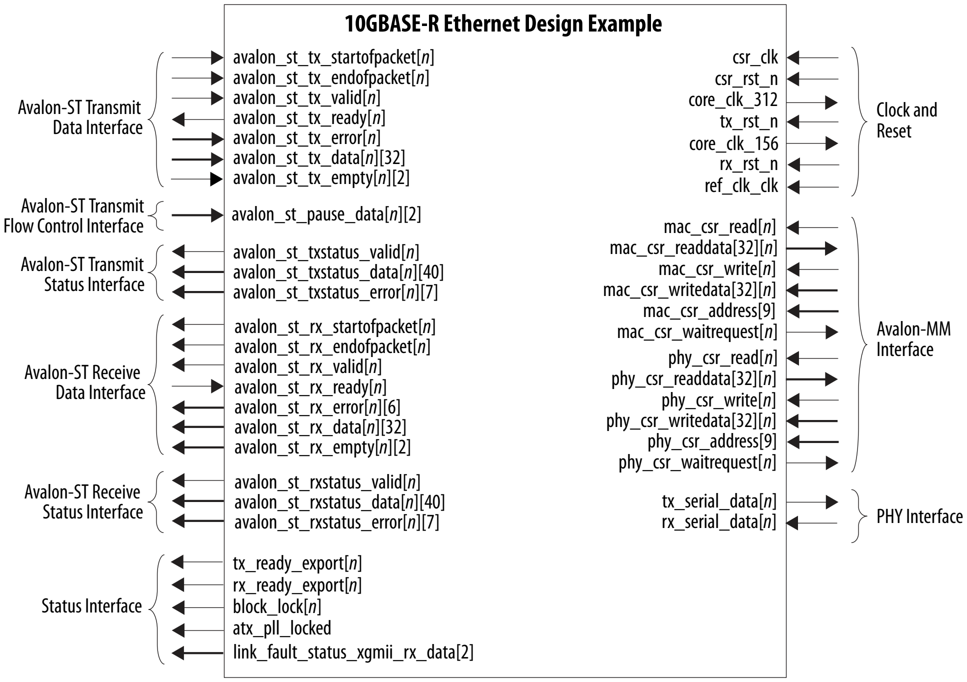
Figure 9. Sample Test Output—Statistics Counters



```
===============================================================================
                      Reading TX Statistics
===============================================================================


clr : 0x00000000

framesOK : 0x00000000000186a0

framesErr : 0x0000000000000000

framesCRCErr : 0x0000000000000000

octetsOK : 0x0000000003bb2e10

pauseMACCtrlFrames : 0x0000000000000000

ifErrors : 0x0000000000000000

unicastFramesOK : 0x00000000000186a0

unicastFramesErr : 0x0000000000000000

multicastFramesOK : 0x0000000000000000

multicastFramesErr : 0x0000000000000000

broadcastFramesOK : 0x0000000000000000

b
```

2.6. Interface Signals

Figure 10. Interface Signals of the 10GBASE-R Ethernet Design Example



n : Number of channels

Related Information

Interface Signals Description on page 29
For more information on each interface signal.

2.7. Configuration Registers

You can access the 32-bit configuration registers of the design components through the Avalon-MM interface.

Table 6. Register Map

Byte Offset	Block
0x0000_0000 – 0x0001_CFFF	Reserved
0x0001_D000 – 0xFFFF_FFFF	Client Logic
Channel 0	
0x0000_0000	MAC
0x0000_8000	PHY
0x0000_d400	RX SC FIFO
0x0000_d600	TX SC FIFO
0x0000_c000	Packet Generator and Checker

continued...

Byte Offset	Block
Channel 1	
0x0001_0000	MAC
0x0001_8000	PHY
0x0001_d400	RX SC FIFO
0x0001_d600	TX SC FIFO
0x0001_c000	Packet Generator and Checker

Related Information

Configuration Registers Description on page 35
For more information on each configuration register.

3. 10M/100M/1G/2.5G/5G/10G (USXGMII) Ethernet Design Example for Intel Cyclone 10 GX Devices

The 10G USXGMII Ethernet design example demonstrates the functionalities of the LL 10GbE MAC Intel FPGA IP core operating at 10M, 100M, 1G, 2.5G, 5G, and 10G.

Generate the design example from the **Example Design** tab of the LL 10GbE Intel FPGA IP parameter editor.

3.1. Features

- Supports dual Ethernet channel operating at 10M, 100M, 1G, 2.5G, 5G, and 10G.
- On the transmit and receive paths:
  - Provides packet monitoring system.
  - Reports Ethernet MAC statistics counter.
- Supports testing using different types of Ethernet packet transfer protocol.

3.2. Hardware and Software Requirements

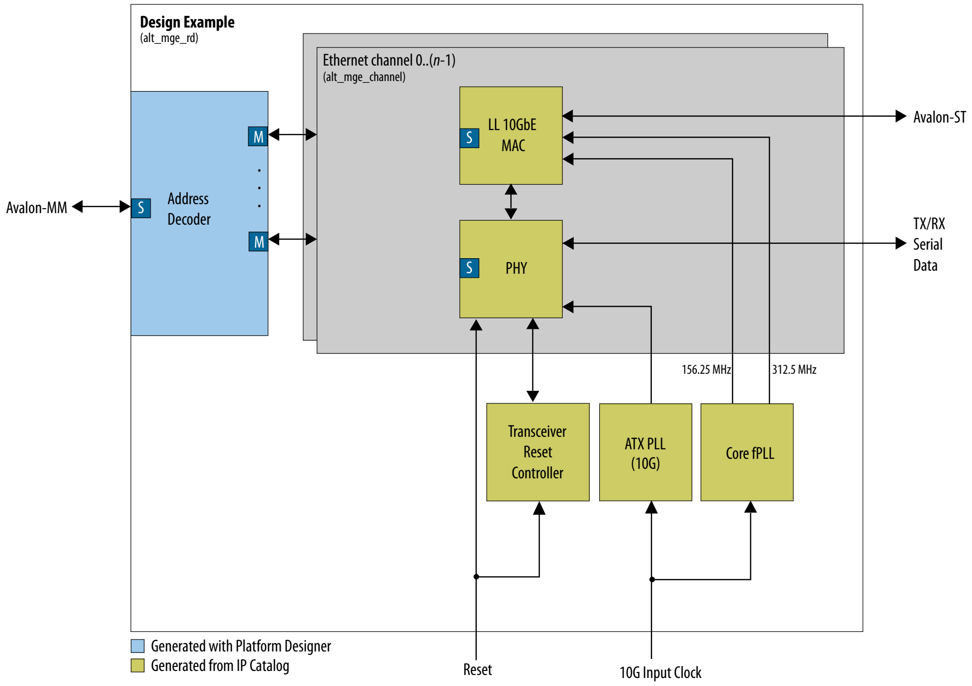
Intel uses the following hardware and software to test the design example in a Linux system:

- Intel Quartus Prime Pro Edition software
- ModelSim-AE, ModelSim-SE, NCSim (Verilog only), VCS, and Xcelium simulators
- For hardware testing:
  - Intel Cyclone 10 GX FPGA Development Kit (10CX220YF780E5G)
  - FMC loopback card

3.3. Functional Description

The design example consists of various components. The following block diagram shows the design components and the top-level signals of the design example.

Figure 11. Block Diagram—10M/100M/1G/2.5G/5G/10G (USXGMII) Ethernet Design Example



3.3.1. Design Components

Table 7. Design Components

Component	Description
LL 10GbE MAC	The Low Latency Ethernet 10G MAC Intel FPGA IP core with the following configuration: • **Speed**: 10M/100M/1G/2.5G/5G/10G (USXGMII) • **Datapath options**: TX & RX • **Enable ECC on memory blocks**: Not selected • **Enable supplementary address**: Selected • **Enable statistics collection**: Selected • **Statistics counters**: Memory-based • **TX and RX datapath Reset/Default To Enable**: Selected • **Use legacy XGMII Interface**: Not selected • **Use legacy Avalon Memory-Mapped Interface**: Not selected • **Use legacy Avalon Streaming Interface**: Selected
PHY	The 1G/2.5G/5G/10G Multi-rate Ethernet PHY Intel FPGA IP with the following configuration: • **Speed**: 10M/100M/1G/2.5G/5G/10G • **Connect to MGBASE-T PHY**: Not selected • **Connect to NBASE-T PHY**: Selected • **Reference clock frequency for 10GbE (MHz)**: 644.53125 • **Enable Altera Debug Master Endpoint**: Not selected • **Enable capability registers**: Not selected • **Enable control and status registers**: Not selected • **Enable PRBS soft accumulators**: Not selected

continued...

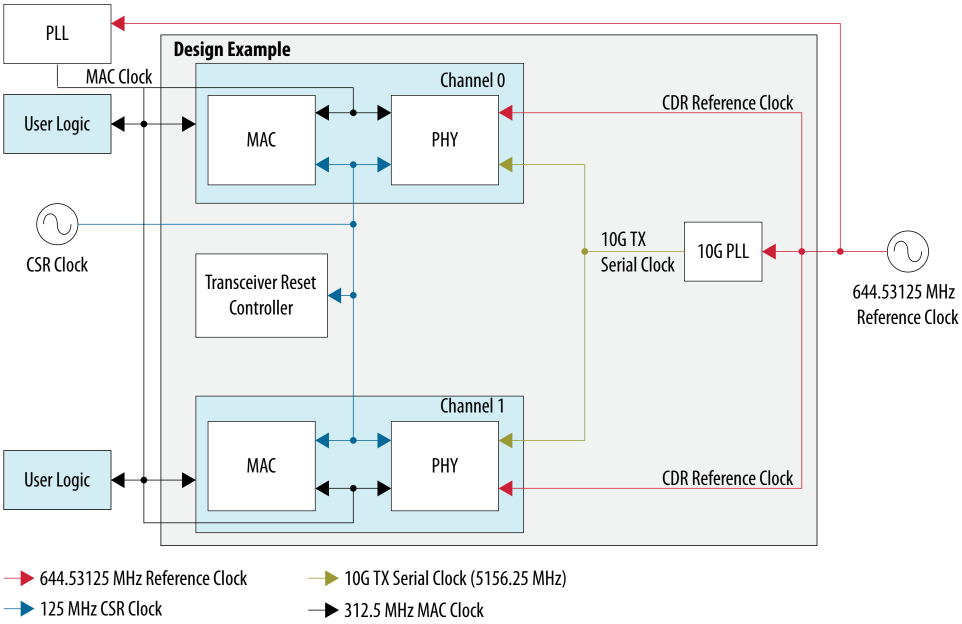
Component	Description
Channel address decoder	Decodes the addresses of the components in each Ethernet channel, such as PHY and LL 10GbE MAC.
Multi-channel address decoder	Decodes the addresses of the components used by all channels.
Top address decoder	Decodes the addresses of the top-level components, such as the Traffic Controller.
Transceiver Reset Controller	The Transceiver PHY Reset Controller Intel FPGA IP core. Resets the transceiver.
ATX PLL	Generates a TX serial clock for the Intel Cyclone 10 GX transceiver.
Core fPLL	Generates clocks for all design components.

Related Information

Low Latency Ethernet 10G MAC Intel FPGA IP User Guide
For more information on the MAC parameters.

3.3.2. Clocking Scheme

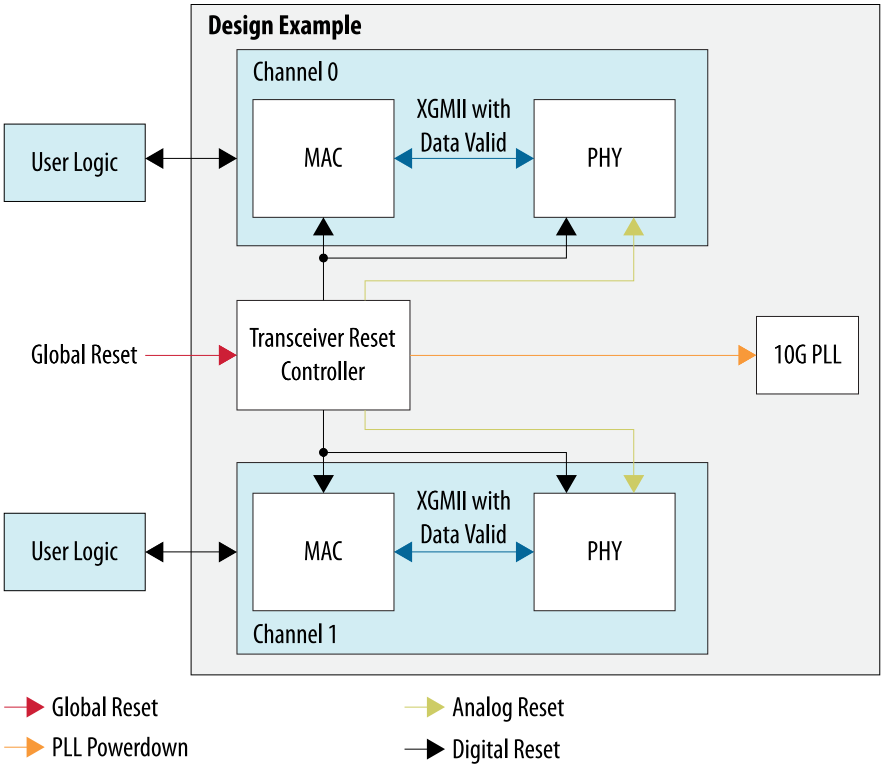
Figure 12. Clocking Scheme for 10M/100M/1G/2.5G/5G/10G (USXGMII) Ethernet Design Example



3.3.3. Reset Scheme

The global reset signal of the design example is asynchronous and active-high. Asserting this signal resets all channels and their components. Upon power-up, reset the design example.

Figure 13. Reset Scheme for 10M/100M/1G/2.5G/5G/10G (USXGMII) Ethernet Design Example



3.4. Simulation

Related Information

Compiling and Simulating the Design on page 8
Provides information on the procedure and testbench.

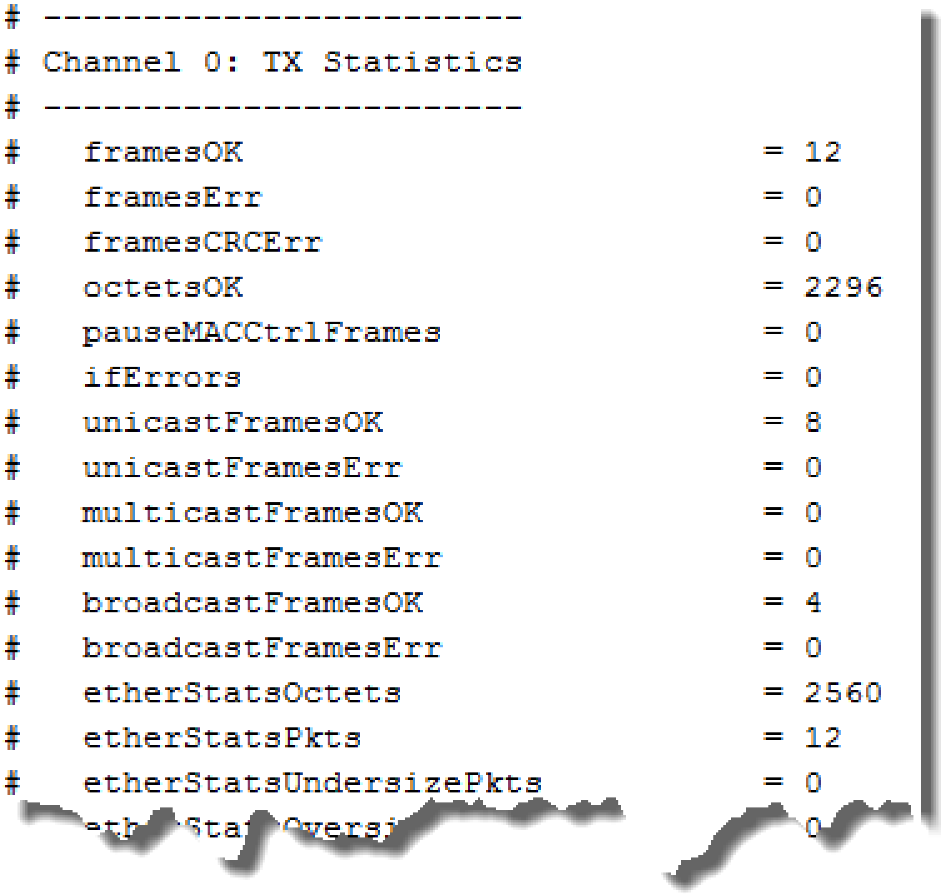
3.4.1. Test Case

The simulation test case performs the following steps:

1. Starts up the example design with an operating speed of 10G.
2. Configures the MAC, PHY, and FIFO buffer for both channels.
3. Waits until the design example asserts the `channel_tx_ready` and `channel_rx_ready`signals for both channels.
4. Sends the following packets:
   - 64-byte packet
   - 1518-byte packet
   - 100-byte packet
5. Repeats steps 2 to 4 for 10M, 100M, 1G, 2.5G, and 5G.

When simulation ends, the values of the MAC statistics counters are displayed in the transcript window. The transcript window also displays PASSED if the RX Avalon-ST interface of channel 0 received all packets successfully, all statistics error counters are zero, and the RX MAC statistics counters are equal to the TX MAC statistics counters.

Figure 14. Sample Simulation Output



```
# -------------------------
# Channel 0: TX Statistics
# -------------------------
#   framesOK                        = 12
#   framesErr                       = 0
#   framesCRCErr                    = 0
#   octetsOK                        = 2296
#   pauseMACCtrlFrames              = 0
#   ifErrors                        = 0
#   unicastFramesOK                 = 8
#   unicastFramesErr                = 0
#   multicastFramesOK               = 0
#   multicastFramesErr              = 0
#   broadcastFramesOK               = 4
#   broadcastFramesErr              = 0
#   etherStatsOctets                = 2560
#   etherStatsPkts                  = 12
#   etherStatsUndersizePkts         = 0
    eth  Stat  Oversi                 0
```

3.5. Hardware Testing

Follow the procedure at the provided link to test the design example in the selected hardware.

In the Clock Control application, which is part of the development kit, set the following frequencies:

- Y2—644.53125 MHz
- U10, OUT1—125 MHz

Related Information

- Compiling and Testing the Design in Hardware on page 10
  More information on the procedure and hardware setup.

- Clock Controller

  More information on using the Clock Controller application.

3.5.1. Test Procedure

Follow these steps to test the design examples in hardware:

1. Run the following command in the system console to start the test.

   TEST_EXT_LB <*channel*> <*speed*> <*burst_size*>

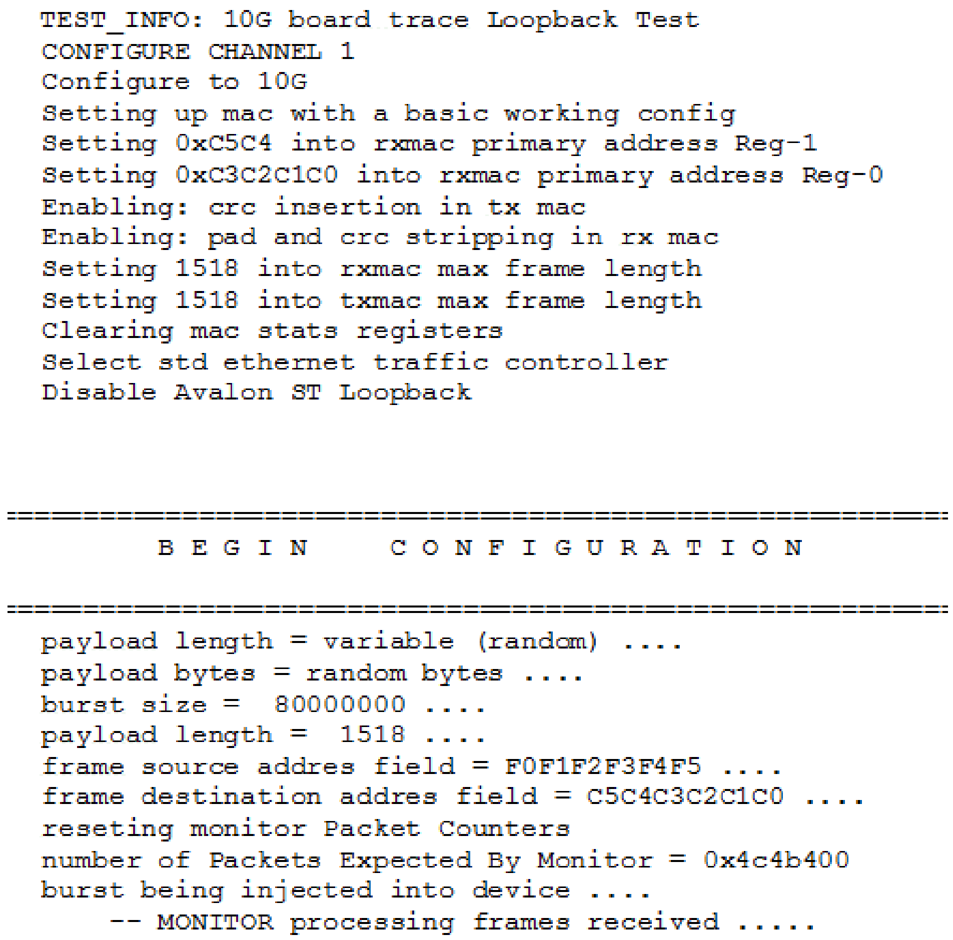
   Example: `TEST_EXT_LB 0 10G 80000000`

Table 8. Command Parameters

Parameter	Valid Values	Description
`channel`	0, 1	The channel number to test.
`speed`	10M, 100M, 1G, 2P5G, 5G, 10G	The PHY speed.
`burst_size`	An integer value	The number of packets to generate for the test.

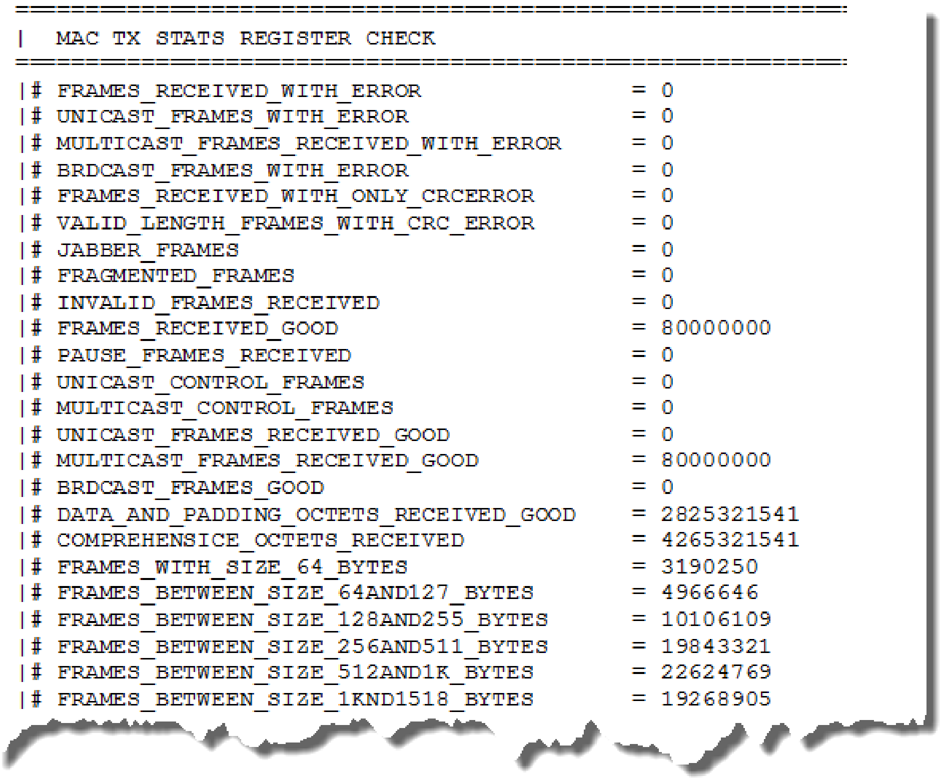
2. When the test is completed, observe the output displayed. The following diagrams show excerpts of the output, which shows that the Ethernet packet monitor block receives the same number of packets generated without error, and the TX and RX statistics counters.

Figure 15. Sample Test Output—Ethernet Packet Monitor



```
  TEST_INFO: 10G board trace Loopback Test
  CONFIGURE CHANNEL 1
  Configure to 10G
  Setting up mac with a basic working config
  Setting 0xC5C4 into rxmac primary address Reg-1
  Setting 0xC3C2C1C0 into rxmac primary address Reg-0
  Enabling: crc insertion in tx mac
  Enabling: pad and crc stripping in rx mac
  Setting 1518 into rxmac max frame length
  Setting 1518 into txmac max frame length
  Clearing mac stats registers
  Select std ethernet traffic controller
  Disable Avalon ST Loopback



=========================================================
         B E G I N    C O N F I G U R A T I O N

=========================================================
  payload length = variable (random) ....
  payload bytes = random bytes ....
  burst size =  80000000 ....
  payload length =  1518 ....
  frame source addres field = F0F1F2F3F4F5 ....
  frame destination addres field = C5C4C3C2C1C0 ....
  reseting monitor Packet Counters
  number of Packets Expected By Monitor = 0x4c4b400
  burst being injected into device ....
      -- MONITOR processing frames received .....
```

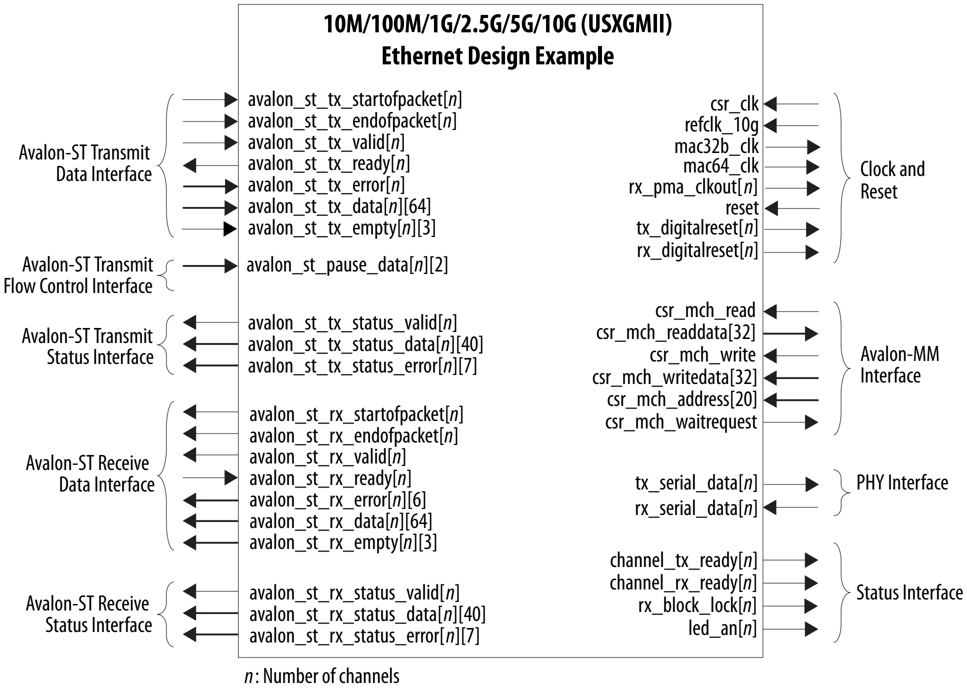
Figure 16. Sample Test Output—Statistics Counters



```
===========================================================
|  MAC TX STATS REGISTER CHECK
===========================================================
|# FRAMES_RECEIVED_WITH_ERROR                = 0
|# UNICAST_FRAMES_WITH_ERROR                 = 0
|# MULTICAST_FRAMES_RECEIVED_WITH_ERROR      = 0
|# BRDCAST_FRAMES_WITH_ERROR                 = 0
|# FRAMES_RECEIVED_WITH_ONLY_CRCERROR        = 0
|# VALID_LENGTH_FRAMES_WITH_CRC_ERROR        = 0
|# JABBER_FRAMES                             = 0
|# FRAGMENTED_FRAMES                         = 0
|# INVALID_FRAMES_RECEIVED                   = 0
|# FRAMES_RECEIVED_GOOD                      = 80000000
|# PAUSE_FRAMES_RECEIVED                     = 0
|# UNICAST_CONTROL_FRAMES                    = 0
|# MULTICAST_CONTROL_FRAMES                  = 0
|# UNICAST_FRAMES_RECEIVED_GOOD              = 0
|# MULTICAST_FRAMES_RECEIVED_GOOD            = 80000000
|# BRDCAST_FRAMES_GOOD                       = 0
|# DATA_AND_PADDING_OCTETS_RECEIVED_GOOD     = 2825321541
|# COMPREHENSICE_OCTETS_RECEIVED             = 4265321541
|# FRAMES_WITH_SIZE_64_BYTES                 = 3190250
|# FRAMES_BETWEEN_SIZE_64AND127_BYTES        = 4966646
|# FRAMES_BETWEEN_SIZE_128AND255_BYTES       = 10106109
|# FRAMES_BETWEEN_SIZE_256AND511_BYTES       = 19843321
|# FRAMES_BETWEEN_SIZE_512AND1K_BYTES        = 22624769
|# FRAMES_BETWEEN_SIZE_1KND1518_BYTES        = 19268905
```

3.6. Interface Signals

Figure 17. Interface Signals of the 10M/100M/1G/2.5G/5G/10G (USXGMII) Ethernet Design Example



3.7. Configuration Registers

You can access the 32-bit configuration registers of the design components through the Avalon-MM interface.

Table 9. Register Map

Byte Offset	Block
0x00_0000	Reserved
Channel 0	
0x02_0000	Reserved
0x02_4000	PHY
0x02_6000	Native PHY Reconfiguration
0x02_8000	MAC
Channel 1	
0x03_0000	Reserved
0x03_4000	PHY
0x03_6000	Native PHY Reconfiguration

continued...

Byte Offset	Block
0x03_8000	MAC
Traffic Controller	
0x10_0000	Traffic Controller

4. Interface Signals Description

Use the following tables to find the description of the signals in the LL 10GbE MAC Intel FPGA IP design examples. The pinout diagram for each design example specifies the width of the signals.

4.1. Clock and Reset Interface Signals

Table 10. Clock and Reset Interface Signals

Signal	Direction	Width	Description
csr_clk	In	1	125 MHz configuration clock for the Avalon-MM interface.
ref_clk_clk	In	1	322.265625 MHz clock for the TX PLL.
core_clk_312	Out	1	312.5 MHz clock for the fast domain.
core_clk_156	Out	1	156.25-MHz clock for the slow domain.
master_reset_n	In	1	Assert this asynchronous and active-low signal to reset the whole design example.
csr_rst_n	In	1	Active-low reset signal for the Avalon-MM interface.
tx_rst_n	In	1	Active-low reset signal for the TX datapath.
rx_rst_n	In	1	Active-low reset signal for the RX datapath.
tx_digitalreset	In	[NUM_CHANNELS]	Asynchronous and active-high signal to reset PCS TX portion of the transceiver PHY.
rx_digitalreset	In	[NUM_CHANNELS]	Asynchronous and active-high signal to reset PCS RX portion of the transceiver PHY.
tx_analogreset	In	[NUM_CHANNELS]	Asynchronous and active-high signal to reset PMA TX portion of the transceiver PHY.
rx_analogreset	In	[NUM_CHANNELS]	Asynchronous and active-high signal to reset PMA RX portion of the transceiver PHY.

4.2. Avalon-MM Interface Signals

Table 11. Avalon-MM Interface Signals

Signal	Direction	Description
write csr_mac_write csr_phy_write csr_mch_write	In	Assert this signal to request a write.
read csr_mac_read csr_phy_read csr_mch_read	In	Assert this signal to request a read.
address csr_mac_address csr_phy_address csr_mch_address	In	Use this bus to specify the register address you want to read from or write to.
writedata csr_mac_writedata csr_phy_writedata csr_mch_writedata	In	Carries the data to be written to the specified register.
readdata csr_mac_readdata csr_phy_readdata csr_mch_readdata	Out	Carries the data read from the specified register.
waitrequest csr_mac_waitrequest csr_phy_waitrequest csr_mch_waitrequest	Out	When asserted, this signal indicates that the IP core is busy and not ready to accept any read or write requests.

4.3. Avalon-ST Interface Signals

Table 12. Avalon-ST Interface Signals

Signal	Direction	Width	Description
avalon_st_tx_startofpacket[]	In	[NUM_CHANNELS]	Assert this signal to indicate the beginning of the TX data.
avalon_st_tx_endofpacket[]	In	[NUM_CHANNELS]	Assert this signal to indicate the end of the TX data.
avalon_st_tx_valid[]	In	[NUM_CHANNELS]	Assert this signal to indicate that avalon_st_tx_data[] and other signals on this interface are valid.
avalon_st_tx_ready[]	Out	[NUM_CHANNELS]	When asserted, indicates that the MAC IP core is ready to accept data. The reset value of this signal is non-deterministic.
avalon_st_tx_error[]	In	[NUM_CHANNELS]	Assert this signal to indicate that the current TX packet contains errors.
avalon_st_tx_data[][]	In	[NUM_CHANNELS][*m*]	TX data from the client.

continued...

Signal	Direction	Width	Description
			m is 64 when the **Use legacy Avalon Streaming Interface** parameter is selected. Otherwise, *m* is 32.
avalon_st_tx_empty[][]	In	[NUM_CHANNELS][*m*]	Use this signal to specify the number of empty bytes in the cycle that contain the end of the TX data. *m* is 3 when the **Use legacy Avalon Streaming Interface** parameter is selected. Otherwise, *m* is 2. • 0x0—All bytes are valid. • 0x1—The last byte is invalid. • 0x2—The last two bytes are invalid. • 0x3—The last three bytes are invalid.
avalon_st_rx_startofpacket[]	Out	[NUM_CHANNELS]	When asserted, indicates the beginning of the RX data.
avalon_st_rx_endofpacket[]	Out	[NUM_CHANNELS]	When asserted, indicates the end of the RX data.
avalon_st_rx_valid[]	Out	[NUM_CHANNELS]	When asserted, indicates that the avalon_st_rx_data[] signal and other signals on this interface are valid.
avalon_st_rx_ready[]	In	[NUM_CHANNELS]	Assert this signal when the client is ready to accept data.
avalon_st_rx_error[][]	Out	[NUM_CHANNELS][6]	When set to 1, the respective bits indicate an error type: • Bit 0—PHY error. For 10 Gbps, the data on xgmii_rx_data contains a control error character (FE). For 10 Mbps,100 Mbps,1 Gbps, gmii_rx_err or mii_rx_err is asserted. • Bit 1—CRC error. The computed CRC value differs from the received CRC. • Bit 2—Undersized frame. The receive frame length is less than 64 bytes. • Bit 3—Oversized frame. The receive frame length is more than MAX_FRAME_SIZE. • Bit 4—Payload length error. The actual frame payload length is different from the value in the length/type field. • Bit 5—Overflow error. The receive FIFO buffer is full while it is still receiving data from the MAC IP core.
avalon_st_rx_data[][]	Out	[NUM_CHANNELS][*m*]	RX data to the client. *m* is 64 when the **Use legacy Avalon Streaming Interface** parameter is selected. Otherwise, *m* is 32.
avalon_st_rx_empty[][]	Out	[NUM_CHANNELS][*m*]	Contains the number of empty bytes during the cycle that contain the end of the RX data. *m* is 3 when the **Use legacy Avalon Streaming Interface** parameter is selected. Otherwise, *m* is 2.
avalon_st_tx_status_valid[]	Out	[NUM_CHANNELS]	When asserted, this signal qualifies the avalon_st_txstatus_data[] and avalon_st_txstatus_error[] signals.
avalon_st_tx_status_data[][]	Out	[NUM_CHANNELS][40]	Contains information about the TX frame.

continued...

Signal	Direction	Width	Description
			• Bits 0 to 15—Payload length. • Bits 16 to 31—Packet length. • Bit 32—When set to 1, indicates a stacked VLAN frame. • Bit 33—When set to 1, indicates a VLAN frame. • Bit 34—When set to 1, indicates a control frame. • Bit 35—When set to 1, indicates a pause frame. • Bit 36—When set to 1, indicates a broadcast frame. • Bit 37—When set to 1, indicates a multicast frame. • Bit 38—When set to 1, indicates a unicast frame. • Bit 39—When set to 1, indicates a PFC frame.
`avalon_st_tx_status_error[][]`	Out	`[NUM_CHANNELS][7]`	When set to 1, the respective bit indicates the following error type in the RX frame. • Bit 0—Undersized frame. • Bit 1—Oversized frame. • Bit 2—Payload length error. • Bit 3—Unused. • Bit 4—Underflow. • Bit 5—Client error. • Bit 6—Unused. The error status is invalid when an overflow occurs.
`avalon_st_rxstatus_valid[]`	Out	`[NUM_CHANNELS]`	When asserted, this signal qualifies the `avalon_st_rxstatus_data[]` and `avalon_st_rxstatus_error[]` signals. The MAC IP core asserts this signal in the same clock cycle the `avalon_st_rx_endofpacket` signal is asserted.

continued...

Signal	Direction	Width	Description
avalon_st_rxstatus_data[][]	Out	[NUM_CHANNELS][40]	Contains information about the RX frame. • Bits 0 to 15—Payload length. • Bits 16 to 31—Packet length. • Bit 32—When set to 1, indicates a stacked VLAN frame. • Bit 33—When set to 1, indicates a VLAN frame. • Bit 34—When set to 1, indicates a control frame. • Bit 35—When set to 1, indicates a pause frame. • Bit 36—When set to 1, indicates a broadcast frame. • Bit 37—When set to 1, indicates a multicast frame. • Bit 38—When set to 1, indicates a unicast frame. • Bit 39—When set to 1, indicates a PFC frame.
avalon_st_rxstatus_error[][]	Out	[NUM_CHANNELS][7]	When set to 1, the respective bit indicates the following error type in the RX frame. • Bit 0—Undersized frame. • Bit 1—Oversized frame. • Bit 2—Payload length error. • Bit 3—Unused. • Bit 4—Underflow. • Bit 5—Client error. • Bit 6—Unused. The error status is invalid when an overflow occurs.
avalon_st_pause_data[][]	In	[NUM_CHANNELS][2]	This signal takes effect when the register bits, tx_pauseframe_enable[2:1], are both set to the default value 0. Set this signal to the following values to trigger the corresponding actions. • 0x0—Stops pause frame generation. • 0x1—Generates an XON pause frame. • 0x2—Generates an XOFF pause frame. The MAC IP core sets the pause quanta field in the pause frame to the value in the tx_pauseframe_quanta register. • 0x3—Reserved.

Related Information

Avalon-ST Data Interface Clocks, Low Latency Ethernet 10G MAC Intel FPGA IP User Guide

4.4. PHY Interface Signals

Table 13. PHY Interface Signals

Signal	Direction	Description
rx_serial_data	In	RX serial input data
tx_serial_data	Out	TX serial output data

4.5. Status Interface

Table 14. Status Interface Signals

Signal	Direction	Description
`block_lock`	Out	Asserted when the link synchronization is successful.

`channel_ready`
`channel_tx_ready`
`channel_rx_ready`
`tx_ready_export`
`rx_ready_export`
	Out	Asserted when the channel is ready for data transmission.

`atx_pll_locked`	Out	Asserted when the TX PLL is locked.

5. Configuration Registers Description

5.1. Register Access Definition

Table 15. Types of Register Access

Access	Definition
RO	Read only.
RW	Read and write.
RWC	Read, and write and clear. The user application writes 1 to the register bit(s) to invoke a defined instruction. The IP core clears the bit(s) upon executing the instruction.

5.2. Low Latency Ethernet 10G MAC

This topic lists the byte offsets the MAC registers.

Table 16. Primary MAC Address

Byte Offset	R/W	Name	HW Reset
0x2008	RW	primary_mac_addr0	0x0
0x200C	RW	primary_mac_addr1	0x0

Table 17. TX Configuration and Status Registers

Byte Offset	R/W	Name	HW Reset
0x4000	RW	tx_packet_control	0x0
0x4004	RO	tx_packet_status	0x0
0x4100	RW	tx_pad_control	0x1
0x4200	RW	tx_crc_control	0x3
0x4400	RW	tx_preamble_control	0x0
0x6004	RW	tx_frame_maxlength	0x5EE(1518)
0x4300	RO	tx_underflow_counter0	0x0
0x4304	RO	tx_underflow_counter1	0x0

Table 18. Flow Control Registers

Byte Offset	R/W	Name	HW Reset
0x4500	RW	tx_pauseframe_control	0x0
0x4504	RW	tx_pauseframe_quanta	0x0
0x4508	RW	tx_pauseframe_enable	0x1
0x4680	RW	tx_pfc_priority_enable	0x0
0x4600	RW	pfc_pause_quanta_0	0x0
0x4604	RW	pfc_pause_quanta_1	0x0
0x4608	RW	pfc_pause_quanta_2	0x0
0x460C	RW	pfc_pause_quanta_3	0x0
0x4610	RW	pfc_pause_quanta_4	0x0
0x4614	RW	pfc_pause_quanta_5	0x0
0x4618	RW	pfc_pause_quanta_6	0x0
0x461C	RW	pfc_pause_quanta_7	0x0
0x4640	RW	pfc_holdoff_quanta_0	0x1
0x4644	RW	pfc_holdoff_quanta_1	0x1
0x4648	RW	pfc_holdoff_quanta_2	0x1
0x464C	RW	pfc_holdoff_quanta_3	0x1
0x4650	RW	pfc_holdoff_quanta_4	0x1
0x4654	RW	pfc_holdoff_quanta_5	0x1
0x4658	RW	pfc_holdoff_quanta_6	0x1
0x465C	RW	pfc_holdoff_quanta_7	0x1

Table 19. RX Configuration and Status Registers

Byte Offset	R/W	Name	HW Reset
0x0000	RW	rx_transfer_control	0x0
0x0004	RO	rx_transfer_status	0x0
0x0100	RW	rx_padcrc_control	0x1
0x0200	RW	rx_crccheck_control	0x2
0x0400	RW	rx_custom_preamble_forward	0x0
0x0500	RW	rx_preamble_control	0x0
0x2000	RW	rx_frame_control	0x3
0x2004	RW	rx_frame_maxlength	1518
0x2010	RW	rx_frame_spaddr0_0	0x0
0x2014	RW	rx_frame_spaddr0_1	0x0
0x2018	RW	rx_frame_spaddr1_0	0x0

continued...

Byte Offset	R/W	Name	HW Reset
0x201C	RW	rx_frame_spaddr1_1	0x0
0x2020	RW	rx_frame_spaddr2_0	0x0
0x2024	RW	rx_frame_spaddr2_1	0x0
0x2028	RW	rx_frame_spaddr3_0	0x0
0x202C	RW	rx_frame_spaddr3_1	0x0
0x2060	RW	rx_pfc_control	0x1
0x0300	RO	rx_pktovrflow_error	0x0

Table 20. TX and RX Statistics Registers

Byte Offset	R/W	Name	HW Reset
0x7000	RO	tx_stats_clr	0x0
0x3000	RO	rx_stats_clr	0x0
0x7008:0x700C	RO	tx_stats_framesOK	0x0
0x3008:0x300C	RO	rx_stats_framesOK	0x0
0x7010:0x7014	RO	tx_stats_framesErr	0x0
0x3010:0x3014	RO	rx_stats_framesErr	0x0
0x7018:0x701C	RO	tx_stats_framesCRCErr	0x0
0x3018:0x301C	RO	rx_stats_framesCRCErr	0x0
0x7020:0x7024	RO	tx_stats_octetsOK	0x0
0x3020:0x3024	RO	rx_stats_octetsOK	0x0
0x7028:0x702C	RO	tx_stats_pauseMACCtrl_Frames	0x0
0x3028:0x302C	RO	rx_stats_pauseMACCtrl_Frames	0x0
0x7030:0x7034	RO	tx_stats_ifErrors	0x0
0x3030:0x3034	RO	rx_stats_ifErrors	0x0
0x7038:0x703C	RO	tx_stats_unicast_FramesOK	0x0
0x3038:0x303C	RO	rx_stats_unicast_FramesOK	0x0
0x7040:0x7044	RO	tx_stats_unicast_FramesErr	0x0
0x3040:0x3044	RO	rx_stats_unicast_FramesErr	0x0
0x7048:0x704C	RO	tx_stats_multicast_FramesOK	0x0
0x3048:0x304C	RO	rx_stats_multicast_FramesOK	0x0
0x7050:0x7054	RO	tx_stats_multicast_FramesErr	0x0
0x3050:0x3054	RO	rx_stats_multicast_FramesErr	0x0
0x7058:0x705C	RO	tx_stats_broadcast_FramesOK	0x0
0x3058:0x305C	RO	rx_stats_broadcast_FramesOK	0x0
0x7060:0x7064	RO	tx_stats_broadcast_FramesErr	0x0

continued...

Byte Offset	R/W	Name	HW Reset
0x3060:0x3064	RO	rx_stats_broadcast_FramesErr	0x0
0x7068:0x706C	RO	tx_stats_etherStatsOctets	0x0
0x3068:0x306C	RO	rx_stats_etherStatsOctets	0x0
0x7070:0x7074	RO	tx_stats_etherStatsPkts	0x0
0x3070:0x3074	RO	rx_stats_etherStatsPkts	0x0
0x7078:0x707C	RO	tx_stats_etherStatsUndersizePkts	0x0
0x3078:0x307C	RO	rx_stats_etherStatsUndersizePkts	0x0
0x7080:0x7084	RO	tx_stats_etherStatsOversizePkts	0x0
0x3080:0x3084	RO	rx_stats_etherStatsOversizePkts	0x0
0x7088:0x708C	RO	tx_stats_etherStatsPkts64Octets	0x0
0x3088:0x308C	RO	rx_stats_etherStatsPkts64Octets	0x0
0x7090:0x7094	RO	tx_stats_etherStatsPkts65to127Octets	0x0
0x3090:0x3094	RO	rx_stats_etherStatsPkts65to127Octets	0x0
0x7098:0x709C	RO	tx_stats_etherStatsPkts128to255Octets	0x0
0x3098:0x309C	RO	rx_stats_etherStatsPkts128to255Octets	0x0
0x70A0:0x70A4	RO	tx_stats_etherStatsPkts256to511Octets	0x0
0x30A0:0x30A4	RO	rx_stats_etherStatsPkts256to511Octets	0x0
0x70A8:0x70AC	RO	tx_stats_etherStatsPkts512to1023Octets	0x0
0x30A8:0x30AC	RO	rx_stats_etherStatsPkts512to1023Octets	0x0
0x70B0:0x70B4	RO	tx_stats_etherStatPkts1024to1518Octets	0x0
0x30B0:0x30B4	RO	rx_stats_etherStatPkts1024to1518Octets	0x0
0x70B8:0x70BC	RO	tx_stats_etherStatsPkts1519toXOctets	0x0
0x30B8:0x30BC	RO	rx_stats_etherStatsPkts1519toXOctets	0x0
0x70C0:0x70C4	RO	tx_stats_etherStatsFragments	0x0
0x30C0:0x30C4	RO	rx_stats_etherStatsFragments	0x0
0x70C8:0x70CC	RO	tx_stats_etherStatsJabbers	0x0
0x30C8:0x30CC	RO	rx_stats_etherStatsJabbers	0x0
0x70D0:0x70D4	RO	tx_stats_etherStatsCRCErr	0x0
0x30D0:0x30D4	RO	rx_stats_etherStatsCRCErr	0x0
0x70D8:0x70DC	RO	tx_stats_unicastMACCtrlFrames	0x0
0x30D8:0x30DC	RO	rx_stats_unicastMACCtrlFrames	0x0
0x70E0:0x70E4	RO	tx_stats_multicastMACCtrlFrames	0x0
0x30E0:0x30E4	RO	rx_stats_multicastMACCtrlFrames	0x0
0x70E8:0x70EC	RO	tx_stats_broadcastMACCtrlFrames	0x0

continued...

Byte Offset	R/W	Name	HW Reset
0x30E8:0x30EC	RO	rx_stats_broadcastMACCtrlFrames	0x0
0x70F0:0x70F4	RO	tx_stats_PFCMACCtrlFrames	0x0
0x30F0:0x30F4	RO	rx_stats_PFCMACCtrlFrames	0x0

5.3. PHY

5.3.1. 1G/2.5G/5G/10G Multi-rate PHY

This topic lists the byte offsets of the 1G/2.5G/5G/10G Multi-rate variant registers for Intel Cyclone 10 GX devices.

Register Map

You can access the 32-bit configuration registers via the Avalon-MM interface.

Table 21. Register Map Overview

Address Range	Usage	Register Width	Configuration
0x400 : 0x41F	USXGMII	32	1G/2.5G/5G/10G (USXGMII)
0x461	Serial Loopback	32	1G/2.5G/5G/10G (USXGMII)

Register Definitions

Observe the following guidelines when accessing the registers:

- Do not write to reserved or undefined registers.
- When writing to the registers, perform read-modify-write operation to ensure that reserved or undefined register bits are not overwritten.

Table 22. 1G/2.5G/5G/10G Multi-rate Register Definitions

Address	Name	Description	Access	HW Reset Value
0x400	usxgmii_control	Control Register	—	—
		Bit [0]: USXGMII_ENA: • 0: 10GBASE-R mode • 1: USXGMII mode	RW	0
		Bit [1]: USXGMII_AN_ENA is used when USXGMII_ENA is set to 1: • 0: Disables USXGMII Auto-Negotiation and manually configures the operating speed with the USXGMII_SPEED register. • 1: Enables USXGMII Auto-Negotiation, and automatically configures operating speed with link partner ability advertised during USXGMII Auto-Negotiation.	RW	1
		Bit [4:2]: USXGMII_SPEED is the operating speed of the PHY in USXGMII mode and USE_USXGMII_AN is set to 0.	RW	0

continued...

Address	Name	Description	Access	HW Reset Value
		• 3’b000: 10M • 3’b001: 100M • 3’b010: 1G • 3’b011: 10G • 3’b100: 2.5G • 3’b101: 5G • 3’b110: Reserved • 3’b111: Reserved		
		Bit [8:5]: Reserved	—	—
		Bit [9]: RESTART_AUTO_NEGOTIATION Write 1 to restart Auto-Negotiation sequence The bit is cleared by hardware when Auto-Negotiation is restarted.	RWC	0
		Bit [31:10]: Reserved	—	—
0x401	usxgmii_status	Status Register	—	—
		Bit [1:0]: Reserved	—	—
		Bit [2]: LINK_STATUS indicates link status for USXGMII all speeds • 1: Link is established • 0: Link synchronization is lost, a 0 is latched	RO	0
		Bit [4:3]: Reserved	—	—
		Bit [5]: AUTO_NEGOTIATION_COMPLETE A value of 1 indicates the Auto-Negotiation process is completed.	RO	0
		Bit [31:6]: Reserved	—	—
0x402:0x404	Reserved	—	—	—
0x405	usxgmii_partner_ability	Device abilities advertised to the link partner during Auto-Negotiation	—	—
		Bit [6:0]: Reserved	—	—
		Bit [7]: EEE_CLOCK_STOP_CAPABILITY Indicates whether or not energy efficient Ethernet (EEE) clock stop is supported. • 0: Not supported • 1: Supported	RO	0
		Bit [8]: EEE_CAPABILITY Indicates whether or not EEE is supported. • 0: Not supported • 1: Supported	RO	0
		Bit [11:9]: SPEED • 3'b000: 10M • 3'b001: 100M • 3'b010: 1G • 3'b011: 10G • 3'b100: 2.5G • 3'b101: 5G • 3'b110: Reserved • 3'b111: Reserved	RO	0

continued...

Address	Name	Description	Access	HW Reset Value
		Bit [12]: DUPLEX Indicates the duplex mode. • 0: Half duplex • 1: Full duplex	RO	0
		Bit [13]: Reserved	—	—
		Bit [14]: ACKNOWLEDGE A value of 1 indicates that the device has received three consecutive matching ability values from its link partner.	RO	0
		Bit [15]: LINK Indicates the link status. • 0: Link down • 1: Link up	RO	0
		Bit [31:16]: Reserved	—	—
0x406:0x411	Reserved	—	—	—
0x412	usxgmii_link_timer	Auto-Negotiation link timer. Sets the link timer value in bit [19:14] from 0 to 2 ms in approximately 0.05-ms steps. You must program the link timer to ensure that it matches the link timer value of the external NBASE-T PHY IP Core. The reset value sets the link timer to approximately 1.6 ms. Bits [13:0] are reserved and always set to 0.	[19:14]: RW [13:0]: RO	[19:14]: 1F [13:0]: 0
0x413:0x41F	Reserved	—	—	—
0x461	phy_serial_loopback	Configures the transceiver serial loopback in the PMA from TX to RX.	—	—
		Bit [0] • 0: Disables the PHY serial loopback • 1: Enables the PHY serial loopback	RW	0
		Bit [31:1]: Reserved	—	—

6. Low Latency Ethernet 10G MAC Intel Cyclone 10 GX FPGA IP Design Example User Guide Archives

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

IP Core Version	User Guide
18.0	Low Latency Ethernet 10G MAC Intel Cyclone 10 GX FPGA IP Design Example User Guide

7. Document Revision History for the Low Latency Ethernet 10G MAC Intel Cyclone 10 GX FPGA IP Design Example User Guide

Document Version	Intel Quartus Prime Version	Changes
2018.10.03	18.1	• Added *10M/100M/1G/2.5G/10G USXGMII Ethernet Design Example for Intel Cyclone 10 GX Devices* chapter. • Updated the *Quick Start Guide* chapter: — Updated *Hardware and Software Requirements* topic. — Updated Figures: *Example Design Tab* and *Block Diagram of the Hardware Setup*. — Updated Table: *Parameters in the Example Design Tab* to include a note to parameter **Enable ADME support** to clarify that this option is only available from Intel Quartus Prime Pro Edition version 17.1 onwards. • Updated the *10GBASE-R Ethernet Design Example for Intel Cyclone 10 GX Devices* chapter: — Updated the hardware testing description for the *Hardware and Software Requirements* topic. — Updated Figure: *Clocking and Reset Scheme for 10GBASE-R Design Example*. — Updated Table: *Hardware Test Cases* to update the description of the `source gen_conf.tcl` command for FMC loopback test case. • Updated the *Interface Signals Description* chapter: — Updated Table: *Avalon-MM Interface Signals*: • Added the following signals: `csr_mch_write`, `csr_mch_writedata`, `csr_mch_read`, `csr_mch_readdata`, `csr_mch_address`, and `csr_mch_waitrequest`. • Removed the following signals: `csr_write`, `csr_writedata`, `csr_read`, `csr_readdata`, `csr_address`, and `csr_waitrequest`. • Updated the *Configuration Registers Description* chapter: — Added the following topics: • *Register Access Definition* • *1G/2.5G/5G/10G Multi-rate PHY* — Removed the *Register Map* topic.
2018.06.28	18.0	• Renamed the document as *Low Latency Ethernet 10G MAC Intel Cyclone 10 GX FPGA IP Design Example User Guide*. • Made minor editorial updates to the document.
2018.05.16	18.0	Initial release.