



Low Latency 40G for ASIC Proto Ethernet Intel® FPGA IP Design Example User Guide

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

IP Version: **19.1.0**



[Subscribe](#)

[Send Feedback](#)

UG-20296 | 2020.10.05

Latest document on the web: [PDF](#) | [HTML](#)



Contents

1. Quick Start Guide.....	3
1.1. Generating the Design Example.....	4
1.1.1. Design Example Parameters.....	5
1.2. Directory Structure.....	6
1.3. Simulating the Design Example Testbench.....	7
1.4. Compiling and Configuring the Design Example in Hardware.....	7
1.5. Testing the Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Design in Hardware.....	8
2. Design Example Description.....	10
2.1. Features.....	10
2.2. Hardware and Software Requirements.....	11
2.3. Functional Description.....	11
2.4. Simulation.....	11
2.5. Hardware Testing.....	14
2.5.1. Internal Loopback Test.....	15
2.5.2. External Loopback Test.....	16
2.6. Ethernet Toolkit Overview.....	16
2.6.1. Features.....	17
2.7. Low Latency 40G for ASIC Proto Ethernet Design Example Registers.....	18
2.8. Design Example Interface Signals.....	19
3. Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Design Example User Guide Archive.....	21
4. Document Revision History for Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Design Example User Guide.....	22

1. Quick Start Guide

The Low Latency 40G for ASIC Proto Ethernet Intel® FPGA IP core provides a simulation testbench and a hardware design example that supports compilation and hardware testing. When you generate the design example, the Intel Quartus® Prime IP parameter editor automatically creates the files necessary to simulate, compile, and test the design in hardware.

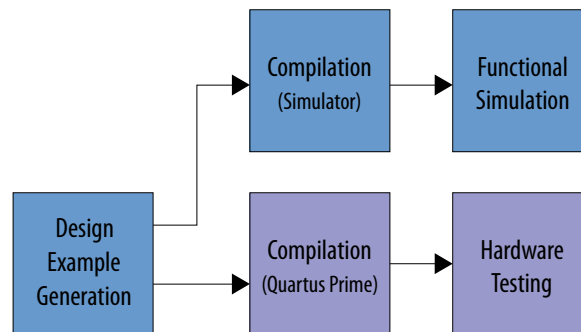
In addition, you can download the compiled hardware design to the Intel device-specific development kit for interoperative testing. The Intel FPGA IP also includes a compilation-only example project that you can use to quickly estimate IP core area and timing.

The Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP supports design example generation with a wide range of parameters. However, the design examples do not cover all possible parameterizations of the Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Core.

Table 1. List of Supported Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Design Example Variants

Variant	Simulation	Compilation-Only Project	Hardware Design Example
MAC and PCS	✓	✓	✓
PCS Only	✓	✓	✓

Figure 1. Development Steps for the Design Example



Related Information

- [Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP User Guide](#)
For detailed information on Low Latency 40G for ASIC Proto Ethernet IP.
- [Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Release Notes](#)
The IP Release Notes list IP changes in a particular release.

1.1. Generating the Design Example

Figure 2. Procedure

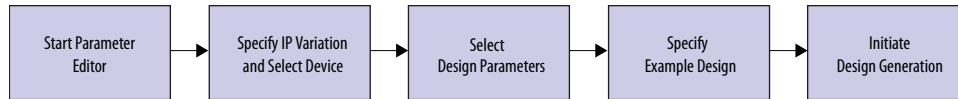
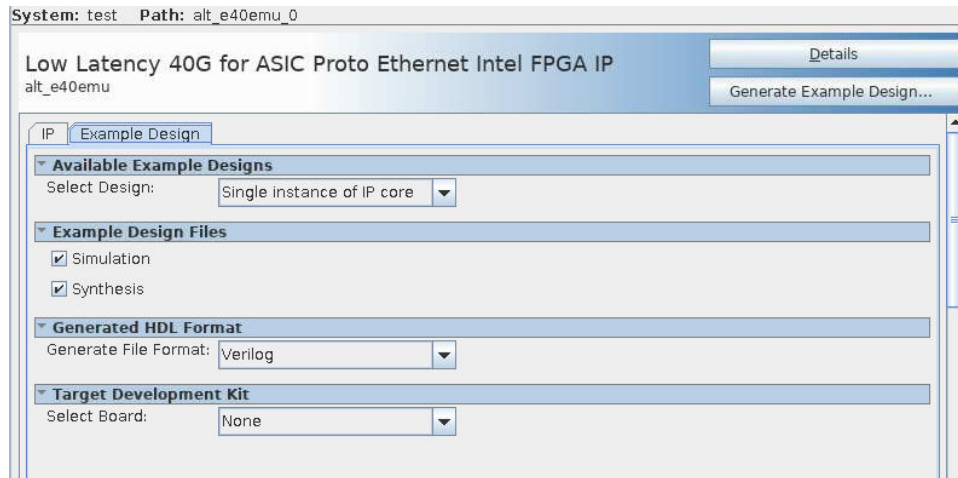


Figure 3. Example Design Tab in the Low Latency 40G for ASIC Proto Ethernet IP Parameter Editor



Follow these steps to generate the hardware design example and testbench:

1. In the Intel Quartus Prime Pro Edition software, click **File** ► **New Project Wizard** to create a new Intel Quartus Prime project, or **File** ► **Open Project** to open an existing Intel Quartus Prime software project. The wizard prompts you to specify a device family and device.

Note: The design example overwrites the selection with the device on the target board. You specify the target board from the menu of design example options in the Example Design tab (Step 8). The design example DUT if you select an H-tile device is 1SG10MHN3F74C2LGSx_Ux, where x is either 1 or 2.

2. In the IP Catalog, locate and select **Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP**. The **New IP Variation** window appears.
3. Specify a top-level name for your custom IP variation. The Intel Quartus Prime IP parameter editor saves the IP variation settings in a file named `<your_ip>.ip`.
4. Click **OK**. The IP parameter editor appears.
5. On the **IP** tab, specify the parameters for your IP core variation.



Note: The Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP design example does not simulate correctly and does not function correctly if you specify any of the following parameters:

- Use external TX MAC PLL
 - **Enable preamble pass-through** turned on
 - **Ready latency** set to the value of 3
 - **Enable TX CRC insertion** turned off
6. On the **Example Design** tab, under **Example Design Files**, enable the **Simulation** option to generate the testbench, and select the **Synthesis** option to generate the compilation-only and hardware design examples.

Note: On the **Example Design** tab, under **Generated HDL Format**, only Verilog HDL is available. This IP core does not support VHDL.

7. Under **Target Development Kit** select the **None**.
8. Click the **Generate Example Design** button. The **Select Example Design Directory** window appears.
9. If you want to modify the design example directory path or name from the defaults displayed (`alt_e40emu_0_example_design`), browse to the new path and type the new design example directory name (`<design_example_dir>`).
10. Click **OK**.

Related Information

[IP Core Parameters](#)

Provides more information about customizing your IP core.

1.1.1. 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 a design from the Preset library, this field shows the selected design.
Example Design Files	The files to generate for the different development phase. <ul style="list-style-type: none"> • 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 development board, the <i>Target Device</i> 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. None: This option excludes the hardware aspects for the design example.

1.2. Directory Structure

The Low Latency 40G for ASIC Proto Ethernet IP core design example file directories contain the following generated files for the design example.

Figure 4. Directory Structure for the Generated Design Example



- The simulation files (testbench for simulation only) are located in `<design_example_dir>/example_testbench`.
- The compilation-only example design is located in `<design_example_dir>/compilation_test_design`.
- The hardware configuration and test files (the hardware design example) are located in `<design_example_dir>/hardware_test_design`.

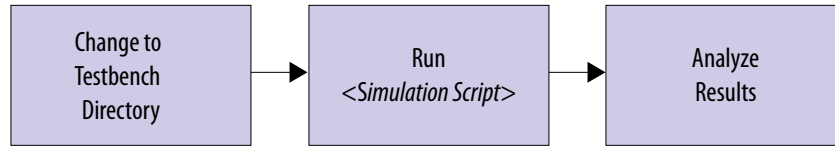
Table 3. Directory and File Descriptions

File Names	Description
eth_ex_40g.qpf	Intel Quartus Prime project file.
eth_ex_40g.qsf	Intel Quartus Prime project settings file.
eth_ex_40g.sdc	Synopsys* Design Constraints file. You can copy and modify this file for your own Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP design.
eth_ex_40g.srf	Intel Quartus Prime project message suppression rule file.
eth_ex_40g.v	Top-level Verilog HDL design example file.
eth_ex_40g_clock.sdc	Synopsys Design Constraints file for clocks.
common/	Hardware design example support files.
hwtest/main.tcl	Main file for accessing System Console.



1.3. Simulating the Design Example Testbench

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 `<design_example_dir>/example_testbench`.
2. Run the simulation script for the supported simulator of your choice. The script compiles and runs the testbench in the simulator.

Table 4. Instructions to Simulate the Testbench

Simulator	Instructions
ModelSim*	In the command line, type <code>vsim -do run_vsim.do</code> . If you prefer to simulate without bringing up the ModelSim GUI, type <code>vsim -c -do run_vsim.do</code> . <i>Note:</i> The ModelSim-AE and ModelSim-ASE simulators cannot simulate this IP core. You must use another supported ModelSim simulator such as ModelSim SE.
VCS*	In the command line, type <code>sh run_vcs.sh</code>
VCS MX	In the command line, type <code>sh run_vcsmx.sh</code> . Use this script when the design contains Verilog HDL and System Verilog with VHDL.
NCSim	In the command line, type <code>sh run_ncsim.sh</code>
Xcelium*	In the command line, type <code>sh run_xcelium.sh</code>

A successful simulation ends with the following message:

```
Simulation Passed.
```

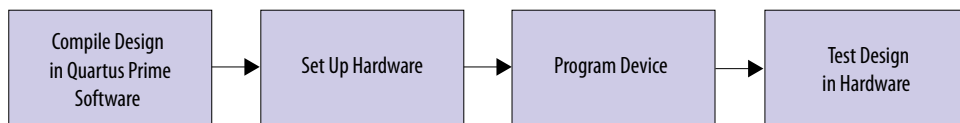
or

```
Testbench complete.
```

After successful completion, you can analyze the results.

1.4. Compiling and Configuring the Design Example in Hardware

The Intel FPGA IP core parameter editor allows you to compile and configure the design example on a target development kit.



To compile and configure a design example on hardware, follow these steps:

1. Launch the Intel Quartus Prime Pro Edition software and select **Processing > Start Compilation** to compile the design.
2. After you generate an SRAM object file `.sof`, follow these steps to program the hardware design example on the Intel device:
 - a. Select **Tools > Programmer**.
 - b. In the Programmer, click **Hardware Setup**.
 - c. Select a programming device.
 - d. Select and add the Intel TX board to your Intel Quartus Prime Pro Edition session.
 - e. Ensure that **Mode** is set to **JTAG**.
 - f. Select the Intel device and click **Add Device**. The Programmer displays a block diagram of the connections between the devices on your board.
 - g. In the row with your `.sof`, check the box for the `.sof`.
 - h. Turn on **Program/Configure** option for the `.sof`.
 - i. Click **Start**.

Related Information

- [Incremental Compilation for Hierarchical and Team-Based Design](#)
- [Programming Intel FPGA Devices](#)

1.5. Testing the Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Design in Hardware

After you compile the Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP core design example and configure it on your Intel device, you can use the System Console to program the IP core and its embedded Native PHY IP core registers.

To turn on the System Console and test the hardware design example, follow these steps:

1. In the Intel Quartus Prime Pro Edition software, select **Tools > System Debugging Tools > System Console** to launch the system console.
2. In the Tcl Console pane, type `cd hwtest` to change directory to `/hardware_test_design/hwtest`.
3. Type `source main.tcl` to open a connection to the JTAG master.

Additional design example commands are available to program the IP core:

- `chkphy_status`: Displays the clock frequencies and PHY lock status.
- `chkmac_stats`: Displays the values in the MAC statistics counters.
- `clear_all_stats`: Clears the IP core statistics counters.
- `start_pkt_gen`: Starts the packet generator.
- `stop_pkt_gen`: Stops the packet generator.
- `sys_reset_digital_analog`: System reset.



- `loop_on`: Turns on internal serial loopback
- `loop_off`: Turns off internal serial loopback.
- `reg_read <addr>`: Returns the IP core register value at `<addr>`.
- `reg_write <addr> <data>`: Writes `<data>` to the IP core register at address `<addr>`.

Follow the test procedure in the *Hardware Testing* section of the design example and observe the test results in the System Console.

Related Information

[Analyzing and Debugging Designs with System Console](#)

2. Design Example Description

The 40G Ethernet design example demonstrates the functions of the Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP core with integrated MAC and PHY solution compliant with the IEEE 802.3 (2010) standard. You can generate the design from the **Example Design** tab in the Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP parameter editor.

To generate the design example, you must first set the parameter values for the IP core variation you intend to generate in your end product. Generating the design example creates a copy of the IP core; the testbench and hardware design example use this variation as the DUT. If you do not set the parameter values for the DUT to match the parameter values in your end product, the design example you generate does not exercise the IP core variation you intend.

Low Latency 40G for ASIC Proto Ethernet IP allows you to implement the IP with or without MAC via **Select USER MAC mode** parameter selection:

- **PCS+MAC**—allows you to integrate the MAC within the IP. The IP integrates the MAC.
- **PCS_Only**—allows you to generate IP without MAC. You need to create a separate user MAC.

Note: The testbench demonstrates a basic test of the IP core. It is not intended to be a substitute for a full verification environment. You must perform more extensive verification of your own Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP design in simulation and in hardware.

Related Information

[Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP User Guide](#)

Provides detailed descriptions of the Low Latency 40G for ASIC Proto Ethernet IP core signals and the interfaces to which they belong.

2.1. Features

- Supports 40G for ASIC Proto Ethernet IP core using the Intel Stratix® 10 device.
- Supports selectable user MAC mode.
- Supports TX CRC insertion.
- Supports preamble pass-through and link training.
- Generates design example with MAC stats counters feature.
- Provides testbench and simulation script.



2.2. Hardware and Software Requirements

To test the example design, use the following hardware and software:

- Intel Quartus Prime Pro Edition software
- System Console
- ModelSim, VCS, VCS MX, NCSim, or Xcelium Simulator

2.3. Functional Description

This section describes the 40G Ethernet MAC/PCS and 40G Ethernet PCS IP core using the Intel device. You can choose to configure MAC within the IP or implement a separate MAC outside of this IP.

Configuring the 40G Ethernet IP with MAC and PCS

In the transmit direction, the MAC accepts client frames and inserts inter-packet gap (IPG), preamble, the start of frame delimiter (SFD), padding, and CRC bits before passing them to the PHY. The PHY encodes the MAC frame as required for reliable transmission over the media to the remote end.

In the receive direction, the PHY passes frames to the MAC. The MAC accepts frames from the PHY, performs checks, strips out the CRC, preamble, and SFD, and passes the rest of the frame to the client.

- In RX preamble pass-through mode, the MAC passes on the preamble and SFD to the client instead of removing them out of frames.
- In RX CRC pass-through mode, the MAC passes on the CRC bytes to the client and asserts the EOP signal in the same clock cycle with the final CRC byte.

Configuring the 40G Ethernet IP with PCS Only

The PHY encodes the user MAC frame to ensure reliable transmission over the media to the remote end via the Media-Independent Interface (MII).

Related Information

[Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP User Guide](#)

Provides detailed descriptions of the Low Latency 40G for ASIC Proto Ethernet IP core signals and the interfaces to which they belong.

2.4. Simulation

The testbench sends traffic through the IP core, exercising the transmit side and receive side of the IP core.

The following figures depict the Low Latency 40G for ASIC Proto Ethernet IP block diagram configuration. Note that the simulation is not affected by the IP's configuration.

Figure 5. Low Latency 40G for ASIC Proto Ethernet IP Design Example Block Diagram with MAC and PCS

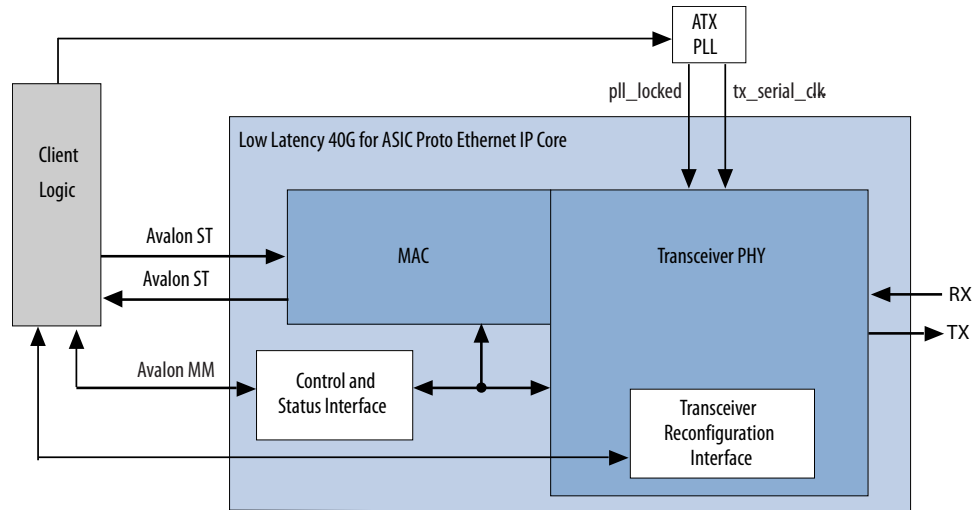
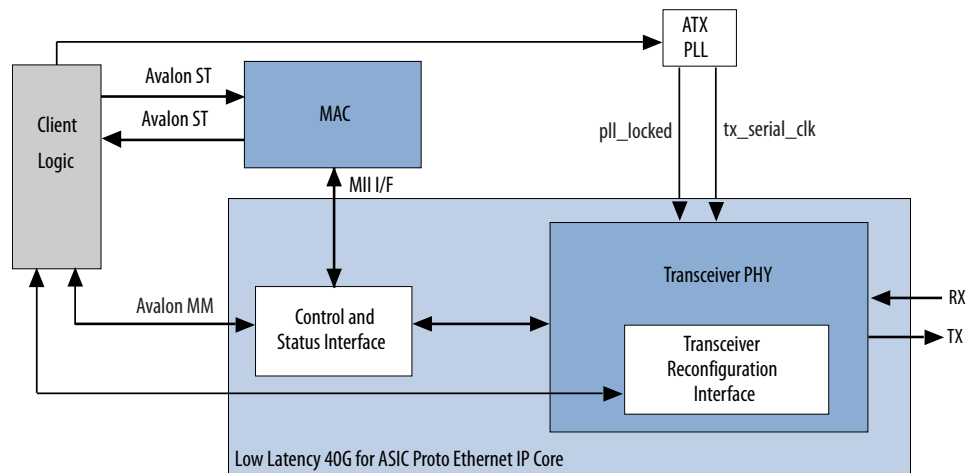


Figure 6. Low Latency 40G for ASIC Proto Ethernet IP Design Example Block Diagram with PCS Only



In both configuration, the simulation design example top-level test file is `basic_avl_tb_top.sv`. This file instantiates and connects an ATX PLL. It includes a task to send and receive 10 packets.

Table 5. Low Latency 40G for ASIC Proto Ethernet Core Testbench File Descriptions

File Names	Description
Testbench and Simulation Files	
<code>basic_avl_tb_top.sv</code>	Top-level testbench file. The testbench instantiates the DUT and runs Verilog HDL tasks to generate and accept packets.
<code>basic_avl_tb_top_nc.sv</code>	Top-level testbench file compatible with the NCSim simulator.
<code>basic_avl_tb_top_msim.sv</code>	Top-level testbench file compatible with the ModelSim simulator.
<i>continued...</i>	



File Names	Description
Testbench Scripts	
run_vsim.do	The Mentor Graphics* ModelSim script to run the testbench.
run_vcs.sh	The Synopsys VCS script to run the testbench.
run_vcsmx.sh	The Synopsys VCS MX script (combined Verilog HDL and System Verilog with VHDL) to run the testbench.
run_ncsim.sh	The Cadence NCSim script to run the testbench.
run_xcelium.sh	The Cadence Xcelium script to run the testbench.

The successful test run displays output confirming the following behavior:

1. Waiting for RX clock to settle
2. Printing PHY status
3. Sending 10 packets
4. Receiving 10 packets
5. Displaying "Testbench complete."

The following sample output illustrates a successful simulation test run:

```
#Waiting for RX alignment
#RX deskew locked
#RX lane alignment locked
#TX enabled
***Sending Packet 1...
***Sending Packet 2...
***Sending Packet 3...
***Sending Packet 4...
***Sending Packet 5...
***Sending Packet 6...
***Sending Packet 7...
***Received Packet 1...
***Sending Packet 8...
***Received Packet 2...
***Sending Packet 9...
***Received Packet 3...
***Sending Packet 10...
***Received Packet 4...
***Received Packet 5...
***Received Packet 6...
***Received Packet 7...
***Received Packet 8...
***Received Packet 9...
***Received Packet 10...
***
*** Testbench complete.
***
#*****
```

Related Information

[Simulating the Design Example Testbench on page 7](#)

2.5. Hardware Testing

In the hardware design example, you can program the IP core in internal serial loopback mode and generate traffic on the transmit side that loops back through the receive side.

Figure 7. Low Latency 40G for ASIC Proto Ethernet IP Hardware Design Example High Level Block Diagram: MAC with PCS variant

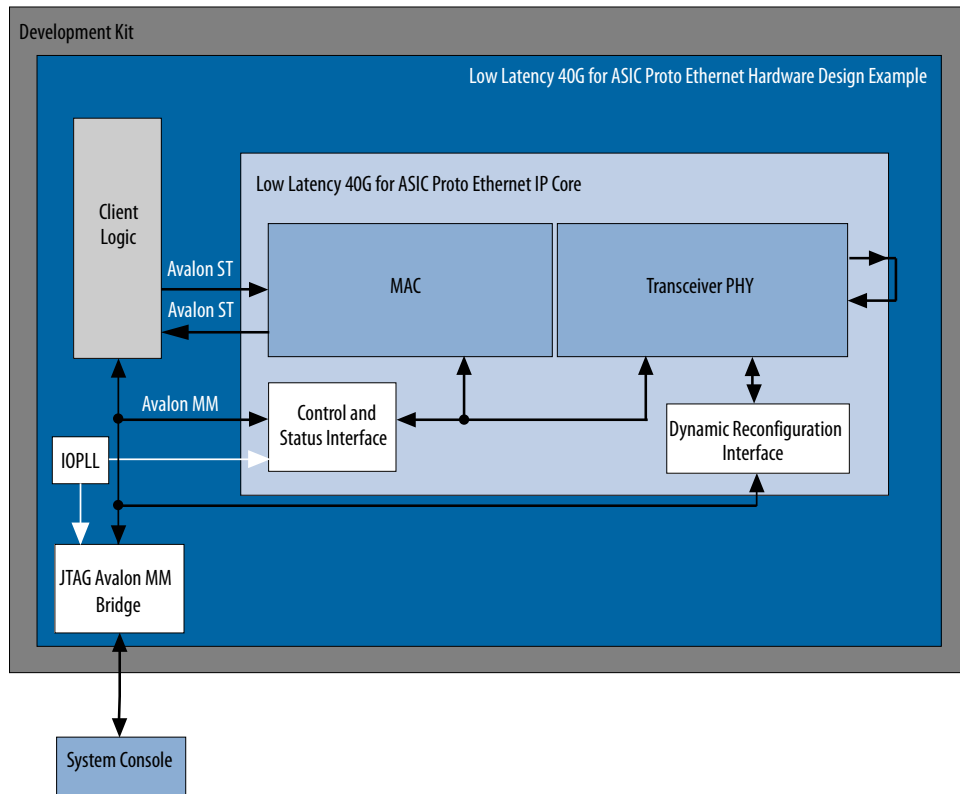
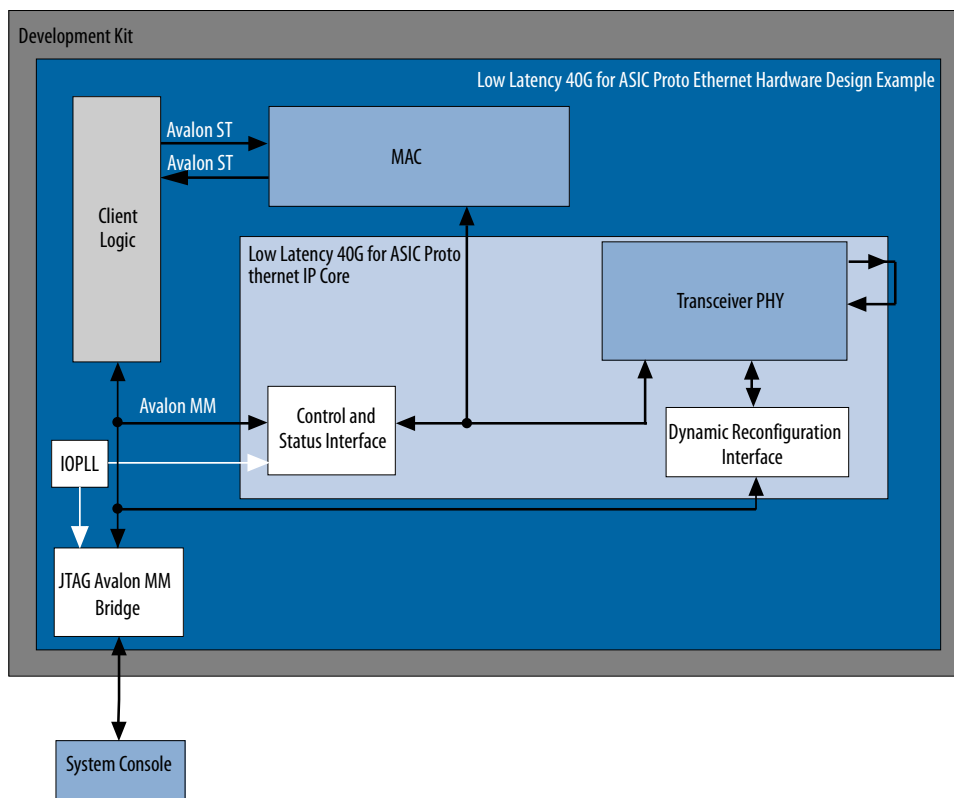


Figure 8. Low Latency 40G for ASIC Proto Ethernet IP Hardware Design Example High Level Block Diagram: PCS variant



The Low Latency 40G for ASIC Proto Ethernet hardware design example includes the following components:

- Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP core.
- Client logic that coordinates the programming of the IP core, and packet generation and checking.
- IOPLL to generate a 100 MHz clock from a 50 MHz input clock to the hardware design example.
- JTAG controller that communicates with the Intel System Console. You communicate with the client logic through the System Console.

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

Related Information

- [Testing the Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Design in Hardware](#) on page 8
- [Analyzing and Debugging Designs with System Console](#)

2.5.1. Internal Loopback Test

Run these steps to perform the internal loopback test:

1. Reset the system.

```
sys_reset_digital_analog
```

2. Display the clock frequency and PHY status.

```
chkphy_status
```

3. Turn on the internal loopback test.

```
loop_on
```

4. Display the clock frequency and PHY status. The `rx_clk` is set to 312.5 MHz and `rx_pcs_ready` is set to 1.

```
chkphy_status
```

5. Start the packet generator.

```
start_pkt_gen
```

6. Stop the packet generator.

```
stop_pkt_gen
```

7. Review the number of transmitted and received packets.

```
chkmac_stats
```

8. Turn off the internal loopback test.

```
loop_off
```

2.5.2. External Loopback Test

Run these steps to perform the external loopback test:

1. Reset the system.

```
sys_reset_digital_analog
```

2. Display the clock frequency and PHY status. The `rx_clk` is set to 312.5 MHz and `rx_pcs_ready` is set to 1.

```
chkphy_status
```

3. Start the packet generator.

```
start_pkt_gen
```

4. Stop the packet generator.

```
stop_pkt_gen
```

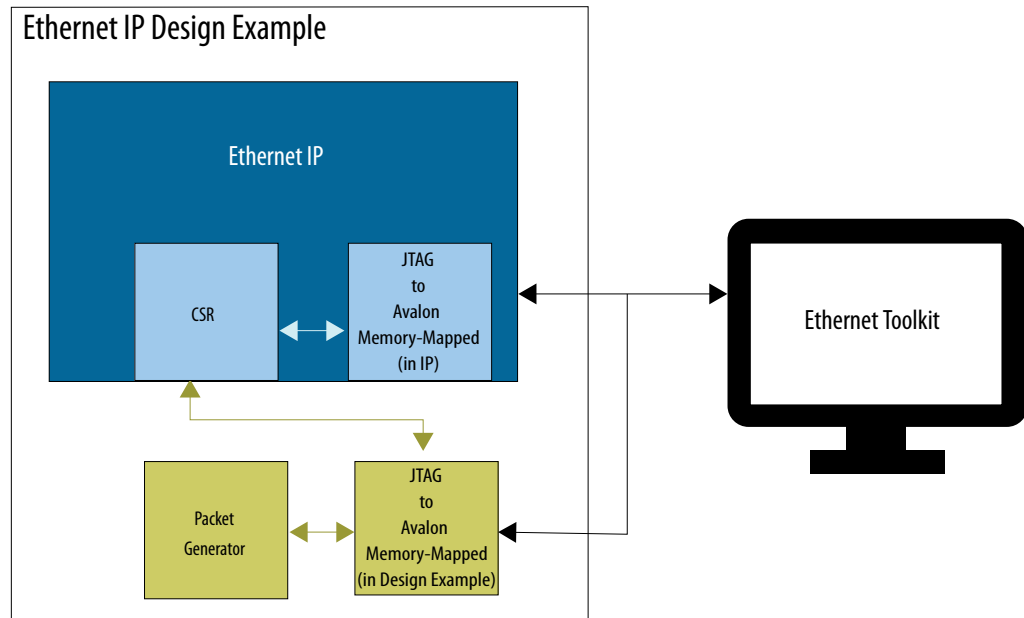
5. Review the number of transmitted and received packets.

```
chkmac_stats
```

2.6. Ethernet Toolkit Overview

The Ethernet Toolkit is a TCL based debugging tool that allows you to interact with an Ethernet Intel FPGA IP in real time.

Figure 9. Block Diagram of the Ethernet Toolkit



You can use the Ethernet Toolkit with hardware design that has standalone Ethernet IP. You can also use the Ethernet Toolkit with an Intel Quartus Prime generated Ethernet IP design example.

2.6.1. Features

The Ethernet Toolkit offers the following features when used with hardware design that has standalone Ethernet IP as well as with an Intel Quartus Prime generated Ethernet IP design example:

- Verifies the status of the Ethernet link.
- Reads and writes to status and configuration registers of the IP.
- Displays the values of TX/RX status and statistics registers.
- Ability to assert and deassert IP resets.
- Verifies the IPs error correction capability.

The Ethernet Toolkit also offers some additional features when used with an Intel Quartus Prime generated Ethernet IP design example:

- Provides access to the example design packet generator.
- Execute testing procedures to verify the functionality of Ethernet IPs.
- Enable and disable MAC loopback.
- Set source and destination MAC addresses.

Related Information

[Ethernet Toolkit User Guide](#)



2.7. Low Latency 40G for ASIC Proto Ethernet Design Example Registers

Table 6. Low Latency 40G for ASIC Proto Ethernet Hardware Design Example Register Map

Lists the memory mapped register ranges for the hardware design example. You access these registers with the `reg_read` and `reg_write` functions in the System Console.

Word Offset	Register Type
0x300-0x3FF	PHY registers
0x400-0x4FF	TX MAC registers
0x500-0x5FF	RX MAC registers
0x800-0x8FF	Statistics Counter registers - TX direction
0x900-0x9FF	Statistics Counter registers - RX direction
0x1000-1016	Packet Client registers

Table 7. Packet Client Registers

You can customize the Low Latency 40G for ASIC Proto Ethernet hardware design example by programming the packet client registers.

Addr	Name	Bit	Description	HW Reset Value	Access
0x1000	PKT_CL_SCRATCH	[31:0]	Scratch register available for testing.		RW
0x1001	PKT_CL_CLNT	[31:0]	Four characters of IP block identification string "CLNT"		RO
0x1008	Packet Size Configure	[29:0]	Specify the transmit packet size in bytes. These bits have dependencies to <code>PKT_GEN_TX_CTRL</code> register. <ul style="list-style-type: none"> Bit [29:16]: Specify the upper limit of the packet size in bytes. This is only applicable to incremental mode. Bit [13:0]: <ul style="list-style-type: none"> For fixed mode, these bits specify the transmit packet size in bytes. For incremental mode, these bits specify the incremental bytes for a packet. 	0x25800040	RW
0x1009	Packet Number Control	[31:0]	Specify the number of packets to transmit from the packet generator.	0xA	RW
0x1010	PKT_GEN_TX_CTRL	[7:0]	<ul style="list-style-type: none"> Bit [0]: Reserved. Bit [1]: Packet generator disable bit. Set this bit to the value of 1 to turn off the packet generator, and reset it to the value of 0 to turn on the packet generator. Bit [2]: Reserved. Bit [3]: Has the value of 1 if the IP core is in MAC loopback mode; has the value of 0 if the packet client uses the packet generator. 	0x6	RW

continued...



Addr	Name	Bit	Description	HW Reset Value	Access
			<ul style="list-style-type: none"> Bit [5:4]: <ul style="list-style-type: none"> 00: Random mode 01: Fixed mode 10: Incremental mode Bit [6]: Set this bit to 1 to use 0x1009 register to turn off packet generator based on a fixed number of packets to transmit. Otherwise, bit [1] of PKT_GEN_TX_CTRL register is used to turn off the packet generator. Bit [7]: <ul style="list-style-type: none"> 1: For transmission without gap in between packets. 0: For transmission with random gap in between packets. 		
0x1011	Destination address lower 32 bits	[31:0]	Destination address (lower 32 bits)	0x56780ADD	RW
0x1012	Destination address upper 16 bits	[15:0]	Destination address (upper 16 bits)	0x1234	RW
0x1013	Source address lower 32 bits	[31:0]	Source address (lower 32 bits)	0x43210ADD	RW
0x1014	Source address upper 16 bits	[15:0]	Source address (upper 16 bits)	0x8765	RW
0x1016	PKT_CL_LOOPB ACK_RESET	[0]	MAC loopback reset. Set to the value of 1 to reset the design example MAC loopback.	1'b0	RW

Related Information

[Low Latency 40G for ASIC Proto Ethernet Control and Status Register Descriptions](#)
Describes the Low Latency 40G for ASIC Proto Ethernet IP core registers.

2.8. Design Example Interface Signals

The Low Latency 40G for ASIC Proto Ethernet testbench is self-contained and does not require you to drive any input signals.

Table 8. Low Latency 40G for ASIC Proto Ethernet Hardware Design Example Interface Signals

Signal	Direction	Comments
clk50	Input	Drive at 50 MHz. The intent is to drive it from a 50 MHz oscillator on the board. The hardware design example routes this clock to the input of an IOPLL on the device and configures the IOPLL to drive a 100 MHz clock internally.
clk_ref	Input	Drive at 644.53125 MHz.
<i>continued...</i>		



Signal	Direction	Comments
cpu_resetn	Input	Resets the IP core. Active low. Drives the global hard reset <code>csr_reset_n</code> to the IP core.
tx_serial[3:0]	Output	Transceiver PHY output serial data.
rx_serial[3:0]	Input	Transceiver PHY input serial data.
user_led[7:0]	Output	Status signals. The hardware design example connects these bits to drive LEDs on the target board. Individual bits reflect the following signal values and clock behavior: <ul style="list-style-type: none">[0]: Main reset signal to IP core[1]: Divided version of <code>clk_ref</code>[2]: Divided version of <code>clk50</code>[3]: Divided version of 100 MHz status clock[4]: <code>tx_lanes_stable</code>[5]: <code>rx_block_lock</code>[6]: <code>rx_am_lock</code>[7]: <code>rx_pcs_ready</code>

Related Information

[Interfaces and Signal Descriptions](#)

Provides detailed descriptions of the Low Latency 40G for ASIC Proto Ethernet IP core signals and the interfaces to which they belong.



3. Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Design Example User Guide Archive

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

Intel Quartus Prime Version	IP Core Version	User Guide
20.2	19.1.0	Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP



4. Document Revision History for Low Latency 40G for ASIC Proto Ethernet Intel FPGA IP Design Example User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2020.10.05	20.3	19.1.0	<ul style="list-style-type: none"> Updated procedure in the <i>Generated the Design Example</i>. Added step to select a target development kit. Revised Select USER MAC mode parameter selection. The available options are PCS+MAC and PCS_Only. Added new topic: <i>Ethernet Toolkit Overview</i>.
2020.07.07	20.2	19.1.0	Initial release.

Intel Corporation. All rights reserved. Agilix, Altera, Arria, Cyclone, Enpirion, Intel, the Intel logo, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered