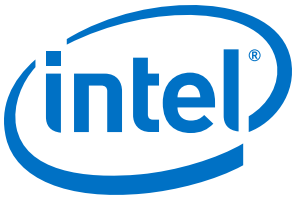


eCPRI Intel® FPGA IP Design Example User Guide

Updated for Intel® Quartus® Prime Design Suite: **20.4**

IP Version: **1.3.0**

Subscribe
Send Feedback

UG-20278 | 2021.02.26
Latest document on the web: **PDF** | **HTML**

Contents

1. Quick Start Guide

The eCPRI Intel FPGA IP provides a simulation testbench and a hardware design example that supports compilation and hardware testing. When you generate the design example, the parameter editor automatically creates the files necessary to simulate, compile, and test the design example in hardware.

The compiled hardware design example runs on:

- Intel® Agilex™ F-Series Transceiver-SoC Development Kit
- Intel Stratix® 10 GX Transceiver Signal Integrity Development Kit for the H-tile design examples
- Intel Stratix 10 TX Transceiver Signal Integrity Development Kit for the E-tile design examples
- Intel Arria® 10 GX Transceiver Signal Integrity Development Kit

Intel provides a compilation-only example project that you can use to quickly estimate IP core area and timing.

The testbench and design example supports 25G and 10G data rates for Intel Stratix 10 H-or E-tile and Intel Agilex E-tile device variations of the eCPRI IP.

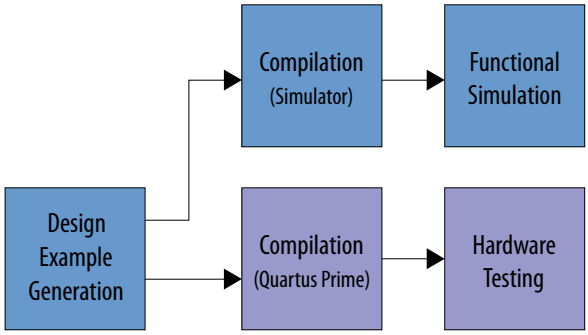
Note: The eCPRI IP design example with interworking function (IWF) is only available for 9.8 Gbps CPRI line bit rate in the current release.

Note: The eCPRI IP design example does not support dynamic reconfiguration for 10G data rate in Intel Arria 10 designs.

The eCPRI Intel FPGA IP core design example supports the following features:

- Internal TX to RX serial loopback mode
- Traffic generator and checker
- Basic packet checking capabilities
- Ability to use System Console to run the design and reset the design for re-testing purpose

Figure 1. Development Steps for the Design Example



Related Information

- eCPRI Intel FPGA IP User Guide
- eCPRI Intel FPGA IP Release Notes

1.1. Hardware and Software Requirements

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

- Intel Quartus® Prime Pro Edition software version 20.4
- System Console
- Modelsim-SE*, VCS*, VCS MX*, NCSim*, Aldec Riviera*, and Xcelium Parallel Simulator*
- Development Kit:
  - Intel Agilex F-Series Transceiver-SoC Development Kit
  - Intel Stratix 10 GX Transceiver Signal Integrity Development Kit for the H-tile device variation design example
  - Intel Stratix 10 TX Transceiver Signal Integrity Development for the E-tile device variation design example
  - Intel Arria 10 GX Transceiver Signal Integrity Development Kit

Related Information

- Intel Agilex F-Series Transceiver-SoC Development Kit User Guide
- Intel Stratix 10 GX Transceiver Signal Integrity Development Kit User Guide
- Intel Stratix 10 TX Transceiver Signal Integrity Development Kit User Guide
- Intel Arria 10 GX Transceiver Signal Integrity Development Kit User Guide

1.2. Generating the Design

Prerequisite: Once you receive the eCPRI web-core IP, save the web-core installer to the local area. Run the installer with Windows/Linux. When prompted, install the web-core to the same location as Intel Quartus Prime folder. The eCPRI Intel FPGA IP now appears in the IP Catalog.

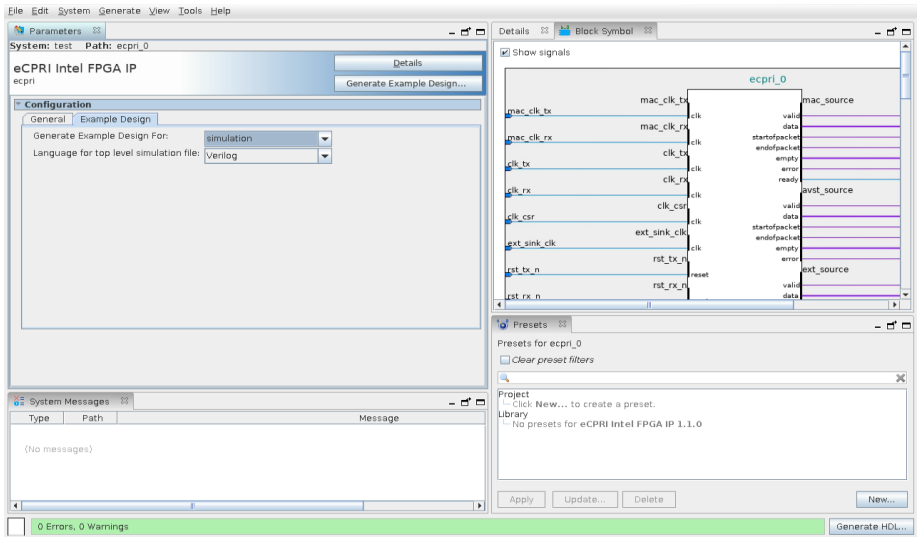
If you do not already have an Intel Quartus Prime Pro Edition project in which to integrate your eCPRI Intel FPGA IP core, you must create one.

1. In the Intel Quartus Prime Pro Edition software, click **File ➤ New Project Wizard** to create a new Intel Quartus Prime project, or click **File ➤ Open Project** to open an existing Intel Quartus Prime project. The wizard prompts you to specify a device.
2. Specify the device family and a device that meets the speed grade requirements.
3. Click **Finish**.
4. In the IP Catalog, locate and double-click **eCPRI Intel FPGA IP**. The **New IP Variant** window appears.

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

1. In the IP Catalog, locate and double-click **eCPRI Intel FPGA IP**. The **New IP Variant** window appears.
2. Click **OK**. The parameter editor appears.

Figure 2. Example Design Tab in the eCPRI Intel FPGA IP Parameter Editor



3. Specify a top-level name <*your_ip*> for your custom IP variation. The parameter editor saves the IP variation settings in a file named <*your_ip*>.ip.
4. Click **OK**. The parameter editor appears.
5. On the **General** tab, specify the parameters for your IP core variation.

   Note:
   - You must turn on **Streaming** parameter in the eCPRI IP parameter editor when you generate the design example with **Interworking Function (IWF) Support** parameter enabled,
   - You must set the **CPRI Line Bit Rate (Gbit/s)** to **Others** when generating the design example with **Interworking Function (IWF) Support** parameter enabled.
6. On the **Example Design** tab, select the **simulation** option to generate the testbench, select the **synthesis** option to generate the hardware example design, and select **synthesis and simulation** option to generate both the testbench and the hardware design example.
7. For **Language for top level simulation file**, select **Verilog** or **VHDL**.

   Note: This option is available only when you select **Simulation** option for your example design.
8. For **Language for top level synthesis file**, select **Verilog** or **VHDL**.

   Note: This option is available only when you select **Synthesis** option for your example design.
9. Click **Generate Example Design**. The **Select Example Design Directory** window appears.
10. If you want to modify the design example directory path or name from the defaults displayed (`ecpri_0_testbench`), browse to the new path and type the new design example directory name.
11. Click **OK**.

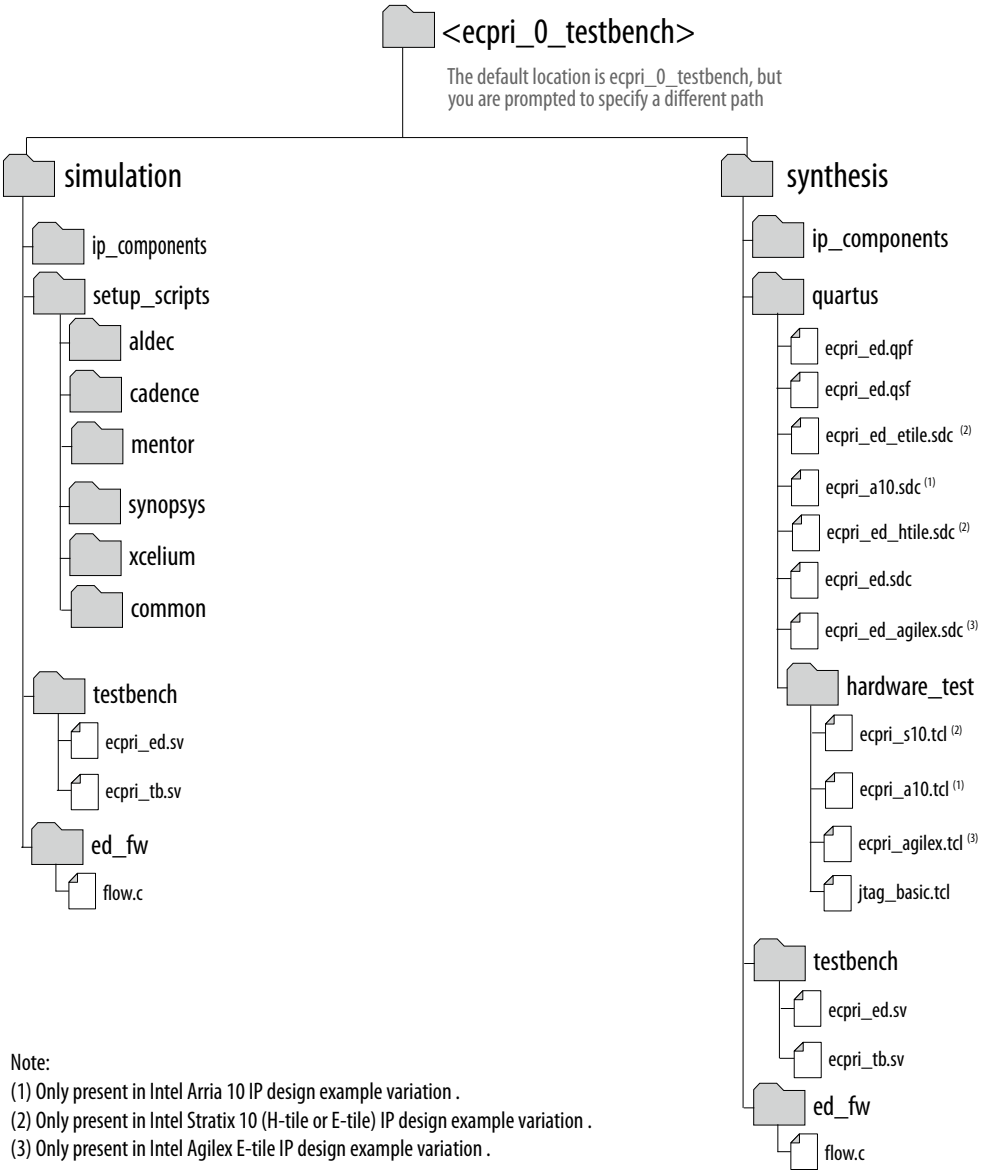
Related Information

eCPRI Intel FPGA IP User Guide

1.3. Directory Structure

The eCPRI IP core design example file directories contain the following generated files for the design example.

Figure 3. Directory Structure of the Generated Example Design



Note:
(1) Only present in Intel Arria 10 IP design example variation .
(2) Only present in Intel Stratix 10 (H-tile or E-tile) IP design example variation .
(3) Only present in Intel Agilex E-tile IP design example variation .

Table 1. eCPRI Intel FPGA IP Core Testbench File Descriptions

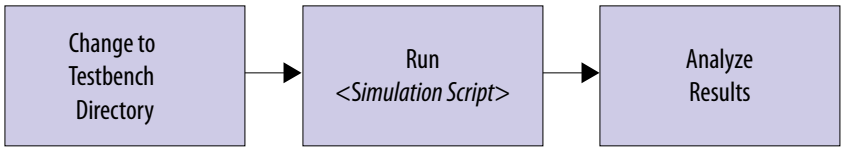
File Names	Description
Key Testbench and Simulation Files	
<design_example_dir>/simulation/testbench/ecpri_tb.sv	Top-level testbench file. The testbench instantiates the DUT wrapper and runs Verilog HDL tasks to generate and accept packets.
<design_example_dir>/simulation/testbench/ecpri_ed.sv	DUT wrapper that instantiates DUT and other testbench components.
<design_example_dir>/simulation/ed_fw/flow.c	C-code source file.
Testbench Scripts	
<design_example_dir>/simulation/setup_scripts/mentor/run_vsim.do	The Mentor Graphics ModelSim* script to run the testbench.
<design_example_dir>/simulation/setup_scripts/synopsys/vcs/run_vcs.sh	The Synopsys VCS script to run the testbench.
<design_example_dir>/simulation/setup_scripts/synopsys/vcsmx/run_vcsmx.sh	The Synopsys VCS MX script (combined Verilog HDL and SystemVerilog with VHDL) to run the testbench.
<design_example_dir>/simulation/setup_scripts/cadence/run_ncsim.sh	The Cadence NCSim script to run the testbench.
<design_example_dir>/simulation/setup_scripts/aldec/run_rivierapro.tcl	The Aldec Riviera script to run the testbench.
<design_example_dir>/simulation/setup_scripts/xcelium/run_xcelium.sh	The Xcelium* script to run the testbench.

Table 2. eCPRI Intel FPGA IP Core Hardware Design Example File Descriptions

File Names	Descriptions
<design_example_dir>/synthesis/quartus/ecpri_ed.qpf	Intel Quartus Prime project file.
<design_example_dir>/synthesis/quartus/ecpri_ed.qsf	Intel Quartus Prime project setting file.
<design_example_dir>/synthesis/quartus/ecpri_ed.sdc	Synopsys Design Constraints files. You can copy and modify these files for your own Intel Stratix 10 design.
<design_example_dir>/synthesis/testbench/ecpri_ed_top.sv	Top-level Verilog HDL design example file.
<design_example_dir>/synthesis/testbench/ecpri_ed.sv	DUT wrapper that instantiates DUT and other testbench components.
<design_example_dir>/synthesis/quartus/ecpri_s10.tcl	Main file for accessing System Console (Available in Intel Stratix 10 H-tile and E-tile designs).
<design_example_dir>/synthesis/quartus/ecpri_a10.tcl	Main file for accessing System Console (Available in Intel Arria 10 designs).
<design_example_dir>/synthesis/quartus/ecpri_agilex.tcl	Main file for accessing System Console (Available in Intel Agilex designs).

1.4. Simulating the Design Example Testbench

Figure 4. Procedure



Follow these steps to simulate the testbench:

1. At the command prompt, change to the testbench simulation directory `<design_example_dir>/simulation/setup_scripts`.
2. Run the simulation script for the supported simulator of your choice. The script compiles and runs the testbench in the simulator. Refer to the table *Steps to Simulate the Testbench*.

   Note: The VHDL language support for simulation is only available with ModelSim and VCS MX simulators. The Verilog language support for simulation is available for all simulators listed in *Table: Steps to Simulate the Testbench*.
3. Analyze the results. The successful testbench sends and receives packets, and displays "PASSED".

Table 3. Steps to Simulate the Testbench

Simulator	Instructions
Mentor Graphics ModelSim	In the command line, type `vsim -do run_vsim.do` If you prefer to simulate without bringing up the ModelSim GUI, type `vsim -c -do run_vsim.do` *Note:* The ModelSim - Intel FPGA Edition simulator does not have the capacity to simulate this IP core. You must use another supported ModelSim simulator such as ModelSim SE.
Cadence NCSim(1)	In the command line, type `sh run_ncsim.sh`
Synopsys VCS	In the command line, type `sh run_vcs.sh`
Synopsys VCSMX	In the command line, type `sh run_vcsmx.sh`
Aldec Riviera	In the command line, type `vsim -c -do run_rivierapro.tcl` *Note:* Only supported in Intel Stratix 10 H-tile design variations.
Xcelium(1)	In the command line, type `sh run_xcelium.sh`

The following sample output illustrates a successful simulation test run of the eCPRI IP design example without IWF feature enabled:

```
_____________________________________________________________
INFO: Out of reset status
_____________________________________________________________


eCPRI TX SOPs count : 0
eCPRI TX EOPs count : 0
eCPRI RX SOPs count : 0
eCPRI RX EOPs count : 0
External PTP TX SOPs count : 0
External PTP TX EOPs count : 0
```

(1) Not supported for eCPRI Intel FPGA IP design example generated with IWF feature enabled.

```
 External MISC TX SOPs count : 0
 External MISC TX EOPs count : 0
 External RX SOPs count : 0
 External RX EOPs count : 0
_____________________________________________________________
 INFO: Start transmitting packets
_____________________________________________________________



_____________________________________________________________
 INFO: Stop transmitting packets
_____________________________________________________________



_____________________________________________________________
 INFO: Checking packets statistics
_____________________________________________________________


 eCPRI SOPs transmitted: 300
 eCPRI EOPs transmitted: 300
 eCPRI SOPs received: 300
 eCPRI EOPs received: 300
 External PTP SOPs transmitted: 128
 External PTP EOPs transmitted: 128
 External MISC SOPs transmitted: 43
 External MISC EOPs transmitted: 43
 External SOPs received: 171
 External EOPs received: 171
_____________________________________________________________
 INFO: Test PASSED
_____________________________________________________________
```

The following sample output illustrates a successful simulation test run of the eCPRI IP design example with IWF feature enabled:

```
Waiting for CPRI achieve HSYNC link up state
# CPRI HSYNC state achieved
#     2011285000ps Write 1 to nego_bitrate_complete
#     2011305000ps Polling PROT_VER
# ____________________________________________________
#     2011325000ps Polling register: 0000000000000000000000a0000010
# ____________________________________________________
#     2211925000ps Write 1 to nego_protol_complete
#     2211945000ps Polling CM_STATUS.rx_fast_cm_ptr_valid
# ____________________________________________________
#     2211965000ps Polling register: 0000000000000000000000a0000020
# ____________________________________________________
#     2266585000ps Write 1 to nego_cm_complete
#     2266625000ps Write 1 to nego_vss_complete
# Waiting for CPRI achieve HSYNC & startup sequence FSM STATE_F
# CPRI HSYNC & startup sequence FSM STATE_F achieved
#          eCPRI version :     1
_____________________________________________________________
     INFO: Out of reset status
_____________________________________________________________

          eCPRI TX SOPs count :           0
          eCPRI TX EOPs count :           0
          eCPRI RX SOPs count :           0
          eCPRI RX EOPs count :           0
          External PTP TX SOPs count :           0
          External PTP TX EOPs count :           0
          External MISC TX SOPs count :           0
          External MISC TX EOPs count :           0
          External RX SOPs count :           0
          External RX EOPs count :           0
 _____________________________________________________________
     INFO: Start transmitting packets
```

```
_____________________________________________________________


     INFO: Waiting for the eCPRI TX traffic transfer to complete
      INFO: eCPRI TX traffic transfer completed
     INFO: Waiting for the eCPRI External TX PTP traffic transfer to
complete
     INFO: eCPRI External TX PTP traffic transfer completed
      INFO: Waiting for the eCPRI External TX Misc traffic transfer to
complete
      INFO: eCPRI External TX Misc traffic transfer completed
_____________________________________________________________
     INFO: Stop transmitting packets
_____________________________________________________________


_____________________________________________________________
     INFO: Checking packets statistics
_____________________________________________________________
         eCPRI SOPs transmitted:          50
         eCPRI EOPs transmitted:          50
         eCPRI SOPs received:             50
         eCPRI EOPs received:             50
         External PTP SOPs transmitted:          64
         External PTP EOPs transmitted:          64
         External MISC SOPs transmitted:         100
         External MISC EOPs transmitted:         100
         External SOPs received:            164
         External EOPs received:            164
_____________________________________________________________
     INFO: Test PASSED

_____________________________________________________________
```

1.4.1. Enabling Dynamic Reconfiguration to the Ethernet IP

By default, the dynamic reconfiguration is disabled in the eCPRI IP design example and it's only applicable to Intel Stratix 10 (E-tile and H-tile) and Intel Agilex (E-tile) design examples.

1. Look for the following line in the `test_wrapper.sv` from the generated `<*design_example_dir*>/simulation/testbench` directory:

   ```
   parameter ETHERNET_DR_EN = 0
   ```

2. Change the value from 0 to 1:

   ```
   parameter ETHERNET_DR_EN = 1
   ```

3. Rerun the simulation using the same generated example design directory.

1.5. Compiling the Compilation-Only Project

To compile the compilation-only example project, follow these steps:

1. Ensure compilation design example generation is complete.
2. In the Intel Quartus Prime Pro Edition software, open the Intel Quartus Prime Pro Edition project <*design_example_dir*>/synthesis/quartus/ecpri_ed.qpf.
3. On the Processing menu, click **Start Compilation**.
4. After successful compilation, reports for timing and for resource utilization are available in your Intel Quartus Prime Pro Edition session. Go to **Processing ➤ Compilation Report** to view the detailed report on compilation.

Related Information

Block-Based Design Flows

1.6. Compiling and Configuring the Design Example in Hardware

To compile the hardware design example and configure it on your Intel device, follow these steps:

1. Ensure hardware design example generation is complete.
2. In the Intel Quartus Prime Pro Edition software, open the Intel Quartus Prime project <*design_example_dir*>/synthesis/quartus/ecpri_ed.qpf.
3. On the Processing menu, click **Start Compilation**.
4. After successful compilation, a `.sof` file is available in <*design_example_dir*>/synthesis/quartus/output_files directory. Follow these steps to program the hardware design example on the Intel FPGA device:
   a. Connect Development Kit to the host computer.
   b. Launch the Clock Control application, which is part of the development kit, and set the new frequencies for the design example. Below is the frequency setting in the Clock Control application:

- If you are targeting your design on Intel Stratix 10 GX SI Development Kit:
  - U5, OUT8- 100 MHz
  - U6, OUT3- 322.265625 MHz
  - U6, OUT4 and OUT5- 307.2 MHz
- If you are targeting your design on Intel Stratix 10 TX SI Development Kit:
  - U1, CLK4- 322.265625 MHz (For 25G data rate)
  - U6- 156.25 MHz (For 10G data rate)
  - U3, OUT3- 100 MHz
  - U3, OUT8- 153.6 MHz
- If you are targeting your design on Intel Agilex F-Series Transceiver-SoC Development Kit:
  - U37, CLK1A- 100 MHz
  - U34, CLK0P- 156.25 MHz
  - U38, OUT2_P- 153.6 MHz
- If you are targeting your design on Intel Arria 10 GX SI Development Kit:
  - U52, CLK0- 156.25 MHz
  - U52, CLK1- 250 MHz
  - U52, CLK3- 125 MHz
  - Y5- 307.2 MHz
  - Y6- 322.265625 MHz

c. On the **Tools** menu, click **Programmer**.

d. In the Programmer, click **Hardware Setup**.

e. Select a programming device.

f. Select and add the Development Kit to which your Intel Quartus Prime Pro Edition session can connect.

g. Ensure that **Mode** is set to **JTAG**.

h. Select the device and click **Add Device**. The Programmer displays a block diagram of the connections between the devices on your board.

i. Load the `.sof` file to your respective Intel FPGA device.

j. Load the Executable and Linking format (`.elf`) file to your Intel Stratix 10 or Intel Agilex device if you plan to perform the dynamic reconfiguration (DR) to switch the data rate between 25G and 10G. Follow the instructions from the Generating and Downloading the Executable and Linking Format (.elf) Programming File on page 34 to generate the `.elf` file.

k. In the row with your `.sof`, check the **Program/Configure** box for the `.sof` file.

l. Click **Start**.

Related Information

- Block-Based Design Flows

- Programming Intel FPGA Devices
- Analyzing and Debugging Designs with System Console
- Intel Agilex F-Series Transceiver-SoC Development Kit User Guide
- Intel Stratix 10 GX Transceiver Signal Integrity Development Kit User Guide
- Intel Stratix 10 TX Transceiver Signal Integrity Development Kit User Guide
- Intel Arria 10 GX Transceiver Signal Integrity Development Kit User Guide

1.7. Testing the eCPRI Intel FPGA IP Design Example

After you compile the eCPRI Intel FPGA IP core design example and configure it on your Intel FPGA 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. After the hardware design example is configured on the Intel device, in the Intel Quartus Prime Pro Edition software, on the **Tools** menu, click **System Debugging Tools ➤ System Console**.
2. In the Tcl Console pane, change directory to `<design_example_dir>/synthesis/quartus/hardware_test` and type the following command to open a connection to the JTAG master and start the test:
   - `source ecpri_agilex.tcl` for Intel Agilex designs
   - `source ecpri_s10.tcl` for Intel Stratix 10 designs
   - `source ecpri_a10.tcl` for Intel Arria 10 designs
3. For your Intel Stratix 10 or Intel Agilex E-tile device variations, you must perform either an internal or external loopback command once after you program the `.sof` file:
   a. Modify TEST_MODE variable in the `flow.c` file to select the loopback mode:

TEST_MODE	Action
0	Serial loopback enable for simulation only
1	Serial loopback enable for hardware only
2	Serial loopback and calibration
3	Calibration only

   You must recompile and regenerate the NIOS II software whenever you change the `flow.c` file.

   b. Regenerate the `.elf` file and program to the board one more time and reprogram the `.sof` file.
4. Test the design operation through the commands supported in the system console script. The system console script provides useful commands for reading statistics and features enabling in the design.

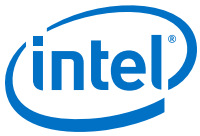


Table 4. System Console Script Commands

Command	Description
loop_on	Enables TX to RX internal serial loopback. Use for Intel Stratix 10 H-tile and Intel Arria 10 devices only.
loop_off	Disables TX to RX internal serial loopback. Use for Intel Stratix 10 H-tile and Intel Arria 10 devices only.
link_init_int_lpbk	Enables TX to RX internal serial loopback within the transceiver and performs the transceiver calibration flow. Applicable to the Intel Stratix 10 E-tile and Intel Agilex E-tile designs only.
link_init_ext_lpbk	Enables TX to RX external loopback and performs the transceiver calibration flow. Applicable to the Intel Stratix 10 E-tile and Intel Agilex E-tile designs only.
traffic_gen_disable	Disables the traffic generator and checker.
chkmac_stats	Displays the statistics for the Ethernet MAC.
read_test_statistics	Display the error statistics for traffic generator and checkers.
ext_continuous_mode_en	Resets the entire design system, and enables the traffic generator to generate continuous traffic packets.
dr_25g_to_10g_etile	Switches the data rate of the Ethernet MAC from 25G to 10G. Use for the Intel Stratix 10 E-tile and Intel Agilex E-tile devices only.
dr_25g_to_10g_htile	Switches the data rate of the Ethernet MAC from 25G to 10G. Use for H-tile devices only
dr_10g_to_25g_etile	Switches the data rate of the Ethernet MAC from 10G to 25G. Use for the Intel Stratix 10 E-tile and Intel Agilex E-tile devices only.
dr_25g_to_10g_htile	Switches the data rate of the Ethernet MAC from 10G to 25G. Use for H-tile devices only.

The following sample output illustrates a successful test run:

```
================================================================================
==============
                        STATISTICS FOR BASE 0x20000000 (Tx)
================================================================================
==============
Fragmented Frames               : 0
Jabbered Frames                 : 0
Any Size with FCS Err Frame     : 0
Right Size with FCS Err Fra     : 0
Multicast data  Err Frames      : 0
Broadcast data Err  Frames      : 0
Unicast data Err  Frames        : 0
Multicast control  Err Frame    : 0
Broadcast control Err  Frame    : 0
Unicast control Err  Frames     : 0
Pause control Err  Frames       : 0


64 Byte Frames                  : 3072
65 - 127 Byte Frames            : 259867903
128 - 255 Byte Frames           : 25986817
256 - 511 Byte Frames           : 0
512 - 1023 Byte Frames          : 0
1024 - 1518 Byte Frames         : 0
1519 - MAX Byte Frames          : 0
> MAX Byte Frames               : 0
```

```
Tx Frame Starts                  : 285857794
Multicast data  OK  Frame        : 285857792
Broadcast data OK   Frame        : 0
Unicast data OK   Frames         : 0
Multicast Control Frames         : 0
Broadcast Control Frames         : 0
Unicast Control Frames           : 0
Pause Control Frames             : 0
Payload Octets OK                : 23648123502
Frame Octets OK                  : 28897511044

================================================================================
==============
                        STATISTICS FOR BASE 0x20200000 (Rx)

================================================================================
==============
Rx Maximum Frame Length          : 9600
Fragmented Frames                : 0
Jabbered Frames                  : 0
Any Size with FCS Err Frame      : 0
Right Size with FCS Err Fra      : 0
Multicast data  Err Frames       : 0
Broadcast data Err  Frames       : 0
Unicast data Err  Frames         : 0
Multicast control  Err Frame     : 0
Broadcast control Err  Frame     : 0
Unicast control Err  Frames      : 0
Pause control Err  Frames        : 0
64 Byte Frames                   : 261
65 - 127 Byte Frames             : 40699313
128 - 255 Byte Frames            : 4069979
256 - 511 Byte Frames            : 0
512 - 1023 Byte Frames           : 0
1024 - 1518 Byte Frames          : 0
1519 - MAX Byte Frames           : 0
> MAX Byte Frames                : 0
Rx Frame Starts                  : 0
Multicast data  OK  Frame        : 44769554
Broadcast data OK   Frame        : 0
Unicast data OK   Frames         : 0
Multicast Control Frames         : 0
Broadcast Control Frames         : 0
Unicast Control Frames           : 0
Pause Control Frames             : 0
Payload Octets OK                : 3703654736
Frame Octets OK                  : 4525786704
eCPRI Error Interrupt: 0
TX SOP Count: 57014964
TX EOP Count: 57147501
RX SOP Count: 57279074
RX EOP Count: 57411153
Checker Errors: 0
Checker Error Counts: 0
EXT PTP TX SOP Count: 256
EXT PTP TX EOP Count: 256
EXT MISC TX SOP Count: 5807054
EXT MISC TX EOP Count: 5820433
EXT RX SOP Count: 5834010
EXT RX EOP Count: 5848179
eCPRI EXT RX AVST Error: 0x00000000
EXT Checker Errors: 0
EXT Checker Error Counts: 0
```

The following is the sample output for the 25G to 10G DR test run:

```
25G transaction finished successfully

DR Successful 25G -> 10G
```

```
 RX PHY Register Access: Checking Clock Frequencies (KHz)

    TXCLK         :15626  (KHZ)
    RXCLK         :15626  (KHZ)
 RX PHY Status Polling

 Rx Frequency Lock Status    0x00000001

 Mac Clock in OK Condition? 0x00000007

 Rx Frame Error        0x00000000

 Rx PHY Fully Aligned?     0x00000001

Read Checker & Ethernet MAC Statistics before enable traffic


==============================================================================
==============
                        STATISTICS FOR BASE 0x20200000 (Tx)

==============================================================================
==============
Fragmented Frames               : 0
Jabbered Frames                 : 0
Any Size with FCS Err Frame     : 0
Right Size with FCS Err Fra     : 0
Multicast data  Err Frames      : 0
Broadcast data Err  Frames      : 0
Unicast data Err  Frames        : 0
Multicast control  Err Frame    : 0
Broadcast control Err  Frame    : 0
Unicast control Err  Frames     : 0
Pause control Err  Frames       : 0
64 Byte Frames                  : 12672
65 - 127 Byte Frames            : 1348759131
128 - 255 Byte Frames           : 134876106
256 - 511 Byte Frames           : 0
512 - 1023 Byte Frames          : 0
1024 - 1518 Byte Frames         : 0
1519 - MAX Byte Frames          : 0
> MAX Byte Frames               : 0
Tx Frame Starts                 : 0
Multicast data  OK  Frame       : 1483647909
Broadcast data OK   Frame       : 0
Unicast data OK   Frames        : 0
Multicast Control Frames        : 0
Broadcast Control Frames        : 0
Unicast Control Frames          : 0
Pause Control Frames            : 0
Payload Octets OK               : 122737685052
Frame Octets OK                 : 149982851838

==============================================================================
==============
                        STATISTICS FOR BASE 0x20200000 (Rx)

==============================================================================
==============
Fragmented Frames               : 0
Jabbered Frames                 : 0
Any Size with FCS Err Frame     : 0
Right Size with FCS Err Fra     : 0
Multicast data  Err Frames      : 0
Broadcast data Err  Frames      : 0
Unicast data Err  Frames        : 0
Multicast control  Err Frame    : 0
Broadcast control Err  Frame    : 0
Unicast control Err  Frames     : 0
Pause control Err  Frames       : 0
64 Byte Frames                  : 12672
```

```
65 - 127 Byte Frames              : 1348759125
128 - 255 Byte Frames             : 134876105
256 - 511 Byte Frames             : 0
512 - 1023 Byte Frames            : 0
1024 - 1518 Byte Frames           : 0
1519 - MAX Byte Frames            : 0
> MAX Byte Frames                 : 0
Rx Frame Starts                   : 0
Multicast data  OK  Frame         : 1483647902
Broadcast data OK   Frame         : 0
Unicast data OK   Frames          : 0
Multicast Control Frames          : 0
Broadcast Control Frames          : 0
Unicast Control Frames            : 0
Pause Control Frames              : 0
Payload Octets OK                 : 122737684462
Frame Octets OK                   : 149982851118
TX SOP Count: 0
TX EOP Count: 0
RX SOP Count: 0
RX EOP Count: 0
Checker Errors: 0
Checker Error Counts: 0
EXT PTP TX SOP Count: 0
EXT PTP TX EOP Count: 0
EXT MISC TX SOP Count: 0
EXT MISC TX EOP Count: 0
EXT RX SOP Count: 0
EXT RX EOP Count: 0
EXT Checker Errors: 0
EXT Checker Error Counts: 0
Configure MAC Destination Addresses for Ethernet Frame
MAC Destination Addresses 0x33445566
MAC Destination Addresses 0x00007788
MAC Destination Addresses 0x11223344
MAC Destination Addresses 0x00005566
MAC Destination Addresses 0x22334455
MAC Destination Addresses 0x00006677
MAC Destination Addresses 0x44556677
MAC Destination Addresses 0x00008899
MAC Destination Addresses 0x66778899
MAC Destination Addresses 0x0000aabb
MAC Destination Addresses 0x778899aa
MAC Destination Addresses 0x0000bbcc
MAC Destination Addresses 0x8899aabb
MAC Destination Addresses 0x0000ccdd
MAC Destination Addresses 0x99aabbcc
MAC Destination Addresses 0x0000ddee
Configure MAC Source Addresses for Ethernet Frame
MAC Source Addresses 0x33445566
MAC Source Addresses 0x00007788
Enable External Continuous Packet Mode
Read Checker & Ethernet MAC Statistics after enable traffic


================================================================================
==============
                        STATISTICS FOR BASE 0x20200000 (Tx)

================================================================================
==============
Fragmented Frames                 : 0
Jabbered Frames                   : 0
Any Size with FCS Err Frame       : 0
Right Size with FCS Err Fra       : 0
Multicast data  Err Frames        : 0
Broadcast data Err  Frames        : 0
Unicast data Err  Frames          : 0
Multicast control  Err Frame      : 0
Broadcast control Err  Frame      : 0
Unicast control Err  Frames       : 0
```

```
Pause control Err  Frames        : 0
64 Byte Frames                   : 12800
65 - 127 Byte Frames             : 1348877612
128 - 255 Byte Frames            : 134887952
256 - 511 Byte Frames            : 0
512 - 1023 Byte Frames           : 0
1024 - 1518 Byte Frames          : 0
1519 - MAX Byte Frames           : 0
> MAX Byte Frames                : 0
Tx Frame Starts                  : 0
Multicast data  OK  Frame        : 1483778363
Broadcast data OK   Frame        : 0
Unicast data OK   Frames         : 0
Multicast Control Frames         : 0
Broadcast Control Frames         : 0
Unicast Control Frames           : 0
Pause Control Frames             : 0
Payload Octets OK                : 122748472560
Frame Octets OK                  : 149996034742

================================================================================
==============
                        STATISTICS FOR BASE 0x20200000 (Rx)

================================================================================
==============
Fragmented Frames                : 0
Jabbered Frames                  : 0
Any Size with FCS Err Frame      : 0
Right Size with FCS Err Fra      : 0
Multicast data  Err Frames       : 0
Broadcast data Err  Frames       : 0
Unicast data Err  Frames         : 0
Multicast control  Err Frame     : 0
Broadcast control Err  Frame     : 0
Unicast control Err  Frames      : 0
Pause control Err  Frames        : 0
64 Byte Frames                   : 12928
65 - 127 Byte Frames             : 1358325058
128 - 255 Byte Frames            : 135832699
256 - 511 Byte Frames            : 0
512 - 1023 Byte Frames           : 0
1024 - 1518 Byte Frames          : 0
1519 - MAX Byte Frames           : 0
> MAX Byte Frames                : 0
Rx Frame Starts                  : 0
Multicast data  OK  Frame        : 1494170686
Broadcast data OK   Frame        : 0
Unicast data OK   Frames         : 0
Multicast Control Frames         : 0
Broadcast Control Frames         : 0
Unicast Control Frames           : 0
Pause Control Frames             : 0
Payload Octets OK                : 123608196298
Frame Octets OK                  : 151046599442
TX SOP Count: 13057508
TX EOP Count: 13319057
RX SOP Count: 13571663
RX EOP Count: 13826107
Checker Errors: 0
Checker Error Counts: 0
EXT PTP TX SOP Count: 128
EXT PTP TX EOP Count: 128
EXT MISC TX SOP Count: 1514619
EXT MISC TX EOP Count: 1540422
EXT RX SOP Count: 1566619
EXT RX EOP Count: 1592754
EXT Checker Errors: 0
EXT Checker Error Counts: 0
result: 0
```

2. Design Example Description

The design example demonstrates the basic functionality of the eCPRI IP core. You can generate the design from the Example Design tab in the eCPRI IP parameter editor.

2.1. Features

- Internal TX and RX serial loopback mode
- Automatically generates fixed size packets
- Basic packet checking capabilities
- Ability to use System Console to test the design and reset the design for re-testing purpose

2.2. Hardware Design Example

Figure 5. Block Diagram for Intel Agilex Designs

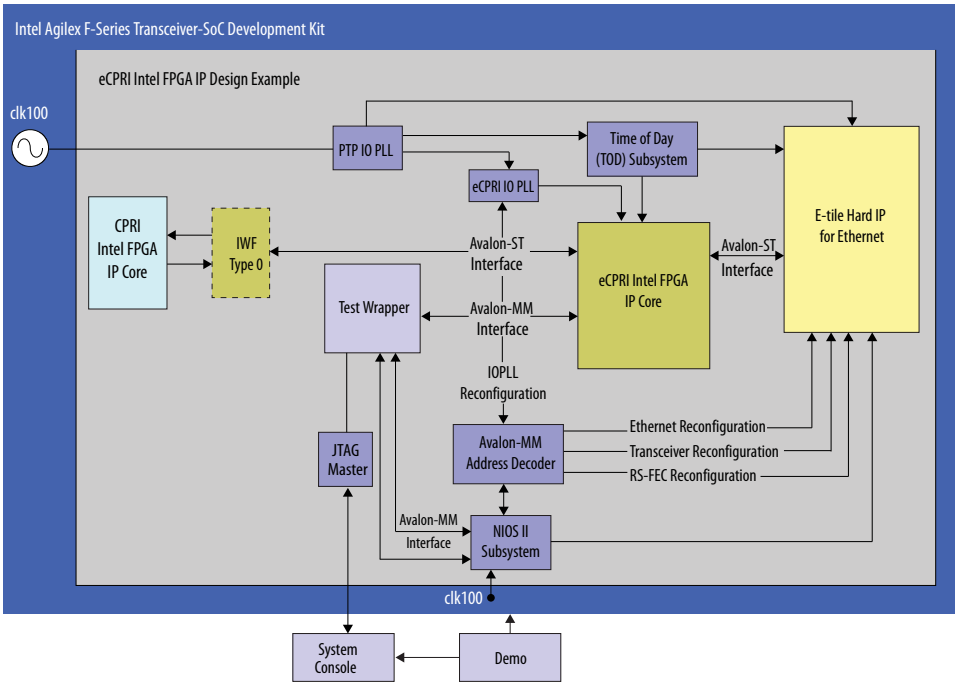


Figure 6. Block Diagram for Intel Stratix 10 Designs

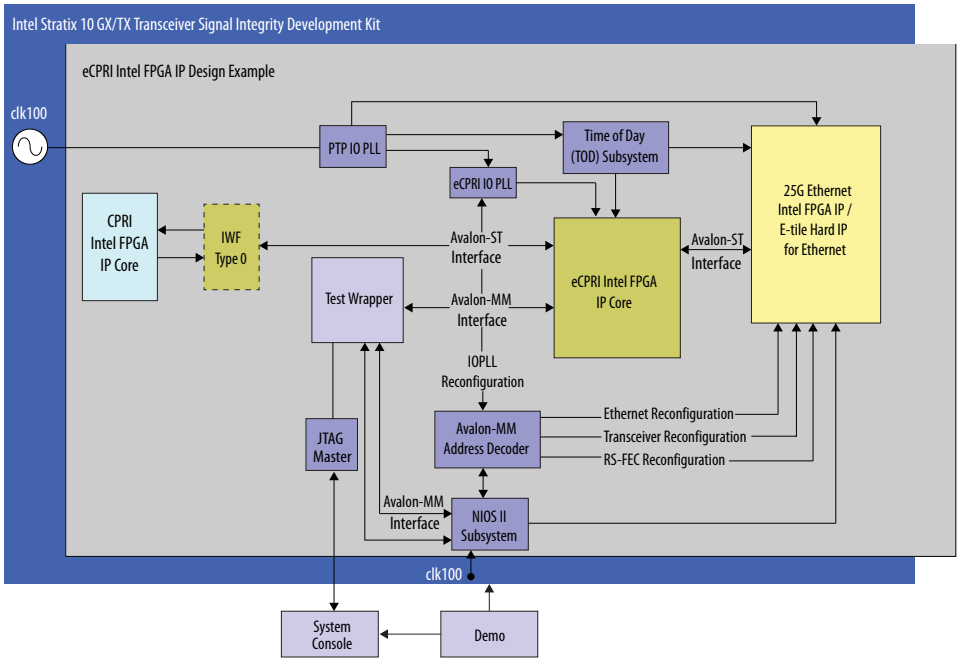
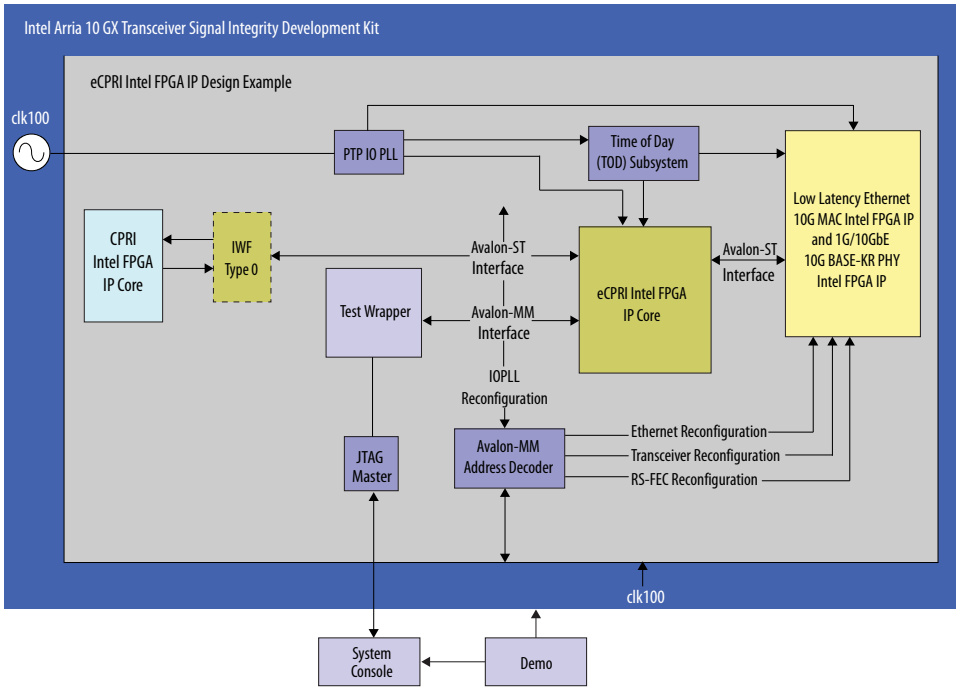


Figure 7. Block Diagram for Intel Arria 10 Designs



The eCPRI Intel FPGA IP core hardware design example includes the following components:

eCPRI Intel FPGA IP

Accepts data from the traffic generators instantiated within the test wrapper and prioritize the data for transmission to the Ethernet IP.

Ethernet IP

- Ethernet Hard IP for Ethernet (Intel Stratix 10 or Intel Agilex E-tile designs)
- 25G Ethernet Intel Stratix 10 IP (Intel Stratix 10 H-tile designs)
- Low Latency Ethernet 10G MAC IP and 1G/10GbE and 10GBASE-KR PHY IP (Intel Arria 10 designs)

Precision Time Protocol (PTP) IO PLL

For Intel Stratix 10 H-tile designs—Instantiated to generate the latency measurement input reference clock for the Ethernet IP and sampling clock for Time of Day (TOD) subsystem. For 25G Ethernet Intel Stratix 10 FPGA IP with the IEEE 1588v2 feature, Intel recommends you to set the frequency of this clock to 156.25 MHz. Refer to the *25G Ethernet Intel Stratix 10 FPGA IP User Guide* and *Intel Stratix 10 H-tile Transceiver PHY User Guide* for more information. The PTP IOPLL also generates the reference clock for the eCPRI IO PLL in the cascading manner.

For Intel Arria 10 designs—Instantiated to generate the 312.5 MHz and 156.25 MHz clock inputs for the Low Latency Ethernet 10G MAC IP and 1G/10GbE, 10GBASE-KR PHY IP, and eCPRI IP .

eCPRI IO PLL

Generates core clock output of 390.625 MHz for the TX and RX path of the eCPRI IP, and traffic components.

Note: This block is only present in the design example generated for Intel Stratix 10 and Intel Agilex devices.

IWF Type 0

Converts CPRI MAC data packet into eCPRI packet. This block sits between the CPRI MAC and eCPRI IP as shown in block diagram above. The conversion works only for message type 0,2, 6, and 7.

Note: The current version of the eCPRI Intel FPGA IP only supports IWF type 0.

When you generate the design example with **Interworking Function (IWF) Support** parameter turned off, the packet traffic flows directly from the test wrapper module to the Avalon-ST source/sink interface and external source/sink interface of the eCPRI IP.

When you generate the design example with **Interworking Function (IWF) Support** parameter turned on, the packet traffic flows to the IWF Avalon-ST sink interface from the test wrapper module first, and coming out from IWF Avalon-ST source interface to the eCPRI Avalon-ST source/sink interface.

CPRI MAC

Provides the CPRI part of the layer 1 and full layer 2 protocols for the transfer of user plane, C&M, and synchronization information between REC and RE as well as between two RE,

CPRI PHY

Provides the remaining part of CPRI layer 1 protocol for line coding, bit error correction/detection, and etc.

Note: The CPRI MAC and CPRI PHY IP instantiated in this design example are configured to be running at single CPRI line rate 9.8 Gbps only. The design example does not support line rate auto-negotiation in the current release.

Test Wrapper

Consists of traffic generators and checkers which generates different set of data packets to the Avalon Streaming (Avalon-ST) interfaces of the eCPRI IP as below:

- eCPRI packets to the Avalon-ST source/sink interfaces (IWF feature disabled):
  - Only supports message type 2.
  - Back-to-back mode generation with incremental pattern mode generation and payload size of 72 bytes for each packet.
  - Configurable via CSR to run in either non-continuous or continuous mode.
  - TX/RX packet statistic status available to access via CSR.
- eCPRI packets to the Avalon-ST source/sink interfaces (IWF feature enabled):
  - Only supports message type 0 in current release.
  - Incremental pattern mode generation with interpacket gap generation and payload size of 240 bytes for each packet.
  - Configurable via CSR to run in either non-continuous or continuous mode.
  - TX/RX packet statistic status available to access via CSR.
- Precision Time Protocol (1588 PTP) packet and non-PTP miscellaneous packets to the External source/sink interfaces:
  - Static Ethernet header generation with pre-defined parameters: Ethertype-0x88F7, Message type- Opcode 0 (Sync), and PTP version-0.
  - Pre-defined pattern mode generation with interpacket gap of 2 cycles and payload size of 57 bytes for each packet.
  - 128 packets are generated in the period of every one second.
  - Configurable via CSR to run in either non-continuous or continuous mode.
  - TX/RX packet statistic status available to access via CSR.
- External non-PTP miscellaneous packets:
  - Static Ethernet Header generation with pre-defined parameter, Ethertype-0x8100 (non-PTP).
  - PRBS pattern mode generation with interpacket gap of 2 cycles and payload size of 128 bytes for each packet.
  - Configurable via CSR to run in either non-continuous or continuous mode.
  - TX/RX packet statistic status available to access via CSR.

Time of Day (TOD) subsystem

Contains two IEEE 1588 TOD modules for both TX and RX, and one IEEE 1588 TOD Synchronizer module generated by Intel Quartus Prime software.

Nios® II Subsystem

Consists of Avalon-MM bridge that allows Avalon-MM data arbitration between Nios® II processor, test wrapper, and Avalon®-MM address decoder blocks.

Nios II is responsible to perform data rate switching based on the output from test wrapper's `rate_switch` register value. This block programs the necessary register once it receives command from the test wrapper.

Note: This block is not present in the design example generated for Intel Arria 10 devices.

System Console

Provides a user-friendly interface for you to do first-level debugging and monitor status of the IP, and the traffic generators and checkers.

Demo Control

This module consists of reset synchronizer modules, and In-system Source and Probe (ISSP) modules for design system debugging and initialization process.

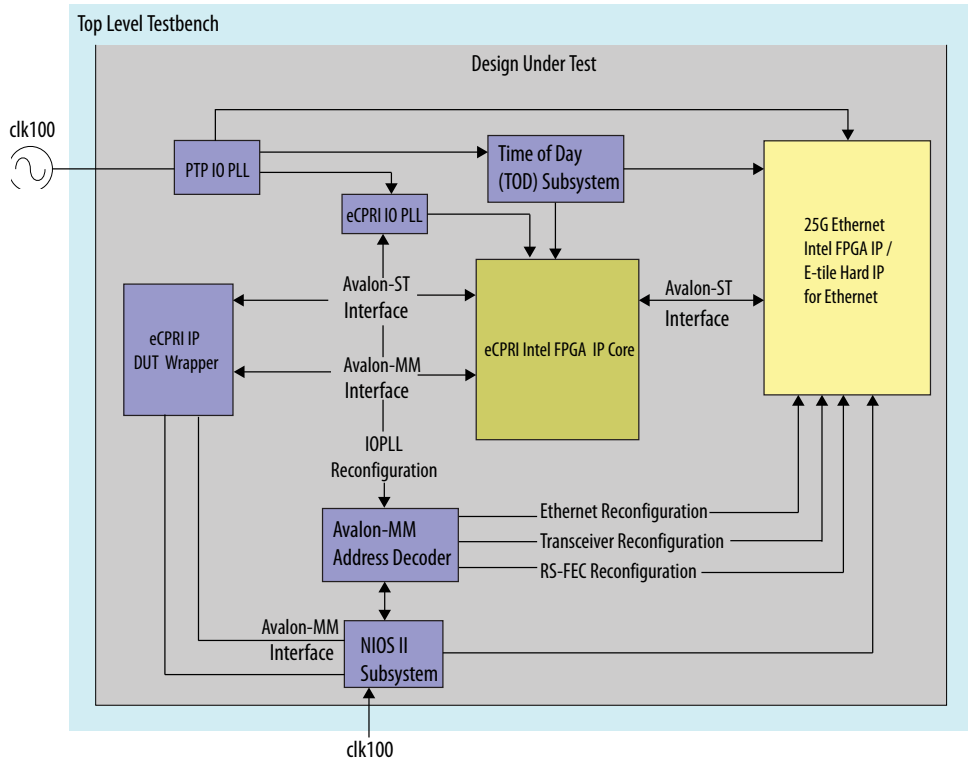
Related Information

- 25G Ethernet Intel Stratix 10 FPGA IP User Guide
- E-tile Hard IP User Guide
- eCPRI Intel FPGA IP User Guide
- 25G Ethernet Intel Stratix 10 FPGA IP Design Example User Guide
- E-tile Hard IP for Intel Stratix 10 Design Examples User Guide
- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- E-Tile Transceiver PHY User Guide
- Intel Stratix 10 10GBASE-KR PHY IP User Guide
- E-tile Hard IP Intel Agilex Design Example User Guide

2.3. Simulation Design Example

The eCPRI design example generates a simulation testbench and simulation files that instantiates the eCPRI Intel FPGA IP core when you select the **Simulation** or **Synthesis & Simulation** option.

Figure 8. eCPRI Intel FPGA IP Simulation Block Diagram



In this design example, the simulation testbench provides basic functionality such as startup and wait for lock, transmit and receive packets.

The successful test run displays output confirming the following behavior:

1. The client logic resets the IP core.
2. The client logic waits for the RX datapath alignment.
3. The client logic transmits packets on the Avalon-ST interface.
4. Receive and checks for the content and correctness of the packets.
5. Display "Test PASSED" message.

2.4. Interface Signals

Table 5. Design Example Interface Signals

Signal	Direction	Description
clk_ref	Input	Reference clock for the Ethernet MAC.

continued...

Signal	Direction	Description
		• For Intel Stratix 10 E-tile and Intel Agilex E-tile designs, 156.25 MHz clock input for the E-tile Ethernet Hard IP core. Connect to `i_clk_ref[0]` in the Ethernet Hard IP. • For Intel Stratix 10 H-tile designs, a 322.2625 MHz clock input for the Transceiver ATX PLL and 25G Ethernet IP. Connect to `pll_refclk0[0]` in the Transceiver ATX PLL and `clk_ref[0]` in 25G Ethernet IP. • For Intel Arria 10 designs, a 322.265625 MHz clock input for the Transceiver ATX PLL and 1G/10GbE and 10GBase-KR PHY IP. Connect to `pll_refclk0[0]` in the Transceiver ATX PLL and `rx_cdr_ref_clk_10g[0]` in the 1G/10GbE and 10G BASE-KR PHY IP.
`tod_sync_sampling_clk`	Input	For Intel Arria 10 designs, a 250 MHz clock input for TOD subsystem.
`clk100`	Input	Management clock. This clock is used to generate `latency_clk` for PTP. Drive at 100 MHz.
`mgmt_reset_n`	Input	Reset signal for Nios II system.
`tx_serial`	Output	TX serial pin.
`rx_serial`	Input	RX serial pin.
`iwf_cpri_ehip_ref_clk`	Input	E-tile CPRI PHY reference clock input. This clock is only present in Intel Stratix 10 E-tile and Intel Agilex E-tile designs. Drive at 153.6 MHz for 9.8 Gbps CPRI line rate.
`iwf_cpri_pll_refclk0`	Output	CPRI TX PLL reference clock. • For Intel Stratix 10 H-tile designs: Drive at 307.2 MHz for CPRI data rate 9.8 Gbps. • For Intel Stratix 10 E-tile and Intel Agilex E-tile designs: Drive at 156.25 MHz for CPRI data rate 9.8 Gbps.
`iwf_cpri_xcvr_cdr_refclk`	Output	CPRI receiver CDR reference clock. This clock is only present in Intel Stratix 10 H-tile designs. Drive at 307.2 MHz for 9.8 Gbps CPRI line rate.
`iwf_cpri_xcvr_txdataout`	Output	CPRI transmit serial data.
`iwf_cpri_xcvr_rxdatain`	Output	CPRI receiver serial data.
`cpri_gmii_clk`	Input	CPRI GMII 125 MHz input clock.

Related Information

PHY Interface Signals
Lists the PHY interface signals of the 25G Ethernet Intel FPGA IP.

2.5. Design Example Register Map

Below is the register mapping for the eCPRI IP core design example:

Table 6. eCPRI Intel FPGA IP Design Example Register Mapping

Address	Register
0x20100000 – 0x201FFFFF[2]	IOPLL Re-configuration Register.
0x20200000 – 0x203FFFFF	Ethernet MAC Avalon-MM Register
0x20400000 – 0x205FFFFF	Ethernet MAC Native PHY Avalon-MM Register
0x20600000 – 0x207FFFFF[2]	Native PHY RS-FEC Avalon-MM Register.
0x40000000 – 0x5FFFFFFF	eCPRI IP Avalon-MM Register
0x80000000 – 0x9FFFFFFF	Ethernet Design Test Generator/Verifier Avalon-MM Register

Table 7. Nios II Register Mapping

The registers in below table are only available in the design example generated for Intel Stratix 10 or Intel Agilex devices.

Address	Register
0x00100000 – 0x001FFFFF	IOPLL Re-configuration Register
0x00200000 – 0x003FFFFF	Ethernet MAC Avalon-MM Register
0x00400000 – 0x005FFFFF	Ethernet MAC Native PHY Avalon-MM Register
0x00600000 – 0x007FFFFF	Native PHY RS-FEC Avalon-MM Register

Note: You can access the Ethernet MAC and Ethernet MAC Native PHY AVMM registers using word offset instead of byte offset.

For detailed information on Ethernet MAC, Ethernet MAC Native PHY, and eCPRI IP core register maps, refer to the respective user guides.

Table 8. eCPRI Intel FPGA IP Hardware Design Example Register Map

Word Offset	Register Type	Access Type
0x0001	Start send data	RW
0x0002	Continuous packet enable	RW
0x0003	Clear error	RW
0x0004	Checker errors	RO
0x0005	TX start of packet (SOP) count	RO
0x0006	TX end of packet (EOP) count	RO
0x0007	RX SOP count	RO
0x0008	RX EOP count	RO
0x0009	Total error count	RO
0x000A	External packets error	RO
0x000B	External PTP packets TX SOP count	RO
0x000C	External PTP packets TX EOP count	RO
0x000D	External misc packets TX SOP count	RO
		continued...

(2) Only available in design example generated for Intel Stratix 10 and Intel Agilex devices.

Word Offset	Register Type	Access Type
0x000E	External misc packets TX EOP count	RO
0x000F	External RX packets SOP count	RO
0x0010	External RX packets EOP count	RO
0x0011	External error count	RO
0x0012	Rate switch: • Bit [7]- Indicates tile: — 1'b0: H-tile — 1'b1: E-tile • Bit [6:4]- Indicates Ethernet data rate switching: — 3'b000: 25G to 10G — 3'b001: 10G to 25G • Bit [0]- Switch rate enable. It's required to set this bit 0 and poll until bit 0 is clear for the rate switching. Only available in Intel Stratix 10 and Intel Agilex designs.	RW
0x0013	Rate switch done. Bit [1] indicates rate switching done. Only available in Intel Stratix 10 and Intel Agilex designs.	RO
0x0020	System configuration status: • Bit [31]: Reserved • Bit [30]: IWF_EN • Bit [29]: STARTUP_SEQ_EN • Bit [4]: EXT_PACKET_EN • Bit [0[: System ready Only present in eCPRI design example generated with IWF feature enabled.	RO
0x0021	CPRI negotiation complete: • Bit [15:12]: Bit rate complete • Bit [11:8]: Fast C&M complete • Bit [7:4]: Protocol complete • Bit [3:0]: Fast VSS complete Only present in eCPRI design example with IWF feature enabled.	RO
0x0025	eCPRI error interrupt. Bit [0] indicates the interrupt.	RO
0x002A	External AVST RX error status: • Bit [0]: Malformed packet. The packet is terminated with a non-terminate control character. When this bit is asserted, bit [1] is also asserted. • Bit [1]: Indicates CRC Error. The computed CRC value differs from the received CRC. • Bit [2]: Undersized frame – The frame size is less than 64 bytes. Frame size = header size + payload size. • Bit [3]: Oversized frame. The frame size is greater than the value specified in the RXMAC_SIZE_CONFIG register. • Bit [4]: Payload length error. If the length field is less than 1535 bytes (0x600 bytes), the received payload length is less than what is advertised in the payload length field.	RO

Related Information

- Control, Status, and Statistics Register Descriptions
  Register information for the 25G Ethernet Stratix 10 FPGA IP
- Reconfiguration and Status Register Descriptions
  Register information for the E-tile Hard IP for Ethernet

- Registers
  Register information for the eCPRI Intel FPGA IP

3. eCPRI Intel FPGA IP Design Example User Guide Archives

IP versions are the same as the Intel Quartus Prime Design Suite software versions up to v19.1. From Intel Quartus Prime Design Suite software version 19.2 or later, IP cores have a new IP versioning scheme.

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
20.3	1.2.0	eCPRI Intel FPGA IP Design Example User Guide
20.1	1.1.0	eCPRI Intel Stratix 10 FPGA IP Design Example User Guide
19.4	1.0.0	eCPRI Intel Stratix 10 FPGA IP Design Example User Guide

4. Document Revision History for eCPRI Intel FPGA IP Design Example User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2021.02.26	20.4	1.3.0	• Added support for the Intel Agilex E-tile devices.
2021.01.08	20.3	1.2.0	• Changed the document title from *eCPRI Intel Stratix 10 FPGA IP Design Example User Guide* to *eCPRI Intel FPGA IP Design Example User Guide*. • Added support for Intel Arria 10 designs. • The eCPRI IP design example is now available with interworking function (IWF) feature support. • Added a note to clarify that eCPRI design example with IWF feature is only available for 9.8 Gbps CPRI line bit rate. • Added conditions in section *Generating the Design* when generating the design example with **Interworking Function (IWF) Support** parameter enabled. • Added sample simulation test run output with IWF feature enabled in section *Simulating the Design Example Testbench*. • Added new section *Enabling Dynamic Reconfiguration to the Ethernet IP*. • Updated hardware test sample output in section *Testing the eCPRI Intel FPGA IP Design Example*. • Updated the *Figure: eCPRI Intel FPGA IP Hardware Design Examples High Level Block Diagram* to include IWF Type 0, CPRI MAC, and CPRI PHY blocks. • Updated *Table: Design Example Interface Signals* to include Intel Arria 10 device and IWF related signals. • Updated *Table: eCPRI Intel FPGA IP Hardware Design Example Register Map*.
2020.06.15	20.1	1.1.0	• Added support for 10G data rate. • `flow.c` file is now available with design example generation to select loopback mode. • Modified the sample output for simulation test run in section *Simulating the Design Example Testbench*. • Added frequency value for running 10G data rate design in section *Compiling and Configuring the Design Example in Hardware*. • Made following changes in section *Testing the eCPRI Intel FPGA IP Design Example*: — Added commands to switch data rate between 10G and 25G — Added sample output for data rate switching — Added TEST_MODE variable information to select loopback in E-tile device variations.

continued...

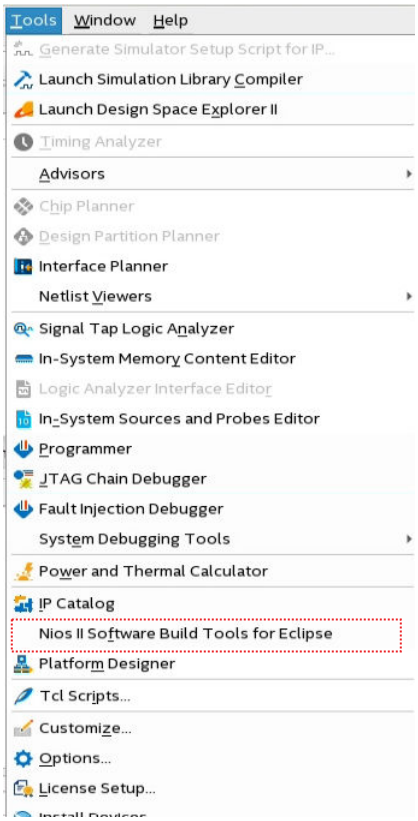
Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Modified *eCPRI Intel FPGA IP Hardware Design Examples High Level Block Diagram* to include new blocks. • Updated *Table: Design Example Interface Signals* to include new signal. • Updated *Design Example Register Map* section. • Added new appendix section:*Generating and Downloading the Executable and Linking Format (.elf) Programming File* .
2020.04.13	19.4	1.0.0	Initial release.

A. Generating and Downloading the Executable and Linking Format (`.elf`) Programming File

This section describes how to generate and download the `.elf` file to the board:

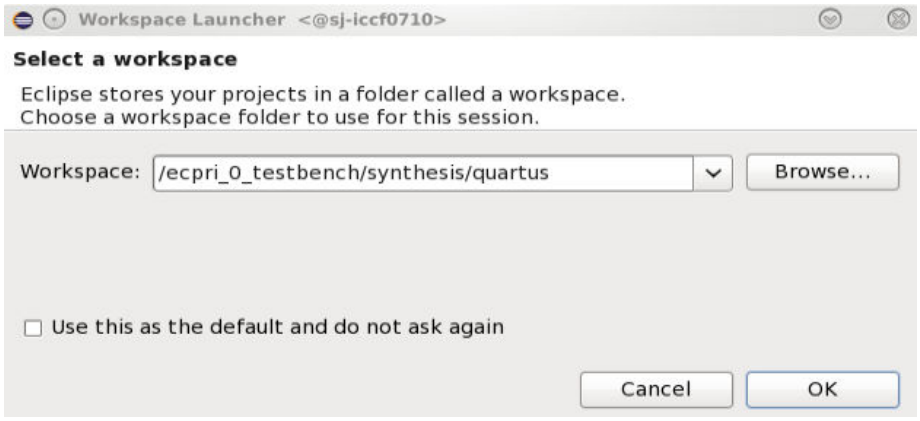
1. Change directory to `<design_example_dir>/synthesis/quatus`.
2. In the Intel Quartus Prime Pro Edition software, click **Open Project** and open `<design_example_dir>/synthesis/quartus/epri_ed.qpf`. Now select **Tools ➤ Nios II Software Build Tools for Eclipse**.

Figure 9. Nios II Software Build Tools for Eclipse



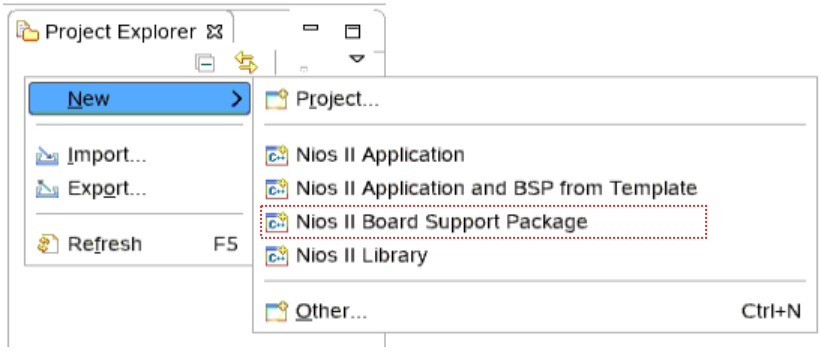
3. The **Workspace Launcher** window prompt appears. In the **Workspace** specify the path as `<design_example_dir>/synthesis/quatus` to store your Eclipse project. The new **Nios II - Eclipse** window appears.

Figure 10. Workspace Launcher Window



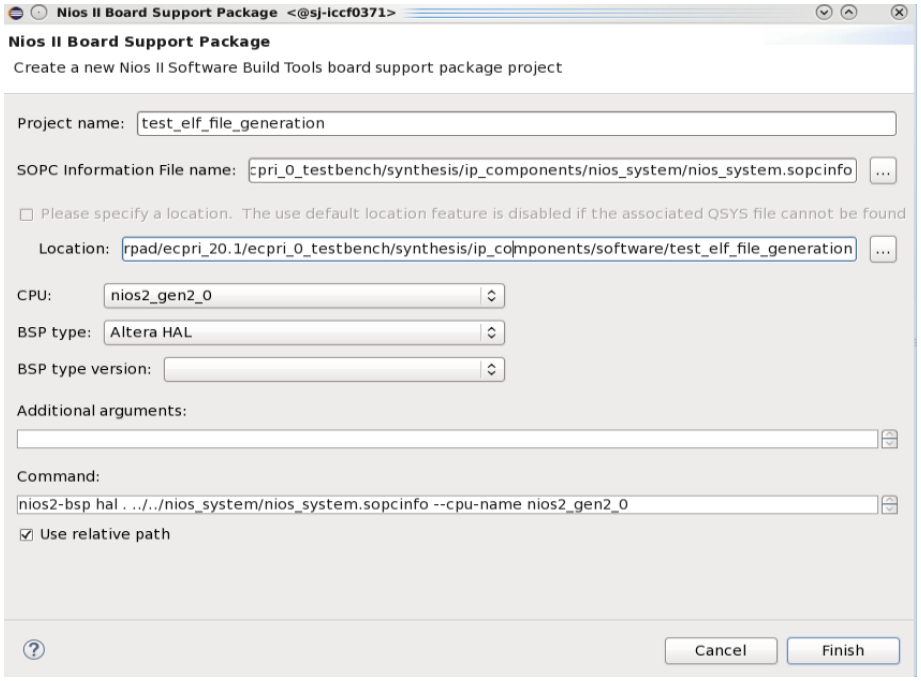
4. In the **Nios II - Eclipse** window, righ-click under **Project Explorer** tab, and select **New ➤ Nios II Board Support Package**. The new window appears.

Figure 11. Project Explorer Tab



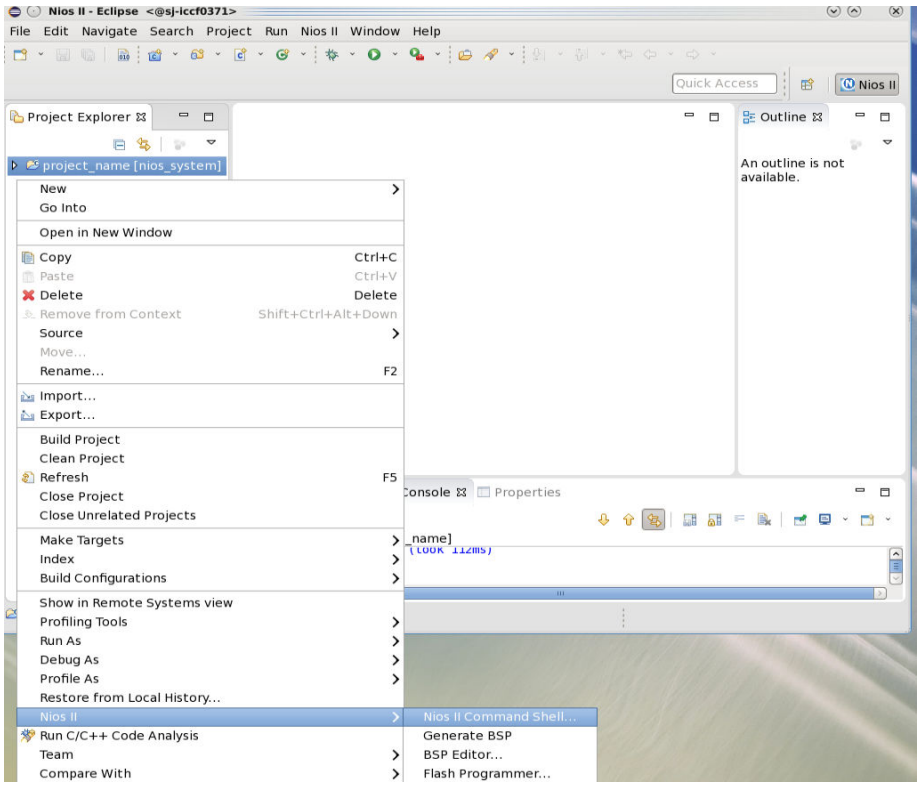
5. In the **Nios II Board Support Package** window:
   - In the **Project name** parameter, specify your desired project name.
   - In the **SOPC Information File name** parameter, browse to the location of `<design_example_dir>/synthesis/ip_components/nios_system/nios_system.sopcinfo` file. Click **Finish**.

Figure 12. Nios II Board Support Package Window



6. The newly created project appears under **Project Explorer** tab in **Nios II - Eclipse** window. Right-click under **Project Explorer** tab, and select **Nios II ➤ Nios II Command Shell**.

Figure 13. Project Explorer- Nios II Command Shell



7. In the **Nios II Command Shell**, type the three following commands:

```
nios2-bsp hal bsp ../../nios_system/nios_system.sopcinfo
```

```
nios2-app-generate-makefile --app-dir app --bsp-dir bsp --elf-name\
nios_system.elf --src-dir ../../../ed_fw
```

```
make --directory=app
```

8. The `.elf` file is generated in the following location: `<design_example_dir>/synthesis/ip_components/software/<desired_project_name>/app`.
9. Type the following command in the **Nios II Command Shell** to download the `.elf` to the board:
   - For Intel Stratix 10:

     ```
     nios2-download -g -r -c 1 -d 2 --accept-bad-sysid app/nios_system.elf
     ```

   - For Intel Agilex:

     ```
     nios2-download -g -r -c 1 -d 1 --accept-bad-sysid app/nios_system.elf
     ```