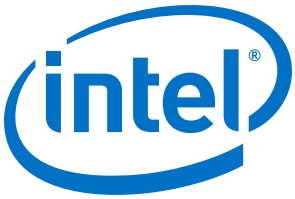


Interlaken (2nd Generation) Intel® Agilex™ FPGA IP Design Example User Guide

Updated for Intel® Quartus® Prime Design Suite: **19.3**

IP Version: **19.2.1**

Subscribe
Send Feedback

UG-20239 | 2019.09.30
Latest document on the web: **PDF** | **HTML**

Contents

1. Quick Start Guide

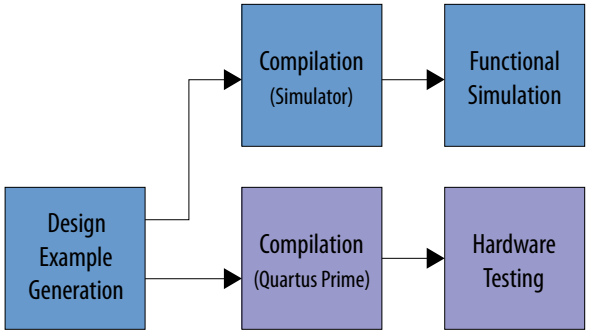
The Interlaken (2nd Generation) FPGA IP core provides a simulation testbench and a hardware design example that supports compilation and hardware testing. When you generate the design example, the parameter editor automatically creates the files necessary to simulate, compile, and test the design in hardware.

The testbench and design example supports NRZ and PAM4 mode for E-tile devices. The Interlaken (2nd Generation) FPGA IP core generates design examples for all supported combinations of number of lanes and data rates.

The Interlaken (2nd Generation) IP core design example supports the following features:

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

Figure 1. Development Steps for the Design Example



Related Information

Interlaken (2nd Generation) FPGA IP User Guide

1.1. Hardware and Software Requirements

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

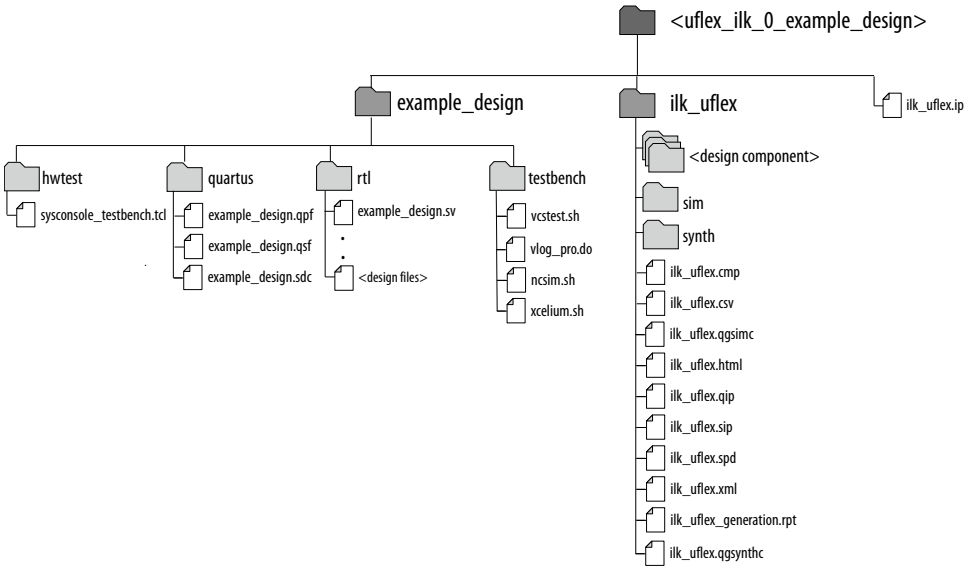
- Intel® Quartus® Prime Pro Edition software version 19.3
- System Console
- Modelsim-SE*, VCS*, NCSim* and Xcelium Parallel Simulator*

Note: The Intel Agilex™ only supports simulation in the Intel Quartus Prime 19.3 Release.

1.2. Directory Structure

The Interlaken (2nd Generation) IP core design example file directories contain the following generated files for the design example.

Figure 2. Directory Structure of the Generated Example Design for Intel Agilex



The hardware configuration, simulation, and test files are located in

<design_example_installation_dir>`/uflex_ilk_0_example_design`.

Table 1. Interlaken (2nd Generation) IP Core Hardware Design Example File Descriptions

These files are in the `quartus` directory.

File Names	Description
`example_design.qpf`	Intel Quartus Prime project file.
`example_design.qsf`	Intel Quartus Prime project settings file
`example_design.sdc` `jtag_timing_template.sdc`	Synopsys Design Constraint file. You can copy and modify for your own design.
`sysconsole_testbench.tcl`	Main file for accessing System Console

Table 2. Interlaken (2nd Generation) IP Core Testbench File Description

This file is in the `rtl` directory.

File Name	Description
`top_tb.sv`	Top-level testbench file.

Table 3. Interlaken (2nd Generation) IP Core Testbench Scripts

These files are in the `testbench` directory.

File Name	Description
`ncsim.sh`	The Cadence NCSim script to run the testbench.
`vcstest.sh`	The Synopsys VCS script to run the testbench.
`vlog_pro.do`	The Mentor Graphics ModelSim script to run the testbench.
`xcelium.sh`	The Cadence NCSim script to run the testbench.

1.3. Hardware Design Example Components

The example design connects system and PLL reference clocks and required design components. The example design configures the IP core in internal loopback mode and generates packets on the IP core TX user data transfer interface. The IP core sends these packets on the internal loopback path through the transceiver. The example design is only available for simulation in the Intel Quartus Prime 19.2 Release.

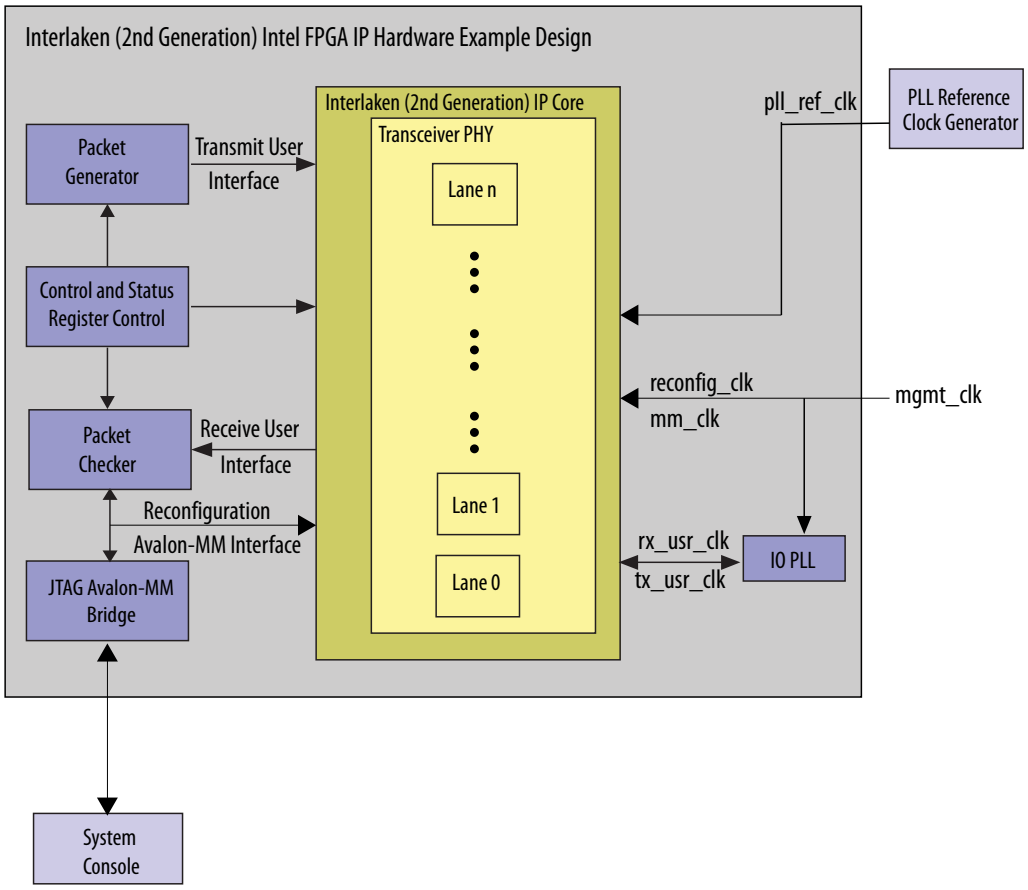
After the IP core receiver receives the packets on the loopback path, it processes the Interlaken packets and transmits them on the RX user data transfer interface. The example design checks that the packets received and transmitted match.

The hardware example design includes external PLLs. You can examine the clear text files to view sample code that implements one possible method to connect external PLLs to the Interlaken (2nd Generation) FPGA IP.

The Interlaken (2nd Generation) hardware design example includes the following components:

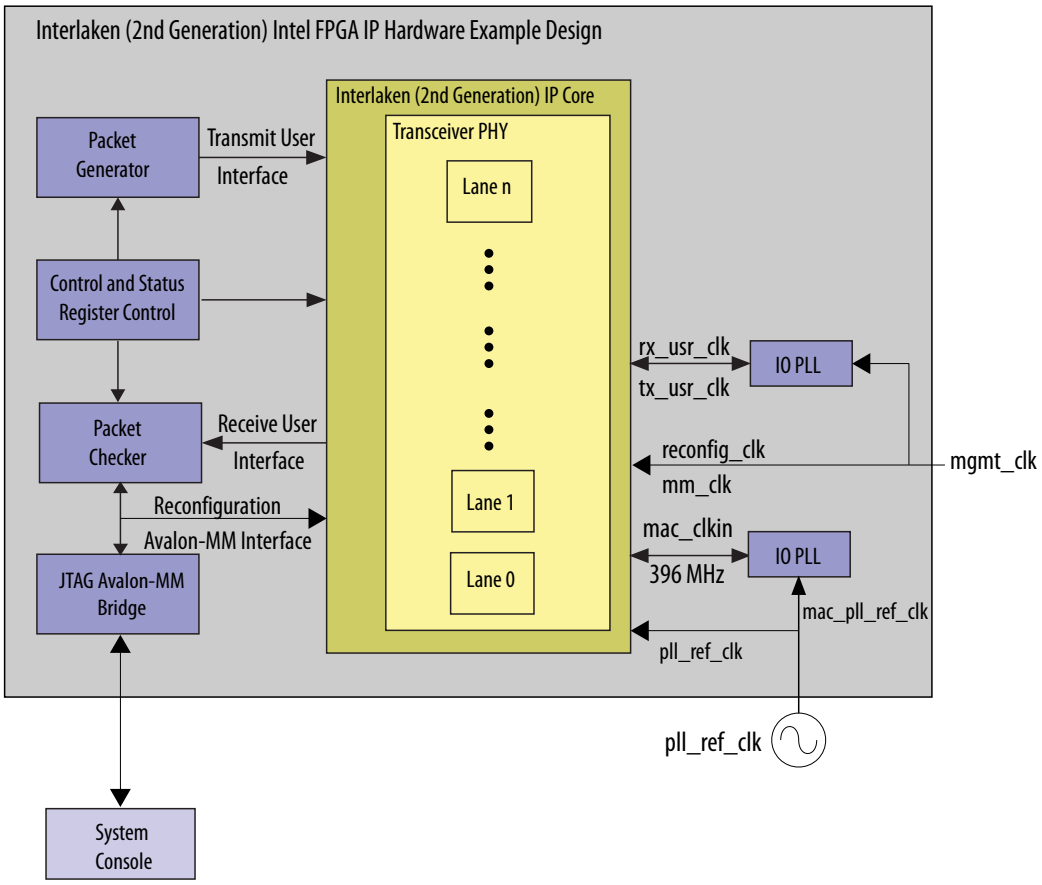
1. Interlaken (2nd Generation) FPGA IP
2. Packet Generator and Packet Checker
3. JTAG controller that communicates with System Console. You communicate with the client logic through the System Console.

Figure 3. Interlaken (2nd Generation) Hardware Design Example High Level Block Diagram for E-tile NRZ Mode Variations



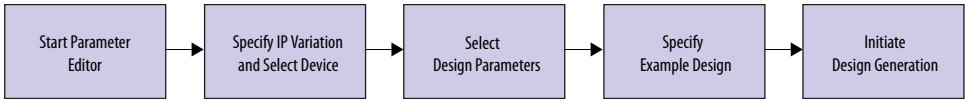
The Interlaken (2nd Generation) hardware design example that targets an E-tile PAM4 mode variations requires an additional clock `mac_clkin` that the IO PLL generates. This PLL must use the same reference clock that drives the `pll_ref_clk`.

Figure 4. Interlaken (2nd Generation) Hardware Design Example High Level Block Diagram for E-tile PAM4 Mode Variations



1.4. Generating the Design

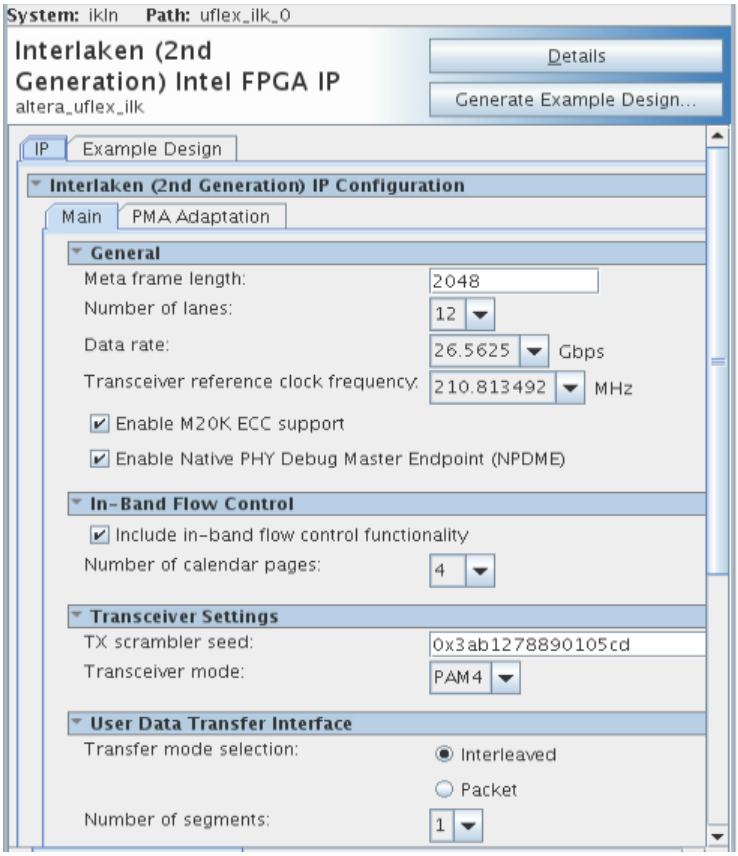
Figure 5. Procedure



Follow these steps to generate the hardware example design 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 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 **Agilex** and select device for your design.
3. In the IP Catalog, locate and double-click **Interlaken (2nd Generation) Intel FPGA IP**. The **New IP Variant** window appears.
4. 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.
5. Click **OK**. The parameter editor appears.

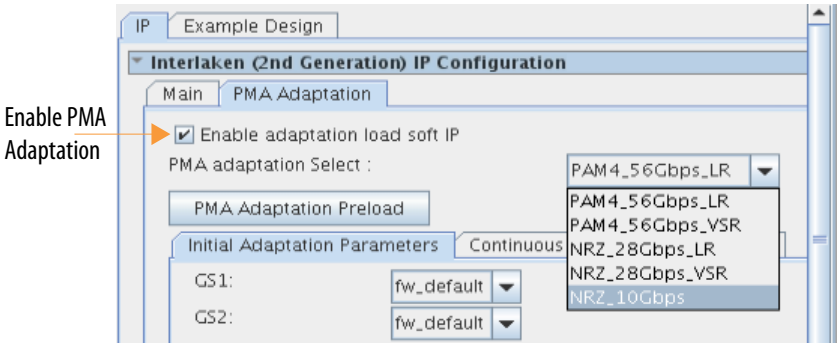
Figure 6. Parameter Editor



6. On the **IP** tab, specify the parameters for your IP core variation.

   Note: For E-tile device variations, you must select the **Enable adaptation load soft IP** option on the **PMA Adaptation** tab to enable the design example for PAM4 adaptation customization feature.

Figure 7. Enable PMA Adaptation



7. On the **Example Design** tab, select the **Simulation** option to generate the testbench, and select the **Synthesis** option to generate the hardware example design.

Note: You must select at least one of the **Simulation** or **Synthesis** options generate the Example Design Files.

8. For **Generated HDL Format**, only **Verilog** is available.
9. For **Target Development Kit** select the appropriate option.

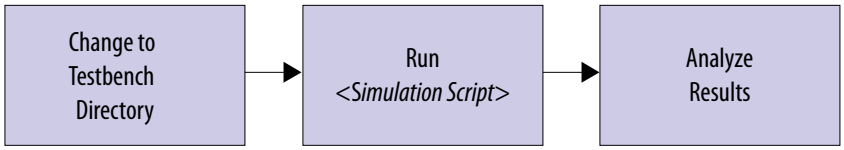
   Note: An Intel Agilex development board is not available in the current release. You can only simulate your design in the current release.

10. Click **Generate Example Design**. The **Select Example Design Directory** window appears.
11. If you want to modify the design example directory path or name from the defaults displayed (`uflex_ilk_0_example_design`), browse to the new path and type the new design example directory name.
12. Click **OK**.

1.5. Simulating the Design Example Testbench

Refer to *Interlaken (2nd Generation) Hardware Design Example High Level Block for E-tile NRZ Mode Variations* and *Interlaken (2nd Generation) Hardware Design Example High Level Block for E-tile PAM4 Mode Variations* block diagrams of the simulation testbench.

Figure 8. Procedure



Follow these steps to simulate the testbench:

1. At the command prompt, change to the testbench simulation directory. The directory is `<design_example_installation_dir>/example_design/testbench` for Intel Agilex devices.
2. Run the simulation script for the supported simulator of your choice. The script compiles and runs the testbench in the simulator. Your script should check that the SOP and EOP counts match after simulation is complete. Refer to the table *Steps to Run Simulation*.

Table 4. Steps to Run Simulation

Simulator	Instructions
ModelSim-SE*	In the command line, type `-do vlog_pro.do` If you prefer to simulate without bringing up the ModelSim GUI, type `vsim -c -do vlog_pro.do`
VCS	In the command line, type `sh vcstest.sh`
NCSim	In the command line, type `sh ncsim.sh`
Xcelium* Parallel Simulator	In the command line, type `sh xcelium.sh`

3. Analyze the results. A successful simulation sends 100 packets, receives 100 packets, and displays `"Test PASSED"`.

   The testbench for the design example completes the following tasks:

- Instantiates the Interlaken (2nd Generation) Intel FPGA IP.
- Prints PHY status.
- Checks metaframe synchronization (SYNC_LOCK) and word (block) boundaries (WORD_LOCK) .
- Waits for individual lanes to be locked and aligned.
- Starts transmitting packets.
- Checks packet statistics:
  - CRC24 errors
  - SOPs
  - EOPs

The following sample output illustrates a successful simulation test run:

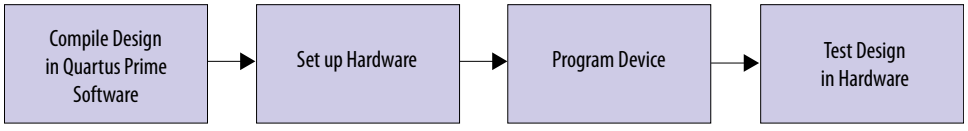
```
****************************************
                INFO: Waiting for lanes to be aligned
                      All of the receiver lanes are aligned and are ready
to receive traffic.
************************************************

************************************************
                INFO: Start transmitting packets
************************************************

************************************************
                INFO: Stop transmitting packets
************************************************

************************************************
                INFO: Checking packets statistics
************************************************

                CRC 24 errors reported: 0
                SOPs transmitted: 100
                EOPs transmitted: 100
                SOPs received: 100
                EOPs received: 100
                ECC error count: 0

************************************************
                INFO: Test PASSED
************************************************
```

Related Information

Hardware Design Example Components on page 5

1.6. Compiling and Configuring the Design Example in Hardware

Figure 9. Procedure



To compile and run a demonstration test on the hardware example design, follow these steps:

1. Ensure hardware example design generation is complete.
2. In the Intel Quartus Prime Pro Edition software, open the Intel Quartus Prime project <design_example_installation_dir>/example_design/quartus/example_design.qpf>.
3. On the **Processing** menu, click **Start Compilation**.
4. You cannot download your compiled Interlaken (2nd Generation) to an Intel Agilex development board in the current release.

Related Information

- Programming Intel FPGA Devices
- Analyzing and Debugging Designs with System Console

1.7. Testing the Hardware Design Example

After you compile the Interlaken (2nd Generation) Intel FPGA IP core design example and configure your device, you can use the System Console to program the IP core and its embedded Native PHY IP core registers.

Note: Hardware testing is not available for Intel Agilex devices in the current release.

Follow these steps to bring up the System Console and test the hardware design example:

1. In the Intel Quartus Prime Pro Edition software, on the **Tools** menu, click **System Debugging Tools ➤ System Console**.
2. Change to the *<design_example_installation_dir>*example_design/hwtest directory.
3. To open a connection to the JTAG master, type the following command:

```
source sysconsole_testbench.tcl
```

4. You can turn on internal serial loopback mode with the following design example commands:
   a. `stat`: Prints general status info.
   b. `sys_reset`: Resets the system.
   c. `loop_on`: Turns on internal serial loopback.
   d. `run_example_design`: Runs the design example.

   Note: You must run the `loop_on` command before the `run_example_design` command for internal serial loopback mode. The `run_example_design` runs the following commands in a sequence: `sys_reset->stat->gen_on->stat->gen_off`

   Note: When you select the **Enable adaptation load soft IP** option, the `run_example_design` command performs the initial adaptation calibration on RX side by running the `calibration_recipe_load` command.
5. You can turn off internal serial loopback mode with the following design example command:

a. `loop_off`: Turns off internal serial loopback.

6. You can program the IP core with the following additional design example commands:
   a. `gen_on`: Enables packet generator.
   b. `gen_off`: Disables packet generator.
   c. `run_test_loop`: Runs the test for <*N*> times for E-tile NRZ and PAM4 variations.
   d. `clear_err`: Clears all sticky error bits.
   e. `set_test_mode` <*min_okt_size*> <*max_pkt_size*> <*step*> <*num_to_run*>: Sets up test to run in a specific mode.
   f. `get_test_mode`: Prints the current test mode.
   g. `set_burst_size` <*burst_size*>: Sets burst size in bytes.
   h. `get_burst_size`: Prints burst size information.

   The following sample output illustrates a successful test run:

```
Elapsed 1 sec since powerup

            ************************************************
            INFO: Transmitting packets
            ************************************************

            ************************************************
            INFO: INFO: Checking received packets statistics
            ************************************************


            === STATUS REPORT ===


            TX KHz:402832
            RX KHz: 402045
            Freq locks: 0x000fff
            TX PLL lock: 0x010001
            Align: 0x00c10f
            Rx LOA: 0x00000
            Tx LOA: 0x00000

            word lock: 0x000fff
            sync lock: 0x000fff

            CRC32 errors: 0
            CRC24 errors: 0
            Checker errors: 0

            FIFO err flags: 0x000000
            SOP count: 100
            EOP count: 100
            ECC corrected: 0
            ECC error: 0

            Elaspsed 1 sec since powerup
```

2. Design Example Description

The design example demonstrates the functionalities of the Interlaken IP core.

Related Information

Interlaken (2nd Generation) FPGA IP User Guide

2.1. Design Example Behavior

To test the design in hardware, type the following commands in the System Console::

1. Source the setup file:

```
% source  <design_example>uflex_ilk_0_example_design/example_design/hwtest/
sysconsole_testbench.tcl
```

2. Run the test:

```
% run_example_design
```

3. The Interlaken (2nd Generation) hardware design example completes the following steps:
   a. Resets the Interlaken (2nd Generation) IP.
   b. Configures the Interlaken (2nd Generation) IP in internal loopback mode.
   c. Sends a stream of Interlaken packets with predefined data in the payload to the TX user data transfer interface of the IP core.
   d. Checks the received packets and reports the status. The packet checker included in the hardware design example provides the following basic packet checking capabilities:

- Checks that the transmitted packet sequence is correct.
- Checks that the received data matches the expected values by ensuring both the start of packet (SOP) and end of packet (EOP) counts align while data is being transmitted and received.

2.2. Interface Signals

Table 5. Design Example Interface Signals

Port Name	Direction	Width (Bits)	Description
clk50	Input	1	System clock input. Clock frequency must be 50 MHz.
pll_ref_clk	Input	1	Transceiver reference clock. Drives the RX CDR PLL.

continued...

Port Name	Direction	Width (Bits)	Description
rx_pin	Input	Number of lanes	Receiver SERDES data pin.
tx_pin	Output	Number of lanes	Transmit SERDES data pin.
usr_pb_reset_n	Input	1	System reset.

Related Information

Interface Signals

2.3. Register Map

Table 6. Design Example Register Map

Offset	Name	Access	Description
8'h00	Reserved		
8'h01	Reserved		
8'h02	System PLL reset	RO	Following bits indicates system PLL reset request and enable value: • Bit [0] - sys_pll_rst_req • Bit [1] - sys_pll_rst_en
8'h03	RX lane aligned	RO	Indicates the RX lane alignment.
8'h04	WORD locked	RO	[NUM_LANES-1:0] – Word (block) boundaries identification.
8'h05	Sync locked	RO	[NUM_LANES-1:0] – Metaframe synchronization.
8'h06 - 8'h09	CRC32 error count	RO	Indicates the CRC32 error count.
8'h0A	CRC24 error count	RO	Indicates the CRC24 error count.
8'h0B	Overflow/Underflow signal	RO	Following bits indicate: • Bit [3] - TX underflow signal • Bit [2] - TX overflow signal • Bit [1] - RX overflow signal
8'h0C	SOP count	RO	Indicates the number of SOP.
8'h0D	EOP count	RO	Indicates the number of EOP
8'h0E	Error count	RO	Indicates the number of following errors: • Loss of lane alignment • Illegal control word • Illegal framing pattern • Missing SOP or EOP indicator
8'h0F	send_data_mm_clk	RW	Write 1 to bit [0] to enable the generator signal.
8'h10	Reserved		
8'h11	System PLL lock	RO	Bit [0] indicates PLL lock indication.
8'h14	TX SOP count	RO	Indicates number of SOP generated by the packet generator.

continued...

Offset	Name	Access	Description
8'h15	TX EOP count	RO	Indicates number of EOP generated by the packet generator.
8'h39	ECC error count	RO	Indicates number of ECC errors.
8'h40	ECC corrected error count	RO	Indicates number of corrected ECC errors.

Note:

- Design Example register address starts with 0x20** while the Interlaken IP core register address starts with 0x10**.
- Access code: RO—Read Only, and RW—Read/Write.
- System console reads the design example registers and reports the test status on the screen.

3. Document Revision History for Interlaken (2nd Generation) Intel Agilex FPGA IP Design Example User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2019.09.30	19.3	19.2.1	Removed `clk100`. The `mgmt_clk` serves as a reference clock to the IO PLL in the following: • *Figure: Interlaken (2nd Generation) Hardware Design Example High Level Block Diagram for E-tile NRZ Mode Variations*. • *Figure: Interlaken (2nd Generation) Hardware Design Example High Level Block Diagram for E-tile PAM4 Mode Variations*.
2019.07.01	19.2	19.2	Initial release.