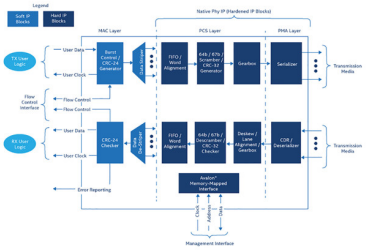


intel Interlaken (2nd Generation) Agilex FPGA IP Design Example User Guide

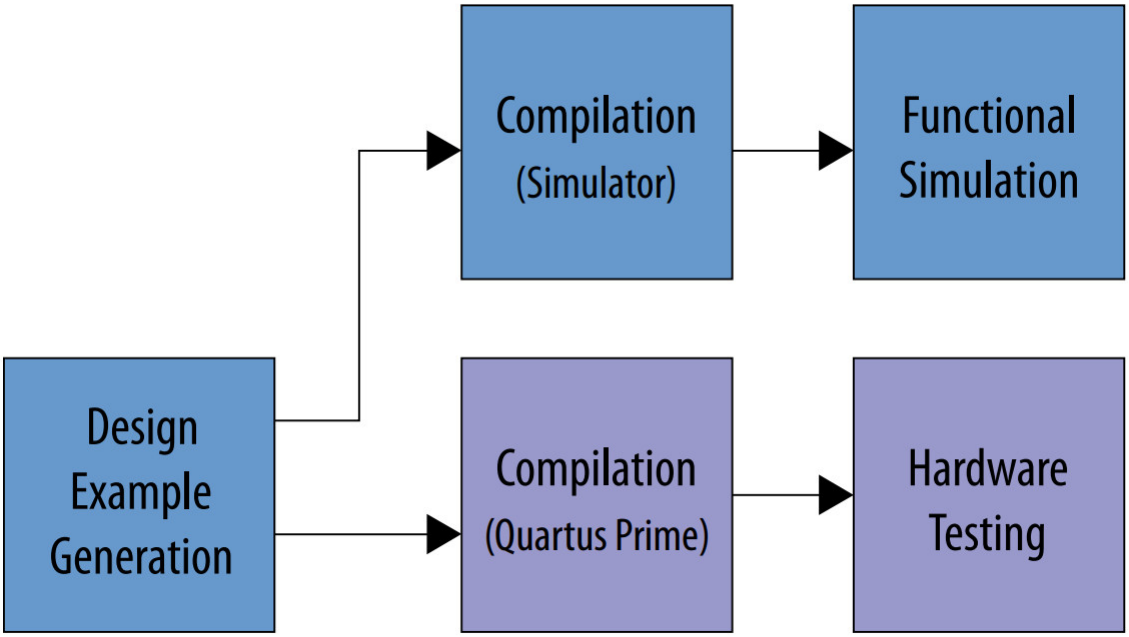
Interlaken (2nd Generation) Intel ® Agilex™ FPGA IP Design Example User Guide

Contents

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 design example is also available for Interlaken Look-aside feature.
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.

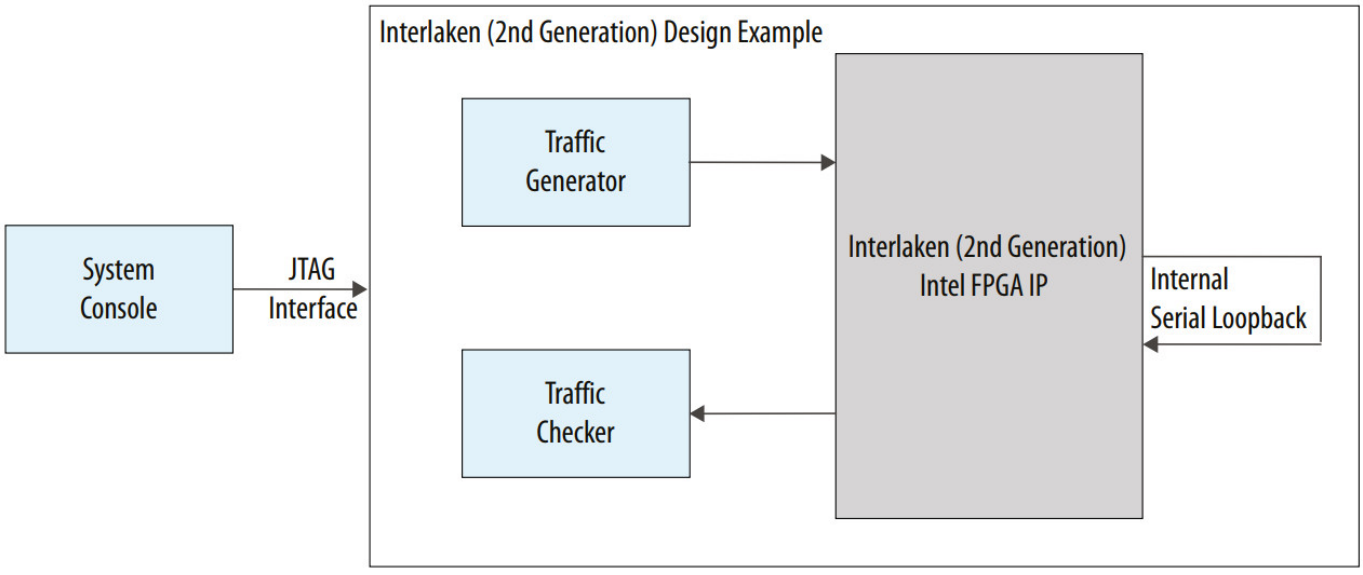
Figure 1. Development Steps for the Design Example



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 2. High-level Block Diagram for Interlaken (2nd Generation) Design Example



Related Information

- Interlaken (2nd Generation) FPGA IP User Guide
- Interlaken (2nd Generation) Intel FPGA IP Release Notes

1.1. Hardware and Software Requirements

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

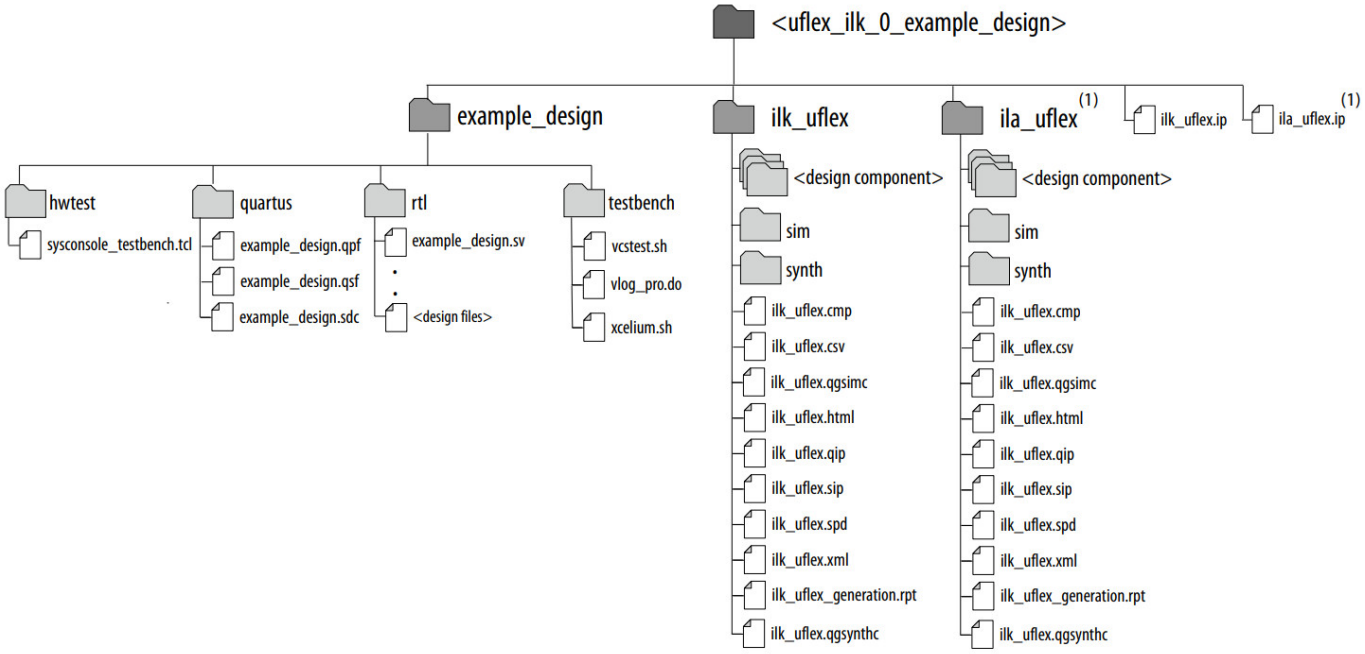
- Intel® Prime Pro Edition software version 21.3
- System Console
- Supported simulators:
  — Siemens* EDA ModelSim* SE or QuestaSim*
  — Synopsys* VCS*
  — Cadence* Xcelium*
- Intel Agilex® Quartus™ F-Series Transceiver-SoC Development Kit (AGFB014R24A2E2V)

Related Information
Intel Agilex F-Series Transceiver-SoC Development Kit User Guide

1.2. Directory Structure

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

Figure 3. Directory Structure of the Generated Interlaken (2nd Generation) Example Design



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 <design_example_installation_dir>/uflex_ilk_0_example_design/ example_design/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 m odify 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 <design_example_installation_dir>/uflex_ilk_0_example_design/ example_design/rtl directory.

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

Table 3. nterlaken (2nd Generation) IP Core Testbench Scripts
These files are in the <design_example_installation_dir>/uflex_ilk_0_example_design/ example_design/testbench directory.

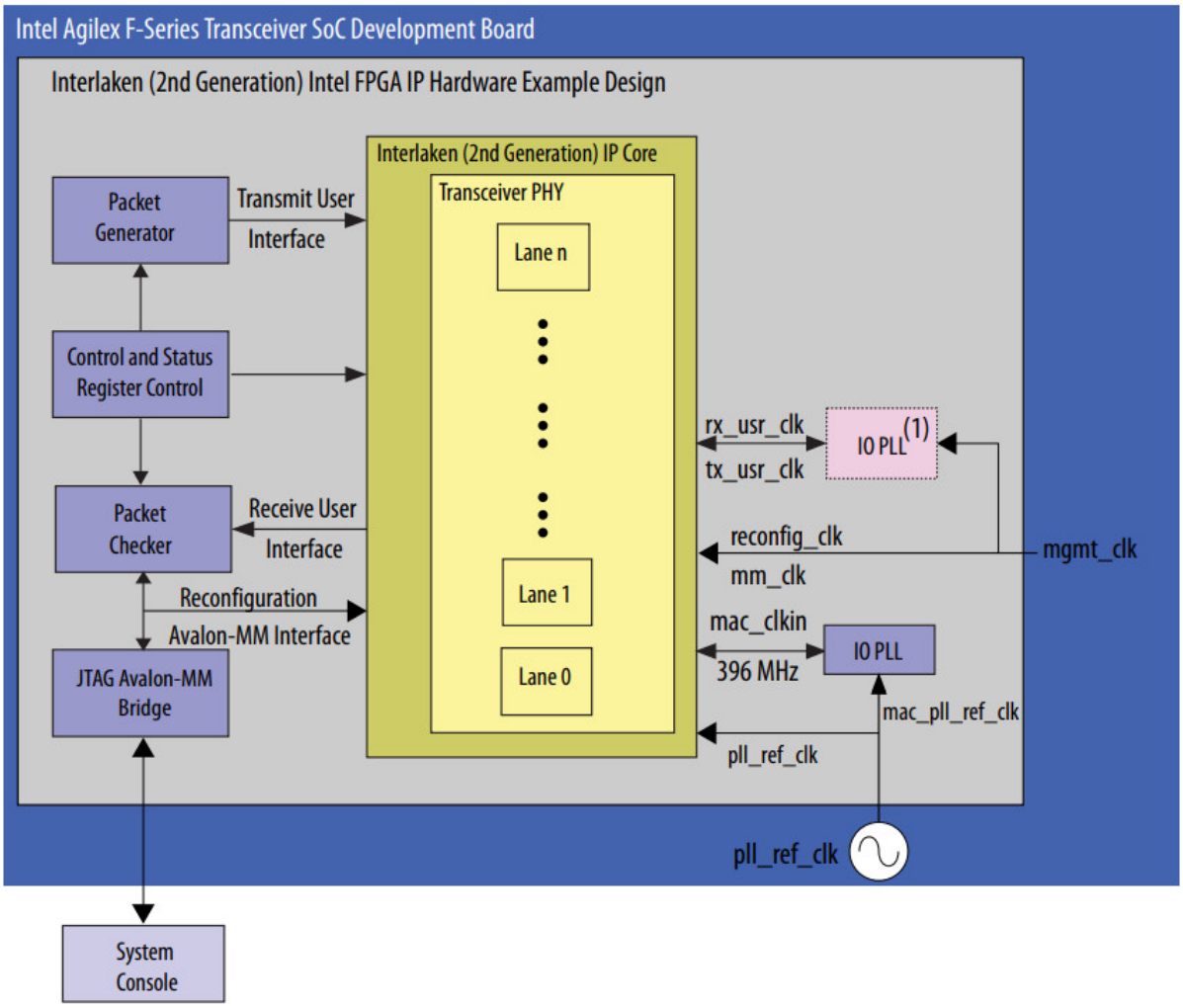
File Name	Description
vcstest.sh	The VCS script to run the testbench.
vlog_pro.do	The ModelSim SE or QuestaSim script to run the testb ench.
xcelium.sh	The Xcelium script to run the testbench.

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.
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.

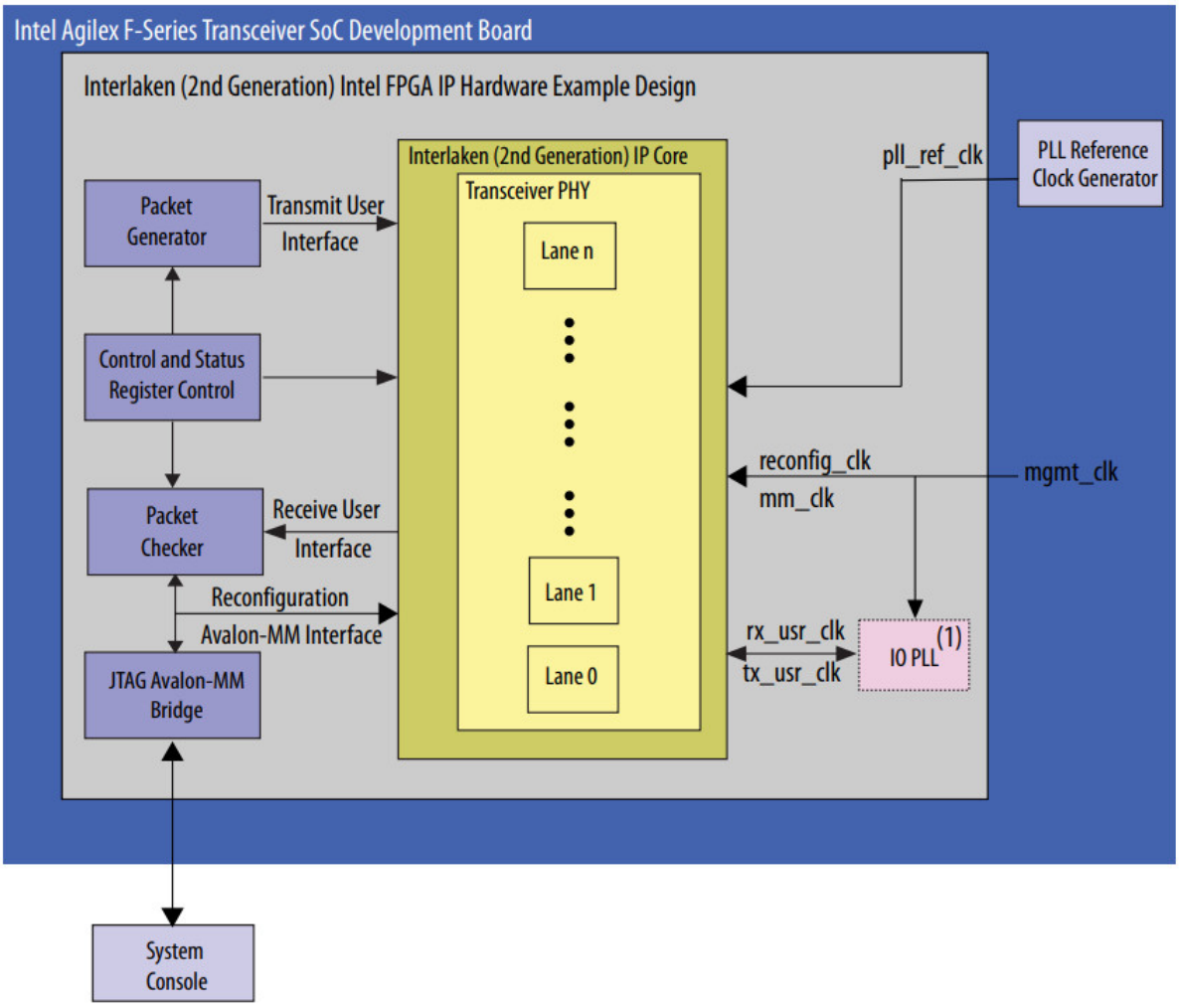
Figure 4. Interlaken (2nd Generation) Hardware Design Example High Level Block Diagram for E-tile NRZ Mode Variations



(1) IO PLL is not present if you generate design example for Interlaken Look-aside mode.

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



(1) IO PLL is not present if you generate design example for Interlaken Look-aside mode.

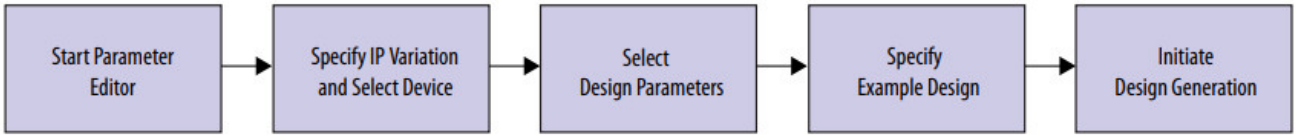
For E-tile PAM4 mode variations, when you enable the Preserve unused transceiver channels for PAM4 parameter, an additional reference clock port is added (pll_ref_clk [1]). This port must be driven at the same frequency as defined in the IP parameter editor (Reference clock frequency for preserved channels). The Preserve unused transceiver channels for PAM4 is optional. The pin and related constraints assigned to this clock is visible in the QSF when you select Intel Stratix® 10 or Intel Agilex development kit for design generation.
For design example simulation, the testbench always defines same frequency for pll_ref_clk[0] and pll_ref_clk[1].

Related Information

Intel Agilex F-Series Transceiver-SoC Development Kit User Guide

1.4. Generating the Design

Figure 6. Procedure

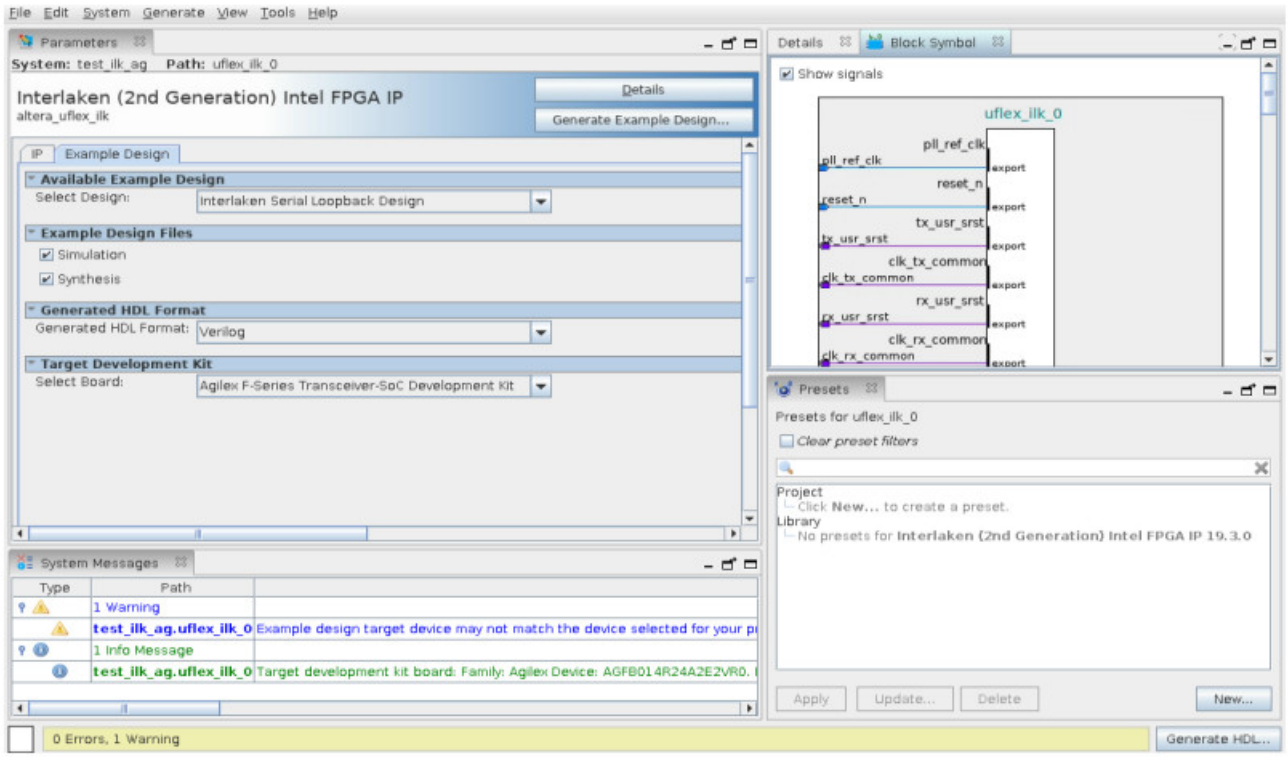


Follow these steps to generate the hardware example design and testbench:

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 7. Example Design Tab in the Interlaken (2nd Generation) Intel FPGA IP Parameter Editor



6. On the IP tab, specify the parameters for your IP core variation.
7. On the PMA Adaptation tab, specify the PMA adaptation parameters if you plan to use PMA adaptation for your E-tile device variations.

   This step is optional:

   • Select Enable adaptation load soft IP option.

   Note: You must enable Enable Native PHY Debug Master Endpoint (NPDME) option on the IP tab when PMA adaptation is enabled.

   • Select a PMA adaptation preset for PMA adaptation Select parameter.

   • Click PMA Adaptation Preload to load the initial and continuous adaptation parameters.

   • Specify the number of PMA configurations to support when multiple PMA configurations are enabled using Number of PMA configuration parameter.

   • Select which PMA configuration to load or store using Select a PMA configuration to load or store.

   • Click Load adaptation from selected PMA configuration to load the selected PMA configuration settings.

   For more information about the PMA adaptation parameters, refer to the E-tile Transceiver PHY User Guide.
8. 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.
9. For Generated HDL Format, only Verilog is available.
10. For Target Development Kit select the appropriate option.

    Note: The Intel Agilex F-Series Transceiver SoC Development Kit option is only available when your project specifies Intel Agilex device name starting with AGFA012 or AGFA014. When you select the Development Kit option, the pin assignments are set according to the Intel Agilex Development Kit device part number AGFB014R24A2E2V and may differ from your selected device. If you intend to test the design on hardware on

a different PCB, select No development kit option and make the appropriate pin assignments in the .qsf file.

11. Click Generate Example Design. The Select Example Design Directory window appears.
12. 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.
13. Click OK.

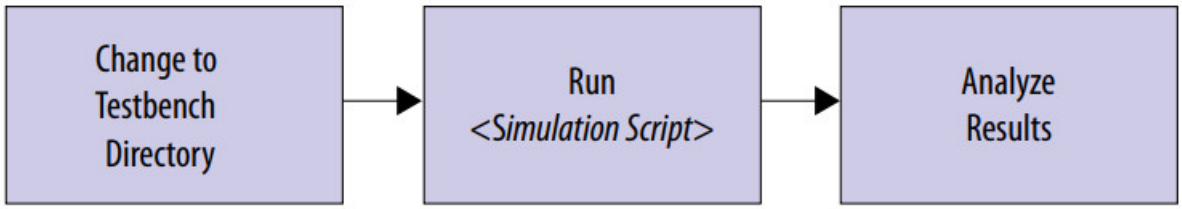
Related Information

- **Intel Agilex F-Series Transceiver-SoC Development Kit User Guide**
- **E-tile Transceiver PHY User Guide**

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 or Questa Sim	In the command line, type -do vlog_pro.do. If you prefer to simulate without bring ing up the ModelSim GUI, type vsim -c -do vlog_pro.do
VCS	In the command line, type sh vcstest.sh
Xcelium	In the command line, type sh xcelium.sh

3. Analyze the results. A successful simulation sends and receives 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 in Interlaken mode:

```
**********************************************
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
**************************************************
```

Note: The Interlaken design example simulation testbench sends 100 packets and receives 100 packets.
The following sample output illustrates a successful simulation test run in Interlaken Look-aside mode:

```
Check TX and RX Counter equal or not.
-----------------------------------------------
READ_MM: address 4000014 = 00000001.
-----------------------------------------------
De-assert Counter equal bit.
-----------------------------------------------
WRITE_MM: address 4000001 gets 00000001.
WRITE_MM: address 4000001 gets 00000000.
-----------------------------------------------
RX_SOP COUNTER.
-----------------------------------------------
READ_MM: address 400000c = 0000006a.
-----------------------------------------------
RX_EOP COUNTER.
READ_MM: address 400000d = 0000006a.
-----------------------------------------------
READ_MM: address 4000010 = 00000000.
-----------------------------------------------
Display Final Report.
-----------------------------------------------
0 Detected Error
0 CRC24 errors reported
106 SOPs transmitted
```

106 EOPs transmitted
106 SOPs received
106 EOPs received

Finish Simulation

TEST PASSED

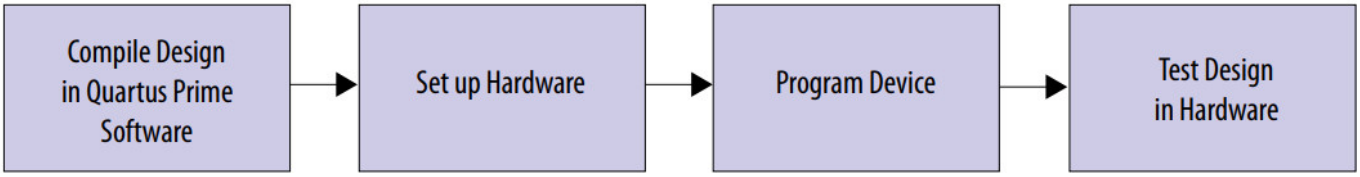
Note: The number of packets (SOPs and EOPs) varies per lane in Interlaken Lookaside design example simulation sample output.

Related Information

Hardware Design Example Components on page 6

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. After successful compilation, a .sof file is available in your specified directory.

   Follow these steps to program the hardware example design on the Intel Agilex device:
5. Connect Intel Agilex F-Series Transceiver-SoC Development Kit to the host computer.

   b. Launch the Clock Control application, which is part of the development kit, and set new frequencies for the design example. Below is the frequency setting in the Clock Control application:

   • Si5338 (U37), CLK1- 100 MHz

   • Si5338 (U36), CLK2- 153.6 MHz

   • Si549 (Y2), OUT- Set to the value of pll_ref_clk (1) per your design requirement.

   c. On the Tools menu, click Programmer.

   d. In the Programmer, click Hardware Setup.

   e. Select a programming device.

   f. Select and add the Intel Agilex F-Series Transceiver-SoC Development Kit to which your Intel Quartus Prime session can connect.

   g. Ensure that Mode is set to JTAG.

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

   i. In the row with your .sof, check the box for the .sof.

   j. Check the box in the Program/Configure column.

   k. Click Start.

Related Information

- **Programming Intel FPGA Devices** on page 0
- **Analyzing and Debugging Designs with System Console**
- **Intel Agilex F-Series Transceiver-SoC Development Kit User Guide**

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.
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 loop_on command before run_example_design command.
   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 run_load_PMA_configuration 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 successful test prints HW_TEST:PASS message. Below is the passing criteria for a test run:

- No errors for CRC32, CRC24, and checker.
- Transmitted SOPs and EOPs should be match with received.

The following sample output illustrates a successful test run in Interlaken mode:
INFO: INFO: Stop generating packtes
==== STATUS REPORT ====
TX KHz : 402813
RX KHz : 402813

Freq locks : 0x0000ff
TX PLL lock : 0x000001
Align : 0x00c10f
Rx LOA : 0x000000
Tx LOA : 0x000000
word lock : 0x0000ff
sync lock : 0x0000ff
CRC32 errors : 0
CRC24 errors : 0
Checker errors : 0
FIFO err flags : 0x000000
SOPs transmitted : 1087913770
EOPs transmitted : 1087913770
SOPs received : 1087913770
EOPs received : 1087913770
ECC corrected : 0
ECC error : 0
Elapsed 161 sec since powerup
HW_TEST : PASS
The successful test prints HW_TEST : PASS message. Below is the passing criteria for a test run:

- No errors for CRC32, CRC24, and checker.
- Transmitted SOPs and EOPs should be match with received.

The following sample output illustrates a successful test run in Interlaken Lookaside mode:
INFO: INFO: Stop generating packtes
==== STATUS REPORT ====
TX KHz : 402813
RX KHz : 402812
Freq locks : 0x000fff
TX PLL lock : 0x000001
Align : 0x00c10f
Rx LOA : 0x000000
Tx LOA : 0x000000
word lock : 0x000fff
sync lock : 0x000fff
CRC32 errors : 0
CRC24 errors : 0
Checker errors : 0
SOPs transmitted : 461
EOPs transmitted : 461
SOPs received : 461
EOPs received : 461
Elapsed 171 sec since powerup
HW_TEST : PASS

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
mgmt_clk	Input	1	System clock input. Clock frequency must be 100 MHz.
pll_ref_clk /pll_ref_clk[1:0] (2)	Input	2-Jan	Transceiver reference clock. Drives the RX CDR PLL.

Port Name	Direction	Width (Bits)	Description
			pll_ref_clk[1] is only available when you enable **Preserve unused** Note: **transceiver channels for PAM4** parameter in E-tile PAM4 mode IP variations.
rx_pin	Input	Number of lanes	Receiver SERDES data pin.
tx_pin	Output	Number of lanes	Transmit SERDES data pin.
rx_pin_n	Input	Number of lanes	Receiver SERDES data pin. This signal is only available in E-tile PAM 4 mode device variations.
tx_pin_n	Output	Number of lanes	Transmit SERDES data pin. This signal is only available in E-tile PAM 4 mode device variations.
mac_clk_pll_ref	Input	1	This signal must be driven by a PLL and must use the same clock source that drives the pll_ref_clk. This signal is only available in E-tile PAM 4 mode device variations.
usr_pb_reset_n	Input	1	System reset.

Related Information
[Interface Signals]
2.3. Register Map
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.

Table 6. Design Example Register Map for Interlaken Design Example

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 id entification.

(2) When you enable Preserve unused transceiver channels for PAM4 parameter, an additional reference clock port is added to preserve the unused PAM4 slave channel.

Offset	Name	Access	Description
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	Checker error		Indicates the checker error. (SOP data error, Cha nnel number error, and PLD data error)
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.
8'h15	TX EOP count	RO	Indicates number of EOP generated by the packet generator.
8'h16	Continuous packet	RW	Write 1 to bit [0] to enable the continuous packet.
8'h39	ECC error count	RO	Indicates number of ECC errors.
8'h40	ECC corrected error count	RO	Indicates number of corrected ECC errors.

Table 7. Design Example Register Map for Interlaken Look-aside Design Example
Use this register map when you generate the design example with Enable Interlaken Look-aside mode parameter turned on.

Offset	Name	Access	Description
8’h00	Reserved		
8’h01	Counter reset	RO	Write 1 to bit [0] to clear TX and RX counter equal bit.
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 id entification.
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.

Offset	Name	Access	Description
8’h0B	Reserved		
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	Checker error	RO	Indicates the checker error. (SOP data error, Cha nnel number error, and PLD data error)
8’h11	System PLL lock	RO	Bit [0] indicates PLL lock indication.
8’h13	Latency count	RO	Indicates number of latency.
8’h14	TX SOP count	RO	Indicates number of SOP generated by the packet generator.
8’h15	TX EOP count	RO	Indicates number of EOP generated by the packet generator.
8’h16	Continuous packet	RO	Write 1 to bit [0] to enable the continuous packet.
8’h17	TX and RX counter equal	RW	Indicates TX and RX counter are equal.
8’h23	Enable latency	WO	Write 1 to bit [0] to enable latency measurement.
8’h24	Latency ready	RO	Indicates latency measurement are ready.

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

For the latest and previous versions of this user guide, refer to the **Interlaken (2nd Generation) Intel Agilex FPGA IP Design Example User Guide** HTML version. Select the version and click Download. If an IP or software version is not listed, the user guide for the previous IP or software version applies.
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.

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

Document Version	Intel Quartus Prime Version	IP Version	Changes
2022.08.03	21.3	20.0.1	Corrected the device OPN for the Intel Agilex F-Series Transceiver-SoC Development Kit.
2021.10.04	21.3	20.0.1	• Added support for QuestaSim simulator. • Removed support for NCSim simulator.
2021.02.24	20.4	20.0.1	• Added information about preserving the unused transceiver channel for PAM4 in section: Hardware Design Example Components. • Added the pll_ref_clk[1] signal description in section: Interface Signals.
2020.12.14	20.4	20.0.0	• Updated sample hardware test output for Interlaken mode and Interlaken Look-aside mode in section Testing the Hardware Design Example. • Updated register map for Interlaken Look-aside design example in section Register Map. • Added a passing criteria for a successful hardware test run in section Testing the Hardware Design Example.
2020.10.16	20.2	19.3.0	Corrected command to run the initial adaptation calibration on RX side in Testing the Hardware Design Example section.

2020.06.22	20.2	19.3.0	• The design example is available for Interlaken Look- aside mode. • Hardware testing of the design example is availa ble for Intel Agilex device variations. • Added Figure: High-level Block Diagram for Inter laken (2nd Generation) Design Example. • Updated following sections: – Hardware and Software Requirements – Directory Structure • Modified the following figures to include Interlake n Look-aside related update: – Figure: Interlaken (2nd Generation) Hardware D esign Example High Level Block Diagram for E- tile NRZ Mode Variatio ns – Figure: Interlaken (2nd Generation) Hardware D esign Example High Level Block Diagram for E- til e PAM4 Mode Variations • Updated Figure: IP Parameter Editor. • Added information about the frequency settings i n the clock control application in section Compiling and Configuring the Design Example in Hardware.

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Added test run outputs for the Interlaken Look- a side in the following sections: – Simulating the Design Example Testbench – Testing the Hardware Design Example • Added following new signals in Interface Signals section: – mgmt_clk – rx_pin_n – tx_pin_n – mac_clk_pll_ref • Added register map for Interlaken Look-aside de sign example in section: Register Map.
2019.09.30	19.3	19.2.1	Removed clk100. The mgmt_clk serves as a refer ence clock to the IO PLL in the following: • Figure: Interlaken (2nd Generation) Hardware D esign Example High Level Block Diagram for E-tile NRZ Mode Variations. • Figure: Interlaken (2nd Generation) Hardware D esign Example High Level Block Diagram for E-tile PAM4 Mode Variations.
2019.07.01	19.2	19.2	Initial release.

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

Online Version

Send Feedback
ID: 683800
UG-20239
Version: 2022.08.03

Documents / Resources

	intel Interlaken (2nd Generation) Agilex FPGA IP Design Example [pdf] User Guide Interlaken 2nd Generation Agilex FPGA IP Design Example, Interlaken, 2nd Generation Agilex FPGA IP Design Example, Agilex FPGA IP Design Example, IP Design Example