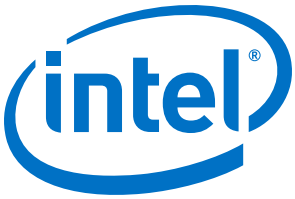


Interlaken (2nd Generation) Intel® FPGA IP User Guide

Updated for Intel® Quartus® Prime Design Suite: **18.0.1**

UG-20035 | 2018.07.16
Latest document on the web: **PDF** | **HTML**

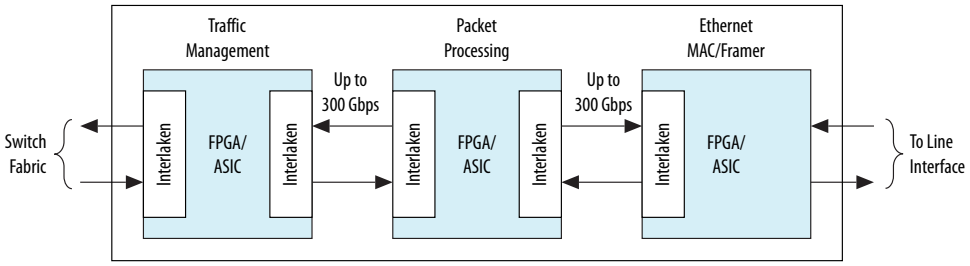
Contents

1. About this IP Core

Interlaken is a high-speed serial communication protocol for chip-to-chip packet transfers. The Interlaken (2nd Generation) Intel® FPGA IP implements the *Interlaken Protocol Specification, Revision 1.2*. It supports multiple combinations of number of lanes (4 to 12) and lane rates from 6.25 gigabits per second (Gbps) to 53.125 Gbps, on Intel Stratix® 10 devices, providing raw bandwidth of 25 Gbps to 300 Gbps.

Interlaken provides low I/O count compared to earlier protocols, supporting scalability in both number of lanes and lane speed. Other key features include flow control, low overhead framing, and extensive integrity checking. The Interlaken IP core incorporates a physical coding sublayer (PCS), a physical media attachment (PMA), and a media access control (MAC) block.

Figure 1. Typical Interlaken Application



Related Information

- Interlaken IP Core (2nd Generation) Design Example User Guide
  Describes a simulating testbench and a hardware example design that supports compilation and hardware testing, to help you understand usage.
- Interlaken Protocol Specifications

1.1. Features

The Interlaken (2nd Generation) IP core has the following features:

- Compliant with the Interlaken Protocol Specification, Revision 1.2.
- Supports 4, 6, and 12 serial lanes in configurations that provide up to 318.75 Gbps raw bandwidth.
- Supports per-lane data rates of 6.25, 10.3125, 12.5, 25.3, 25.8 and 53.125 Gbps using Intel FPGA on-chip high-speed transceivers.
- Supports dynamically configurable BurstMax and BurstMin values.
- Supports Packet mode and Interleaved mode for user data transfer.
- Supports up to 256 logical channels in out-of-the-box configuration.

- Supports optional user-controlled in-band flow control with 1, 2, 4, 8, or 16 16-bit calendar pages.
- Supports optional out-of-band flow control blocks.
- Supports memory block ECC.
- Supports per-lane data rate of 53.125 Gbps using pulse amplitude modulation (PAM4) mode in Intel Stratix 10 E-Tile variations.

Table 1. IP Core Supported Combinations of Number of Lanes and Data Rate

The following combinations are supported in Intel Quartus® Prime Pro Edition 18.0.1

Device	IP Core Supported Combinations	
	Number of Lanes	Lane Rate (Gbps)
Intel Stratix 10 L-Tile	4	6.25
	12	10.3125
	12	12.5
Intel Stratix 10 H-Tile	4	6.25
	6	25.3
	6	25.8
	12	10.3125
	12	12.5
	12	25.3
	12	25.8
Intel Stratix 10 E-Tile (Non-Return-to-Zero (NRZ))	6	25.3
	6	25.8
	12	12.5
	12	25.3
	12	25.8
Intel Stratix 10 E-Tile (PAM4)	12	26.5625 *Note:* To obtain 6x53.125 Gbps speed in PAM4 mode, you must select 12x26.5625 Gbps combination in Intel Quartus Prime Pro Edition 18.0.1

Table 2. IP Core Theoretical Raw Aggregate Bandwidth

The following combinations are supported in Intel Quartus Prime Pro Edition 18.0.1

Number of Lanes	Lane Rate (Gbps)						Number of Words	Data Width (bits)
	6.25	10.3125	12.5	25.3	25.8	26.5625		
4	25	-	-	-	-	-	4	256
12	-	-	-	151.8	154.8	318.75	8	512
12	-	123.75	150	-	-	-	8	512
12	-	-	-	303.6	309.6	-	16	1024

Related Information

Interlaken Protocol Specifications

1.2. Device Family Support

The following lists the device support level definitions for Intel FPGA IP cores:

- **Advance support** — The IP core is available for simulation and compilation for this device family. Timing models include initial engineering estimates of delays based on early post-layout information. The timing models are subject to change as silicon testing improves the correlation between the actual silicon and the timing models. You can use this IP core for system architecture and resource utilization studies, simulation, pinout, system latency assessments, basic timing assessments (pipeline budgeting), and I/O transfer strategy (data-path width, burst depth, I/O standards tradeoffs).
- **Preliminary support** — The IP core is verified with preliminary timing models for this device family. The IP core meets all functional requirements, but might still be undergoing timing analysis for the device family. It can be used in production designs with caution.
- **Final support** — The IP core is verified with final timing models for this device family. The IP core meets all functional and timing requirements for the device family and can be used in production designs.

Table 3. Device Family Support

Device Family	Support
Intel Stratix 10	Advance

1.3. Performance and Resource Utilization

Lists the resources and expected performance for selected variations of the Interlaken IP core using the Intel Quartus Prime Pro Edition 18.0.1. The numbers of ALMs and logic registers are rounded up to the nearest 100.

Table 4. FPGA Resource Utilization

Device	Parameters		Resource Utilization			
	Number of Lanes	Data/Lane Rate (Gbps)	ALMs needed	Logic Registers		M20K Blocks
				Primary	Secondary	
Intel Stratix 10 L-Tile	4	6.25	8300	16600	4800	28
	12	10.3125	18300	39400	9700	52
	12	12.5	18400	39200	9700	52
Intel Stratix 10 H-Tile	6	25.3	19700	39700	9600	52
	12	25.3	32900	69800	14000	100
Intel Stratix 10 E-Tile (NRZ)	6	25.3	28000	49500	12800	52
	12	12.5	45000	83300	19400	73
	12	25.3	48500	92000	19700	100
Intel Stratix 10 E-Tile (PAM4)	6	53.125	61300	111900	22600	100

1.4. Release Information

Table 5. IP Core Release Information

Item	Value	
Version	Intel Quartus Prime Pro Edition 18.0.1	
Release Date	2018.07.16	
Ordering Code	**Aggregate Bandwidth**	**Ordering Code**
	20G to <100G	IP-ILKN/50G
	100G to <200G	IP-ILKN/100G
	200G to <400G	IP-ILKN/200G

2. Getting Started

The following sections explain how to install, parameterize, simulate, and initialize the Interlaken IP core.

Related Information

- Introduction to Intel FPGA IP Cores
  Provides general information about all Intel FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.
- Generating a Combined Simulator Setup Script
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- Project Management Best Practices
  Guidelines for efficient management and portability of your project and IP files.

2.1. Installing and Licensing Intel FPGA IP Cores

The Intel Quartus Prime software installation includes the Intel FPGA IP library. This library provides many useful IP cores for your production use without the need for an additional license. Some Intel FPGA IP cores require purchase of a separate license for production use. The Intel FPGA IP Evaluation Mode allows you to evaluate these licensed Intel FPGA IP cores in simulation and hardware, before deciding to purchase a full production IP core license. You only need to purchase a full production license for licensed Intel IP cores after you complete hardware testing and are ready to use the IP in production.

The Intel Quartus Prime software installs IP cores in the following locations by default:

Figure 2. IP Core Installation Path

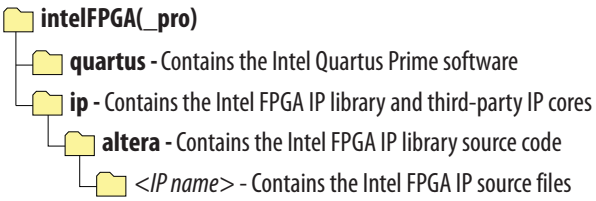


Table 6. IP Core Installation Locations

Location	Software	Platform
<drive>:\intelFPGA_pro\quartus\ip\altera	Intel Quartus Prime Pro Edition	Windows*
<home directory>:/intelFPGA_pro/quartus/ip/altera	Intel Quartus Prime Pro Edition	Linux*

2.1.1. Intel FPGA IP Evaluation Mode

The free Intel FPGA IP Evaluation Mode allows you to evaluate licensed Intel FPGA IP cores in simulation and hardware before purchase. Intel FPGA IP Evaluation Mode supports the following evaluations without additional license:

- Simulate the behavior of a licensed Intel FPGA IP core in your system.
- Verify the functionality, size, and speed of the IP core quickly and easily.
- Generate time-limited device programming files for designs that include IP cores.
- Program a device with your IP core and verify your design in hardware.

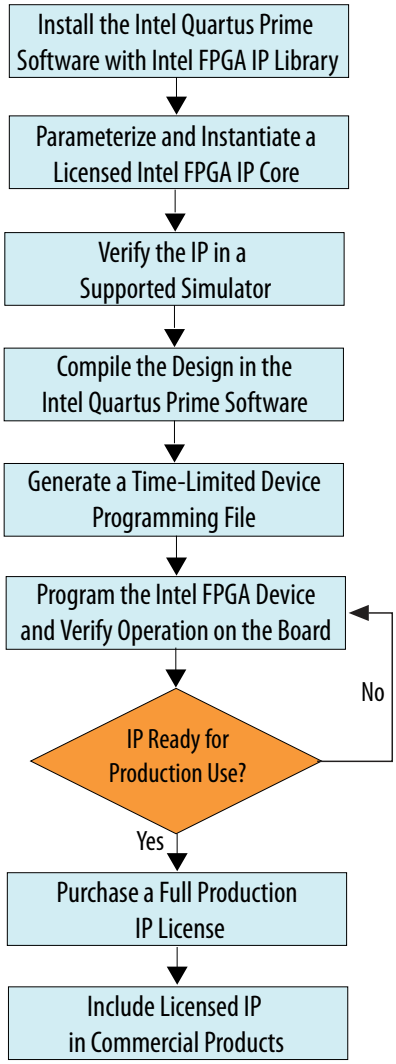
Intel FPGA IP Evaluation Mode supports the following operation modes:

- **Tethered**—Allows running the design containing the licensed Intel FPGA IP indefinitely with a connection between your board and the host computer. Tethered mode requires a serial joint test action group (JTAG) cable connected between the JTAG port on your board and the host computer, which is running the Intel Quartus Prime Programmer for the duration of the hardware evaluation period. The Programmer only requires a minimum installation of the Intel Quartus Prime software, and requires no Intel Quartus Prime license. The host computer controls the evaluation time by sending a periodic signal to the device via the JTAG port. If all licensed IP cores in the design support tethered mode, the evaluation time runs until any IP core evaluation expires. If all of the IP cores support unlimited evaluation time, the device does not time-out.
- **Untethered**—Allows running the design containing the licensed IP for a limited time. The IP core reverts to untethered mode if the device disconnects from the host computer running the Intel Quartus Prime software. The IP core also reverts to untethered mode if any other licensed IP core in the design does not support tethered mode.

When the evaluation time expires for any licensed Intel FPGA IP in the design, the design stops functioning. All IP cores that use the Intel FPGA IP Evaluation Mode time out simultaneously when any IP core in the design times out. When the evaluation time expires, you must reprogram the FPGA device before continuing hardware verification. To extend use of the IP core for production, purchase a full production license for the IP core.

You must purchase the license and generate a full production license key before you can generate an unrestricted device programming file. During Intel FPGA IP Evaluation Mode, the Compiler only generates a time-limited device programming file (`<`*project name*`>_time_limited.sof`) that expires at the time limit.

Figure 3. Intel FPGA IP Evaluation Mode Flow



Note: Refer to each IP core's user guide for parameterization steps and implementation details.

Intel licenses IP cores on a per-seat, perpetual basis. The license fee includes first-year maintenance and support. You must renew the maintenance contract to receive updates, bug fixes, and technical support beyond the first year. You must purchase a full production license for Intel FPGA IP cores that require a production license, before generating programming files that you may use for an unlimited time. During Intel FPGA IP Evaluation Mode, the Compiler only generates a time-limited device programming file (`<project name>_time_limited.sof`) that expires at the time limit. To obtain your production license keys, visit the Self-Service Licensing Center or contact your local Intel FPGA representative.

The Intel FPGA Software License Agreements govern the installation and use of licensed IP cores, the Intel Quartus Prime design software, and all unlicensed IP cores.

Related Information

- Intel Quartus Prime Licensing Site
- Intel FPGA Software Installation and Licensing

2.2. Generated File Structure

The Intel Quartus Prime Pro Edition software generates the following IP core output file structure.

For more information about the file structure of the design example, refer to the *Interlaken IP Core (2nd Generation) Design Example User Guide*.

Figure 4. IP Core Generated Files

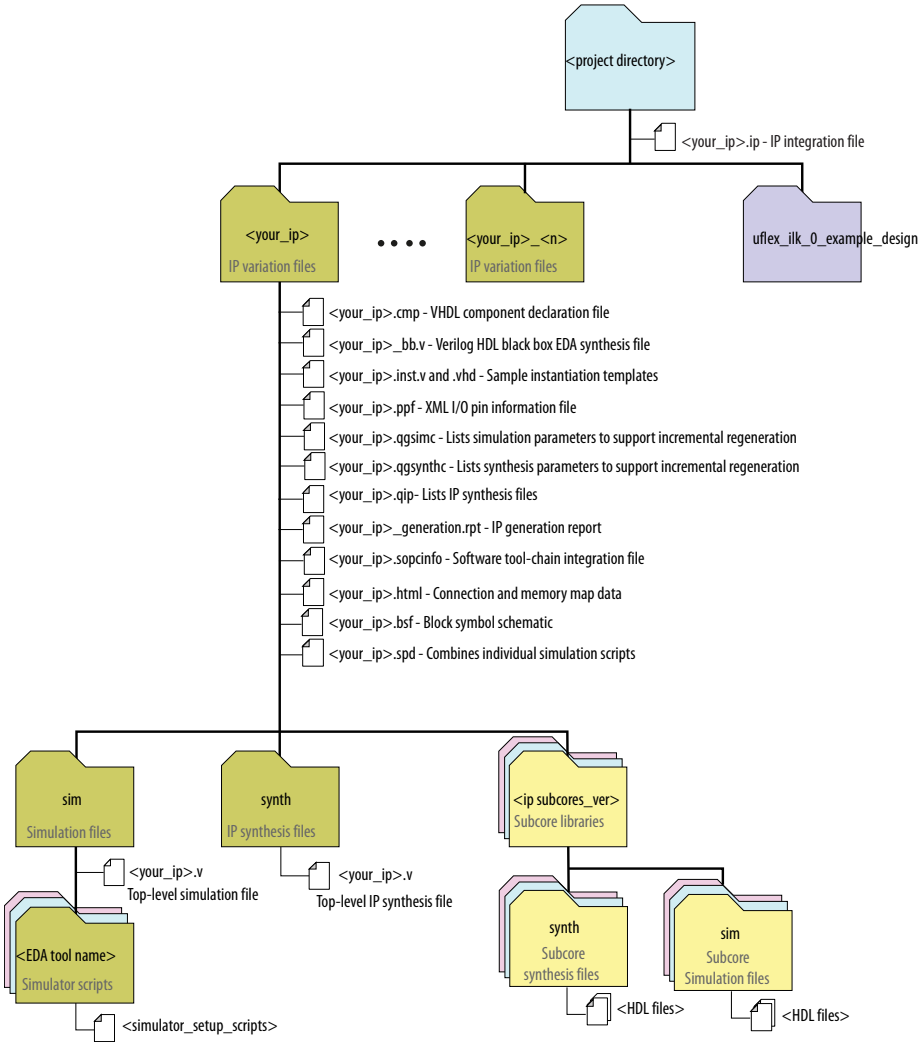


Table 7. IP Core Generated Files

File Name	Description
<*your_ip*>.ip	The top-level IP variation file. <*your_ip*> is the name that you give your IP variation.
<*your_ip*>.cmp	The VHDL Component Declaration (**.cmp**) file is a text file that contains local generic and port definitions that you can use in VHDL design files. This IP core does not support VHDL. However, the Intel Quartus Prime Pro Edition software generates this file.
<*your_ip*>.html	A report that contains connection information, a memory map showing the address of each slave with respect to each master to which it is connected, and parameter assignments.
<*your_ip*>_generation.rpt	IP or Platform Designer generation log file. A summary of the messages during IP generation.
<*your_ip*>.qgsimc	Lists simulation parameters to support incremental regeneration.
<*your_ip*>.qgsynthc	Lists synthesis parameters to support incremental regeneration.
<*your_ip*>.qip	Contains all the required information about the IP component to integrate and compile the IP component in the Intel Quartus Prime software.
<*your_ip*>.sopcinfo	Describes the connections and IP component parameterizations in your Platform Designer system. You can parse its contents to get requirements when you develop software drivers for IP components.
<*your_ip*>.csv	Contains information about the upgrade status of the IP component.
<*your_ip*>.bsf	A Block Symbol File (.**bsf**) representation of the IP variation for use in Intel Quartus Prime Block Diagram Files (**.bdf**).
<*your_ip*>.spd	Required input file for `ip-make-simscript` to generate simulation scripts for supported simulators. The **.spd** file contains a list of files generated for simulation, along with information about memories that you can initialize.
<*your_ip*>.ppf	The Pin Planner File (**.ppf**) stores the port and node assignments for IP components created for use with the Pin Planner.
<*your_ip*>_bb.v	You can use the Verilog black-box (**_bb.v**) file as an empty module declaration for use as a black box.
<*your_ip*>_inst.v or _inst.vhd	HDL example instantiation template. You can copy and paste the contents of this file into your HDL file to instantiate the IP variation. This IP core does not support VHDL. However, the Intel Quartus Prime Pro Edition software generates the `_inst.vhd` file.
<*your_ip*>.v	HDL files that instantiate each submodule or child IP core for synthesis or simulation.
mentor/	Contains a ModelSim* script `msim_setup.tcl` to set up and run a simulation.
synopsys/vcs/ synopsys/vcsmx/	Contains a shell script `vcs_setup.sh` to set up and run a VCS* simulation. Contains a shell script `vcsmx_setup.sh` and `synopsys_ sim.setup` file to set up and run a VCS MX* simulation.
cadence/	Contains a shell script `ncsim_setup.sh` and other setup files to set up and run an NCSim* simulation.
xcelium/	Contains a shell script `xcelium_setup.sh` to set up and run simulation.
submodules/	Contains HDL files for the IP core submodules.
<child IP cores>/	For each generated child IP core directory, Platform Designer generates `synth/` and `sim/` sub-directories.

Related Information

Interlaken IP Core (2nd Generation) Design Example User Guide

2.3. Specifying the IP Core Parameters and Options

The IP parameter editor allows you to quickly configure your custom IP variation. Perform the following steps to specify IP core options and parameters in the Intel Quartus Prime Pro Edition software.

The Interlaken IP core is not supported in Platform Designer. You must use the IP Catalog accessible from the Intel Quartus Prime Pro Edition **Tools** menu. The Interlaken IP core does not support VHDL simulation models. You must specify the Verilog HDL for both synthesis and simulation models.

1. In the Intel Quartus Prime Pro Edition software, click **File ➤ New Project Wizard** to create a new Intel Quartus Prime project, or **File ➤ Open Project** to open an existing Intel Quartus Prime project. The wizard prompts you to specify a device. Select **Stratix 10 (GX/SX/MX)** as your target device.
2. In the IP Catalog (**Tools ➤ IP Catalog**), locate and double-click **Interlaken (2nd Generation) Intel FPGA IP**. The **New IP Variant** window appears.
3. Specify a top-level name for your custom IP variation. The parameter editor saves the IP variation settings in a file named <*your_ip*>.ip.
4. Click **Create**. The parameter editor appears.
5. On the **IP** tab, specify the parameters and options for your IP variation, including one or more of the following. Refer to *Parameter Settings* for information about specific IP core parameters.
   - Specify parameters defining the IP core functionality, port configurations, and device-specific features.
   - Specify options for processing the IP core files in other EDA tools.
6. Click **Generate HDL**. The **Generation** dialog box appears.
7. Specify output file generation options, and then click **Generate**. The IP variation files generate according to your specifications.
8. Optionally, click **Generate Example Design** tab in the parameter editor to generate a demonstration testbench and example design for your IP core variation.

   Note: To generate the demonstration testbench and example design, you must specify Verilog HDL for both synthesis and simulation models.
9. Click **Finish**. The parameter editor adds the top-level .ip file to the project automatically. If you are prompted to manually add the .ip file to the project, click **Project ➤ Add/Remove Files in Project** to add the file.
10. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports.

Related Information

- Interlaken IP Core (2nd Generation) Design Example User Guide
  Describes a simulating testbench and a hardware example design that supports compilation and hardware testing, to help you understand usage.
- Parameter Settings on page 15

2.4. Simulating the IP Core

You can simulate your Interlaken IP core variation using any of the vendor-specific IEEE encrypted functional simulation models which are generated in the new `<instance name>/sim/<simulator>` subdirectory of your project directory.

The Interlaken IP core supports the Synopsys VCS, Mentor Graphics Modelsim-SE*, and Cadence NCSim and Xcelium Parallel simulators. The Interlaken IP core generates only a Verilog HDL simulation model and testbench. The IP core parameter editor appears to offer you the option of generating a VHDL simulation model, but this IP core does not support a VHDL simulation model or testbench.

For more information about functional simulation models for Intel FPGA IP cores, refer to the *Simulating Intel FPGA Designs* chapter in *Quartus Prime Pro Edition Handbook Volume 3: Verification*.

Related Information

Simulating Intel FPGA Designs

2.5. Compiling the Full Design and Programming the FPGA

You can use the **Start Compilation** command on the **Processing** menu in the Intel Quartus Prime software to compile your design. After successfully compiling your design, program the targeted Intel device with the Programmer and verify the design in hardware.

Related Information

- Programming Intel FPGA Devices
- Design Compilation

2.6. Integrating Your IP Core in Your Design

2.6.1. Pin Assignment

When you integrate your IP core instance in your design, you must make appropriate pin assignments. You do not need to specify pin assignments for simulation. However, you should make the pin assignments before you compile, to provide direction to the Fitter and to specify the signals that should be assigned to device pins. While compiling the IP core alone, you can create virtual pins to avoid making specific pin assignments for top-level signals. When you are ready to map the design to hardware, you can change to the correct pin assignments.

2.6.2. Adding the External PLL

The Interlaken (2nd Generation) IP core variations that target an Intel Stratix 10 L-Tile or H-Tile device require an external TX transceiver PLL to drive the TX transceiver clock, in order to compile and to function correctly in hardware. In many cases, the same PLL can be shared with other transceivers in your design.

You can create an external transceiver PLL from the IP Catalog:

- Select **L-Tile/H-Tile Transceiver ATX PLL Intel Stratix 10 FPGA IP**.
- In the parameter editor, set the following parameter values:
  - Set **PLL output frequency** to one half the per-lane data rate of the IP core variation.
  - Set **PLL auto mode reference clock frequency (integer)** to the value you select for the transceiver reference clock frequency (`pll_ref_clk`) parameter in the Interlaken (2nd Generation) IP parameter editor.
  - Set **VCCR_GXB and VCCT_GXB Supply Voltage for the transceiver** to the same value you specify in the Interlaken (2nd Generation) IP parameter editor.

You must connect `tx_serial_clock` output from the ATX PLL to `tx_serial_clk` input of your Interlaken (2nd Generation) IP core.

3. Parameter Settings

You customize the Interlaken IP core by specifying parameters in the IP parameter editor.

Table 8. Interlaken IP Core Parameter Settings: IP Tab

Parameter	Supported Values	Default Setting	Description
General			
Meta frame length (words)	64-8192 words	2048	This parameter specifies the length of the meta frame, in 64-bit (8-byte) words. You must enter this paramter value in power of two. For example, 64, 128, 256 etc. Smaller values for this parameter shorten the time to achieve lock. Larger values reduce overhead while transferring data, after lock is achieved.
Number of lanes	4, 6, 12	12	This parameter specifies the number of lanes available for Interlaken communication. The Interlaken IP core supports various combinations of number of lanes and lane rates. Ensure that your parameter settings specify a supported combination. Refer to *Table: IP Core Supported Combinations of Number of Lanes and Data Rate*
Data rate	6.25, 10.3125, 12.5, 25.3, 25.8 and 26.5625(1) Gbps	10.3125 Gbps	This parameter specifies the data rate on each lane. All lanes have the same data rate (lane rate). The Interlaken IP core supports various combinations of number of lanes and lane rates. Ensure that your parameter settings specify a supported combination. Refer to *Table: IP Core Supported Combinations of Number of Lanes and Data Rate*
Transceiver reference clock frequency	Multiple	412.5 MHz	This parameter specifies the expected frequency of the `pll_ref_clk` input clock.

continued...

(1) This data rate is only available when you select PAM4 option for **XCVR Mode** parameter in Intel Stratix 10 E-Tile variations.

<table>
<tr><th>Parameter</th><th>Supported Values</th><th>Default Setting</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Data Rate per Lane (Gbps)</th><th>Valid Frequencies (MHz)</th></tr>
<tr><td>10.3125</td><td>206.25, 257.8125, 322.265625, 412.5, 515.625, 644.53125</td></tr>
<tr><td>12.5, 6.25</td><td>156.25, 195.3125, 250, 312.5, 390.625, 500, 625</td></tr>
<tr><td>25.3</td><td>126.4, 158.0, 197.5, 252.8, 320.0, 395.0, 486.153846(2), 505.6</td></tr>
<tr><td>25.8</td><td>159.135802, 201.40625, 250.291262, 322.25, 402.8125, 495.769231(2) 500.582524</td></tr>
<tr><td>26.5625
Note: To obtain 6x53.125 Gbps speed in PAM4 mode, you must select 12x26.5625 Gbps combination in Intel Quartus Prime Pro Edition 18.0.1</td><td>156.25, 210.813492, 312.5, 390.62, 491.898148</td></tr>
<tr><td colspan="2">If the actual frequency of the <code>pll_ref_clk</code> input clock does not match the value you specify for this parameter, the design fails in both simulation and hardware.</td></tr>
</table>
</td></tr>
<tr><td>Enable M20K ECC support</td><td>On/Off</td><td>Off</td><td>This parameter specifies whether your Interlaken IP core variation supports the ECC feature in the M20K memory blocks that are configured as part of the IP core.
You can turn this parameter on to enable single-error correct, double-adjacent-error correct, and triple-adjacent-error detect ECC functionality in the M20K memory</td></tr>
</table>

continued...

(2) Only available in NRZ mode of Intel Stratix 10 E-Tile device variations

Parameter	Supported Values	Default Setting	Description
			blocks configured in your IP core. This feature enhances data reliability but increases latency and resource utilization.
Enable Native XCVR PHY ADME	On/Off	Off	This parameter specifies whether your Interlaken IP core variation supports the ADME feature. This parameter exposes debugging features of the Intel Stratix 10 Native PHY IP core that specifies the transceiver settings in the Interlaken IP core.
In-Band Flow Control			
Include in-band flow control functionality	On/Off	Off	This parameter specifies whether your Interlaken IP core includes an in-band flow control block.
Number of calender pages	1, 2, 4, 8, and 16	1	This parameter specifies the number of 16-bit pages of in-band flow control data that your Interlaken IP core supports.This parameter is available if you turn on **Include in-band flow control functionality**. Each 16-bit calendar page includes 16 in-band flow control bits. The application determines the interpretation of the in-band flow control bits. The IP core supports a maximum of 256 channels with in-band flow control. If your design requires a different number of pages, select the lowest supported number of pages which is larger than the number required, and ignore any unused pages. For example, if your configuration requires three in-band flow control calendar pages, you can set this parameter to 4 and use pages 3, 2, and 1 while ignoring page 0.
Transceiver Settings			
Transceiver Tile	L-Tile, H-Tile, E-Tile	H-Tile	Specifies the transceiver tile on your target Intel Stratix 10 device. The **Device** setting of the Intel Quartus Prime Pro Edition project in which you generate the IP core determines the transceiver tile type.
XCVR Mode	NRZ, PAM4	NRZ	Specifies the transceiver mode. This parameter is available only in IP core variations that target an Intel Stratix 10 E-Tile device.
Tx Scrambler seed	-	0x3ab1278890105cd	This parameter specifies the initial scrambler state. If a single Interlaken IP Core is configured on your device, you can use the default value of this parameter.

continued...

Parameter	Supported Values	Default Setting	Description
			If multiple Interlaken IP Cores are configured on your device, you must use a different initial scrambler state for each IP core to reduce crosstalk. Try to select random values for each Interlaken IP core, such that they have an approximately even mix of ones and zeros and differ from the other scramblers in multiple spread out bit positions.
VCCR_GXB and VCCT_GXB supply voltage for the Transceivers	1.0V, 1.1V	1.0V	This parameter specifies the VCCR_GXB and VCCT_GXB transceiver supply voltage. Set this parameter value to 1.1V for 25.3 and 25.8 Gbps data rate.
User Data Transfer Interface			
Transfer mode selection	Interleaved and Packet	Interleaved	This parameter specifies whether the Interlaken transmitter expects incoming traffic to the TX user data transfer interface to be interleaved or packet based.

Related Information

- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Features on page 3
  For more information on IP core supported combinations of lanes and data rate.
- Intel Stratix 10 E-Tile Transceiver PHY User Guide

4. Functional Description

The Interlaken IP core provides the functionality described in the *Interlaken Protocol Specification, Revision 1.2*.

4.1. Interfaces

The Interlaken IP core supports the following interfaces:

- **User Data Transfer Interface**

  The user data transfer interface, also known as application interface, provides up to 256 logical channels of communication to and from the Interlaken link. This interface is similar to the Avalon-ST interface which supports data bursts or packets, which are carried in the Interlaken MetaFrame Payload.

- **Interlaken Interface**

  The Interlaken interface complies with the *Interlaken Protocol Specification, Revision 1.2*. It is the high-speed transceiver interface to an Interlaken link.

- **Out-of-Band Flow Control Interface**

  The optional out-of-band flow control interface conforms to the out-of-band requirements in *Section 5.3.4.2, Out-of-Band Flow Control, of the Interlaken Protocol Specification, Revision 1.2*.

- **Management Interface**

  The management interface provides access to the Interlaken IP core internal status and control registers. This interface does not provide access to the hard PCS registers on the device. This interface complies with the Avalon Memory-Mapped (Avalon-MM) specification defined in the *Avalon Interface Specifications*.

- **Transceiver Control Interfaces**

  The Interlaken IP core provides several interfaces to control the transceiver. The transceiver control interfaces in your Interlaken IP core variation depend on the device family the variation targets. The Interlaken IP core supports the following transceiver control interfaces:

  — **External PLL Interface**

    The Interlaken IP core variations that target an Intel Stratix 10 L- or H-Tile device require an external transceiver PLL to function correctly in hardware. The Interlaken IP core variations that target an Intel Stratix 10 E-Tile device include a PLL and do not require an external PLL.

  — **Transceiver Reconfiguration Interface**

    The Intel Stratix 10 transceiver reconfiguration interface provides access to the registers in the embedded Intel Stratix 10 Native PHY IP core. This interface provides direct access to the hard PCS registers on the device. This interface complies with the Avalon Memory-Mapped (Avalon-MM) specification defined in the *Avalon Interface Specifications*.

Related Information

- Interlaken Protocol Specifications
- Avalon Interface Specifications
- Interface Signals on page 38
- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide

4.2. IP Core Clocks

Table 9. Interlaken IP Core Clocks

Clock Name	Device	Direction	Description
pll_ref_clk	Intel Stratix 10 H-, L- and E-Tile	Input	Reference clock for the RX CDR PLL in IP core variations that target an Intel Stratix 10 device.
tx_serial_clk[NUM_LANES-1:0]	Intel Stratix 10 H- and L-Tile	Input	Clocks for the individual transceiver channels in Interlaken IP core variations that target an Intel Stratix 10 device.
rx_usr_clk	Intel Stratix 10 H-, L- and E-Tile	Input	Clock for the receive application interface.
tx_usr_clk	Intel Stratix 10 H-, L- and E-Tile	Input	Clock for the transmit application interface.
mm_clk	Intel Stratix 10 H-, L- and E-Tile	Input	Management clock for Interlaken IP core register access.
reconfig_clk	Intel Stratix 10 H-, L- and E-Tile	Input	Management clock for Intel Stratix 10 hard PCS register access, including access for Intel Stratix 10 transceiver reconfiguration and testing features.
clk_tx_common	Intel Stratix 10 H-, L- and E-Tile	Output	Transmit PCS common lane clock driven by the SERDES transmit PLL.
clk_rx_common	Intel Stratix 10 H-, L- and E-Tile	Output	Receive PCS common lane clock driven by the CDR in transceiver.
mac_clkin	Intel Stratix 10 E-Tile (PAM4 only)	Input	This signal must be driven by a PLL and shares the same PLL output clock.

Related Information

- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide

4.3. Interleaved and Packet Modes

You can configure the Interlaken IP core to accept interleaved data transfers from the application on the TX user data transfer interface, or to not accept interleaved data transfers on this interface. If the IP core can accept interleaved data transfers, it is in Interleaved mode. If the IP core does not accept interleaved data transfers, it is in Packet mode. The value you specify for the **Transfer mode selection** parameter in the IP core parameter editor determines the IP core transmit mode.

In Packet mode, the Interlaken IP core performs *Optional Scheduling Enhancement based on Section 5.3.2.1.1 of the Interlaken Protocol Specification, Revision 1.2*. The IP core ignores the `itx_sob` and `itx_eob` signals. Instead, the IP core performs optional enhanced scheduling based on the settings of `BurstMax` and `BurstMin`.

In Interleaved mode, you can achieve full capability of channelization and per-channel flow control offered by the Interlaken IP core. In this mode, you can interleave bursts of different channels. It allows more efficient use of bandwidth. The Interlaken IP core inserts burst control words on the Interlaken link based on the `itx_sob` and `itx_eob` inputs. The internal optional enhanced scheduling is disabled and the BurstMax and BurstMin values are ignored. BurstShort is still in effect. To avoid overflowing the transmit FIFO, you should not send a burst that is longer than 1024 bytes.

In Interleaved mode or in Packet mode, the Interlaken IP core is capable of accepting non-interleaved data on the TX user data transfer interface (`itx_din_words`). However, if the IP core is in Interleaved mode, the application must drive the `itx_sob` and `itx_eob` inputs correctly.

In Interleaved mode or in Packet mode, the Interlaken IP core can generate interleaved data transfers on the RX user data transfer interface (`irx_dout_words`). The application must be able to accept interleaved data transfers if the Interlaken link partner transmits them on the Interlaken link. In this case, the Interlaken link partner must send traffic in Interleaved mode that confirms with the Interlaken IP core `BurstShort` value.

Note: The transmitter (link partner) needs to send packets with a minimum packet size of 64 bytes.

4.4. High Level System Flow

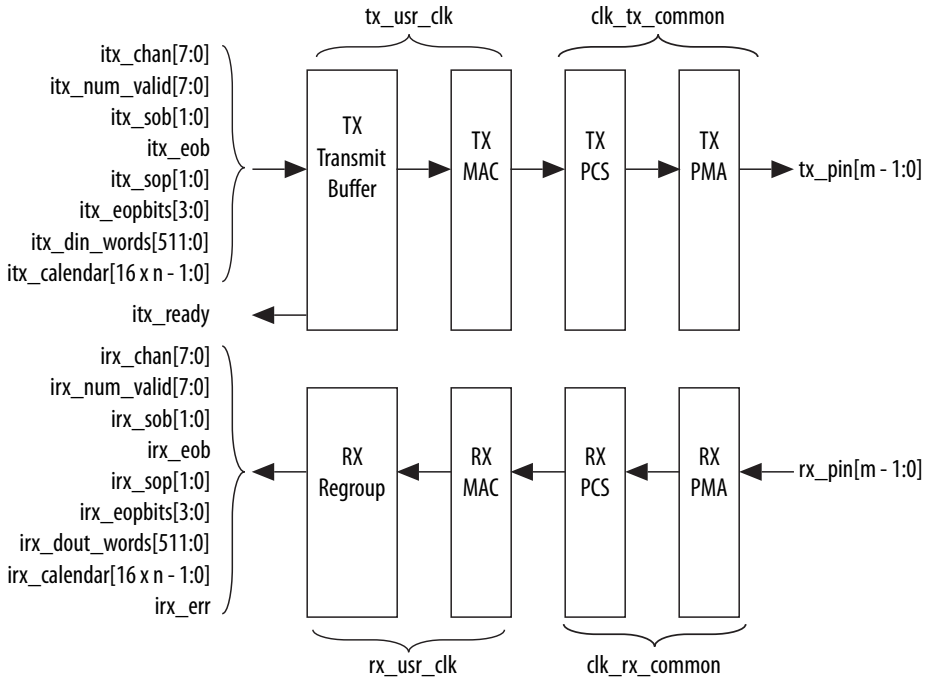
The Interlaken IP core consists of two paths:

- Interlaken TX path
- Interlaken RX path

Each path includes MAC, PCS, and PMA blocks. The PCS blocks are implemented in hard IP.

Figure 5. Interlaken IP Core Block Diagram

The figure illustrates the 8-word data transfer scenario.



Related Information

- Avalon Interface Specifications
- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide

4.4.1. Interlaken TX Path

The Interlaken IP core accepts application data from up to 256 channels and combines it into a single data stream in which data is labeled with its source channel. The Interlaken TX MAC and PCS blocks format the data into protocol-compliant bursts and insert Idle words where required.

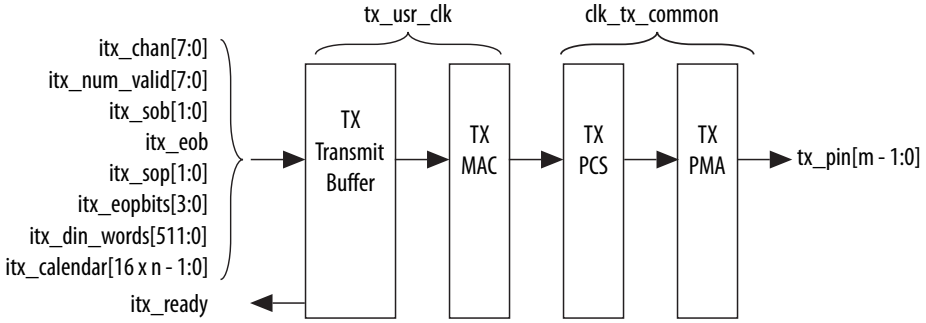
4.4.1.1. Transmit Path Blocks

The Interlaken IP core transmit data path has the following four main functional blocks:

- TX Transmit Buffer
- TX MAC
- TX PCS
- TX PMA

Figure 6. Interlaken IP Core Transmit Path Blocks

The figure illustrates the 8-word data transfer scenario.



TX Transmit Buffer

The Interlaken IP core TX transmit buffer aligns the incoming user application data, `itx_data` in the IP core internal format.

TX MAC

The Interlaken IP core TX MAC performs the following functions:

- Inserts burst and idle control words in the incoming data stream. Burst delineation allows packet interleaving in the Interlaken protocol.
- Performs flow adaption of the data stream, repacking the data to ensure the maximum number of words is available on each valid clock cycle.
- Calculates and inserts CRC24 bits in all burst and idle words.
- Inserts calendar data in all burst and idle words, if you configure in-band flow control.
- Stripes the data across the PCS lanes. Configurable order, default is MSB of the data goes to lane 0.
- Buffers data between the application and the TX PCS block in the TX FIFO buffer. The TX PCS block uses the FIFO buffer to recover bandwidth when the number of words delivered to the transmitter is less than the full width.

TX PCS

In the Intel Stratix 10 L- and H-Tile device variations, TX PCS logic is an embedded hard macro and does not consume FPGA soft logic elements. The FPGA soft logic implements TX PCS in Intel Stratix 10 E-Tile devices. In PAM4 mode, the Intel Stratix 10 E-Tile device variations contain a soft logic transcoder block to work with RS FEC of the TX PMA. The Interlaken IP core TX PCS block performs the following functions for each lane:

- Inserts the meta frame words in the incoming data stream.
- Calculates and inserts the CRC32 bits in the meta frame diagnostic words.
- Scrambles the data according to the scrambler seed and the protocol-specified polynomial.
- Performs 64B/67B encoding.

TX PMA

The Interlaken IP core TX PMA serializes the data and sends it out on the Interlaken link. TX PMA contains RS FEC block in PAM4 mode of Intel Stratix 10 E-Tile devices.

Note: Normal operation of Interlaken (2nd Generation) Intel FPGA IP produces skew of 131 UI or smaller. There is a theoretical chance that on a given reset, channel-to-channel skew can be as much as 259 UI. The Interlaken (2nd Generation) Intel FPGA IP supports significantly more than 259 UI skew, as can many products in the market. If your Interlaken receiver cannot tolerate a skew beyond 134 UI transmit skew in the Interlaken interop guide, please contact mySupport.

Pin out your Interlaken IP core to exist within a single E-Tile. If you have to pin out your core across multiple E-Tiles, please contact mySupport.

4.4.1.2. In-Band Calendar Bits on Transmit Side

When you turn on **Include in-band flow control functionality**,

- The `itx_calendar` input signal supports in-band flow control. It is synchronous with `tx_usr_clk`, but does not align with the packets on the user data interface. The Interlaken IP core reads the `itx_calendar` bits and encodes them in control words (Burst control words and Idle control words) opportunistically.
- The Interlaken IP core transmits each page of the `itx_calendar` bits on the Interlaken link in a separate control word, starting with the most significant page and working through the pages, in order, to the least significant page.
- The Interlaken IP core fills each flow control bit in each control word with the value of 1.

If you hold all the calendar bits at one, you indicate an XON setting for each channel. You should set the calendar bits to 1 to indicate that the Interlaken link partner does not need to throttle the data it transfers to the Interlaken IP core. Set this value by default if you choose not to use the in-band flow control feature of the Interlaken IP core. If you decide to turn off any channel, you must drive the corresponding bits of `itx_calendar` with zero (the XOFF setting) for that channel.

Consider an example where the number of calendar pages is four and itx_calendar bits are set to the value 64'h1111_2222_3333_4444. In this example, the **Number of calendar pages** parameter is set to four, and therefore the width of the `itx_calendar` signal is 4 x 16 = 64 bits. Each of these bits is a calendar bit. The transmission begins with the page with the value of 16'h1111 and works through the pages in order until the least significant page with the value of 16'h4444.

In this example, four control words are required to send the full set of 64 calendar bits from the `itx_calendar` signal. The Interlaken IP core automatically sets the Reset Calendar bit[56] of the next available control word to the value of one, to indicate the start of transmission of a new set of calendar pages, and copies the most significant page (16'h1111 in this example) to the In-Band Flow Control bits[55:40] of the control word. It maps the most significant bit of the page to the control word bit[55] and the least significant bit of the page to the control word bit[40].

Table 10. Value of Reset Calendar Bit and In-band Flow Control Bits in the Example

The table shows the value of the Reset Calendar bit and the In-Band Flow Control bits in the four Interlaken link control words that transmit the 64'h1111_2222_3333_4444 value of `itx_calendar`.

Control Word	Reset Calendar Bit (bit [56])	In-Band Flow Control Bits (bits [55:40])
First	1	16'b0001000100010001 (16'h1111)
Second	0	16'b0010001000100010 (16'h2222)
Third	0	16'b0011001100110011 (16'h3333)
Fourth	0	16'b0100010001000100 (16'h4444)

For details of the control word format, refer to the *Interlaken Protocol Specification,* Revision 1.2.

The IP core supports `itx_calendar` widths of **1**, **2**, **4**, **8**, and **16** 16-bit calendar pages. You configure the width in the IP core parameter editor.

By convention, in a standard case, each calendar bit corresponds to a single data channel. However, the IP core assumes no default usage. You must map the calendar bits to channels or link status according to your specific application needs. For example, if your design has 64 physical channels, but only 16 priority groups, you can use a single calendar page and map each calendar bit to four physical channels. As another example, for a different application, you can use additional calendar bits to pass quality-of-service related information to the Interlaken link partner.

If your application flow-controls a channel, you are responsible for dropping the relevant packet. Intel supports the transfer of the `itx_calendar` values you provide without examining the data that is affected by in-band flow control of the Interlaken link.

4.4.1.3. Transmit User Data Interface Examples

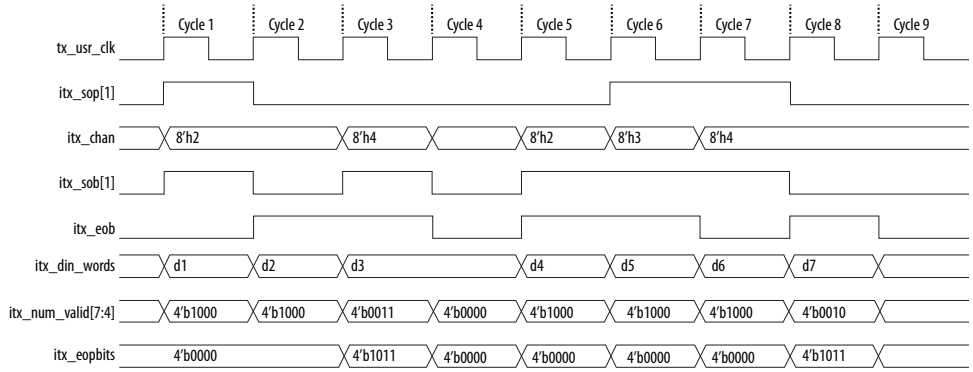
The following examples illustrate how to use the Interlaken IP core TX user data interface:

Interleaved Mode Example

In Interleaved Mode, you are responsible for scheduling the burst. You need to drive an extra pair of signals, Start of Burst (SOB) and End of Burst (EOB), to indicate the burst boundary. You can send the traffic in packet order or interleaved order, as long as you set the SOB and EOB flags correctly to establish the data boundaries.

Figure 7. Packet Transfer on Transmit Interface in Interleaved Single Segment Mode

The figure shows the timing diagram for an interleaved data transfer (8-word) in Interleaved mode.



This example illustrates the expected behavior of the Interlaken IP core application interface transmit signals during data transfers from the application to the IP core on the TX user data transfer interface in interleaved single segment mode. Since only the single segment mode is supported in interleaved mode, the `itx_sob[1]` and `itx_num_valid[7:4]` are valid signals.

In cycle 1, the application asserts `itx_sop[1]` and `itx_sob[1]`, indicating that this cycle is both the start of the burst and the start of the packet. The value the application drives on `itx_chan` indicates the data originates from channel 2.

In cycle 2, the application asserts `itx_eob`, indicating the data the application transfers to the IP core in this clock cycle is the end of the burst. (`itx_chan` only needs to be valid when `itx_sob[1]` or `itx_sop[1]` is asserted). `itx_num_valid[7:4]` indicates all eight words are valid. However, the data in this cycle is not end of packet data. The application is expected to transfer at least one additional data burst in this packet, possibly interleaved with one or more bursts in packets from different data channels.

Cycle 3 is a short burst with both `itx_sob[1]` and `itx_eob` asserted. The application drives the value of three on `itx_num_valid[7:4]` to indicate that three words of the eight-word `itx_din_words` data bus are valid. The data is packed in the most significant words of `itx_din_words`.The application drives the value of 4'b1011 on `itx_eopbits` to indicate that the data the application transfers to the IP core in this cycle are the final words of the packet, and that in the final word of the packet, only three bytes are valid data. The value the application drives on `itx_chan` indicates this burst originates from channel 4.

In cycle 4, the `itx_num_valid[7:4]` signal has the value of zero, which means this cycle is an idle cycle.

In cycle 5, the application sends another single-cycle data burst from channel 2, by asserting `itx_sob[1]` and `itx_eob` to indicate this data is both the start and end of the burst. The application does not assert `itx_sop[1]`, because this burst is not start of packet data. `itx_eopbits` has the value of 4'b0000, indicating this burst is also not end of packet data. This data follows the data burst transfered in cycles 1 and 2, within the same packet from channel 2.

In cycle 6, the application sends a start of packet, single-cycle data burst from channel 3.

In cycles 7 and 8, the application sends a two-cycle data packet in one two-cycle burst. In cycle 8, the second data cycle, the application drives the value of two on `itx_num_valid[7:4]` and the value of 4'b1011 on `itx_eopbits`, to tell the IP core that in this clock cycle, the two most significant words of the data symbol contain valid data and the remaining words do not contain valid data, and that in the second of these two words, only the three most significant bytes contain valid data.

In Interleaved Mode, you can transfer a packet without interleaving as long as the channel number does not toggle during the same packet transfer. However, you must still assert the `itx_sob` and `itx_eob` signals correctly to maintain the proper burst boundaries.

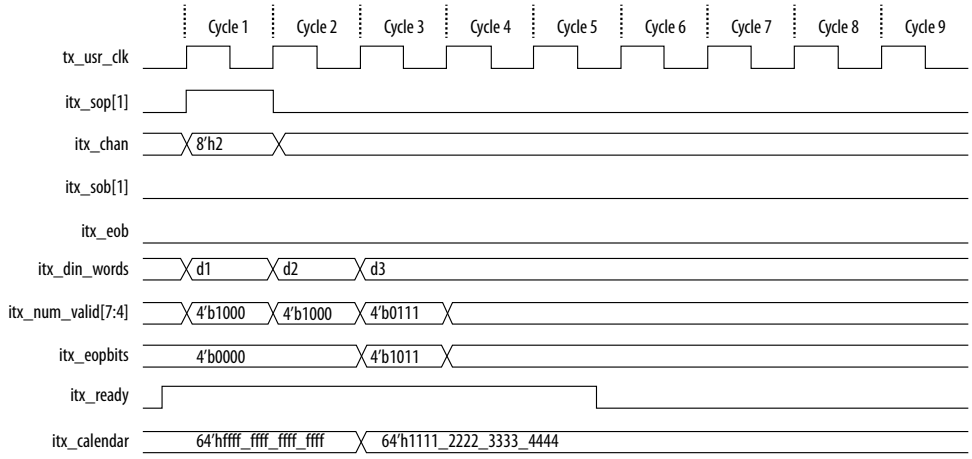
If you do not drive the `itx_sob` and `itx_eob` signals, the Interlaken IP Core does not operate properly and the transmit FIFO may overflow, since in this mode the internal logic is looking for `itx_sob` and `itx_eob` assertion for insertion of proper burst control words.

Packet Mode Operation Example

This example illustrates the expected behavior of the Interlaken IP core application interface transmit signals during a packet transfer in packet mode.

Figure 8. Packet Transfer on Transmit Interface in Packet Mode

The figure illustrates a packet mode data transfer (8-word) of 179 bytes on the transmit interface into the IP core. In this mode, the Interlaken IP core ignores the `itx_sob` and `itx_eob` input signals.



To start a transfer, you assert `itx_sop[1]` when you have data ready on `itx_din_words`. At the following rising edge of the clock, the IP core detects that `itx_sop[1]` is asserted, indicating that the value on `itx_din_words` in the current cycle is the start of an incoming data packet. When you assert `itx_sop[1]`, you must also assert the correct value on `itx_chan` to tell the IP core the data channel source of the data. In this example, the value 2 on `itx_chan` tells the IP core that the data originates from channel number 2.

During the SOP cycle (labeled with data value d1) and the cycle that follows the SOP cycle (labeled with data value d2), you must hold the value of `itx_num_valid[7:4]` at 4'b1000. In the following clock cycle, labeled with data value d3, you must hold the following values on critical input signals to the IP core:

- `itx_num_valid[7:4]` at the value of 4'b0111 to indicate the current data symbol contains seven 64-bit words of valid data.
- `itx_eopbits[3]` high to indicate the current cycle is an EOP cycle.
- `itx_eopbits[2:0]` at the value of 3'b011 to indicate that only three bytes of the final valid data word are valid data bytes.

This signal behavior correctly transfers a data packet with the total packet length of 179 bytes to the IP core, as follows:

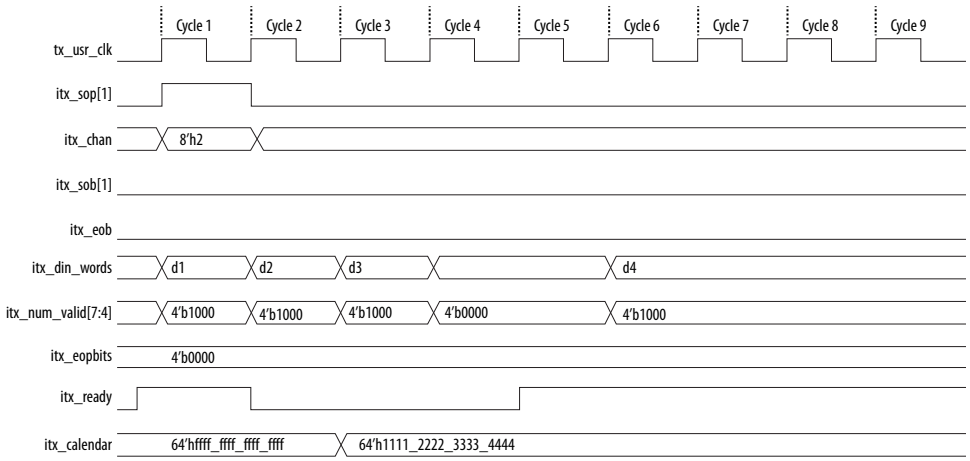
- In the SOP cycle, the IP core receives 64 bytes of valid data (d1).
- In the following clock cycle, the IP core receives another 64 bytes of valid data (d2).
- In the third clock cycle, the EOP cycle, the IP core receives 6 full words (6 x 8 = 48 bytes) and three bytes of valid data, for a total of 51 valid bytes.

The total packet length is 64 + 64 + 51 = 179 bytes.

Back-Pressured Packet Transfer Example

This example illustrates the expected behavior of the Interlaken application interface transmit signals during a packet transfer with back pressure.

Figure 9. Packet Transfer on Transmit Interface with Back Pressure



In this example, the Interlaken IP core accepts the first four data symbols (256 bytes) of a data packet. The clock cycles in which the application transfers the data values d2 and d3 to the Interlaken IP Core are grace-period cycles following the Interlaken IP core's de-assertion of `itx_ready`.

The Interlaken IP core supports up to 4 cycles of grace period, enabling you to register the input data and control signals, as well as the `itx_ready` signal, without changing functionality. The grace period supports your design in achieving timing closure more easily. In any case you must ensure that you hold `itx_num_valid` at the value of 0 when you are not driving data.

You can think of this interface as a FIFO write interface. When `itx_num_valid[7:4]` is nonzero, both data and control information (including `itx_num_valid[7:4]` itself) are written to the transmit side data interface. The `itx_ready` signal is the inverse of a hypothetical FIFO-almost-full flag. When `itx_ready` is high, the Interlaken IP core is ready to accept data. When `itx_ready` is low, you can continue to send data for another 6 to 8 clock cycles of `tx_usr_clk`.

4.4.2. Interlaken RX Path

The Interlaken IP core receives data on the Interlaken link, monitors and removes Interlaken overhead, and provides user data and calendar information to the application. Calendar information is available only if you turn on Include in-band flow control block in the Interlaken parameter editor.

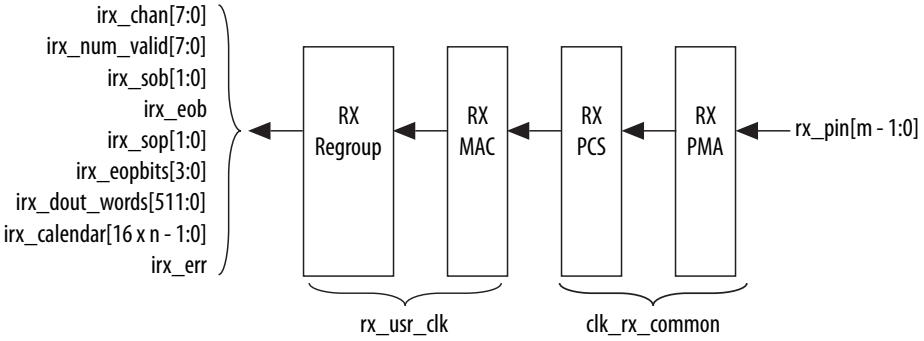
4.4.2.1. Receive Path Blocks

The Interlaken IP core receive data path has the following four main functional blocks:

- RX PMA
- RX PCS
- RX MAC
- RX Regroup Block

Figure 10. Interlaken IP Core Receive Path Blocks

The figure illustrates the 8-word data transfer scenario.



RX PMA

The Interlaken IP core RX PMA deserializes data that the IP core receives on the serial lines of the Interlaken link. RX PMA contains RS FEC block in PAM4 mode of Intel Stratix 10 E-Tile devices.

RX PCS

In Intel Stratix 10 L- and H- Tile device variations, RX PCS logic is an embedded hard macro and does not consume FPGA soft logic elements. The FPGA soft logic implements RX PCS in Intel Stratix 10 E-Tile devices. In PAM4 mode, the Intel Stratix 10 E-Tile device variations contain a soft logic transcoder block to work with RS FEC of the RX PMA. The Interlaken IP core RX PCS block performs the following functions to retrieve the data:

- Detects word lock and word synchronization.
- Checks running disparity.
- Reverses gear-boxing and 64/67B encoding.
- Descrambles the data.
- Delineates meta frame boundaries.
- Performs CRC32 checking.
- Sends lane status information to the calendar and status blocks, if **Include in-band flow control functionality** is turned on.

RX MAC

To recover a packet or burst, the RX MAC takes data from each of the PCS lanes and reassembles the packet or burst. The Interlaken IP core RX MAC performs the following functions:

- Data de-striping, including lane alignment and burst assembly from the PCS lanes.
- CRC24 validation.
- Calendar recovery, if **Include in-band flow control functionality** is turned on.

RX Regroup Block

The Interlaken IP core RX regroup block translates the IP core internal data format to the outgoing user application data `irx_data` format.

4.4.2.2. In-Band Calendar Bits on Receive Side

The Interlaken IP core receiver logic decodes incoming control words (both Burst control words and Idle control words) on the incoming Interlaken link. If you turn on **Include in-band flow control functionality**, the receiver logic extracts the calendar pages from the In-Band Flow Control bits and assembles them into the `irx_calendar` output signal. If you turn off **Include in-band flow control functionality**, the IP core sets all the bits of `irx_calendar` to the value of 1, indicating that the IP core is not flow controlling the incoming data on the Interlaken link.

The Interlaken IP core receives the most significant calendar page in a control word with the Reset Calendar bit set, indicating the beginning of the calendar page sequence. The mapping of bits from the control words to the `irx_calendar` output signal is consistent with the mapping of bits from the `itx_calendar` input signal to the control words.

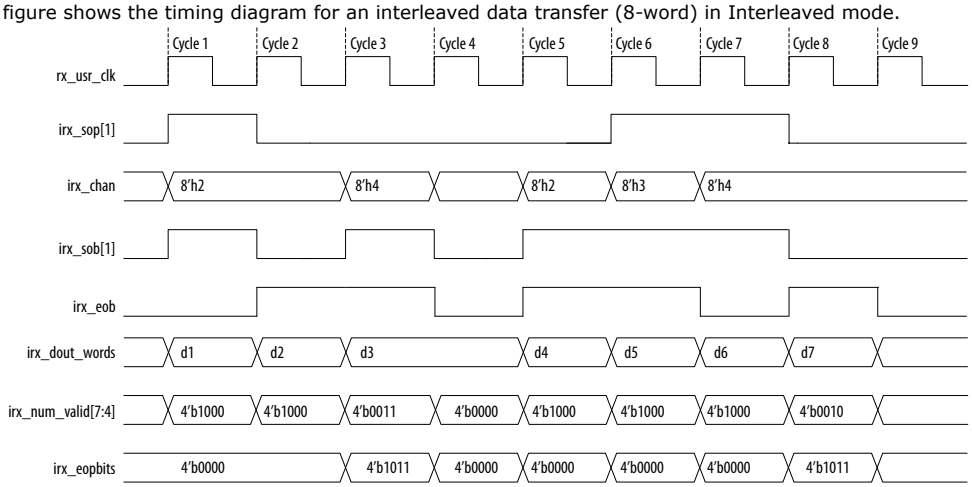
On the RX side, your application is responsible for mapping the calendar pages to the corresponding channels, according to any interpretation agreed upon with the Interlaken link partner application in sideband communication. On the TX side, your application is responsible for throttling the data it transfers to the TX user data transfer interface, in response to the agreed upon interpretation of the `irx_calendar` bits.

4.4.2.3. Receive User Data Interface Example

The Interlaken IP Core can generate interleaved data transfers on the RX user data transfer interface. The IP core always toggles the `irx_sob` and `irx_eob` signals to indicate the beginning of the burst and end of the burst. In single segment mode, only `irx_sob[1]` toggles. Since only the single segment interleaved mode is supported, the `irx_sob[1]` and `irx_num_valid[7:4]` are the valid signals.

Figure 11. Interlaken IP Core Receiver Side Single Segment Example

The figure shows the timing diagram for an interleaved data transfer (8-word) in Interleaved mode.



This example illustrates the expected behavior of the Interlaken IP core application interface receive signals during data transfers from the IP core to the application on the RX user data transfer interface in interleaved single segment mode.

In cycle 1, the IP core asserts `irx_sop[1]` and `irx_sob[1]`, indicating that this cycle is both the start of the burst and the start of the packet. The first word is MSB aligned at the top. The value the IP core drives on `irx_chan` indicates the data targets channel 2. You must sample `irx_chan` during cycles in which `irx_sop[1]` is asserted. The `irx_chan` output signal is not guaranteed to remain valid for the duration of the burst.

In cycle 2, the IP core asserts `irx_eob`, indicating the data the IP core transfers to the application in this clock cycle is the end of the burst. `irx_num_valid[7:4]` indicates all eight words are valid. However, the data in this cycle is not end of packet data. The IP core transfers at least one additional data burst in this packet, possibly interleaved with one or more bursts in packets that target different data channels.

Cycle 3 is a short burst with both `irx_sop[1]` and `irx_eob` asserted. The IP core drives the value of three on `irx_num_valid[7:4]` to indicate that three words of the eight-word `irx_dout_words` data bus are valid. The data is packed in the most significant words of `irx_dout_words`.The IP core drives the value of 4'b1011 on `irx_eopbits` to indicate that the data the IP core transfers to the application in this cycle are the final words of the packet, and that in the final word of the packet, only three bytes are valid data. The value the IP core drives on `irx_chan` indicates this burst targets channel 4.

In cycle 4, the `irx_num_valid[7:4]` signal has the value of zero, which means this cycle is an idle cycle.

In cycle 5, the IP core sends another single-cycle data burst to channel 2, by asserting `irx_sob[1]` and `irx_eob` to indicate this data is both the start and end of the burst. The IP core does not assert `irx_sop[1]`, because this burst is not start of packet data. `irx_eopbits` has the value of 4'b0000, indicating this burst is also not end of packet data. This data follows the data burst transfered in cycles 1 and 2, within the same packet the IP core is sending to channel 2.

In cycle 6, the IP core sends a start of packet, single-cycle data burst to channel 3.

In cycles 7 and 8, the IP core sends a two-cycle data packet in one two-cycle burst. In cycle 8, the second data cycle, the IP core drives the value of two on `irx_num_valid[7:4]` and the value of 4'b1011 on `irx_eopbits`, to tell the application that in this clock cycle, the two most significant words of the data symbol contain valid data and the remaining words do not contain valid data, and that in the second of these two words, only the three most significant bytes contain valid data.

4.4.2.4. RX Errored Packet Handling

The Interlaken IP Core provides information about errored packets on the RX user data transfer interface through the following output signals:

- `irx_eopbits[3:0]`—If this signal has the value of 4'b0001, an error indication arrived with the packet on the incoming Interlaken link: the EOP_Format field of the control word following the final burst of the packet on the Interlaken link has this value, which indicates an error and EOP.
- `irx_err`— The Interlaken IP Core checks the integrity of incoming packets on the Interlaken link, and reports some packet corruption errors it detects on the RX user data transfer interface in the `irx_err` output signal. The IP core asserts the `irx_err` output signal synchronously with the `irx_eob` signal. The IP core asserts this signal only if it can determine the burst in which the error occurred. If the IP core cannot determine the burst in which the error occurred, it does not assert the `irx_err` in response.

In both cases, the application is responsible for discarding the relevant packet.

The `irx_err` signal reflects CRC24 errors that are associated with a data or control burst and is aligned with `irx_eob`.

The `irx_err` signal indicates where an error occurs. The IP core asserts this signal only if an error occurs in an identifiable burst. Corruption can occur at the SOP of the current packet, in some later cycle in the payload of the current packet, in a packet that is interleaved with the current packet, or in the current EOP cycle. However, the IP core asserts the `irx_err` signal only in a subset of these cases, If the current EOP cycle data is corrupted so badly that the EOP indication is missing, or if an error occurs during an IDLE cycle, the IP core does not assert the `irx_err` signal.

For CRC24 errors, you should use the `crc24_err` status signal, rather than relying on the `irx_err` signal, in the following situations:

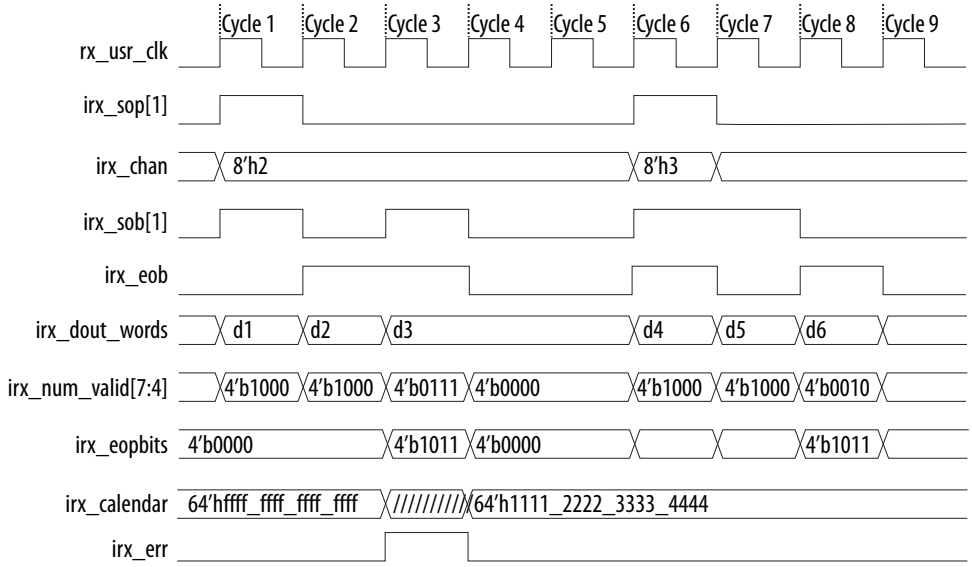
- If you monitor the link when only Idle control words are being received (no data is flowing), you should monitor the real time status signal `crc24_err`.
- If you maintain a count of CRC24 errors, you should monitor the number of times that the real time status signal `crc24_err` is asserted.

4.4.2.4.1. Example With Errors and In-Band Calendar Bits

This example illustrates the expected behavior of the Interlaken IP core application interface receive signals during a packet transfer with CRC or other errors. In the example, the errored packet transfer is followed by two idle cycles and a non-errored packet transfer.

Figure 12. Interlaken IP Core Receiver Side With irx_err Errors

This figure illustrates the attempted transfer of a 179-byte packet on the RX user data transfer interface to channel 2, after the Interlaken IP core receives the packet on the Interlaken link and detects corruption. Following the errored packet, the IP core transfers an uncorrupted packet to channel 3.



In cycle 1, the Interlaken IP core asserts `irx_sop[1]` when data is ready on `irx_dout_words`. When the Interlaken IP core asserts `irx_sop[1]`, it also asserts the correct value on `irx_chan` to tell the application the data channel destination of the data. In this example, the value 2 on `irx_chan` tells the application that the data should be sent to channel number 2.

During the SOP cycle (labeled with data value d1) and the cycle that follows the SOP cycle (labeled with data value d2), the Interlaken IP core holds the value of `irx_num_valid[7:4]` at 4'b1000. In the following clock cycle, labeled with data value d3, the Interlaken IP core holds the following values on critical output signals:

- `irx_num_valid[7:4]` at the value of 4'b0111 to indicate the current data symbol contains seven 64-bit words of valid data.
- `irx_eopbits[3]` high to indicate the current cycle is an EOP cycle.
- `irx_eopbits[2:0]` at the value of 3'b011 to indicate that only three bytes of the final valid data word are valid data bytes.

This signal behavior, in the absence of the `irx_err` flag, would correctly transfer a data packet with the total packet length of 179 bytes from the Interlaken IP core. However, the Interlaken IP core marks the burst as errored by asserting the `irx_err` signal, even though the `irx_eopbits` signal would appear to indicate the packet is valid.

The application is responsible for discarding the errored packet when it detects that the IP core has asserted the `irx_err` signal. Following the corrupted packet, the IP core waits two idle cycles and then transfers a valid 139-byte packet.

4.5. IP Core Reset

The Interlaken IP core variations have a single asynchronous reset, the `reset_n` signal. The Interlaken IP core manages the initialization sequence internally. After you de-assert `reset_n` (raise it after asserting it low), the IP core automatically goes through the entire reset sequence.

Note: Intel recommends that you hold the `reset_n` signal low for at least the duration of eight `mm_clk` cycles, to ensure the reset sequence proceeds correctly.

Following completion of the reset sequence internally, the Interlaken IP core begins link initialization. If your IP core and its Interlaken link partner initialize the link successfully, you can observe the assertion of the lane and link status signals according to the Interlaken specification. For example, you can monitor the `tx_lanes_aligned`, `sync_locked`, `word_locked`, and `rx_lanes_aligned` output status signals.

In Intel Stratix 10 devices, the required wait time from de-asserting the `reset_n` signal to safely accessing the IP core registers is a function of the internal reset controller.

Related Information

- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide

4.6. M20K ECC Support

If you turn on **Enable M20K ECC support** in your Intel Stratix 10 Interlaken IP core variation, the IP core takes advantage of the built-in device support for ECC checking in all M20K blocks configured in the IP core on the device. The feature performs single-error correct, double-adjacent-error correct, and triple-adjacent-error detect ECC functionality in the M20K memory blocks configured in your IP core.

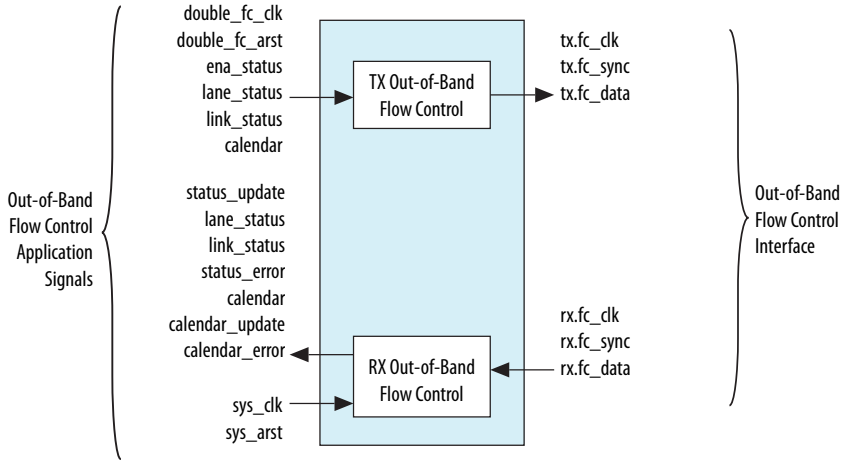
This feature enhances data reliability but increases latency and resource utilization. Without the ECC feature, a single M20K memory block can support a data path width of 40 bits. With the ECC feature, eight of those bits are dedicated to the ECC, and an M20K memory block can support a maximum data path width of 32 bits. Therefore, when M20K ECC support is turned on the IP core configures additional M20K memory blocks. The ECC check adds latency to the path through the memory block, and increases the amount of device memory used by your IP core.

4.7. Out-of-Band Flow Control

The Interlaken IP core includes logic to provide the out-of-band flow control functionality described in the *Interlaken Protocol Specification, Revision 1.2*, Section 5.3.4.2. This optional feature is intended for applications that require transmission rate control.

Figure 13. Out-of-Band Flow Control Block Interface

This figure lists the signals on the four interfaces of the out-of-band flow control block.



The out-of-band flow control block is provided as two separate modules that can be stitched to the Interlaken IP core and user logic. You can optionally instantiate these blocks in your own custom logic. To enable the use of these out-of-band modules, the signals on the far left side of the figure must be connected to user logic, and the signals on the far right side of the figure should be connected to the complementary flow control blocks of the Interlaken link partner.

You must connect the out-of-band flow control receive and transmit interface signals to device pins.

Table 11. Out-of-Band Flow Control Block Clocks

Clock Name	Interface	Direction	Recommended Frequency (MHz)	Description
`rx.fc_clk`	RX Out-of-band	Input	100	Clocks the incoming out-of-band flow control interface signals described in the Interlaken specification. This clock is received from an upstream TX out-of-band flow control block associated with the Interlaken link partner. The recommended frequency for the RX `fc_clk` clock is 100 MHz, which is the maximum frequency allowed by the Interlaken specification.
`tx.fc_clk`	TX Out-of-band	Output	100	Clocks the outgoing out-of-band flow control interface signals described in the Interlaken specification. This clock is generated by the out-of-band flow control block and sent to a downstream RX out-of-band flow control block associated with the Interlaken link partner. The frequency of this clock must be half the frequency of the `double_fc_clk` clock. The recommended frequency for the TX `fc_clk` clock is 100 MHz, which is the maximum frequency allowed by the Interlaken specification.
`sys_clk`	RX Application	Input	200	Clocks the outgoing calendar and status information on the application side of the block. The frequency of this clock must be at least double the frequency of

continued...

Clock Name	Interface	Direction	Recommended Frequency (MHz)	Description
				the RX input clock fc_clk. Therefore, the recommended frequency for the sys_clk clock is 200 MHz.
double_fc_clk	TX Application	Input	200	Clocks the incoming calendar and status information on the application side of the block. The frequency of this clock must be double the frequency of the TX output clock fc_clk. Therefore, the recommended frequency for the double_fc_clk clock is 200 MHz.

The transmit out-of-band flow control interface receives calendar and status information, and transmits flow control clock, data, and sync signals. The TX Out-of-Band Flow Control Interface Signals table describes the transmit out-of-band flow control interface signals specified in the *Interlaken Protocol Specification, Revision 1.2*. The *TX Out-of-Band Flow Control Block Signals for Application Use* table describes the signals on the application side of the TX out-of-band flow control block.

Table 12. TX Out-of-Band Flow Control Interface Signals

Signal Name	Direction	Width (Bits)	Description
tx.fc_clk	Output	1	Output reference clock to a downstream out-of-band RX block. Clocks the fc_data and fc_sync signals. You must connect this signal to a device pin.
tx.fc_data	Output	1	Output serial data pin to a downstream out-of-band RX block. You must connect this signal to a device pin.
tx.fc_sync	Output	1	Output sync control pin to a downstream out-of-band RX block. You must connect this signal to a device pin.

Table 13. TX Out-of-Band Flow Control Block Signals for Application Use

Signal Name	Direction	Width (Bits)	Description
double_fc_clk	Input	1	Reference clock for generating the flow control output clock fc_clk. The frequency of the double_fc_clk clock must be double the intended frequency of the TX fc_clk output clock.
double_fc_arst	Input	1	Asynchronous reset for the out-of-band TX block.
ena_status	Input	1	Enable transmission of the lane status and link status to the downstream out-of-band RX block. If this signal is asserted, the lane and link status information is transmitted on fc_data. If this signal is not asserted, only the calendar information is transmitted on fc_data.
lane_status	Input	Number of Lanes	Lane status to be transmitted to a downstream out-of-band RX block if ena_status is asserted. Width is the number of lanes.
link_status	Input	1	Link status to be transmitted to a downstream out-of-band RX block if ena_status is asserted.
calendar	Input	16	Calendar status to be transmitted to a downstream out-of-band RX block.

The receive out-of-band flow control interface receives input flow control clock, data, and sync signals and sends out calendar and status information. The RX Out-of-Band Flow Control Interface Signals table describes the receive out-of-band flow control

interface signals specified in the *Interlaken Protocol Specification, Revision 1.2*. The *RX Out-of-Band Flow Control Block Signals for Application Use* describes the signals on the application side of the RX out-of-band flow control block.

Table 14. RX Out-of-Band Flow Control Interface Signals

Signal Name	Direction	Width (Bits)	Description
rx.fc_clk	Input	1	Input reference clock from an upstream out-of-band TX block. This signal clocks the fc_data and fc_sync signals. You must connect this signal to a device pin.
rx.fc_data	Input	1	Input serial data pin from an upstream out-of-band TX block. You must connect this signal to a device pin.
rx.fc_sync	Input	1	Input sync control pin from an upstream out-of-band TX block. You must connect this signal to a device pin.

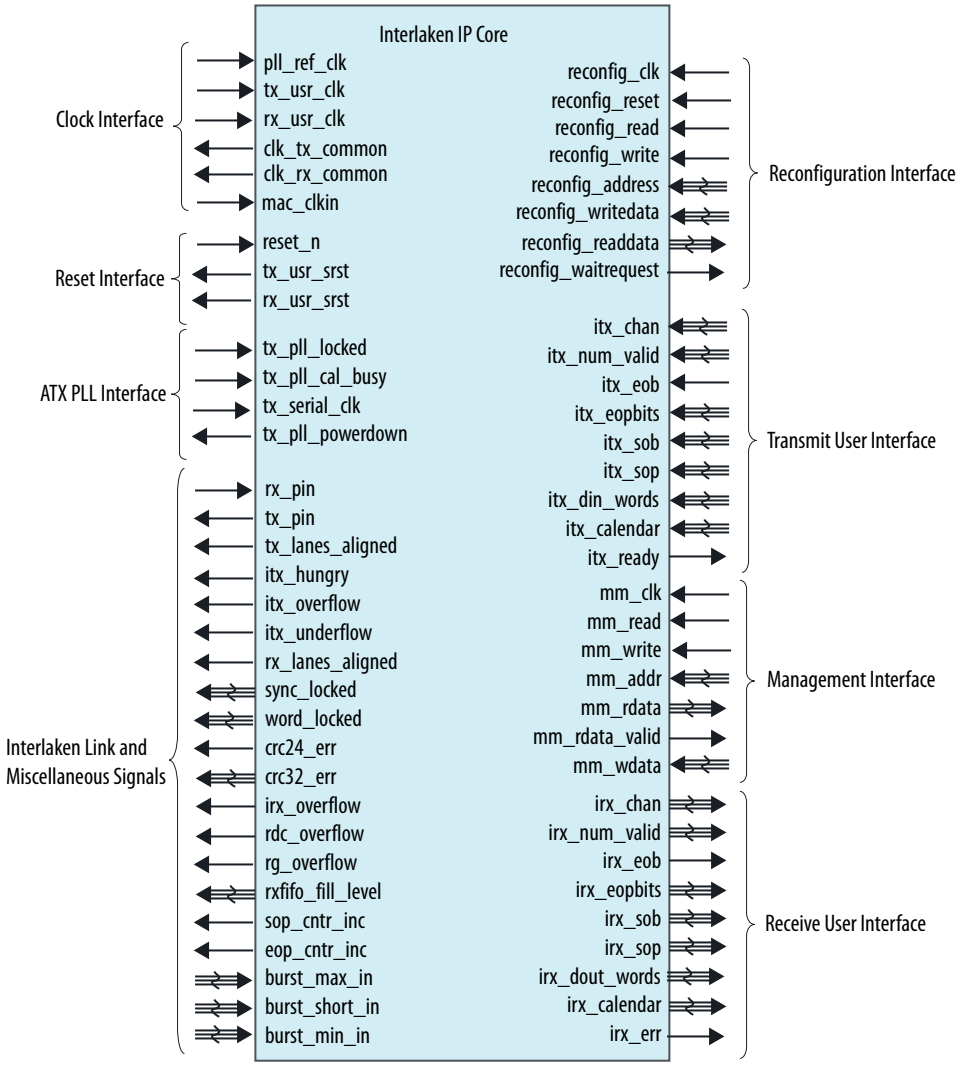
Table 15. RX Out-of-Band Flow Control Block Signals for Application Use

Signal Name	Direction	Width (Bits)	Description
sys_clk	Input	1	Reference clock for capturing RX calendar, lane status, and link status. Frequency must be at least double the frequency of the TX fc_clk input clock.
sys_arst	Input	1	Asynchronous reset for the out-of-band RX block.
status_update	Output	1	Indicates a new value without CRC4 errors is present on at least one of lane_status or link_status in the current sys_clk cycle. The value is ready to be read by the application logic.
lane_status	Output	Number of Lanes	Lane status bits received from an upstream out-of-band TX block on fc_data. Width is the number of lanes.
link_status	Output	1	Link status bit received from an upstream out-of-band TX block on fc_data.
status_error	Output	1	Indicates corrupt lane or link status. A new value is present on at least one of lane_status or link_status in the current sys_clk cycle, but the value has at least one CRC4 error.
calendar	Output	16	Calendar bits received from an upstream out-of-band TX block on fc_data.
calendar_update	Output	1	Indicates a new value without CRC4 errors is present on calendar in the current sys_clk cycle. The value is ready to be read by the application logic.
calendar_error	Output	1	Indicates corrupt calendar bits. A new value is present calendar in the current sys_clk cycle, but the value has at least one CRC4 error.

5. Interface Signals

The IP core communicates with the surrounding design through multiple external signals.

Figure 14. IP Core Interface Signals



5.1. Clock and Reset Interface Signals

Table 16. Clock and Reset Interface Signals

<table>
<tr><th>Signal Name</th><th>Width (Bits)</th><th>I/O Direction</th><th>Description</th></tr>
<tr><td rowspan="7">pll_ref_clk</td><td rowspan="7">1</td><td rowspan="7">Input</td><td>Transceiver reference clock for the RX CDR PLL in IP core variations that target a Intel Stratix 10 device. The sets of valid frequencies vary with the per-lane data rate of the transceivers as following:</td></tr>
<tr><td><table>
<tr><th>Per-Lane Data Rate (Gbps)</th><th>Valid pll_ref_clk Frequencies (MHz)</th></tr>
<tr><td>10.3125</td><td>206.25, 257.8125, 322.265625, 412.5, 515.625, 644.53125</td></tr>
<tr><td>12.5/6.25</td><td>156.25, 195.3125, 250, 312.5, 390.625, 500, 625</td></tr>
<tr><td>25.3</td><td>126.4, 158.0, 197.5, 252.8, 320.0, 395.0, 486.153846(3), 505.6</td></tr>
<tr><td>25.8</td><td>159.135802, 201.40625, 250.291262, 322.25, 402.8125, 495.769231(3), 500.582524</td></tr>
<tr><td>26.5625(4)</td><td>156.25, 210.813492, 312.5, 390.62, 491.898148</td></tr>
</table></td></tr>
<tr><td></td></tr>
<tr><td>tx_usr_clk</td><td>1</td><td>Input</td><td>Transmit side user data interface clock. This signal must be driven by clk_tx_common for maximum performance. The lower frequency of tx_clk increases the latency of data path.</td></tr>
<tr><td>rx_usr_clk</td><td>1</td><td>Input</td><td>Receive side user data interface clock. This signal must be driven by clk_rx_common for maximum performance. The lower frequency of rx_clk increases the latency of data path.</td></tr>
<tr><td>clk_tx_common</td><td>1</td><td>Output</td><td>Transmit PCS common lane clock driven by the SERDES transmit PLL.
The frequency is given by the following equation:
<code>Frequency of clk_tx_common = transceiver data rate / PMA_WIDTH</code>
For example, the clock rate is 322 MHz at 10.3125 Gbps transceiver speed with 32 bits PMA_WIDTH.</td></tr>
<tr><td>clk_rx_common</td><td>1</td><td>Output</td><td>Receive PCS common lane clock driven by CDR in transceiver.
The frequency is given by the following equation:
<code>Frequency of clk_rx_common = transceiver data rate / PMA_WIDTH</code>
For example, the clock rate is 322 MHz at 10.3125 Gbps transceiver speed with 32 bits PMA_WIDTH.</td></tr>
<tr><td>reset_n</td><td>1</td><td>Input</td><td>Active-low asynchronous reset signal.</td></tr>
</table>

continued...

(3) Only available in NRZ mode of Intel Stratix 10 E-Tile device variations

(4) Only available in PAM4 mode of Intel Stratix 10 E-Tile device variations.

Signal Name	Width (Bits)	I/O Direction	Description
tx_usr_srst	1	Output	Transmit side reset output signal. Indicates the transmit side user data interface is resetting. This signal is synchronous with tx_usr_clk.
rx_usr_srst	1	Output	Receive side reset output. Indicates the receive side user data interface is resetting. This signal is synchronous with rx_usr_clk.
mac_clkin	1	Input	This clock signal is only available in IP core variations that target Intel Stratix 10 E-Tile PAM4 device variations. This signal must be driven by a PLL and shares the same PLL output clock.

Related Information

- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide

5.2. Transmit User Interface Signals

Table 17. Transmit User Interface Signals

Signal Name	Width (Bits)	I/O Direction	Description
itx_chan	8	Input	Transmit logic channel number. The IP core supports up to 256 channels. The IP core samples this value only when a bit of itx_sop or itx_sob is high and itx_num_valid has a non-zero value.
itx_num_valid	Variable	Input	Indicates the number of valid 64-bit words in the current packet in the current data symbol. The width of the itx_num_valid depends on the parameter internal_words. • If internal_words=4, then width=3 bits. • If internal_words=8, then width=8 bits. • If internal_words=16, then width=10 bits. If internal_words is equal to 8: • itx_num_valid[7:4] indicates the number of valid words in itx_din_words[511:0]. The value can vary from 4'b0000 to 4'b1000. • itx_num_valid[3:0] indicates the number of valid words in itx_din_words[255:0]. The value must be either 4'b0100 or 4'b0000. When itx_num_valid[3:0] has the value of 4'b0100, you must also hold itx_sop[0] high. When itx_ready is not asserted, it is non-valid cycle, hence the value of itx_num_valid[7:4] and itx_num_valid[3:0] must be set to 4'b0000. In valid cycles, you must set the value of itx_num_valid[7:4] as following: • 4'b1000: if all eight words contain valid data from the current packet. • 4'b0xxx: where xxx indicates the number of valid words that are part of the current packet, if the number is less than eight. Data is always MSB aligned (left aligned). For example, the value of 4'b0111 indicates that word 0 (bit [63:0]) is not valid.
itx_eob	1	Input	Indicates the current data symbol contains the end of the burst (EOB). When itx_eopbits[3] or itx_eopbits[0] is valid, the itx_eob is valid too, but not vice-versa. Whenever parameter TX_PKTMOD_ONLY is set to 0, you must provide this signal. Otherwise, when parameter TX_PKTMOD_ONLY is set to 1, the IP core ignores this signal. You are responsible to comply with the **BurstMax** and **BurstMin** setting.

continued...

Signal Name	Width (Bits)	I/O Direction	Description
itx_eopbits	4	Input	Indicates whether the current data symbol contains the end of a packet (EOP) with or without an error, and specifies the number of valid bytes in the current end-of-packet, non-error 8-byte data word, if relevant. You must set the value of itx_eopbits as following: • 4b'0000: no end of packet, no error. • 4b'0001: Error and end of packet. • 4b'1xxx: End of packet. xxx indicates the number of valid bytes in the final valid 8-byte word of the packet, as following: — 000: all 8 bytes are valid. — 001: 1 byte is valid. — ... — 111: 7 bytes are valid. All other values (4'b01xx, 4'b001x) are undefined. The valid bytes always start in bit positions [63:56] of the final valid data word of the packet.
itx_sob	2 (or 1 for 4-word use case)	Input	Indicates the current data symbol contains the start of a burst (SOB). If the IP core is in Interleaved mode, you are responsible for providing this start of the burst signal. If the IP core is in Packet mode, the IP core ignores this signal. The IP core samples the itx_chan signal during this cycle. This signal has the following valid values: • 2'b00—The current data symbol does not contain the start of a burst. • 2'b10— If itx_sob[1] has the value of 1, the start-of-burst aligns with the most significant byte (byte 63) of the data. • 2'b01— If itx_sob[0] has the value of 1, the start-of-burst aligns with byte 31 of the data. Typically, you use this mode for sending interleaved packets. However, you can still send non-interleaved packets as long as you provide the itx_sob and itx_eob signal values. You are responsible to comply with the **BurstMax** and **BurstMin** setting. If the burst you send is too large, it can overflow the Interlaken transmit buffer.
itx_sop	2 (or 1 for 4-word use case)	Input	Indicates the current data symbol on itx_din_words contains the start of a packet (SOP). This signal has the following valid values: • 2'b00—The current data symbol does not contain the start of a packet. • 2'b10— If itx_sop[1] has the value of 1, the start-of-packet aligns with the most significant byte (byte 63) of the data. • 2'b01— If itx_sop[0] has the value of 1, the start-of-packet aligns with byte 31 of the data.
itx_din_words	Variable	Input	The 64-bit words of input data (one data symbol). The width of the itx_dout_words depends on the parameter internal_words. • If internal_words=4, then width=256 bits. • If internal_words=8, then width=512 bits. • If internal_words=16, then width=1024 bits. When tx_idle or tx_sob is one, the Interlaken IP core ignores matching data word in tx_data. The first and last data word is in [511:448] and [63:0] respectively.
itx_calendar	N * 16	Input	Multiple pages (16 bits per page) of calendar input bits. The IP core copies these bits to the in-band flow control bits in N control words that it sends on the Interlaken link. **N** is the value of the **Number of calendar pages** parameter, which can be any of 1, 2, 4, 8. or 16. This signal is synchronous with tx_usr_clk, although it is not part of the user data transfer protocol.
itx_ready	1	Output	Flow control signal to back pressure transmit traffic. When this signal is high, you can send traffic to the IP core. When this signal is low, you should stop sending traffic to the IP core within one to four cycles. You should provide itx_num_valid only after itx_ready is asserted.

5.3. Receive User Interface Signals

Table 18. Receive User Interface Signals

Signal Name	Width (Bits)	I/O Direction	Description
irx_chan	8	Output	Receive logic channel number. The IP core supports up to 256 channels. The Interlaken IP core samples this value only when a bit of irx_sop or irx_sob is high and irx_num_valid has a non-zero value.
irx_num_valid	Variable	Output	Indicates the number of valid 64-bit words in the current packet in the current data symbol. The width of the irx_num_valid depends on the parameter internal_words. • If internal_words=4, then width=3 bits. • If internal_words=8, then width=8 bits. • If internal_words=16, then width=10 bits. If internal_words is equal to 8: • irx_num_valid[7:4] indicates the number of valid words in irx_din_words[511:0]. The value can vary from 4'b0000 to 4'b1000. • irx_num_valid[3:0] indicates the number of valid words in irx_din_words[255:0]. The value must be either 4'b0100 or 4'b0000. When irx_num_valid[3:0] has the value of 4'b0100, the IP core also holds itx_sop[0] high. For all non-valid cycles, IP core sets the value of irx_num_valid[7:4] and irx_num_valid[3:0]to 4'b0000. In valid cycles, the IP core sets the value of irx_num_valid[7:4] as following: • 4'b1000: if all eight words contain valid data from the current packet. • 4'b0xxx: where xxx indicates the number of valid words that are part of the current packet, if the number is less than eight. Data is always MSB aligned (left aligned). For example, the value of 4'b0111 indicates that word 0 (bit [63:0]) is not valid.
irx_eob	1	Output	Indicates the end of the burst (EOB). This signal toggles in Packet Mode and in Interleaved Mode.
irx_eopbits	4	Output	Indicates whether the current data symbol contains the end of a packet (EOP) with or without an error, and specifies the number of valid bytes in the current end-of-packet, non-error 8-byte data word, if relevant. IP core sets the value of irx_eopbits as following: • 4b'0000: no end of packet, no error. • 4b'0001: Error and end of packet. • 4b'1xxx: End of packet. xxx indicates the number of valid bytes in the final valid 8-byte word of the packet, as following: — 000: all 8 bytes are valid. — 001: 1 byte is valid. — ... — 111: 7 bytes are valid. All other values (4'b01xx, 4'b001x) are undefined. The valid bytes always start in bit positions [63:56] of the final valid data word of the packet.
irx_sob	2 (or 1 for 4-word use case)	Output	Indicates the start of a burst (SOB). This signal toggles in Packet Mode and in Interleaved Mode. This signal has the following valid values:

continued...

Signal Name	Width (Bits)	I/O Direction	Description
			• 2'b00—The current data symbol does not contain the start of a burst. • 2'b10— If `irx_sob[1]` has the value of 1, the start-of-burst aligns with the most significant byte (byte 63) of the data. • 2'b01— If `irx_sob[0]` has the value of 1, the start-of-burst aligns with byte 31 of the data.
`irx_sop`	2 (or 1 for 4-word use case)	Output	Indicates the current data symbol on `irx_dout_words` contains the start of a packet (SOP). This signal has the following valid values: • 2'b00—The current data symbol does not contain the start of a packet. • 2'b10— If `irx_sop[1]` has the value of 1, the start-of-packet aligns with the most significant byte (byte 63) of the data. • 2'b01— If `irx_sop[0]` has the value of 1, the start-of-packet aligns with byte 31 of the data.
`irx_dout_words`	Variable	Output	The 64-bit words of input data (one data symbol). The width of the `itx_din_words` depends on the parameter `internal_words`. • If `internal_words=4`, then width=256 bits. • If `internal_words=8`, then width=512 bits. • If `internal_words=16`, then width=1024 bits. When `rx_idle` or `rx_sob` is one, the matching rx_data words are invalid. The first and last data word is in [511:448] and [63:0] respectively. When `irx_ num_valid` has the value of zero, you should ignore `irx_dout_words`.
`irx_calendar`	N * 16	Output	Multiple pages (16 bits per page) of calendar input bits. The value is the in-band flow control bits from N control words on the incoming Interlaken link. **N** is the value of the **Number of calendar pages** parameter, which can be any of 1, 2, 4, 8. or 16. This signal is synchronous with `rx_usr_clk`, although it is not part of the user data transfer protocol.
`irx_err`	1	Output	Indicates an errored packet. This signal is valid only when `irx_eob` is asserted.

5.4. Management Interface Signals

Table 19. Management Interface Signals

Signal Name	Width (Bits)	I/O Direction	Description
`mm_clk`	1	Input	Management clock. Clocks the register accesses. It is also used for clock rate monitoring and some analog calibration procedures. You must run this clock at a frequency in the range of 100 MHz–125 MHz.
`mm_read`	1	Input	Read access to the register ports.
`mm_write`	1	Input	Write access to the register ports.
`mm_addr`	16	Input	Address to access the register ports.
`mm_rdata`	32	Output	When `mm_rdata_valid` is high, `mm_rdata` holds valid read data.
`mm_rdata_valid`	1	Output	Valid signal for `mm_rdata`.
`mm_wdata`	32	Input	When `mm_write` is high, `mm_wdata` holds valid write data.

5.5. Reconfiguration Interface Signals

Table 20. Reconfiguration Interface Signals

Signal Name	Width (Bits)	I/O Direction	Description
reconfig_clk	1	Input	Intel Stratix 10 transceiver reconfiguration interface clock.
reconfig_reset	1	Input	Active-high synchronous reset. Assert this signal to reset the Intel Stratix 10 transceiver reconfiguration interface.
reconfig_read	1	Input	Read access to the Intel Stratix 10 hard PCS registers.
reconfig_write	1	Input	Write access to the Intel Stratix 10 hard PCS registers.
reconfig_address	RECONF_ADDR+10	Input	Address to access the hard PCS registers. This signal holds both the hard PCS register offset and the transceiver channel being addressed.
reconfig_writedata	32	Input	When reconfig_write is high, reconfig_writedata holds valid write data.
reconfig_readdata	32	Output	After user logic asserts the reconfig_read signal, when the IP core deasserts the signal, reconfig_readdata holds valid read data.
reconfig_waitrequest	1	Output	Busy signal for reconfig_readdata.

Related Information

- Avalon Interface Specifications
- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide

5.6. Interlaken Link and Miscellaneous Signals

Table 21. SERDES Pins

Signal Name	Width (Bits)	I/O Direction	Description
rx_pin	Number of lanes	Input	Each bit represents the differential pair on an RX Interlaken lane.
tx_pin	Number of lanes	Output	Each bit represents the differential pair on a TX Interlaken lane.

Table 22. Real-Time Transmitter Status Signals

Signal Name [5]	Width (Bits)	I/O Direction	Description
tx_lanes_aligned	1	Output	Indicates whether all of the transmitter lanes are aligned and are ready to send traffic.
itx_overflow	1	Output	An error flag indicating that the Transmit buffer is currently overflowing. This signal is asserted for the duration of the overflow condition: it is asserted in the first clock cycle in which the overflow occurs, and remains asserted until the Transmit buffer pointers indicate that no overflow condition exists.
itx_underflow	1	Output	An error flag indicating that the Transmit buffer is currently underflowed. In normal operation, this signal may be asserted temporarily immediately after the Interlaken IP core comes out of reset. It is asserted as a single cycle wide pulse.

Table 23. Real-Time Receiver Status Signals

Signal Name [6]	Width (Bits)	I/O Direction	Description
rx_lanes_aligned	1	Output	Indicates whether all of the receiver lanes are aligned and are ready to receive traffic.
sync_locked	Number of lanes	Output	Receive lane has locked on the remote transmitter Meta Frame. These signals are level signals: all bits are expected to stay high unless a problem occurs on the serial line.
word_locked	Number of lanes	Output	Receive lane has identified the 67-bit word boundaries in the serial stream. These signals are level signals: all bits are expected to stay high unless a problem occurs on the serial line.
crc24_err	1	Output	A CRC24 error flag covering both control word and data word. You can use this signal to count the number of CRC24 errors. This signal is asserted as a single cycle wide pulse.
crc32_err	Number of lanes	Output	An error flag indicating diagnostic CRC32 failures per lane. This signal is asserted as a single cycle wide pulse.
irx_overflow	0	Output	This signal is tied to 0 and it is not used.
rdc_overflow	0	Output	This signal is tied to 0 and it is not used.
rg_overflow	1	Output	An error flag indicating that the Reassembly FIFO is currently overflowed. The Reassembly FIFO is the receiver FIFO that feeds directly to the user data interface.
rxfifo_fill_level	RXFIFO_ADDR_ WIDTH	Output	The fill level of the Reassembly FIFO, in units of 64-bit words. The width of this signal is the value of the RXFIFO_ADDR_WIDTH parameter, which is 12 by default. You can use this signal to monitor when the RX Reassembly FIFO is empty.

continued...

(5) Synchronous with tx_usr_clk.

(6) Synchronous with rx_usr_clk.

Signal Name [6]	Width (Bits)	I/O Direction	Description
sop_cntr_inc	1	Output	A pulse indicating that the IP core receiver user data interface received a start-of- packet (SOP). You can use this signal to increment a count of SOPs the application observes on the receive interface.
eop_cntr_inc	1	Output	A pulse indicating that the IP core receiver user data interface received an end-of-packet (EOP). You can use this signal to increment a count of EOPs the application observes on the receive interface.
nad_cntr_inc	0	Output	This signal is tied to 0 and it is not used.

Table 24. Burst Control Settings

Signal Name	Width (Bits)	I/O Direction	Description
burst_max_in	4	Input	Encodes the **BurstMax** parameter for the IP core. The actual value of the **BurstMax** parameter must be a multiple of 64 bytes. While traffic is present, this input signal should remain static. However, when no traffic is present, you can modify the value of the burst_max_in signal to modify the **BurstMax** value of the IP core. The IP core supports the following valid values for this signal: • 4'b0010: 128 bytes • 4'b0100: 256 bytes • 4'b1000: 512 bytes[7]
burst_short_in	4	Input	Encodes the **BurstShort** parameter for the IP core. The IP core supports the following valid value for this parameter: • 4'b0001: 32 bytes • 4'b0010: 64 bytes In general, the presence of the **BurstMin** parameter makes the BurstShort parameter obsolete.
burst_min_in	4	Input	Encodes the **BurstMin** parameter for the IP core. The IP core supports the following valid values for this signal: • 4'b0000: Disable optional enhanced scheduling. If you disable enhanced scheduling, performance is non-optimal. • 4'b0001: 32 bytes[8] • 4'b0010: 64 bytes • 4'b0100: 128 bytes The **BurstMin** parameter should have a value that is less than or equal to half of the value of the **BurstMax** parameter. Intel recommends that you modify the value of this input signal only when no traffic is present on the TX user data interface. You do not need to reset the IP core.

Table 25. ECC Status Signals

Signal Name	Width (Bits)	I/O Direction	Description
itx_eccstatus	2	Output	Indicates the TX ECC status.

continued...

[6] Synchronous with rx_usr_clk.

[7] This value is not supported in internal_words=4.

[8] This value is not supported in internal_words=8 and internal_words=16.

Signal Name	Width (Bits)	I/O Direction	Description
			• Bit 1: Correctable error status • Bit 0: Uncorrectable error status
irx_eccstatus	2	Output	Indicates the RX ECC status. • Bit 1: Correctable error status • Bit 0: Uncorrectable error status

5.7. External PLL Interface Signals

The external PLL interface signals are available in the Intel Stratix 10 H- and L- Tile device variations. In the Intel Stratix 10 E-Tile device variations, tie the input signals low and float the output signals.

Table 26. ATX PLL Interface Signals

Signal Name	Width (Bits)	I/O Direction	Description
tx_pll_locked	1	Input	PLL-locked indication from external TX PLL.
tx_pll_cal_busy	1	Input	PLL-busy indication from external TX PLL.
tx_serial_clk	NUM_LANES	Input	High-speed clock for Intel Stratix 10 transceiver channel, provided from external TX PLL.
tx_pll_powerdown	1	Output	Output signal from the IP core internal reset controller. The IP core asserts this signal to tell the external PLLs to power down.

Related Information

- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide

6. Register Map

The Interlaken IP core control registers are 32 bits wide and are accessible to you using the management interface, an Avalon-MM interface which conforms to the *Avalon Interface Specifications*. This table lists the registers available in the IP core. All unlisted locations are reserved.

Table 27. IP Core Register Map

Offset	Name	R/W	Description
8'h0	PCS_BASE	RO	[31:8] – Constant "HSi" ASCII [7:0] – version number Despite its name, this register does not encode the hard PCS base address.
8'h1	LANE_COUNT	RO	Number of lanes
8'h3	ELAPSED_SEC	RO	[23:0] - Elapsed seconds since power up. The IP core calculates this value from the management interface clock (`mm_clk`) for diagnostic purposes. During continuous operation, this value rolls over every 194 days.
8'h4	TX_EMPTY	RO	[NUM_LANES–1:0] – Transmit FIFO status (empty)
8'h5	TX_FULL	RO	[NUM_LANES–1:0] – Transmit FIFO status (full)
8'h6	TX_PEMPTY	RO	[NUM_LANES–1:0] – Transmit FIFO status (partially empty)
8'h7	TX_PFULL	RO	[NUM_LANES–1:0] – Transmit FIFO status (partially full)
8'h8	RX_EMPTY	RO	[NUM_LANES–1:0] – Receive FIFO status (empty)
8'h9	RX_FULL	RO	[NUM_LANES–1:0] – Receive FIFO status (full)
8'hA	RX_PEMPTY	RO	[NUM_LANES–1:0] – Receive FIFO status (partially empty)
8'hB	RX_PFULL	RO	[NUM_LANES–1:0] – Receive FIFO status (partially full)
8'hC	MAC_CLK_KHZ	RO	MAC clock frequency (kHz). This register is only available in Intel Stratix 10 E-Tile PAM4 device variations.
8'hD	RX_KHZ	RO	RX recovered clock frequency (kHz)
8'hE	TX_KHZ	RO	TX serial clock frequency (kHz)
8'h10	PLL_LOCKED	RO	In Intel Stratix 10 H- and L-Tile device variations: • Bit[0] – Transmit PLL lock indication. One lock indicator per transceiver block. Bits that correspond to unused transceiver block PLLs are forced to 1. In Intel Stratix 10 E-Tile device variations: • Bit[16] – MAC clock PLL lock indication. • Bit[0] – Transmit PLL lock indication. One lock bit for all transceivers.
8'h11	FREQ_LOCKED	RO	[NUM_LANES–1:0] – Clock data recovery is frequency locked on the inbound data stream

continued...

Offset	Name	R/W	Description
8'h12	LOOPBACK	RW	In Intel Stratix 10 H- and L-Tile device variations: [NUM_LANES–1:0] – For each lane, write a 1 to activate internal TX to RX serial loopback mode, or write a 0 to disable the loopback for normal operation. In Intel Stratix 10 E-Tile device variations: Interlaken Intel FPGA IP core does not support this function. To enable internal serial loopback, perform Avalon®-MM read/write to E-Tile registers.
8'h13	RESET	RW	Bit 9 : 1 = Force lock to data mode (Only in Intel Stratix 10 H- and L-Tile device variations) Bit 8 : 1 =Force lock to reference mode (Only in Intel Stratix 10 H- and L- Tile device variations) Bit 7 : 1 = Synchronously clear the TX-side error counters and sticky flags Bit 6 : 1 = Synchronously clear the RX-side error counters and sticky flags The normal operating state for this register is all zeros, to allow automatic reset control. These bits are intended primarily for hardware debugging use. Bits 6 and 7 are convenient for monitoring long stretches of error-free operation.
8'h20	ALIGN	RO	Bit 12 : RSFEC AM sync align (Only available in PAM4 Intel Stratix 10 E-Tile devices, not valid in NRZ mode) Bit 0 : TX lanes are aligned Bit 1 : RX lanes are aligned.
8'h21	WORD_LOCK	RO	[NUM_LANES–1:0] – Word (block) boundaries have been identified in the RX stream.
8'h22	SYNC_LOCK	RO	[NUM_LANES–1:0] – Metaframe synchronization has been achieved.
8'h23	CRC0	RO	4 bit counters indicating CRC errors in lanes [7:0]. These saturates at F, and you clear them by setting bit 6 in the RESET register.
8'h24	CRC1	RO	4 bit counters indicating CRC errors in lanes [15:8]. These saturates at F, and you clear them by setting bit 6 in the RESET register.
8'h25	CRC2	RO	4 bit counters indicating CRC errors in lanes [23:16]. These saturates at F, and you clear them by setting bit 6 in the RESET register.
8'h26	CRC3	RO	4 bit counters indicating CRC errors in lanes [31:24]. These saturates at F, and you clear them by setting bit 6 in the RESET register.
8'h28	RX_LOA	RO	Bit [0] – Sticky flag indicating loss of RX side lane-to-lane alignment since this bit was last cleared through the RESET register. Typically, the IP core asserts this bit in case of a catastrophic problem such as one or more lanes going down.
8'h29	TX_LOA	RO	Bit [0] – Sticky flag indicating loss of TX side lane to lane alignment since this bit was last cleared through the RESET register. Typically, the IP core asserts this bit in case of a TX FIFO underflow / overflow caused by a significant deviation from the expected data flow rate through the TX PCS.
8'h38	CRC32_ERR_INJECT	RW	[NUM_LANES–1:0] - When a bit has the value of 1, the IP core injects CRC32 errors on the corresponding TX lane. When it has the value of 0, the IP core does not inject errors on the TX lane. You must maintain each bit at the value of 1 for the duration of a Meta Frame, at least, to ensure that the IP core transmits at least one CRC32 error.

continued...

Offset	Name	R/W	Description
8'h40	TX_READY_XCVR	RO	[NUM_LANES-1:0] - Transceiver transmit channels are ready. This register is only available in Intel Stratix 10 E-Tile variations.
8'h41	RX_READY_XCVR	RO	[NUM_LANES-1:0] - Transceiver receive channels are ready. This register is only available in Intel Stratix 10 E-Tile variations.
12'h80	ILKN_FEC_XCODER_TX_ILLEGAL_STATE	RO	This register is only available in PAM4 Intel Stratix 10 E-Tile variations. Transcoder detects illegal framing bits [66:64] of the Interlaken frame layer words. Write 0 to clear.
12'h81	ILKN_FEC_XCODER_RX_UNCOR_FECCW	RO	This register is only available in PAM4 Intel Stratix 10 E-Tile variations. FEC indicates uncorrectable FEC code word error. Write 0 to clear.

Related Information

Avalon Interface Specifications

7. Test Features

Depending on the features you turn on in the IP parameter editor, your Interlaken IP core supports the following test features:

Note: These test features are available to designs that target Intel Stratix 10 L- and H-Tile devices only.

7.1. Internal Serial Loopback Mode

The Interlaken IP core supports an internal TX to RX serial loopback mode.

To turn on internal serial loopback:

- Reset the IP core by asserting and then deasserting the active low `reset_n` signal.
- After reset completes, set the value of bits [NUM_LANES-1:0] of the LOOPBACK register at offset 0x12 to all ones.

  Note: Refer to "IP Core Reset" for information about the required wait period for register access.
- Monitor the RX lanes aligned bit (bit 0) of the ALIGN register at offset 0x20 or the `rx_lanes_aligned` output signal. After the RX lanes are aligned, the IP core is in internal serial loopback mode.

Resetting the IP core turns off internal serial loopback. To turn off internal serial loopback:

- Reset the IP core by asserting and then deasserting the active low `reset_n` signal. Resetting the IP core sets the value of bits [NUM_LANES-1:0] of the LOOPBACK register at offset 0x12 to all zeros.
- Monitor the RX lanes aligned bit (bit 0) of the ALIGN register at offset 0x20 or the `rx_lanes_aligned` output signal. After the RX lanes are aligned, the IP core is in normal operational mode.

Related Information

IP Core Reset on page 34

7.2. External Loopback Mode

The Interlaken IP core operates correctly in an external loopback configuration.

To put the IP core in external loopback mode, connect the TX lanes to the RX lanes of the IP core. This mode does not require any special programming of the IP core.

8. Document Revision History for Interlaken (2nd Generation) Intel FPGA IP User Guide

Document Version	Intel Quartus Prime Version	Changes
2018.07.16	18.0.1	• Added support for the Intel Stratix 10 devices with E-Tile transceivers. • Added 53.125 Gbps data rate support for Intel Stratix 10 E-tile devices in PAM4 mode. • Added the new supported combinations of number of lanes and data rates in *Table: IP Core Supported Combinations of Number of Lanes and Data Rate*. • Updated the *Table: Performance and Resource Utilization* for Intel Stratix 10 E-Tile devices in NRZ and PAM4 mode. • Added new parameter **XCVR Mode** in *Table: Interlaken IP Core Parameter Settings: IP Tab*. • Added new **Transceiver reference clock frequency** values for 25.3, 25.8, and 26.5625 Gbps data rate in the *Table: Interlaken IP Core Parameter Settings: IP Tab* and *Table: Clock and Reset Interface Signals*. • Added clock signal `mac_clkin` in *Table: Interlaken IP Core Clocks* for Intel Stratix 10 E-Tile PAM4 devices. • Updated *Table: IP Core Register Map* for Intel Stratix 10 E-Tile devices.
2018.05.07	18.0	• Renamed the document as *Interlaken (2nd Generation) Intel FPGA IP User Guide* • Added new 25.8 Gbps data rate support for number of lanes 6 and 12. • Added Cadence Xcelium Parallel simulator support. • Added new section *Integrating Your IP Core in Your Design* explaining how to make appropriate pin assignments and add external PLL. • Added the **Transceiver reference clock frequency** for 25.8 Gbps data rate in the *Table: Interlaken IP Core Parameter Settings: IP Tab* and *Table: Clock and Reset Interface Signals*. • Modified default setting value for the **Tx Scrambler Seed** parameter in *Table: Interlaken IP Core Parameter Settings: IP Tab* • Clarified the direction of the IP core clocks in *Table: Interlaken IP Core Clocks*.

Date	Version	Changes
November 2017	2017.11.06	• Updated for Intel Quartus Prime Pro Edition 17.1 release. • Updated support for the Intel Stratix 10 devices with L-Tile and H-Tile transceivers in Table: *IP Core Supported Combinations of Number of Lanes and Data Rate*. • Added the resource utilization numbers for 25.3 Gbps data rate in Table: *FPGA Resource Utilization*. • Added support for Cadence NCSim simulator.

continued...

Date	Version	Changes
		• Added new parameter **Transceiver Tile** in Table: *Interlaken IP Core Parameter Settings: IP Tab*. • Removed 412.5 MHz `pll_ref_clk` frequency support for 25.3 Gbps data rate in Table: *Clock and Reset Interface Signals*. • Added new signal `nad_cntr_inc` in Table: *Real-Time Receiver Status Signals*.
May 2017	2017.05.08	• Updated the resource utilization in *Table: FPGA Resource Utilization*. • Added the new supported combinations of number of lanes and data rate (6x25.3G and 12x25.3G) in *Table: IP Core Supported Combinations of Number of Lanes and Data Rate* • Corrected the steps for *Specifying the IP Core Parameters and Options*. • Added the transceiver reference clock frequency for 25.3 Gbps data rate in the *Table: Interlaken IP Core Parameter Settings* and *Table: Clock and Reset Interface Signals*.
December 2016	2016.12.19	• Dynamic reconfiguration support is now available for Intel Stratix 10 devices. • Added a new parameter `VCCR_GXB and VCCT_GXB supply voltage for the transceivers` in the table: *Interlaken IP Core Parameter Settings*.
August 2016	2016.08.08	Initial version for Quartus Prime Pro – Stratix 10 Edition Beta software.