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

Updated for Intel® Quartus® Prime Design Suite: **21.3**

IP Version: **20.0.1**

Online Version
Send Feedback

UG-20035

ID: 683396
Version: 2021.10.04

Contents

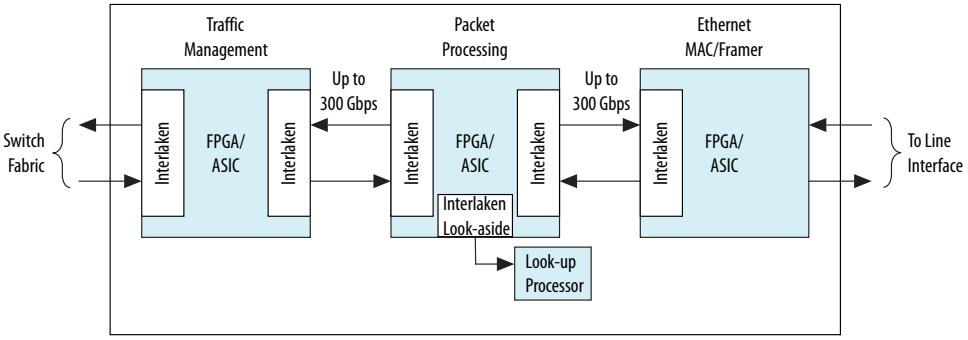
1. Introduction

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, v1.2*. This IP also implements Interlaken Look-aside protocol compliant to *Interlaken Look-Aside Protocol Definition, v1.1*. Interlaken 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 and Intel Agilex™ devices, providing raw bandwidth up 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 incorporates a physical coding sublayer (PCS), a physical media attachment (PMA), and a media access control (MAC) block.

Interlaken Look-aside is a scalable protocol that allows interoperability between a datapath device and a Look-aside co-processor with packet transfer rates up to 300 Gbps.

Figure 1. Typical Interlaken Application



Related Information

- Interlaken (2nd Generation) Intel Stratix 10 FPGA IP Design Example User Guide
  Describes a simulating testbench and a hardware example design that supports compilation and hardware testing.
- Interlaken Protocol Specification, v1.2
- Interlaken (2nd Generation) Intel FPGA IP User Guide Archives on page 99
- Interlaken (2nd Generation) Intel Agilex FPGA IP Design Example User Guide
- Interlaken Look-Aside Protocol Definition, v1.1

1.1. Features

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

- General features:
  - Compliant with the Interlaken Protocol Specification, Revision 1.2.
  - Supports 4, 6, 8, 10, 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, 25.28, 25.78, 25.78125, and 53.125 Gbps using Intel FPGA on-chip high-speed transceivers.
- User interface features:
  - 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 multi-segment user interface.
- Flow-control features:
  - Supports optional out-of-band flow control blocks.
  - Supports optional user-controlled in-band flow control with 1, 2, 4, 8, or 16 16-bit calendar pages.
  - Supports error correction code (ECC) for memory block implementation with the IP.
- Line-side features:
  - Supports per-lane data rate of 53.125 Gbps using pulse amplitude modulation (PAM4) mode in E-tile variations.
  - Supports per lane data rates of 6.25, 10.3125, 12.5, 25.28, 25.78, and 25.78125 Gbps using non-return-to-zero (NRZ) mode in E-tile variations.
- PHY features:
  - Supports PMA adaptation.
- Interlaken Look-aside mode:
  - Supports per lane data rates of 6.25, 10.3125, 12.5, 25.28, 25.78 and 25.78125 Gbps using NRZ and 53.125 Gbps using PAM4 mode in E-tile variations.
  - Supports packet mode for user data transfer.
  - Available only in Intel Stratix 10 and Intel Agilex E-tile device variations in current version of the Intel Quartus® Prime software release.

    Note: Contact Intel Support to request Interlaken Look-aside mode access in IP variations that target an Intel Stratix 10 H- and L-tile device.

Table 1. IP Supported Combinations of Number of Lanes and Data Rates

The following combinations are supported in the current version of the Intel Quartus Prime Pro Edition software.

Throughout this table, ILK refers to Interlaken mode and ILA refers to Interlaken Look-aside mode.

Device	Supported Mode	IP Supported Combinations	
		Number of Lanes	Lane Rate (Gbps)(1)
L-tile	ILK only	4	6.25
			12.5
			25.28
			25.78
			25.78125
		8	12.5
		12	10.3125
			12.5
H-tile	ILK only	4	6.25
			12.5
			25.28
			25.78
			25.78125
		6	25.28
			25.78
			25.78125
		8	12.5
			25.28
			25.78
			25.78125
		10	12.5
			25.28
			25.78
			25.78125
		12	10.3125
			12.5
			25.28
			25.78
			25.78125

continued...

(1) You can customize the data rates depending on the tiles. Refer to the KDB for information on how to customize the data rate.

Device	Supported Mode	IP Supported Combinations	
		Number of Lanes	Lane Rate (Gbps)(1)
E-tile (NRZ)	ILK and ILA	4	6.25
			12.5
			25.28
			25.78
			25.78125
		6	25.28
			25.78
			25.78125
		8	12.5
			25.28
			25.78
			25.78125
		10	12.5
			25.28
			25.78
			25.78125
		12	10.3125
			12.5
			25.28
			25.78
			25.78125
E-tile (PAM4)	ILK and ILA	12	26.5625(2)

Table 2. IP Theoretical Raw Aggregate Bandwidth

The following combinations are supported in the current version of the Intel Quartus Prime Pro Edition software:

Number of Lanes	Lane Rate (Gbps)						
	6.25	10.3125	12.5	25.28	25.78	25.78125	26.5625
4	25	-	50	101.12	103.12	103.125	-
6	-	-	-	151.68	154.68	154.6875	-
8	-	-	100	202.24	206.24	206.25	-
10	-	-	125	252.8	257.8	257.8125	-
12	-	123.75	150	303.36	309.36	309.375	318.75

(1) You can customize the data rates depending on the tiles. Refer to the KDB for information on how to customize the data rate.

(2) To obtain 6x53.125 Gbps speed in PAM4 mode, select 12x26.5625 Gbps.

Related Information

Interlaken Protocol Specification, v1.2

1.2. Device Family Support

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

- **Advance support** — The IP 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 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 is verified with preliminary timing models for this device family. The IP 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 is verified with final timing models for this device family. The IP 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	Final
Intel Agilex	Advance

1.3. Performance and Resource Utilization

This section covers the resources and expected performance numbers for selected variations of the Interlaken IP core using the Intel Quartus Prime Pro Edition software. The number of ALMs and logic registers are rounded up to the nearest 100. Your results may slightly vary depending on the device you select.

For a comprehensive list of supported configurations, refer to *Table 1. IP Supported Combinations of Number of Lanes and Data Rates*

Table 4. Intel Stratix 10 FPGA Resource Utilization in Interlaken Mode

The following numbers were obtained using the Intel Quartus Prime Pro Edition software version 20.3.

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	11500	22700	4400	32
		12.5	11900	24600	4000	32
		25.28	12000	24900	4400	32
		25.78	12000	24800	4200	32
		25.78125	12000	25000	4200	32
	8	12.5	23000	50400	7200	56
	10	12.5	29000	64300	8300	65

continued...

Device	Parameters		Resource Utilization			
	Number of Lanes	Data/Lane Rate (Gbps)	ALMs needed	Logic Registers		M20K Blocks
				Primary	Secondary	
Intel Stratix 10 H-tile	12	10.3125	24100	51300	7600	56
		12.5	24000	50800	8100	56
	4	6.25	11500	22900	4200	32
		12.5	12200	24900	4200	32
		25.28	12400	25500	4000	32
		25.78	12400	25500	4200	32
		25.78125	12400	25400	4100	32
	6	25.28	23800	50700	7300	56
		25.78	23800	50500	7300	56
		25.78125	23700	50400	7700	56
	8	12.5	23800	51000	7300	56
		25.28	24200	52000	6400	56
		25.78	24300	51200	6800	56
		25.78125	24100	52400	6300	56
	10	12.5	29000	64600	8000	65
		25.28	35700	84600	7500	104
		25.78	35700	81100	7600	104
		25.78125	35700	80300	8700	104
	12	10.3125	25200	52700	7400	56
		12.5	24200	51000	7900	56
		25.28	38600	88000	9800	104
		25.78	38700	87400	10400	104
		25.78125	38600	88000	9200	104
Intel Stratix 10 E-tile (NRZ)	4	6.25	17300	33500	5600	32
		12.5	17300	33500	5700	32
		25.28	17500	33600	5800	32
		25.78	17500	33400	5800	32
		25.78125	17500	33200	5900	32
	6	25.28	31400	63000	10300	56
		25.78	31400	63300	9900	56
		25.78125	31400	63500	10400	56
	8	12.5	34000	68200	10400	56
		25.28	34500	68400	40400	56
		25.78	34500	68600	9700	56

continued...

Device	Parameters		Resource Utilization			
	Number of Lanes	Data/Lane Rate (Gbps)	ALMs needed	Logic Registers		M20K Blocks
				Primary	Secondary	
		25.78125	34500	67900	10500	56
	10	12.5	42700	86700	12000	65
		25.28	48500	101400	12500	104
		25.78	48500	100600	13200	104
		25.78125	48500	100800	12800	104
	12	10.3125	51800	104900	14200	77
		12.5	49700	102600	14300	89
		25.28	54000	112700	15200	104
		25.78	54000	112000	15650	104
		25.78125	54000	112200	15200	104
Intel Stratix 10 E-tile (PAM4)	12	26.5625	67000	137600	17900	104

Table 5. Intel Agilex FPGA Resource Utilization in Interlaken Mode

The following numbers were obtained using the Intel Quartus Prime Pro Edition software version 20.3:

Device	Parameters		Resource Utilization			
	Number of Lanes	Data/Lane Rate (Gbps)	ALMs needed	Logic Registers		M20K Blocks
				Primary	Secondary	
Intel Agilex E-tile (NRZ)	4	6.25	17700	33500	7700	32
		12.5	17500	33400	7800	32
		25.28	17800	34000	8000	32
		25.78	17700	34000	7800	32
		25.78125	17900	34000	8000	32
	6	25.28	31800	63700	13700	56
		25.78	31700	63700	13600	56
		25.78125	31700	63500	13800	56
	8	12.5	34200	68200	13300	56
		25.28	34600	68900	14100	56
		25.78	34600	69000	13900	56
		25.78125	34600	69100	13900	56
	10	12.5	43200	86600	16000	65
		25.28	48600	120100	17600	104
		25.78	48600	102000	18000	104
		25.78125	48500	101600	18000	104
	12	10.3125	52000	105100	18000	77

continued...

Device	Parameters		Resource Utilization			
	Number of Lanes	Data/Lane Rate (Gbps)	ALMs needed	Logic Registers		M20K Blocks
				Primary	Secondary	
		12.5	50000	102700	18300	89
		25.28	54200	113800	20600	104
		25.78	54100	113300	20500	104
		25.78125	54300	113800	20700	104
Intel Agilex E-tile (PAM4)	12	26.5625	67700	137400	26000	104

Table 6. Intel Stratix 10 E-tile Resource Utilization Numbers in Interlaken Look-aside Mode

The following numbers were obtained using the Intel Quartus Prime Pro Edition software version 20.3:

Device	Parameters		Resource Utilization			
	Number of Lanes	Data/Lane Rate (Gbps)	ALMs	Logic Registers		M20 Blocks
				Primary	Secondary	
Intel Stratix 10 E-tile (NRZ)	4	6.25	11900	19700	3300	4
		12.5	11800	19500	3400	4
		25.28	11900	19600	3300	4
		25.78	11800	19600	3300	4
		25.78125	11900	19600	3300	4
	6	25.28	17600	29300	4700	4
		25.78	17400	29000	4800	4
		25.78125	17500	29000	4800	4
	8	12.5	23200	39300	6300	4
		25.28	23400	39100	6100	4
		25.78	23500	39200	6100	4
		25.78125	23400	39000	6300	4
	10	12.5	29600	51000	7900	4
		25.28	29600	50000	7900	4
		25.78	29700	50000	7900	4
		25.78125	29900	50500	7600	4
	12	10.3125	35700	62600	9000	4
		12.5	35800	62200	9200	4
		25.28	35800	61500	8800	4
		25.78	36000	61000	8900	4
		25.78125	35900	61000	9000	4
Intel Stratix 10 E-tile (PAM4)	12	26.5625	49400	86500	11800	4

Table 7. Intel Agilex E-tile Resource Utilization Numbers in Interlaken Look-aside Mode

The following numbers were obtained using the Intel Quartus Prime Pro Edition software version 20.3:

Device	Parameters		Resource Utilization			
	Number of Lanes	Data/Lane Rate (Gbps)	ALMs	Logic Registers		M20 Blocks
				Primary	Secondary	
Intel Agilex E-tile (NRZ)	4	6.25	11800	19000	4600	4
		12.5	11800	19400	4300	4
		25.28	11900	19500	4600	4
		25.78	11700	19300	4500	4
		25.78125	11800	19300	4600	4
	6	25.28	17200	28500	6900	4
		25.78	17100	28500	6700	4
		25.78125	17200	28600	6900	4
	8	12.5	23400	39200	8400	4
		25.28	23500	39000	9000	4
		25.78	23200	38800	9000	4
		25.78125	23400	39000	8800	4
	10	12.5	29500	50000	10700	4
		25.28	29900	50100	11700	4
		25.78	29700	49900	11200	4
		25.78125	29800	50000	11400	4
	12	10.3125	35700	61200	12600	4
		12.5	35900	61200	12600	4
		25.28	35800	60900	13000	4
		25.78	35900	60600	13100	4
		25.78125	35800	60800	13500	4
Intel Agilex E-tile (PAM4)	12	26.5625	49200	86000	17200	4

Related Information

Features on page 5

1.4. Flexible Lanes Support

In the current version of the Intel Quartus Prime Pro Edition software, the IP parameter editor prompts you to the recommended user clock frequency for the combination of number of lanes and data rates. The software derives the user clock frequency based on the configuration you select. The user clock frequency maps to the `tx_usr_clk` and `rx_usr_clk` signals.

Table 8. Recommended User Clock Frequency

Number of Lanes	Data/Lane Rate (Gbps)	Number of Segments	User Clock Frequency (MHz)	
			H-tile	E-tile
4	6.25	1	200	100
	12.5		200	
	25.28		400	
	25.78		400	
	25.78125		400	
6	25.28	1, 2	300	
	25.78		300	
	25.78125		300	
8	12.5	1, 2	200	
	25.28		400	
	25.78		400	
	25.78125		400	
10	12.5	1, 2	250	
	25.28	1, 2, 4	250	
	25.78	1, 2, 4	300	
	25.78125	1, 2, 4	300	
12	10.3125	1	300	250
		2	250	
	12.5	1, 2	300	
	25.28	1	350	
		2, 4	300	
	25.78	1, 2, 4	350	
	25.78125	1, 2, 4	350	
	25.5625 (Only in E-tile PAM4 mode IP variations)	1, 2, 4	N/A	350

1.5. Round-trip Latency

The following table includes the round-trip latency numbers for specific variants. The latency numbers were measured for the longest logical datapath for two highest lane rate and number of lanes variants, with FIFO level at 50 for the first packet.

Table 9. Round-trip Latency Numbers

Device	Number of Lanes	Lane Rate (Gbps)	Interlaken		Interlaken Look-aside	
			Number of Segments	Latency (Number of tx_usr_clk cycles)	Number of Segments	Latency (Number of tx_usr_clk cycles)
E-tile (NRZ)	12	25.78	4	260	1	104
E-tile (PAM4)	12	26.5625	4	368	1	228

1.6. Release Information

Intel FPGA IP versions match the Intel Quartus Prime Design Suite software versions until v19.1. Starting in Intel Quartus Prime Design Suite software version 19.2, Intel FPGA IP has a new versioning scheme.

The Intel FPGA IP version (X.Y.Z) number can change with each Intel Quartus Prime software version. A change in:

- X indicates a major revision of the IP. If you update the Intel Quartus Prime software, you must regenerate the IP.
- Y indicates the IP includes new features. Regenerate your IP to include these new features.
- Z indicates the IP includes minor changes. Regenerate your IP to include these changes.

Table 10. Interlaken (2nd Generation) Intel FPGA IP Core Release Information

Item	Value		
IP Version	20.0.1		
Intel Quartus Prime Version	21.3		
Release Date	2021.10.04		
Ordering Code	**Aggregate Bandwidth**	**Ordering Code**	**Product ID**
	20G to <100G	IP-ILKN/50G	010E
	100G to <200G	IP-ILKN/100G	
	200G to <400G	IP-ILKN/200G	
	Note: The ordering code for Interlaken Look-aside IP solution is same as the Interlaken (2nd Generation) Intel FPGA IP.		

Related Information

Interlaken (2nd Generation) Intel FPGA IP Release Notes
Describes changes to the IP in a particular release.

2. Getting Started

The following sections explain how to install, parameterize, simulate, and initialize the Interlaken (2nd Generation) Intel FPGA IP.

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 (Intel Quartus Prime Pro Edition)
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- Project File 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

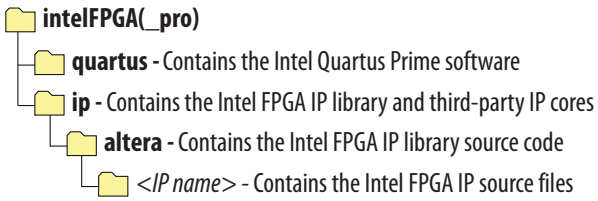


Table 11. IP Core Installation Locations

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.

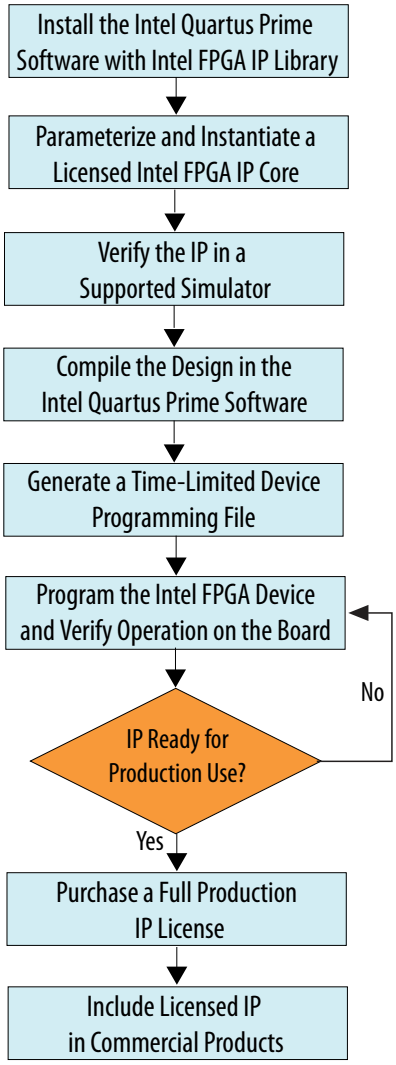
Intel FPGA IP Evaluation Mode supports the following operation modes:

- **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.

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 FPGA Licensing Support Center
- Introduction to Intel FPGA Software Installation and Licensing

2.2. Generated File Structure

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

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

Figure 4. IP Generated Files

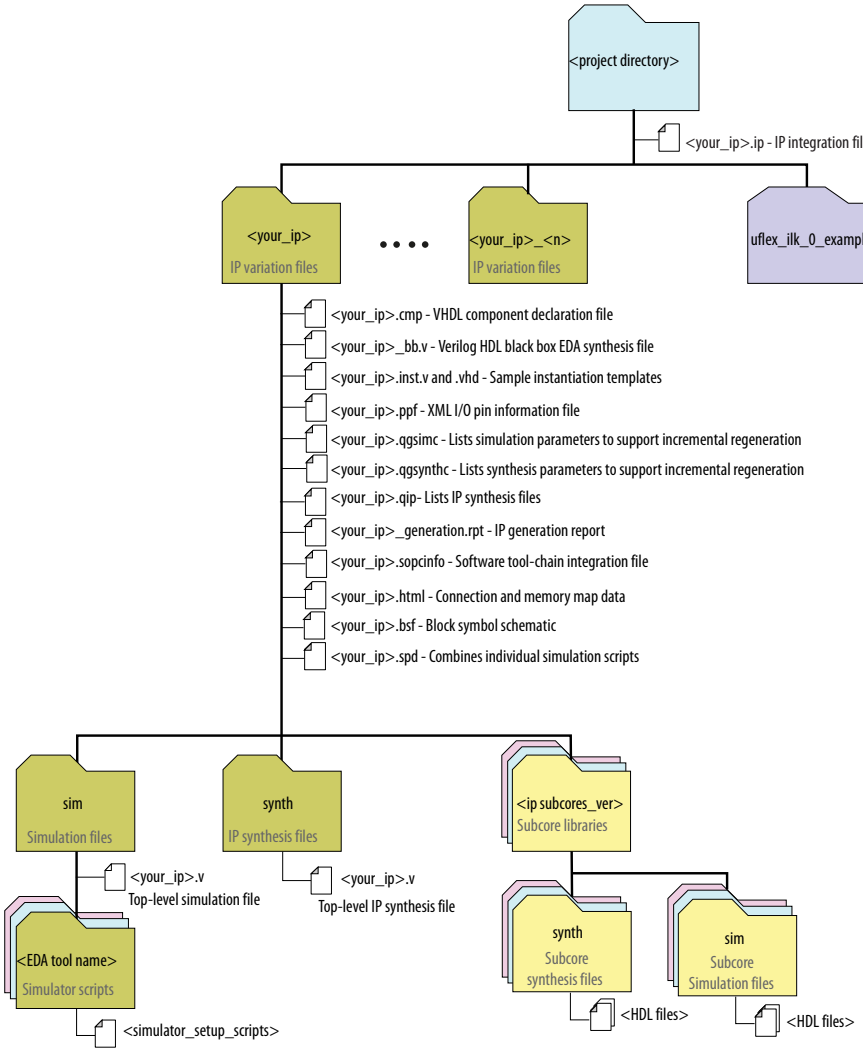


Table 12. IP Generated Files

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 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 does not support VHDL. However, the Intel Quartus Prime Pro Edition software generates the `_inst.vhd` file.
<*your_ip*>.v	HDL files that instantiates each submodule or child IP for synthesis or simulation.
mentor/	Contains a ModelSim* SE or Questa* 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.
xcelium/	Contains a shell script `xcelium_setup.sh` to set up and run a Xcelium* simulation.
submodules/	Contains HDL files for the IP core submodules.
<*child IP cores*>/	For each generated child IP directory, Platform Designer generates `synth/` and `sim/` sub-directories.

Related Information

- Interlaken (2nd Generation) Intel Stratix 10 FPGA IP Design Example User Guide

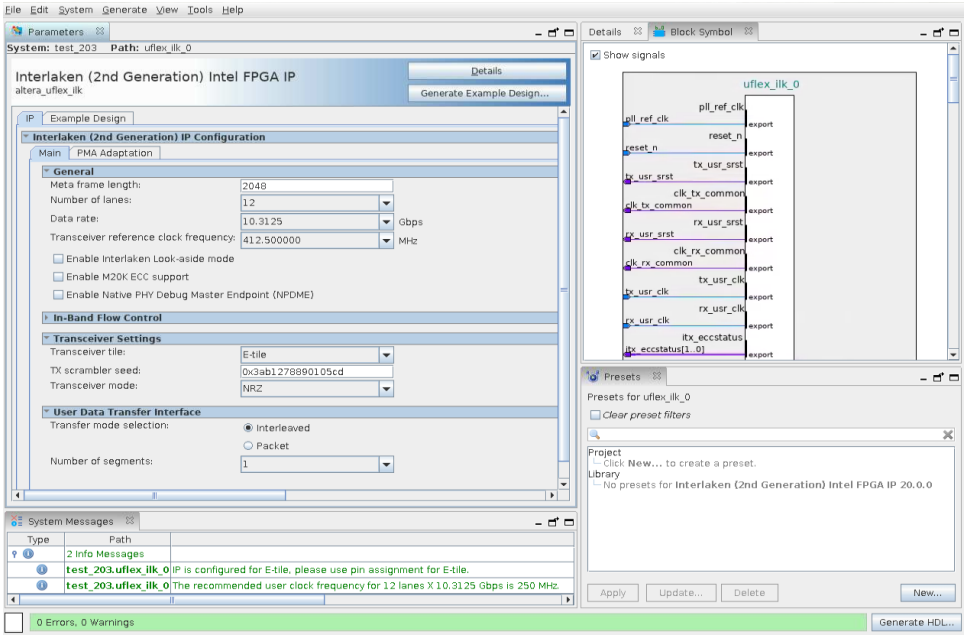
- Interlaken (2nd Generation) Intel Agilex FPGA IP Design Example User Guide

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 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 does not support VHDL simulation models. You must specify the Verilog HDL for both synthesis and simulation models.

Figure 5. IP Parameter Editor



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.
2. Select the device family either **Stratix 10 (GX/SX/MX/TX)** or **Agilex (FB/FA)** as your target device.
3. In the IP Catalog (**Tools ➤ IP Catalog**), locate and double-click **Interlaken (2nd Generation) Intel FPGA IP**. The **New IP Variant** window appears.
4. 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.
5. Click **Create**. The parameter editor appears.
6. 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.

7. Click **Generate HDL**. The **Generation** dialog box appears.
8. Specify output file generation options, and then click **Generate**. The IP variation files generate according to your specifications.
9. 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.
10. 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.
11. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports.

Related Information

- Interlaken (2nd Generation) Intel Stratix 10 FPGA IP Design Example User Guide
  Describes a simulating testbench and a hardware example design that supports compilation and hardware testing.
- Main Parameters on page 26
- Interlaken (2nd Generation) Intel Agilex FPGA IP Design Example User Guide

2.4. Simulating the IP Core

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

The Interlaken IP supports the following simulators:

- Siemens* EDA ModelSim SE or Questa
- Synopsys* VCS
- Synopsys VCS MX
- Cadence* Xcelium

The Interlaken IP generates a Verilog HDL and VHDL simulation model and testbench. The IP core parameter editor offers you the option of generating a Verilog HDL or VHDL simulation model for the IP core, but the IP core design example 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 *Intel Quartus Prime Pro Edition User Guide: Third-party Simulation*.

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 Interlaken IP 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. Pin assignments provide direction to the Fitter and specify the signals that should be assigned to device pins. While compiling only the IP core, 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.

Related Information

GX and GXT Channel Placement Guidelines

2.6.2. Adding the External PLL

The Interlaken (2nd Generation) IP core variations that target an 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 the **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.

For more information about ATX PLL connection and how to instantiate the ATX PLL, refer to the *Instantiating the ATX PLL IP Core* in the *Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide*.

To use fPLL as an external transceiver PLL, select **L-Tile/H-Tile fPLL Intel Stratix 10 FPGA IP** from the IP Catalog and set the parameters using the instructions in the *Instantiating the fPLL IP Core* in the *Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide.*

The Interlaken (2nd Generation) IP core variations that target an E-tile device contains transceiver PLLs and do not require an external PLL for the transceivers. These transceiver PLLs require a reference clock (`pll_ref_clk`). Refer to the *E-Tile Transceiver PHY User Guide* and *Interlaken (2nd Generation) Design Example User Guide* for the reference clock connections.

The E-tile PAM4 mode variations require an additional `mac_clkin` input clock generated by a PLL. This PLL must use the same reference clock source that drives the `pll_ref_clk`. Refer to *Figure: Interlaken (2nd Generation) Hardware Design Example High Level Block Diagram for E-tile PAM4 Mode Variations* in *Interlaken (2nd Generation) Intel FPGA IP Design Example User Guide* for more information about `mac_clkin` connections.

Related Information

- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Interlaken (2nd Generation) Intel Stratix 10 FPGA IP Design Example User Guide
- E-Tile Transceiver PHY User Guide
- Interlaken (2nd Generation) Intel Agilex FPGA IP Design Example User Guide

2.6.3. PMA Adaptation Flow

The Interlaken (2nd Generation) Intel FPGA IP does not start PMA adaptation in L- and H-tile device variations. If you want to enable PMA adaptation, you must start through transceiver PHY reconfiguration interface. Refer to *Adaptation Control- Start* section of the *Intel Stratix 10 L- and H-tile Transceiver PHY User Guide* for information on how to start PMA adaptation through transceiver PHY reconfiguration interface.

For your E-tile Interlaken IP core variations, perform the following steps to start PMA adaptation:

1. Set the operation mode:
   - Write 0x200 = 0x1F.
   - Write 0x201 = 0x00.
   - Write 0x202 = 0x00
   - Write 0x203 = 0x93. This picks the opcode for `SET_OPERATION_MODE`.

   Note: Values set for 0x200 and 0x201 are just examples. Refer to *Figure: Loading PMA Configuration Register SET_OPERATION_MODE* in *E-tile Transceiver PHY User Guide* for 32-bits decoding and set it according to your design intent.
2. Load recipe[3]:

[3] If load recipe is required.

- Write 0x40143 = 0x80. Loads the PMA configuration to all channels (0x40143 and 0x40144 can only be accessed from channel 0).
- Read 0x40144[0] until it reports 0x1. This ensures the process is not timed out.

3. Start adaptation:
   - Write 0x200 = 0xF2.
   - Write 0x201 = 0x03.
   - Write 0x202 = 0x01.
   - Write 0x203 = 0x96.

   Note: Values set for 0x200, 0x201, and 0x202 are just examples. Refer to *Figure: Loading PMA Configuration Register START_ADAPTATION* in *E-tile Transceiver PHY User Guide* for 32-bits decoding and set it according to your design intent.

   Note: PRBS will not be disabled automatically after calibration if you have already enabled it in `SET_OPERATION_MODE` and `START_ADAPTATION`. You need to turn it off manually.

4. Check ical status:

   `check_stat_cal` command can be run after `start_cal` command to return the calibration status:
   - Write 0x200 = 0x01.
   - Write 0x201 = 0x00.
   - Write 0x202 = 0x00.
   - Write 0x203 = 0x97.

   Refer to *Figure: Loading PMA Configuration Register CHECK_CAL_STAT* in *E-tile Transceiver PHY User Guide*.

5. Poll on 0x204 and process the following status:
   - If 0x204 = 0x91, PMA adaptation in progress
   - If 0x204 = 0x80, PMA adaptation done
   - If 0x204 = 0xe2, Continuous adaptation is running
   - If 0x204 = 0xa2, One time adaptation in progress

Related Information

- Adaptation Control- Start
- E-tile Transceiver PHY User Guide

3. Parameter Settings

3.1. Main Parameters

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

Table 13. General

Parameter	Supported Values	Default Setting	Description
Meta fame length	128-8192	2048	Specifies the meta frame length in 64-bit (8-byte) words. Must be a power of two. Smaller values shorten the time to achieve lock. Larger values reduce the overhead while transferring data, after the clock data recover (CDR) circuit achieves lock.
Number of lanes	4, 6, 8, 10, 12	12	This parameter specifies the number of lanes available for Interlaken communication. The Interlaken IP supports various combinations of number of lanes and lane rates. Ensure that your parameter settings specify a supported combination. Refer to *Table: IP Supported Combinations of Number of Lanes and Data Rate* in this document.
Data rate	6.25, 10.3125, 12.5, 25.28, 25.78, 25.78125, and 26.5625(4) Gbps	10.3125 Gbps	This parameter specifies the data rate on each lane. The Interlaken IP supports various combinations of number of lanes and lane rates. Ensure that your parameter settings specify a supported combination. Refer to *Table: IP Supported Combinations of Number of Lanes and Data Rate* in this document.
Transceiver reference clock frequency	Multiple	412.5 MHz	This parameter specifies the expected frequency of the `pll_ref_clk` input clock. If the actual frequency of the `pll_ref_clk` input clock does not match the value you specify for this parameter, the design fails in both simulation and hardware.

Per-Lane Data Rate (Gbps)	Valid Frequencies (MHz)
6.25	156.25, 195.3125, 250, 312.5, 390.625, 480.76923(6), 500(5), 625(5)

continued...

(4) This data rate is only available when you select PAM4 option for **Transceiver Mode** parameter in E-tile variations.

(5) Only available in H-tile and L-tile device variations.

Parameter	Supported Values	Default Setting	Description	
			Per-Lane Data Rate (Gbps)	**Valid Frequencies (MHz)**
			10.3125	156.25, 206.25, 257.8125, 322.265625, 412.5, 491.071428(6), 515.625(5), 644.53125(5)
			12.5	156.25, 195.3125, 250, 312.5, 390.625, 500, 625(5)
			25.28	126.4(5), 158.0, 197.5, 252.8, 320.0(5), 395.0, 486.153846(6), 505.6(5)
			25.78	159.135802, 201.40625, 250.291262(5), 322.25, 402.8125, 495.769231(6), 500.582524(5)
			25.78125	159.143519(5), 159.143518(6), 201.416016(5), 201.416015(6), 250.303398(5), 322.265625, 402.832031, 500.606796(5), 495.793269(6)
			26.5625(7)	156.25, 210.813492, 312.5, 390.625, 491.898148

continued...

(6) Only available in NRZ E-tile device variations.

(7) Only available in PAM4 E-tile device variations.

Parameter	Supported Values	Default Setting	Description
Enable Interlaken Look-aside mode	On/Off	Off	Enable this option to configure the IP core in Interlaken Look-aside mode. This option is only available in E-tile device variations. *Note:* This mode is only supported in non-striper mode.
Enable M20K ECC support	On/Off	Off	This parameter specifies whether your Interlaken IP variation supports the ECC feature in the M20K memory blocks that are configured as part of the IP. 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 blocks configured in your IP. This feature enhances data reliability but increases latency and resource utilization. The IP does not include M20K ECC support when you turn on **Enable Interlaken Look-aside** option.
Enable Native PHY Debug Master Endpoint (NPDME)	On/Off	Off	This parameter specifies whether your Interlaken IP variation supports the NPDME feature. This parameter exposes debugging features of the Intel Stratix 10 Native PHY IP that specifies the transceiver settings in the Interlaken IP.

Table 14. In-Band Flow Control

The parameters in table below are not available when you turn on **Enable Interlaken Look-aside mode** parameter.

Parameter	Supported Values	Default Setting	Description
Include in-band flow control functionality	On/Off	Off	This parameter specifies whether your Interlaken IP includes an in-band flow control block.
Number of calender pages	1, 2, 4, 8, 16	1	This parameter specifies the number of 16-bit pages of in-band flow control data that your Interlaken IP 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 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.

Table 15. Transceiver Settings

Transceiver Settings Parameter	Supported Values	Default Setting	Description
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 determines the transceiver tile type.

continued...

Transceiver Settings Parameter	Supported Values	Default Setting	Description
			This parameter is not available in Intel Agilex device variations. The software selects E-tile by default for Intel Agilex device variations.
TX scrambler seed	—	0x3ab1278890105cd	This parameter specifies the initial scrambler state. If you configure a single Interlaken IP on your device, you can use the default value of this parameter. If you configure multiple Interlaken IPs, you must use a different initial scrambler state for each IP to reduce crosstalk. Select random values for each Interlaken IP. Each scrambler seed should have an approximately even mix of ones and zeros. The scrambler seeds should differ from the other scramblers in multiple spread out bit positions.
Transceiver mode	NRZ, PAM4	NRZ	Specifies the transceiver mode. This parameter is available only in IP variations that target an Intel Stratix 10 E-tile device.
Preserve unused transceiver channels for PAM4	On/Off	Off	This option can be used to preserve the unused PAM4 slave channel.(8) This parameter is only available in E-tile PAM4 mode IP variations.
Reference clock frequency for preserved channels	125 MHz to 500 MHz	156 MHz	When you enable **Preserve unused transceiver channels for PAM4**, an additional reference clock port is added to preserve the unused PAM4 slave channel. Use this parameter to set the frequency of reference clock.
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 25.28 and 25.78 Gbps data rate. This parameter is not available in IP variations that target an Intel Stratix 10 and Intel Agilex E-tile device.

Table 16. User Data Transfer Interface

Parameter	Supported Values	Default Setting	Description
Number of Segments	1, 2, 4	1	This parameter enables 1, 2, or 4 segments depending on the total number of words. This parameter is grayed out when you turn on **Enable Interlaken Look-aside mode** parameter.

Related Information

- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Features on page 5
  For more information on IP core supported combinations of lanes and data rate.
- E-Tile Transceiver PHY User Guide

(8) For more details, refer to Unused Transceiver Channels in High-Speed PAM4 Mode.

3.2. PMA Adaptation Parameters

After device configuration, you may need to trigger PMA adaptation on the RX interface to achieve optimal performance. The PMA interface performs adaptation for initial adaptation and continuous adaptation which runs continuously in background.

Figure 6. PMA Adaptation Options

You specify settings on the **PMA Adaptation** tab for both initial and continuous adaptation. PMA Adaptation is only available for E-tile variants.

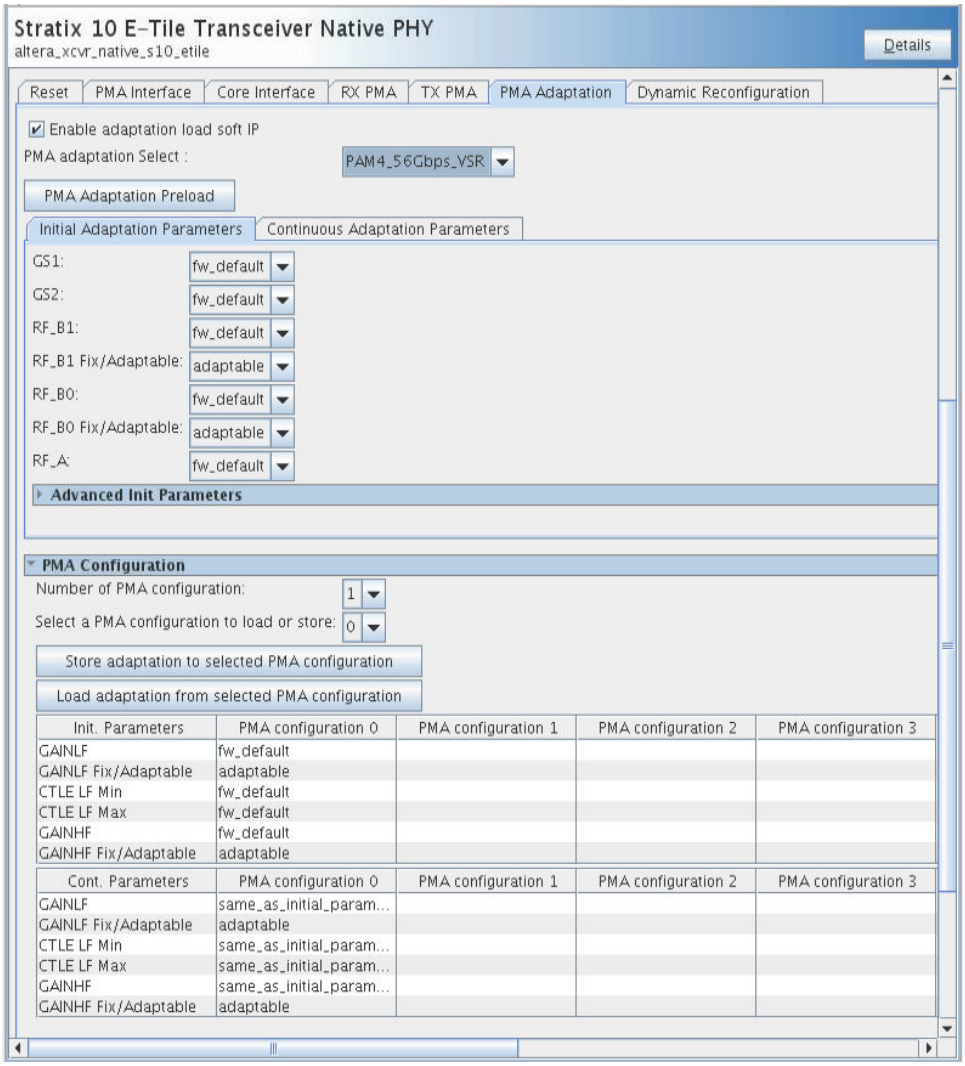


Table 17. PMA Adaptation

Parameter	Value	Description
Enable adaptation load soft IP	**On/Off**	Enables PMA adaptation parameter customization.
PMA adaptation select	**PAM4_56Gbps_LR** **PAM4_56Gbps_VSR**	Selects pre-defined PMA configuration parameters.

continued...

Parameter	Value	Description
	NRZ_28Gbps_LR **NRZ_28Gbps_VSR** **NRZ_10Gbps_25/15/10dB**	

Table 18. Initial Adaptation Parameters

Parameter	Value	Description
GS1	**Fw_default, 0, 1, 2, 3**	CTLE Low Frequency Gain Shaping 1.
GS2	**Fw_default, 0, 1, 2, 3**	CTLE Low Frequency Gain Shaping 2.
RF_B1	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8**	RF_B1 Setting.
RF_B1 Fix/Adaptable	**Fixed, Adaptable**	RF_B1 Setting.
RF_B0	**Fw_default, 0, 1, 2, 3, 4, 5**	RF_B0 Setting.
RF_B0 Fix/Adaptable	**Fixed, Adaptable**	RF_B0 Setting.
RF_A	**Fw_default, 100, 110, 120, 130, 140, 150, 160**	RF_A Setting.
GAINLF	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	CTLE Low Frequency Gain.
GAINLF Fixed/ Adaptable	**Fixed, Adaptable**	CTLE Low Frequency Gain.
CTLE LF Min	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Limits CTLE LF minimum.
CTLE LF Max	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Limits CTLE LF maximum.
GAINHF	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Inputs CTLE High Frequency Gain.
GAINHF Fixed/ Adaptable	**Fixed, Adaptable**	Inputs CTLE High Frequency Gain - Fix or Adaptable options.
CTLE HF Min	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Limits CTLE HF minimum.
CTLE HF Max	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Limits CTLE HF maximum.
RF_P2	**Fw_default, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10**	RF_P2 setting.
RF_P2 Fixed/Adaptable	**Fixed, Adaptable**	RF_P2 setting - Fix or Adaptable options.
RF_P2 Min	**Fw_default, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10**	Limits RF_P2 minimum.
RF_P2 Max	**Fw_default, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10**	Limits RF_P2 maximum.
RF_P1	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	RF_P1 setting.
RF_P1 Fixed/Adaptable	**Fixed, Adaptable**	RF_P1 setting - Fix or Adaptable options.
RF_P1 Min	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Limits RF_P1 minimum.
RF_P1 Max	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Limits RF_P1 maximum.

continued...

Parameter	Value	Description
RF_P0	**Fw_default, -15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Sets RF_P0.
RF_P0 Fixed/Adaptable	**Fixed, Adaptable**	Limits RF_P0 - Fix or Adaptable options.
RF_B0T	**Fw_default, 10, 15, 20, 25, 30, 35, 40, 45, 50**	Controls the rate RF_B0 adapts.

Related Information

My Intel Support

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 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® Streaming (Avalon-ST) interface which supports data bursts or packets, which are carried in the Interlaken meta frame 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.

Interlaken Look-aside Interface

The Interlaken Look-aside interface complies with the *Interlaken Look-Aside Protocol Specification, Revision 1.1*. It is the high-speed transceiver interface to an Interlaken link.

Management Interface

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

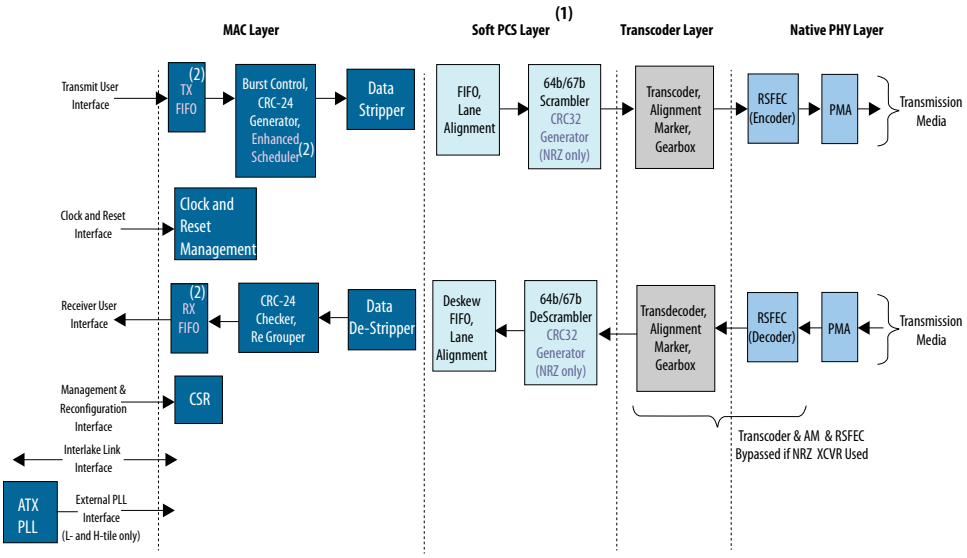
- **External PLL Interface**

  The Interlaken IP variations that target an Intel Stratix 10 L-tile or H-tile device require an external transceiver PLL to function correctly in hardware. The Interlaken IP variations that target an Intel Stratix 10 or Intel Agilex E-tile device include transceiver PLLs and do not require an external PLL. The E-tile PAM4 mode variations require an additional `mac_clkin` input clock generated by a PLL. This PLL must use the same reference clock source that drives the pll_ref_clk.

- **Transceiver Reconfiguration Interface**

  The transceiver reconfiguration interface provides access to the registers in the embedded Native PHY IP. This interface provides direct access to the hard PCS registers on the device. This interface complies with the Avalon-MM specification defined in the *Avalon Interface Specifications*.

Figure 7. Interlaken (2nd Generation) Intel FPGA IP High Level System Overview



(1) The PCS layer is a soft layer for E-tile devices and hard layer for L- and H-tile devices.
(2) TX FIFO, RX FIFO, and Ehanced Scheduler blocks are not present in Interlaken Look-aside solution.

Related Information

- Interlaken Protocol Specification, v1.2
- Avalon Interface Specifications
- Interface Signals on page 72
- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- E-Tile Transceiver PHY User Guide
- Interlaken Look-Aside Protocol Definition

4.2. IP Clocks

Table 19. Interlaken IP Core Clocks

Clock Name	Device	Direction	Description
`pll_ref_clk`	• Intel Stratix 10 L-, H-, and E-tile • Intel Agilex E-tile	Input	Reference clock for the RX CDR PLL.
`tx_serial_clk[NUM_LANES-1:0]`	Intel Stratix 10 L- and H-Tile	Input	Clocks for the individual transceiver channels in Interlaken IP.
`rx_usr_clk`	• Intel Stratix 10 L-, H-, and E-tile • Intel Agilex E-tile	Input	Clock for the receive application interface. *Note:* This clock is not present in Interlaken Look-aside IP core variations.
`tx_usr_clk`	• Intel Stratix 10 L-, H-, and E-tile • Intel Agilex E-tile	Input	Clock for the transmit application interface. *Note:* This clock is not present in Interlaken Look-aside IP core variations.
`reconfig_clk`	• Intel Stratix 10 L-, H-, and E-tile • Intel Agilex E-tile	Input	Management clock for hard PCS register access, including access for transceiver reconfiguration and testing features. Refer to the appropriate *Transceiver PHY User Guide* for the frequency of the `reconfig_clk`.
`mm_clk`	• Intel Stratix 10 L-, H-, and E-tile • Intel Agilex E-tile	Input	Management clock for Interlaken IP core register access. Intel recommends that you use the `mm_clk` value same as the `reconfig_clk`.
`clk_tx_common`	• Intel Stratix 10 L-, H-, and E-tile • Intel Agilex E-tile	Output	Transmit PCS common lane clock driven by the SERDES transmit PLL.
`clk_rx_common`	• Intel Stratix 10 L-, H-, and E-tile • Intel Agilex E-tile	Output	Receive PCS common lane clock driven by the CDR in transceiver.
`mac_clkin`	Intel Stratix 10 and Intel Agilex E-Tile (PAM4 only)	Input	This signal must be driven by a PLL. This PLL must use the same clock source that drives the `pll_ref_clk` . The value of `mac_clkin` signal is 396 MHz.

Related Information

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

4.3. High Level Data Path Flow for Interlaken Mode

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 8. Interlaken IP Core Block Diagram for L- , H- and E-Tile NRZ Mode Device Variations

The figure illustrates the 8-word data transfer scenario. This figure uses the following conventions:

- n = Number of calendar pages
- m = Number of lanes

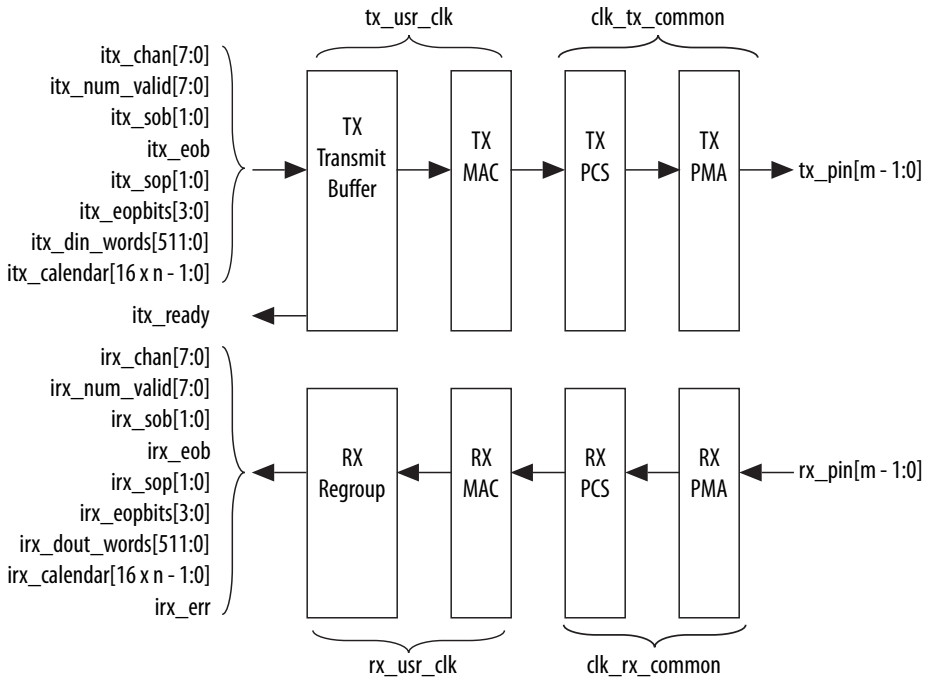
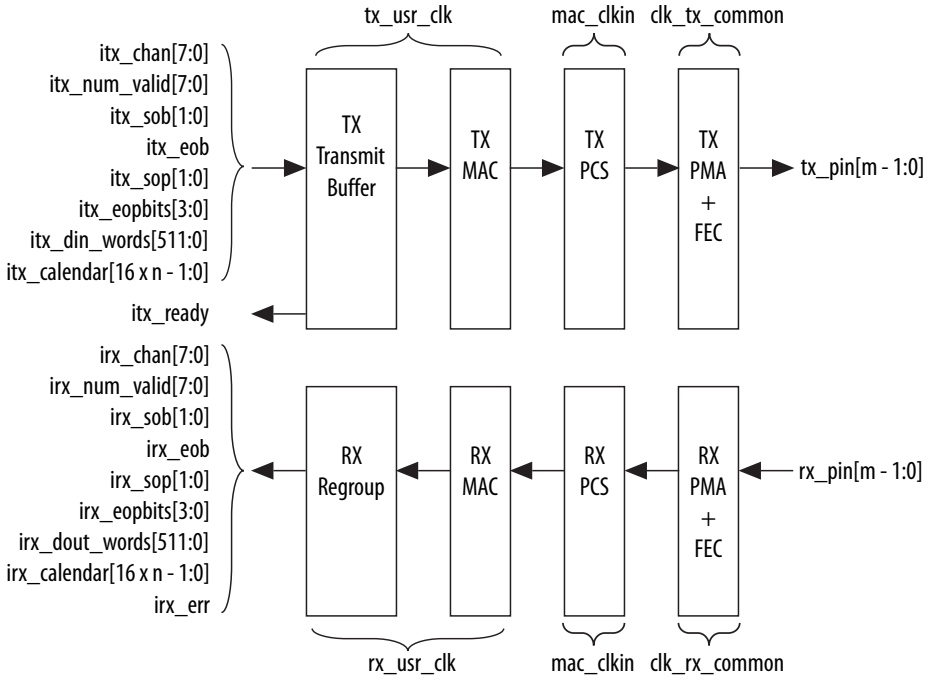


Figure 9. Interlaken IP Core Block Diagram for E-Tile PAM4 Mode Device Variations

The figure illustrates the 16-word data transfer scenario. This figure uses the following conventions:

- n = Number of calendar pages
- m = Number of lanes



Related Information

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

4.3.1. 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.

4.3.1.1. 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 10. Interlaken IP Core Transmit Path Blocks for L- , H- and E-Tile NRZ Mode Device Variations

The following figure illustrates the 8-word data transfer scenario:

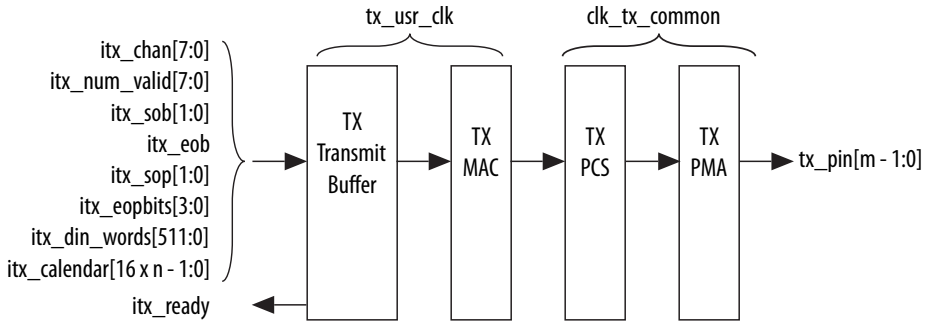
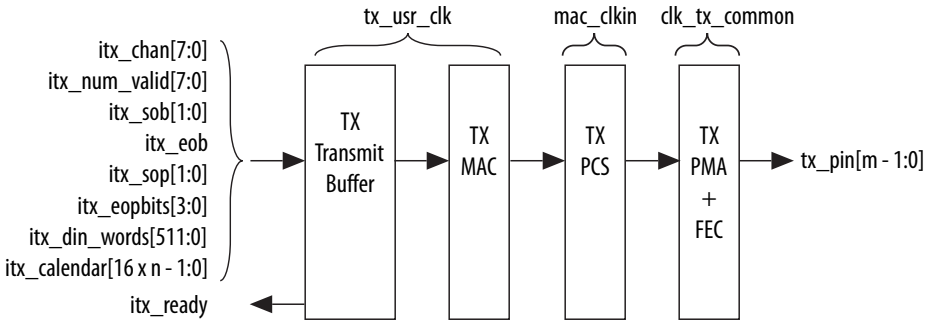


Figure 11. Interlaken IP Core Transmit Path Blocks for E-Tile PAM4 Mode Device Variations

The following 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_din_words` 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.
- Stripes and de-stripes between the user data (data width) and the number of lanes. Refer to *Table 2: IP Core Theoretical Raw Aggregate Bandwidth* in this document for more information on supported combinations.
- 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 L- and H-tile device variations, TX PCS logic is an embedded hard macro and does not consume FPGA soft logic elements. In E-tile device variations, the FPGA soft logic implements TX PCS. In PAM4 mode, the E-tile device variations contain a soft logic transcoder block to work with RS FEC (544, 514) 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.
- Performs asynchronous operations and transmission lane alignment using TX Align FIFO.
- Performs the Interlaken transcoding function to support the RS FEC (544, 514) of the TX PMA in PAM4 mode applications.

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 E-tile devices and three RS FEC (544,514) blocks in 6x 53.125 Gbps PAM4 mode configuration. Each RS FEC block serves four FEC channels in the aggregate mode.

Note: For E-tile variants: The *Interlaken Alliance Interoperability Recommendations v1.11 section 2.5.2* suggests doubling the skew budget from 107UI to 214UI for data lane bit rates higher than 6.25Gbps. TX skew of the Interlaken (2nd Generation) Intel FPGA IP on Intel Stratix 10 or Intel Agilex E-Tile devices is usually less than 130UI. There is a theoretical possibility that on a given reset, channel-to-channel skew can be as much as 258 UI. This is not observed in practical scenerio. A reset of the Interlaken (2nd Generation) Intel FPGA IP recovers the skew to less than 214UI as per the *Interlaken Alliance Interoperability Recommendations*. The Interlaken (2nd Generation) Intel FPGA IP supports significantly more than 258 UI skew tolerance, as can many products in the market.

For L-tile and H-tile variants, soft PCS bonding is implemented, PMA and PCS bonding is not enabled. The TX skew is about 39 UI.

Pin out your Interlaken IP core to exist within a single tile of Intel Stratix 10 device. If you have to pin out your core across multiple tiles, please contact Intel Premier Support.

Related Information

- E-Tile Transceiver PHY User Guide
- Device Family Support on page 9
  For more information on supported combinations of number of lanes and data rates.

4.3.1.2. In-Band Calendar Bits on Transmit Side

When you turn on **Include in-band flow control functionality**[9],

- 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].

[9] The **Include in-band flow control functionality** option is not available when you turn on **Enable Interlaken Look-aside mode** parameter.

Table 20. 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.

4.3.2. 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.

4.3.2.1. 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 12. Interlaken IP Core Receive Path Blocks for L-, H- and E-Tile NRZ Mode Device Variations

The figure illustrates the eight word data transfer scenario. This figure uses the following conventions:

- n = Number of calender pages
- m = Number of lanes

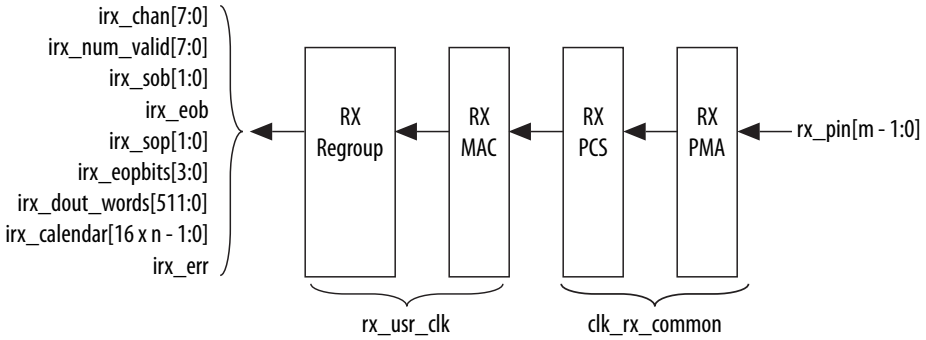
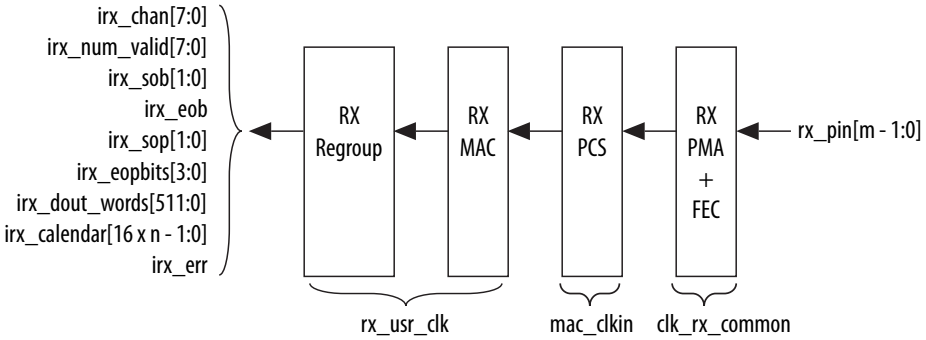


Figure 13. Interlaken IP Core Receive Path Blocks for E-Tile PAM4 Mode Device Variations

The figure illustrates the eight word data transfer scenario. This figure uses the following conventions:

- n = Number of calender pages
- m = Number of lanes



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

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

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 E-tile devices and three RS FEC (544,514) blocks in 6x 53.125 Gbps PAM4 mode configuration. Each RS FEC block serves four FEC channels in the aggregate mode.

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 E-tile devices. In PAM4 mode, the 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.
- Performs asynchronous operations and receiver alignment using RX Align FIFO.
- Performs the Interlaken inverse transcoding function on the data received from the RX RS FEC (544, 514) in E-tile PAM4 mode device variations.

For more information about error conditions, refer to the `ILKN_FEC_XCODER_TX_ILLEGAL_STATE` (offset 0x80) and `ILKN_FEC_XCODER_RX_UNCOR_FECCW` (offset 0x81) registers. You can also obtain more details from the FEC status, FEC correctable and uncorrectable registers documented in the *RS-FEC Registers* section of the *E-Tile Transceiver PHY User Guide*.

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_dout_words` format.

For details on transceiver initialization, please refer to the *Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide* or the *E-Tile Transceiver PHY User Guide*.

Related Information

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

4.3.2.2. 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**[10], 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.

4.3.2.3. 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

(10) The **Include in-band flow control functionality** option is not available when you turn on **Enable Interlaken Look-aside mode** parameter.

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:

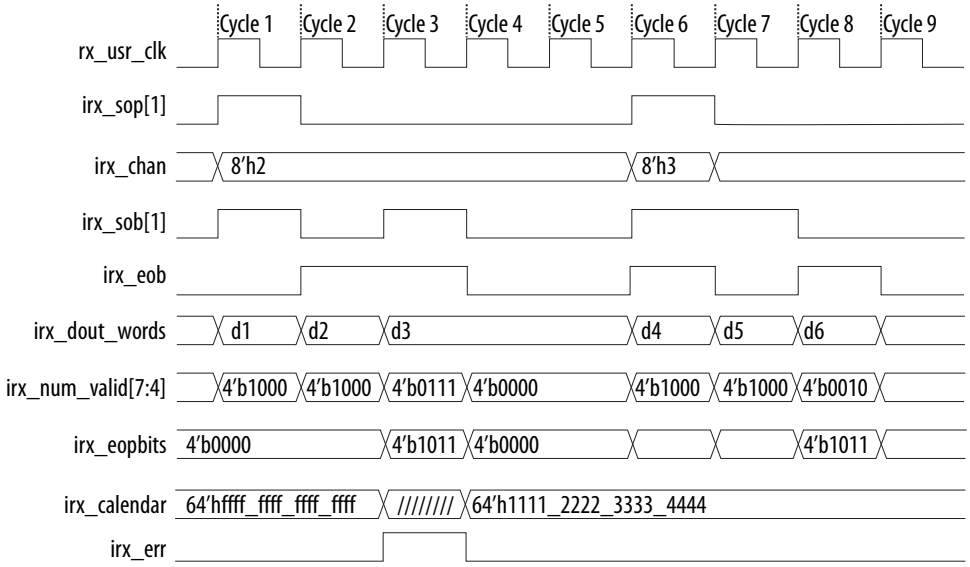
- If you monitor the link when only idle control words are being received (no data is flowing), you should monitor the real time status signal `crc24_err`.
- If you maintain a count of CRC24 errors, you should monitor the number of times that the real time status signal `crc24_err` is asserted.

4.3.2.3.1. 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 14. 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.

4.3.3. Unused Transceiver Channels

Unused transceiver channels can degrade in performance over time. To preserve the performance of unused transceiver channels, the Intel Quartus Prime software can switch the TX and RX channels on and off at a low frequency using a reference clock.

For more information about preserving the performance of unused transceiver channels refer to *Unused or Idle Transceiver Channels* section in the *Intel Stratix 10 L- and H-tile Transceiver PHY User Guide* and *Unused Transceiver Channels* section in the *E-Tile Transceiver PHY User Guide*.

Related Information

- Unused or Idle Transceiver Channels
- Unused Transceiver Channels

4.4. High Level Data Path Flow for Interlaken Look-aside Mode

The Interlaken Look-aside mode consists of two paths:

- Interlaken Look-aside TX path
- Interlaken Look-aside RX path

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

Figure 15. Interlaken Look-aside Mode Block Diagram for E-Tile NRZ Mode Device Variations

The figure illustrates the eight word data transfer scenario. This figure uses the following conventions:

- m = Number of lanes

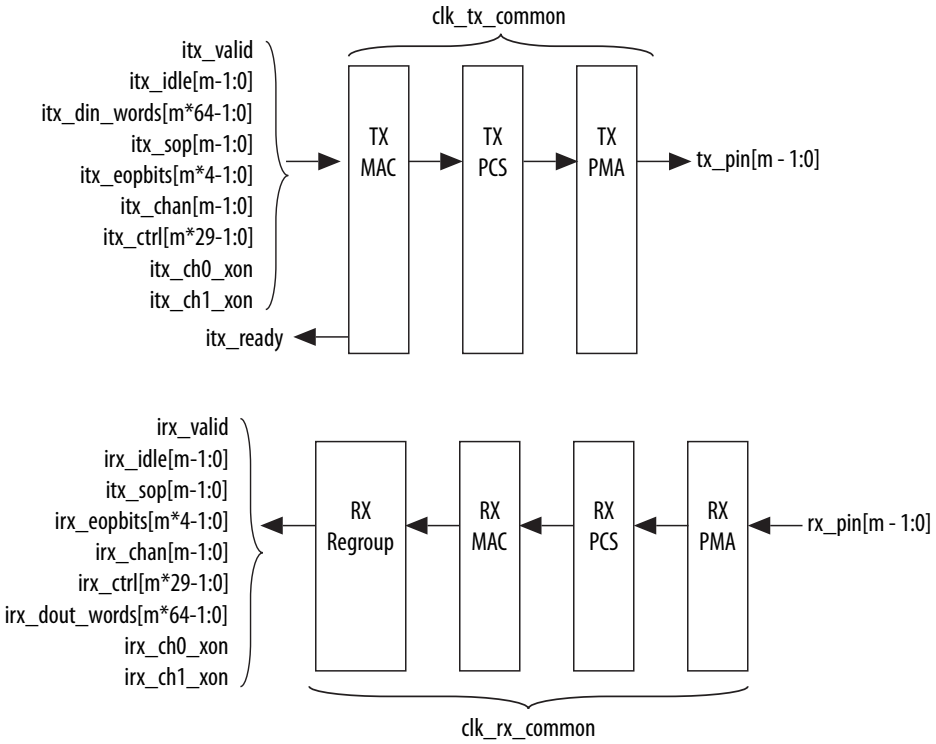
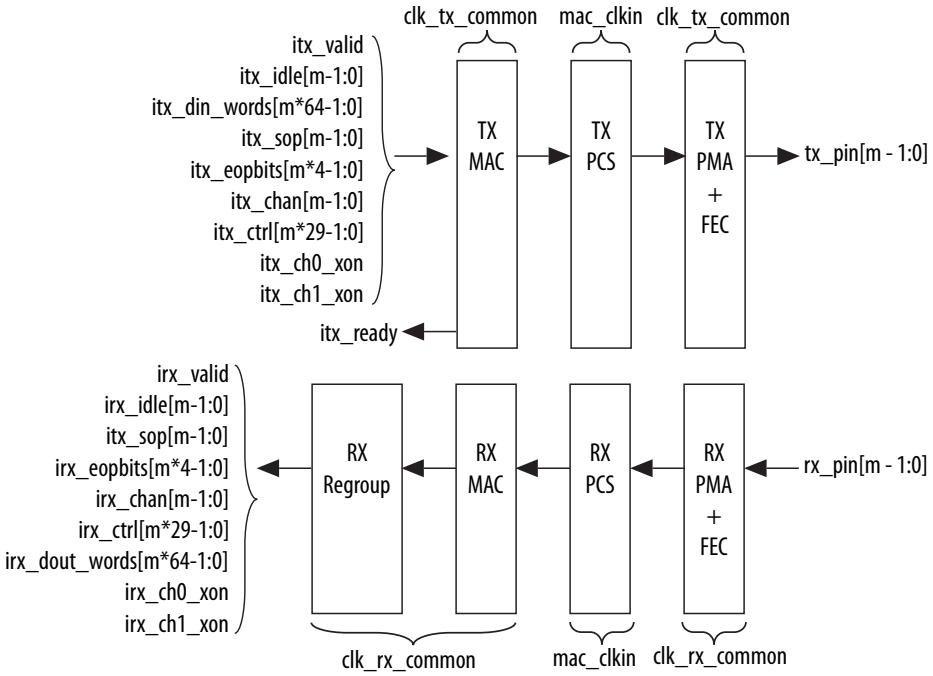


Figure 16. Interlaken Look-aside Mode Block Diagram for E-Tile PAM4 Mode Device Variations

The figure illustrates the eight word data transfer scenario. This figure uses the following conventions:

- m = Number of lanes



4.4.1. Interlaken Look-aside TX Path

4.4.1.1. Transmit Path Blocks

The Interlaken Look-aside mode transmit data path has the following three main functional blocks:

- TX MAC
- TX PCS
- TX PMA

Figure 17. Interlaken Look-aside IP Core Transmit Path Blocks for L- , H-, and E-Tile NRZ Mode Device Variations

The figure illustrates the eight word data transfer scenario.

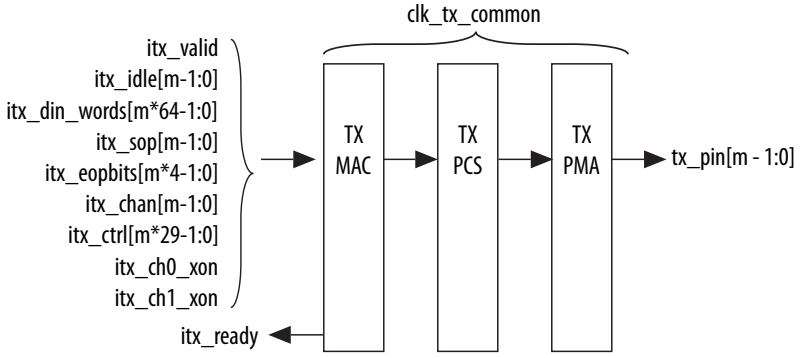
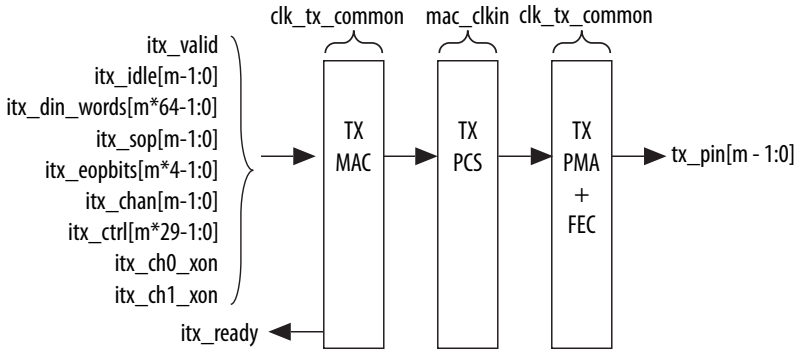


Figure 18. Interlaken Look-aside IP Core Transmit Path Blocks for E-Tile PAM4 Mode Device Variations

The figure illustrates the eight word data transfer scenario.



TX MAC

The Interlaken Look-aside IP core TX MAC performs the following functions:

- Populates burst and idle control words in the incoming data stream to align with the Interlaken Look-aside 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.

TX PCS

In E-tile device variations, the FPGA soft logic implements TX PCS. In PAM4 mode, the E-tile device variations contain a soft logic transcoder block to work with RS FEC (544, 514) 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.
- Performs asynchronous operations and transmission lane alignment using TX Align FIFO.
- Performs the Interlaken transcoding function to support the RS FEC (544, 514) of the TX PMA in PAM4 mode applications.

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 E-tile devices and three RS FEC (544,514) blocks in 6x 53.125 Gbps PAM4 mode configuration. Each RS FEC block serves four FEC channels in the aggregate mode.

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 Intel Premier Support.

Pin out your Interlaken IP core to exist within a single tile of Intel Stratix 10 device. If you have to pin out your core across multiple tiles, please contact Intel Premier Support.

4.4.2. Interlaken Look-aside RX Path

4.4.2.1. Receive Path Blocks

The Interlaken Look-aside mode receive data path has the following four main functional blocks:

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

Figure 19. Interlaken Look-aside IP Core Receive Path Blocks for E-Tile NRZ Mode Device Variations

The figure illustrates the eight word data transfer scenario. This figure uses the following conventions:

- m= Number of lanes

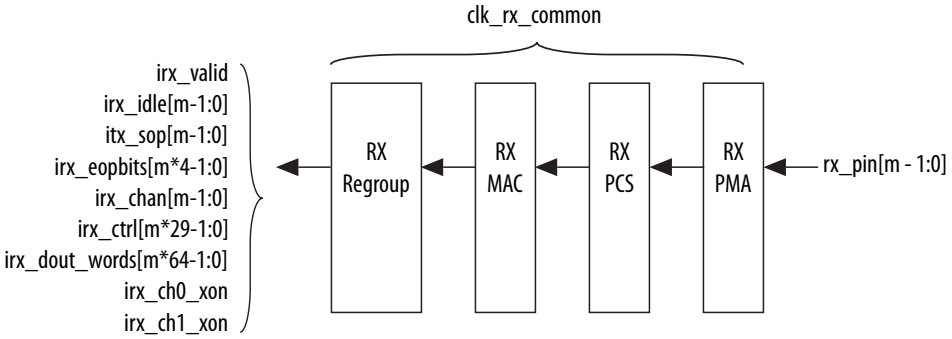
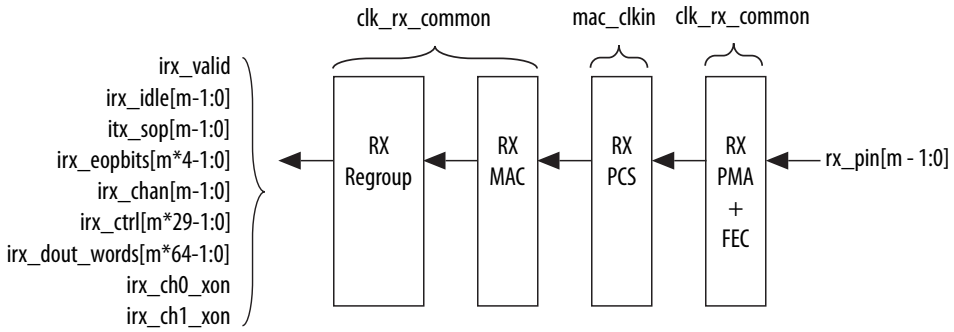


Figure 20. Interlaken Look-aside IP Core Receive Path Blocks for E-Tile PAM4 Mode Device Variations

The figure illustrates the eight word data transfer scenario. This figure uses the following conventions:

- m= Number of lanes



RX Regroup

The Interlaken Look-aside IP core RX regroup block translates the IP core internal data format to the outgoing user application data

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.

RX PCS

The FPGA soft logic implements RX PCS in E-tile devices. In PAM4 mode, the 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.
- Performs asynchronous operations and receiver alignment using RX Align FIFO.
- Performs the Interlaken inverse transcoding function on the data received from the RX RS FEC (544, 514) in E-tile PAM4 mode device variations.

For more information about error conditions, refer to the `ILKN_FEC_XCODER_TX_ILLEGAL_STATE` (offset 0x80) and `ILKN_FEC_XCODER_RX_UNCOR_FECCW` (offset 0x81) registers. You can also obtain more details from the FEC status, FEC correctable and uncorrectable registers documented in the *RS-FEC Registers* section of the *E-Tile Transceiver PHY User Guide*.

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

4.4.2.2. RX Error Handling

In Interlaken Look-aside mode, the IP checks for CRC24 errors in incoming data stream. For CRC24 errors, you should use `crc24_err` signal similar to Interlaken mode. The `crc24_err` signal can appear after `irx_valid` signal.

If there is a data word just before CRC24 errored control word, the IP copies the CRC24 error bit to the location of the last word as well as original location of CRC24 error.

There is no checking on the validity of each field of control word other than CRC24. The corrupted bit exposed to the user interface without modification. Ignore burst which has CRC24 error. Use error handling for invalid fields.

4.5. Modes of Operation

4.5.1. 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.

Note: The Interlaken Look-aside feature transfers data only in packet mode. It does not support interleaved 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.

4.5.2. Interlaken Mode

The following sections cover examples of user data interfaces in Interlaken mode.

4.5.2.1. Transmit User Data Interface Examples

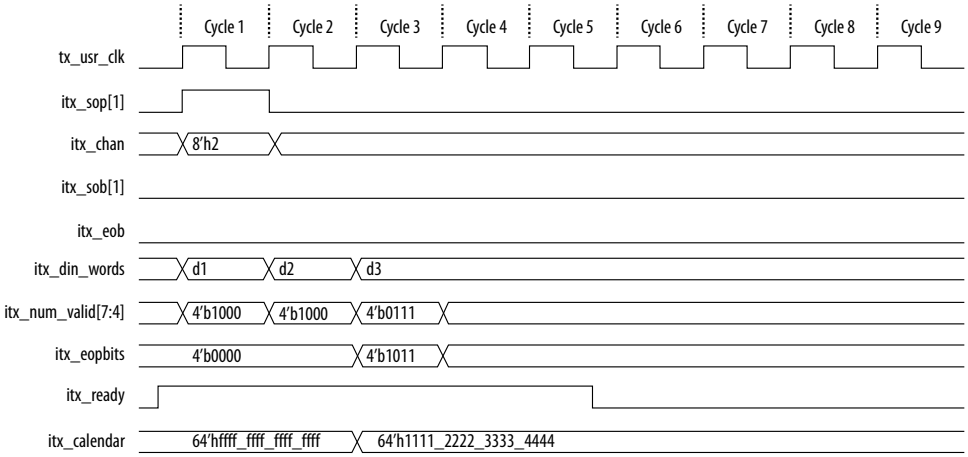
The following examples illustrate how to use the Interlaken IP core TX user data interface:

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 21. Packet Transfer on Transmit Interface in Packet Mode

The figure illustrates a packet mode data transfer (eight 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.

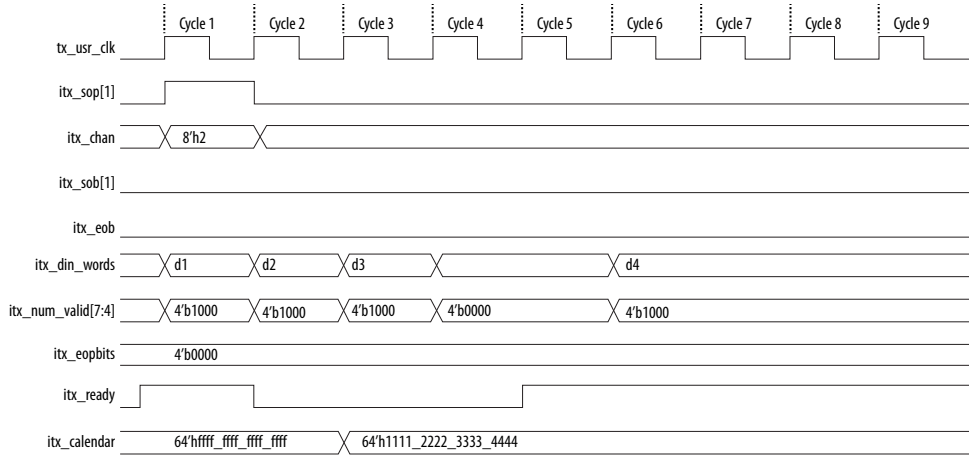
The total packet length is 64 + 64 + 51 = 179 bytes.

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 22. Packet Transfer on Transmit Interface with Back Pressure

The figure illustrates timing diagram for packet mode data transfer with back-pressure on the transmit interface.



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.

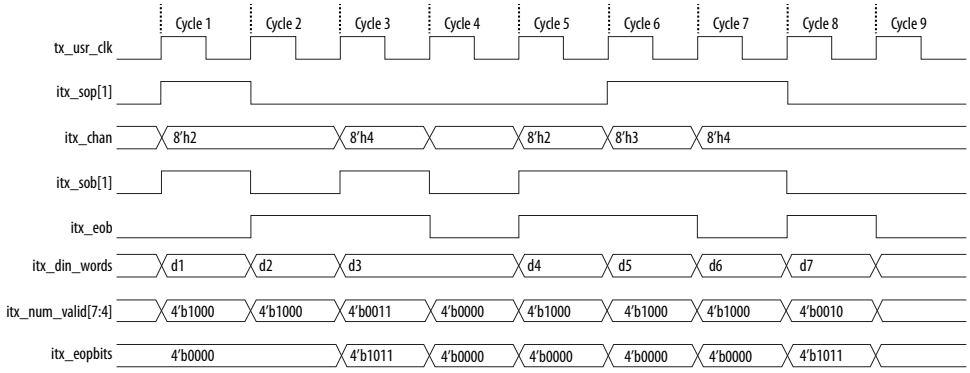
You can think of this interface as a FIFO write interface. When `itx_num_valid[7:4]` is nonzero, both data and control information (including `itx_num_valid[7:4]` itself) are written to the transmit side data interface. The `itx_ready` signal is the inverse of a hypothetical FIFO-almost-full flag. When `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 six to eight clock cycles of `tx_usr_clk`.

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, if you set the SOB and EOB flags correctly to establish the data boundaries.

Figure 23. Packet Transfer on Transmit Interface in Interleaved Single Segment Mode

The figure shows the timing diagram for an interleaved data transfer (eight 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 of that channel. 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 if 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.

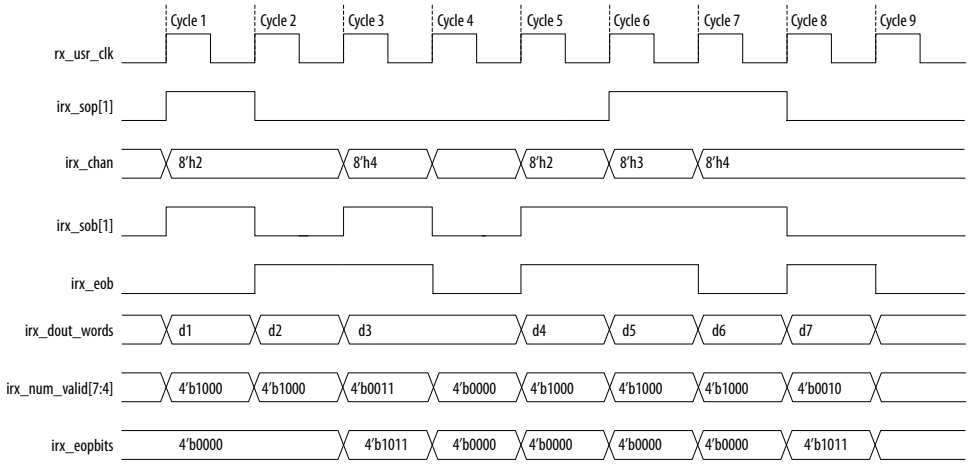
If you do not drive the `itx_sob` and `itx_eob` signals, the Interlaken IP Core does not operate properly and the transmit FIFO may overflow, since in this mode the internal logic is looking for `itx_sob` and `itx_eob` assertion for insertion of proper burst control words.

4.5.2.2. 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 24. 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.

4.5.3. Interlaken Look-aside Mode

The following sections cover examples of user data interfaces in Interlaken Look-aside mode. The IP only supports packet mode for user data transfer when you enable Interlaken Look-aside feature.

4.5.3.1. Transmit User Data Interface Example

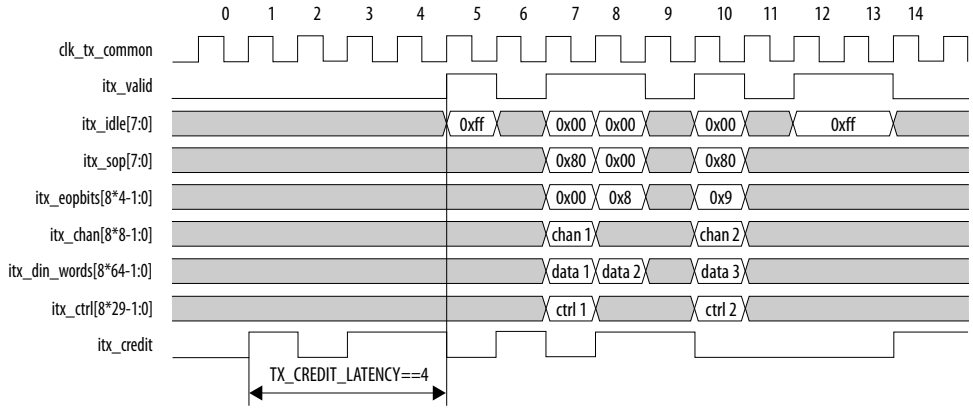
The following example illustrate how to use Interlaken Look-aside TX user data interface during a normal packet transfer:

Packet Mode Transfer Example

This example illustrates the expected behavior of the Interlaken Look-aside application interface transmit signals during a packet transfer in packet mode.

Figure 25. Packet Transfer on Transmit Interface in Packet Mode

The following figure illustrates two packet transfer of 120 bytes and 49 bytes on the transmit interface with number of lanes is equal to eight and TX_CREDIT_LATENCY is equal to four.



You must assert `itx_valid` after TX_CREDIT_LATENCY cycle from the assertion of `itx_credit` even when `itx_ready` deasserted.

In cycle 5, the IP core detects assertion of `itx_idle` and all of `itx_idle` bits are asserted indicating that eight idle control words can be inserted in this slot.

In cycle 6, `itx_valid` signal is deasserted indicating that all of other signals are not valid. `itx_valid` signal is deasserted because TX_CREDIT_LATENCY cycles before, `itx_credit` was deasserted.

In cycle 7, `itx_valid` is asserted and `itx_idle` is 0x0, which indicates `itx_sop`, `itx_eopbits`, and `itx_din_words` signals. Assert the `itx_sop[7]` to indicate the start of packet. Burst control word can be inserted at this location and seven data

words in itx_din_words[447:0] should follow it. Corresponding itx_chan[7] indicates the channel associated with this data packet following the control word. The IP ignores the rest of itx_chan [6:0] bits.

In cycle 8, the transfer of data packet should be terminated since itx_eopbits[3:0] is not zero. The seven data words in itx_data[511:0] should be transferred as a part of data packet. itx_chan is invalid in this cycle because itx_sop bits are not set. itx_eopbits[3:0] can be set to 4'b1000 to indicate the end of packet and last word contains 8 bytes. This eopbits field can be set to EOP_Format field of following burst control words.

In cycle 9, the itx_valid should be de-asserted because before the TX_CREDIT_LATECNY cycles, itx_credit was deasserted.

In cycle 10, the transfer of new data packet starts and this can be terminated since itx_eopbits[3:0] is not zero. In this cycle, user sends 49 bytes ti the IP core. The itx_eopbits[3:0]= 4'b10001 indicates that the last word (tx_data[63:0]) only contains one byte. This itx_eopbits[3:0] value can be set to EOP_Format field of following IDLE control word.

In cycle 11, itx_valid should be asserted because before TX_CREDIT_LATENCY cycles, itx_credit was deasserted.

In cycle 12, user logic asserts itx_idle[7:0] indicating that it wants to terminate the data burst and send eight idle control words.

This signal behavior correctly transfer first data packet as follows:

- In cycle 7, the IP core receives burst control word and 56 bytes of valid data (data 1)
- In cycle 8, the IP core receives 64 bytes of valid data (data 2)

The total packet length for the first packet is 56 + 64 = 120 bytes.

This signal behavior correctly transfer second data packet as follows:

- In cycle 7, the IP core receives another burst control word and 49 bytes of valid data (data 3)

The total packet length for the first packet is 49 bytes.

4.5.3.2. Receive User Data Interface Examples

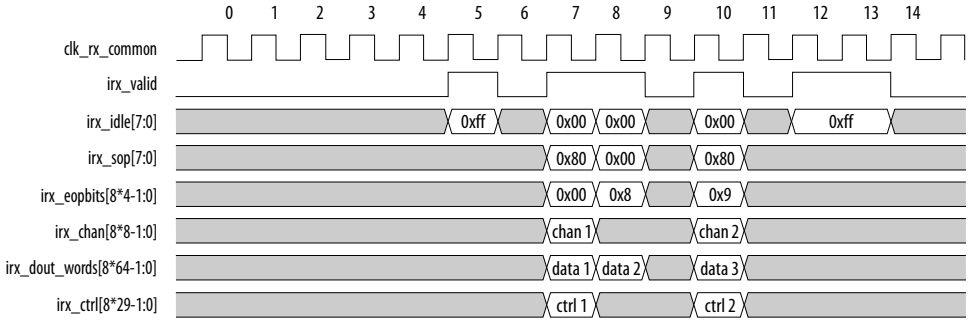
The Interlaken Look-aside mode only supports packet mode data transfer. The following example illustrate how to use Interlaken Look-aside RX user data interface during a normal packet transfer:

Packet Mode Transfer Example

This example illustrates the expected behavior of the Interlaken Look-aside application interface receive signals during a packet transfer in packet mode.

Figure 26. Packet Transfer on Receive Interface

The following figure illustrates two packet transfer of 120 bytes and 49 bytes on the transmit interface with number of lanes is equal to eight and TX_CREDIT_LATENCY is equal to four.



From cycle 0 to cycle 4, there is no valid data drop by the IP as `irx_valid` deassert.

In cycle 5, the IP core asserts `irx_valid`, and all of `irx_idle` bits indicating that `irx_dout_words` carry idle control words.

In cycle 6, the IP deasserts `irx_valid` signal again indicating all of other signals are invalid.

In cycle 7, the IP core asserts `irx_valid` with `irx_idle[7:0]` is equal to 8'h0, which indicates `irx_sop`, `irx_eopbits`, and `irx_dout_words` carry valid data. The assertion of `itx_sop[7]` indicates the start of packet at the leftmost words. Corresponding `irx_chan[7]` indicates the channel associated with this data packet following this control word. With `irx_eopbits` is 0x0, `irx_dout_words` represents the actual data words of packet.

In cycle 8, the assertion of `irx_eopbits[3]` terminates the transfer of data, which happens on the rightmost words. All data words on `irx_dout_words[511:0]` are the last words of the packet. `irx_eopbits[3:0]`== 4'b1000 indicates all 8 bytes of the rightmost data word are valid.

In cycle 9, IP core deasserts `irx_valid` signal again indicating that all of the other signals are not valid.

In cycle 10, IP core flushes new packet with assertion of `irx_sop[7]` and `irx_valid`. The packet terminated in the same cycle as well with assertion of `irx_eopbits[3]`. The `rx_eopbits[3:0]`== 4'b1001 indicates only 1 byte of the rightmost words is valid. In this cycle, IP core outputs 49 bytes of data which is accessible from `irx_dout_words[447:56]`.

In cycle 11, IP core deasserts `irx_valid` signal again indicating that all of other signals are not valid.

In cycle 12, IP core asserts `irx_valid` and all of `irx_idle` bits indicating that `irx_dout_words` carry idle control words.

This signal behavior correctly transfers first data packet as follows:

- In cycle 7, the IP core outputs burst control word and 56 bytes of valid data (data 1).
- In cycle 8, the IP core output 64 bytes of valid data (data 2).

The total packet length for the first packet is 56+64= 120 bytes.

This signal behavior correctly transfers second data packet as follows:

- In cycle 10, the IP core outputs burst control word and 49 bytes of valid data (data 3).

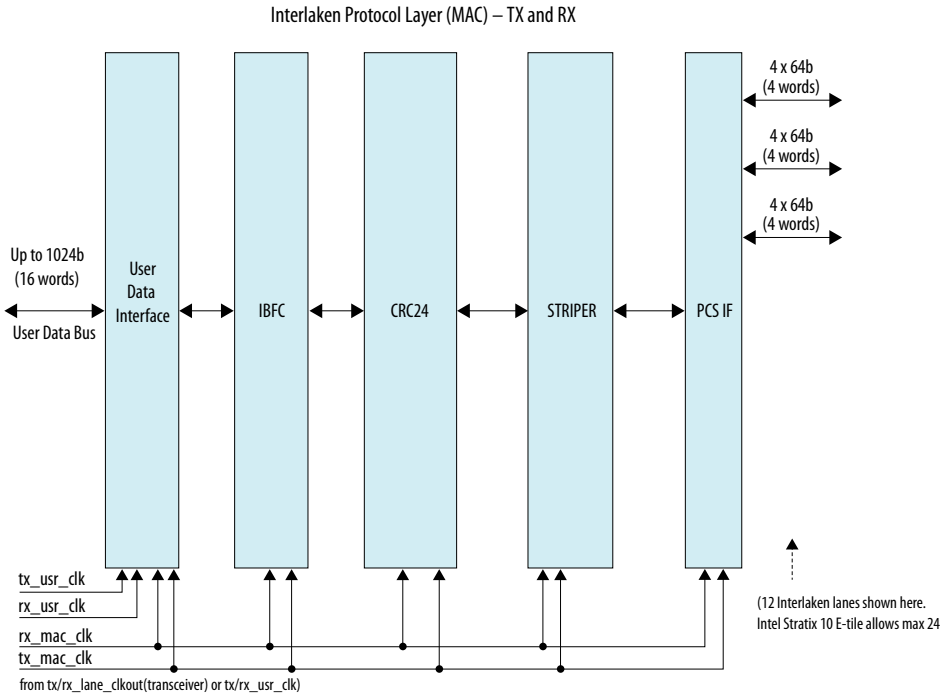
The total packet length for the first packet is 49 bytes.

4.5.4. Multi-Segment Mode

This section describes the functionality of the multi-segment feature of the Interlaken (2nd Generation) IP. This feature enables you to make better use of the transmit and receive bandwidth. For package sizes smaller than the user data width, this feature becomes important to provide better bandwidth efficiency.

The multi-segment mode feature is not available with Interlaken Look-aside.

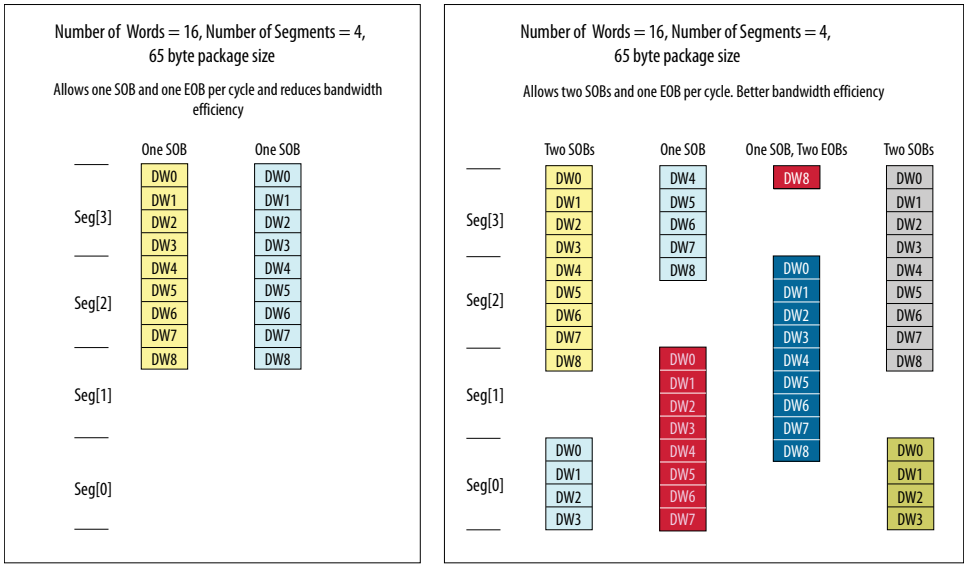
Figure 27. Interlaken Protocol Layer (MAC)- TX and RX Block Diagram



The user data interface block contains the multi-segment blocks, RX regroup, and TX transmit buffer. In the TX direction, the TX transmit buffer inserts necessary control words based on the user interface control information before sending downstream for further processing. In the RX direction, the striper strip offs the control words of the data to generate the user data and the extracted control word generates the user interface information.

The bandwidth efficiency of the four-segment case is significantly better than the single-segment case. Each segment supports both **Interleaved** and **Packet** transfer modes when you operate the IP in multi-segment mode.

Figure 28. Four Segment versus Single Segment Bandwidth Efficiency Comparison



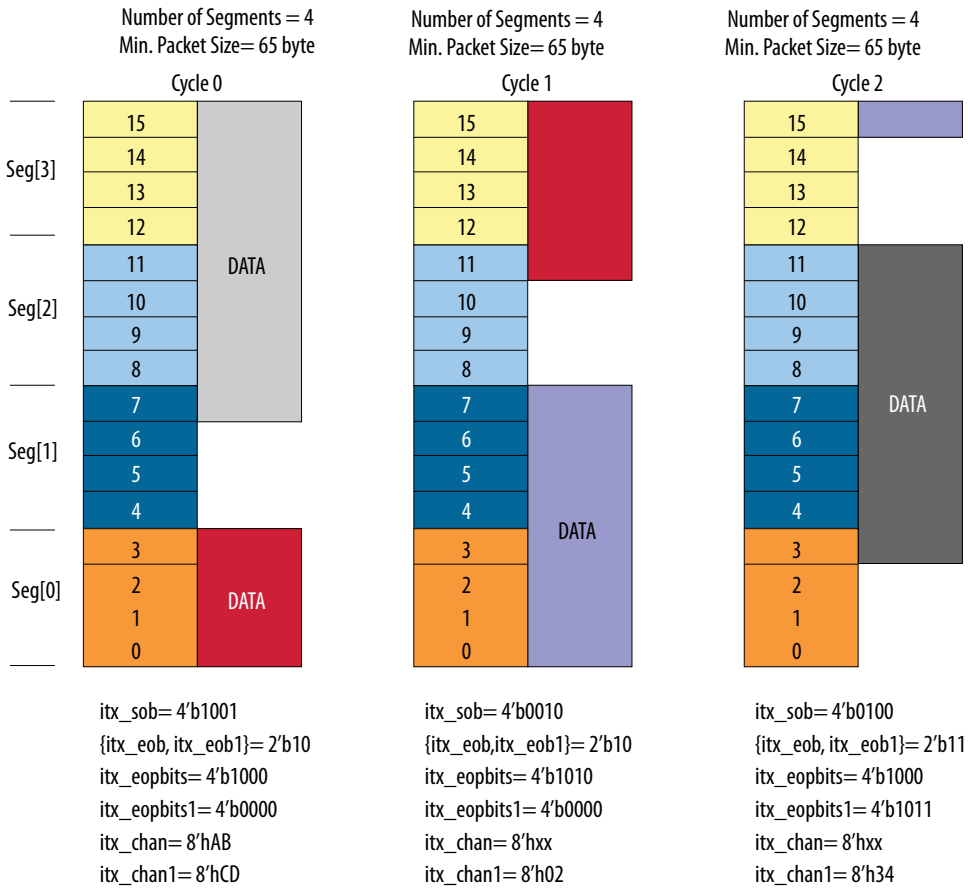
4.5.4.1. Multi-Segment Interface Example

The following example illustrates how to use the Interlaken IP core TX and RX multi-segment interface:

TX Multi-Segment

This example illustrates the expected behavior of multi-segment mode Interlaken IP core at the TX user interface during a 65 byte packet transfer in four-segment interleave mode.

Figure 29. TX Multi-Segment User Interface Example



Even though there are 4-bit allocated to the sob and sop bus, only a maximum of two segment chunks are supported in any given cycle. Because of this reason, you need only two-bit eob, two sets of eopbits and channel numbers.

In this example, in cycle 0, the second segment starts at segment 0 while the first segment data occupies segment 3, 2, and 1. In cycle 1, the first segment of data occupies segment 3 and 2 while the second segment of data occupies the segment 0 and 1.

There are two burst starting in cycle 0. In sob= 4'b1001, the sob[3] refers to the start of a burst at word 15. The sob[0] refers to another start of a burst at word 3. The packet ends in cycle 0. Hence, eopbits=4'b1000 indicates the last word containing eight bytes. eob=1'b1 is for the first segment chunk. The eopbits1 indicates no end of packet since eopbits1[3]=0. The two channel numbers are h'AB and h'CD, respectively.

There is one burst ends and one burst starts in cycle 1. In sob= 4'b0010, the burst in the first segment continues the second segment of the last cycle. There is no sob[3] set in this cycle. The sob[1]=1 of the second segment refers to the start of a burst at word 7. The packet (first segment chunk) ends in this cycle, hence eopbits=4'b1010 indicating the last word contains two bytes. The eob=1'b1 is for the first segment

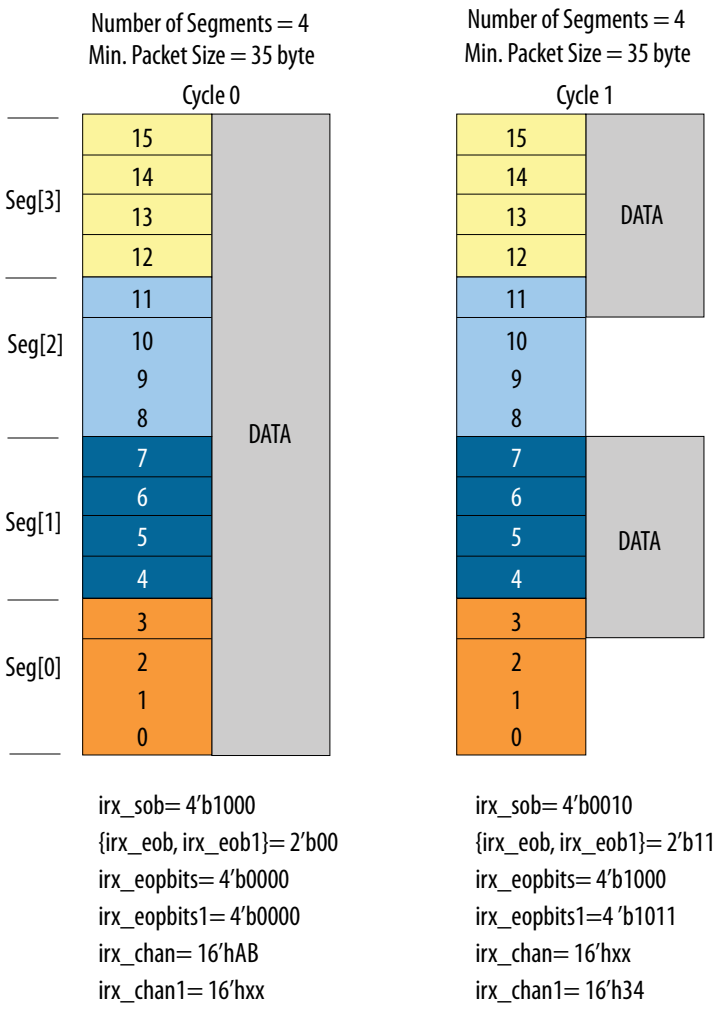
chunk. The eopbits1 indicates no EOP since eopbits1[3]=0. Only the channel number for the second segment chunk is valid, which is 'h02. This channel number corresponds to the burst indicated by sob[1]=1'b1. A channel number is always associated to a start of a burst.

There is one burst ends and one burst starts in cycle 2. In sob= 4'b0100, the burst in the first segment continues the second segment of the last cycle. There is no sob[3] set in this cycle. The packet ends in this cycle, hence, eopbits=4'h1000, indicates the last word containing eight bytes. The eob=1'b1 is for the first segment chunk. The sob[2]=1 of the second segment chunk refers to the start of burst at word 11. The packet ends in this cycle, hence eopbits1=4'b1011 indicates the last word containing three bytes. Also, eob1=1'b1 is the EOB for the second segment chunk. Only the channel number for the second segment chunk is valid here, which is 'h34. This channel number corresponds to the burst indicated by sob[2]=1'b1.

RX Multi-Segment

Below example illustrates behavior of multi-segment mode of the Interlaken IP core at the RX user interface.

Figure 30. RX Multi-Segment User Interface Example



4.6. Performance

You can measure the performance of Interlaken (2nd Generation) IP core in terms of the percentage of the raw bandwidth.

You can calculate the bandwidth performance by multiplying raw bandwidth with the efficiency factor using the formula below:

Actual Bandwidth = Raw Bandwidth * Efficiency Factor

The Efficiency Factor can be calculated using the formula below:

Efficiency Factor = (Encoding Efficiency) * (Framing Efficiency) * (Alignment Efficiency) * (Meta frame Efficiency) * 100% , where:

- Encoding Efficiency: 64B/67B encoding
- Framing Efficiency: The impact of the 8-byte control word overhead as a percentage of the frame or cell size
- Alignment Efficiency: The impact of invalid characters inserted to pad the end of a frame to an 8-byte word boundary
- Meta frame Efficiency: created by the Synchronization, Scrambler, State, Diagnostic, and Skip Words (assuming a meta frame length of 2K words, and not counting optional insertion of Idle Control Words for rate matching)

User efficiency examines how well the application logic occupies the full data width of the TX user data bus. This efficiency depends on the application logic implementation and transactions. Add the user efficiency to the efficiency factor equation to include efficiency degradation due to this user interface effect. You can use average efficiency numbers. Refer to *Performance* section of the Interlaken Protocol Definitions for more details.

The multi-segment feature of the Interlaken IP core improves the user efficiency by utilizing the user data bus more effectively.

Refer to the following example to understand how to calculate bandwidth performance.

Sample Calculation Example

This example is for the following input conditions:

- Number of lanes= 8
- Clock frequency = 395 MHz
- Meta frame size= 2048 word of 64 bit/word
- Constant burst size = 124 bytes

Use the following steps to calculate total bandwidth efficiency and actual bandwidth:

1. Calculate the raw bandwidth using the formula below:

   Raw bandwidth= 8 lanes * 64 bits/lane * 395 MHz = 202.2 Gbps

   This example assumes constant burst size. You can use an average burst size numbers for average bandwidth efficiency calculations.
2. Calculate each required efficiency to determine efficiency factor:

- Encoding efficiency = 64 bits/67 bits= 0.955

  Use 64B/67B encoding to count encoding efficiency.
- Framing efficiency = 16 words/17 words= 0.941

  124 bytes of data contain 16 Interlaken data words. One Interlaken burst control word is added to this 16 data words. Hence, the total number of words per bursts in 17.
- Alignment Efficiency = 124 bytes/128 bytes = 0.968

  The last Interlaken data word contains only 4 bytes of valid data instead of the full 8 bytes. In this 128 bytes (16-word) burst, only 124 byte data is valid.
- Meta frame Efficiency = 2024 words/2048 words= 0.99

  The 2048 Interlaken words contain four meta frame control words.

3. Total bandwidth efficiency = (Encoding Efficiency) * (Framing Efficiency) * (Alignment Efficiency) * (Meta frame Efficiency) * 100% = 0.955 * 0.941 * 0.968 * 0.99= 0.869= 86.9%
4. Actual bandwidth = Raw Bandwidth * Total bandwidth efficiency = 202.2 * 0.869 = 175.8 Gbps

4.7. 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.

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.

For details on transceiver initialization, please refer to the *Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide* or the *E-Tile Transceiver PHY User Guide*.

Related Information

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

4.8. M20K ECC Support

If you turn on **Enable M20K ECC support** in your 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.

Note: The IP does not include the ECC feature in the M20K blocks in Interlaken Look-aside IP mode.

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.

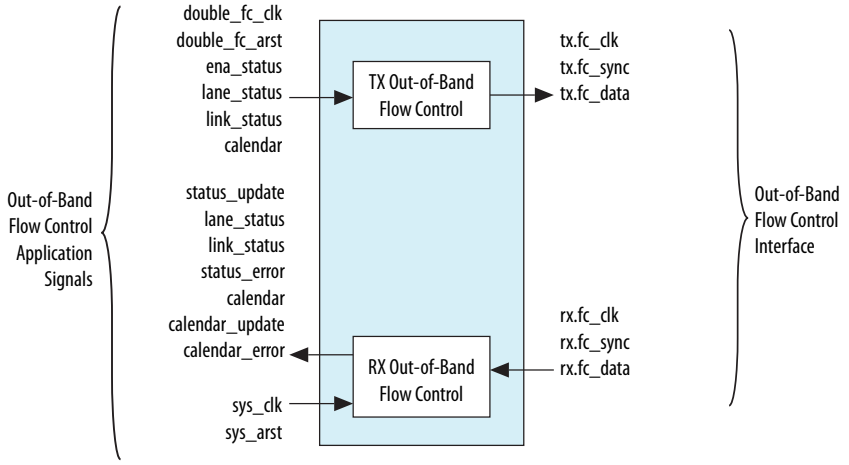
4.9. Out-of-Band Flow Control

The Interlaken IP core design example provides example logic to include the out-of-band flow control functionality. You can find this logic in the following location: `<design_example_installation_dir>/example_design/rtl`. The RX out-of-band flow control module is called `ilk_oob_flow_rx` and the TX out-of-band flow control is called `ilk_oob_flow_tx`. Alternatively, you can design and implement your own custom out-of-band control logic. 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*. This optional feature is intended for applications that require transmission rate control.

Note: The IP core does not include out-of-band flow control logic block when you generate the design example with Interlaken Look-aside feature enabled.

Figure 31. 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 21. 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 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 22. 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 23. 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 24. 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 25. 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.

Note: Throughout this chapter, ILK refers to Interlaken mode and ILA refers to Interlaken Look-aside mode.

Figure 32. IP Core Interface Signals

5.1. Clock and Reset Interface Signals

Table 26. Clock and Reset Interface Signals

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
`pll_ref_clk` / `pll_ref_clk[1:0]`(11)	ILK and ILA	1/2	Input	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:

continued...

<table>
<tr><th>Signal Name</th><th>Feature Support</th><th>Width (Bits)</th><th>I/O Direction</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><th>Per-Lane Data Rate (Gbps)</th><th>Valid pll_ref_clk Frequencies (MHz)</th></tr>
<tr><td>6.25</td><td>156.25, 195.3125, 250, 312.5, 390.625, 480.76923(13), 500(12), 625(12)</td></tr>
<tr><td>10.3125</td><td>156.25, 206.25, 257.8125, 322.265625, 412.5, 491.071428(13), 515.625(12), 644.53125(12)</td></tr>
<tr><td>12.5</td><td>156.25, 195.3125, 250, 312.5, 390.625, 500, 625(12)</td></tr>
<tr><td>25.28</td><td>126.4(12), 158.0, 197.5, 252.8, 320.0(12), 395.0, 486.153846(13), 505.6(12)</td></tr>
<tr><td>25.78</td><td>159.135802, 201.40625, 250.291262(12), 322.25, 402.8125, 495.769231(13), 500.582524(12)</td></tr>
<tr><td>25.78125</td><td>159.143519(12), 159.143518(13), 201.416016(12), 201.416015(13), 250.303398(12), 322.265625, 402.832031, 500.606796(12), 495.793269(13)</td></tr>
<tr><td>26.5625(14)</td><td>156.25, 210.813492, 312.5, 390.625, 491.898148</td></tr>
</table>
Note: pll_ref_clk[1] is only available when you enable Preserve unused transceiver channels for PAM4 parameter in E-tile PAM4 mode IP variations.
</td></tr>
<tr><td>tx_usr_clk</td><td>ILK only</td><td>1</td><td>Input</td><td>Transmit side user data interface clock. The lower frequency of tx_usr_clk increases the latency of data path.</td></tr>
<tr><td>rx_usr_clk</td><td>ILK only</td><td>1</td><td>Input</td><td>Receive side user data interface clock. The lower frequency of rx_usr_clk increases the latency of data path.</td></tr>
<tr><td>clk_tx_common</td><td>ILK and ILA</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.
The valid frequencies per lane data rate are as below:</td></tr>
</table>

continued...

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

(12) Only available in H-tile and L-tile device variations.

(13) Only available in NRZ E-tile device variations.

(14) Only available in PAM4 E-tile device variations.

<table>
<tr><th>Signal Name</th><th>Feature Support</th><th>Width (Bits)</th><th>I/O Direction</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><th rowspan="2">Data Rate (Gbps)</th><th colspan="2">E-tile</th><th colspan="2">L- and H-tile</th></tr>
<tr><th>PMA_WIDTH</th><th>clk_tx_common</th><th>PMA_WIDTH</th><th>clk_tx_common</th></tr>
<tr><td>6.25</td><td>64</td><td>97.656250</td><td>32</td><td>195.3125</td></tr>
<tr><td rowspan="2">10.3125</td><td rowspan="2">64</td><td rowspan="2">161.132813</td><td>32 (For number of segments = 1)</td><td>322.265625</td></tr>
<tr><td>64 (For number of segments = 2)</td><td>161.1328125</td></tr>
<tr><td rowspan="2">12.5</td><td rowspan="2">64</td><td rowspan="2">195.3125</td><td>64</td><td>195.3125</td></tr>
<tr><td>40 (Only for number of segments =1 and 12x12.5 Gbps combination)</td><td>312.5</td></tr>
<tr><td>25.28</td><td>64</td><td>395.0</td><td>64</td><td>395.0</td></tr>
<tr><td>25.78</td><td>64</td><td>402.8125</td><td>64</td><td>402.8125</td></tr>
<tr><td>25.78125</td><td>64</td><td>402.8320</td><td>64</td><td>402.8320</td></tr>
<tr><td>26.5625</td><td>64</td><td>415.039063</td><td>N/A</td><td>N/A</td></tr>
</table>
</td></tr>
<tr><td>clk_rx_common</td><td>ILK and ILA</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. The valid frequencies for clk_rx_common are same as clk_tx_common.</td></tr>
<tr><td>reset_n</td><td>ILK and ILA</td><td>1</td><td>Input</td><td>Active-low asynchronous reset signal.</td></tr>
<tr><td>tx_usr_srst</td><td>ILK and ILA</td><td>1</td><td>Output</td><td>Transmit-side reset output signal. Indicates the transmit side user data interface is resetting. This signal is synchronous with tx_usr_clk.</td></tr>
<tr><td>rx_usr_srst</td><td>ILK and ILA</td><td>1</td><td>Output</td><td>Receive-side reset output. Indicates the receive side user data interface is resetting. This signal is synchronous with rx_usr_clk.</td></tr>
<tr><td>mac_clkin</td><td>ILK and ILA</td><td>1</td><td>Input</td><td>This clock signal is only available in IP core variations that target an E-tile PAM4 device variations. This signal must be driven by a PLL and must use the same clock source that drives the pll_ref_clk.</td></tr>
</table>

Related Information

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

5.2. Transmit User Interface Signals

Table 27. Transmit User Interface Signals

The input signals are synchronous to `clk_tx_commom` and output signals are synchronous to `clk_rx_common`.

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
`itx_chan`	ILK	8	Input	Transmit logic channel number for the first segment chunk. 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.
	ILA	[Number of lanes-1:0]		Indicates one of two channel numbers associated with the data burst following control symbol. Valid when `itx_sop` is equal to 1 for corresponding symbol.
`itx_chan1`	ILK only	8	Input	Transmit logic channel number for the second segment chunk. The IP core supports up to 256 channels. DUAL or QUAD segment interface defines this signal. 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`	ILK only	Variable	Input	Indicates the number of valid 64-bit words in the current packet in the current data symbol. This signal is not available if you turn on **Enable Interlaken Look-aside mode** in the Interlaken parameter editor. The width of the `itx_num_valid` depends on the parameter number of words and **Number of segments** For single segment, • If number of words=4, then width=3 using `itx_num_valid[2:0]` • If number of words=8, then width=4 using `itx_num_valid[7:4]` • If number of words=16, then width=5 using `itx_num_valid[9:5]` For multi-segment, • If number of words=8, then width=8 using `itx_num_valid[7:4]` (first segment chunk) and `itx_num_valid[3:0]` (second segment chunk) • If number of words=16, then width=10 using `itx_num_valid[9:5]` (first segment chunk) and `itx_num_valid[4:0]` (second segment chunk) If number of words is equal to 8 and **Number of segments** is equal to 2: • `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 can vary from 4'b0000 to 4'b0100. In non-valid cycle, the value of `itx_num_valid[7:4]` and `itx_num_valid[3:0]` must be set to 4'b0000. In the end of burst cycle (`itx_eob=1`), if the value of `itx_num_valid[7:4]` is equal to or less than 4'0100 and the parameter **Number of Segments** is set to 2 , then the value of `itx_num_valid[3:0]` can be set to values from 4'b0000 to 4'b0100. See the table below for valid values. If the value of `itx_num_valid[3:0]` is set to non-zero, you must set the value of `itx_sob[0]` to 1'b1.

continued...

<table>
<tr><th>Signal Name</th><th>Feature Support</th><th>Width (Bits)</th><th>I/O Direction</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><th>itx_num_valid[7:4]</th><th>itx_num_valid[3:0]</th></tr>
<tr><td>1, 2, 3, 4</td><td>0, 1, 2, 3, 4</td></tr>
<tr><td>5, 6, 7, 8</td><td>0</td></tr>
</table>
If number of words is equal to 8 and Number of segments is equal to 4:

<code>itx_num_valid[7:4]</code> indicates the number of valid words in <code>itx_din_words[511:0]</code>. The value can vary from 4’b0000 to 4’b1000.
<code>itx_num_valid[3:0]</code> indicates the number of valid words in <code>itx_din_words[383:0]</code>. The value can vary from 4’b0000 to 4’b0110.

In non-valid cycle, the value of <code>itx_num_valid[7:4]</code> and <code>itx_num_valid[3:0]</code> must be set to 4’b0000.
In the end of burst cycle (<code>itx_eob=1</code>), if the value of <code>itx_num_valid[7:4]</code> is equal to or less than 4'0110 and the parameter Number of segments is set to 4 , then the value of <code>itx_num_valid[3:0]</code> can be set to 4'b0000 to 4'b0110. See the table below for valid values. If the value of <code>itx_num_valid[3:0]</code> is set to non-zero, you must set the value of <code>itx_sob[0]</code> to 1'b1.
<table>
<tr><th>itx_num_valid[7:4]</th><th>itx_num_valid[3:0]</th></tr>
<tr><td>1, 2</td><td>0, 1, 2, 3, 4, 5, 6</td></tr>
<tr><td>3, 4</td><td>0, 1, 2, 3, 4</td></tr>
<tr><td>5, 6</td><td>0, 1, 2</td></tr>
<tr><td>7, 8</td><td>0</td></tr>
</table>
If number of words is equal to 16 and Number of segments is equal to 2:
<table>
<tr><th>itx_num_valid[9:5]</th><th>itx_num_valid[4:0]</th></tr>
<tr><td>1, 2, 3, 4, 5, 6, 7, 8</td><td>0, 1, 2, 3, 4, 5, 6, 7, 8</td></tr>
<tr><td>9, 10, 11, 12, 13, 14, 15, 16</td><td>0</td></tr>
</table>
If number of Words is equal to 16 and Number of segments is equal to 4:
<table>
<tr><th>itx_num_valid[9:5]</th><th>itx_num_valid[4:0]</th></tr>
<tr><td>1, 2, 3, 4</td><td>0 to 12</td></tr>
<tr><td>5, 6, 7, 8</td><td>0 to 8</td></tr>
<tr><td>9, 10, 11, 12</td><td>0 to 4</td></tr>
<tr><td>13, 14, 15, 16</td><td>0</td></tr>
</table>
</td></tr>
<tr><td><code>itx_eob</code></td><td>ILK only</td><td>1</td><td>Input</td><td>Indicates the current data symbol contains the end of the burst (EOB) for the first segment chunk.
This signal is not available if you turn on Enable Interlaken Look-aside mode in the Interlaken parameter editor.
This signal is used in SINGLE, DUAL or QUAD segment mode.</td></tr>
</table>

continued...

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
				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.
itx_eob1	ILK only	1	Input	Indicates the current data symbol contains the end of the burst (EOB) for the second segment chunk. This signal is used in DUAL or QUAD segment mode. 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.
itx_eopbits	ILK	4	Input	Number of bytes at the end of packet for the first segment chunk. 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.
	ILA	[Number of lanes*4-1:0]		Specifies the number of valid bytes of the corresponding data symbol and indicates the end of packet transfer (EOP). 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.
itx_eopbits1	ILK only	4	Input	Number of bytes at the end of packet for the second segment chunk. 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_eopbits1 as following:

continued...

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
				• 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`	ILK only	1, 2 or 4	Input	Indicates the current data symbol contains the start of a burst (SOB). This signal is not available if you turn on **Enable Interlaken Look-aside mode** in the Interlaken parameter editor. 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. • [1]— single segment • [1:0]—dual segment • [3:0]—four segment {segment 3, segment 2, segment 1, segment 0} defines the segment order with segment 3 starts at the most significant bit location. Using four segment as example, the signal has the following valid values: • [3]: indicates SOB for the first segment chunk. • [2:0]: only one bit can be set to indicate SOB for the second segment chunk. For example: If number of words= 16, **Number of segments**=4, `tx_num_valid[9:5]` =3, `tx_num_valid[4:0]`= 9, then the second segment starts at `word[11]`, `sob[3:0]`= 4'1100 If number of words= 16, **Number of segments**=4, `tx_num_valid[9:5]` =9, `tx_num_valid[4:0]`= 4, then the second segment starts at `word[3]`, `sob[0]`= 4'1001
`itx_sop`	ILK	1, 2 or 4	Input	Indicates the current data symbol on `itx_din_words` contains the start of a packet (SOP). • [1]— single segment • [1:0]—dual segment • [3:0]—four segment {segment 3, segment 2, segment 1, segment 0} defines the segment order with segment 3 starts at the most significant bit location. Using four segment as example, the signal has the following valid values: • [3]: indicates SOP for the first segment chunk. • [2:0]: only one bit can be set to indicate SOP for the second segment chunk. For example: If number of words, **Number of segments**=4, `tx_num_valid[9:5]` =3, `tx_num_valid[4:0]`= 9, then the second segment starts at `word[11]`, `sop[3:0]`= 4'1100 If number of words= 16, **Number of segments**=4, `tx_num_valid[9:5]` =9, `tx_num_valid[4:0]`= 4, then the second segment starts at `word[3]`, `sop[0]`= 4'1001

continued...

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
	ILA	[Number of lanes-1:0]		Each bit indicates the SOP for the data burst following corresponding control symbol.
itx_din_words	ILK	Variable	Input	The 64-bit words of input data (one data symbol). The width of the itx_dout_words depends on the parameter number of words. • If number of words=4, then width=256 bits. • If number of words=8, then width=512 bits. • If number of words=16, then width=1024 bits. The first and last data word is in [511:448] and [63:0] respectively.
	ILA	[Number of lanes*64-1:0]		The 64-bits words of input data (one data symbol). When itx_idle is equal to one, the ILA IP core ignores matching data word in itx_din_words. The width of the itx_din_words depends on parameter number of lanes: • If number of lanes=4, then width=256 bits. • If number of lanes=6, then width=284 bits. • If number of lanes=8, then width=512 bits. • If number of lanes=10, then width=640 bits. • If number of lanes=12, then width=768 bits. Example: For 8 lanes, the first data word is in [511:448] and last data word is in [63:0].
itx_calendar	ILK only	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	ILK	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.
	ILA			The signal indicates IP readiness to accept user data. When this signal is high, you can send traffic to the IP core. When this signal is low, it indicates tx_lanes_aligned and/or rx_lanes_aligned deasserted.
itx_ch0_xon	ILA only	1	Output	Indicates channel 0 flow control. This signal is available only if you turn on **Enable Interlaken Look-aside mode** in the Interlaken parameter editor.
itx_ch1_xon	ILA only	1	Output	Indicates channel 1 flow control. This signal is available only if you turn on **Enable Interlaken Look-aside mode** in the Interlaken parameter editor.
itx_valid	ILA only	1	Input	Valid signal for entire input bus. The pattern of itx_valid should match that of itx_credit. The latency between itx_credit anditx_valid should be fixed and specified by the parameter TX_CREDIT_LATENCY. This signal is available only if you turn on **Enable Interlaken Look-aside mode** in the Interlaken parameter editor.
itx_idle	ILA only	[Number of lanes-1:0]	Input	Each bit indicates unused 64-bit words in the current data symbol and it is also not part of a burst. The IP inserts IDLE control word in this location. The IDLE equals to one may imply

continued...

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
				the end of a previous burst. It is not necessary to have IDLE between bursts. It is legal to immediately start a new burst after last data word of previous burst. This signal is available only if you turn on **Enable Interlaken Look-aside mode** in the Interlaken parameter editor.
itx_ctrl	ILA only	[Number of lanes*29-1:0]	Input	Eight 29 bits itx_ctrl signal indicates the application specific 0 (15 bits), specific 1 (6 bits) and specific 2 (8 bits) data for ILA. ILA IP core samples this value only when associated itx_sop is valid. The most significant 29 control bits data is aligned with the most significant bit of itx_sop. {specific 0, specific 1, specific 2} = {[28:14],[13:8],[7:0]} This signal is available only if you turn on **Enable Interlaken Look-aside mode** in the Interlaken parameter editor.
itx_credit	ILA only	1	Output	Flow control signal to give backpressure to the transmit traffic. User logic asserts itx_valid after TX_CREDIT_LATENCY cycles regardless of itx_ready state and/or existence of user data. This signal is available only if you turn on **Enable Interlaken Look-aside mode** in the Interlaken parameter editor.

5.3. Receive User Interface Signals

Table 28. Receive User Interface Signals

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
irx_chan	ILK	8	Output	Receive logic channel number for the first segment chunk. 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.
	ILA	[Number of lanes-1]		
irx_chan1	ILK only	8	Output	Receive logic channel number for the second segment chunk. The IP core supports up to 256 channels. The Interlaken IP core samples this value only when a bit of irx_sob[0] is high and irx_num_valid of the second segment chunk has a non-zero value.
irx_num_valid	ILK only	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 number of words. For single segment, • If number of words=4, then width=3 using irx_num_valid[2:0] • If number of words=8, then width=8 using irx_num_valid[7:4] • If number of words=16, then width=5 using irx_num_valid[9:5] For multi-segment,

continued...

<table>
<tr><th>Signal Name</th><th>Feature Support</th><th>Width (Bits)</th><th>I/O Direction</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>

If number of words=8, then width=8 using <code>irx_num_valid[7:4]</code> (first segment chunk) and <code>irx_num_valid[3:0]</code> (second segment chunk)
If number of words=16, then width=10 using <code>irx_num_valid[9:5]</code> (first segment chunk) and <code>irx_num_valid[4:0]</code> (second segment chunk)

If number of words is equal to 8 and Number of segments is equal to 2:

<code>irx_num_valid[7:4]</code> indicates the number of valid words in <code>irx_din_words[511:0]</code>. The value can vary from 4’b0000 to 4’b1000.
<code>irx_num_valid[3:0]</code> indicates the number of valid words in <code>irx_din_words[255:0]</code>. The values can vary from 4’b0000 to 4’b0100.

For all non-valid cycles, IP core sets the value of <code>irx_num_valid[7:4]</code> and <code>irx_num_valid[3:0]</code>to 4’b0000.
In the end of burst cycle (<code>irx_eob=1</code>), if the value of <code>irx_num_valid[7:4]</code> is equal to or less than 4'b0100 and the Number of segments parameter is set to 2, you can set the value of <code>irx_num_valid[3:0]</code> from 4'b0000 to 4'b0100. If the value of <code>irx_num_valid[3:0]</code> is not equal to zero, you must set the value of <code>irx_sop[0]</code> or <code>irx_sob[0]</code> to 1'b1.
<table>
<tr><th><code>irx_num_valid[7:4]</code></th><th><code>irx_num_valid[3:0]</code></th></tr>
<tr><td>1, 2, 3, 4</td><td>0, 1, 2, 3, 4</td></tr>
<tr><td>5, 6, 7, 8</td><td>0</td></tr>
</table>
If number of words is equal to 8 and Number of segments is equal to 4:

<code>irx_num_valid[7:4]</code> indicates the number of valid words in <code>irx_din_words[511:0]</code>. The value can vary from 4’b0000 to 4’b1000.
<code>irx_num_valid[3:0]</code> indicates the number of valid words in <code>irx_din_words[383:0]</code>. The value can vary from 4'b0000 to 4'b0110.

For all non-valid cycles, the value of <code>irx_num_valid[7:4]</code> and <code>irx_num_valid[3:0]</code> must be set to 4’b0000.
In the end of burst cycle (<code>irx_eob=1</code>), if the value of <code>irx_num_valid[7:4]</code> is equal or less than 4'b0110 and the Number of segments parameter is set to 4, you can set the value of <code>irx_num_valid[3:0]</code> from 4'b0000 to 4'b0110. If the value of <code>irx_num_valid[3:0]</code> is not equal to zero, you must set the value of <code>irx_sop[0]</code> or <code>irx_sob[0]</code> to 1'b1.
</td></tr>
</table>

continued...

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
				`irx_num_valid[7:4]` / `irx_num_valid[3:0]` 1, 2, / 0, 1, 2, 3, 4, 5, 6 3, 4 / 0, 1, 2, 3, 4 5, 6 / 0, 1, 2 7, 8 / 0 If number of words is equal to 16 and **Number of segments** is equal to 2: `irx_num_valid[7:4]` / `irx_num_valid[3:0]` 1, 2, 3, 4, 5, 6, 7, 8 / 0, 1, 2, 3, 4, 5, 6, 7, 8 9, 10, 11, 12, 13, 14, 15, 16 / 0 If number of words is equal to 16 and **Number of segments** is equal to 4: `irx_num_valid[7:4]` / `irx_num_valid[3:0]` 1, 2, 3,4 / 0 to 12 5, 6, 7, 8 / 0 to 8 9, 10, 11, 12 / 0 to 4 13, 14, 15, 16 / 0
`irx_eob`	ILK only	1	Output	Indicates the end of the burst (EOB) for the first segment chunk. This signal toggles in Packet Mode and in Interleaved Mode. This signal is not available if you turn on **Enable Interlaken Look-aside mode** in the Interlaken parameter editor.
`irx_eob1`	ILK only	1	Output	Indicates the end of the burst (EOB) for the second segment chunk. Only DUAL or QUAD segment interface defines this signal.
`irx_eopbits`	ILK	4	Output	Specifies the number of bytes at the end of packet (EOP) for the first segment chunk. Indicates whether the current data symbol contains the 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:

continued...

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
				• 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.
	ILA	[Number of lanes*4 -1:0]		Specifies the number of valid bytes of the corresponding data symbol AND Indicates the end of packet transfer (EOP). IP core sets the value of `itx_eopbits` as follows: • 4b'0000: no End-of-Packet • 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 in the burst. Bits [2:0] are encoded 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.
`irx_eopbits1`	ILK only	4	Output	Specifies number of bytes at the end of packet (EOP) for the second segment chunk. You must set the value of `irx_eopbits1` 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`	ILK only	1, 2 or 4	Output	Indicates the start of a burst (SOB). This signal toggles in Packet Mode and in Interleaved Mode. If the IP core is in Interleaved mode, you are responsible for providing the start of the burst signal. If the IP core is in Packet mode, the IP core ignores this signal. The IP core samples the `irx_chan` signal during this cycle.

continued...

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
				• [1]— single segment • [1:0]—dual segment • [3:0]—four segment {segment 3, segment 2, segment 1, segment 0} defines the segment order with segment 3 starts at the most significant bit location (left aligned). Using four segment as example, the signal has the following valid values: • [3]: indicates SOB for the first segment. • [2:0]: only one bit can be set to indicate SOB for the second segment. For example: If number of words= 16, **Number of segments**=4, `rx_num_valid[9:5]`= 3, `rx_num_valid[4:0]`= 9, then the second segment starts at `word[11]`, `sob[3:0]`= 4'1100 If number of words= 16, **Number of segments**=4, `rx_num_valid[9:5]`= 9, `rx_num_valid[4:0]`= 4, then the second segment starts at `word[3]`, `sob[3:0]`= 4'1001
`irx_sop`	ILK	1, 2 or 4	Output	Indicates the current data symbol on `irx_din_words` contains the start of a packet (SOP). • [1]— single segment • [1:0]—dual segment • [3:0]—four segment {segment 3, segment 2, segment 1, segment 0} defines the segment order with segment 3 starts at the most significant bit location (left aligned). Using four segment as example, the signal has the following valid values: • [3]: indicates SOP for the first segment. • [2:0]: only one bit can be set to indicate SOP for the second segment. For example: If number of words= 16, **Number of segments**=4, `rx_num_valid[9:5]`= 3, `rx_num_valid[4:0]`= 9, then the second segment starts at `word[11]`, `sop[3:0]`= 4'1100 If number of words= 16, **Number of segments**=4, `rx_num_valid[9:5]`= 9, `rx_num_valid[4:0]`= 4, then the second segment starts at `word[3]`, `sop[0]`= 4'1001
	ILA	[Number of lanes-1:0]		Each bit indicates the start of a packet (SOP) for the data burst following corresponding control symbol.
`irx_dout_words`	ILK	Variable	Output	The 64-bit words of input data (one data symbol). The width of the `itx_din_words` depends on the parameter `external_words`.

continued...

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
				• If number of words=4, then width=256 bits. • If number of words=8, then width=512 bits. • If number of words=16, then width=1024 bits. 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`.
	ILA	[Number of lanes*64-1:0]		The 64-bit words of input data (one data symbol). The width of the `itx_din_words` depends on the parameter number of lanes. • If number of lanes=4, then width=256 bits. • If number of lanes=6, then width=284 bits. • If number of lanes=8, then width=512 bits. • If number of lanes=10, then width=640 bits. • If number of lanes=12, then width=768 bits. For 8 lanes,the first data word is in [511:448] and last data word is in [63:0].
`irx_calendar`	ILK only	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`	ILK only	1	Output	Indicates an errored packet. This signal is valid only when `irx_eob` is asserted.
`irx_err1`	ILK only	1	Output	Indicates an errored packet in second segment chunk. This signal is valid only when `irx_eob1` is asserted. This signal is only valid in DUAL or QUAD mode.
`irx_ch0_xon`	ILA only	1	Output	Indicates channel 0 flow control.
`irx_ch1_xon`	ILA only	1	Output	Indicates channel 1 flow control.
`irx_valid`	ILA only	1	Output	Valid signal for entire output bus
`irx_idle`	ILA only	[Number of lanes-1:0]	Output	Each bit indicates unused 64-bit words in the current data symbol and it is also not part of a burst. User logic can ignore the `irx_dout_words`, `irx_sop`, `irx_eopbits`, and `irx_chan` when `irx_idle` is equal to one. The `irx_idle` is equal to one may imply the end of a previous burst.
`irx_ctrl`	ILA only	[Number of lanes*29-1:0]	Output	Eight 29 bits signal indicates the application specific 0 (15 bits), specific 1 (6 bits) and specific 2 (8 bits) data for Interlaken Look-aside mode. This signal can be valid only when `irx_sop` is asserted. The most significant 29 control bits data is aligned with the most significant bit of `irx_sop`. {specific 0, specific 1, specific 2} = {[28:14],[13:8],[7:0]}

5.4. Management Interface Signals

The management interface signals are available for the Avalon-MM (AVMM) interface.

Table 29. Management Interface Signals

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
mm_clk	ILK and ILA	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

The reconfiguration interface signals are available for the AVMM interface.

Table 30. Reconfiguration Interface Signals

<table>
<tr><th>Signal Name</th><th>Feature Support</th><th>Width (Bits)</th><th>I/O Direction</th><th>Description</th></tr>
<tr><td>reconfig_clk</td><td rowspan="10">ILK and ILA</td><td>1</td><td>Input</td><td>Intel Stratix 10 transceiver reconfiguration interface clock.</td></tr>
<tr><td>reconfig_reset</td><td>1</td><td>Input</td><td>Active-high synchronous reset. Assert this signal to reset the Intel Stratix 10 transceiver reconfiguration interface.</td></tr>
<tr><td>reconfig_read</td><td>1</td><td>Input</td><td>Read access to the Intel Stratix 10 hard PCS registers.</td></tr>
<tr><td>reconfig_write</td><td>1</td><td>Input</td><td>Write access to the Intel Stratix 10 hard PCS registers.</td></tr>
<tr><td>reconfig_address</td><td>• RECONF_ADDR +11 (For L-Tile and H-Tile device variations)
• RECONF_ADDR +20 (For E-tile device variations)
Refer to the table below to find out the value of RECONF_ADDR:
<table><tr><th>Lanes</th><th>RECONF_ADDR (Bits)</th></tr><tr><td>4</td><td>2</td></tr><tr><td>6</td><td>3</td></tr><tr><td>8</td><td>3</td></tr><tr><td>10</td><td>4</td></tr><tr><td>12</td><td>4</td></tr></table></td><td>Input</td><td>Address to access the hard PCS registers. This signal holds both the hard PCS register offset and the transceiver channel being addressed.
The E-tile PAM4 mode IP variations customize the most significant bit (MSB) of the reconfig_address as follows:
• There are three Native PHY available in an E-tile PAM4 mode Interlaken design. The reconfig_address[22:21] bits select which native PHY to access as below:
— 2'b00- 1st Naive PHY
— 2'b01- 2nd Native PHY
— 2'b10- 3rd Native PHY
— 2'b11- Reserved
• The reconfig_address[23] bit selects RS-FEC registers when set to 1 and PMA registers when set to 0.
• The active bits within the reconfig_address[20:0] differs when you access RS-FEC registers as compared to PMA registers. Refer to the PMA Register Map section of the E-tile Transceiver PHY User Guide for more information on this.
In E-tile NRZ mode IP variations, the MSB of reconfig_address (e.g., bit [23] for 12 lanes NRZ) is always unused while the remaining bits are used to access PMA registers.</td></tr>
<tr><td>reconfig_writedata</td><td>32</td><td>Input</td><td>When reconfig_write is high, reconfig_writedata holds valid write data.</td></tr>
<tr><td>reconfig_readdata</td><td>32</td><td>Output</td><td>After user logic asserts the reconfig_read signal, when the IP core deasserts the signal, reconfig_readdata holds valid read data.</td></tr>
<tr><td>reconfig_waitrequest</td><td>1</td><td>Output</td><td>Busy signal for reconfig_readdata.</td></tr>
</table>

For information on Intel Stratix 10 L- and H-tile Transceiver PHY registers, refer to the *Logical View of the L-tile/H-tile Transceiver Registers* section.

For information on PMA and RS-FEC registers of the E-tile transceiver PHY, refer to the *PMA Register Map* section.

Related Information

- Avalon Interface Specifications
- Logical View of the L-Tile/H-Tile Transceiver Registers
- PMA Register Map
- E-tile Transceiver PHY User Guide

5.6. Interlaken Link and Miscellaneous Signals

Table 31. SERDES Pins

In PMA4 mode, the Interlaken IP uses only six even channels out of twelve E-tile channels. The other six odd channels cannot be used. For more information on transceiver channels, refer to the E-tile Transceiver PHY User Guide.

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
rx_pin	ILK and ILA	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.
rx_pin_n		Number of lanes	Input	For the PAM4 loopback example design, rx_pin_n receives data from tx_pin_n.
tx_pin_n		Number of lanes	Output	For the PAM4 loopback example design, tx_pin_n drives data to rx_pin_n.

Table 32. Real-Time Transmitter Status Signals

Signal Name [15]	Feature Support	Width (Bits)	I/O Direction	Description
tx_lanes_aligned	ILK and ILA	1	Output	Indicates whether all of the transmitter lanes are aligned and are ready to send traffic.
itx_overflow	ILK only	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	ILK only	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.

[15] Synchronous with tx_usr_clk.

Table 33. Real-Time Receiver Status Signals

Signal Name [16]	Feature Support	Width (Bits)	I/O Direction	Description
rx_lanes_aligned	ILK and ILA	1	Output	Indicates whether all of the receiver lanes are aligned and are ready to receive traffic.
sync_locked	ILK and ILA	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	ILK and ILA	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	ILK and ILA	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 in E-tile IP core variations and as a multi-cycle wide pulse in L-tile/H-tile IP core variations.
crc32_err	ILK and ILA	Number of lanes	Output	An error flag indicating diagnostic CRC32 failures per lane. This signal is asserted as a single cycle wide pulse in E-tile IP core variations and as a multi-cycle wide pulse in L-tile/H-tile IP core variations.
irx_overflow	ILK only	0	Output	This signal is tied to 0 and it is not used.
rdc_overflow	ILK only	0	Output	This signal is tied to 0 and it is not used.
rg_overflow	ILK only	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	ILK only	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.
sop_cntr_inc	ILK only	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	ILK only	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.
sop_cntr_inc1	ILK only		Output	A pulse indicating that the IP core receiver user data interface received a start-of- packet (SOP) on second segment chunk. You can use this signal to increment a count of SOPs the application observes on the receive interface. This signal is only available in muti-segment mode of IP core variations.

continued...

[16] Synchronous with rx_usr_clk.

Signal Name [16]	Feature Support	Width (Bits)	I/O Direction	Description
eop_cntr_inc1	ILK only		Output	A pulse indicating that the IP core receiver user data interface received an end-of-packet (EOP) on second segment chunk. You can use this signal to increment a count of EOPs the application observes on the receive interface. This signal is only available in muti-segment mode of IP core variations.
nad_cntr_inc	ILK only	0	Output	This signal is tied to 0 and it is not used.
rx_xcoder_uncor_fec cw	ILK and ILA	[XCODER_ LANES-1:0]]	Output	Indicates uncorrectable FEC code word. This signal is also accessible through status register and may not align with the irx_data signal. This signal is only available in PAM4 mode of E-tile device variations.

Table 34. Burst Control Settings

These signals are not available in Interlaken Look-aside mode.

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[17]
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[18] • 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.

[16] Synchronous with rx_usr_clk.

[17] This value is not supported for number of words=4.

[18] This value is not supported for number of words= 8 and number of words= 16 .

Table 35. ECC Status Signals

These signals are not available in Interlaken Look-aside mode.

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

Related Information

E-Tile Transceiver PHY User Guide

5.7. External PLL Interface Signals

The external PLL interface signals are only available in the L-Tile and H-Tile device variations. For the E-tile device variations, tie the input signals low. The output signals can float.

Table 36. ATX PLL Interface Signals

Signal Name	Feature Support	Width (Bits)	I/O Direction	Description
tx_pll_locked	ILK and ILA	1	Input	PLL-locked indication from external TX PLL. This signal is only available in Intel Stratix 10 H-tile device variations.
tx_pll_cal_busy		1	Input	PLL-busy indication from external TX PLL. This signal is only available in Intel Stratix 10 H-tile device variations.
tx_serial_clk		NUM_LANES	Input	High-speed clock for transceiver channel, provided from external TX PLL This signal is only available in Intel Stratix 10 H-tile device variations.
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. This signal is only available in Intel Stratix 10 H-tile device variations.
mac_pll_locked		1	Input	Lock indicator for the PLL that generates mac_clkin. This signal is only available in PAM4 mode of E-Tile device variations.

Related Information

Intel Stratix 10 L- and H-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 37. IP Core Register Map

Offset	Name	R/W	Description
16'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.
16'h1	LANE_COUNT	RO	Number of lanes
16'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.
16'h4	TX_EMPTY	RO	[NUM_LANES–1:0] – Transmit FIFO status (empty)
16'h5	TX_FULL	RO	[NUM_LANES–1:0] – Transmit FIFO status (full)
16'h6	TX_PEMPTY	RO	[NUM_LANES–1:0] – Transmit FIFO status (partially empty)
16'h7	TX_PFULL	RO	[NUM_LANES–1:0] – Transmit FIFO status (partially full)
16'h8	RX_EMPTY	RO	[NUM_LANES–1:0] – Receive FIFO status (empty)
16'h9	RX_FULL	RO	[NUM_LANES–1:0] – Receive FIFO status (full)
16'hA	RX_PEMPTY	RO	[NUM_LANES–1:0] – Receive FIFO status (partially empty)
16'hB	RX_PFULL	RO	[NUM_LANES–1:0] – Receive FIFO status (partially full)
16'hC	MAC_CLK_KHZ	RO	MAC clock frequency (kHz). This register is only available in E-tile PAM4 mode variations.
16'hD	RX_KHZ	RO	RX recovered clock frequency (kHz) *Note:* This register assumes mm_clk frequency of 100 MHz, and scales accordingly if the mm_clk is not equal to 100 MHz.
16'hE	TX_KHZ	RO	TX serial clock frequency (kHz)
16'h10	PLL_LOCKED	RO	In L-tile and H-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 E-tile device variations: • Bit[16] – MAC clock PLL lock indication. • Bit[0] – Transmit PLL lock indication. One lock bit for all transceivers.

continued...

Offset	Name	R/W	Description
16'h11	FREQ_LOCKED	RO	[NUM_LANES–1:0] – Clock data recovery is frequency locked on the inbound data stream
16'h12	LOOPBACK	RW	In Intel Stratix 10 L-tile and H-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 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. For more information refer to the *Register Map* section of the *E-Tile Transceiver PHY User Guide*.
16'h13	RESET	RW	Bit 9 : 1 = Force lock to data mode (Only in Intel Stratix 10 L- and H-tile device variations) Bit 8 : 1 =Force lock to reference mode (Only in Intel Stratix 10 L- and H- 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.
16'h20	ALIGN	RO	Bit 12 : RSFEC AM sync align (Only available in Intel Stratix 10 E-tile PAM4 mode device variations, not valid in NRZ mode) Bit 0 : TX lanes are aligned Bit 1 : RX lanes are aligned.
16'h21	WORD_LOCK	RO	[NUM_LANES–1:0] – Word (block) boundaries have been identified in the RX stream.
16'h22	SYNC_LOCK	RO	[NUM_LANES–1:0] – Metaframe synchronization has been achieved.
16'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.
16'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.
16'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.
16'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.
16'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.
16'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.

continued...

Offset	Name	R/W	Description
16'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.
16'h80	ILKN_FEC_XCODER_TX_ILLEGAL_STATE	RO	This register is only available in E-tile PAM4 mode variations. Transcoder detects illegal framing bits [66:64] of the Interlaken frame layer words. This is sticky bit.
16'h81	ILKN_FEC_XCODER_RX_UNCOR_FECCW	RO	This register is only available in E-tile PAM4 mode variations. FEC indicates uncorrectable FEC code word error. This register saturates at 4'b1111 for each lane.

Related Information

- Avalon Interface Specifications
- Logical View of the L-Tile/H-Tile Transceiver Registers
- PMA Register Map

7. Test Features

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

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 in L- and H-tile device variations:

- 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 1) 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.

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.

To turn on internal serial loopback in E-tile device variations:

- Reconfigure the PMA settings by using PMA attribute code 0x0008 of the transceiver PHY reconfiguration interface.
  1. Write 0x84[7:0] = 0x01
  2. Write 0x85[7:0] =0x01
  3. Write 0x86[7:0] = 0x08
  4. Write 0x87[7:0] = 0x00
  5. Write 0x90[0] = 1'b1
  6. Read 0x8A[7]. It should be 1
  7. Read 0x8B[0] until it changes to 0
  8. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value
- Perform the initial RX equalizer adaption calibration steps. Refer to the *PMA Receiver Equalization Adaption Usage Model* section in the *E-Tile Transceiver PHY User Guide*.
- Reset the IP core by asserting and then deasserting the active low `reset_n` signal.
- Monitor the RX lanes aligned bit (bit 1) 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.

To turn off internal serial loopback:

- 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.
- Reconfigure the PMA settings to turn off the serial loop back mode by using PMA attribute code 0x0008 of the transceiver PHY reconfiguration interface..
  1. Write 0x84[7:0] = 0x00
  2. Write 0x85[7:0] =0x00
  3. Write 0x86[7:0] = 0x08
  4. Write 0x87[7:0] = 0x00
  5. Write 0x90[0] = 1'b1
  6. Read 0x8A[7]. It should be 1
  7. Read 0x8B[0] until it changes to 0
  8. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value

Related Information

- IP Core Reset on page 67
- PAM Receiver Equalization Adaption Usage Model

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 on the FPGA board. This mode does not require any special programming of the IP core.

8. Interlaken (2nd Generation) Intel FPGA IP User Guide Archives

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

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

Intel Quartus Prime Pro Edition Version	IP Core Version	User Guide
20.4	20.0.1	Interlaken (2nd Generation) FPGA IP User Guide
20.3	20.0.0	Interlaken (2nd Generation) FPGA IP User Guide
20.2	19.3.0	Interlaken (2nd Generation) FPGA IP User Guide
19.3	19.2.1	Interlaken (2nd Generation) FPGA IP User Guide
19.2	19.2	Interlaken (2nd Generation) FPGA IP User Guide
18.1.1	18.1.1	Interlaken (2nd Generation) Intel Stratix 10 FPGA IP User Guide
18.1	18.1	Interlaken (2nd Generation) Intel Stratix 10 FPGA IP User Guide
18.0.1	18.0.1	Interlaken (2nd Generation) FPGA IP User Guide
18.0	18.0	Interlaken (2nd Generation) Intel FPGA IP User Guide
17.1	17.1	Interlaken IP Core (2nd Generation) User Guide

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

Document Version	Intel Quartus Prime Version	IP Version	Changes
2021.10.04	21.3	20.0.1	• Added support for Questa simulator. • Removed support for NCSim simulator.
2021.02.24	20.4	20.0.1	• Added the following parameters: — **Preserve unused transceiver channels for PAM4** — **Reference clock frequency for preserved channels** • Added the `pll_ref_clk[1]` signal description in section: *Clock and Reset Interface Signals*. • Clarified the information about TX skew in section: *Transmit Path Blocks*.
2020.10.16	20.3	20.0.0	• Changed the data rates support from 25.3 Gbps to 25.28 Gbps and 25.8 Gbps to 25.78 Gbps. • The IP now supports new data rate 25.78125 Gbps. • Updated *Table: IP Supported Combinations of Number of Lanes and Data Rates* to clarify the supported data rate/ number of lanes combination for Interlaken look-aside IP variations. • Updated the data rate support information in the following sections: — *Features* — *Flexible Lanes Support* — *Round-trip Latency* — *Parameter Settings* • Updated *Table: IP Theoretical Raw Aggregate Bandwidth* for 25.28, 25.78, and 25.78125 Gbps data rates. • Updated *Performance and Resource Utilization* section for Intel Quartus Prime software version 20.3. • Added transceiver reference clock frequency for the 25.78125 Gbps data rate in the following sections: — *Main Parameters* — *Clock and Reset Interface Signals* • Updated section *Release Information*. • Updated *Figure: IP Parameter Editor*. • Clarified that **Number of Segments** parameter is grayed out for Interlaken Look-aside IP variations in *Table: User Data Interface*. • Updated the description of the `reconfig_address` signal for the E-tile PAM4 mode IP variations in *Table: Reconfiguration Interface Signals*.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
2020.06.22	20.2	19.3.0	• The IP now supports Interlaken Look-aside feature. • Updated the following section to include Interlaken Look-aside information: — *Figure: Typical Interlaken Application* — *Features* — *Performance and Resource Utilization* — *Figure: IP Parameter Editor* — *Figure: Interlaken (2nd Generation) Intel FPGA IP High Level System Overview* • The IP now supports 10x12.5 Gbps combination in H- and E-tile IP core variations. • Clarified the user clock frequency values for H- and E-tile device variations in *Table: Recommended User Clock Frequency*. • Added new parameter **Enable Interlaken Look-aside mode** in *chapter: Parameter Settings*. • Updated values of `pll_ref_clk` frequencies for H- and E-tile device variations in *chapter: Parameter Settings*. • Added new section *PMA Adaptation Flow*. • Added following new sections: — *High Level Data Path Flow for Interlaken Look-aside Mode* — *Interlaken Look-aside Mode* • Clarified that the design example provides logic to include out-of-band flow control functionality. • Modified *Figure: IP Core Interface Signals* to include new signals related to Interlaken Look-aside. • Added new signals related to Interlaken Look-aside in *Chapter: Interface Signals* and differentiate availability of the signals in two different modes of the IP. • Added new table *Section: Clock and Reset Interface Signals* to include frequency values for `clk_tx_common`. • Removed `itx_hungry` signal. • Removed *Table: Transceiver Interface Signals*.
2019.09.27	19.3	19.2.1	• Added public support for Intel Agilex E-tile variants. • Updated *Performance and Resource Utilization* for L-, H-, and E-tile device variations. • Added topic *Round trip Latency*. • Renamed and updated the *Figure: Interlaken (2nd Generation) Intel FPGA IP High-Level System Overview*.
2019.07.01	19.2	19.2	Made the following changes: • Added EAP support for Intel Agilex E-tile variants. • Added information about the *PMA Adaptation* parameters available for the E-tile transceiver. • Added topic on preserving performance in unused E-tile transceivers. • Updated *Performance and Resource Utilization* for E-tile transceivers. • Added definitions of `tx_pin_n` and `rx_pin_n` signals. The PAM4 loopback example design uses these signals. • Added *Interlaken Clock Domains* figure to *IP Core Clocks* topic.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
2018.12.24	18.1.1	18.1.1	• Updated section *Features*. • 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*. • Added new section *Flexible Lanes Support*. • Added new parameter **Number of Segment** in *Table: Interlaken IP Core Parameter Settings: IP Tab*. • Added new section *Multi-Segment Mode*. • Updated section *Transmit User Interface Signals* and *Receive User Interface Signals*. • Added the following new signals in *Table: Transmit User Interface Signals*: — `itx_eob1` — `itx_eopbits1` — `itx_chan1` • Added the following new signals in *Table: Receive User Interface Signals*: — `irx_eob1` — `irx_eopbits1` — `irx_chan1` — `irx_err1` — `irx_err` • Added new signal `mac_pll_locked` in *Table: ATX PLL Interface Signals*.
2018.09.24	18.1	18.1	• Renamed the document title as *Interlaken (2nd Generation) Intel Stratix 10 FPGA IP User Guide* • Added VHDL simulation model and testbench support for Interlaken (2nd Generation) IP core. • Added information about additional clock `mac_clkin` for E-tile PAM4 mode variations in *Adding the External PLL* section and updated the description about this signal in *Table: Interlaken IP Core Clocks*. • Added new Interlaken IP core block diagrams for E-tile PAM4 mode in *High Level System Flow* section. • Made the following changes in *Transmit Path Blocks* section: — Added *Figure: Interlaken IP Core Transmit Path Blocks for E-tile PAM4 Mode Device Variations* — Updated information in TX MAC, TX PCS, and TX PMA sections for E-tile device variations. • Made the following changes in *Receive Path Blocks* section: — Added *Figure: Interlaken IP Core Transmit Path Blocks for E-tile PAM4 Mode Device Variations* — Updated information in RX MAC, RX PCS, and RX PMA sections for E-tile device variations. • Added new section *Performance* to showcase how to calculate bandwidth performance. • Updated signal description in *Transmit User Interface Signals* and *Receive User Interface Signals*.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Clarified the AVMM interface signals are available for the H-, L- and E-tile device variations. • Added steps to turn on internal serial loopback mode in IP core variations that target an E-tile devices. • Added following new registers related to E-tile device variations in *Register Map* section: — TX_READY_XCVR — RX_READY_XCVR — ILKN_FEC_XCODER_TX_ILLEGAL_STATE — ILKN_FEC_XCODER_RX_ILLEGAL_STATE
2018.07.16	18.0.1	18.0.1	• Added support for the 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 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 E-tile devices.
2018.05.07	18.0	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*.
Added the resource utilization numbers for 25.3May 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.