R-Tile Avalon® Streaming Intel® FPGA IP for PCI Express* User Guide

Updated for Intel® Quartus® Prime Design Suite: **23.2**

IP Version: **10.0.0**

Online Version

Send Feedback

683501

UG-20316

2023.06.26

Contents

1. About the R-Tile Avalon® Streaming Intel® FPGA IP for PCI Express

1.1. Overview

R-Tile is an FPGA companion tile that supports PCI Express* configurations up to Gen5 x16 in Endpoint (EP), Root Port (RP) and Transaction Layer Packet (TLP) Bypass modes. Gen3, Gen4 and Gen5 configurations are natively supported. R-Tile also supports up to 16 SerDes channels through a PHY Interface for PCI Express (PIPE) v5.1.1 in SerDes Architecture mode.

R-Tile serves as a companion tile for the Intel Agilex® 7 devices.

1.1.1. Standards and Specifications Compliance

- PCI Express Base Specification Revision 5.0, Version 1.0
- Single Root I/O Virtualization and Sharing Specification, Revision 1.1
- Address Translation Services, Revision 1.1
- Virtual I/O Device (VIRTIO) Version 1.0
- PHY Interface for PCI Express Specification, Version 5.1.1

  Note: This interface is accessible when the IP is in PIPE mode.

1.2. Features

The R-Tile Avalon® streaming Intel FPGA IP for PCI Express supports the following features:

- Includes a complete protocol stack including the Transaction, Data Link, and Physical Layers implemented as a Hard IP.
- Supports Root Port (RP), Endpoint (EP) and TLP Bypass (BP) modes.

Table 1. Configurations Natively Supported by R-Tile

Endpoint = EP; Root Port = RP; TLP Bypass = BP

Configuration	Application Interface Data Width (bits)	EP/RP/BP
Gen5/Gen4/Gen3 x16	1024	EP/RP/BP
Gen4/Gen3 x16	512 (1)	EP/RP/BP
Gen5/Gen4/Gen3 x8x8	512	EP/RP/BP
Gen4/Gen3 x8x8	256 (1)	EP/RP/BP

continued...

Configuration	Application Interface Data Width (bits)	EP/RP/BP
Gen5/Gen4/Gen3 x4x4x4x4	256	EP/RP/BP
Gen4/Gen3 x4x4x4x4	128 [(1)]	EP/RP/BP
PIPE Direct	64 bits per Lane	N/A

Note: Gen1/Gen2 configurations are supported via link down-training.

Note: When selecting Gen3 or Gen4 configurations, the R-Tile Avalon streaming Intel FPGA IP for PCI Express continues to advertise its capabilities as a device compliant with the 5.0 PCI Express Base Specification.

Table 2. Topologies Supported by R-Tile

Topology\Lane #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x16	Port 0 (EP/RP/BP)															
x8x8	Port 0 (EP/RP/BP)								Port 1 (EP/RP/BP)							
x4x4 x4x4	Port 2 (EP/RP/BP)				Port 0 (EP/RP/BP)				Port 1 (EP/RP/BP)				Port 3 (EP/RP/BP)			
PIPE Direct	PIPE Direct															

Note: Port 2 is only available in the following OPNs:

— AGIx027R29AxxxxR2

— AGIx027R29AxxxxR3

— AGIx027R29BxxxxR3

— AGIx023R18AxxxxR0

— AGIx041R29DxxxxR0

— AGIx041R29DxxxxR1

For additional details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

• The following PIPE Direct bundle modes are supported and are selectable via the **PIPE Direct Mode** menu in the Parameter Editor in Intel® Quartus® Prime:

— 1x16

— 2x8

— 4x4

— 8x2

— 16x1

[(1)] These configurations are only available in the following OPNs: AGIx027R29AxxxxR2, AGIx027R29AxxxxR3, AGIx027R29BxxxxR3, AGIx023R18AxxxxR0, AGIx041R29DxxxxR0, AGIx041R29DxxxxR1. For more details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview

- 2x4 : 1x8
- 4x2 : 1x8
- 8x1 : 1x8
- 1x8 : 2x4
- 4x2 : 2x4
- 8x1 : 2x4
- 1x8 : 4x2
- 2x4 : 4x2
- 8x1 : 4x2
- 1x8 : 8x1
- 2x4 : 8x1
- 4x2 : 8x1

Note: 1x16 means all 16 PIPE Direct channels act in a bundle mode. 16x1 means all 16 channels act as independent channels. 2x8 means the channels are in two 8-channel bundles. 2x4 : 1x8 means the channels in the Lower 8 lanes are organized as two 4-channel bundles, while the channels in the Upper 8 lanes are in an 8-channel bundle. Refer to PIPE Direct Reset Sequence for reset considerations when lanes 0-8 are not in use by the Soft IP Controller.

- Static port bifurcation (x8x8, x4x4x4x4).
- Supports Precision Time Measurement (PTM) (Endpoint only).

  Note: Only Ports 0 and 1 support PTM.
- Supports TLP Bypass mode in Upstream or Downstream configuration.
  - Supports one x16, two x8, or four x4 interfaces.
- Supports up to 512-byte maximum payload size (MPS).
- Supports up to 4096-byte (4 KB) maximum read request size (MRRS).
- Single Virtual Channel (VC).
- Latency Tolerance Reporting (LTR).
- Page Request Services (PRS).

  Note: Only Ports 0 and 1 support PRS.
- MSI and MSI-X.

  Note: Only Ports 0 and 1 support MSI and MSI-X.
- Completion Timeout Ranges.
- Atomic Operations (FetchAdd/Swap/CAS).
- Extended Tag Support.
  - 10-bit Tag Support (Maximum of 768 outstanding tags (x16) / 512 outstanding tags (x8/x4) at any given time, for all functions combined).
- Separate Refclk with Independent Spread Spectrum Clocking (SRIS).
- Separate Refclk with no Spread Spectrum Clocking (SRNS).
- Common Refclk architecture.
- PCI Express Advanced Error Reporting (PF only).

Note: Advanced Error Reporting is always enabled in the R-Tile Avalon streaming Intel FPGA IP for PCIe.

- ECRC generation and checking (when the IP is not in TLP Bypass mode).
  — Application logic needs to handle ECRC generation and checking when the IP is in TLP Bypass mode.
- Data bus parity protection.
- Supports D0 and D3 device power management states.
- Lane Margining at Receiver.
- Retimers presence detection.

- User packet interface with separate header, data and prefix.
- User packet interface with a split-bus architecture where the header, data and prefix busses consist of four segments each (x16 mode only).
- Up to 768 outstanding Non-Posted requests (x16 core only).
- Up to 512 outstanding Non-Posted requests (x8 and x4 cores).
- Summary of outstanding Non-Posted requests supported when 8-bit tags or 10-bit tags are enabled:

Table 3. Outstanding Non-Posted Requests Supported

Ports	Active Cores	8-bit Tags	10-bit Tags
0	x16	256	768 (*)
1	x8	256	512
2 and 3	x4	256	512

Note: (*): Use tags 256 to 1023.

- Completion timeout interface.
  — The PCIe Hard IP can optionally track outgoing non-posted packets to report completion timeout information to the application.
- You cannot change the pin allocations for the R-Tile Avalon streaming Intel FPGA IP for PCI Express in the Intel Quartus Prime project. However, this IP does support lane reversal per port (x16, x8, x4_0, x4_1) and polarity inversion on the PCB by default.
- Supports Autonomous Hard IP mode.
  — This mode allows the PCIe Hard IP to communicate with the Host before the FPGA configuration and entry into User mode are complete.

    Note: Unless Readiness Notifications mechanisms are used, the Root Complex or the system software must allow at least one second after a Conventional Reset of a device before it may determine that a device that fails to return a Successful Completion status for a valid Configuration Request is a broken device. This period is independent of how quickly Link training completes.

- Supports CvP Init and CvP Update.

  Note: For Gen3, Gen4 and Gen5 x16 variants, Port 0 (corresponding to lanes 0 - 15) supports the CvP features. For Gen3, Gen4 and Gen5 x8 variants, only Port 0 (corresponding to lanes 0 - 7) supports the CvP features. Port 1 (corresponding to lanes 8 - 15) does not support CvP.
- **VCS*, VCS* MX, Siemens EDA QuestaSim*, and Xcelium* are the only simulators supported in the latest release of Intel Quartus Prime.**

  Note: The Xcelium* simulator support is only available in the following OPNs:
  - AGIx027R29AxxxxR2
  - AGIx027R29AxxxxR3
  - AGIx027R29BxxxxR3
  - AGIx023R18AxxxxR0
  - AGIx041R29DxxxxR0
  - AGIx041R29DxxxxR1

  For more details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.
- The R-Tile PHY layer does not support sending a Beacon signal.

Note: Throughout this User Guide, the term Avalon-ST may be used as an abbreviation for the Avalon streaming interface or IP.

1.2.1. Multifunction and Virtualization Features

The R-Tile Avalon streaming Intel FPGA IP for PCI Express supports the following multifunction and virtualization features:

- SR-IOV support (Port 0 and Port 1 only).

  Note: Ports 0 and 1 can support 8 PFs and 2K VFs. Ports 2 and 3 do not support SR-IOV, and only support 1 PF.
- Access Control Service (ACS) capability.

  Note: ACS support for Ports 2 and 3 is only available in the following OPNs:
  - AGIx027R29AxxxxR2
  - AGIx027R29AxxxxR3
  - AGIx027R29BxxxxR3
  - AGIx023R18AxxxxR0
  - AGIx041R29DxxxxR0
  - AGIx041R29DxxxxR1

  For additional details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.
- Alternative Routing-ID Interpretation (ARI).
- Function Level Reset (FLR).

  Note: Only Ports 0 and 1 support FLR.
- TLP Processing Hint (TPH).

  Note: TPH supports the "No Steering Tag (ST)" mode only.

- Address Translation Services (ATS).
- Process Address Space ID (PasID).

  Note: Scalable IO and Shared Virtual Memory (SVM) may be available in a future Intel Quartus Prime release.
- Configuration Intercept Interface (CII).
- Soft VirtIO support.

  Note: VirtIO support for Ports 2 and 3 is only available in the following OPNs:
  - AGIx027R29AxxxxR2
  - AGIx027R29AxxxxR3
  - AGIx027R29BxxxxR3
  - AGIx023R18AxxxxR0
  - AGIx041R29DxxxxR0
  - AGIx041R29DxxxxR1

  For additional details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

1.3. Release Information

Table 4. R-Tile Avalon streaming Intel FPGA IP for PCI Express Release Information

Item	Description
IP Version	10.0.0
Intel Quartus Prime Version	23.2
Release Date	June 2023
Ordering Codes	No IP ordering code is required

The IP version (X.Y.Z) number may change from one Intel Quartus Prime software version to another. A change in:

- X indicates a major revision of the IP. If you update your 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.

1.4. Device Family Support

The following terms define device support levels for Intel FPGA IP cores:

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

Table 5. Device Family Support

Device Family	Support Level
Intel Agilex 7	Advance support
Other device families	No support Refer to the *Intel PCI Express Solutions* web page on the Intel website for support information on other device families.

Note: In the Intel Quartus Prime latest release, the R-Tile Avalon Streaming Intel FPGA IP for PCI Express is only available in the Intel Agilex 7 I-Series and M-Series devices.

1.5. Performance and Resource Utilization

The following table shows the recommended FPGA fabric speed grades for all the configurations that the R-Tile Avalon streaming IP core supports.

Table 6. Intel Agilex 7 Recommended FPGA Fabric Speed Grades for All Avalon Streaming Widths and Frequencies

Configuration	Application Clock Frequency (MHz)	Recommended FPGA Fabric Speed Grades	Note
Gen5 1x16 EP/RP/BP	400 MHz (2) 425 MHz 450 MHz 475 MHz 500 MHz	-1, -2, -3 (2)	
Gen4 1x16 EP/RP/BP	250 MHz 275 MHz 300 MHz	-1, -2, -3	

continued...

(2) -3 devices only support an Application Clock frequency up to 400 MHz.

Configuration	Application Clock Frequency (MHz)	Recommended FPGA Fabric Speed Grades	Note
	400 MHz [2] 425 MHz 450 MHz 475 MHz 500 MHz	-1, -2, -3 [2]	
Gen3 1x16 EP/RP/BP	250 MHz 275 MHz 300 MHz	-1, -2, -3	
Gen5 2x8 EP/RP/BP	400 MHz [2] 425 MHz 450 MHz 475 MHz 500 MHz	-1, -2, -3 [2]	
Gen4 2x8 EP/RP/BP	250 MHz 275 MHz 300 MHz	-1, -2, -3	
	400 MHz [2] 425 MHz 450 MHz 475 MHz 500 MHz	-1, -2, -3 [2]	[3]
Gen3 2x8 EP/RP/BP	250 MHz 275 MHz 300 MHz	-1, -2, -3	
Gen5 4x4 EP/RP/BP	400 MHz [2] 425 MHz 450 MHz 475 MHz 500 MHz	-1, -2, -3 [2]	
Gen4 4x4 EP/RP/BP	250 MHz 275 MHz 300 MHz	-1, -2, -3	
	400 MHz [2] 425 MHz 450 MHz 475 MHz 500 MHz	-1, -2, -3 [2]	[3]
Gen3 4x4 EP/RP/BP	250 MHz 275 MHz 300 MHz	-1, -2, -3	
PIPE Direct	500	-1, -2	

[3] This topology is only available in the following OPNs: AGIx027R29AxxxxR2, AGIx027R29AxxxxR3, AGIx027R29BxxxxR3, AGIx023R18AxxxxR0, AGIx041R29DxxxxR0, AGIx041R29DxxxxR1. For additional details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

The following table shows the typical resource utilization information for selected configurations.

The resource usage is based on the Avalon streaming IP core top-level entity (`intel_rtile_pcie_ast`) that includes IP core soft logic implemented in the FPGA fabric.

Table 7. Resource Utilization Information for the R-Tile Avalon Streaming IP

Link Configuration	Device Family	ALMs	M20Ks	Dedicated Logic Registers
Gen5 x16	Intel Agilex 7	11721	0	32819
Gen4 x16	Intel Agilex 7	11617	0	28127
Gen3 x16	Intel Agilex 7	11617	0	28127
16-channel PIPE Direct	Intel Agilex 7	2257	0	1836

For more details on the R-Tile Avalon Streaming design example, refer to R-Tile Avalon Streaming Intel FPGA IP for PCI Express Design Example User Guide.

1.6. IP Core Support Levels

The following table shows the support levels of the R-Tile Avalon streaming Intel FPGA IP for PCI Express IP core in Intel Agilex 7 devices.

Table 8. R-Tile Avalon streaming Intel FPGA IP for PCIe Support Matrix for Intel Agilex 7 Devices

Support level keys: S = simulation, C = compilation, T = timing, H = hardware, N/A = configuration not supported

Configuration	PCIe IP Support		
	EP	RP	BP UP/DN
16-channel PIPE Direct	N/A	N/A	N/A
Gen5 x16 1024-bit	SCTH	SCTH	SCTH
Gen4 x16 1024-bit	SCTH	SCTH	SCTH
Gen3 x16 1024-bit	SCTH	SCTH	SCTH
Gen4 x16 512-bit [(4)]	SCTH	SCTH	SCTH
Gen3 x16 512-bit [(4)]	SCTH	SCTH	SCTH
Gen5 x8/x8 512-bit	SCTH	SCTH	SCTH
Gen4 x8/x8 512-bit	SCTH	SCTH	SCTH
Gen3 x8/x8 512-bit	SCTH	SCTH	SCTH
Gen4 x8/x8 256-bit [(4)]	SCTH	SCTH	SCTH

continued...

(4) These configurations are only available in the following OPNs: AGIx027R29AxxxxR2, AGIx027R29AxxxxR3, AGIx027R29BxxxxR3, AGIx023R18AxxxxR0, AGIx041R29DxxxxR0, AGIx041R29DxxxxR1. For more details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

Configuration	PCIe IP Support		
	EP	RP	BP UP/DN
Gen3 x8/x8 256-bit [4]	SCTH	SCTH	SCTH
Gen5 x4/x4/x4/x4 256-bit	SCTH	SCTH	SCTH
Gen4 x4/x4/x4/x4 256-bit	SCTH	SCTH	SCTH
Gen3 x4/x4/x4/x4 256-bit	SCTH	SCTH	SCTH
Gen4 x4/x4/x4/x4 128-bit [4]	SCTH	SCTH	SCTH
Gen3 x4/x4/x4/x4 128-bit [4]	SCTH	SCTH	SCTH

Note: PIO design examples are available only in the x16 and 2x8 EP modes in the 22.2 release of Intel Quartus Prime. For additional details, refer to R-Tile Avalon Streaming Intel FPGA IP for PCI Express* Design Example User Guide.

2. IP Architecture and Functional Description

2.1. Overview

R-Tile can be configured in one of three primary modes of operation:

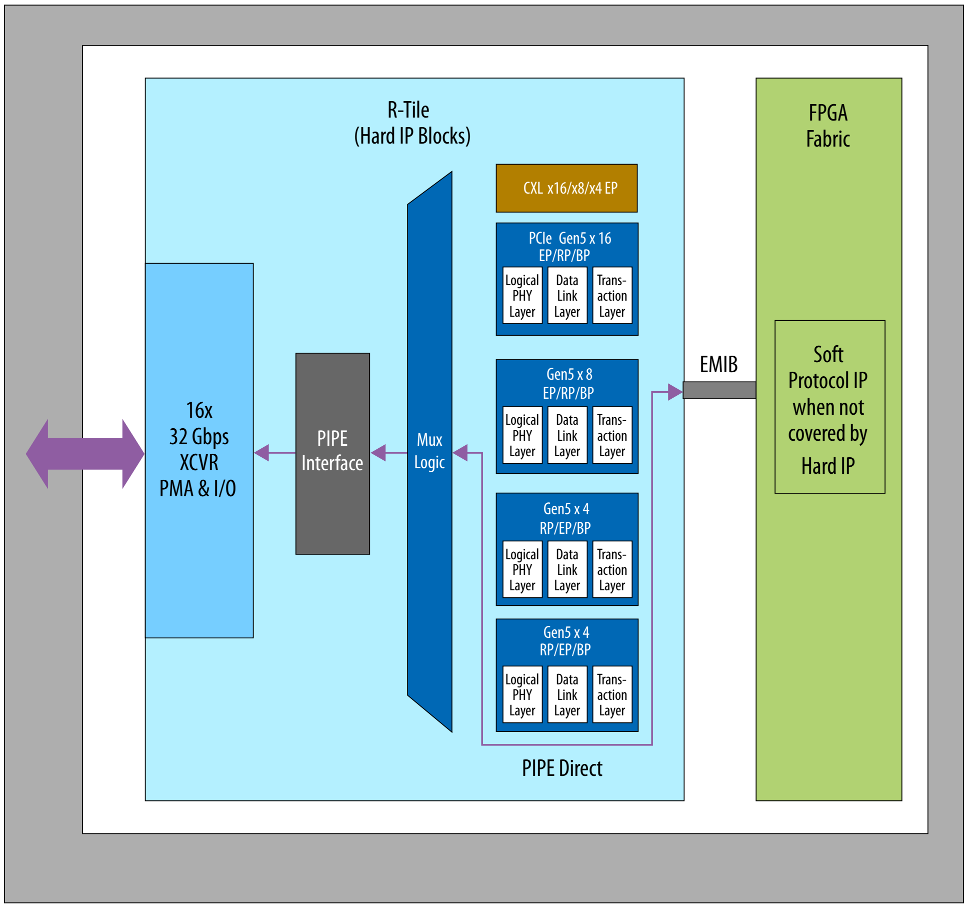
- PCIe Hard IP mode: This mode includes support for (up to Gen5) Endpoint (EP), Root Port (RP) or TLP Bypass (16 lanes maximum). When configured in this mode, R-Tile includes a complete protocol stack, including the Transaction, Data Link and Physical Layers.
- PIPE Direct (protocol controller bypass) for FPGA user custom application needs. In this mode, both the PCIe and CXL controller stacks are entirely bypassed, and the PIPE SerDes mode interface is exported across the Embedded Multi-die Interconnect Bridge (EMIB) to the FPGA fabric. This mode allows you to implement your own custom controllers in soft IP.
- Compute Express Link (CXL).

  Note: For more details on the Intel Agilex 7 R-Tile Compute Express Link 1.1 Intel FPGA IP and the corresponding design examples, refer to:

  — Intel Agilex 7 R-Tile Compute Express Link 1.1 Intel FPGA IP User Guide. This document is available in the Intel Resource and Documentation Center (RDC). You can download a copy or ask your local Intel Field Applications engineer to download one using the asset number 763328.

  — Intel Agilex 7 R-Tile Compute Express Link 1.1 Intel FPGA IP Design Example User Guide. This document is available in the Intel Resource and Documentation Center (RDC). The asset number is 763513.

Figure 1. R-Tile Top-Level Block Diagram



2.2. PCIe Hard IP Mode

In this mode, the four cores (one x16 core, one x8 core and two x4 cores) in the PCIe Hard IP can be configured to support the following topologies:

Table 9. Configuration Modes Supported by the R-Tile Avalon Streaming Intel FPGA IP for PCI Express

Configuration Mode	Native IP Mode	Endpoint (EP) / Root Port (RP) / TLP Bypass (BP)	Active Hard IP Cores
Configuration Mode 0	Gen3 x16 or Gen4 x16 or Gen5 x16	EP/RP/BP	x16
Configuration Mode 1	Gen3 x8/Gen3 x8 or Gen4 x8/Gen4 x8 or Gen5 x8/Gen5 x8	EP/RP/BP	x16, x8
Configuration Mode 2	Gen3 x4/Gen3 x4/Gen3 x4/Gen3 x4 or Gen4 x4/Gen4 x4/Gen4 x4/Gen4 x4 or Gen5 x4/Gen5 x4/Gen5 x4/Gen5 x4	EP/RP/BP	x16, x8, x4_0, x4_1
Configuration Mode 3	PIPE Direct (with a maximum of 16 channels)	N/A	None

In Configuration Mode 0, only the x16 core is active, and it operates in x16 mode (in Gen3, Gen4 or Gen5).

In Configuration Mode 1, the x16 core and x8 core are active, and they operate as two Gen3 x8 cores, two Gen4 x8 cores or two Gen5 x8 cores.

Note: **In Configuration Mode 1, when you use only one of the x8 bifurcated ports, you must ensure that the other bifurcated port's lanes are not physically connected. If you connect both x8 bifurcated ports to a x16 Root Port/Switch device, it is non-deterministic which x8 port will be trained.**

In Configuration Mode 2, all four cores (x16, x8, x4_0, x4_1) are active, and they operate as four Gen3 x4 cores, four Gen4 x4 cores or four Gen5 x4 cores.

Note: In Configuration Mode 2, for the latest release of Intel Quartus Prime, the x4_0 core is disabled for AGI OPNs with the suffix R0. For these devices, the maximum number of active x4 cores for Configuration Mode 2 is three (with these active cores being the x16, x8 and x4_1 cores, all configured as x4 cores). However, AGI OPNs with the R2 or R3 suffix, and AGM OPNs can support the x16, x8, x4_1 and x4_0 cores all being active while in Configuration Mode 2. For additional details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

Each of the cores has its own Avalon-ST interface to the user logic. The number of IP-to-User Logic interfaces exposed to the FPGA fabric are different based on the configuration modes.

2.2.1. Clocking

In the PCIe Hard IP mode, the R-Tile Avalon Streaming Intel FPGA IP for PCI Express has four primary clock domains:

- PHY clock domain (i.e. `core_clk` domain): this clock is synchronous to the SerDes parallel clock.
- EMIB/FPGA fabric interface clock domain (i.e. `pld_clk` domain): this clock is derived depending on the OPN being used and the parameter **Enable Independent perst pins**. If this parameter is set to disabled, this clock is derived from `refclk0`. On the other hand, if **Enable Independent perst pins** is set to enabled, this clock is derived from `refclk2`. Refer to the Intel Agilex® 7 Device Family Pin Connection Guidelines for further details on the implementation of the `refclk` pins available for your specific OPN.
- Application clock domain (`coreclkout_hip`) for in-band signals: this clock is an output from the R-Tile IP, and it has the same frequency as `pld_clk`.
- Application clock domain (`slow_clk`) for sideband signals: this clock is another output from the R-Tile IP. It is a divide-by-2/4 version of `coreclkout_hip`.

Figure 2. Clock Domains in PCIe Modes

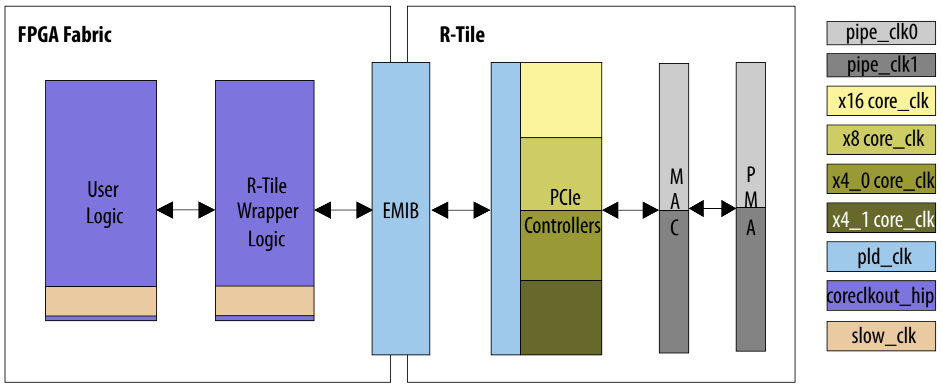


Table 10. PHY Clock and Application Clock Frequencies

Mode	PHY Clock Frequency	Application Clock Frequency
PCIe Gen1	1000 MHz	Gen1 is supported only via link down-training and not natively. Hence, the application clock frequency depends on the configuration you choose in the IP Parameter Editor. For example, if you choose a Gen3 configuration, the application clock frequency is 250 MHz - 300 MHz.
PCIe Gen2	1000 MHz	Gen2 is supported only via link down-training and not natively. Hence, the application clock frequency depends on the configuration you choose in the IP Parameter Editor. For example, if you choose a Gen3 configuration, the application clock frequency is 250 MHz - 300 MHz.
PCIe Gen3	1000 MHz	250 MHz - 500 MHz (*)
PCIe Gen4	1000 MHz	250 MHz - 500 MHz (*)
PCIe Gen5	1000 MHz	400 MHz - 500 MHz

Note: (*) The highest frequencies at Gen3 and Gen4 for the Application Clock Frequency are only available in the following OPNs:

- AGIx027R29AxxxxR2
- AGIx027R29AxxxxR3
- AGIx027R29BxxxxR3
- AGIx023R18AxxxxR0
- AGIx041R29DxxxxR0
- AGIx041R29DxxxxR1

For additional details, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

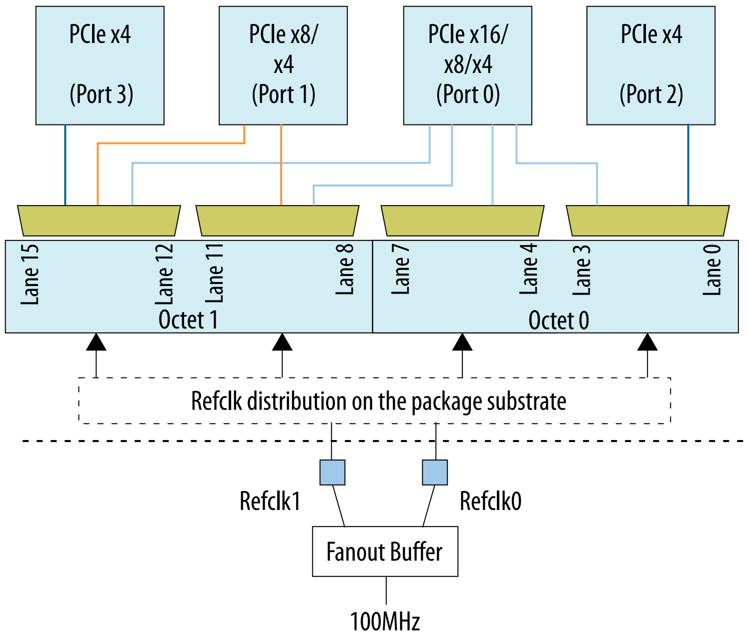
Note: For a link down-training scenario when R-Tile is configured at Gen3, Gen4 or Gen5 and the link gets down-trained to a lower speed, the application clock frequency will continue to run at the configured frequency set in the **PLD Clock Frequency** parameter. For example, when the **PCIe Hard IP Mode** parameter is set as Gen5 1x16 and the **PLD Clock Frequency** parameter as 500 MHz, the PLD clock frequency will continue to run at 500 MHz even if the link is down-trained to Gen4 or less.

R-Tile may have up to three reference clock inputs at the package level, `refclk0`, `refclk1`, and `refclk2`, depending on the OPN being used. Refer to the Intel Agilex 7 Device Family Pin Connection Guidelines for further details on the implementation of the `refclk` pins available for your specific OPN.

The connection requirements for these input clocks depend on the Configuration mode being used and the **Enable Independent perst pins** parameter. You must consider the following guidelines:

- Connect a 100 MHz reference clock source to `refclk0` and `refclk1`.
- If using Configuration Mode 0 (1x16) or Configuration Mode 2 (4x4), drive `refclk0` and `refclk1` using a single clock source.

Figure 3. Using a Single 100 MHz Clock Source in 1x16 and 4x4 Modes



In Configuration Mode 1 (2x8), you can drive the `refclk0` and `refclk1` inputs with either a single 100 MHz clock source as shown above, or two independent 100 MHz sources (see Using Independent 100 MHz Clock Sources in 2x8 Mode) depending on your system architecture. For example, if your system has each x8 port connected to a separate CPU/Root Complex, it may be required to drive these `refclk` inputs using independent clock sources. In that case, if the parameter **Enable Independent perst pins** is set to disabled, the `refclk0` input for Port 0 must always be running because it feeds the reference clock for the R-Tile core PLL that controls the data transfers between the R-Tile and FPGA fabric via the EMIB. If this clock goes down, Port 0 link will go down and Port 1 will not be able to communicate with the FPGA fabric.

On the other hand, if the parameter **Enable Independent perst pins** is set to enabled, the `refclk2` input must always be running because it feeds the R-Tile core PLL that controls the data transfers between the R-Tile and FPGA fabric via the EMIB.

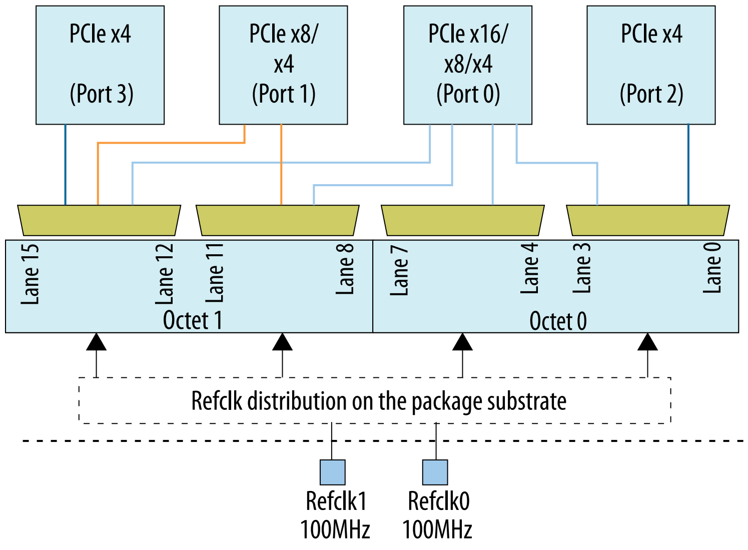
Following are the guidelines for implementing two independent refclks in Configuration Mode 1 (2x8) with the parameter **Enable Independent perst pins** set to disabled:

- If the link can handle two separate reference clocks, drive the `refclk0` of R-Tile with the on-board free-running oscillator.
- If the link needs to use a common reference clock, then PERST# needs to indicate the stability of this reference clock. If this reference clock goes down, the entire R-Tile must be reset.

Following are the guidelines for implementing two independent refclks in Configuration Mode 1 (2x8) with the parameter **Enable Independent perst pins** set to enabled:

- Drive the additional `refclk2` of R-Tile with an on-board free-running oscillator.
- The pin `pin_perst_n` indicates the stability of this reference clock. If this reference clock goes down, the entire R-Tile must be reset.

Figure 4. Using Independent 100 MHz Clock Sources in 2x8 Mode



2.2.2. Reset

Follow the guidelines below for a proper reset implementation for the R-Tile Avalon Streaming Intel FPGA IP for PCI Express:

- The `pin_perst_n` signal affects the entire R-Tile. Toggling `pin_perst_n` will affect all the active cores.
- When using the Configuration Mode 1 (2x8 Endpoint only) and based on the OPN used in the Intel Quartus project, you have the additional PERST# pins: `pin_perst0` and `pin_perst1`, to independently reset port 0 and port 1 respectively. Refer to section Independent PERST pin for additional information and the OPNs that support this feature.

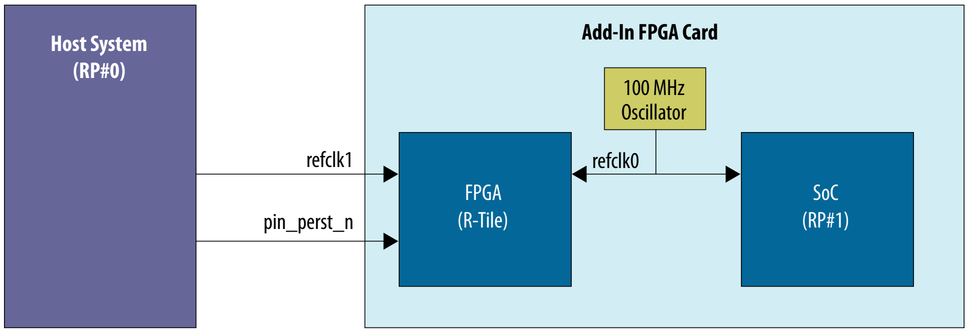
  Note: Refer to the Intel Agilex 7 Device Family Pin Connection Guidelines for additional details on the correct implementation and usage of `pin_perst0` and `pin_perst1`.
- The `pin_perst_n` signal must qualify that both `refclk0` and `refclk1` are stable, when using the following Configuration Modes:
  - Configuration Mode 0 (1x16)
  - Configuration Mode 1 (2x8) with the parameter **Enable Independent perst pins** set to disabled
  - Configuration Mode 2 (4x4)

  In case one of the reference clocks becomes stable later, deassert `pin_perst_n` after this reference clock becomes stable.
- The `pin_perst0` must qualify the `refclk0` stability, and `pin_perst1` must qualify the `refclk1` stability when using Configuration Mode 1 (2x8 Endpoint only) with the parameter **Enable Independent perst pins** set to enabled.
- `pin_perst_n` assertion is required for the Autonomous mode functionality in the R-Tile Avalon Streaming Intel FPGA IP for PCIe. In Autonomous mode (enabled by default), the IP can successfully link up upon the release of `pin_perst_n` regardless of the FPGA fabric configuration and will send out Completion TLPs with the Configuration Retry Status (CRS) set until the FPGA fabric is configured and ready.
- `pin_perst_n` assertion should be avoided during a functional-level reset or before a functional-level reset is completed since this could impact the link training process. In case this occurs, a cold reset would be required to properly complete the link training process.
- The `pX_reset_status_n_o` signal from the R-Tile Avalon Streaming Intel FPGA IP for PCI Express includes an accumulative characteristic related to the number of back-to-back `pin_perst_n` assertions. Each back-to-back `pin_perst_n` event will be queued and executed one after the other, affecting the total time it takes for the R-Tile Avalon Streaming Intel FPGA IP for PCI Express to come out of reset and deassert the `pX_reset_status_n_o` signal. For additional information on the `pX_reset_status_n_o` signal, refer to Resets.

2.2.2.1. Single PERST

The following is an example where a single PERST# (pin_perst_n) is driven with independent `refclk0` and `refclk1`. In this example, the add-in card (FPGA and Soc) is powered up first. The R-Tile `refclk0` is fed by the on-board free-running oscillator and the `refclk1` driven by the Host becomes stable later. Hence, the PERST# is connected to the Host.

Figure 5. Single PERST# Connection in Bifurcated 2x8 Mode



2.2.2.2. Independent PERST Pins

Depending on the OPN being used, the R-Tile Avalon Streaming Intel FPGA IP for PCIe allows the flexibility to handle the independent reset operation for each of the active PCIe cores when Configuration Mode 1 (2x8 Endpoint only) is selected. For more information on the Configuration Modes, refer to Configuration Modes Supported by the R-Tile Avalon Streaming Intel FPGA IP for PCI Express.

The OPNs that support the additional independent PERST pins at the package level are: AGI*041R29D*R1 (for example, AGIB041R29D1E2VR1).

When you enable the **Enable Independent Perst pins** parameter in the IP Parameter Editor, the additional `pin_perst0_n`, `pin_perst1_n` at the package level must be used. In addition, the `pX_warm_pest_n_i` optional ports become available.

Consider the following guidelines for handling independent reset operations:

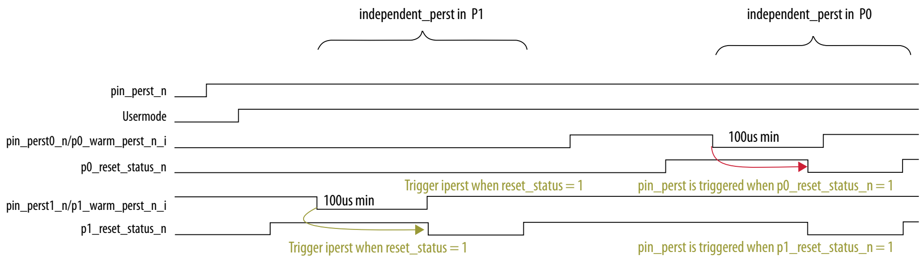
- The `pin_perst0_n`, `pin_perst1_n` input ports can trigger a cold reset. This clears sticky bits and resets the physical layer.
- The `pin_perst_n` continues to affect the entire R-Tile. Toggling `pin_perst_n` affects both port 0 and port 1.
- `pX_warm_perst_n_i` input ports can trigger a warm reset. This will not clear the sticky bits but will reset the physical layer.

Input Port Used	Sticky Bits Clearing	Non-Sticky Bits Clearing	PHY Lane Reset
`pin_perst_n`	Yes	Yes	Yes
`pin_perst0_n`, `pin_perst1_n`	Yes	Yes	Yes
`pX_warm_perst_n_i`	No	Yes	Yes

- `pin_perst_n` has the highest priority for reset over `pin_perst0_n`, `pin_perst1_n` or `pX_warm_perst_n_i` ports.
- When `pin_perst_n` is deasserted (i.e., high), the `pin_perst0_n` and `pin_perst1_n` input ports can be used to trigger a cold reset operation in each of the PCIe cores independently.

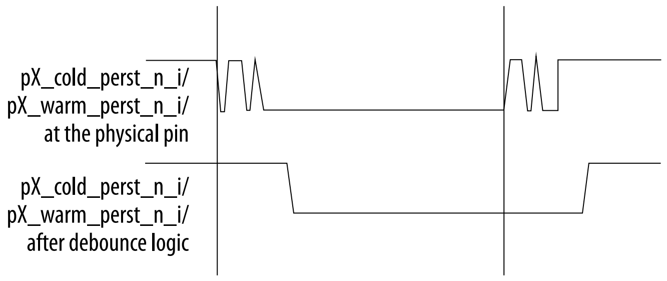
- When pin_perst_n, pin_perst0_n and pin_perst1_n are deasserted (i.e., high), the pX_warm_perst_n_i input ports can be used to trigger a warm reset operation in each of the PCIe cores independently.

Figure 6. pX_warm_perst_n_i vs pin_perst_n/pin_perst0_n/pin_perst1_n behavior


independent_perst in P1
independent_perst in P0
pin_perst_n
Usermode
pin_perst0_n/p0_warm_perst_n_i
p0_reset_status_n
pin_perst1_n/p1_warm_perst_n_i
p1_reset_status_n
100us min
100us min
Trigger iperst when reset_status = 1
pin_perst is triggered when p0_reset_status_n = 1
Trigger iperst when reset_status = 1
pin_perst is triggered when p1_reset_status_n = 1

- Concurrent assertions of the reset input ports pin_perst_n, pin_perst0_n, pin_perst1_n and pX_warm_perst_n_i are not supported.
- Usage of the pX_warm_perst_n_i to perform a warm reset to one of the active cores must happen only after the deassertion (i.e. high) of the corresponding pX_reset_status_n port. As an example, in Configuration Mode 2 (2x8), in order to trigger an independent warm reset operation on p0_warm_perst_n_i, the p0_reset_status_n must be deasserted (i.e. high).
- As is the case with pin_perst_n, once pin_perst0_n or pin_perst1_n has been asserted (i.e. low), the assertion needs to be held for a minimum of 100ms.
- As is the case with pin_perst_n, once pX_warm_perst_n_i has been asserted (i.e. low), the assertion needs to be held for a minimum of 100ms.
- pX_warm_perst_n_i assertion should be avoided during a functional-level reset or before a functional-level reset is completed since this could impact the link training process. In case this occurs, assert the corresponding pin_perst0_n or pin_perst1_n to properly complete the link training process.
- When the pX_warm_perst_n_i ports are routed to General Purpose I/Os (GPIOs), the Application logic must implement a debounce logic to prevent the switch bouncing and triggering unintentional assertions. The debounce logic consists of a counter that waits for the signal to stabilize before propagating it to the targeted port. In case these ports are not routed to GPIOs and are being used by internal fabric logic only, the debounce logic is not necessary.

Figure 7. pX_warm_perst_n_i Signal Before and After Debounce Logic



2.2.2.2.1. REFCLK and PERST Guidelines When Using Independent PERST Pins

In Configuration Mode 1 (2x8 Endpoint only), the independent PERST and REFCLK are available with:

- The IP parameter **Enable Independent Perst pins** set to enabled.
- The clock source for `refclk2` must be always running.
- The `pin_perst_n` pin must qualify the stability of `refclk2`.
- The `pin_perst_n` pin has the highest priority and toggling it affects both Port 0 and Port 1.
- For Port 0 (P0):
  - The clock source comes from `refclk0`.
  - The reset source can come from `pin_perst0_n` at the package level or `p0_warm_perst_n_i`.
- For Port 1 (P1):
  - The clock source comes from `refclk1`.
  - The reset source can come from `pin_perst1_n` at the package level or `p1_warm_perst_n_i`.

2.2.2.3. Independent GPIO PERST

The R-Tile Avalon Streaming Intel FPGA IP for PCIe allows further flexibility to handle independent reset operation for each of the active PCIe cores. The active PCIe cores depend on the Configuration Mode selected for the IP. For more information on the Configuration Modes, refer to Configuration Modes Supported by the R-Tile Avalon Streaming Intel FPGA IP for PCI Express.

When you enable the **Enable Independent GPIO Perst** parameter in the IP Parameter Editor, the additional `pX_cold_perst_n_i`, `pX_warm_perst_n_i`, and `pX_ip_rst_n_o` ports become available.

Consider the following guidelines for handling independent reset operations:

- `pin_perst_n` or `pX_cold_perst_n_i` input ports can trigger a cold reset. This will clear sticky bits and reset the physical layer.
- `pX_warm_perst_n_i` input ports can trigger a warm reset. This will not clear the sticky bits but will reset the physical layer.

Input Port Used	Sticky Bits Clearing	Non-Sticky Bits Clearing	PHY Lane Reset
pin_perst_n	Yes	Yes	Yes
pX_cold_perst_n_i	Yes	Yes	Yes
pX_warm_perst_n_i	No	Yes	Yes

- `pin_perst_n` has the highest priority for reset over the `pX_cold_perst_n_i` or `pX_warm_perst_n_i` ports.
- When `pin_perst_n` is asserted (i.e. low), all active PCIe cores will reset.
- When `pin_perst_n` is deasserted (i.e. high), the `pX_cold_perst_n_i` input ports can be used to trigger a cold reset operation on each of the PCIe cores independently.

- When pin_perst_n is deasserted (i.e. high), the pX_warm_perst_n_i input ports can be used to trigger a warm reset operation on each of the PCIe cores independently.

Figure 8. pX_cold_perst_n_i/pX_warm_perst_n_i vs. pin_perst_n Behavior

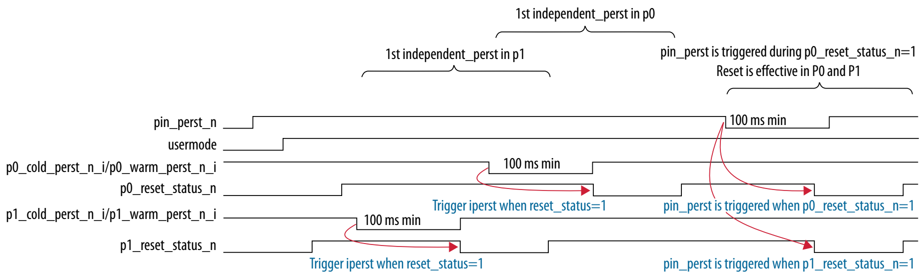
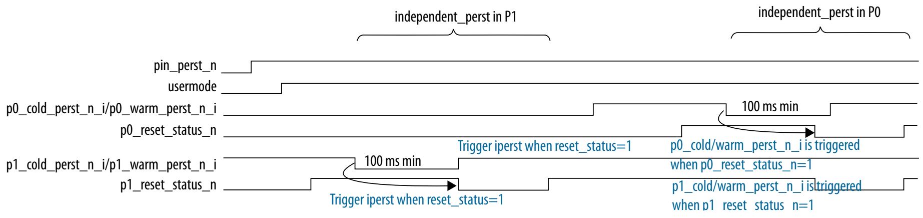

1st independent_perst in p0
1st independent_perst in p1
pin_perst is triggered during p0_reset_status_n=1
Reset is effective in P0 and P1
pin_perst_n
usermode
p0_cold_perst_n_i/p0_warm_perst_n_i
p0_reset_status_n
p1_cold_perst_n_i/p1_warm_perst_n_i
p1_reset_status_n
100 ms min
100 ms min
100 ms min
Trigger iperst when reset_status=1
pin_perst is triggered when p0_reset_status_n=1
Trigger iperst when reset_status=1
pin_perst is triggered when p1_reset_status_n=1

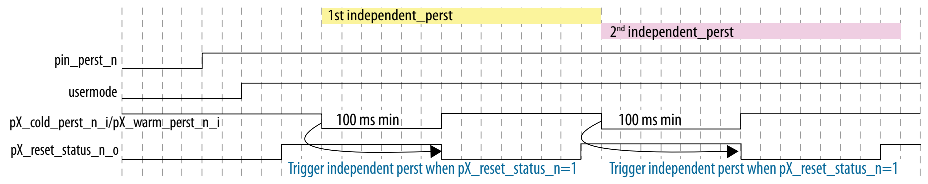
Figure 9. pX_cold_perst_n_i/pX_warm_perst_n_i Behavior


independent_perst in P1
independent_perst in P0
pin_perst_n
usermode
p0_cold_perst_n_i/p0_warm_perst_n_i
p0_reset_status_n
p1_cold_perst_n_i/p1_warm_perst_n_i
p1_reset_status_n
100 ms min
100 ms min
Trigger iperst when reset_status=1
p0_cold/warm_perst_n_i is triggered when p0_reset_status_n=1
p1_cold/warm_perst_n_i is triggered when p1_reset_status_n=1
Trigger iperst when reset_status=1

- Concurrent assertions of the reset input ports pin_perst_n, pX_cold_perst_n_i and pX_warm_perst_n_i are not supported.
- Usage of the pX_cold_perst_n_i to perform a cold reset to one of the active cores must happen only after the deassertion (i.e. high) of the corresponding pX_reset_status_n port. As an example, in Configuration Mode 2 (x8x8), in order to trigger an independent cold reset operation on p0_cold_perst_n_i, the p0_reset_status_n must be deasserted (i.e. high).
- Usage of the pX_warm_perst_n_i to perform a warm reset to one of the active cores must happen only after the deassertion (i.e. high) of the corresponding pX_reset_status_n port. As an example, in Configuration Mode 2 (x8x8), in order to trigger an independent warm reset operation on p0_warm_perst_n_i, the p0_reset_status_n must be deasserted (i.e. high).
- Similarly to the pin_perst_n, once pX_cold_perst_n_i has been asserted (i.e. low), the assertion needs to be held for a minimum of 100ms.

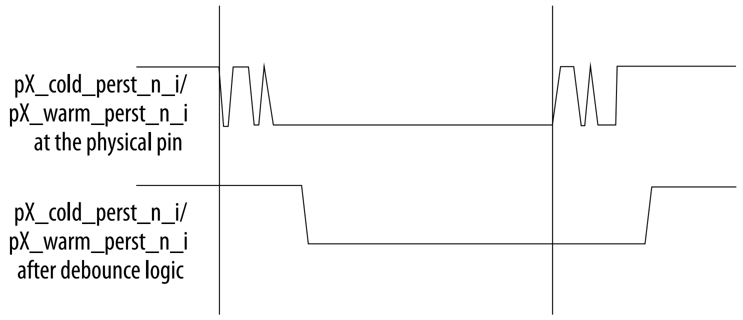
- Similarly to the `pin_perst_n`, once `pX_warm_perst_n_i` has been asserted (i.e. low), the assertion needs to be held for a minimum of 100ms.

Figure 10. Interval Between Subsequent Independent PERST Operations


1st independent_perst
2nd independent_perst
pin_perst_n
usermode
pX_cold_perst_n_i/pX_warm_perst_n_i
pX_reset_status_n_o
100 ms min
100 ms min
Trigger independent perst when pX_reset_status_n=1
Trigger independent perst when pX_reset_status_n=1

- `pX_cold_perst_n_i` or `pX_warm_perst_n_i` assertion should be avoided during a functional-level reset or before a functional-level reset is completed since this could impact the link training process. In case this occurs, assert `pin_perst_n` to properly complete the link training process.
- When the `pX_cold_perst_n_i` or `pX_warm_perst_n_i` ports are routed to General Purpose I/Os (GPIOs), the Application logic must implement a debounce logic to prevent the switch bouncing and triggering unintentional assertions. The debounce logic consists of a counter that waits for the signal to stabilize before propagating it to the targeted port. In case these ports are not routed to GPIOs and are being used by internal fabric logic only, the debounce logic is not necessary.

Figure 11. `pX_cold_perst_n_i/pX_warm_perst_n_i` Signals Before and After Debounce Logic



2.2.2.3.1. REFCLK and PERST Guidelines when Using Independent GPIO PERST

- In Configuration Mode 0 (1x16), the independent PERST and independent REFCLK are available with:
  - The clock coming from a single source connected to `refclk0` and `refclk1`.
  - The reset coming from `pin_perst_n`.
- In Configuration Mode 1 (2x8), the independent PERST and REFCLK are conditional with:
  - Port 0 (P0):
    - The clock source comes from `refclk0`.
    - The reset source can come from `pin_perst_n`, `p0_cold_perst_n_i` or `p0_warm_perst_n_i`.
  - Port 1 (P1):
    - The clock source comes from `refclk1`.
    - The reset source can come from `pin_perst_n`, `p1_cold_perst_n_i` or `p1_warm_perst_n_i`.
  - `refclk0` needs to be always active. A separate reference clock architecture is suggested for this use case. Refer to the *PCI Express Base Specification 5.0* for additional details on this architecture.
  - As stated in the Independent PERST section, `pin_perst_n` has the highest priority for reset over `pX_cold_perst_n_i` or `pX_warm_perst_n_i`, and toggling `pin_perst_n` affects all the ports.

- In Configuration Mode 2 (4x4), the independent PERST feature is available, but REFCLKs are not independent.
  - Port 0 (P0):
    - The clock source comes from `refclk0`.
    - The reset can come from `pin_perst_n`, `p0_cold_perst_n_i` or `p0_warm_perst_n_i`.
  - Port 1 (P1):
    - The clock source comes from `refclk1`.
    - The reset can come from `pin_perst_n`, `p1_cold_perst_n_i` or `p1_warm_perst_n_i`.
  - Port 2 (P2):
    - The clock source comes from `refclk0`.
    - The reset can come from `pin_perst_n`, `p2_cold_perst_n_i` or `p2_warm_perst_n_i`.
  - Port 3 (P3):
    - The clock source comes from `refclk1`.
    - The reset can come from `pin_perst_n`, `p3_cold_perst_n_i` or `p3_warm_perst_n_i`.
  - Both `refclk0` and `refclk1` need to be always active. A separate reference clock architecture is suggested for this use case. Refer to the *PCI Express Base Specification 5.0* for additional details on this architecture.
  - As stated in the Independent PERST section, `pin_perst_n` has the highest priority for reset over `pX_cold_perst_n_i` or `pX_warm_perst_n_i`, and toggling `pin_perst_n` affects all the ports.

2.2.3. PCI Express Protocol Stack

2.2.3.1. PMA

The R-Tile Avalon-ST IP for PCI Express contains Physical Medium Attachment (PMA) for handling the Physical layer (PHY) packets. Note that the R-Tile PMA implements the PIPE SerDes Architecture mode, and the Physical Coding Sublayer (PCS) responsibilities are implemented by the logic PHY/MAC layer. The PMA receives and transmits high-speed serial data on the serial lanes. Since the PMA implements the PIPE SerDes Architecture mode, the logic PHY performs functions like data encoding and decoding, scrambling and descrambling, block synchronization etc. The logic PHY and PCIe controller are only available when the PCIe Hard IP mode is used. Refer to PIPE Direct Mode on page 33 for a functional description of the PIPE mode.

The R-Tile PMA consists of two octets. Each octet contains a pair of transmit PLLs and eight SerDes lanes capable of running up to 32 GT/s to perform the various TX and RX functions.

The Slow PLL generates the required transmit clocks for Gen1/Gen2 speeds, while the Fast PLL generates the required clocks for Gen3/Gen4/Gen5 speeds.

The PMA performs functions such as serialization/deserialization, clock data recovery, and analog front-end functions such as Continuous Time Linear Equalizer (CTLE), Decision Feedback Equalizer (DFE) and transmit equalization.

The transmitter consists of a 3-tap equalizer with one tap of pre-cursor, one tap of main cursor and one tap of post-cursor.

The receiver consists of attenuation (ATT), CTLE, Voltage gain amplifier (VGA) and DFE blocks that are adaptive for Gen3/Gen4/Gen5 speeds. RX Lane Margining is supported by the PHY. The Lane Margining supports timing and voltage margining.

Timing margining capabilities are as follows:

- Maximum Timing Offset: -0.5UI to +0.5UI
- Number of timing steps: 63
- Independent left and right timing margining is not supported.
- Independent Error Sampler is not supported (lane margining may produce logical errors in the data stream and cause the LTSSM to go to the Recovery state).

Voltage margining capabilities are as follows:

- Maximum Voltage Offset: -120mV to +120mV
- Number of voltage steps: 127

Related Information

PHY Interface for PCI Express (PIPE) Base Specification

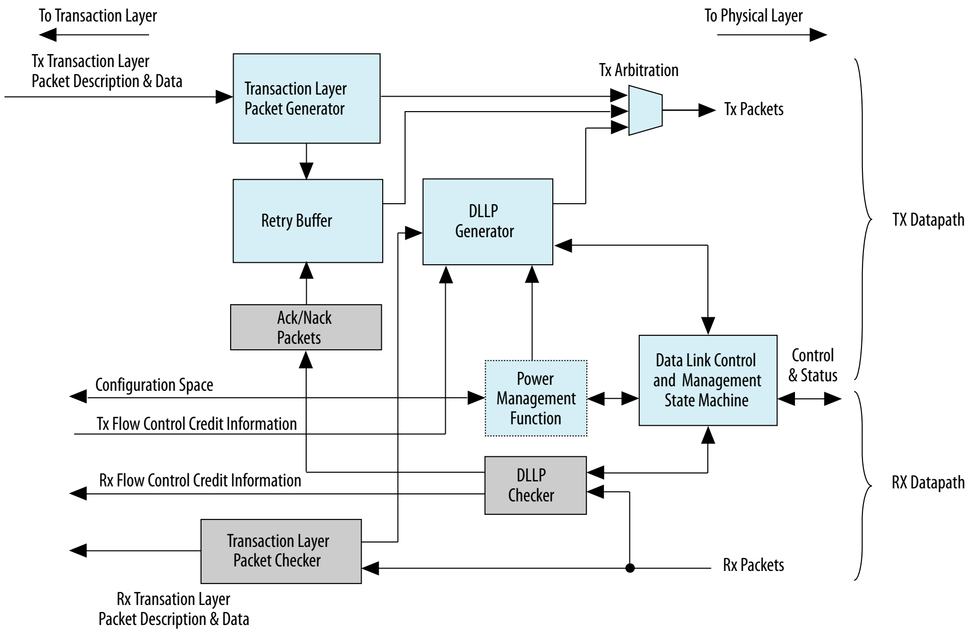
2.2.3.2. Data Link Layer Overview

The Data Link Layer (DLL) is located between the Transaction Layer and the Physical Layer. It maintains packet integrity and communicates (by DLL packet transmission) at the PCI Express link level.

The DLL implements the following functions:

- Link management through the reception and transmission of DLL Packets (DLLP), which are used for the following functions:
  - Power management of DLLP reception and transmission
  - To transmit and receive ACK/NAK packets
  - Data integrity through the generation and checking of CRCs for TLPs and DLLPs
  - TLP retransmission in case of NAK DLLP reception or replay timeout, using the retry (replay) buffer
  - Management of the retry buffer
  - Link retraining requests in case of error through the Link Training and Status State Machine (LTSSM) of the Physical Layer

Figure 12. Data Link Layer


To Transaction Layer
To Physical Layer
Tx Transaction Layer Packet Description & Data
Transaction Layer Packet Generator
Tx Arbitration
Tx Packets
Retry Buffer
DLLP Generator
TX Datapath
Ack/Nack Packets
Configuration Space
Power Management Function
Data Link Control and Management State Machine
Control & Status
Tx Flow Control Credit Information
Rx Flow Control Credit Information
DLLP Checker
RX Datapath
Transaction Layer Packet Checker
Rx Packets
Rx Transation Layer Packet Description & Data

The DLL has the following sub-blocks:

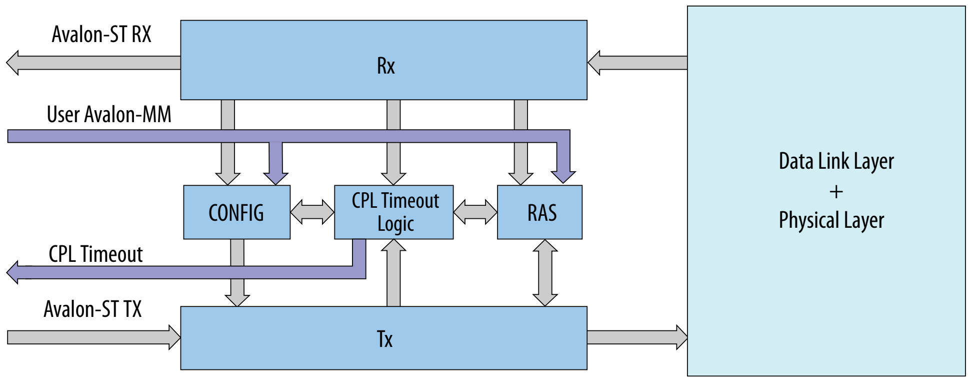
- Data Link Control and Management State Machine—This state machine connects to both the Physical Layer's LTSSM state machine and the Transaction Layer. It initializes the link and flow control credits and reports status to the Transaction Layer.
- Power Management—This function handles the handshake to enter low power mode. Such a transition is based on register values in the Configuration Space and received Power Management (PM) DLLPs. For more details on the power states supported by the R-Tile Avalon-ST IP for PCIe, refer to section Power Management Interface on page 110.
- Data Link Layer Packet Generator and Checker—This block is associated with the DLLP's 16-bit CRC and maintains the integrity of transmitted packets.
- Transaction Layer Packet Generator—This block generates transmit packets, including a sequence number and a 32-bit Link CRC (LCRC). The packets are also sent to the retry buffer for internal storage. In retry mode, the TLP generator receives the packets from the retry buffer and generates the CRC for the transmit packet.
- Retry Buffer—The retry buffer stores TLPs and retransmits all unacknowledged packets in the case of NAK DLLP reception. In case of ACK DLLP reception, the retry buffer discards all acknowledged packets.

- ACK/NAK Packets—The ACK/NAK block handles ACK/NAK DLLPs and generates the sequence number of transmitted packets.
- Transaction Layer Packet Checker—This block checks the integrity of the received TLP and generates a request for transmission of an ACK/NAK DLLP.
- TX Arbitration—This block arbitrates transactions, prioritizing in the following order:
  - — Initialize FC Data Link Layer packet
  - — ACK/NAK DLLP (high priority)
  - — Update FC DLLP (high priority)
  - — PM DLLP
  - — Retry buffer TLP
  - — TLP
  - — Update FC DLLP (low priority)
  - — ACK/NAK FC DLLP (low priority)

2.2.3.3. Transaction Layer Overview

The following figure shows the major blocks in the R-Tile Avalon-ST IP for PCI Express Transaction Layer:

Figure 13. R-Tile Avalon-ST IP for PCI Express Transaction Layer Block Diagram



The RAS (Reliability, Availability, and Serviceability) block includes a set of features to maintain the integrity of the link.

For example: Transaction Layer inserts an optional ECRC in the transmit logic and checks it in the receive logic to provide End-to-End data protection.

When the application logic sets the TLP Digest (TD) bit in the Header of the TLP, the R-Tile Avalon-ST IP for PCIe will append the ECRC automatically.

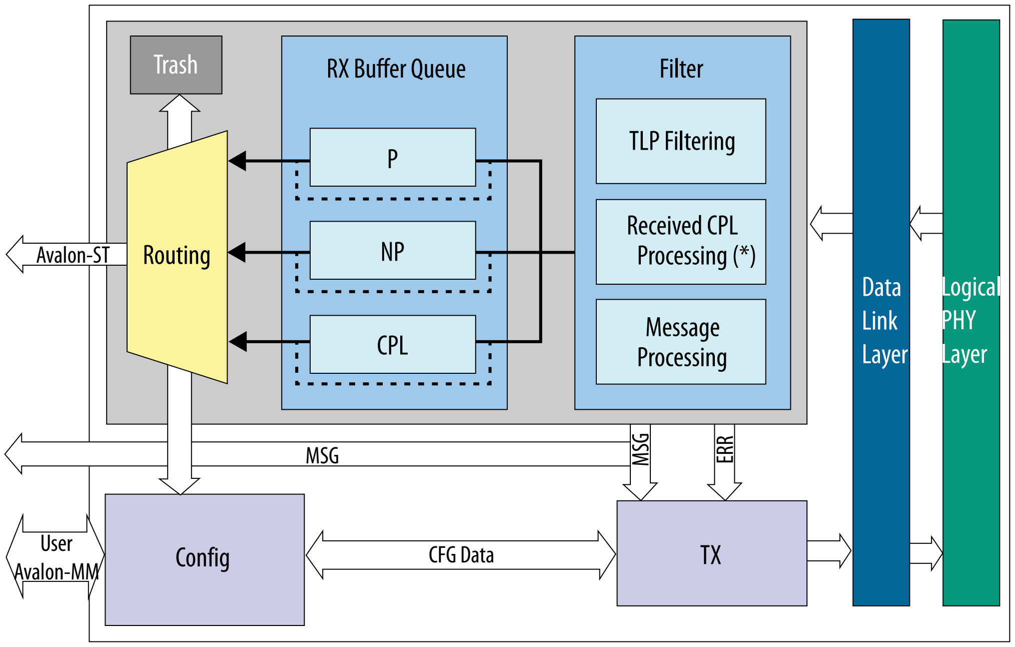
Note: In TLP Bypass mode, the PCIe Hard IP does not generate/check the ECRC and will not remove it if the received TLP has the ECRC.

The TX block sends out the TLPs that it receives as-is. It also sends the information about non-posted TLPs to the CPL Timeout Block for CPL timeout detection.

The R-Tile Avalon-ST IP for PCI Express RX block consists of two main blocks:

- Filtering block: This module checks if the TLP is good or bad and generates the associated error message and completion. It also tracks received completions and updates the completion timeout (CPL timeout) block.
- RX Buffer Queue: The R-Tile IP for PCIe has separate queues for posted/non-posted transactions and completions. This avoids head-of-queue blocking on the received TLPs and provides flexibility to extract TLPs according to the PCIe ordering rules.

Figure 14. R-Tile Avalon-ST IP for PCI Express RX Block Overview



Note: The Received CPL Processing block includes the CPL tracking mechanism.

Note: The Avalon-ST interface uses a split-bus architecture. In the x16 and x8 configurations, the 1024-bit Avalon-ST data bus consists of four segments of 256-bit data. This is done to improve the bandwidth efficiency of this interface. With this split-bus architecture, multiple TLP packets can be transmitted or received in a single clock cycle. For more details, refer to Avalon Streaming Interface.

2.2.3.3.1. Deferrable Memory Write (DMWr)

Deferrable Memory Write (DMWr) transactions are a new type of TLP supported by the PCI Specifications. This new feature allows the completer to return an acknowledgement to the requester of the DMWr transaction and provides the completer a mechanism to temporarily refuse the request. For additional details, refer to the Deferrable Memory Write (DMWr) ECN.

In the R-Tile Avalon Streaming Intel FPGA IP for PCIe, the requirements for supporting this Deferrable Memory Write feature are:

1. During the credit initialization phase between the IP and the Application logic, the Application logic needs to advertise infinite credits for Non-Posted Data (NPD) transactions to the IP. Note that the Application logic can still throttle the Non-

Posted traffic with the credits for the Non-Posted Headers (NPH). Refer to Credit Initialization for more details on the procedure required to advertise infinite credits.

2. With infinite credits being advertised between the R-Tile Avalon Streaming IP and the Application logic, the minimum size of the Application logic Rx buffer for Non-Posted Data (NPD) transactions needs to be equal to the amount of Non-Posted Header (NPH) advertised credits multiplied by 128 bytes.

Note: DMWr support is only available in the following OPNs:

- AGIx027R29AxxxxR2
- AGIx027R29AxxxxR3
- AGIx027R29BxxxxR3
- AGIx023R18AxxxxR0
- AGIx041R29DxxxxR0
- AGIx041R29DxxxxR1

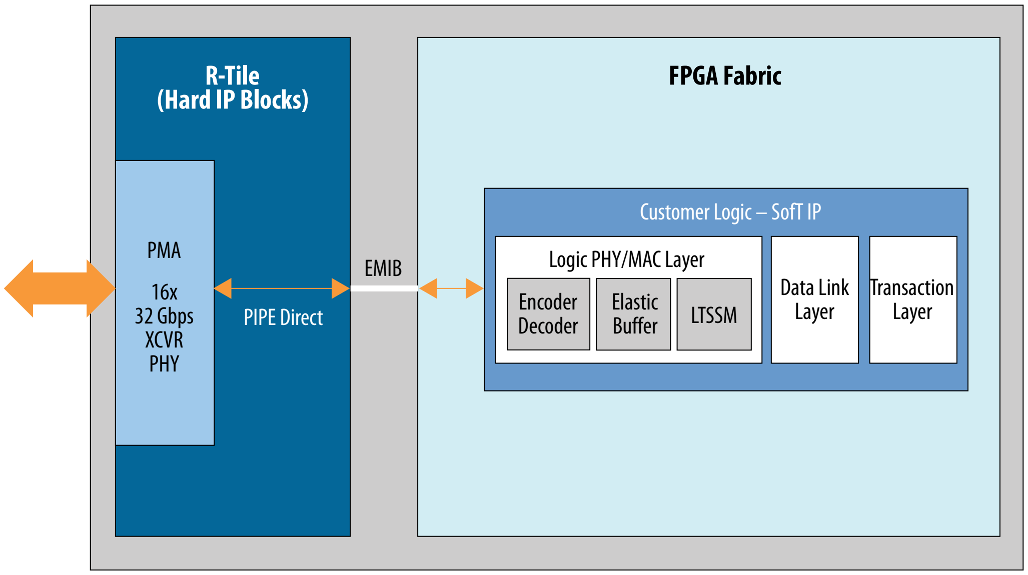
For more details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

2.3. PIPE Direct Mode

In PIPE Direct mode, application logic is responsible for implementing the Transaction Layer, Data Link Layer and the logic PHY/MAC (including the 8b/10b, 128b/130b Encoder/Decoder, Elastic Buffer, Link Training and Status State Machine (LTSSM), etc.) in your application logic in the FPGA fabric. Note that in PIPE Direct mode, R-Tile implements the SerDes Architecture mode, and the PCS responsibilities must be implemented in the Soft IP logic PHY/MAC layer. Only the PMA layer inside the R-Tile IP for PCIe is active as shown in the following figure.

Note: When implementing a Soft IP Controller using the R-Tile IP for PCI Express in PIPE Direct mode, the FPGA device must be fully configured in order to start the link training sequence since the Controller is in the FPGA fabric.

Figure 15. PIPE Direct Mode Top-Level Block Diagram



2.3.1. Clocking

In PIPE Direct mode, the logic PHY/MAC, Data Link Layer and Transaction Layer inside the R-Tile IP for PCIe are not active. The only clock domains are `pipe_direct_pld_tx_clk_out_o` and `lnX_pipe_direct_pld_rx_clk_out_o`, which are clock outputs from the R-Tile PMA layer to the FPGA fabric.

Figure 16. Clock Domains in PIPE Mode

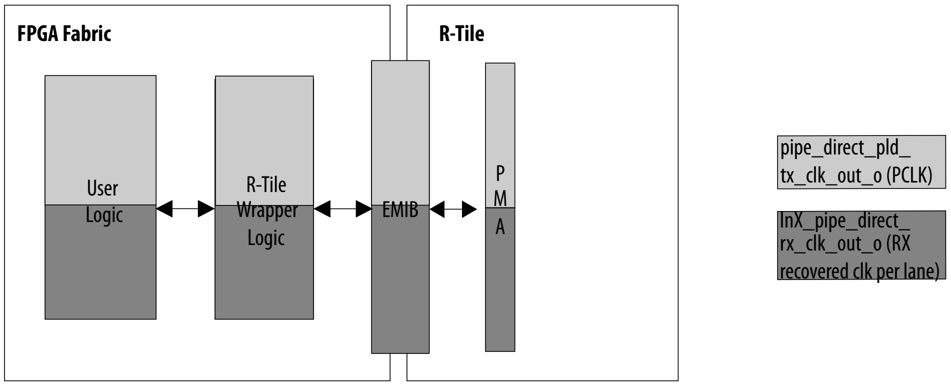

FPGA Fabric
R-Tile
User Logic
R-Tile Wrapper Logic
EMIB
P M A
pipe_direct_pld_tx_clk_out_o (PCLK)
lnX_pipe_direct_rx_clk_out_o (RX recovered clk per lane)

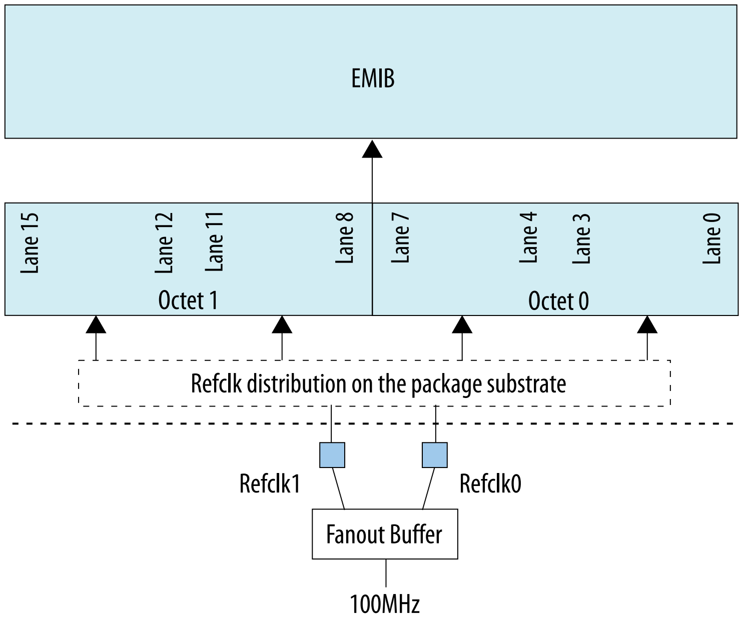
Table 11. PHY Clock and Application Clock Frequencies

Mode	PHY Clock Frequency	Application Clock Frequency
PIPE Direct	TX: 1000 MHz	TX: 500 MHz
	RX: Gen1: 250 MHz Gen2: 500 MHz Gen3: 250 MHz Gen4: 500 MHz Gen5: 1000 MHz	RX: Gen1: 125 MHz Gen2: 250 MHz Gen3: 125 MHz Gen4: 250 MHz Gen5: 500 MHz

R-Tile has two reference clock inputs at the package level, `refclk0` and `refclk1`.

You must connect a 100 MHz reference clock source to these two inputs. In PIPE Direct mode, you must drive the two refclk inputs from the same clock source. However, if Octet 1 is not used, `refclk1` can be tied to ground.

Figure 17. Using a Single 100 MHz Clock Source (for Endpoint and Root Port)



2.3.2. Reset

There is only one PERST# pin (`pin_perst_n`) on the R-Tile. Therefore, toggling `pin_perst_n` will affect the entire R-Tile. This pin also resets the EMIB interface. If the R-Tile x16 port is bifurcated into multiple ports, toggling `pin_perst_n` will affect all the ports . To reset each port individually, use the `pin_perst_n` in conjunction with the individual lane resets (`lnX_pipe_direct_pld_pcs_rst_n_i`).

The `lnX_pipe_direct_pld_pcs_rst_n_i` signals reset the per-lane TX/RX interfaces.

In case your Soft IP Controller is only using lanes 8-15, it is required to perform part of the reset sequence on lane 0 until one of the 8-15 lanes has completed its reset sequence. For further details regarding the reset sequence, refer to PIPE Direct Reset Sequence.

In PIPE mode, the per-lane `lnX_pipe_direct_pld_pcs_rst_n_i` signals must be gated by the per-lane `lnX_pipe_direct_tx_transfer_en_o` signal.

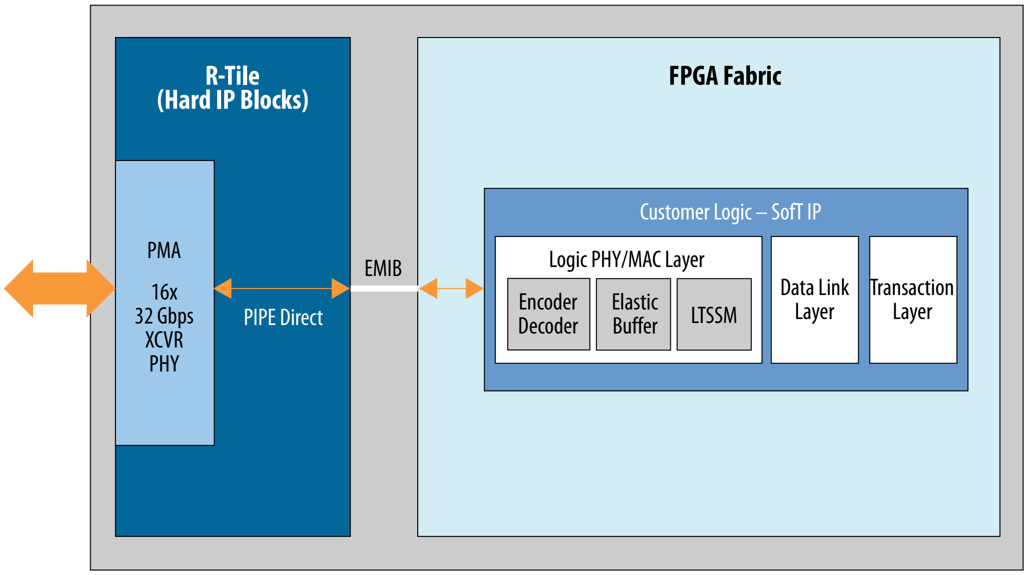
For more details, refer to PIPE Direct Mode on page 120.

2.3.3. PIPE Layer

R-Tile supports up to 16 SerDes channels through a PHY Interface for PCI Express (PIPE) v5.1.1 in SerDes Architecture mode, with 64/80 bits available to the fabric across the EMIB interface. For more details, refer to Data Signals on page 121. The R-Tile PIPE Serdes mode does not include the logic PHY/MAC layer. You must implement the logic PHY/MAC layer (including the 8b/10b, 128b/130b encoding/decoding, elastic buffer, Link Training and Status State Machine (LTSSM), etc.) in the FPGA fabric. Note that in PIPE Direct mode, R-Tile implements the SerDes Architecture mode, and the PCS responsibilities must be implemented in the Soft IP logic PHY/MAC layer.

The figure below shows the block diagram of the R-Tile in PIPE Direct mode:

Figure 18. R-Tile Avalon-ST IP for PCI Express in PIPE Direct Mode



Refer to the PIPE 5.1.1 spec for more information on the PIPE SerDes architecture.

The R-Tile Avalon-ST IP for PCI Express configured in PIPE Direct mode contains a Physical Medium Attachment (PMA) block for handling the Physical layer (PHY) packets. The PMA receives and transmits high-speed serial data on the serial lanes.

The R-Tile PMA consists of two octets. Each octet contains a pair of transmit PLLs and eight SerDes lanes capable of running up to 32 GT/s to perform the various TX and RX functions.

The Slow PLL generates the required transmit clocks for Gen1/Gen2 speeds, while the Fast PLL generates the required clocks for Gen3/Gen4/Gen5 speeds.

The PMA performs functions such as serialization/deserialization, clock data recovery, and analog front-end functions such as Continuous Time Linear Equalizer (CTLE), Decision Feedback Equalizer (DFE) and transmit equalization.

The transmitter consists of a 3-tap equalizer with one tap of pre-cursor, one tap of main cursor and one tap of post-cursor.

The receiver consists of attenuation (ATT), CTLE, Voltage gain amplifier (VGA) and a DFE blocks that are adaptive for Gen3/Gen4/Gen5 speeds. For PIPE mode, the Soft IP Controller in the application logic will perform the lane margining capability. Timing margining capabilities/parameters are as described in PMA on page 28.

3. Advanced Features

3.1. PCIe Port Bifurcation and PHY Channel Mapping

The PCIe* controller IP contains a set of port bifurcation muxes to remap the four controller PIPE lane interfaces to the shared 16 PCIe PHY lanes. The table below shows the relationship between PHY lanes and the port mapping.

Table 12. Port Bifurcation and PHY Channel Mapping

Bifurcation Mode	Port 0 (x16)	Port 1 (x8)	Port 2 (x4)	Port 3 (x4)
1 x16	0 - 15	NA	NA	NA
2 x8	0 - 7	8 - 15	NA	NA
4 x4	4 - 7	8 - 11	0 - 3	12 - 15

Note: Port 2 is only available in the following OPNs:

- AGIx027R29AxxxxR2
- AGIx027R29AxxxxR3
- AGIx027R29BxxxxR3
- AGIx023R18AxxxxR0
- AGIx041R29DxxxxR0
- AGIx041R29DxxxxR1

For additional details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

Note: For more details on the bifurcation modes, refer to PCIe Hard IP Mode on page 16.

3.2. Virtualization Support

The two components of the virtualization support of the R-Tile IP for PCIe are:

- Single-root I/O virtualization (SR-IOV)
- VirtIO

3.2.1. SR-IOV Support

The R-Tile IP for PCIe supports SR-IOV. The endpoint port controllers in the IP support up to eight physical functions (PF) and 2048 virtual functions (VF) per SR-IOV endpoint. The RP port controllers in the IP do not support SR-IOV, and only support one PF.

Note: Ports 0 and 1 can support eight PFs and 2048 VFs. Ports 2 and 3 do not support SR-IOV, and only support one PF.

The VF configuration space registers are hardened in the R-Tile. The specific VF-based work queues and interrupt tables must be implemented in the FPGA fabric by the user application.

For more details on the configuration space registers required for virtualization support, refer to R-Tile Configuration Register Maps.

3.2.1.1. SR-IOV Supported Features List

Table 13. SR-IOV Supported Features

Feature	Support
SR-IOV	Supported by the Port 0 and Port 1 controllers (in EP mode). Not supported by the Port 2 and Port 3 controllers (in RP mode).
MSI	Supported in PFs only. Not supported in VFs. No Per Vector Masking (PVM). If you need PVM, you must use MSI-X. *Note:* When SR-IOV is enabled, either MSI or MSI-X must be enabled.
MSI-X	Supported by all PFs. For SR-IOV, PFs and VFs are always MSI-X capable. *Note:* VFs share a common Table Size. VF Table BIR/Offset and PBA BIR/Offset are fixed at compile time. *Note:* When SR-IOV is enabled, either MSI or MSI-X must be enabled.
Function Level Reset (FLR)	Supported by all PFs/VFs. Required for all SR-IOV functions.
Extended Tags	Supported by all PFs/VFs. The Extended Tags feature allows the TLP Tag field to be 8-bit, thus allowing the support of 256 tags. Note that the application is restricted to a max of 256 outstanding tags, at any given time, for all functions combined. The application logic is responsible for implementing the tag generation/tracking functions. This feature is reflected in the **Extended Tag Field Supported** in the **Device Capabilities** register. By default, this field is set to 1 in every physical function enabled in the Intel FPGA R-Tile IP for PCI Express.
10-bit Tags	Supported by all PFs/VFs. x16 port supports 10-bit tag completer and requester capabilities. x8 port supports 10-bit tag completer capability.
AER	PFs are always AER capable. No AER implemented for VFs.
Atomic Ops	Requester capability is supported by all PFs/VFs. Completer capability is supported. Compare and Swap (CAS) AtomicOps are also supported. They can handle up to 128-bit operands.
Internal Error Reporting	Supported by all PFs (because all PFs are AER capable). No support for VFs (because VFs do not support AER).

continued...

Feature	Support
TLP Processing Hints	2-bit Processing Hint and 8-bit Steering Tag are supported by all PFs/VFs. TPH Prefixes are not supported. You can optionally choose to enable the TPH Requestor capability. However, the IP is always TPH Completer capable.
ID-Based ordering	Supported by all PFs/VFs. However, the IP core does NOT perform the reordering. The Application Layer must do this. The IP core only provides the IDO Request & Completion Enable bits in the Device Control 2 register. This gives the application permission to set the Attr bits in Requests and Completions that it transmits. *Note:* Reordering capability on the RX side may be limited by your bypass queue. On the TX side, the IP core does not set the IDO bits on internally generated TLPs.
Relaxed Ordering	Implemented on the RX side. This feature is always active. On the TX side, reordering is done by the application.
Alternative Routing ID Interpretation (ARI)	EP (PFs/VFs) is always ARI capable. This is a device-level option (all lanes or none will support ARI). In addition, RP will always be ARI capable (ARI Forwarding Supported bit is always 1).
Address Translation Service (ATS)	Supported by all EP PFs/VFs.
Page Request Service Interface (PRI)	Supported by all EP PFs/VFs.
User Extensions (Customer VSEC)	Supported by all PFs/VFs.
Gen3 Receiver Impedance (3.0 ECN)	Supported
Device Serial Number	Supported
Completion Timeout Ranges (Device Capabilities 2)	All ranges are supported.
Data Link Layer Active Reporting Capability (Link Capabilities)	This capability is always supported in RP mode, but not in EP mode.
Surprise Down Error Reporting Capability (Link Capabilities)	Supported
PM-PCI Power Management	Only D0 / D3 Hot states are supported.
ASPM (L0s/L1)	Not supported
Process Address Space ID (PASID)	Supported
TLP prefix	Supported, mainly for PASID
Latency Tolerance Reporting (LTR)	Supported (only for PASID)
Access Control Services	Supported

3.2.1.2. Implementation

The VF configuration space is implemented in R-Tile logic, and does not require FPGA fabric resources.

Accessing VF PCIe Information:

Due to the limited number of pins between R-Tile and the FPGA fabric, the PCIe configuration space for VFs is not directly available to the user application.

User application can use the following methods to retrieve necessary information (bus master enable, MSI-X etc…):

- Monitor specific VF registers using the Configuration Intercept Interface (for more details, refer to section Configuration Intercept Interface on page 108).
- Read/write specific VF registers using the Hard IP Reconfiguration Interface (for more details, refer to section Hard IP Reconfiguration Interface on page 103).

Accessing VF PCIe Information:

VF IDs are calculated within R-Tile. User application has sideband signals `pX_rx_st_vfnum_o[10:0]` and `pX_rx_st_vfactive_o` with the TLP to identify the associated VFs within the PFs.

BDF Assignments:

When SR-IOV is enabled, the ARI capability is always enabled.

The R-Tile IP for PCIe automatically calculates the completer/requester ID on the Transmit side.

User application needs to provide the VF and PF information in the Header as shown below:

- `pX_tx_st_hdr_sn_i[127]`: must be set to 0
- `pX_tx_st_hdr_sn_i[83]`: `pX_tx_st_vfactive_i`
- `pX_tx_st_hdr_sn_i[82:80]`: `pX_tx_st_pfnum_i[2:0]`
- `pX_tx_st_hdr_sn_i[95:84]`: `pX_tx_st_vfnum_i[10:0]`

In the following example, VF3 of PF1 is receiving and sending a request:

For the Receive TLP:

`pX_rx_st_pfnum_o` = 1h indicating that a VF associated with PF1 is making the request.

`pX_rx_st_vfnum_o` = 3h, and `pX_rx_st_vfactive_o` = 1 indicating that VF3 of PF1 is the active VF.

For the Transmit TLP of VF3 associated with PF1:

- `pX_tx_st_hdr_sn_i[83]` = 1h
- `pX_tx_st_hdr_sn_i[82:80]` = 1h
- `pX_tx_st_hdr_sn_i[95:84]` = 3h

3.2.1.2.1. VF Error Flag Interface (for x16/x8 Cores Only)

The VFs, with no AER support, are required to generate Non-Fatal error messages. The IP does not generate any error message. It is up to the user application logic to generate appropriate messages when specific error conditions occur.

The R-Tile IP for PCIe makes necessary signals available to the user application logic to generate these messages. The Completion Timeout Interface (described in section Completion Timeout Interface on page 107) and the signals listed in the table below provide the necessary information to generate Non-Fatal error messages.

Table 14. VF Error Flag Interface

Signal Name	Direction	Description	Clock Domain	EP/RP/BP
X16: vf_err_poisonedwrreq_s0/1_o X8: vf_err_poisonedwrreq_s0_o	O	Indicates a Poisoned Write Request is received.	coreclkout_hip	EP
X16: vf_err_poisonedcompl_s0/1_o X8: vf_err_poisonedcompl_s0_o	O	Indicates a Poisoned Completion is received.	coreclkout_hip	EP
X16: vf_err_ur_posted_s0/1_o X8: vf_err_ur_posted_s0_o	O	Indicates the IP core received a Posted UR request.	coreclkout_hip	EP
X16: vf_err_ca_postedreq_s0/1_o X8: vf_err_ca_postedreq_s0_o	O	Indicates the IP core received a Posted CA request.	coreclkout_hip	EP
X16: vf_err_vf_num_s0/1_o[10:0] X8: vf_err_vf_num_s0_o[10:0]	O	Indicates the VF number for which the error is detected.	coreclkout_hip	EP
X16: vf_err_func_num_s0/1_o[2:0] X8: vf_err_func_num_s0_o[2:0]	O	Indicates the physical function number associated with the VF that has the error.	coreclkout_hip	EP
vf_err_overflow_o	O	Indicates a VF error FIFO overflow and a loss of an error report. The overflow can happen when coreclkout_hip is slower than the default value. If coreclkout_hip is running at the default frequency, the overflow will not happen.	coreclkout_hip	EP

continued...

Signal Name	Direction	Description	Clock Domain	EP/RP/BP
user_sent_vfnonfatalmsg_s0_i	I	Indicates the user application sent a non-fatal error message in response to an error detected.	coreclkout_hip	EP
user_vfnonfatalmsg_vfnum_s0_i[10:0]	I	Indicates the VF number for which the error message was generated. This bus is valid when user_sent_vfnonfatalmsg_s0_i is high.	coreclkout_hip	EP
user_vfnonfatalmsg_func_num_s0_i[2:0]	I	Indicates the PF number associated with the VF with the error. This bus is valid when user_sent_vfnonfatalmsg_s0_i is high.	coreclkout_hip	EP

3.2.1.3. VF to PF Mapping

VF to PF mapping always starts from the lowest possible PF number. For instance, if the IP has 2 PFs, wherein PF0 has 64 VFs and PF1 has 16 VFs, VF1 to VF64 are mapped to PF0, and VF65 to VF80 are mapped to PF1.

Currently, the IP core only supports the following PF/VF combinations:

Table 15. Supported PF/VF Combinations

Number of PFs	Number of VFs per PF (PF0/PF1/PF2/PF3/PF4/PF5/PF6/PF7)	Total VFs
1	8	8
1	16	16
1	32	32
1	64	64
1	128	128
1	256	256
1	512	512
2	16/16	32
2	32/32	64
2	128/128	256
2	256/256	512
2	32/0	32
2	0/32	32
2	64/0	64
2	0/64	64
2	128/0	128
2	0/128	128
2	256/0	256
2	0/256	256

continued...

Number of PFs	Number of VFs per PF (PF0/PF1/PF2/PF3/PF4/PF5/PF6/PF7)	Total VFs
2	512/0	512
2	0/512	512
2	1024/0	1024
2	0/1024	1024
2	2048/0	2048
2	0/2048	2048
4	128/0/0/0	128
4	0/128/0/0	128
4	256/0/0/0	256
4	0/256/0/0	256
4	1024/0/0/0/0	1024
4	0/1024/0/0	1024
8	256/0/0/0/0/0/0/0	256
8	0/256/0/0/0/0/0/0	256

For example, the row that shows the combination of four PFs, 256 VFs, and the notation `256/0/0/0` in the `Number of VFs per PF` column indicates that all 256 VFs are mapped to PF0, while no VF is mapped to PF1, PF2 or PF3.

Note: SR-IOV permutations allow any PF to be assigned the initial VF allocation.

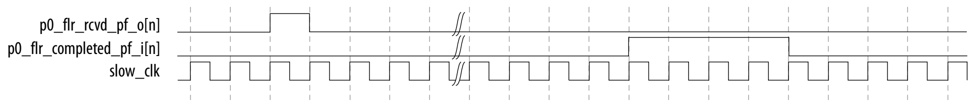
3.2.1.4. Function Level Reset (FLR)

Use the FLR interface to reset individual SR-IOV functions. The PCIe Hard IP supports FLR for both PFs and VFs. If the FLR is for a specific VF, the received packets for that VF are no longer valid. The flr_* interface signals are provided to the application interface for this purpose. When the flr_rcvd* signal is asserted, it indicates that an FLR is received for a particular PF/VF. Application logic needs to perform its FLR routine and send the completion status back on the flr_completed* interface. The Hard IP will wait for the flr_completed* status to re-enable the VF. Prior to that event, the Hard IP will respond to all transactions to the function that is reset by the FLR with completions with an Unsupported Request (UR) status.

Note: Only Ports 0 and 1 support FLR.

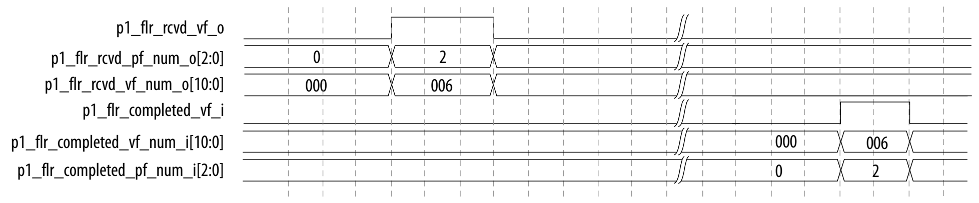
The following figure shows the timing diagram for an FLR event targeting a PF (PF[n] in this example):

Figure 19. FLR for PF



Here is the timing diagram for an FLR event targeting a VF:

Figure 20. FLR for VF


p1_flr_rcvd_vf_o
p1_flr_rcvd_pf_num_o[2:0]
p1_flr_rcvd_vf_num_o[10:0]
p1_flr_completed_vf_i
p1_flr_completed_vf_num_i[10:0]
p1_flr_completed_pf_num_i[2:0]
0
2
000
006
000
006
0
2

3.2.2. VirtIO Support

3.2.2.1. VirtIO Supported Features List

- VirtIO devices are implemented as PCI Express devices.
- Support 8 PFs and 2K VFs VirtIO capability structure.
    - For Port 2 and Port 3, VirtIO is only supported in the following OPNs:
        - AGIx027R29AxxxxR2
        - AGIx027R29AxxxxR3
        - AGIx027R29BxxxxR3
        - AGIx023R18AxxxxR0
        - AGIx041R29DxxxxR0
        - AGIx041R29DxxxxR1

      For additional details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.
- Configuration Intercept Interface in the R-Tile IP for PCIe (EP mode only) is provided for VirtIO transport.
- Five VirtIO device configuration structures are supported:
    - Common configuration
    - Notifications
    - ISR Status
    - Device-specific configuration (optional)
    - PCI configuration access
- Location of each structure is specified using a vendor-specific PCI capability located in the PCI configuration space of the device.
- VirtIO capability structure uses little-endian format.
- All fields of the VirtIO capability structure are read-only for the driver by default.

- Support PFs and VFs FLR

  Note: The Read/Write registers in the VF and PF VirtIO capability registers are not reset by FLR.
- Supports x16 and x8 cores.
- MSI is not supported with VirtIO.

  Note: In the 22.1 release of Intel Quartus Prime, when you enable the VirtIO feature, all the PFs in the Hard IP must support VirtIO. The configuration where some PFs support VirtIO and some PFs support MSI is not available.

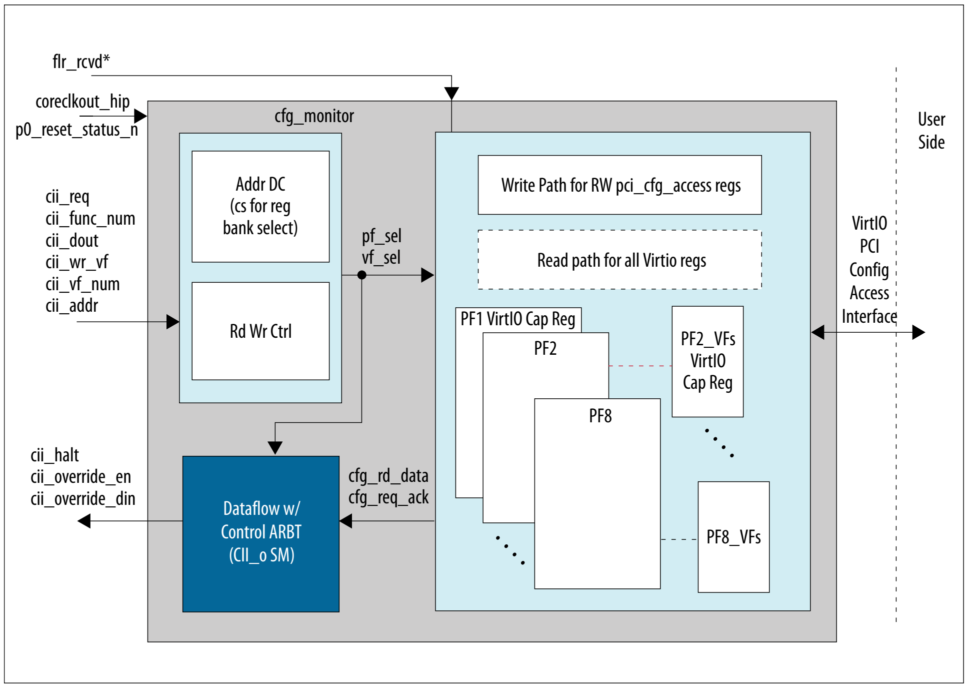
3.2.2.2. Overview

The VirtIO PCI configuration access capability creates an alternative access method to the common configuration, notifications, ISR, and device-specific configuration structure regions. This interface provides a means for the driver to access the VirtIO device region of Physical Functions (PFs) or Virtual Functions (VFs).

VirtIO is an industry standard for software-based virtualization that is supported natively by Linux*. In VirtIO, software implements the virtualization stack, whereas in the case of SR-IOV, this stack is implemented mostly in hardware.

Below is the block diagram of the Soft IP which implements the VirtIO capability for PFs and VFs. This Soft IP block is automatically included when the VirtIO feature is enabled in the IP Parameter Editor.

Figure 21. VirtIO Soft IP Block Diagram



3.2.2.3. Parameters

For a detailed discussion of the VirtIO-related parameters, refer to the section VirtIO Parameters on page 146.

3.2.2.4. VirtIO PCI Configuration Access Interface

To access a VirtIO device region, pci_cfg_data will provide a window of size cap.length (1, 2 or 4 Bytes) into the given cap.bar (0x0 – 0x5) at offset cap.offset (multiple of cap.length). Detailed interfaces mapping for the user application logic are shown in the following table.

As for the VirtIO device, upon detecting a driver write access to pci_cfg_data, the user application side's VirtIO device must execute a write access at cap.offset at the BAR selected by cap.bar using the first cap.length bytes from pci_cfg_data. Moreover, upon detecting a driver read access to pci_cfg_data, the user application side's VirtIO device must execute a read access of length cap.length at cap.offset at the BAR selected by cap.bar and store the first cap.length bytes in pci_cfg_data.

Table 16. VirtIO PCI Configuration Access Interface

Name	Direction	Description	Clock Domain
`pX_virtio_pcicfg_vfaccess_o` where X = 0, 1	O	Indicates the driver access is to a VF. The corresponding Virtual Function is identified from the value of `virtio_pcicfg_vfnum_o`.	`slow_clk`
`pX_virtio_pcicfg_vfnum_o[VFNUM_WIDTH-1:0]` where X = 0, 1	O	Indicates the corresponding Virtual Function number associated with the current Physical Function that the driver's write or read access is targeting. Validated by `virtio_pcicfg_vfaccess_o` and by driver write access to pci_cfg_data, or driver read access to pci_cfg_data.	`slow_clk`
`pX_virtio_pcicfg_pfnum_o[PFNUM_WIDTH-1:0]` where X = 0, 1	O	Indicates the corresponding Physical Function number that the driver's write or read access is targeting. Validated by driver write access to pci_cfg_data, or driver read access to pci_cfg_data.	`slow_clk`
`pX_virtio_pcicfg_bar_o[7:0]` where X = 0, 1	O	Indicates the BAR holding the PCI configuration access structure. The driver sets the BAR to access by writing to cap.bar. Values 0x0 to 0x5 specify a BAR belonging to the function located beginning at 10h in the PCI Configuration Space. The BAR can be either 32-bit or 64-bit. Validated by driver write access to pci_cfg_data, or driver read access to pci_cfg_data. The corresponding PF or VF is identified from the value of `virtio_pcicfg_p/vfnum_o`.	`slow_clk`
`pX_virtio_pcicfg_length_o[31:0]` where X = 0, 1	O	Indicates the length of the structure. The length may include padding, or fields unused by the driver, or future extensions. The driver sets the size of the access by writing 1, 2 or 4 to cap.length. Validated by driver write access to pci_cfg_data, or driver read access to pci_cfg_data. The corresponding PF or VF is identified from the value of `virtio_pcicfg_p/vfnum_o`.	`slow_clk`

continued...

Name	Direction	Description	Clock Domain
pX_virtio_pcicfg_baroffset_o[31:0] where X = 0, 1	O	Indicates where the structure begins relative to the base address associated with the BAR. The driver sets the offset within the BAR by writing to cap.offset. Validated by driver write access to pci_cfg_data, or driver read access to pci_cfg_data. The corresponding PF or VF is identified from the value of virtio_pcicfg_p/vfnum_o.	slow_clk
pX_virtio_pcicfg_cfgdata_o[31:0] where X = 0, 1	O	Indicates the data for BAR access. The pci_cfg_data will provide a window of size cap.length into the given cap.bar at offset cap.offset. Validated by driver write access to pci_cfg_data, or driver read access to pci_cfg_data. The corresponding PF or VF is identified from the value of virtio_pcicfg_p/vfnum_o.	slow_clk
pX_virtio_pcicfg_cfgwr_o where X = 0, 1	O	Indicates driver write access to pci_cfg_data. The corresponding PF or VF is identified from the value of virtio_pcicfg_p/vfnum_o.	slow_clk
pX_virtio_pcicfg_cfgrd_o where X = 0, 1	O	Indicates driver read access to pci_cfg_data. The corresponding PF or VF is identified from the value of virtio_pcicfg_p/vfnum_o.	slow_clk
pX_virtio_pcicfg_appvfnum_i[VFNUM_WIDTH-1:0] where X = 0, 1	I	Indicates the corresponding Virtual Function number associated with the current Physical Function that the application config data storage is for. Validated by virtio_pcicfg_rdack_i.	slow_clk
pX_virtio_pcicfg_apppfnum_i[PFNUM_WIDTH-1:0] where X = 0, 1	I	Indicates the corresponding Physical Function number that the application config data storage is for. Validated by virtio_pcicfg_rdack_i.	slow_clk
pX_virtio_pcicfg_rdack_i where X = 0, 1	I	Indicates an application read access data ack to store the config data in pci_cfg_data. Usually the reasonable ack latency is no more than 10 cycles. The corresponding Virtual Function is identified from the value of virtio_pcicfg_appvfnum_i.	slow_clk
pX_virtio_pcicfg_rdbe_i[3:0] where X = 0, 1	I	Indicates application enabled bytes within virtio_pcicfg_data_i. Validated by virtio_pcicfg_rdack_i. The corresponding Virtual Function is identified from the value of virtio_pcicfg_appvfnum_i. For the current implementation of the Hard IP, these signals are tied high.	slow_clk
pX_virtio_pcicfg_data_i[31:0] where X = 0, 1	I	Indicates application data to be stored in PCI Configuration Access data registers. Validated by virtio_pcicfg_rdack_i and virtio_pcicfg_rdbe_i.	slow_clk

Name	Direction	Description	Clock Domain
		The corresponding Virtual Function is identified from the value of `virtio_pcicfg_appvfnum_i`.	

3.2.2.5. Registers

The following VirtIO capability structure registers references apply to each PF and VF. Addresses shown are register addresses.

Table 17. PF/VF Capability Link List

Capability	Start Byte Address	Last Byte Address	DW Count
Type0	0x00	0x3F	16
PM (PF only)	0x40	0x47	2
Reserved	0x48	0x4F	2
VirtIO Common Configuration	0x50	0x5F	4
VirtIO ISR Configuration	0x60	0x6F	4
PCIe	0x70	0xAB	15
Reserved	0xAC	0xAF	1
MSIX	0xB0	0xBB	3
Reserved	0xBC	0xBF	1
VirtIO Notify Configuration	0xC0	0xD3	5
VirtIO Device-Specific Configuration	0xD4	0xE3	4
VirtIO PCI Configuration Access	0xE4	0xF7	5
Reserved	0xF8	0xFF	2

Table 18. VirtIO Common Configuration Capability Structure

Address	Name	Description
014	Common Configuration Capability Register	Capability ID, next capability pointer, capability length
015	BAR Indicator Register	Lower 8 bits indicate which BAR holds the structure
016	BAR Offset Register	Indicates starting address of the structure within the BAR
017	Structure Length Register	Indicates length of structure
VirtIO Notifications Capability Structure		
030	Notifications Capability Register	Capability ID, next capability pointer, capability length
031	BAR Indicator Register	Lower 8 bits indicate which BAR holds the structure
032	BAR Offset Register	Indicates starting address of the structure within the BAR
033	Structure Length Register	Indicates length of structure
034	Notify Off Multiplier	Multiplier for queue_notify_off

continued...

Address	Name	Description
VirtIO ISR Status Capability Structure		
018	ISR Status Capability Register	Capability ID, next capability pointer, capability length
019	BAR Indicator Register	Lower 8 bits indicate which BAR holds the structure
020	BAR Offset Register	Indicates starting address of the structure within the BAR
021	Structure Length Register	Indicates length of structure
VirtIO Device-Specific Capability Structure (Optional)		
035	Device Specific Capability Register	Capability ID, next capability pointer, capability length
036	BAR Indicator Register	Lower 8 bits indicate which BAR holds the structure
037	BAR Offset Register	Indicates starting address of the structure within the BAR
038	Structure Length Register	Indicates length of structure
VirtIO PCI Configuration Access Structure		
039	PCI Configuration Access Capability Register	Capability ID, next capability pointer, capability length
040	BAR Indicator Register	Lower 8 bits indicate which BAR holds the structure
041	BAR Offset Register	Indicates starting address of the structure within the BAR
042	Structure Length Register	Indicates length of structure
043	PCI Configuration Data	Data for BAR access

3.2.2.5.1. VirtIO Common Configuration Capability Register (Address: 0x012)

The capability register identifies that this is a vendor-specific capability. It also identifies the structure type.

Table 19. VirtIO Common Configuration Capability Register

Bit Location	Description	Access Type	Default Value
31:24	Configuration Type	RO	0x01
23:16	Capability Length	RO	0x10
15:8	Next Capability Pointer	RO	0x58
7:0	Capability ID	RO	0x09

3.2.2.5.2. VirtIO Common Configuration BAR Indicator Register (Address: 0x013)

The Bar Indicator field holds the values 0x0 to 0x5 specifying a Base Address register (BAR) belonging to the function located beginning at 10h in PCI Configuration Space. The BAR is used to map the structure into the memory space. Any other value is reserved for future use.

Table 20. VirtIO Common Configuration BAR Indicator Register

Bit Location	Description	Access Type	Default Value
31:24	Padding	RO	0x00
23:16	Padding	RO	0x00
15:8	Padding	RO	0x00
7:0	BAR Indicator	RO	Settable through the IP Parameter Editor

3.2.2.5.3. VirtIO Common Configuration BAR Offset Register (Address: 0x014)

This register indicates where the structure begins relative to the base address associated with the BAR. The alignment requirements of the offset are indicated in each structure-specific section.

Table 21. VirtIO Common Configuration BAR Offset Register

Bit Location	Description	Access Type	Default Value
31:0	BAR Offset	RO	Settable through the IP Parameter Editor

3.2.2.5.4. VirtIO Common Configuration Structure Length Register (Address 0x015)

The length register indicates the length of the structure. The length may include padding, fields unused by the driver, or future extensions.

Table 22. VirtIO Common Configuration Structure Length Register

Bit Location	Description	Access Type	Default Value
31:0	Structure Length	RO	Settable through the IP Parameter Editor

3.2.2.5.5. VirtIO Notifications Capability Register (Address: 0x016)

The capability register identifies that this is a vendor-specific capability. It also identifies the structure type.

Table 23. VirtIO Notifications Capability Register

Bit Location	Description	Access Type	Default Value
31:24	Configuration Type	RO	0x02
23:16	Capability Length	RO	0x14
15:8	Next Capability Pointer	RO	0xBC
7:0	Capability ID	RO	0x09

3.2.2.5.6. VirtIO Notifications BAR Indicator Register (Address: 0x017)

The Bar Indicator field holds the values 0x0 to 0x5 specifying a Base Address register (BAR) belonging to the function located beginning at 10h in PCI Configuration Space. The BAR is used to map the structure into memory space. Any other value is reserved for future use.

Table 24. VirtIO Notifications BAR Indicator Register

Bit Location	Description	Access Type	Default Value
31:24	Padding	RO	0x00
23:16	Padding	RO	0x00
15:8	Padding	RO	0x00
7:0	BAR Indicator	RO	Settable through the IP Parameter Editor

3.2.2.5.7. VirtIO Notifications BAR Offset Register (Address: 0x018)

This register indicates where the structure begins relative to the base address associated with the BAR. The alignment requirements of the offset are indicated in each structure-specific section.

Table 25. VirtIO Notifications BAR Offset Register

Bit Location	Description	Access Type	Default Value
31:0	BAR Offset	RO	Settable through the IP Parameter Editor

3.2.2.5.8. VirtIO Notifications Structure Length Register (Address: 0x019)

The length register indicates the length of the structure. The length may include padding, fields unused by the driver, or future extensions.

Table 26. VirtIO Notifications Structure Length Register

Bit Location	Description	Access Type	Default Value
31:0	Structure Length	RO	Settable through the IP Parameter Editor

3.2.2.5.9. VirtIO Notifications Notify Off Multiplier Register (Address: 0x01A)

The notify off multiplier register indicates the multiplier for queue_notify_off in the structure.

Table 27. VirtIO Notifications Notify Off Multiplier Register

Bit Location	Description	Access Type	Default Value
31:0	Multiplier for queue_notify_off	RO	Settable through the IP Parameter Editor

3.2.2.5.10. VirtIO ISR Status Capability Register (Address: 0x02F)

The capability register identifies that this is a vendor-specific capability. It also identifies the structure type.

Table 28. VirtIO ISR Status Capability Register

Bit Location	Description	Access Type	Default Value
31:24	Configuration Type	RO	0x03
23:16	Capability Length	RO	0x10
15:8	Next Capability Pointer	RO	If Device-Specific Capability is present, then points to 0xCC, else points to 0xDC.
7:0	Capability ID	RO	0x09

3.2.2.5.11. VirtIO ISR Status BAR Indicator Register (Address: 0x030)

The Bar Indicator field holds the values 0x0 to 0x5 specifying a Base Address register (BAR) belonging to the function located beginning at 10h in PCI Configuration Space. The BAR is used to map the structure into memory space. Any other value is reserved for future use.

Table 29. VirtIO ISR Status BAR Indicator Register

Bit Location	Description	Access Type	Default Value
31:24	Padding	RO	0x00
23:16	Padding	RO	0x00
15:8	Padding	RO	0x00
7:0	BAR Indicator	RO	Settable through the IP Parameter Editor

3.2.2.5.12. VirtIO ISR Status BAR Offset Register (Address: 0x031)

This register indicates where the structure begins relative to the base address associated with the BAR. The alignment requirements of the offset are indicated in each structure-specific section.

Table 30. VirtIO ISR Status BAR Offset Register

Bit Location	Description	Access Type	Default Value
31:0	BAR Offset	RO	Settable through the IP Parameter Editor

3.2.2.5.13. VirtIO ISR Status Structure Length Register (Address: 0x032)

The length register indicates the length of the structure. The length may include padding, fields unused by the driver, or future extensions.

Table 31. VirtIO ISR Status Structure Length Register

Bit Location	Description	Access Type	Default Value
31:0	Structure Length	RO	Settable through the IP Parameter Editor

3.2.2.5.14. VirtIO Device Specific Capability Register (Address: 0x033)

The capability register identifies that this is vendor-specific capability. It also identifies the structure type.

Table 32. VirtIO Device Specific Capability Register

Bit Location	Description	Access Type	Default Value
31:24	Configuration Type	RO	0x04
23:16	Capability Length	RO	0x10
15:8	Next Capability Pointer	RO	If this capability is present, then points to 0xDC.
7:0	Capability ID	RO	0x09

3.2.2.5.15. VirtIO Device Specific BAR Indicator Register (Address: 0x034)

The BAR Indicator field holds the values 0x0 to 0x5 specifying a Base Address register (BAR) belonging to the function located beginning at 10h in PCI Configuration Space. The BAR is used to map the structure into memory space. Any other value is reserved for future use.

Table 33. VirtIO Device Specific BAR Indicator Register

Bit Location	Description	Access Type	Default Value
31:24	Padding	RO	0x00
23:16	Padding	RO	0x00
15:8	Padding	RO	0x00
7:0	BAR Indicator	RO	Settable through the IP Parameter Editor

3.2.2.5.16. VirtIO Device Specific BAR Offset Register (Address 0x035)

This register indicates where the structure begins relative to the base address associated with the BAR. The alignment requirements of the offset are indicated in each structure-specific section.

Table 34. VirtIO Device Specific BAR Offset Register

Bit Location	Description	Access Type	Default Value
31:0	BAR Offset	RO	Settable through the IP Parameter Editor

3.2.2.5.17. VirtIO Device Specific Structure Length Register (Address: 0x036)

The length register indicates the length of the structure. The length may include padding, fields unused by the driver, or future extensions.

Table 35. VirtIO Device Specific Structure Length Register

Bit Location	Description	Access Type	Default Value
31:0	Structure Length	RO	Settable through the IP Parameter Editor

3.2.2.5.18. VirtIO PCI Configuration Access Capability Register (Address: 0x037)

The capability register identifies that this is a vendor-specific capability. It also identifies the structure type.

Table 36. VirtIO PCI Configuration Access Capability Register

Bit Location	Description	Access Type	Default Value
31:24	Configuration Type	RO	0x05
23:16	Capability Length	RO	0x14
15:8	Next Capability Pointer	RO	0x00
7:0	Capability ID	RO	0x09

3.2.2.5.19. VirtIO PCI Configuration Access BAR Indicator Register (Address: 0x038)

The BAR Indicator field holds the values 0x0 to 0x5 specifying a Base Address register (BAR) belonging to the function located beginning at 10h in PCI Configuration Space. The BAR is used to map the structure into memory space. Any other value is reserved for future use.

Table 37. VirtIO PCI Configuration Access BAR Indicator Register

Bit Location	Description	Access Type	Default Value
31:24	Padding	RO	0x00
23:16	Padding	RO	0x00
15:8	Padding	RO	0x00
7:0	BAR Indicator	RW	Undefined

3.2.2.5.20. VirtIO PCI Configuration Access BAR Offset Register (Address: 0x039)

This register indicates where the structure begins relative to the base address associated with the BAR. The alignment requirements of the offset are indicated in each structure-specific section.

Table 38. VirtIO PCI Configuration Access BAR Offset Register

Bit Location	Description	Access Type	Default Value
31:0	BAR Offset	RW	Undefined

3.2.2.5.21. VirtIO PCI Configuration Access Structure Length Register (Address: 0x03A)

The length register indicates the length of the structure. The length may include padding, fields unused by the driver, or future extensions.

Table 39. VirtIO PCI Configuration Access Structure Length Register

Bit Location	Description	Access Type	Default Value
31:0	Structure Length	RW	Undefined

3.2.2.5.22. VirtIO PCI Configuration Access Data Register (Address: 0x03B)

The PCI configuration data register indicates the data for BAR access.

Table 40. VirtIO PCI Configuration Access Data Register

Bit Location	Description	Access Type	Default Value
31:0	PCI Configuration Data	RW	Undefined

3.3. TLP Bypass Mode

The R-Tile Avalon-ST IP for PCIe includes a TLP Bypass mode for both downstream and upstream ports to allow the implementation of advanced features such as:

- The upstream port or the downstream port of a switch.
- A custom implementation of a Transaction Layer to meet specific user requirements.

Table 41. Supported TLP Bypass Configurations

UP = upstream port; DN = downstream port

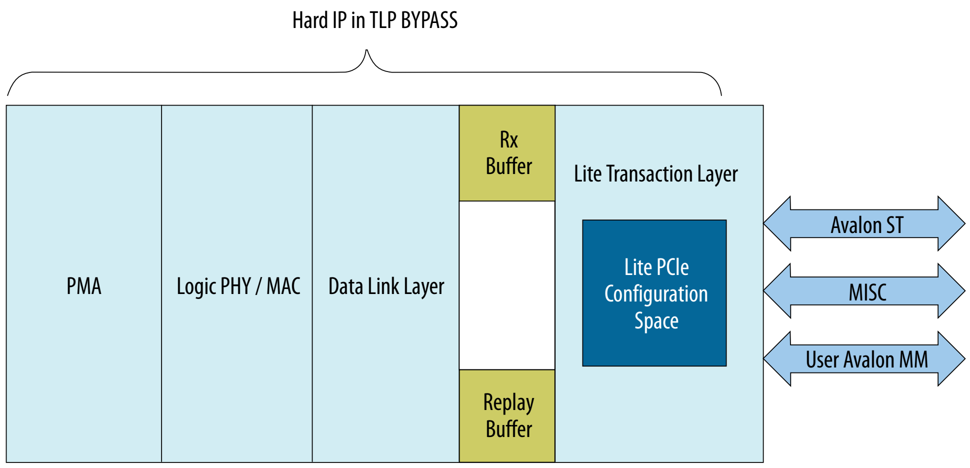
IP Mode	Port Mode
X16	UP DN
X8/X8	UP/UP UP/DN DN/UP DN/DN
X4/X4/X4/X4	UP/UP/UP/UP DN/DN/DN/DN

3.3.1. Overview

When the TLP Bypass feature is enabled, the R-Tile Avalon-ST IP does not process received TLPs internally but outputs them to the user application (this includes the processing of Configuration TLPs, which are forwarded to the application logic). This allows the application to implement a custom Transaction Layer.

When in TLP Bypass mode, the same Avalon Streaming interface is used. This includes the regular Tx and Rx interfaces along with their corresponding flow control interfaces for the credit handling. In addition, although most of the transaction layer is bypassed, there is a Lite Transaction Layer still active which interfaces with the application logic through the Hard IP Reconfiguration Interface on page 103 to access the set of PCIe registers related to link operation that continues to be implemented on the R-Tile Avalon Streaming IP. This set of registers is referred to in the figure below as the Lite PCIe Configuration Space. For details on these registers, refer to Hard IP Reconfiguration Interface.

Figure 22. R-Tile Avalon-ST IP in TLP Bypass Mode



In TLP bypass mode, R-Tile supports the autonomous Hard IP feature. It responds to configuration accesses before the FPGA fabric enters user mode with Completions with a CRS code.

Note that in TLP Bypass mode, the PCIe Hard IP does not generate/check the ECRC. However, you can enable the IP to remove it if the received TLP has the ECRC. The steps on how to do this are described in section ECRC on page 61.

Note: In TLP bypass mode, CvP init and update are not supported.

3.3.2. Register Settings for the TLP Bypass Mode

When TLP Bypass mode is enabled, some error detection is still performed in the Physical and Link Layers inside the Hard IP. Per PCIe specification, the Hard IP must report these errors on the configuration space registers (in the AER Capability Structure). The R-Tile IP for PCIe includes two registers called TLPBYPASS_ERR_EN and TLPBYPASS_ERR_STATUS to report errors detected while in TLP Bypass mode. You must use the Hard IP Reconfiguration Interface to access these registers. For more details on the signals in this interface, refer to Hard IP Reconfiguration Interface.

TLPBYPASS_ERR_EN and TLPBYPASS_ERR_STATUS are part of the configuration and status register.

3.3.2.1. TLPBYPASS_ERR_EN (Address 0x1314)

This register allows you to enable or disable error reporting. When this feature is disabled, the TLPBYPASS_ERR_ STATUS bit associated with an error is not set when the error is detected.

Table 42. TLPBYPASS_ERR_EN Register (Address 0x1314)

Name	Bits	Reset Value	Access Mode	Register Description
Reserved	[31:20]	12’b0	RO	Reserved
`k_cfg_uncor_internal_err_sts_en`	[19]	1'b1	RW	Enable error indication on `serr_out_o` for Uncorrectable Internal Error.

continued...

Name	Bits	Reset Value	Access Mode	Register Description
k_cfg_corrected_internal_err_sts_en	[18]	1'b1	RW	Enable error indication on serr_out_o for Corrected Internal Error.
k_cfg_rcvr_overflow_err_sts_en	[17]	1'b1	RW	Enable error indication on serr_out_o for Receiver Overflow Error.
k_cfg_fc_protocol_err_sts_en	[16]	1'b1	RW	Enable error indication on serr_out_o for Flow Control Protocol Error.
k_cfg_mlf_tlp_err_sts_en	[15]	1'b1	RW	Enable error indication on serr_out_o for Malformed TLP Error.
k_cfg_surprise_down_err_sts_en	[14]	1'b1	RW	Enable error indication on serr_out_o for Surprise Down Error.
k_cfg_dl_protocol_err_sts_en	[13]	1'b1	RW	Enable error indication on serr_out_o for Data Link Protocol Error.
k_cfg_replay_number_rollover_err_sts_en	[12]	1'b1	RW	Enable error indication on serr_out_o for REPLAY_NUM Rollover Error.
k_cfg_replay_timer_timeout_err_st_en	[11]	1'b1	RW	Enable error indication on serr_out_o for Replay Timer Timeout Error.
k_cfg_bad_dllp_err_sts_en	[10]	1'b1	RW	Enable error indication on serr_out_o for Bad DLLP Error.
k_cfg_bad_tlp_err_sts_en	[9]	1'b1	RW	Enable error indication on serr_out_o for Bad TLP Error.
k_cfg_rcvr_err_sts_en	[8]	1'b1	RW	Enable error indication on serr_out_o for Receiver Error.
Reserved	[7:1]	7'b0	RO	Reserved
k_cfg_ecrc_err_sts_en	[0]	1'b1	RW	Enable error indication on serr_out_o for ECRC Error.

3.3.2.2. TLPBYPASS_ERR_STATUS (Address 0x1310)

When an error is detected, Intel recommends that you read the PF0 AER register inside R-Tile to get detailed information about the error. To clear the previous error status, you need to clear TLPBYPASS_ERR_STATUS and the corresponding correctable and uncorrectable error status registers in the AER capability structure. After doing that, you can get the new error update from this register.

Table 43. TLPBYPASS_ERR_STATUS Register (Address 0x1310)

Name	Bits	Reset Value	Access Mode	Register Description
Reserved	[31:20]	12’b0	RO	Reserved
cfg_uncor_internal_err_sts	[19]	1'b0	W1C	Uncorrectable Internal Error
cfg_corrected_internal_err_sts	[18]	1'b0	W1C	Corrected Internal Error
cfg_rcvr_overflow_err_sts	[17]	1'b0	W1C	Receiver Overflow Error
cfg_fc_protocol_err_sts	[16]	1'b0	W1C	Flow Control Protocol Error

continued...

Send Feedback

Name	Bits	Reset Value	Access Mode	Register Description
`cfg_mlf_tlp_err_sts`	[15]	1'b0	W1C	Malformed TLP Error
`cfg_surprise_down_err_sts`	[14]	1'b0	W1C	Surprise Down Error. Available in downstream mode only.
`cfg_dl_protocol_err_sts`	[13]	1'b0	W1C	Data Link Protocol Error
`cfg_replay_number_rollover_err_sts`	[12]	1'b0	W1C	REPLAY_NUM Rollover Error
`cfg_replay_timer_timeout_err_sts`	[11]	1'b0	W1C	Replay Timer Timeout Error
`cfg_bad_dllp_err_sts`	[10]	1'b0	W1C	Bad DLLP Error
`cfg_bad_tlp_err_sts`	[9]	1'b0	W1C	Bad TLP Error
`cfg_rcvr_err_sts`	[8]	1'b0	W1C	Receiver Error
Reserved	[7:1]	7'b0	RO	Reserved
`cfg_ecrc_err_sts`	[0]	1'b0	W1C	ECRC Error

3.3.3. Hard IP Reconfiguration Interface

For more details on the signals in this interface, refer to the section Hard IP Reconfiguration Interface on page 103.

The majority of the PCIe standard capability structures are implemented in the application logic outside of the R-Tile Avalon-ST IP.

However, the following PCIe capability structures are still implemented inside the R-Tile Avalon Streaming IP:

- Power management capability structure
- A portion of the PCI Express capability structure is implemented inside R-Tile. However, the following registers within this structure still need to be implemented by the application logic:
  - PCI Express Capability List register
  - PCI Express Capabilities register
  - Device Capabilities register
  - Device Control register
  - Device Status register
- Secondary PCI Express extended capability structure
- Data link feature extended capability structure
- Physical layer 16.0 GT/s and 32.0 GT/s extended capabilities structures
- Lane margining at the receiver extended capability structure
- Advanced error reporting extended capability structure

The application can only access PCIe controller registers through the Hard IP Reconfiguration interface.

Table 44. Capability Registers to be Updated by the Application Logic via the Hard IP Reconfiguration Interface

Capability	Comments
Power Management Capability	Need to write back since it is required to trigger a PCI-PM entry.
PCI Express Capability	All the PCIe capabilities, control and status registers are for configuring the device. Write-back is required.
Secondary PCI Express Capability	Secondary PCIe Capability is required for configuring the device.
Data Link Feature Extended Capability	Data Link Capability is device specific.
Physical Layer 16.0 GT/s Extended Capability	Physical Layer 16G Capability is device specific.
Lane Margining at the Receiver Extended Capability	Margining Extended Capability is device specific.
Advanced Error Reporting Capability	Write-back to error status registers is required for TLP Bypass.

3.3.4. Avalon Streaming Interface

For more details on the signals in this interface, refer to Avalon Streaming Interface on page 67.

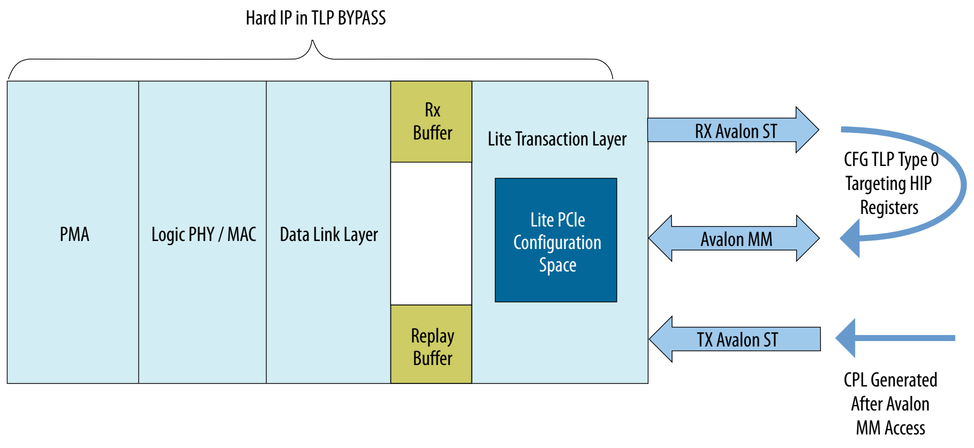
3.3.4.1. Configuration TLP

The R-Tile IP forwards any received Type0/1 Configuration TLP to the Avalon-ST RX streaming interface. Application logic has the responsibility to respond with a Completion TLP with a Completion code of Successful Completion (SC), Unsupported Request (UR), Configuration Request Retry Status (CRS), or Completer Abort (CA).

If a Configuration TLP needs to update a register in the Lite PCIe configuration space in the R-Tile PCIe Hard IP (shown in the figure below), you need to use the Hard IP Reconfiguration Interface.

The application logic needs to prevent link programming side effects such as writing into low-power states before sending the Completion associated with the request. The application logic can check the `pX_tx_ehp_deallocate_empty_o` signal after the Completion enters the TX streaming interface to confirm that the TLP has been sent. For more details on the Hard IP Reconfiguration Interface, refer to Hard IP Reconfiguration Interface.

Figure 23. Configuration TLP Received by R-Tile IP for PCIe Targeting the Hard IP Internal Registers



3.3.4.2. Transmit Interface

All TLPs transmitted by the application through the TX streaming interface are sent out as-is, without any tracking for completion. The R-Tile IP for PCIe does not perform any check on the TLPs. Your application logic is responsible for sending TLPs that comply with the PCIe specifications.

3.3.4.3. Receive Interface

ALL TLPs received by the IP are transmitted to the application through the RX streaming interface (except Malformed TLPs).

Please refer to the Packets Forwarded to the User Application in TLP Bypass Mode on page 205 Appendix for detailed information.

All PCIe protocol errors leading up to designating a TLP packet as a good packet or not will be detected by the Hard IP and communicated to user logic to take appropriate action in terms of error logging and escalation. The IP does not generate any error message internally, since this is the responsibility of the user logic.

3.3.4.4. Malformed TLP

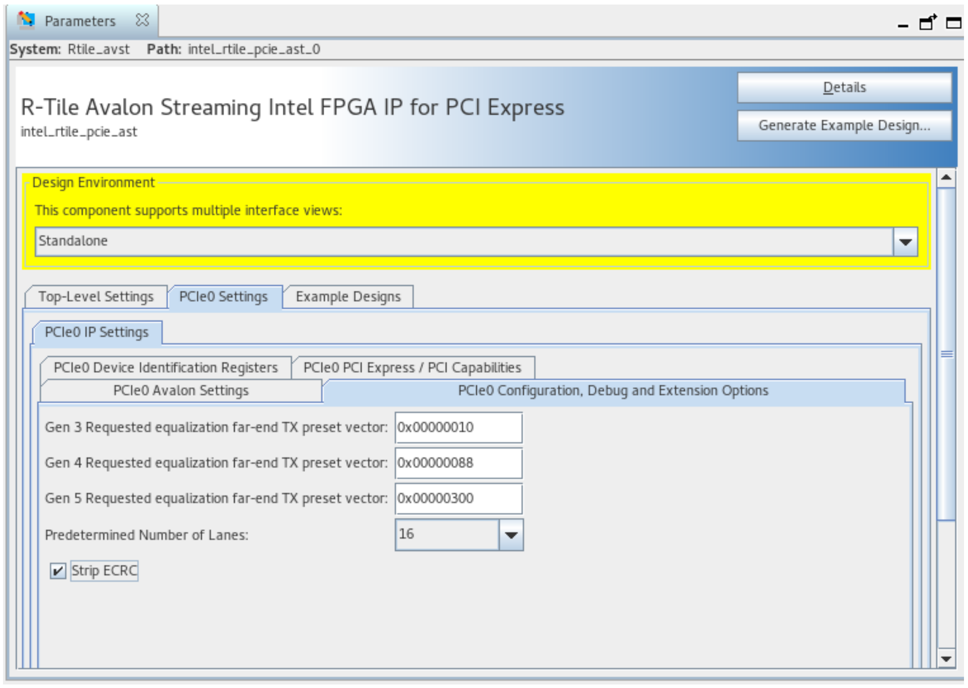
In TLP Bypass mode, a malformed TLP is dropped in the R-Tile IP for PCIe and its event is logged in the AER capability registers. R-Tile also notifies you of this event by asserting the `serr_out_o` signal.

Refer to the *PCI Express Base Specification* for the definition of a malformed TLP.

3.3.4.5. ECRC

In TLP Bypass mode, the R-Tile Avalon Streaming Intel FPGA IP for PCIe does not check the ECRC for received TLPs nor generate the ECRC for transmitted TLPs. In addition, you can configure the IP to strip the ECRC from the TLP payload (when the TD bit is set) by enabling the **Strip ECRC** option in the **PCIeN Configuration, Debug and Extension Options** tab. Note that this option is only available in this tab when the IP is put in TLP Bypass mode (by choosing either the **Upstream** or **Downstream** option for the **Port Mode** parameter in the **Top-Level Settings** tab. Note that with the **Strip ECRC** option enabled, the `cfg_ecrc_err_sts` field in the TLPBYPASS_ERR_STATUS register will be set when there is an ECRC error. However, there is no mechanism to identify the specific TLP with the ECRC error from the application logic.

Figure 24. Strip ECRC Option


Parameters
Details
Generate Example Design...
Top-Level Settings
PCIe0 Settings
Example Designs
PCIe0 IP Settings
PCIe0 Device Identification Registers
PCIe0 PCI Express / PCI Capabilities
PCIe0 Avalon Settings
PCIe0 Configuration, Debug and Extension Options

System: Rtile_avst Path: intel_rtile_pcie_ast_0

R-Tile Avalon Streaming Intel FPGA IP for PCI Express
intel_rtile_pcie_ast

Design Environment
This component supports multiple interface views:
Standalone

Gen 3 Requested equalization far-end TX preset vector: 0x00000010
Gen 4 Requested equalization far-end TX preset vector: 0x00000088
Gen 5 Requested equalization far-end TX preset vector: 0x00000300
Predetermined Number of Lanes: 16
[✓] Strip ECRC

4. Interfaces

This section focuses mainly on the signal interfaces that the R-Tile IP for PCIe uses to communicate with the Application Layer in the FPGA fabric core. However, it also briefly covers the Serial Data Interface, which allows the IP to communicate with the link partner across the PCIe link.

4.1. Clocks and Resets

4.1.1. Clocks

Table 45. Clocks

Name	Direction	Description	EP/RP/BP/ PIPE-D	Clock Frequency
`coreclkout_hip`	Output	This clock drives the Data Link, Transaction, and Application Layers. For the Application Layer, the frequency depends on the data rate and the number of lanes as specified.	EP/RP/BP	Native Gen5: 400/425/450/475/500 MHz Native Gen4: 250/275/300 MHz Native Gen3: 250/275/300 MHz The frequency can change depending on whether the IP is operating in single-width or double-width mode. *Note:* Single-width mode is not supported in the 21.4 release of Intel Quartus Prime.
`refclk0`, `refclk1`, `refclk2`	Input	These are the input reference clocks for the IP core. They must be free-running clocks. You may experience an error while reconfiguring or performing a CvP Update operation on your device if there is no stable free-running clock. These reference clocks are not available to the FPGA user logic. The only clocks available to the user logic from R-Tile are `coreclkout_hip`, `slow_clk` in the PCIe mode, and `pipe_direct_pld_tx_clk_out_o` and `LnX_pipe_direct_pld_rx_clk_out_o` in the PIPE mode.	EP/RP/BP	100 MHz ± 100 ppm *Note:* For Gen5-capable systems, the clock frequency is 100 MHz ± 100 ppm

continued...

Name	Direction	Description	EP/RP/BP/ PIPE-D	Clock Frequency
`reconfig_clk`	Input	This clock is required for proper speed change operation in PIPE Direct mode.	PIPE-D	100 MHz (Recommended) 50 MHz - 125 MHz (Permitted Range)
`slow_clk`	Output	This is the clock for sideband signals.	EP/RP/BP	Divide-by-2 or divide-by-4 clock derived from `coreclkout_hip`. Use the **Slow Clock Divider** option in the Parameter Editor to choose between the divide-by-2 or divide-by-4 versions of `coreclkout_hip` for this clock.
`pipe_direct_pld_tx_clk_out_o`	Output	This is the TX clock for the PIPE Direct mode.	PIPE-D	500 MHz
`LnX_pipe_direct_pld_rx_clk_out_o` (X is the lane number and ranges from 0 to 15. There is one clock output per lane.)	Output	This is the RX clock for the PIPE Direct mode. It is the per-lane CDR recovery clock.	PIPE-D	The clock frequency depends on the lane rate (Gen1 to Gen5). • Gen1 = 125 MHz • Gen2 = 250 MHz • Gen3 = 125 MHz • Gen4 = 250 MHz • Gen5 = 500 MHz

4.1.2. Resets

Table 46. Resets

Name	Direction	Description	EP/RP/BP/ PIPE-D	Asynchronous / Synchronous
`pin_perst_n`	Input	This is the reset signal from the board. This pin is not available to the FPGA user logic. If you want to use the PERST# signal in user logic or in the Intel Signal Tap tool, you need to use the `pin_perst_n_o` signal.	EP/RP/BP	Asynchronous
`pin_perst0_n`, `pin_perst1_n`	Input	These are the reset signals from the board. These inputs ports are only available when using the Configuration Mode 1(2x8) and the parameter **Enable Independent Perst Pins** is set to Enable.	EP	Asynchronous
`pin_perst_n_o`	Output	This output signal to the FPGA fabric indicates if PERST# is asserted.	EP/RP/BP	Asynchronous
`ninit_done`	Input	A "1" on this active-low signal indicates that the FPGA device is not yet fully configured. A "0" indicates the device has been configured and is in normal operating mode. You need to instantiate the Reset Release IP and connect the output of that IP to `ninit_done`.	EP/RP/BP	Asynchronous

continued...

Name	Direction	Description	EP/RP/BP/ PIPE-D	Asynchronous / Synchronous
pX_reset_status_n_o	Output	This active-low signal is held low until pin_perst_n has been deasserted and the PCIe Hard IP has come out of reset. This signal is synchronous to coreclkout_hip. When port bifurcation is used, there is one such signal for each Avalon Streaming interface. Signals for different interfaces are differentiated by the prefixes p<n>. Traffic between the user logic in the FPGA core and the IP can start when pX_reset_status_n_o is asserted high.	EP/RP/BP	Synchronous to coreclkout_hip.
pX_slow_reset_status_n_o	Output	This is the equivalent signal for pX_reset_status_n_o in the slow_clk domain.	EP/RP/BP	Synchronous to slow_clk.
pX_cold_perst_n_i	Input	When enabled, these active-low signals independently trigger cold resets to individual PCIe Controllers. If these inputs are not used, they should be tied off to 1.	EP/RP/BP	Synchronous to coreclkout_hip.
pX_warm_perst_n_i	Input	When enabled, these active-low signals independently trigger warm resets to individual PCIe Controllers. If these inputs are not used, they should be tied off to 1.	EP/RP/BP	Synchronous to coreclkout_hip.
pX_ip_rst_n_o	Output	These active-low output signals are exposed to the Application logic and indicate the status of the Hard Reset Controller triggering resets to individual PCIe Controllers.	EP/RP/BP	Synchronous to coreclkout_hip.
LnX_pipe_direct_reset_status_n (X = 0 - 15)	Output	This active-low per-lane signal is held low until the PHY RX path is out of reset, and when deasserted is an indication to the application logic that the RX data transfer is beginning.	PIPE-D	Synchronous to pipe_direct_pld_tx_clk_out_o.
LnX_pipe_direct_pld_pcs_rst_n_i (X = 0 - 15)	Input	This is the per-lane PHY channel reset signal. The Soft IP Controller must release this signal after the per-lane LnX_pipe_direct_tx_transfer_en_o signal is asserted. Follow the reset sequence as shown in PIPE Direct Reset Sequence on page 131.	PIPE-D	Asynchronous

4.2. Serial Data Interface

In PCI Express modes, R-Tile natively supports 4, 8, or 16 PCIe lanes. Each lane includes a TX differential pair and an RX differential pair. Data is striped across all available lanes. The number of lanes will be different based on the topology which indicates how many lanes are used.

In PIPE Direct mode, R-Tile supports 16 PIPE Direct lanes. Each lane includes a TX differential pair and an RX differential pair. Data is striped across all available lanes. The number of lanes will be different based on the topology which indicates how many lanes are used.

Table 47. Serial Data Interface

Name	Direction	Description

`tx_p_out[15:0]`
`tx_n_out[15:0]`
	Output	Transmit serial data outputs using the High Speed Differential I/O standard.

`rx_p_in[15:0]`
`rx_n_in[15:0]`
	Input	Receive serial data inputs using the High Speed Differential I/O standard.

4.3. PCI Express Mode

In PCI Express mode, only the PCI Express controller stack is active. The four PCI Express cores (x16, x8, x4_0 and x4_1) interface with the application logic in the FPGA fabric via Avalon streaming interfaces. You can determine which core each interface in this section belongs to by looking at the prefixes in the signal names:

- p0 : x16 core
- p1 : x8 core
- p2 : x4_0 core
- p3 : x4_1 core

Note: The x4_0 core is only available in the following OPNs:

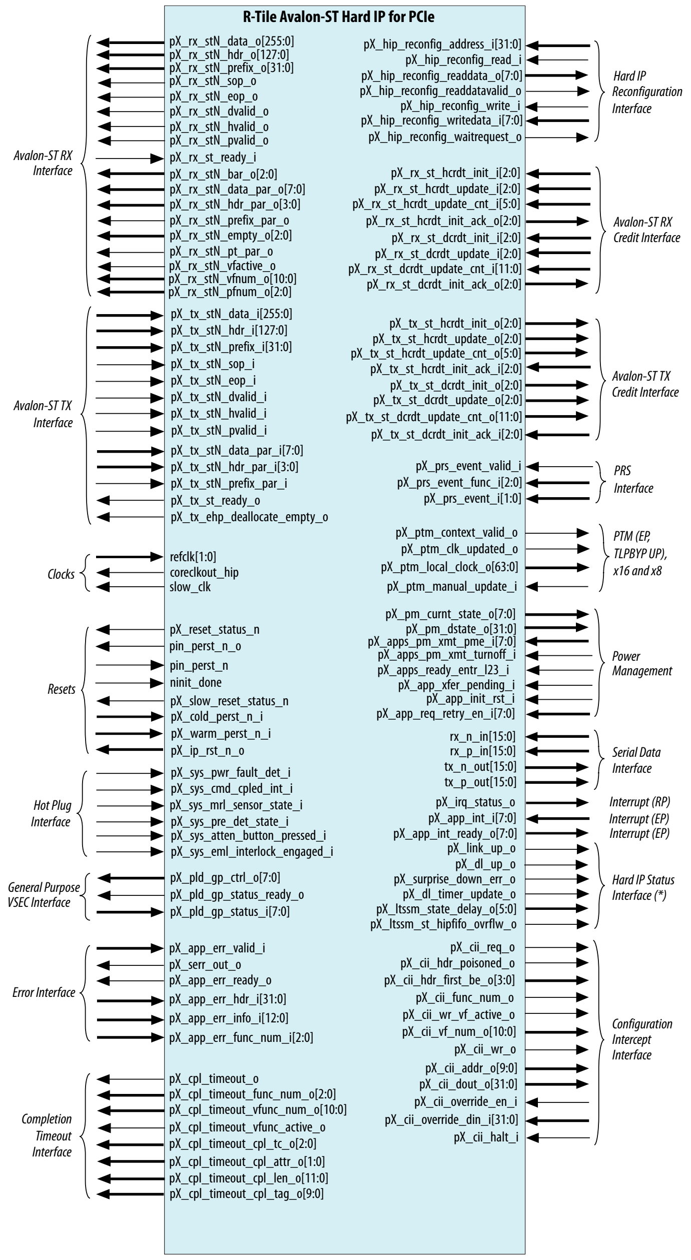
- AGIx027R29AxxxxR2
- AGIx027R29AxxxxR3
- AGIx027R29BxxxxR3
- AGIx023R18AxxxxR0
- AGIx041R29DxxxxR0
- AGIx041R29DxxxxR1

For additional details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

R-Tile Top-Level Block Diagram in PCI Express Mode below shows the top-level signals of this IP. Note that the signal names in the figure will get the appropriate prefixes pn (where n = 0, 1, 2 or 3) depending on which of the supported topologies (x16, x8x8, x4x4x4x4) the R-Tile Avalon streaming Intel FPGA IP for PCIe is in.

The only cases where the interface signal names do not get the pn prefixes are the interfaces that are common for all the cores, like clocks and resets.

Figure 25. R-Tile Top-Level Block Diagram in PCI Express Mode



Note: pX: X is port number, ranges from 0 to 3.

stN: N is segment number, ranges from 0 to 3.

4.3.1. Avalon Streaming Interface

Each of the PCIe cores has its own Avalon streaming interface to the user logic in the FPGA fabric. The number of IP-to-User Logic interfaces exposed to the FPGA fabric are different based on the topologies:

Table 48. IP to FPGA Fabric Interfaces Summary

Topology	Avalon-ST Interface Count	Data Width (each Interface)	Header Width (each Interface)	TLP Prefix Width (each Interface)	Application Clock Frequency	Note
Gen5 1x16 EP/RP/BP	1	1024-bit (four 256-bit segments)	512-bit (four 128-bit segments)	128-bit (four 32-bit segments)	400 MHz / 425 MHz / 450 MHz / 475 MHz / 500 MHz	
Gen4 1x16 EP/RP/BP	1	1024-bit (four 256-bit segments)	512-bit (four 128-bit segments)	128-bit (four 32-bit segments)	250 MHz / 275 MHz / 300 MHz	
		512-bit (two 256-bit segments)	256-bit (two 128-bit segments)	64-bit (two 32-bit segments)	400 MHz / 425 MHz / 450 MHz / 475 MHz / 500 MHz	
Gen3 1x16 EP/RP/BP	1	1024-bit (four 256-bit segments)	512-bit (four 128-bit segments)	128-bit (four 32-bit segments)	250 MHz / 275 MHz / 300 MHz	
		512-bit (two 256-bit segments)	256-bit (two 128-bit segments)	64-bit (two 32-bit segments)	400 MHz / 425 MHz / 450 MHz / 475 MHz / 500 MHz	(5)
Gen5 2x8 EP/RP/BP	2	512-bit (two 256-bit segments)	256-bit (two 128-bit segments)	64-bit (two 32-bit segments)	400 MHz / 425 MHz / 450 MHz / 475 MHz / 500 MHz	
Gen4 2x8 EP/RP/BP	2	512-bit (two 256-bit segments)	256-bit (two 128-bit segments)	64-bit (two 32-bit segments)	250 MHz / 275 MHz / 300 MHz	
		256-bit (one 256-bit segment)	128-bit (one 128-bit segment)	32-bit (one 32-bit segment)	400 MHz / 425 MHz / 450 MHz / 475 MHz / 500 MHz	(5)
Gen3 2x8 EP/RP/BP	2	512-bit (two 256-bit segments)	256-bit (two 128-bit segments)	64-bit (two 32-bit segments)	250 MHz / 275 MHz / 300 MHz	
		256-bit (one 256-bit segment)	128-bit (one 128-bit segment)	32-bit (one 32-bit segment)	250 MHz / 275 MHz / 300 MHz	(5)
Gen5 4x4 EP/RP/BP	4	256-bit (two 128-bit segments)	256-bit (two 128-bit segments)	64-bit (two 32-bit segments)	400 MHz / 425 MHz / 450 MHz / 475 MHz / 500 MHz	
Gen4 4x4 EP/RP/BP	4	256-bit (two 128-bit segments)	256-bit (two 128-bit segments)	64-bit (two 32-bit segments)	400 MHz / 425 MHz / 450 MHz / 475 MHz / 500 MHz	

continued...

(5) This topology is only available in the following OPNs: AGIx027R29AxxxxR2, AGIx027R29AxxxxR3, AGIx027R29BxxxxR3, AGIx023R18AxxxxR0, AGIx041R29DxxxxR0, AGIx041R29DxxxxR1. For more details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

Topology	Avalon-ST Interface Count	Data Width (each Interface)	Header Width (each Interface)	TLP Prefix Width (each Interface)	Application Clock Frequency	Note
		128-bit (one 128-bit segment)	128-bit (one 128-bit segment)	32-bit (one 32-bit segment)	400 MHz / 425 MHz / 450 MHz / 475 MHz / 500 MHz	(5)
Gen3 4x4 EP/RP/BP	4	256-bit (two 128-bit segments)	256-bit (two 128-bit segments)	64-bit (two 32-bit segments)	250 MHz / 275 MHz / 300 MHz	
		128-bit (one 128-bit segment)	128-bit (one 128-bit segment)	32-bit (one 32-bit segment)	250 MHz / 275 MHz / 300 MHz	(5)

The R-Tile PCIe Hard IP provides an Avalon Streaming-like interface with separate header and data to improve the bandwidth utilization.

The Avalon Streaming interface has different data bus widths depending on the link width configuration of the PCIe IP.

Table 49. Avalon Streaming Interface Data and Header Bus Widths per Port

Link Width	Link Speed	Data Width (Bits)	Header Width (Bits)	TLP Prefix Width (Bits)	Note
x16	Gen5	1024 (4 x 256)	512 (4 x 128)	128 (4 x 32)	
	Gen4	1024 (4 x 256)	512 (4 x 128)	128 (4 x 32)	
		512 (2 x 256)	256 (2 x 128)	64 (2 x 32)	
	Gen3	1024 (4 x 256)	512 (4 x 128)	128 (4 x 32)	
		512 (2 x 256)	256 (2 x 128)	64 (2 x 32)	(6)
x8	Gen5	512 (2 x 256)	256 (2 x 128)	64 (2 x 32)	
	Gen4	512 (2 x 256)	256 (2 x 128)	64 (2 x 32)	
		256 (1 x 256)	128 (1 x 128)	32 (1 x 32)	(6)
	Gen3	512 (2 x 256)	256 (2 x 128)	64 (2 x 32)	
		256 (1 x 256)	128 (1 x 128)	32 (1 x 32)	(6)
x4	Gen5	256 (2 x 128)	256 (2 x 128)	64 (2 x 32)	
	Gen4	256 (2 x 128)	256 (2 x 128)	64 (2 x 32)	
		128 (1 x 128)	128 (1 x 128)	32 (1 x 32)	(6)
	Gen3	256 (2 x 128)	256 (2 x 128)	64 (2 x 32)	
		128 (1 x 128)	128 (1 x 128)	32 (1 x 32)	(6)

(6) This topology is only available in the following OPNs: AGIx027R29AxxxxR2, AGIx027R29AxxxxR3, AGIx027R29BxxxxR3, AGIx023R18AxxxxR0, AGIx041R29DxxxxR0, AGIx041R29DxxxxR1. For more details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

4.3.1.1. TLP Header and Data Alignment for the Avalon streaming RX and TX Interfaces

The TLP prefix, header and data are sent and received on the Avalon streaming TX and RX interfaces. There are four Avalon streaming TX/RX interfaces corresponding to the four PCIe cores. The signal names on these interfaces are prefixed with "p0" for the x16 core, "p1" for the x8 core, "p2" for the x4_0 core and "p3" for the x4_1 core.

Each of these interfaces can contain up to four segments. The signal names for these segments are notated with *_st0_*, *_st1_*, *_st2_* and *_st3_*.

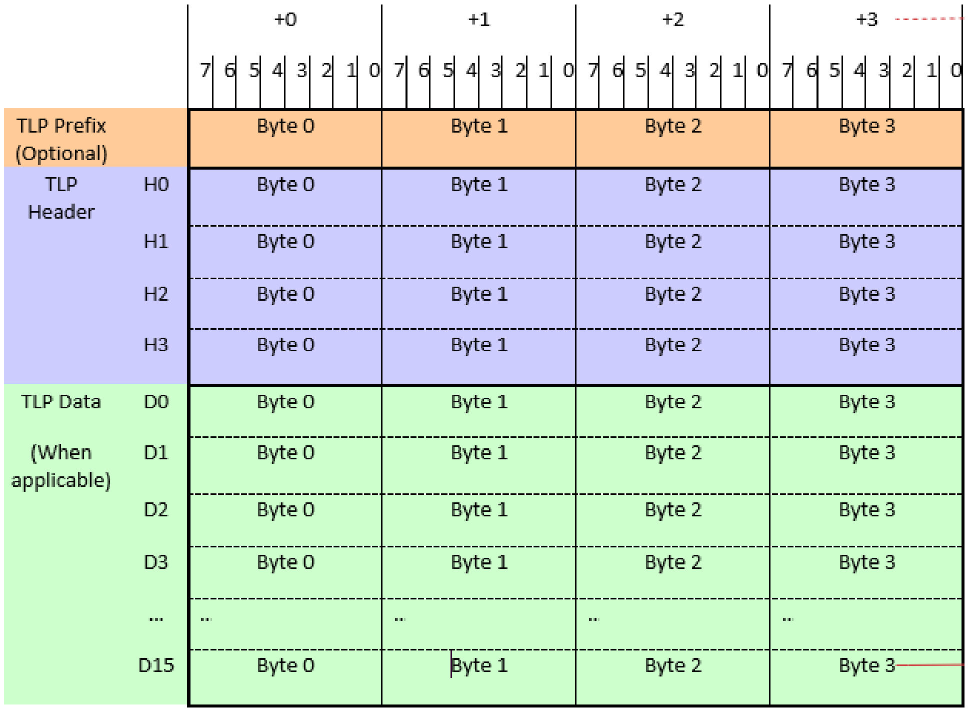
A Start of Packet (SOP) symbol demarcates the start of a packet. A TLP with or without data is terminated by an End of Packet (EOP) symbol. An EOP happening on the st0, st1, st2 or st3 segment implies that the last data with data valid = 1 is located on the corresponding data bus.

A TLP without data is qualified by a header valid indicator. A TLP with data is qualified by a data valid indicator. There is also a prefix valid indicator to qualify a TLP Prefix.

If a global valid signal is required for backward compatibility to P-tile, the header valid ORed with data valid will provide the global valid indicator.

This is defined to enable pipelined Header/Prefix and Data transfers to meet the bandwidth target. For example, while transferring the Data for one TLP, the Header and Prefix for the next TLP can also be transferred.

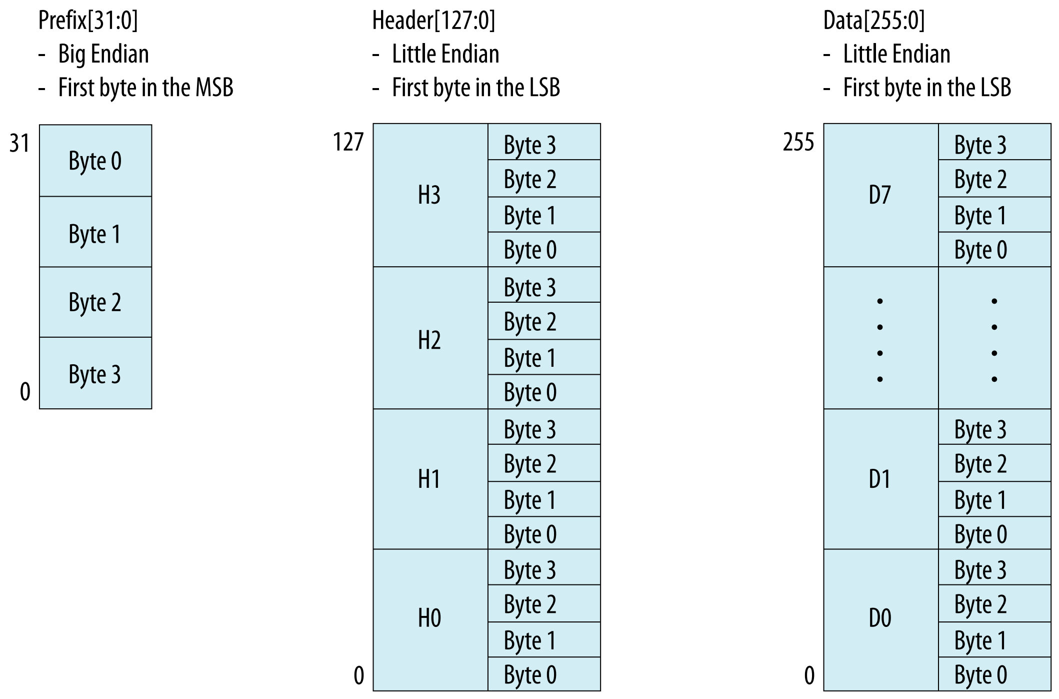
Figure 26. Generic TLP Format



The Avalon streaming Header and TLP Prefix bus packet format follows the TLP packet format as defined by the PCIe specification for Memory, Configuration and Message TLPs. Refer to TLP Prefix, Header and Data when PCIe Header Format Checkbox is Disabled and TLP Prefix, Header and Data when PCIe Header Format Checkbox is Enabled below for more details.

In the R-Tile Parameter Editor, there is a checkbox labeled **PCIe Header format** (in the **PCIe Avalon Settings** tab) that allows you to specify whether the Header will follow the Big Endian format or Little Endian format. If this checkbox is not enabled, the Header follows the Little Endian format as shown in TLP Prefix, Header and Data when PCIe Header Format Checkbox is Disabled (note that the diagram shows the Prefix, Header and Data for one segment of the Avalon streaming interface, not the whole interface):

Figure 27. TLP Prefix, Header and Data when PCIe Header Format Checkbox is Disabled



If the **PCIe Header format** checkbox is enabled, the Header follows the Big Endian format as shown in TLP Prefix, Header and Data when PCIe Header Format Checkbox is Enabled:

Figure 28. TLP Prefix, Header and Data when PCIe Header Format Checkbox is Enabled

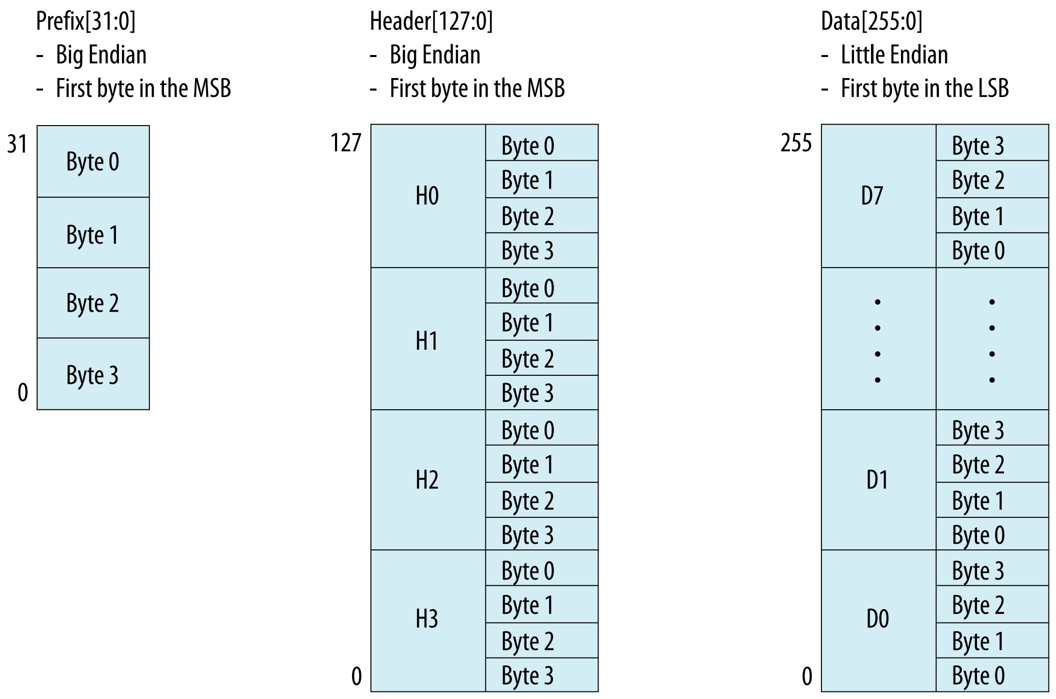


Table 50. IP to FPGA Fabric Interface Summary

Configuration	Link Rate	Width Mode	Header	Data	Application Clock Frequency (MHz)
1x16	Gen5	Double-Width	st3_hdr[127:0] st2_hdr[127:0] st1_hdr[127:0] st0_hdr[127:0]	st3_data[255:0] st2_data[255:0] st1_data[255:0] st0_data[255:0]	400 MHz 425 MHz 450 MHz 475 MHz 500 MHz
	Gen4	Double-Width	st3_hdr[127:0] st2_hdr[127:0] st1_hdr[127:0] st0_hdr[127:0]	st3_data[255:0] st2_data[255:0] st1_data[255:0] st0_data[255:0]	250 MHz 275 MHz 300 MHz
		Single-Width [7]	st1_hdr[127:0] st0_hdr[127:0]	st1_data[255:0] st0_data[255:0]	400 MHz 425 MHz 450 MHz 475 MHz 500 MHz

continued...

[7] Single-width mode is only available in the following OPNs: AGIx027R29AxxxxR2, AGIx027R29AxxxxR3, AGIx027R29BxxxxR3, AGIx023R18AxxxxR0, AGIx041R29DxxxxR0, AGIx041R29DxxxxR1. For additional details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

Configuration	Link Rate	Width Mode	Header	Data	Application Clock Frequency (MHz)
	Gen3	Double-Width	st3_hdr[127:0] st2_hdr[127:0] st1_hdr[127:0] st0_hdr[127:0]	st3_data[255:0] st2_data[255:0] st1_data[255:0] st0_data[255:0]	250 MHz 275 MHz 300 MHz
		Single-Width (7)	st1_hdr[127:0] st0_hdr[127:0]	st1_data[255:0] st0_data[255:0]	400 MHz 425 MHz 450 MHz 475 MHz 500 MHz
2x8	Gen5	Double-Width	st1_hdr[127:0] st0_hdr[127:0]	st1_data[255:0] st0_data[255:0]	400 MHz 425 MHz 450 MHz 475 MHz 500 MHz
	Gen4	Double-Width	st1_hdr[127:0] st0_hdr[127:0]	st1_data[255:0] st0_data[255:0]	250 MHz 275 MHz 300 MHz
		Single-Width (7)	st0_hdr[127:0]	st0_data[255:0]	400 MHz 425 MHz 450 MHz 475 MHz 500 MHz
	Gen3	Double-Width	st1_hdr[127:0] st0_hdr[127:0]	st1_data[255:0] st0_data[255:0]	250 MHz 275 MHz 300 MHz
		Single-Width (7)	st0_hdr[127:0]	st0_data[255:0]	250 MHz 275 MHz 300 MHz
4x4	Gen5	Double-width	st1_hdr[127:0] st0_hdr[127:0]	st1_data[127:0] st0_data[127:0]	400 MHz 425 MHz 450 MHz 475 MHz 500 MHz
	Gen4	Double-width	st1_hdr[127:0] st0_hdr[127:0]	st1_data[127:0] st0_data[127:0]	400 MHz 425 MHz 450 MHz 475 MHz 500 MHz
		Single-Width (7)	st0_hdr[127:0]	st0_data[127:0]	400 MHz 425 MHz 450 MHz 475 MHz 500 MHz
	Gen3	Double-width	st1_hdr[127:0] st0_hdr[127:0]	st1_data[127:0] st0_data[127:0]	250 MHz 275 MHz 300 MHz
		Single-Width (7)	st0_hdr[127:0]	st0_data[127:0]	250 MHz 275 MHz 300 MHz

Note:

- Refer to Section 4.3.1.3.1 and Section 4.3.1.4.1 for Application logic guidelines for the proper handling of the Avalon Streaming interface.

4.3.1.2. Credit Control

4.3.1.2.1. Credit Interface - Data and Header

The credit interface is used to implement flow control for the data movement between the user application interface and each IP block. Each header type (P,NP,CPL) and data type (P,NP,CPL) has an independent credit handling. One data credit consists of 16 bytes. One header credit includes the TLP Header, 1DW prefix (if present) and the digest (if present).The section below describes the credit initialization and update flow.

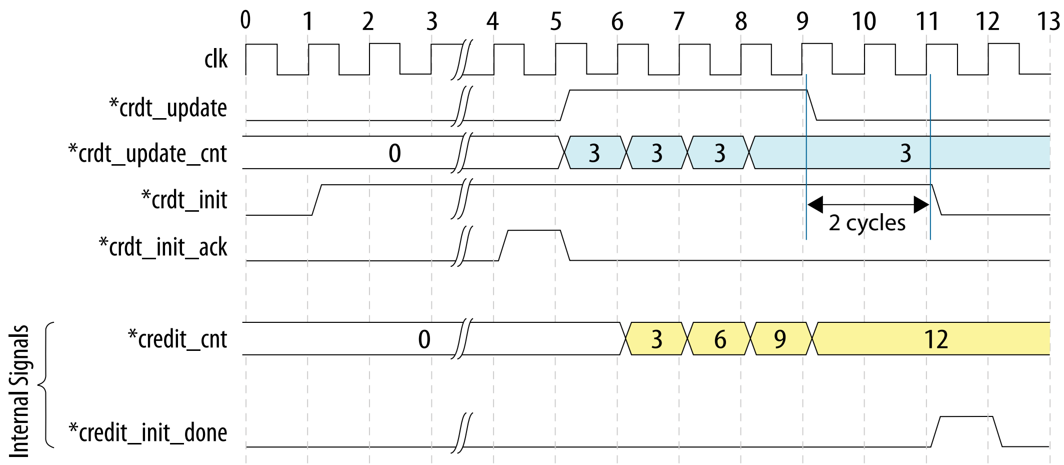
4.3.1.2.2. Credit Initialization

Each side will initialize its credit counters during the initialization stage. In this stage, the sink communicates its total buffer size available to the source. The sink initiates the initialization phase by asserting the `*crdt_init` signal. If the source is ready for initialization, it asserts `*crdt_init_ack` in response. After receiving `*crdt_init_ack`, the sink asserts the `*crdt_update` signal to communicate the buffer size available to receive transactions.

The source increments its internal credit counter by `*crdt_update_cnt` for every cycle it encounters an assertion of the `*update` signal for the header credit.

The deassertion of `*init` signals the completion of the initialization phase. There is a required minimum of 2 clock cycles separation between the deassertion of the `*update` and `*init` signals.

Figure 29. Credit Initialization



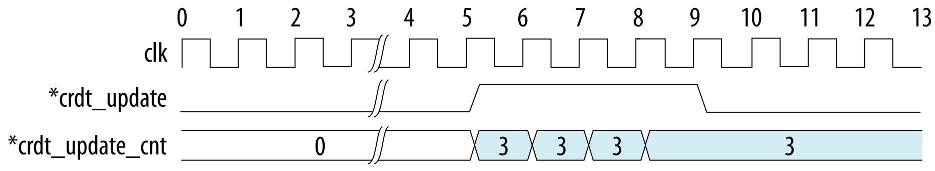
Note: The internal signals in the figure above are not exposed to the application logic.

Re-initialization is generally not required. However, if required (for example, if the application logic wants to change the buffer size), it can be done only when the source and sink are both in IDLE state. The IDLE state for the sink is when its receiver buffer is empty. For the source, the IDLE state is when it has nothing to transmit when the initialization request is made.

4.3.1.2.3. Credit Update/Release

Once the initialization is completed, the source will decrement the credit available for transmission whenever it transmits a transaction. The source also increments the credit available count by `*crdt_update_cnt` whenever the sink asserts the `*update` signal.

Figure 30. Credit Update Timings



4.3.1.2.4. RX Flow Control Interface

The RX flow control interface provides information on the application's available RX buffer space for Posted (P), Non-Posted (NP) and Completion (CPL) transactions to the PCIe Hard IP. It reports the space available in number of credits as specified by the PCIe Specification.

The Application logic must ensure that enough credits are provided to the R-Tile Avalon-ST IP for PCIe during the Credit Initialization phase. This is to prevent a performance impact caused by the lack of credits available between the IP and the Application logic. Depending on the number of P, NP and CPL transactions happening at the PCIe link level, a lack of credit scenario may happen if the number of credits advertised by the Application logic is less than the credits advertised by the R-Tile Avalon-ST IP for PCIe to the link partner.

Flow control credits are available for the following TLP categories:

- Posted (P) transactions: TLPs that do not required a response.
- Non-poseted (NP) transactions: TLPs that require a completion.
- Completions (CPL): TLPs that respond to non-posted transactions.

Table 51. Credits Advertised by the R-Tile Avalon-ST IP for PCIe to the Link Partner

Note: The credit unit in this table follows the PCIe Specification where 1 credit = 4 DWs = 16 Bytes.

Port	Hard IP Mode	Posted Headers	Posted Data	Non-Posted Headers	Non-Posted Data	Completion Headers	Completion Data
Port 0	Endpoint	784	1456	784	392	0 (infinite)	0 (infinite)
	Root Port	784	1456	784	392	0 (infinite)	0 (infinite)
	Upstream Port	784	1456	784	392	1024	2816
	Downstream Port	784	1456	784	392	1024	2816

continued...

Port	Hard IP Mode	Posted Headers	Posted Data	Non-Posted Headers	Non-Posted Data	Completion Headers	Completion Data
Port 1	Endpoint	392	760	392	196	0 (infinite)	0 (infinite)
	Root Port	392	760	392	196	0 (infinite)	0 (infinite)
	Upstream Port	392	760	392	196	512	1408
	Downstream Port	392	760	392	196	512	1408
Port 2	Endpoint	224	444	224	112	0 (infinite)	0 (infinite)
	Root Port	224	444	224	112	0 (infinite)	0 (infinite)
	Upstream Port	224	444	224	112	256	704
	Downstream Port	224	444	224	112	256	704
Port 3	Endpoint	224	444	224	112	0 (infinite)	0 (infinite)
	Root Port	224	444	224	112	0 (infinite)	0 (infinite)
	Upstream Port	224	444	224	112	256	704
	Downstream Port	224	444	224	112	256	704

For more information on how credit control in general is implemented in this IP, refer to Credit Control on page 74.

Note: In some systems, there are broadcast Message TLPs being sent by the link partner during the PCIe enumeration process. Failing to properly initialize and return the corresponding credits from the Application logic to the R-Tile Avalon Streaming IP may cause enumeration issues due to the priority order between Message TLPs and the Completions required for Configuration TLPs. Application logic must ensure proper credits are returned to the R-Tile Avalon Streaming IP for any TLP that it receives.

Table 52. Categorization of Transaction Types

TLP Type	Category
Memory Write	Posted
Memory Read	Non-Posted
Memory Read Lock	
I/O Read	
I/O Write	
Configuration Read	
Configuration Write	
Fetch and Add AtomicOp	
Message	Posted
Completion	Completion

continued...

TLP Type	Category
Completion with Data	
Completion Lock	
Completion Lock with Data	

Table 53. RX Flow Control Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
pX_rx_st_hcrdt_update_i[2:0] where X = 0, 1, 2, 3 (IP core number)	Input	Indicates credit is made available for the different types of Header. Each Header (including the TLP Prefix, if any) consumes one credit. [0] : Posted Header (PH) [1] : Non-Posted Header (NPH) [2] : Completion Header (CPLH) To advertise infinite credits for a specific TLP type between the IP and the Application logic, the corresponding bit within this signal bus must be asserted for one clock cycle, with a value of 0 on the corresponding bits for the targeted TLP type in the pX_rx_st_hcrdt_update_cnt_i signal bus during the credit initialization phase.	EP/RP/BP	coreclkout_hip
pX_rx_st_hcrdt_update_cnt_i[5:0] where X = 0, 1, 2, 3 (IP core number)	Input	Indicates number of credits released. [1:0] : number of PH credits released [3:2] : number of NPH credits released [5:4] : number of CPLH credits released Valid when the corresponding pX_rx_st_hcrdt_update_i bit = 1. The maximum number of credits released is three. To advertise infinite credits for a specific TLP type between the IP and the Application logic, the corresponding bits within this signal bus must be set to 0 during the credit initialization phase (when the corresponding pX_rx_st_hcrdt_update_i bit is asserted for one clock cycle).	EP/RP/BP	coreclkout_hip
pX_rx_st_hcrdt_init_i[2:0] where X = 0, 1, 2, 3 (IP core number)	Input	Credit Initialization indicator. These signals remain high for entire initialization phase. A High to Low transition indicates the completion of the credit initialization phase. [0] : PH [1] : NPH [2] : CPLH	EP/RP/BP	coreclkout_hip
pX_rx_st_hcrdt_init_ack_o[2:0] where X = 0, 1, 2, 3 (IP core number)	Output	Indicates the Host is ready for the credit initialization phase [0] : PH [1] : NPH [2] : CPLH	EP/RP/BP	coreclkout_hip
pX_rx_st_dcrdt_update_i[2:0] where X = 0, 1, 2, 3 (IP core number)	Input	Indicates credit is made available for the different types of Data. [0] : Posted Data (PD) [1] : Non-Posted Data (NPD) [2] : Completion Data (CPLD)	EP/RP/BP	coreclkout_hip

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
		To advertise infinite credits for a specific TLP type between the IP and the Application logic, the corresponding bit within this signal bus must be asserted for one clock cycle, with a value of 0 on the corresponding bits for the targeted TLP type in the `pX_rx_st_dcrdt_update_cnt_i` signal bus during the credit initialization phase.		
`pX_rx_st_dcrdt_update_cnt_i[11:0]` where X = 0, 1, 2, 3 (IP core number)	Input	Indicates number of credits released. [3:0] : number of PD credits released [7:4] : number of NPD credits released [11:8] : number of CPLD credits released Valid when the corresponding `pX_rx_st_dcrdt_update_i` bit = 1. The maximum number of credits released is 15. To advertise infinite credits for a specific TLP type between the IP and the Application logic, the corresponding bits within this signal bus must be set to 0 during the credit initialization phase (when the corresponding `pX_rx_st_dcrdt_update_i` bit is asserted for one clock cycle). *Note:* The initial RX NPD credits must be equal or larger than the Maximum Payload Size.	EP/RP/BP	`coreclkout_hip`
`pX_rx_st_dcrdt_init_i[2:0]` where X = 0, 1, 2, 3 (IP core number)	Input	Credit Initialization indicator. These signals remain high for entire initialization phase. A High to Low transition indicates the completion of the credit initialization phase. [0] : PD [1] : NPD [2] : CPLD	EP/RP/BP	`coreclkout_hip`
`pX_rx_st_dcrdt_init_ack_o[2:0]` where X = 0, 1, 2, 3 (IP core number)	Input	Indicates the Host is ready for the credit initialization phase [0] : PD [1] : NPD [2] : CPLD	EP/RP/BP	`coreclkout_hip`

4.3.1.2.5. TX Flow Control Interface

Before a TLP can be transmitted, flow control logic verifies that the link partner's RX port has sufficient buffer space to accept it. The TX Flow Control interface reports the link partner's available RX buffer space to the Application. It reports the space available in units called Flow Control credits for posted, non-posted and completion TLPs (as defined in the RX Flow Control Interface section).

For more information on how credit control in general is implemented in this IP, refer to Credit Control on page 74.

Table 54. TX Flow Control Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
`pX_tx_st_hcrdt_update_o[2:0]` where	Output	Indicates credit is made available for the different types of Header.	EP/RP/BP	`coreclkout_hip`

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
X = 0, 1, 2, 3 (IP core number)		Each Header (including the TLP Prefix, if any) consumes one credit. [0] : Posted Header (PH) [1] : Non-Posted Header (NPH) [2] : Completion Header (CPLH) When the link partner advertises infinite credits, this signal will get asserted for one clock cycle, with a value of 0 on `pX_tx_st_hcrdt_update_cnt_o` during the credit initialization phase.		
`pX_tx_st_hcrdt_update_cnt_o[5:0]` where X = 0, 1, 2, 3 (IP core number)	Output	Indicates number of credits released. [1:0] : number of PH credits released [3:2] : number of NPH credits released [5:4] : number of CPLH credits released Valid when the corresponding `pX_tx_st_hcrdt_update_o` bit = 1. The maximum number of credits released is three. When the link partner advertises infinite credits, this signal will reflect a 0 during the credit initialization phase (when `pX_tx_st_hcrdt_update_o` is asserted for one clock cycle).	EP/RP/BP	`coreclkout_hip`
`pX_tx_st_hcrdt_init_o[2:0]` where X = 0, 1, 2, 3 (IP core number)	Output	Credit Initialization indicator. These signals remain high for entire initialization phase. A High to Low transition indicates the completion of the credit initialization phase. [0] : PH [1] : NPH [2] : CPLH	EP/RP/BP	`coreclkout_hip`
`pX_tx_st_hcrdt_init_ack_i[2:0]` where X = 0, 1, 2, 3 (IP core number)	Input	Indicates the Application logic is ready for the credit initialization phase [0] : PH [1] : NPH [2] : CPLH	EP/RP/BP	`coreclkout_hip`
`pX_tx_st_dcrdt_update_o[2:0]` where X = 0, 1, 2, 3 (IP core number)	Output	Indicates credit is made available for the different types of Data. [0] : Posted Data (PD) [1] : Non-Posted Data (NPD) [2] : Completion Data (CPLD) When the link partner advertises infinite credits, this signal will get asserted for one clock cycle, with a value of 0 on `pX_tx_st_dcrdt_update_cnt_o` during the credit initialization phase.	EP/RP/BP	`coreclkout_hip`
`pX_tx_st_dcrdt_update_cnt_o[11:0]` where X = 0, 1, 2, 3 (IP core number)	Output	Indicates number of credits released. [3:0] : number of PD credits released [7:4] : number of NPD credits released [11:8] : number of CPLD credits released Valid when the corresponding `pX_tx_st_dcrdt_update_o` bit = 1. The maximum number of credits released is 15.	EP/RP/BP	`coreclkout_hip`

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
		When the link partner advertises infinite credits, this signal will reflect a 0 during the credit initialization phase (when pX_tx_st_dcrdt_update_o is asserted for one clock cycle.		
pX_tx_st_dcrdt_init_o[2:0] where X = 0, 1, 2, 3 (IP core number)	Output	Credit Initialization indicator. These signals remain high for entire initialization phase. A High to Low transition indicates the completion of the credit initialization phase. [0] : PD [1] : NPD [2] : CPLD	EP/RP/BP	coreclkout_hip
pX_tx_st_dcrdt_init_ack_i[2:0] where X = 0, 1, 2, 3 (IP core number)	Input	Indicates the Application logic is ready for the credit initialization phase [0] : PD [1] : NPD [2] : CPLD	EP/RP/BP	coreclkout_hip

4.3.1.3. Avalon Streaming RX Interface

The Application Layer receives data from the Transaction Layer of the R-Tile PCI Express IP core over the Avalon Streaming RX interface. For R-Tile, the rx_st_ready_i has to be always high. The buffer control in the application logic needs to be handled by the RX Flow Control interface. Refer to RX Flow Control Interface on page 75 for more details.

This interface supports four rx_st_sop_o signals and four rx_st_eop_o signals per cycle when the R-Tile IP is operating in a 1x16 configuration.

In Configuration Mode 0 (1x16) with a double-width configuration, the core provides four segments with each one having 256 bits of data (pX_rx_stN_data_o[255:0]), 128 bits of header (pX_rx_stN_hdr_o[127:0]), and 32 bits of TLP prefix (pX_rx_stN_prefix_o[31:0]). If this core is configured in the 1x16 mode, the data bus becomes a 1024-bit bus.

In Configuration Mode 1 (2x8) with a double-width configuration, there are still four Avalon Streaming segments (two for each x8 port).

Parity generation is done via a 32:1 XOR (i.e. there is one parity bit for every 32 data, header or prefix bits).

Table 55. Avalon Streaming RX Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
pX_rx_stN_data_o[W:0] where X = 0,1,2,3 (IP core number) and W varies based on the core. N = 0,1,2,3 (segment number)	Output	This is the Receive data bus. The Application Layer receives data from the Transaction Layer of the IP core on this bus.	EP/RP/BP	coreclkout_hip
pX_rx_stN_hdr_o[127:0] where	Output	This is the received header, which follows the TLP header format of the PCIe specifications.	EP/RP/BP	coreclkout_hip

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain

X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)				

pX_rx_stN_prefix_o[31:0] where
X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)
	Output	This is the first TLP prefix received, which follows the TLP prefix format of the PCIe specifications. PASID is supported.
		These signals are valid when the corresponding rx_st_sop_o is asserted.
		The TLP prefix uses a Big Endian implementation (i.e, the Fmt field is in bits [31:29] and the Type field is in bits [28:24]).
		If no prefix is present for a given TLP, that dword (including the Fmt field) is all zeros.
			EP/RP/BP	coreclkout_hip

pX_rx_stN_sop_o where
X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)
	Output	Signals the first cycle of the TLP when asserted in conjunction with the corresponding bit of rx_stN_valid_o.
		rx_stN_sop_o: When asserted, signals the start of a TLP on rx_stN_data_o[255:0].
		For example, when asserted, rx_st2_sop_o signals the start of a TLP on rx_st2_data_o[255:0].
			EP/RP/BP	coreclkout_hip

pX_rx_stN_eop_o where
X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)
	Output	Signals the last cycle of the TLP when asserted in conjunction with the corresponding bit of rx_stN_valid_o.
		rx_stN_eop_o: When asserted, signals the end of a TLP on rx_stN_data_o[255:0].
		For example, when asserted, rx_st2_eop_o signals the end of a TLP on rx_st2_data_o[255:0].
			EP/RP/BP	coreclkout_hip

pX_rx_stN_dvalid_o where
X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)
	Output	These signals qualify the rx_stN_data_o signals going into the Application Layer.	EP/RP/BP	coreclkout_hip

pX_rx_stN_hvalid_o where
X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)
	Output	These signals qualify the rx_stN_hdr_o signals going into the Application Layer.	EP/RP/BP	coreclkout_hip

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
pX_rx_stN_pvalid_o where X = 0,1,2,3 (IP core number) N = 0,1,2,3 (segment number)	Output	These signals qualify the rx_stN_prefix_o signals going into the Application Layer.	EP/RP/BP	coreclkout_hip
pX_rx_stN_data_par_o[Z:0] where X = 0,1,2,3 (IP core number) and Z varies based on the core. N = 0,1,2,3 (segment number)	Output	Parity signals for rx_stN_data_o.	EP/RP/BP	coreclkout_hip
pX_rx_stN_hdr_par_o[3:0] where X = 0,1,2,3 (IP core number) N = 0,1,2,3 (segment number)	Output	Parity signals for rx_stN_hdr_o.	EP/RP/BP	coreclkout_hip
pX_rx_stN_prefix_par_o where X = 0,1,2,3 (IP core number) N = 0,1,2,3 (segment number)	Output	Parity signals for rx_stN_prefix_o.	EP/RP/BP	coreclkout_hip
pX_rx_st_ready_i	Input	Indicates the Application Layer is ready to accept data. This signal should always be set to 1. The Flow Control on the RX side is handled through the Credit Control Interface.	EP/RP/BP	coreclkout_hip
pX_rx_stN_empty_o[2:0] where X = 0,1,2,3 (IP core number) N = 0,1,2,3 (segment number)	Output	Specifies the number of dwords that are empty during cycles when the rx_stN_eop_o signals are asserted. These signals are not valid when the rx_stN_eop_o signals are not asserted.	EP/RP/BP	coreclkout_hip
pX_rx_stN_bar_o[2:0] where X = 0,1,2,3 (IP core number) N = 0,1,2,3 (segment number)	Output	Specify the BAR for the TLP being output. These outputs are valid when both rx_stN_sop_o and rx_stN_valid_o are asserted.	EP/RP	coreclkout_hip
pX_rx_stN_vfactive_o where X = 0,1,2,3 (IP core number) N = 0,1,2,3 (segment number)	Output	When asserted, these signals indicate that the received TLP is targeting a virtual function. When these signals are deasserted, the received TLP is targeting a physical function and the rx_stN_pfnum_o signals indicate the function number.	EP/RP	coreclkout_hip

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
		These signals are valid when the corresponding `rx_stN_sop_o` is asserted.		
`pX_rx_stN_vfnum_o[10:0]` where X = 0,1,2,3 (IP core number) N = 0,1,2,3 (segment number)	Output	Specify the target VF number for the received TLP. The application uses this information for both request and completion TLPs. For a completion TLP, these bits specify the VF number of the requester for this completion TLP. These signals are valid when `rx_stN_vf_active_o` and the corresponding `rx_stN_sop_o` are asserted.	EP/RP	`coreclkout_hip`
`pX_rx_stN_pfnum_o[2:0]` where X = 0,1,2,3 (IP core number) N = 0,1,2,3 (segment number)	Output	Specify the target physical function number for the received TLP. These signals are valid when the corresponding `rx_stN_sop_o` is asserted.	EP/RP	`coreclkout_hip`

As an example, Avalon Streaming RX Interface Timings below shows the behavior of the Avalon Streaming RX interface with multiple TLPs, and each of those TLPs spanning across multiple segments. The following text describes the waveforms per clock cycle:

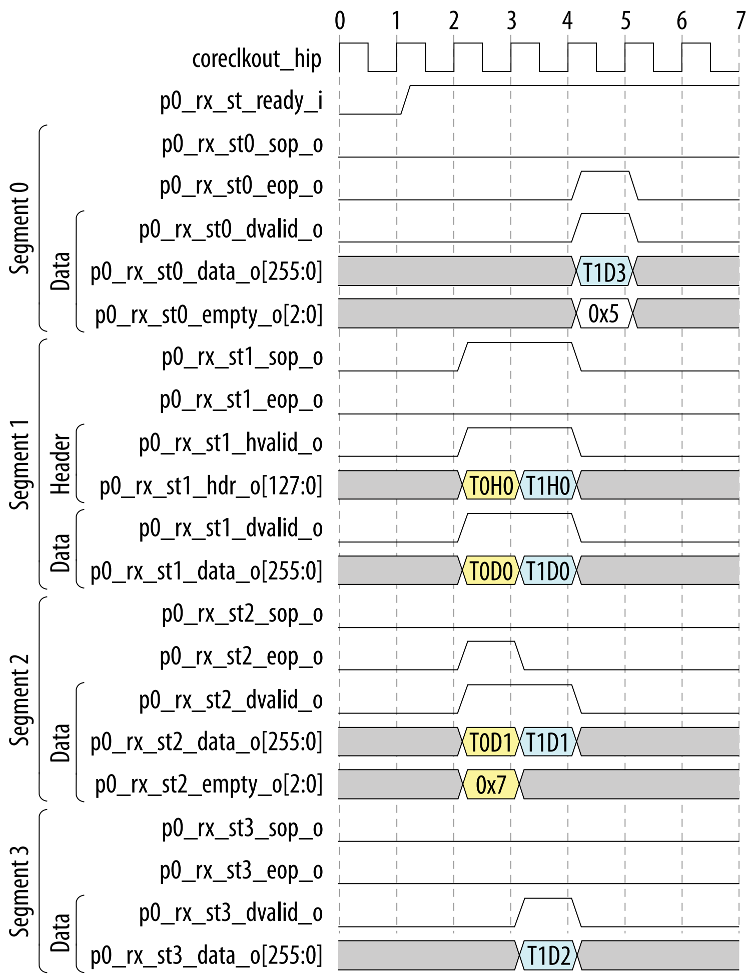
1. Clock cycle 1: Application logic asserts `p0_rx_st_ready_i` signal. This signal must be set high all the time. The RX flow control must be handled by the Application logic using the RX Flow Control interface (refer to RX Flow Control Interface on page 75 for details).
2. Clock cycle 2:
    a. The start of the first TLP (T0) arrives in segment 1, when `p0_rx_st1_sop_o` is asserted.
    b. The signal `p0_rx_st1_hvalid_o` is asserted to validate the header of this first TLP (T0H0) in the `p0_rx_st1_hdr_o` bus.
    c. The signal `p0_rx_st1_dvalid_o` is asserted to validate the data of this first TLP (T0D0) in the `p0_rx_st1_data_o` bus.
    d. The end of this first TLP (T0) is in segment 2, denoted by the assertion of `p0_rx_st2_eop_o`.
    e. The signal `p0_rx_st2_dvalid_o` is asserted to validate the data of this first TLP (T0D1) in the `p0_rx_st2_data_o` bus.
    f. The bus `p0_rx_st2_empty_o` indicates the number of dwords that are not valid in the `p0_rx_st2_data_o` bus (T0D1).
3. Clock cycle 3:

   a. The next TLP (T1), arrives in segment 1, as denoted by the assertion of `p0_rx_st1_sop_o`.
   b. The signal `p0_rx_st1_hvalid_o` is asserted to validate the header of this TLP (T1H0) in the `p0_rx_st1_hdr_o` bus.
   c. The signal `p0_rx_st1_dvalid_o` is asserted to validate the data of this TLP (T1D0) in the `p0_rx_st1_data_o` bus.
   d. The signal `p0_rx_st2_dvalid_o` is asserted to validate the data of this TLP (T1D1) in the `p0_rx_st2_data_o` bus.
   e. The signal `p0_rx_st3_dvalid_o` is asserted to validate the data of this TLP (T1D2) in the `p0_rx_st3_data_o` bus.

4. Clock cycle 4:
   a. The end of the T1 TLP is in segment 0, denoted by the assertion of `p0_rx_st0_eop_o`.
   b. The signal `p0_rx_st0_dvalid_o` is asserted to validate the data of this TLP (T1D3) in the `p0_rx_st0_data_o` bus.
   c. The bus `p0_rx_st0_empty_o` indicates the number of dwords that are not valid in the `p0_rx_st0_data_o` bus (T1D3).

The next TLP arrives in the next clock cycle in segment 1 and finishes in segment 0.

Figure 31. Avalon Streaming RX Interface Timings



4.3.1.3.1. Application Logic Guidelines for the Avalon Streaming RX Interface

The following guidelines must be considered by the Application logic:

- The `pX_rx_st_ready_i` signal has to be always high. The buffer control and backpressure need to be handled with the RX Flow Control interface. Refer to RX Flow Control Interface for more details.
- The start of a packet (pX_rx_stN_sop_o) may occur in any of the segments (_stN_).
- For a single TLP spanning across multiple segments, the application logic needs to process the TLP in the order of the segment index (segment st0 → st1 → st2 → st3 → st0).

- For multiple TLPs arriving on the same clock cycle, the application logic needs to process the TLPs in the order of the segment index (i.e. segment st0 → st1 → st2 → st3 → st0).
- The R-Tile PCIe IP does not use segment 2 and segment 3 if segment 0 AND segment 1 are unused. Note that this behavior only applies in the following OPNs:
  - AGIx027R29AxxxxR2
  - AGIx027R29AxxxxR3
  - AGIx027R29BxxxxR3
  - AGIx023R18AxxxxR0
  - AGIx041R29DxxxxR0
  - AGIx041R29DxxxxR1

  For more details on OPN decoding, refer to the *Available Options* section of the Intel Intel Agilex 7 FPGAs and SoCs Device Overview.
- There is a maximum of three SOPs (pX_rx_stN_sop_o) in a single clock cycle. The following table describes the possible combinations across segments:

Table 56. Possible Combinations of Three pX_rx_stN_sop_o on a Single Clock Cycle

pX_rx_st0_sop_o	pX_rx_st0_eop_o	pX_rx_st1_sop_o	pX_rx_st1_eop_o	pX_rx_st2_sop_o	pX_rx_st2_eop_o	pX_rx_st3_sop_o	pX_rx_st3_eop_o
1'b1	1'b1	1'b1	1'b1	1'b1	1'b1	1'b0	1'b0
1'b1	1'b1	1'b1	1'b1	1'b1	1'b0	1'b0	1'b1
1'b1	1'b1	1'b1	1'b1	1'b1	1'b0	1'b0	1'b0
1'b1	1'b1	1'b1	1'b0	1'b0	1'b1	1'b1	1'b1
1'b1	1'b1	1'b1	1'b0	1'b0	1'b1	1'b1	1'b0
1'b1	1'b0	1'b0	1'b1	1'b1	1'b1	1'b1	1'b1
1'b1	1'b0	1'b0	1'b1	1'b1	1'b1	1'b1	1'b0

4.3.1.4. Avalon Streaming TX Interface

The Application Layer transfers data to the Transaction Layer of the R-Tile PCI Express IP core over the Avalon-ST TX interface. The R-Tile PCI Express IP core must assert pX_tx_st_ready_o before transmission begins.

If the R-Tile PCI Express IP core is configured in Configuration Mode 0 (1x16) with a double-width configuration, there are four segments with a 256-bit data width that allows multiple TLPs per cycle. This means there are four pX_tx_stN_sop_i signals and four pX_tx_stN_eop_i signals for Configuration Mode 0 (1x16).

This interface also does not follow a fixed latency between the pX_tx_st_ready_o and pX_tx_stN_dvalid_i signals as specified by the Avalon Interface Specifications.

The R-Tile PCI Express core when in Configuration Mode 0 (1x16) and in a double-width configuration provides four segments with each one having 256 bits of data (pX_tx_stN_data_i[255:0]), 128 bits of header (pX_tx_stN_hdr_i[127:0]), and 32 bits of TLP prefix (pX_tx_stN_prefix_i[31:0]). If the core is configured in Configuration Mode 0 (1x16), all four segments are used, so the data bus becomes a

1024-bit bus altogether, consisting of pX_tx_st0_data_i[255:0], pX_tx_st1_data_i[255:0], pX_tx_st2_data_i[255:0], and pX_tx_st3_data_1[255:0].

Parity generation is done via a 32:1 XOR (i.e. there is one parity bit for every 32 data, header or prefix bits).

Table 57. Avalon Streaming TX Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock Domain

pX_tx_stN_data_i[255:0] where
X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)
	Input	Application Layer data for transmission. The data bus is organized in multiple 256-bit segments. In x16 mode, all four segments are used to effectively form a 1024-bit data bus. In x8 mode, two segments are used to form a 512-bit data bus. In x4 mode, each 256-bit segment is an independent data bus.
		The Application Layer must provide a properly formatted TLP on the TX interface. The data is valid when the corresponding tx_stN_valid_i signal is asserted.
		The mapping of message TLPs is the same as the mapping of Transaction Layer TLPs with 4-dword headers. The number of data cycles must be correct for the length and address fields in the header. Issuing a packet with an incorrect number of data cycles results in the TX interface hanging and becoming unable to accept further requests.
		Note: There must be no Idle cycle between the tx_stN_sop_i and tx_stN_eop_i cycles unless there is backpressure with the deassertion of tx_st_ready_o.
			EP/RP/BP	coreclkout_hip

pX_tx_stN_hdr_i[127:0] where
X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)
	Input	This is the header to be transmitted, which follows the TLP header format of the PCIe specifications. Consider the following guidelines:
		• When the R-Tile Avalon Streaming Intel FPGA IP for PCIe is configured in EP or RP mode, it automatically calculates the Completer/Requester ID and the Application logic does not need to provide this information as part of the TLP header being transmitted. Note that this guideline does not apply when the IP is configured in TLP Bypass mode.
		• When the R-Tile Avalon Streaming Intel FPGA IP for PCIe is configured in EP mode and SR-IOV is enabled, follow the guidelines stated in the *BDF Assignments* section of SR-IOV Support Implementation.
		These signals are valid when the corresponding tx_stN_sop_i signal is asserted.
			EP/RP/BP	coreclkout_hip

pX_tx_stN_prefix_i[31:0] where
X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)
	Input	This is the TLP prefix to be transmitted, which follows the TLP prefix format of the PCIe specifications. PASID is supported.
		These signals are valid when the corresponding tx_stN_sop_i signal is asserted.
		The TLP prefix uses a Big Endian implementation (i.e. the Fmt field is in bits [31:29] and the Type field is in bits [28:24]).
			EP/RP/BP	coreclkout_hip

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
		If no prefix is present for a given TLP, that dword, including the Fmt field, is all zeros.		

pX_tx_stN_sop_i where
X = 0,1,2,3 (IP core number)
N = 0,2 (segment number)
	Input	Indicate the first cycle of a TLP when asserted in conjunction with the corresponding bit of tx_stN_valid_i. For the x16 configuration:
		• tx_st3_sop_i: When asserted, indicates the start of a TLP in tx_st3_data_i[255:0].
		• tx_st2_sop_i: When asserted, indicates the start of a TLP in tx_st2_data_i[255:0].
		• tx_st1_sop_i: When asserted, indicates the start of a TLP in tx_st1_data_i[255:0].
		• tx_st0_sop_i: When asserted, indicates the start of a TLP in tx_st0_data_i[255:0].
		These signals are asserted for one clock cycle per each TLP. They also qualify the corresponding tx_stN_hdr_i and tx_stN_tlp_prfx_i signals.
		Note: pX_tx_stN_sop_i pulses can only be sent on segments 0 or 2 (st0 or st2).
			EP/RP/BP	coreclkout_hip

pX_tx_stN_eop_i where
X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)
	Input	Indicate the last cycle of a TLP when asserted in conjunction with the corresponding bit of tx_stN_valid_i. For the x16 configuration:
		• tx_st3_eop_i: When asserted, indicates the end of a TLP in tx_st3_data_i[255:0].
		• tx_st2_eop_i: When asserted, indicates the end of a TLP in tx_st2_data_i[255:0].
		• tx_st1_eop_i: When asserted, indicates the end of a TLP in tx_st1_data_i[255:0].
		• tx_st0_eop_i: When asserted, indicates the end of a TLP in tx_st0_data_i[255:0].
		These signals are asserted for one clock cycle per each TLP.
			EP/RP/BP	coreclkout_hip

pX_tx_stN_dvalid_i where
X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)
	Input	Qualify the data of the corresponding segment of tx_stN_data_i into the IP core on ready cycles.
		To facilitate timing closure, Intel recommends that you register both the tx_st_ready_o and tx_stN_dvalid_i signals.
			EP/RP/BP	coreclkout_hip

pX_tx_stN_hvalid_i where
X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)
	Input	Qualify the header of the corresponding segment of tx_stN_data_i into the IP core on ready cycles.
		To facilitate timing closure, Intel recommends that you register both the tx_st_ready_o and tx_stN_hvalid_i signals.
			EP/RP/BP	coreclkout_hip

pX_tx_stN_pvalid_i where X = 0,1,2,3 (IP core number)	Input	Qualify the prefix of the corresponding segment of tx_stN_data_i into the IP core on ready cycles.	EP/RP/BP	coreclkout_hip

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
N = 0,1,2,3 (segment number)		To facilitate timing closure, Intel recommends that you register both the tx_st_ready_o and tx_stN_pvalid_i signals.		
pX_tx_stN_data_par_i[Z:0] where X = 0,1,2,3 (IP core number) and Z varies based on the core. N = 0,1,2,3 (segment number)	Input	Parity for tx_stN_data_i. Bit [0] corresponds to tx_stN_data_i[31:0], bit [1] corresponds to tx_stN_data_i[63:32], and so on. By default, the PCIe Hard IP generates the parity for the TX data.	EP/RP/BP	coreclkout_hip

pX_tx_stN_hdr_par_i[3:0] where
X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)
	Input	Parity for tx_stN_hdr_i.
		By default, the PCIe Hard IP generates the parity for the TX header.
			EP/RP/BP	coreclkout_hip

pX_tx_stN_prefix_par_i where
X = 0,1,2,3 (IP core number)
N = 0,1,2,3 (segment number)
	Input	Parity for tx_stN_tlp_prfx_i.
		By default, the PCIe Hard IP generates the parity for the TX TLP prefix.
			EP/RP/BP	coreclkout_hip

pX_tx_st_ready_o where X = 0,1,2,3 (IP core number)	Output	Indicates that the PCIe Hard IP is ready to accept data. When tx_st_ready_o is driven high by the R-Tile Avalon Streaming Intel FPGA IP for PCIe, the Application logic may assert high the tx_stN_valid_i signal and transfer data. When tx_st_ready_o is driven low by the R-Tile Avalon Streaming Intel FPGA IP for PCIe, the Application logic must drive low the tx_stN_valid_i within a maximum of 16 clock cycles. Refer to Avalon Streaming TX Interface pX_tx_st_ready_o Behavior for additional information. The pX_tx_st_ready_o signal can be deasserted in the following conditions: • The LTSSM is not in L0 state. • A TLP retry is in progress. • The R-Tile Avalon-ST IP is busy sending internally generated TLPs. • The internal R-Tile TX FIFO is full.	EP/RP/BP	coreclkout_hip

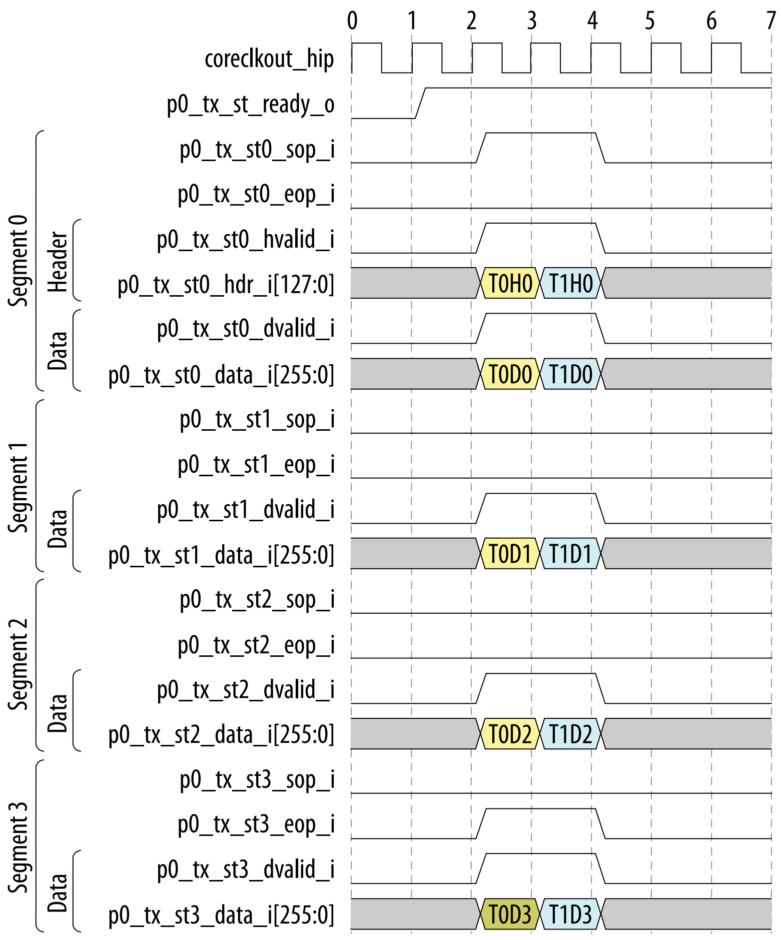
As an example, Avalon Streaming TX Interface Timings below shows the behavior of the Avalon Streaming TX interface in a back-to-back TLPs scenario with data spanning across multiple segments. The following text describes the waveforms per clock cycle:

1. Clock cycle 1: The R-Tile Intel FPGA IP for PCI Express asserts p0_tx_st_ready_o signal, indicating the Hard IP is ready to accept TLPs from the Application logic.
2. Clock cycle 2:

a. The start of the first TLP (T0) is in segment 0, indicated by the assertion of `p0_tx_st0_sop_i`.
b. The signal `p0_tx_st0_hvalid_i` is asserted to validate the header of this first TLP (T0H0) in the `p0_tx_st0_hdr_i` bus.
c. The signal `p0_tx_st0_dvalid_i` is asserted to validate the data of this first TLP (T0D0) in the `p0_tx_st0_data_i` bus.
d. The signal `p0_tx_st1_dvalid_i` is asserted to validate the next portion of the data of this first TLP (T0D1) in the `p0_tx_st1_data_i` bus.
e. The signal `p0_tx_st2_dvalid_i` is asserted to validate the next portion of the data of this first TLP (T0D2) in the `p0_tx_st2_data_i` bus.
f. The signal `p0_tx_st3_dvalid_i` is asserted to validate the final portion of the data of this first TLP (T0D3) in the `p0_tx_st3_data_i` bus.
g. The end of this first TLP (T0) is in segment 3, denoted by the assertion of `p0_tx_st3_eop_i`.

3. Clock cycle 3:
    a. The next TLP (T1), arrives in segment 0, as denoted by `p0_tx_st0_sop_i` staying high.
    b. The signal `p0_tx_st0_hvalid_i` is asserted to validate the header of this TLP (T1H0) in the `p0_tx_st0_hdr_i` bus.
    c. The signal `p0_tx_st0_dvalid_i` is asserted to validate the data of this TLP (T1D0) in the `p0_tx_st0_data_i` bus.
    d. The signal `p0_tx_st1_dvalid_i` is asserted to validate the next portion of the data of this TLP (T1D1) in the `p0_tx_st1_data_i` bus.
    e. The signal `p0_tx_st2_dvalid_i` is asserted to validate the next portion of the data of this TLP (T1D2) in the `p0_tx_st2_data_i` bus.
    f. The signal `p0_tx_st3_dvalid_i` is asserted to validate the final portion of the data of this TLP (T1D2) in the `p0_tx_st3_data_i` bus.
    g. The end of this TLP (T1) is in segment 3, denoted by `p0_tx_st3_eop_i` staying high.

Figure 32. Avalon Streaming TX Interface Timings



Note: For Configuration Mode 0 (1x16), the start of a TLP (`pX_tx_stN_sop_i`) can only happen on segment 0 (`st0`) or segment 2 (`st2`).

4.3.1.4.1. Application Logic Guidelines for the Avalon Streaming TX Interface

The following guidelines must be considered by the Application logic:

- Application logic must adhere to the requirements for the `pX_tx_st_ready_o` signal behavior as outlined in Avalon Streaming TX Interface `pX_tx_st_ready_o` Behavior.
- Transmission of a TLP must be uninterrupted when `pX_tx_st_ready_o` is asserted. The application must not deassert `pX_tx_stN_valid_i` between `pX_tx_stN_sop_i` and `pX_tx_stN_eop_i` unless there is backpressure from the R-Tile PCIe IP core indicated by the deassertion of `pX_tx_st_ready_o`.

  Note: Failing to meet this guideline may cause the transmission of a TLP with an invalid LCRC.
- For the Configuration Mode 0 (1x16) in double-width mode, the start of a TLP (`pX_tx_stN_sop_i`) can only happen in segment 0 (st0) or segment 2 (st2) (i.e. a given TLP cannot start on segment 1 or segment 3).

- For Configuration Mode 0 (1x16) in double-width mode, the header for segment 2 (st2_hdr) is allowed depending on the utilization for segment 0 and segment 1. Refer to the table below for the allowed conditions. Note that the table does not include all the signals for the Avalon Streaming TX interface. It only shows the relevant signals to highlight the valid cases where a TLP can be started on segment 2.

Table 58. Possible Combinations for the pX_tx_st2_sop_i in Configuration Mode 0 (1x16) Double-width Mode

pX_tx_st0_sop_i	pX_tx_st0_eop_i	pX_tx_st0_hvalid_i	pX_tx_st0_dvalid_i	pX_tx_st1_sop_i	pX_tx_st1_eop_i	pX_tx_st1_hvalid_i	pX_tx_st1_dvalid_i	pX_tx_st2_sop_i	pX_tx_st2_hvalid_i	pX_tx_st2_dvalid_i
1'b1	1'b1	1'b1	1'b1	1'b0	1'b0	1'b0	1'b0	1'b1	1'b1	1'b1
1'b1	1'b0	1'b1	1'b1	1'b0	1'b1	1'b0	1'b1	1'b1	1'b1	1'b1
1'b0	1'b1	1'b0	1'b1	1'b0	1'b0	1'b0	1'b0	1'b1	1'b1	1'b1
1'b0	1'b0	1'b0	1'b1	1'b0	1'b1	1'b0	1'b1	1'b1	1'b1	1'b1

- For Configuration Mode 1 (2x8) and Configuration Mode 2 (4x4) in double-width mode, the header for segment 1 (st1_hdr) is allowed depending on the utilization for segment 0. Refer to the table below for the allowed conditions. Note that the table does not include all the signals for the Avalon Streaming TX interface. It only shows the relevant signals to highlight the valid cases where a TLP can be started on segment 1.

Table 59. Possible Combinations for the pX_tx_st1_sop_i in Configuration Mode 1 (2x8) and Configuration Mode 2 (4x4), Both in Double-width Mode

pX_tx_st0_sop_i	pX_tx_st0_eop_i	pX_tx_st0_hvalid_i	pX_tx_st0_dvalid_i	pX_tx_st1_sop_i	pX_tx_st1_hvalid_i	pX_tx_st1_dvalid_i
1'b1	1'b1	1'b1	1'b1	1'b1	1'b1	1'b1
1'b0	1'b1	1'b0	1'b1	1'b1	1'b1	1'b1

- For a single TLP spanning across multiple segments, the application logic needs to send the TLP in the order of the segment index (segment st0 → st1 → st2 → st3 → st0).
- If the TLP length of the TLP being transmitted is greater than the segment size, the segment used to assert the pX_tx_stN_eop_i signal is dictated by the TLP length.
- If the TLP length being transmitted is less than the segment size (255 bits), the corresponding pX_tx_stN_eop_i signal needs to happen in the same segment where pX_tx_stN_sop_i is being asserted.
- The maximum number of clock cycles allowed between the deassertion of pX_tx_st_ready_o and pX_tx_stN_valid_i is 16 coreclkout_hip cycles.
- For Configuration Mode 0 (1x16) in single-width mode, only one segment can be used per clock cycle (i.e. st0_hdr/st0_data or st1_hdr/st1_data). In addition, If segment 1 is used, st0_data must be used by the previous TLP.

4.3.1.4.2. Avalon Streaming TX Interface pX_tx_st_ready_o Behavior

The following timing diagram illustrates the behavior of pX_tx_st_ready_o, which is deasserted to pause the data transmission to the R-Tile PCI Express IP core, and then reasserted. The application logic deasserts pX_tx_stN_valid_i 16 clock cycles after pX_tx_st_ready_o is deasserted (from the point marked with the letter **a** to the point marked with the letter **b**). This is the maximum number of clock cycles allowed between the deassertion of pX_tx_st_ready_o and pX_tx_stN_valid_i.

When the R-Tile PCI Express IP Core reasserts the pX_tx_st_ready_o signal (point marked with letter **c** to point marked with letter **d**), the following two cases must be considered:

Case 1: If there is a TLP suspended due to the deassertion of pX_tx_st_ready_o, then the maximum number of clock cycles for pX_tx_stN_valid_i to go high after the assertion of pX_tx_st_ready_o is one (as illustrated in the following figure).

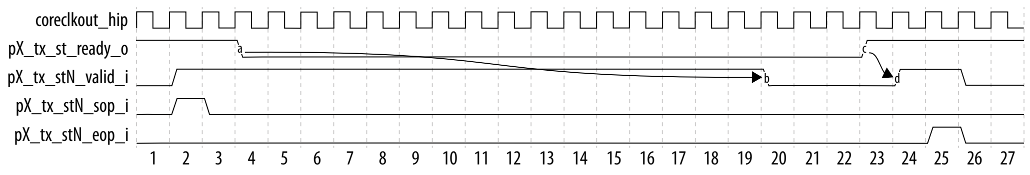
Case 2: If there is no TLP suspended due to the deassertion of pX_tx_st_ready_o, then there is no requirement, and the application logic can reassert pX_tx_stN_valid_i as soon as it has a TLP available to transmit.

The application must not deassert pX_tx_stN_valid_i between pX_tx_stN_sop_i and pX_tx_stN_eop_i unless there is backpressure from the R-Tile PCIe IP indicated by the deassertion of pX_tx_st_ready_o.

Note: Failing to meet this guideline may cause the transmission of a TLP with an invalid LCRC.

Note: This is an additional requirement for the R-Tile PCI Express IP core that does not follow the Avalon-ST standard.

Figure 33. Avalon Streaming TX Interface pX_tx_st_ready_o Behavior



4.3.1.5. Tag Allocation

For the x16 Controller (Port 0), the R-Tile Avalon Streaming Hard IP supports the 10-bit tag Requester capability. It supports up to 768 outstanding Non-Posted Requests (NPRs) with valid tag values ranging from 256 to 1023.

For the x8 (Port 1) and x4 Controllers (Port 2/3) the Hard IP supports the 10-bit tag Requester capability. They support up to 512 outstanding Non-Posted Requests (NPRs) with valid tag values ranging from 256 to 768.

When enabling both 10-bit tags and 8-bit tags, the LSB 8 bits of the 8-bit tags cannot be shared with the LSB 8 bits of the 10-bit tags.

Note that all PFs and their associated VFs share the same tag space. This means that different PFs and VFs cannot have outstanding tags having the same tag values.

In the TLP bypass mode, there is no restriction on the tag allocation since the R-Tile PCIe Hard IP does not do any tag management. Hence, 10-bit tags can be used without any restriction across all the cores.

4.3.1.5.1. Completion Buffer Size

R-Tile Hard IP for PCIe implements Completion (Cpl) buffers for header and data for each PCIe core. In Endpoint mode, when Completion credits are infinite, user application needs to manage the number of outstanding requests to prevent overflow and lost Completions.

Table 60. Completion Buffer Size

Completion Buffer	OPNs = AGIx027R29AxxxxR0, AGIx027R29AxxxxR1		OPNs = AGIx027R29AxxxxR2, AGIx027R29AxxxxR3, AGIx027R29BxxxxR3, AGIx023R18AxxxxR0, AGIx041R29DxxxxR0, AGIx041R29DxxxxR1	
	Depth	Width	Depth	Width
Port 0 Cpl header	572	N/A	1444	N/A
Port 0 Cpl data	2016	512	2016	512
Port 1 Cpl header	572	N/A	1144	N/A
Port 1 Cpl data	2016	256	2016	256
Port 2 Cpl header	286	N/A	572	N/A
Port 2 Cpl data	1730	128	2016	128
Port 3 Cpl header	286	N/A	572	N/A
Port 3 Cpl data	1730	128	2016	128

Application Layer should send a request when there are sufficient completion buffer entries. For a Memory Read request, the Completer is allowed to break up the response for a single Read Request into multiple Completions, so the Application Layer needs to account for the maximum possible Completions returned.

Below are a few examples for the amount of Completion buffer entries that would be required based on the Memory Read request size and the Read Completion Boundary (RCB) configuration.

Table 61. Completion Buffer Entries Examples

Memory Read Request Examples	Completion Buffer Entries Required					
	Port 0		Port 1		Port 2/3	
	Header	Data	Header	Data	Header	Data
A Memory Read request with an address of 1_0000h and length of C0h bytes (192 decimal) can be completed by a Root Complex with an RCB value of 64 bytes with one of the following combinations of Completions (bytes):	3	3	3	6	3	12

continued...

Memory Read Request Examples	Completion Buffer Entries Required					
	Port 0		Port 1		Port 2/3	
	Header	Data	Header	Data	Header	Data
• 192 • 128, 64 • 64, 128 • **64, 64, 64 → Maximum number of completions**						
A Memory Read request with an address of 1_0000h and length of C0h bytes (192 decimal) can be completed by a Root Complex with an RCB value of 128 bytes with one of the following combinations of Completions (bytes): • 192 • **128, 64 → Maximum number of completions**	2	3	2	6	2	12
A Memory Read request with an address of 1_0020h and length of 100h bytes (256 decimal) can be completed by a Root Complex with an RCB value of 64 bytes with one of the following combinations of Completions (bytes): • 256 • 32, 224 • 32, 64, 160 • 32, 64, 64, 96 • **32, 64, 64, 64, 32 → Maximum number of completions** • 32, 64, 128, 32 • 32, 128, 96 • 32, 128, 64, 32 • 96, 160 • 96, 128, 32 • 96, 64, 96 • 96, 64, 64, 32 • 160, 96 • 160, 64, 32 • 224, 32	5	4	5	8	5	16

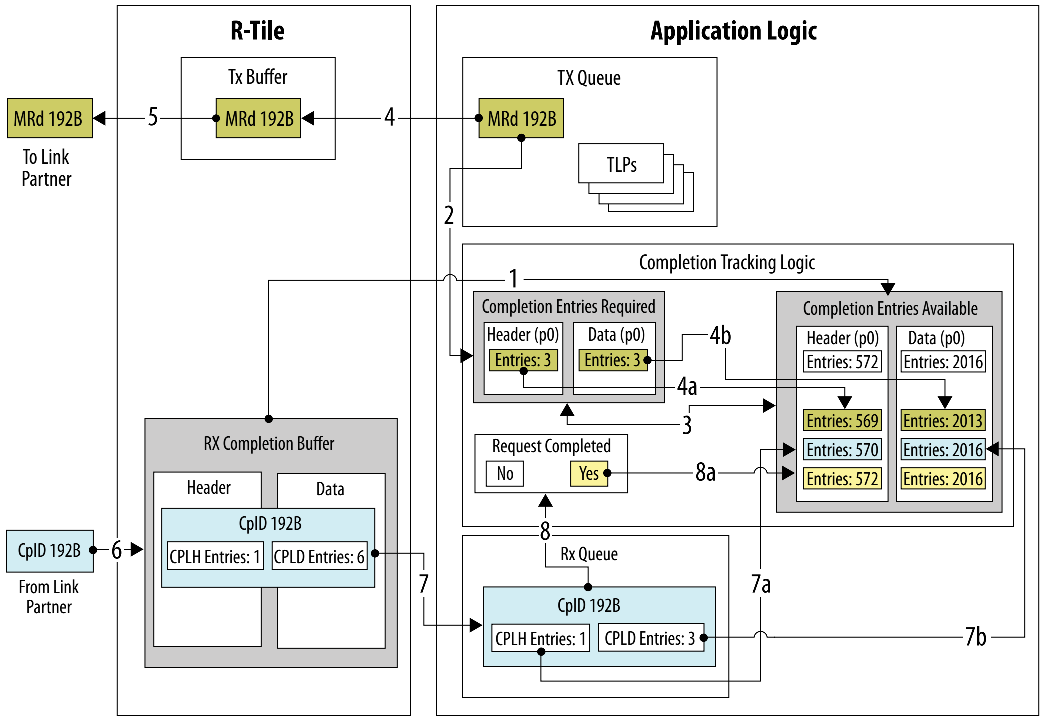
Based on the examples from the table above, the Suggested Flow for Application Logic to Track Completion Buffer Entries and Schedule NP Requests to R-Tile IP for PCIe below provides the flow in the Application logic to track the completion buffer entries and based on this, schedule Non-Posted (NP) requests to the R-Tile Intel FPGA IP for PCI Express.

For illustration purposes, Suggested Flow for Application Logic to Track Completion Buffer Entries and Schedule NP Requests to R-Tile IP for PCIe is showing the R-Tile Intel FPGA IP for PCIe express configured in Configuration Mode 0 (1x16), with the Application logic requesting a Memory Read of 192 Bytes and getting a single completion of 192 Bytes in the response from the link partner.

The flow is as follows:

1. When `dl_up` signal goes from LOW to HIGH, the Application logic initializes the available completion buffer entries based on CPL buffer size table above (Completion Buffer Size).
2. When there is an NP request added to transmission queue of the Application logic, the Completion Tracking logic calculates the required completion buffer entries for the request, considering the maximum number of completions that could be received. For this example, it is 3 completions of 64 Bytes. (Refer to the table with the examples provided in Completion Buffer Entries Examples).
3. The Completion Tracking logic checks the required completion buffer entries against the available completion buffer entries.
4. If there are sufficient buffer entries available, the Application logic schedules the Memory Read request for transmission.
    a. The Completion Tracking logic updates the available completion header buffer entries.
    b. The Completion Tracking logic updates the available completion data buffer entries.
5. The R-Tile Intel FPGA IP for PCI Express sends the NP TLP to the link partner.
6. The link partner responds to the MRd request with a single Completion with a data payload of 192B.
7. When the Completion packet is received by the Application Layer in its Rx Queue,
    a. The Completion Tracking logic updates the available completion header buffer entries.
    b. The Completion Tracking logic updates the available completion data buffer entries.
8. The Completion Tracking logic verifies if the original NP request has been fully completed with the Completion received.
    a. If the request has been completed, the Completion Tracking logic adjusts the available completion buffer entries based on the unused entries allocated to the request previously.
    b. In case of multiple Completions for steps 8 and 8a, the Completion Tracking Logic will update the completion buffer entries for each Completion TLP received accordingly.

Figure 34. Suggested Flow for Application Logic to Track Completion Buffer Entries and Schedule NP Requests to R-Tile IP for PCIe



4.3.2. Precision Time Measurement (PTM) Interface (Endpoint Only)

PTM Interface Signals shows the PTM interface signals between the IP and the FPGA Application logic.

The following consideration must be taken into account when implementing the PTM feature with the R-Tile Avalon streaming Intel FPGA IP for PCIe:

- PTM is only supported in Configuration Mode 0 (1x16) or in Configuration Mode 1 (2x8).
- PTM accuracy in the common clock scheme is +/-50 ns.
- PTM accuracy in the separate clock scheme is +/-100 ns.

Table 62. PTM Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock
`pX_ptm_context_valid_o`	Output	When this signal is asserted, it indicates that the value present on the ptm_time bus is valid. Hardware will deassert this bit	EP	`coreclkout_hip`

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock
		whenever a PTM dialogue is requested and an update is in progress.		
pX_clk_updated_o	Output	This one clock pulse is an indication that the PTM dialogue has completed and the results of that operation have been driven on the ptm_time bus.	EP	coreclkout_hip
pX_ptm_local_clock_o[63:0]	Output	This bus contains the calculated master time at t1' as indicated in the PCIe spec plus any latency to do the calculation and to drive the value to the requester.	EP	coreclkout_hip
pX_ptm_manual_update_i	Input	Asserted high for one coreclkout_hip clock when the user application wants to request a PTM handshake to get a snapshot of the latest time.	EP	coreclkout_hip

For more details, refer to *Section 6.22 Precision Time Measurement (PTM) Mechanism* of the *PCI Express Base Specification Revision 5.0 Version 1.0*.

4.3.3. Interrupt Interface

The R-Tile Avalon-ST IP for PCI Express supports Message Signaled Interrupts (MSI), MSI-X interrupts, and legacy interrupts. MSI and legacy interrupts are mutually exclusive.

The user application generates MSI which are single-Dword memory write TLPs to implement interrupts. This interrupt mechanism conserves pins because it does not use separate wires for interrupts. In addition, the single Dword provides flexibility for the data presented in the interrupt message. The MSI Capability structure is stored in the Configuration Space and is programmed using Configuration Space accesses. The user application generates MSI-X messages which are single-Dword memory writes. The MSI-X Capability structure points to an MSI-X table structure and an MSI-X Pending Bit Array (PBA) structure which are stored in memory. This scheme is different than the MSI Capability structure, which contains all the control and status information for the interrupts.

Enable legacy interrupts by programming the Interrupt Disable bit (bit[10]) of the Configuration Space Command to 1'b0. When legacy interrupts are enabled, the IP core emulates INTx interrupts using virtual wires. The `app_int_i` ports control legacy interrupt generation.

4.3.3.1. Legacy Interrupts

Legacy interrupts mimic the original PCI level-sensitive interrupts using virtual wire messages. The R-Tile IP for PCIe signals legacy interrupts on the PCIe link using Message TLPs. The term INTx refers collectively to the four legacy interrupts, INTA#,INTB#, INTC# and INTD#. The R-Tile IP for PCIe asserts `app_int_i` to cause an Assert_INTx Message TLP to be generated and sent upstream. A deassertion of `app_int_i`, i.e a transition of this signal from high to low, causes a Deassert_INTx

Message TLP to be generated and sent upstream. To use legacy interrupts, you must clear the Interrupt Disable bit, which is bit 10 of the Command Register in the configuration header. Then, you must turn off the MSI Enable bit.

Table 63. Legacy Interrupts

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
pX_app_int_i[7:0]	Input	When asserted, these signals indicate an assertion of an INTx message is requested. A transition from high to low indicates a deassertion of the INTx message is requested. This bus is for Endpoints only. Each bit is associated with a corresponding physical function.	EP	slow_clk
pX_app_int_ready_o[7:0]	Output	One bit per physical function. The new app_int_i value should be held until app_int_ready_o=1.	EP	slow_clk
pX_irq_status_o	Output	These signals drive legacy interrupts to the Application Layer in Root Port mode. The source of the interrupt is logged in the Root Port Interrupt Status registers in the Port Configuration and Status registers.	RP	slow_clk

4.3.3.2. MSI

MSI interrupts are signaled on the PCI Express link using a single-dword Memory Write TLP. The user application issues an MSI request (MWr) through the Avalon-ST interface and updates the configuration space register using the MSI interface.

Note: Only Ports 0 and 1 support MSI.

Table 64. MSI Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
pX_msi_pnd_func_i[2:0]	Input	Function number select for the Pending Bits register in the MSI capability structure.	EP/BP	slow_clk
pX_msi_pnd_addr_i[1:0]	Input	Byte select for the Pending Bits Register in the MSI Capability Structure. For example, if msi_pnd_addr_i[1:0] = 00, bits [7:0] of the Pending Bits register will be updated with msi_pnd_byte_i[7:0]. If msi_pnd_addr_i[1:0] = 01, bits [15:8] of the Pending Bits register will be updated with msi_pnd_byte_i[7:0].	EP/BP	slow_clk
pX_msi_pnd_byte_i[7:0]	Input	Indicate that function has a pending associated message.	EP	slow_clk
pX_msi_pnd_ready_o	Output	A value of 0 indicates the endpoint may be servicing another message, and unable to service this master immediately.	EP/BP	slow_clk

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
		A new MSI event should be held until `msi_pnd_ready_o` = 1.		

4.3.3.3. MSI-X

The R-Tile IP for PCIe provides a Configuration Intercept Interface. User soft logic can monitor this interface to get MSI-X Enable and MSI-X function mask related information. User application logic needs to implement the MSI-X tables for all PFs and VFs at the memory space pointed to by the BARs as a part of your Application Layer.

For more details on the MSI-X related information that you can obtain from the Configuration Intercept Interface, refer to the MSI-X Registers section in the Registers chapter.

MSI-X is an optional feature that allows the user application to support large amount of vectors with independent message data and address for each vector.

When MSI-X is supported, you need to specify the size and the location (BARs and offsets) of the MSI-X table and PBA. MSI-X can support up to 2048 vectors per function versus 32 vectors per function for MSI.

A function is allowed to send MSI-X messages when MSI-X is enabled and the function is not masked. The application uses the Configuration Intercept Interface to access this information.

Note: Only Ports 0 and 1 support MSI-X.

When the application needs to generate an MSI-X, it will use the contents of the MSI-X Table (Address and Data) and generate a Memory Write through the Avalon-ST interface.

You can enable MSI-X interrupts by turning on the **Enable MSI-X** option in the **MSI-X** tab under the **PCI Express/PCI Capabilities** tab in the parameter editor. If you turn on the **Enable MSI-X** option, you should implement the MSI-X table structures at the memory space pointed to by the BARs as a part of your Application Layer.

The MSI-X Capability Structure contains information about the MSI-X Table and PBA Structure. For example, it contains pointers to the bases of the MSI-X Table and PBA Structure, expressed as offsets from the addresses in the function's BARs. The Message Control register within the MSI-X Capability Structure also contains the MSI-X Enable bit, the Function Mask bit, and the size of the MSI-X Table.

MSI-X interrupts are standard Memory Writes, therefore Memory Write ordering rules apply.

Example:

Table 65. MSI-X Configuration

MSI-X Vector	MSI-X Upper Address	MSI-X Lower Address	MSI-X Data
0	0x00000001	0xAAAA0000	0x00000001
1	0x00000001	0xBBBB0000	0x00000002
2	0x00000001	0xCCCC0000	0x00000003

Table 66. PBA Table

PBA Table	PBA Entries
Offset 0	0x0

If the application needs to generate an MSI-X interrupt (vector 1), it will read the MSI-X Table information, generate a MWR TLP through the Avalon-ST interface and assert the corresponding PBA bits (bit[1]) in a similar fashion as for MSI generation.

The generated TLP will be sent to address 0x00000001_BBBB0000 and the data will be 0x00000002. When the MSI-X has been sent, the application can clear the associated PBA bits.

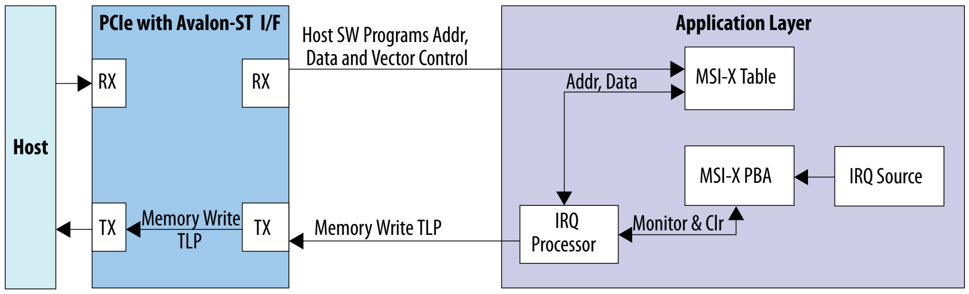
Related Information

Implementing MSI-X for PCI Express in Intel FPGA Devices

4.3.3.3.1. Implementing MSI-X Interrupts

Section 6.8.2 of the *PCI Local Bus Specification* describes the MSI-X capability and table structures. The MSI-X capability structure points to the MSI-X Table structure and MSI-X Pending Bit Array (PBA) registers. The BIOS sets up the starting address offsets and BAR associated with the pointer to the starting address of the MSI-X Table and PBA registers.

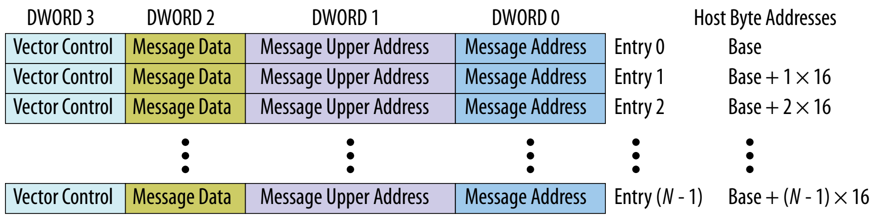
Figure 35. MSI-X Interrupt Components



1. Host software sets up the MSI-X interrupts in the Application Layer by completing the following steps:
   a. Host software reads the `Message Control` register at 0x050 register to determine the MSI-X Table size. The number of table entries is the *<value read> + 1*.

      The maximum table size is 2048 entries. Each 16-byte entry is divided in 4 fields as shown in the figure below. The MSI-X table can be accessed on any BAR configured. The base address of the MSI-X table must be aligned to a 4 KB boundary.
   b. The host sets up the MSI-X table. It programs MSI-X address, data, and masks bits for each entry as shown in the figure below.

Figure 36. Format of MSI-X Table


DWORD 3
DWORD 2
DWORD 1
DWORD 0
Host Byte Addresses
Vector Control
Message Data
Message Upper Address
Message Address
Entry 0
Base
Entry 1
Base + 1 × 16
Entry 2
Base + 2 × 16
Entry (N - 1)
Base + (N - 1) × 16

c. The host calculates the address of the <n^{th}> entry using the following formula:

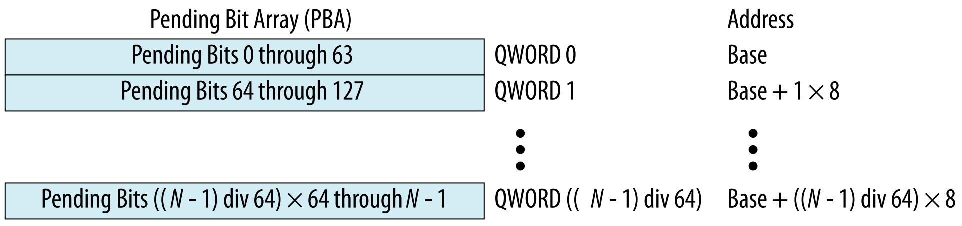
```
nth_address = base address[BAR] + 16<n>
```

2. When Application Layer has an interrupt, it drives an interrupt request to the IRQ Source module.
3. The IRQ Source sets appropriate bit in the MSI-X PBA table.

   The PBA can use qword or dword accesses. For qword accesses, the IRQ Source calculates the address of the <m^{th}> bit using the following formulas:

```
qword address = <PBA base addr> + 8(floor(<m>/64))
qword bit = <m> mod 64
```

Figure 37. MSI-X PBA Table


Pending Bit Array (PBA)
Address
Pending Bits 0 through 63
QWORD 0
Base
Pending Bits 64 through 127
QWORD 1
Base + 1 × 8
Pending Bits ((N - 1) div 64) × 64 through N - 1
QWORD ((N - 1) div 64)
Base + ((N - 1) div 64) × 8

4. The IRQ Processor reads the entry in the MSI-X table.
   a. If the interrupt is masked by the `Vector_Control` field of the MSI-X table, the interrupt remains in the pending state.
   b. If the interrupt is not masked, IRQ Processor sends Memory Write Request to the TX slave interface. It uses the address and data from the MSI-X table. If **`Message Upper Address`** = 0, the IRQ Processor creates a three-dword header. If the **`Message Upper Address`** `> 0`, it creates a 4-dword header.
5. The host interrupt service routine detects the TLP as an interrupt and services it.

Related Information

- Floor and ceiling functions
- PCI Local Bus Specification, Rev. 3.0

4.3.4. Hard IP Reconfiguration Interface

This interface is an Avalon-MM slave interface with a 32-bit address and an 8-bit data bus. You can use this interface to dynamically modify the value of configuration registers. Note that after a warm reset or cold reset, changes made to the configuration registers of the Hard IP via this interface are lost as these registers revert back to their default values.

Note: This interface can be used in Endpoint, Root Port and TLP Bypass modes. However, it must be enabled if Root Port or TLP Bypass mode is selected.

Note: If implementing temperature monitoring, the Application logic must prevent a temperature readout to the R-Tile temperature sensing diode (TSD) while a read or write transaction on the Hard IP Reconfiguration interface is in progress. Refer to the Intel Agilex 7 F-Series and I-Series Power Management User Guide for additional information.

In Root Port mode, the application logic uses this interface to access its PCIe configuration space to perform link control functions (such as Hot Reset, link disable, or link retrain).

In TLP Bypass mode, the Hard IP forwards the received Type0/1 Configuration request TLPs to the application logic, which must respond with Completion TLPs with a status of Successful Completion (SC), Unsupported Request (UR), Configuration Request Retry Status (CRS), or Completer Abort (CA). If a received Configuration request TLP needs to update a PCIe configuration space register, the application logic needs to use the Hard IP Reconfiguration interface to access that PCIe configuration space register.

Table 67. Hard IP Reconfiguration Interface Signals

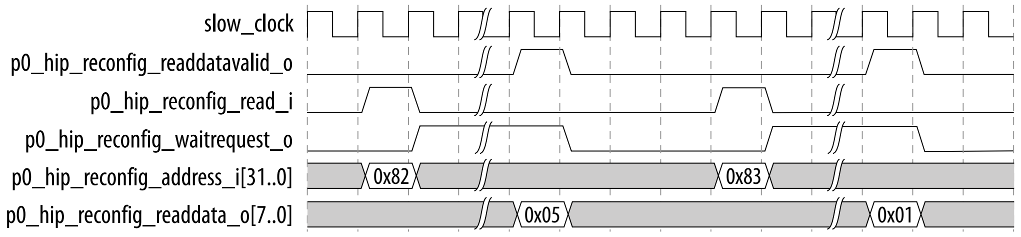
Signal Name	Direction	Description	EP/RP/BP	Clock
`pX_hip_reconfig_readdata_o[7:0]` where X = 0, 1, 2, 3	Output	Avalon-MM read data outputs	EP/RP/BP	`slow_clk`
`pX_hip_reconfig_readdatavalid_o` where X = 0, 1, 2, 3	Output	Avalon-MM read data valid. When asserted, this signal indicates that the data on `hip_reconfig_readdata_o[7:0]` is valid.	EP/RP/BP	`slow_clk`
`pX_hip_reconfig_write_i` where X = 0, 1, 2, 3	Input	Avalon-MM write enable	EP/RP/BP	`slow_clk`
`pX_hip_reconfig_read_i` where X = 0, 1, 2, 3	Input	Avalon-MM read enable. *Note:* This interface is not pipelined. User application must wait for the return of the data from the current read (on `hip_reconfig_readdata_o[7:0]`) before starting another read operation.	EP/RP/BP	`slow_clk`
`pX_hip_reconfig_address_i[31:0] where X = 0, 1, 2, 3`Hard IP Reconfiguration Interface When Performing a Read Operation	Input	Avalon-MM reconfiguration address.	EP/RP/BP	`slow_clk`

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock
		Note: Following is the mapping for the `pX_hip_reconfig_address_i[31:0]` bus: • Virtual Function Number: `pX_hip_reconfig_address_i[31:21]`. • Reserved (must be set to 0): `pX_hip_reconfig_address_i[20]` • Physical Function Number: `pX_hip_reconfig_address_i[19:17]`. • Virtual Function Active: `pX_hip_reconfig_address_i[16]`. • Register Offset (please refer to the column **Address Offset** that is available in the linked collateral in Configuration Space Registers): `pX_hip_reconfig_address_i[15:0]` • **Example 1**: To access the MSI-X Capability Structure for PF1 with SR-IOV disabled, set the `pX_hip_reconfig_address_i` to 0x0002_00B0. • **Example 2**: To access the MSI-X Capability Structure for PF0, set the `pX_hip_reconfig_address_i` to 0x0021_00B0.		
`pX_hip_reconfig_writedata_i[7:0] where X = 0, 1, 2, 3`	Input	Avalon-MM write data inputs	EP/RP/BP	`slow_clk`
`pX_hip_reconfig_waitrequest_o where X = 0, 1, 2, 3`	Output	When asserted, this signal indicates the IP core is not ready to respond to a request.	EP/RP/BP	`slow_clk`

As an example, Hard IP Reconfiguration Interface when performing a read operation shows the behavior of the Hard IP Reconfiguration Interface when performing a read operation to the Current Link Speed and Negotiated Link Width fields of the Link Status Register, when the R-Tile Avalon Streaming Intel FPGA IP for PCIe is configured in Gen5 x16 mode with a single physical function enabled. For additional details on configuration space registers, refer to Configuration Space Registers.

Figure 38. Hard IP Reconfiguration Interface when performing a read operation


slow_clock
p0_hip_reconfig_readdatavalid_o
p0_hip_reconfig_read_i
p0_hip_reconfig_waitrequest_o
p0_hip_reconfig_address_i[31..0]
p0_hip_reconfig_readdata_o[7..0]
0x82
0x83
0x05
0x01

4.3.5. Error Interface

This is an optional interface that allows the Application Layer to report errors to the IP core and vice versa. Specifically, the Application Layer can report the different types of errors defined by the `app_error_info_i` signal to the IP. For Advanced Error Reporting (AER), the Application Layer can provide the information to log the TLP header and the error log request via the `app_err_*` interface.

Note: The AER capability for Physical Functions (PFs) is enabled by default. There is no AER implementation for Virtual Functions (VFs). Use the VF Error Flag Interface instead of AER when using VFs.

Table 68. Error Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
`pX_serr_out_o`	Output	Indicates if a system error is detected. **EP mode**: This signal is asserted when the R-Tile PCIe Hard IP sends a message of a correctable/non-fatal/ fatal error. **RP mode**: A one-clock-cycle pulse on this signal indicates if any device in the hierarchy reports any of the following errors and the associated enable bit is set in the Root Port Control register: ERR_COR, ERR_FATAL, ERR_NONFATAL. This signal is also asserted when an internal error is detected. The source of the error will be logged in the Root Port Error Status registers in the Port Configuration and Status registers. **BP mode**: The transaction layer or data link layer errors detected by the Hard IP core trigger this signal. Detailed information are logged in the Bypass Mode Error Status registers in the Port Configuration and Status registers. **All modes**: This signal is asserted to indicate if a parity error was detected for the received data or for the data crossing between the R-Tile Hard IP and the FPGA Core. **For Tx**: If the GUI option **Enable byte parity ports on Avalon-ST interface** is enabled, there will be a parity check using the parity bits provided on the ports `pX_tx_stN_data_par_i`, `pX_tx_stN_hdr_par_i` and `pX_tx_stN_prefix_par_i`. **For Rx**: There will be a parity check internal to the R-Tile for the data being received. Additionally, If the GUI option **Enable byte parity ports on Avalon-ST interface** is enabled, parity bits will be provided on the	EP/RP/BP	`slow_clk`

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
		ports pX_rx_stN_data_par_o, pX_rx_stN_hdr_par_o and pX_rx_stN_prefix_par_o, to allow an additional parity check in the Application logic to verify data integrity after crossing the EMIB. In case of an error, you can use the Hard IP Reconfiguration interface to read address 0x1319 for additional information on the type of error: [0]: rx_correctable_err [1]: rx_uncorrectable_err [2]: rx_parity_err [3]: tx_correctable_err [4]: tx_uncorrectable_err [5]: tx_parity_err at the front end of the datapath inside the Hard IP. [6]: tx_parity_err at the back end of the datapath inside the Hard IP.		
pX_app_err_valid_i	Input	A one-cycle pulse on this signal indicates that the data on app_err_info_i, app_err_hdr_i, and app_err_func_num_i are valid in that cycle and app_err_hdr_i will be valid during the following four cycles.	EP/RP	slow_clk
pX_app_err_hdr_i[31:0]	Input	This bus contains the header and TLP prefix information for the error TLP. The 128-bit header and 32-bit TLP prefix are sent to the Hard IP over five cycles (32 bits of information are sent in each clock cycle). Cycle 1 : header[31:0] Cycle 2 : header[63:32] Cycle 3 : header[95:64] Cycle 4 : header[127:96] Cycle 5 : TLP prefix	EP/RP	slow_clk
pX_app_err_info_i[13:0]	Input	This error bus carries the following information: • [0]: Malformed TLP • [1]: Receiver overflow • [2]: Unexpected completion • [3]: Completer abort • [4]: Completion timeout • [5]: Unsupported request • [6]: Poisoned TLP received • [7]: AtomicOp egress blocked • [8]: Uncorrectable internal error • [9]: Correctable internal error • [10]: Advisory error	EP/RP	slow_clk

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
		• [11]: TLP prefix blocked • [12]: ACS violation • [13]: ECRC check failed		
x16/x8: pX_app_err_func_num_i[2:0] x4: NA	Input	This bus contains the function number for the function that asserts the error valid signal.	EP/RP	slow_clk
pX_app_err_ready_o	Output	When deasserted, this signal indicates the endpoint may be in the process of servicing another message, and unable to service this Master for back-to-back user input.	EP/RP	slow_clk

4.3.6. Completion Timeout Interface

The R-Tile IP for PCIe features a Completion timeout mechanism to keep track of Non-Posted requests sent by the user application and the corresponding Completions received. When the R-Tile IP detects a Completion timeout, it notifies the user application by asserting the `cpl_timeout_o` signal.

When a Completion timeout happens, the user application can use the Completion Timeout Interface (for each port) to get more detailed information about the event and update the AER capability registers if required. After the completion timeout FIFO becomes empty, the IP core deasserts the `cpl_timeout_o` signal.

Table 69. Completion Timeout Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
pX_cpl_timeout_o	Output	Indicates that the Completion TLP for a request has not been received within the expected time window. The outputs below are valid when cpl_timeout_o is asserted.	EP/RP/BP	slow_clk
pX_cpl_timeout_func_num_o[2:0]	Output	The function number of the timed-out completion.	EP/RP/BP	slow_clk
pX_cpl_timeout_vfunc_num_o[10:0]	Output	Indicates which virtual function (VF) had a completion timeout.	EP/RP/BP	slow_clk
pX_cpl_timeout_vfunc_active_o	Output	Indicates that a virtual function (VF) had a completion timeout.	EP/RP/BP	slow_clk
pX_cpl_timeout_cpl_tc_o[2:0]	Output	The Traffic Class of the timed-out completion.	EP/RP/BP	slow_clk
pX_cpl_timeout_cpl_attr_o[1:0]	Output	The Attributes field of the timed-out completion.	EP/RP/BP	slow_clk
pX_cpl_timeout_cpl_len_o[11:0]	Output	The length (in bytes) of the timed-out completion.	EP/RP/BP	slow_clk
pX_cpl_timeout_cpl_tag_o[9:0]	Output	The Tag field of the timed-out completion.	EP/RP/BP	slow_clk

4.3.7. Configuration Intercept Interface

The Configuration Intercept Interface (CII) allows the application logic to detect the occurrence of a Configuration (CFG) request on the link and to modify its behavior. To provide further flexibility, the R-Tile Intel FPGA IP for PCI Express allows you to define up to three different PCIe configuration space ranges to intercept using this interface (refer to CII Address Ranges in the IP Parameter Editor, where the allowed range is 0x - 0xFFF).

Note: If the VirtIO feature is enabled, the ranges 0x50-0x6E and 0xC0-0xF7 are reserved and only one range will be exposed.

The application logic should detect the CFG request at the rising edge of `cii_req_o`. Due to the latency of the EMIB, the `cii_req_o` can be deasserted many cycles after the deassertion of `cii_halt_i`.

The application logic can use the CII to:

- Delay the processing of a CFG request by the controller. This allows the application to perform any housekeeping task first.
- Overwrite the data payload of a CfgWr request. The application logic can also overwrite the data payload of a CfgRd Completion TLP.

Note: In the P-Tile Avalon Streaming Intel FPGA IP for PCIe, the Configuration Output Interface (tl_cfg) provided a subset of the information stored in the PCIe Configuration space. Starting with the R-Tile Avalon Streaming Intel FPGA IP for PCIe, the CII interface should be used as a replacement for similar functionality. To achieve this, the Application logic should intercept the Configuration TLPs of interest during the enumeration process. Additionally, the Hard IP Reconfiguration Interface (Hard IP Reconfiguration Interface on page 103) is available and can also be used to access the PCIe Configuration space information.

This interface also allows you to implement the Intel Vendor Specific Extended Capability (VSEC) registers. All configuration accesses targeting the Intel VSEC registers (addresses 0xD00 to 0xFFF) are automatically mapped to this interface and can be monitored via this interface.

If you are not using this interface, tie `cii_halt_p0/1` to logic 0.

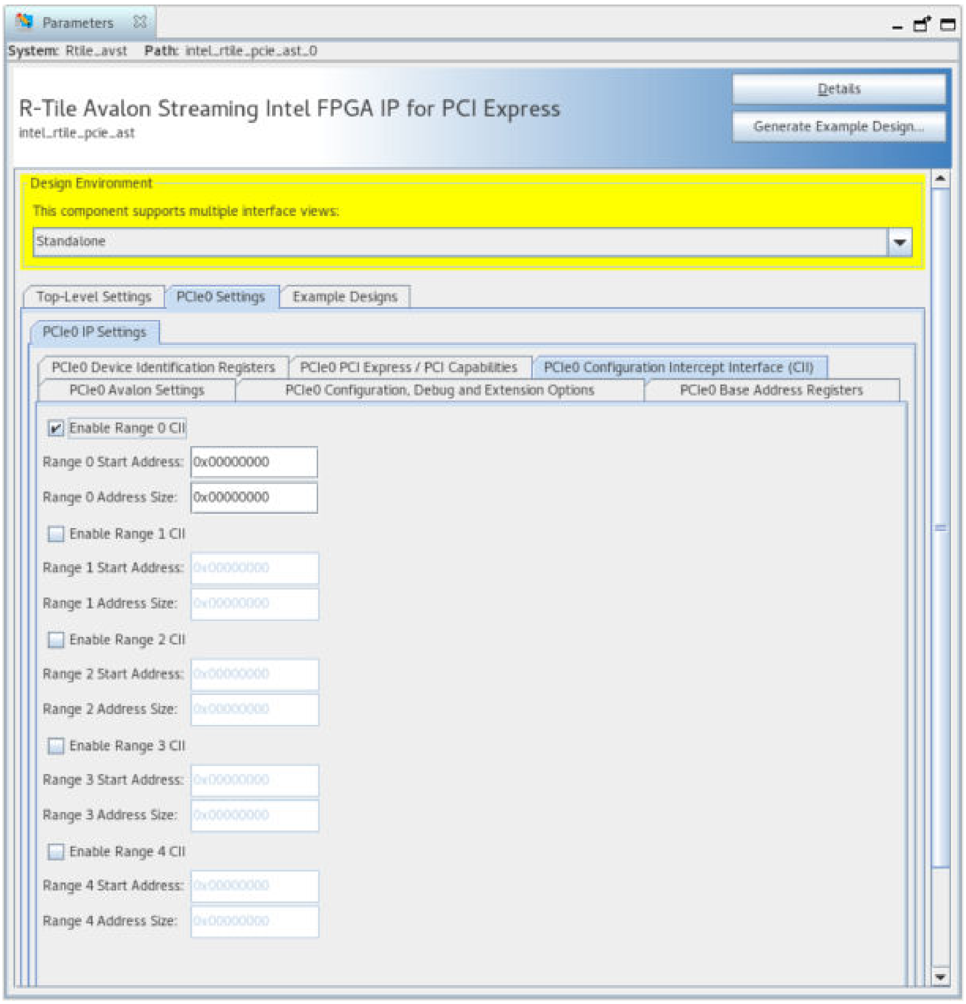
Table 70. Configuration Intercept Interface (CII) Signals

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
`pX_cii_req_o`	Output	Indicates the CFG request is intercepted and all the other CII signals are valid.	EP	`slow_clk`
`pX_cii_hdr_poisoned_o`	Output	The poisoned bit in the received TLP header on the CII.	EP	`slow_clk`
`pX_cii_hdr_first_be_o[3:0]`	Output	The first dword byte enable field in the received TLP header on the CII.	EP	`slow_clk`
`pX_cii_func_num_o[2:0]`	Output	The function number in the received TLP header on the CII.	EP	`slow_clk`

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
pX_cii_wr_o	Output	Indicates that cii_dout_p0/1 is valid. This signal is asserted only for a configuration write request.	EP	slow_clk
pX_cii_wr_vf_active_o	Output	Indicates that the received TLP targets a virtual function (VF) inside the controller, which is identified by cii_vf_num_o[10:0].	EP	slow_clk
pX_cii_vf_num_o[10:0]	Output	Identifies the VF inside the controller that was targeted.	EP	slow_clk
pX_cii_addr_o[9:0]	Output	The double-word register address in the received TLP header on the CII.	EP	slow_clk
pX_cii_dout_o[31:0]	Output	Received TLP payload data from the link partner to your application client. The data is in little endian format. The first received payload byte is in bits [7:0].	EP	slow_clk
pX_cii_override_en_i	Input	Override enable. When the application logic asserts this input, the PCIe Hard IP overrides the CfgWr payload or CfgRd completion using the data supplied by the application logic on cii_override_din_i[31:0].	EP	slow_clk
pX_cii_override_din_i[31:0]	Input	Override data. • CfgWr: this bus contains the data supplied by the application logic to override the write data to the PCIe Hard IP register. • CfgRd: this bus contains the data supplied by the application logic to override the data payload of the Completion TLP.	EP	slow_clk
pX_cii_halt_i	Input	Flow control input signal. When cii_halt_p0/1 is asserted, the PCIe Hard IP halts the processing of CFG requests for the PCIe configuration space registers.	EP	slow_clk

Figure 39. CII Address Ranges in the IP Parameter Editor


Parameters
System: Rtile_avst Path: intel_rtile_pcie_ast_0
R-Tile Avalon Streaming Intel FPGA IP for PCI Express
intel_rtile_pcie_ast
Details
Generate Example Design...
Design Environment
This component supports multiple interface views:
Top-Level Settings
PCIe0 Settings
Example Designs
PCIe0 IP Settings
PCIe0 Device Identification Registers
PCIe0 PCI Express / PCI Capabilities
PCIe0 Configuration Intercept Interface (CII)
PCIe0 Avalon Settings
PCIe0 Configuration, Debug and Extension Options
PCIe0 Base Address Registers

Standalone

[✓] Enable Range 0 CII
Range 0 Start Address: 0x00000000
Range 0 Address Size: 0x00000000
[ ] Enable Range 1 CII
Range 1 Start Address: 0x00000000
Range 1 Address Size: 0x00000000
[ ] Enable Range 2 CII
Range 2 Start Address: 0x00000000
Range 2 Address Size: 0x00000000
[ ] Enable Range 3 CII
Range 3 Start Address: 0x00000000
Range 3 Address Size: 0x00000000
[ ] Enable Range 4 CII
Range 4 Start Address: 0x00000000
Range 4 Address Size: 0x00000000

4.3.8. Power Management Interface

Software programs the device into a D-state by writing to the Power Management Control and Status register in the PCI Power Management Capability Structure. The power management output signals indicate the current power state. The IP core supports the two mandatory power states: D0 (full power) and D3Hot. It does not support the optional D1 and D2 low-power states.

The correspondence between the device power states (D states) and link power states (L states) is as follows:

Table 71. Relationship Between Device and Link Power States

Device Power State	Link Power State
D0	L0
D1 (not supported)	L1

continued...

Device Power State	Link Power State
D2 (not supported)	L1
D3Hot	L1, L2/L3 Ready
D3Cold	L2, L3

The following table shows the support for L2/L3 states in R-Tile.

Table 72. L2/L3 Support in R-Tile

	EP/BP UP	RP/BP DN
L2/L3 entry	Ok	Ok
L2 exit	Host to initiate or Cold Reset	Cold Reset
L3 exit	Cold Reset	Cold Reset

Table 73. Power Management Interface

Signal Name	Direction	Description	Clock Domain	EP/RP/BP
`pm_curnt_state_o[7:0]`	O	Indicates the current power state. • 8'b00000001 : L0 or IDLE • 8'b00000010 : L0s • 8'b00000100 : L1 • 8'b00001000 : L2 • 8'b00010000 : L3 • Other values are invalid.	Async	EP/RP/BP
x16/x8: `pm_dstate_o[31:0]` x4: `pm_dstate_o[3:0]`	O	Power management D-state for each function. • 4'b0001 : D0 • 4'b1000 : D3 Hot • Other values are invalid.	Async	EP/RP/BP
x16/x8: `apps_pm_xmt_pme_i[7:0]` x4: NA	I	The application logic asserts this signal for one cycle to wake up the Power Management Capability (PMC) state machine from a D1, D2, or D3 Hot power state. Upon wake-up, the IP core sends a PM_PME message. This signal needs to be asserted for one clock cycle.	`slow_clk`	EP/BP
x16/x8: `apps_ready_entr_l23_i` x4: NA	I	The application logic asserts this signal to indicate that it is ready to enter the L2/L3 Ready state. The `app_ready_entr_l23_i` signal is provided for applications that must control the L2/L3 Ready entry (in case certain tasks must be performed before going into L2/L3 Ready). The core delays sending PM_Enter_L23 (in response to PM_Turn_Off) until this signal becomes active. This is a level-sensitive signal.	`slow_clk`	EP/BP

continued...

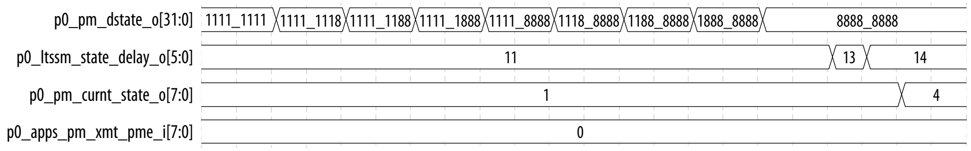
Signal Name	Direction	Description	Clock Domain	EP/RP/BP
apps_pm_xmt_turnoff_i	I	This signal is a pulse input. It is a request from the Application Layer to generate a PM_Turn_Off message. The Application Layer must assert this signal for one clock cycle. The IP core does not return an acknowledgement or grant signal. The Application Layer must not pulse the same signal again until the previous message has been transmitted.	slow_clk	RP
app_init_rst_i	I	The Application Layer uses this signal to request a hot reset to downstream devices. The hot reset request will be sent when a single-cycle pulse (~20ns) is applied to this pin.	Asynchronous	RP
app_req_retry_en_i[7:0]	I	When asserted, the PCIe Hard IP responds to Configuration TLPs with a CRS (Configuration Retry Status) if it has not already responded to a Configuration TLP with non-CRS status since the last reset. The user application can use this signal to hold off on enumeration. This input is not used for Root Ports. This bus applies to both Endpoints when the Hard IP is configured as 2x8. The x4 cores (Ports 2 and 3) also have these pins but they are not used and need to be driven to zero.	Asynchronous	EP
app_xfer_pending_i	I	This signal prevents the entry to L1 or initiates the exit from L1.	Asynchronous	EP/RP/BP

4.3.8.1. D3Hot Entry

The following sequence describes the D3Hot entry procedure. All transmissions on the Avalon Streaming TX and RX interfaces must have completed before the R-Tile PCIe IP core can begin the L1 request (Enter_L1 DLLP). In addition, the RX buffer must be empty and the `app_xfer_pending_i` signal must be deasserted.

1. Software on the host side writes the Power Management Control register to request the entry to D3Hot state.
2. The endpoint stops transmitting requests when it has been taken out of D0. Application logic can use the `pm_dstate_o` signal to monitor the current D state.
3. The link transitions to L1. Application logic can use the `pm_curnt_state_o` signal to monitor the current L state.

Figure 40. Timing Diagram for D3Hot Entry



4.3.8.2. D3Hot Exit

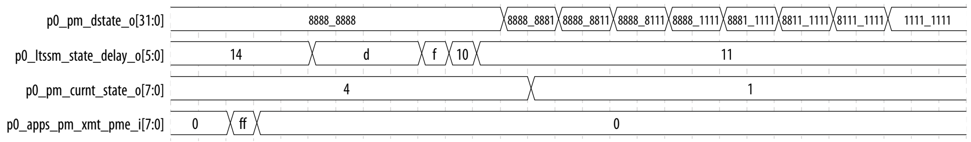
4.3.8.2.1. D3Hot Exit Initiated by Host

The system host will attempt to send a CfgWr to the Power Management Control register to go to D0. This will automatically first cause a transition from L1 to L0 started by the host. Then, the CfgWr will be sent to the Power Management Control register in the R-Tile Avalon Streaming Intel FPGA IP for PCIe to change the power state and pme enable. Alternatively, the host can initiate a link retrain, link disable or hot reset for an L1 exit.

4.3.8.2.2. D3Hot Exit Initiated by EP

For the Endpoint to exit the D3hot state, the PME_en bit in the Power Management Control and Status register needs to be set first by the Host (prior to the D3hot entry). The Application Layer can then request a wake-up event by asserting `apps_pm_xmt_pme_i`, which causes the IP core to transmit a PM_PME message. In addition, the IP core sets the PME_status bit in the Power Management Control and Status register to notify software that it has requested the D3hot exit transition.

Figure 41. Timing Diagram for D3Hot Exit Initiated by EP



4.3.8.3. D3Cold Entry

The following sequence describes the D3Cold entry procedure. All transmissions on the Avalon Streaming TX and RX interfaces must have completed before the R-Tile PCIe IP core can begin the L1 request (Enter_L1 DLLP). In addition, the RX buffer must be empty and the `app_xfer_pending_i` signal must be deasserted.

1. Software on the host side writes the Power Management Control register to request the entry to D3Hot state.
2. The endpoint stops transmitting requests when it has been taken out of D0. Application logic can use the `pm_dstate_o` signal to monitor the current D state.
3. The link transitions to L1. Application logic can use the `pm_curnt_state_o` signal to monitor the current L state.
4. Software on the host side sends the PME_Turn_Off Message to the endpoint to initiate a power down. The delivery of the message TLP causes the link to transition to L0 and the message will also be passed on to the Avalon Streaming RX interface.

5. The R-Tile IP core automatically transmits a PME_TO_Ack Message to acknowledge the Turn Off request.
6. When ready for the power removal D3Cold state, the application logic in the Endpoint asserts `p#_app_ready_entr_l23_i`. The R-Tile IP core will then send the PM_Enter_L23 DLLP and initiate the Link transition to L2/L3 Ready.
7. In this state, power can be removed to transition the link to L3. Alternatively, if supported by the Host system, the link can be transitioned to L2 while maintaining `refclk` and Vaux.

4.3.8.4. D3Cold Exit

When configured as an endpoint, the R-Tile Avalon Streaming IP cannot initiate a transition from D3Cold to D0. Depending on the Link state, the exit transition must be as follows:

From L2:

The system host will attempt to send a CfgWr to the Power Management Control register to go to D0. This will automatically first cause a transition from L2 to L0 started by the host. Then, the CfgWr will be sent to the Power Management Control register in the R-Tile IP to change the power state and pme enable.

From L3:

A cold reset is required to perform a complete FPGA configuration and transition the link to L0.

When the IP is configured as a Root Port, a cold reset is required to perform a complete FPGA configuration and transition the link from L2 or L3 back to L0.

4.3.9. Hard IP Status Interface

This interface includes the signals that are useful for debugging, such as the link status signal, LTSSM state outputs, etc.

Table 74. Hard IP Status Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
`pX_link_up_o`	Output	When asserted, this signal indicates the link is up.	EP/RP/BP	`coreclkout_hip`
`pX_dl_up_o`	Output	When asserted, this signal indicates the Data Link (DL) Layer is active.	EP/RP/BP	`coreclkout_hip`
`pX_ltssm_state_delay_o[5:0]`	Output	Delayed version of the live LTSSM state of the PCIe Hard IP. • 6'h00: S_DETECT_QUIET • 6'h01: S_DETECT_ACT • 6'h02: S_POLL_ACTIVE • 6'h03: S_POLL_COMPLIANCE • 6'h04: S_POLL_CONFIG • 6'h05: S_PRE_DETECT_QUIET • 6'h06: S_DETECT_WAIT	EP/RP/BP	`slow_clk`

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
		• 6'h07: S_CFG_LINKWD_START • 6'h08: S_CFG_LINKWD_ACCEPT • 6'h09: S_CFG_LANENUM_WAIT • 6'h0A: S_CFG_LANENUM_ACCEPT • 6'h0B: S_CFG_COMPLETE • 6'h0C: S_CFG_IDLE • 6'h0D: S_RCVRY_LOCK • 6'h0E: S_RCVRY_SPEED • 6'h0F: S_RCVRY_RCVRCFG • 6'h10: S_RCVRY_IDLE • 6'h11: S_L0 • 6'h12: S_L0S • 6'h13: S_L123_SEND_EIDLE • 6'h14: S_L1_IDLE • 6'h15: S_L2_IDLE • 6'h16: S_L2_WAKE • 6'h17: S_DISABLED_ENTRY • 6'h18: S_DISABLED_IDLE • 6'h19: S_DISABLED • 6'h1A: S_LPBK_ENTRY • 6'h1B: S_LPBK_ACTIVE • 6'h1C: S_LPBK_EXIT • 6'h1D: S_LPBK_EXIT_TIMEOUT • 6'h1E: S_HOT_RESET_ENTRY • 6'h1F: S_HOT_RESET • 6'h20: S_RCVRY_EQ0 • 6'h21: S_RCVRY_EQ1 • 6'h22: S_RCVRY_EQ2 • 6'h23: S_RCVRY_EQ3		
pX_ltssm_st_hipfifo_ovrflw_o	Output	PCIe Hard IP FIFO storing ltssm_state changes is full. State changes may have been dropped prior to the current ltssm_state value change.	EP/RP/BP	slow_clk
pX_surprise_down_err_o	Output	Surprise Down Error indicator.	EP/RP/BP	coreclkout_hip
pX_dl_timer_update_o	Output	This signal asserts when DL Ack/Replay Timers need to be updated due to a change in the Maximum Payload Size, Link Width, or Link Speed.	EP/RP/BP	coreclkout_hip
pX_tx_ehp_deallocate_empty_o	Output	This signal indicates when the PCIe Hard IP Tx FIFO is empty.	EP/RP/BP	coreclkout_hip

4.3.10. Page Request Services (PRS) Interface (Endpoint Only)

When an Endpoint determines that it requires access to a page for which the ATS translation is not available, it sends a Page Request message to request that the page be mapped into system memory.

The PRS interface allows the monitoring of when PRS events happen, what functions these PRS events belong to, and what types of events they are.

The PRS interface is only available in EP mode, and with TLP Bypass disabled.

Note: The R-Tile Avalon Streaming Intel FPGA IP for PCIe only provides the PRS capability. To take advantage of this feature, you need to implement the necessary logic in your application.

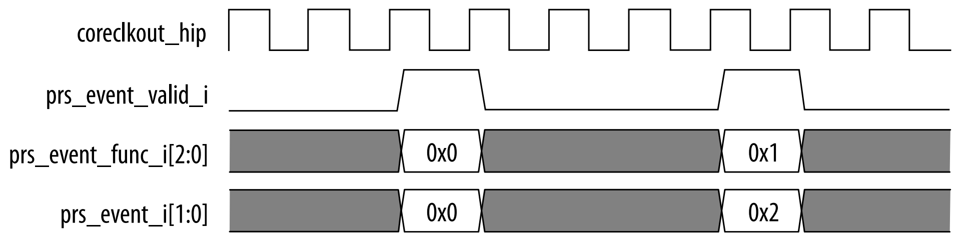
Note: Only Ports 0 and 1 support PRS.

Table 75. PRS Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock
`pX_prs_event_valid_i` where X = 0, 1, 2, 3 (core number)	Input	This signal qualifies `pX_prs_event_func_i` and `pX_prs_event_i`. There is a single-cycle pulse for each PRS event.	EP	`slow_clk`
`pX_prs_event_func_i[2:0]` where X = 0, 1, 2, 3 (core number)	Input	The function number for the PRS event.	EP	`slow_clk`
`pX_prs_event_i[1:0]` where X = 0, 1, 2, 3 (core number)	Input	00 : Indicate that the function has received a PRG response failure. 01: Indicate that the function has received a response with Unexpected Page Request Group Index. 10: Indicate that the function has completed all previously issued page requests and that it has stopped requests for additional pages. Only valid when the PRS enable bit is clear. 11: reserved.	EP	`slow_clk`

Example Timing Diagram for the PRS Event Interface below shows the timing diagram for the PRS event interface when the application layer of function 0 sends an event of PRG response reception, and the application layer of function 1 sends an event stopping requests for additional pages.

Figure 42. Example Timing Diagram for the PRS Event Interface



4.3.11. Function-Level Reset (FLR) Interface (Endpoint Only)

FLR allows specific physical/virtual functions to be reset without affecting other physical/virtual functions or the link they share. This interface is only present in EP mode (for x16/x8 configurations).

Note: Only Ports 0 and 1 support FLR.

Table 76. Function-Level Reset (FLR) Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
pX_flr_rcvd_pf_o[7:0] where X = 0, 1, 2, 3 (IP core number)	Output	Pulse-based signals. Once asserted, the signals remain high until the Application Layer sets the pX_flr_completed_pf_num_i[2:0] high for the associated function. The Application Layer must perform actions necessary to clear any pending transactions associated with the function being reset. The Application Layer must also assert pX_flr_completed_pf_num_i[2:0] to indicate it has completed the FLR actions and is ready to reenable the PF.	EP	slow_clk
pX_flr_rcvd_vf_o where X = 0, 1, 2, 3 (IP core number)	Output	A one-cycle pulse on this signal indicates that an FLR was received from the host targeting a VF.	EP	slow_clk
pX_flr_rcvd_pf_num_o[2:0] where X = 0, 1, 2, 3 (IP core number)	Output	Parent PF number of the VF undergoing FLR.	EP	slow_clk
pX_flr_rcvd_vf_num_o[10:0] where X = 0, 1, 2, 3 (IP core number)	Output	VF number offset of the VF undergoing FLR.	EP	slow_clk
pX_flr_completed_pf_i[7:0] where X = 0, 1, 2, 3 (IP core number)	Input	One bit per PF. A one-cycle pulse on any bit indicates that the application has completed the FLR sequence for the corresponding PF and is ready to be enabled.	EP	slow_clk
pX_flr_completed_vf_i where X = 0, 1, 2, 3 (IP core number)	Input	A one-cycle pulse from the application reenables a VF.	EP	slow_clk
pX_flr_completed_pf_num_i[2:0] where X = 0, 1, 2, 3 (IP core number)	Input	Parent PF number of the VF to re-enable.	EP	slow_clk
pX_flr_completed_vf_num_i[10:0] where X = 0, 1, 2, 3 (IP core number)	Input	VF number offset of the VF to re-enable.	EP	slow_clk
pX_flr_completed_ready_o where X = 0, 1, 2, 3 (IP core number)	Output	Value 0 indicates the previous message is pending. The new FLR completed should be held if pX_flr_completed_ready_o = 0.	EP	slow_clk

4.3.12. SR-IOV VF Error Flag Interface (Endpoint Only)

The VFs, with no AER support, are required to generate Non-Fatal error messages. The IP does not generate any error message. It is up to the user application logic to generate appropriate messages when specific error conditions occur.

The R-Tile IP for PCIe makes necessary signals available to the user application logic to generate these messages. The Completion Timeout Interface and the signals listed in the table below provide the necessary information to generate Non-Fatal error messages.

This interface applies to EP only.

Note: Ports 2 and 3 do not support SR-IOV.

Table 77. RX VF Error Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
pX_vf_err_poisonedwrreq_s0_o pX_vf_err_poisonedwrreq_s1_o where X = 0, 1 (IP core number)	Output	Indicates a Poisoned Write Request is received.	EP	slow_clk
pX_vf_err_poisonedcompl_s0_o pX_vf_err_poisonedcompl_s1_o where X = 0, 1 (IP core number)	Output	Indicates a Poisoned Completion is received.	EP	slow_clk
pX_vf_err_ur_posted_s0_o pX_vf_err_ur_posted_s1_o where X = 0, 1 (IP core number)	Output	Indicates a Posted UR request is received.	EP	slow_clk
pX_vf_err_ca_postedreq_s0_o pX_vf_err_ca_postedreq_s1_o where X = 0, 1 (IP core number)	Output	Indicates a Posted CA request is received.	EP	slow_clk
pX_vf_err_vf_num_s0_o[10:0] pX_vf_err_vf_num_s1_o[10:0] where X = 0, 1 (IP core number)	Output	Indicates the VF number for which the error is detected.	EP	slow_clk
pX_vf_err_func_num_s0_o[2:0] pX_vf_err_func_num_s1_o[2:0] where	Output	Indicates the physical function number associated with the VF that has the error.	EP	slow_clk

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
X = 0, 1 (IP core number)				
pX_vf_err_overflow_o	Output	Indicates a VF error FIFO overflow and a loss of an error report. The overflow can happen when coreclkout_hip is slower than the default value. If coreclkout_hip is running at the default frequency, the overflow will not happen.	EP	slow_clk

Table 78. TX VF Error Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
pX_user_sent_vfnonfatalmsg_i where X = 0, 1 (IP core number)	Input	Indicates the user application sent a non-fatal error message in response to an error detected.	EP	slow_clk
pX_user_vfnonfatalmsg_vfnum_i[10:0] where X = 0, 1 (IP core number)	Input	Indicates the VF number for which the error message was generated. This bus is valid when user_sent_vfnonfatalmsg_s0_i is high.	EP	slow_clk
pX_user_vfnonfatalmsg_func_num_i[2:0] where X = 0, 1 (IP core number)	Input	Indicates the PF number associated with the VF with the error. This bus is valid when user_sent_vfnonfatalmsg_s0_i is high.	EP	slow_clk
pX_user_vfnonfatalmsg_ready_o where X = 0, 1 (IP core number)	Output	Value 0 indicates an input change is pending. The new value should be held. pX_user_vfnonfatalmsg_ready_o = 1 when the interface is ready to accept the new value.	EP	slow_clk

4.3.13. General Purpose VSEC Interface

Table 79. General Purpose VSEC Interface Signals

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
pX_pld_gp_ctrl_o[7:0] where X = 0, 1, 2, 3 (IP core number)	Output	General purpose VSEC control register value.	EP/RP/BP	slow_clk
pX_pld_gp_status_i[7:0] where X = 0, 1, 2, 3 (IP core number)	Input	General purpose VSEC status register value.	EP/RP/BP	slow_clk
pX_pld_gp_status_ready_o where	Output	Value 0 indicates an input change is pending. The new value should be held if pX_pld_gp_status_ready_o = 0.	EP/RP/BP	slow_clk

continued...

Signal Name	Direction	Description	EP/RP/BP	Clock Domain
X = 0, 1, 2, 3 (IP core number)		`pX_pld_gp_status_ready_o` = 1 when the interface is ready to accept the new value.		

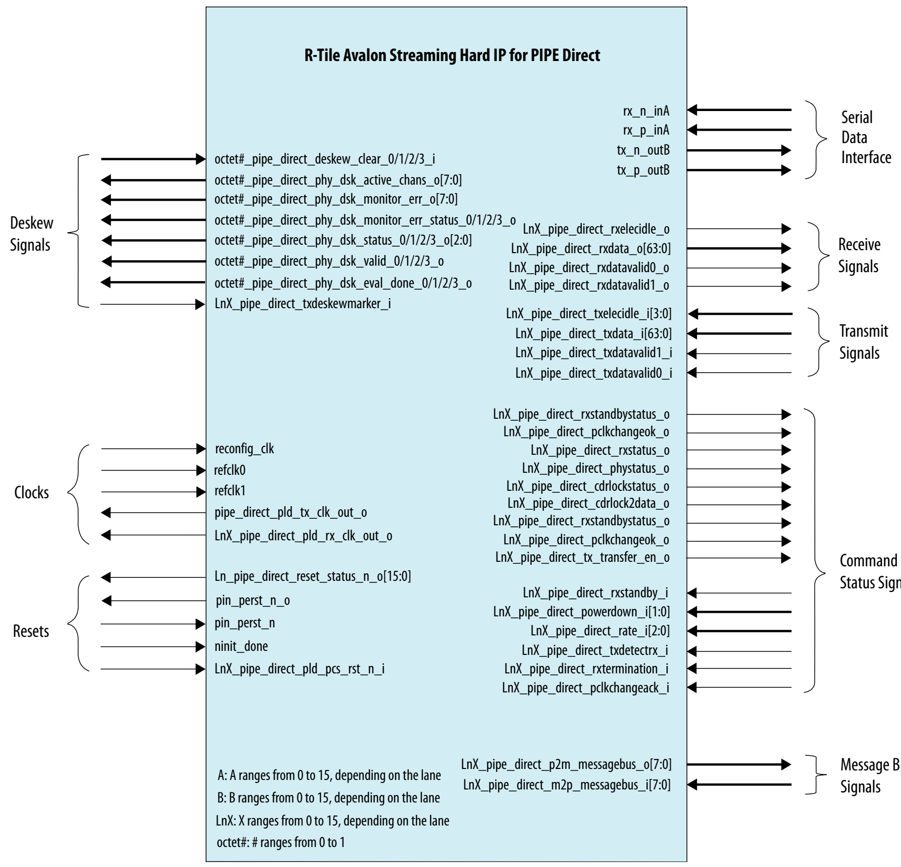
4.4. PIPE Direct Mode

In the PIPE Direct mode, either one or both of the PCIe and CXL controller stacks are entirely bypassed, and the PIPE SerDes mode interface is exported across the Embedded Multi-die Interconnect Bridge (EMIB) to the FPGA fabric. This mode allows you to implement your own custom controllers in soft IP.

In PIPE Direct mode, each entire octet must be configured to the same base mode setting (Gen5 capable). All per-lane variations are configured through the PIPE `Rate[2:0]` and `Powerdown[1:0]` control settings, which are independent per-lane.

Note: In the 22.4 release of Intel Quartus Prime, the PIPE Direct mode does not support design example or testbench generation.

Figure 43. R-Tile Top-Level Block Diagram in PIPE Direct Mode


R-Tile Avalon Streaming Hard IP for PIPE Direct
Deskew Signals
octet#_pipe_direct_deskew_clear_0/1/2/3_i
octet#_pipe_direct_phy_dsk_active_chans_o[7:0]
octet#_pipe_direct_phy_dsk_monitor_err_o[7:0]
octet#_pipe_direct_phy_dsk_monitor_err_status_0/1/2/3_o
octet#_pipe_direct_phy_dsk_status_0/1/2/3_o[2:0]
octet#_pipe_direct_phy_dsk_valid_0/1/2/3_o
octet#_pipe_direct_phy_dsk_eval_done_0/1/2/3_o
LnX_pipe_direct_txdeskewmarker_i
Clocks
reconfig_clk
refclk0
refclk1
pipe_direct_pld_tx_clk_out_o
LnX_pipe_direct_pld_rx_clk_out_o
Resets
Ln_pipe_direct_reset_status_n_o[15:0]
pin_perst_n_o
pin_perst_n
ninit_done
LnX_pipe_direct_pld_pcs_rst_n_i
rx_n_inA
rx_p_inA
tx_n_outB
tx_p_outB
Serial Data Interface
LnX_pipe_direct_rxelecidle_o
LnX_pipe_direct_rxdata_o[63:0]
LnX_pipe_direct_rxdatavalid0_o
LnX_pipe_direct_rxdatavalid1_o
Receive Signals
LnX_pipe_direct_txelecidle_i[3:0]
LnX_pipe_direct_txdata_i[63:0]
LnX_pipe_direct_txdatavalid1_i
LnX_pipe_direct_txdatavalid0_i
Transmit Signals
LnX_pipe_direct_rxstandbystatus_o
LnX_pipe_direct_pclkchangeok_o
LnX_pipe_direct_rxstatus_o
LnX_pipe_direct_phystatus_o
LnX_pipe_direct_cdrlockstatus_o
LnX_pipe_direct_cdrlock2data_o
LnX_pipe_direct_rxstandbystatus_o
LnX_pipe_direct_pclkchangeok_o
LnX_pipe_direct_tx_transfer_en_o
LnX_pipe_direct_rxstandby_i
LnX_pipe_direct_powerdown_i[1:0]
LnX_pipe_direct_rate_i[2:0]
LnX_pipe_direct_txdetectrx_i
LnX_pipe_direct_rxtermination_i
LnX_pipe_direct_pclkchangeack_i
Command and Status Signals
LnX_pipe_direct_p2m_messagebus_o[7:0]
LnX_pipe_direct_m2p_messagebus_i[7:0]
Message Bus Signals
A: A ranges from 0 to 15, depending on the lane
B: B ranges from 0 to 15, depending on the lane
LnX: X ranges from 0 to 15, depending on the lane
octet#: # ranges from 0 to 1

4.4.1. Data Signals

The R-Tile Avalon Streaming Intel FPGA IP for PCI Express in PIPE direct mode is implemented with a SerDes architecture. The IP has a built-in word serializer for the `LnX_pipe_direct_txdata_i[63:0]` signals from the user interface and a word deserializer for the `LnX_pipe_direct_rxdata_o[63:0]` signals to the user interface. For a data rate of 8 GT/s or higher, the IP only exposes 8 bits out of each 10-bit slice for block-encoded data, e.g. bits [7:0] represent byte0, [8:15] represent byte1, [16:31] represent byte 2 etc. Refer to PIPE Direct TX Datapath and PIPE Direct RX Datapath for the `TxData/RxData` signal connection guidance.

4.4.1.1. Transmit Signals

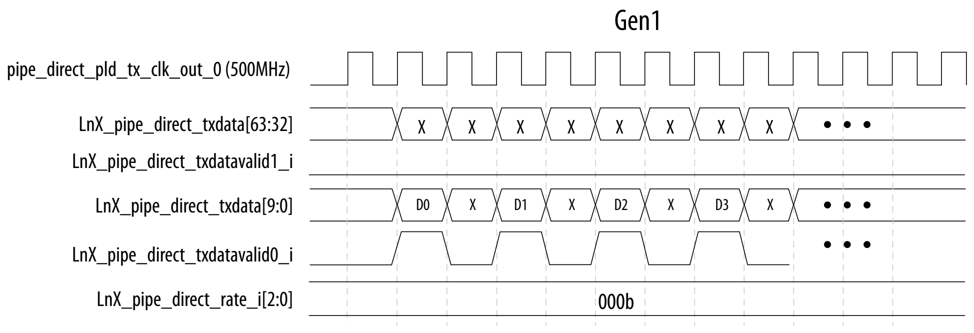
Table 80. PIPE Direct EMIB Data Channel Transmit Signals

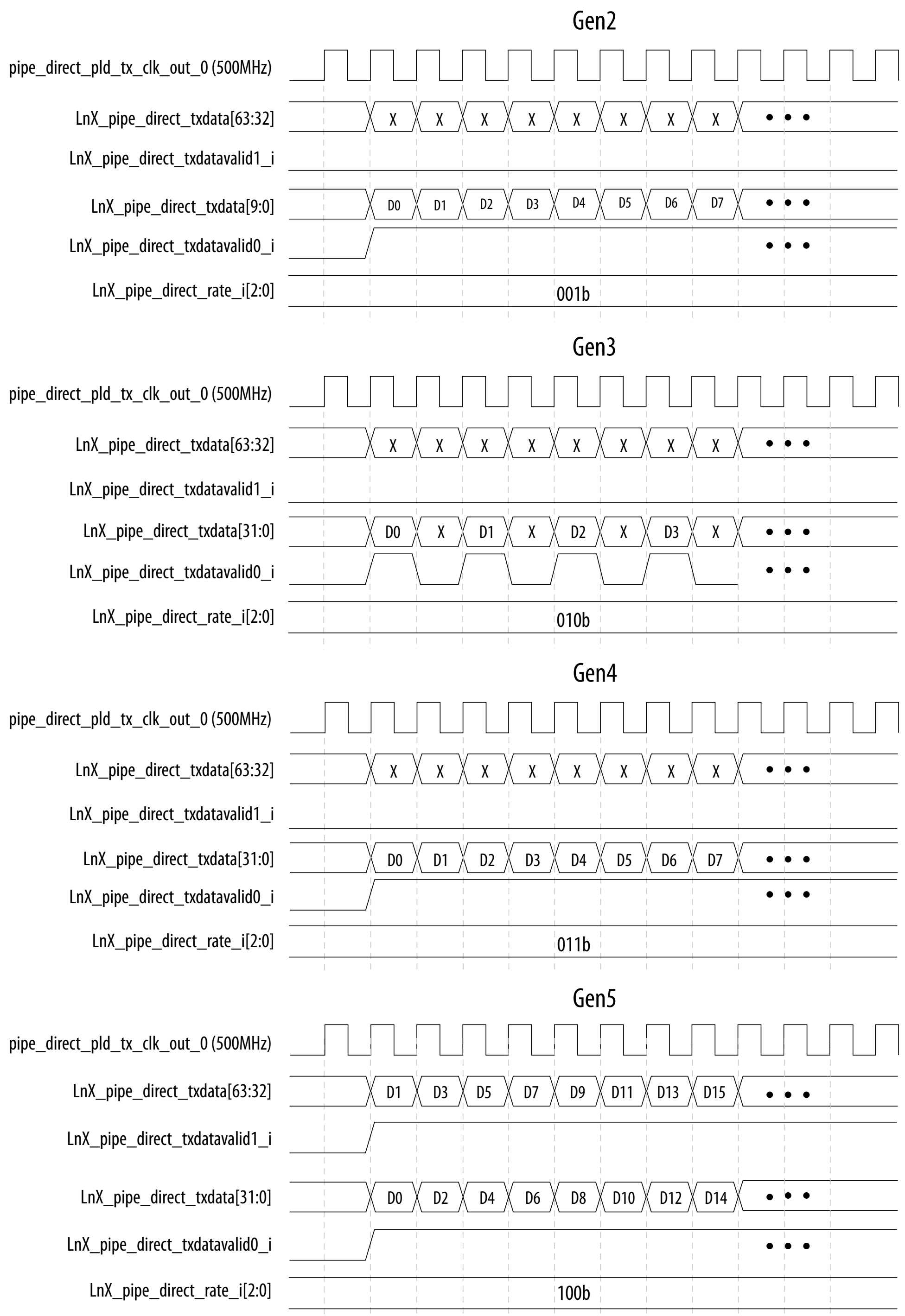
In the signal names, X is the lane number and ranges from 0 to 15.

Signal Name	Direction	Descriptions/Notes	Clock Domain
`lnX_pipe_direct_txelecidle_i[3:0]`	Input	One bit per two Symbols, for a maximum of 8 symbols.	`pipe_direct_pld_tx_clk_out_o`
`lnX_pipe_direct_txdatavalid1_i`	Input	This signal qualifies `txdata[63:32]`.	`pipe_direct_pld_tx_clk_out_o`
`lnX_pipe_direct_txdatavalid0_i`	Input	This signal qualifies `txdata[31:0]`.	`pipe_direct_pld_tx_clk_out_o`
`lnX_pipe_direct_txdata_i[63:0]`	Input	Transmit data bus	`pipe_direct_pld_tx_clk_out_o`

The following timing diagrams provide an illustration of the behaviors of the PIPE Direct TX Datapath signals:

Figure 44. PIPE Direct TX Datapath


Gen1
pipe_direct_pld_tx_clk_out_0 (500MHz)
LnX_pipe_direct_txdata[63:32]
LnX_pipe_direct_txdatavalid1_i
LnX_pipe_direct_txdata[9:0]
LnX_pipe_direct_txdatavalid0_i
LnX_pipe_direct_rate_i[2:0]
X
D0
D1
D2
D3
000b


Gen2
pipe_direct_pld_tx_clk_out_0 (500MHz)
LnX_pipe_direct_txdata[63:32]
X X X X X X X X
LnX_pipe_direct_txdatavalid1_i
LnX_pipe_direct_txdata[9:0]
D0 D1 D2 D3 D4 D5 D6 D7
LnX_pipe_direct_txdatavalid0_i
LnX_pipe_direct_rate_i[2:0]
001b
Gen3
pipe_direct_pld_tx_clk_out_0 (500MHz)
LnX_pipe_direct_txdata[63:32]
X X X X X X X X
LnX_pipe_direct_txdatavalid1_i
LnX_pipe_direct_txdata[31:0]
D0 X D1 X D2 X D3 X
LnX_pipe_direct_txdatavalid0_i
LnX_pipe_direct_rate_i[2:0]
010b
Gen4
pipe_direct_pld_tx_clk_out_0 (500MHz)
LnX_pipe_direct_txdata[63:32]
X X X X X X X X
LnX_pipe_direct_txdatavalid1_i
LnX_pipe_direct_txdata[31:0]
D0 D1 D2 D3 D4 D5 D6 D7
LnX_pipe_direct_txdatavalid0_i
LnX_pipe_direct_rate_i[2:0]
011b
Gen5
pipe_direct_pld_tx_clk_out_0 (500MHz)
LnX_pipe_direct_txdata[63:32]
D1 D3 D5 D7 D9 D11 D13 D15
LnX_pipe_direct_txdatavalid1_i
LnX_pipe_direct_txdata[31:0]
D0 D2 D4 D6 D8 D10 D12 D14
LnX_pipe_direct_txdatavalid0_i
LnX_pipe_direct_rate_i[2:0]
100b

Note: At Gen1 and Gen2 speeds, only the 10 LSB bits from the lower segment of LnX_pipe_direct_txdata bus contain valid data. Bits [63:10] are don't-cares.

4.4.1.2. Receive Signals

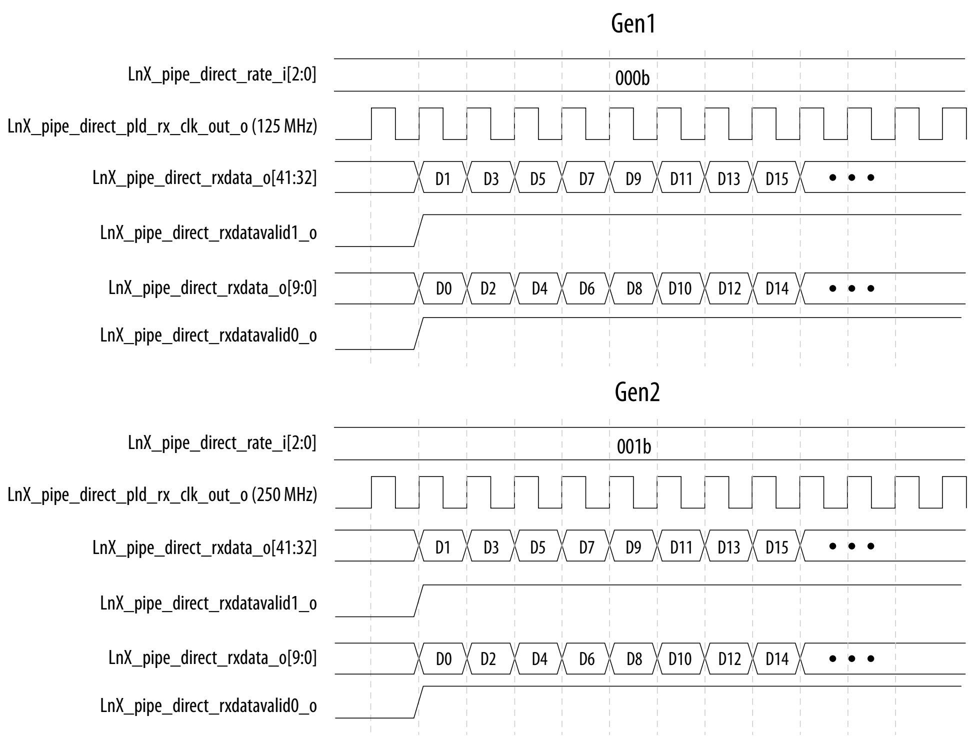
Table 81. PIPE Direct EMIB Data Channel Receive Signals

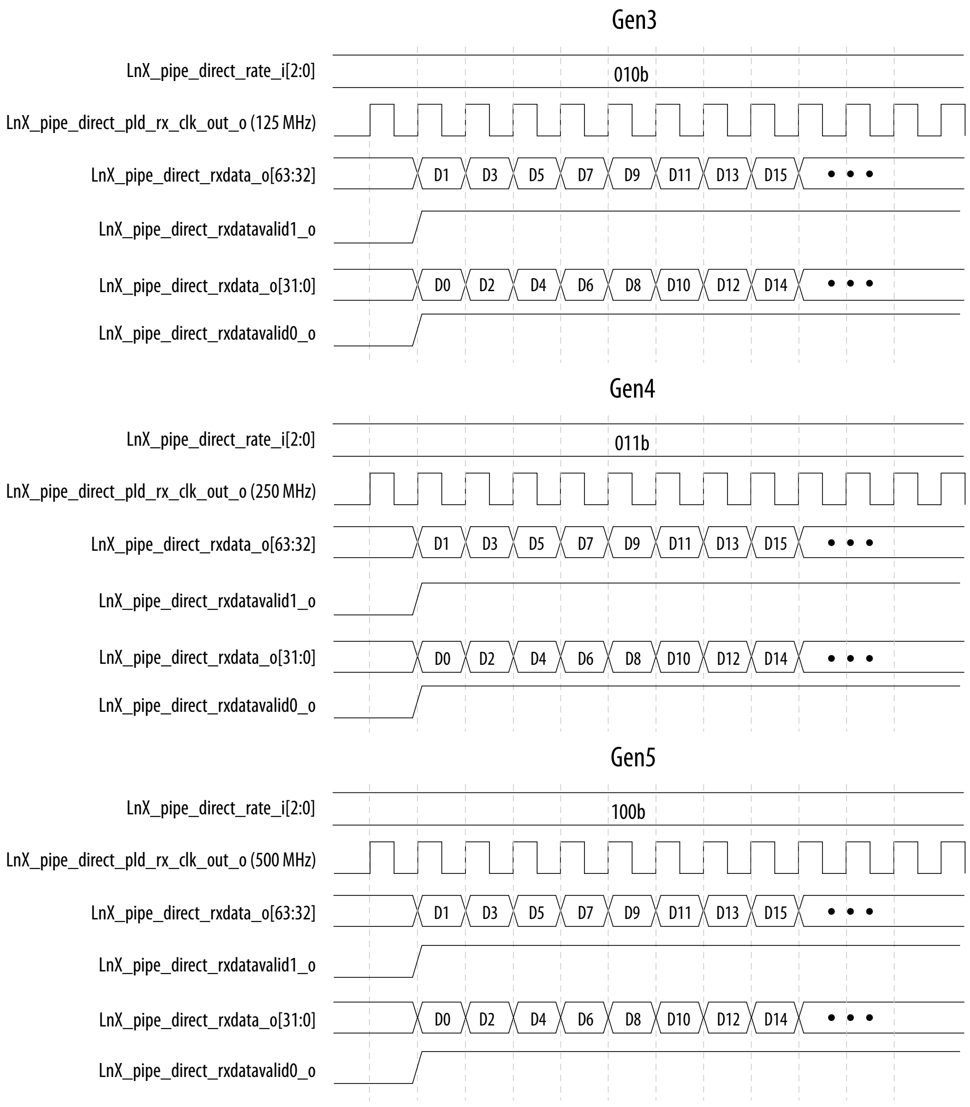
In the signal names, X is the lane number and ranges from 0 to 15.

Signal Names	Direction	Descriptions/Notes	Clock Domain
`lnX_pipe_direct_rxdatavalid1_o`	Output	This signal qualifies `rxdata[63:32]`.	`lnX_pipe_direct_pld_rx_clk_out_o`
`lnX_pipe_direct_rxdatavalid0_o`	Output	This signal qualifies `rxdata[31:0]`.	`lnX_pipe_direct_pld_rx_clk_out_o`
`lnX_pipe_direct_rxdata_o[63:0]`	Output	Receive data bus	`lnX_pipe_direct_pld_rx_clk_out_o`
`lnX_pipe_direct_rxelecIdle_o`	Output	This signal indicates the receiver detection of an Electrical Idle. It is an asynchronous signal. *Note:* This signal may toggle during continuous traffic. Per the PIPE Spec 5.1.1 section 9.4, the Soft IP controller must not rely on this signal for Electrical Idle detection when operating at gen2 or higher speeds. This toggling may not be observed in simulation and is a known limitation of the R-Tile simulation model.	Async

The following timing diagrams provide an illustration of the behaviors of the PIPE Direct RX Datapath signals:

Figure 45. PIPE Direct RX Datapath


Gen1
LnX_pipe_direct_rate_i[2:0]
000b
LnX_pipe_direct_pld_rx_clk_out_o (125 MHz)
LnX_pipe_direct_rxdata_o[41:32]
D1 D3 D5 D7 D9 D11 D13 D15 • • •
LnX_pipe_direct_rxdatavalid1_o
LnX_pipe_direct_rxdata_o[9:0]
D0 D2 D4 D6 D8 D10 D12 D14 • • •
LnX_pipe_direct_rxdatavalid0_o
Gen2
LnX_pipe_direct_rate_i[2:0]
001b
LnX_pipe_direct_pld_rx_clk_out_o (250 MHz)
LnX_pipe_direct_rxdata_o[41:32]
D1 D3 D5 D7 D9 D11 D13 D15 • • •
LnX_pipe_direct_rxdatavalid1_o
LnX_pipe_direct_rxdata_o[9:0]
D0 D2 D4 D6 D8 D10 D12 D14 • • •
LnX_pipe_direct_rxdatavalid0_o


Gen3
LnX_pipe_direct_rate_i[2:0]
010b
LnX_pipe_direct_pld_rx_clk_out_o (125 MHz)
LnX_pipe_direct_rxdata_o[63:32]
D1 D3 D5 D7 D9 D11 D13 D15 • • •
LnX_pipe_direct_rxdatavalid1_o
LnX_pipe_direct_rxdata_o[31:0]
D0 D2 D4 D6 D8 D10 D12 D14 • • •
LnX_pipe_direct_rxdatavalid0_o
Gen4
LnX_pipe_direct_rate_i[2:0]
011b
LnX_pipe_direct_pld_rx_clk_out_o (250 MHz)
LnX_pipe_direct_rxdata_o[63:32]
D1 D3 D5 D7 D9 D11 D13 D15 • • •
LnX_pipe_direct_rxdatavalid1_o
LnX_pipe_direct_rxdata_o[31:0]
D0 D2 D4 D6 D8 D10 D12 D14 • • •
LnX_pipe_direct_rxdatavalid0_o
Gen5
LnX_pipe_direct_rate_i[2:0]
100b
LnX_pipe_direct_pld_rx_clk_out_o (500 MHz)
LnX_pipe_direct_rxdata_o[63:32]
D1 D3 D5 D7 D9 D11 D13 D15 • • •
LnX_pipe_direct_rxdatavalid1_o
LnX_pipe_direct_rxdata_o[31:0]
D0 D2 D4 D6 D8 D10 D12 D14 • • •
LnX_pipe_direct_rxdatavalid0_o

Note: At Gen1 and Gen2 speeds, only the 10 LSB bits in the upper and lower segments of the LnX_pipe_direct_rxdata_o bus contain valid data. Bits [31:10] and [63:42] are don't-cares.

4.4.2. Command and Status Signals

Table 82. PIPE Direct EMIB Data Channel Command and Status Signals

In the signal names, X is the lane number and ranges from 0 to 15.

Signal Name	Direction	Descriptions/Notes	Clock Domain
`lnX_pipe_direct_rxstandby_i`	Input	Synchronous `rxstandby` signal	`pipe_direct_pld_tx_clk_out_o`
`lnX_pipe_direct_powerdown_i[1:0]`	Input	PHY power state control signals	`pipe_direct_pld_tx_clk_out_o`

continued...

Signal Name	Direction	Descriptions/Notes	Clock Domain
lnX_pipe_direct_rate_i[2:0]	Input	Gen1-5 rate change control signals: 000: Gen1 001: Gen2 010: Gen3 011: Gen4 100: Gen5	pipe_direct_pld_tx_clk_out_o
lnX_pipe_direct_txdetectrx_i	Input	Receiver detect control signal	pipe_direct_pld_tx_clk_out_o
lnX_pipe_direct_rxtermination_i	Input	Controls the presence of receiver terminations. This is a PIPE signal mainly intended for USB usage. Intel recommends driving this signal high (default). • 0 = Terminations removed. • 1 = Terminations present.	pipe_direct_pld_tx_clk_out_o
lnX_pipe_direct_pclkchangeack_i	Input	Asserted by the MAC when a PCLK rate change or, if required, width change is complete and stable.	pipe_direct_pld_tx_clk_out_o
lnX_pipe_direct_tx_transfer_en_o	Output	This signal indicates when the EMIB is ready in PIPE mode. The soft IP controller must release the per-lane lnX_pipe_direct_pld_pcs_rst_n_i signal out of reset after the per-lane lnX_pipe_direct_tx_transfer_en_o signal is asserted.	pipe_direct_pld_tx_clk_out_o
lnX_pipe_direct_rxstandbystatus_o	Output	Indicates whether the PHY is active or in standby mode. • 0 = Active • 1 = Standby	pipe_direct_pld_tx_clk_out_o
lnX_pipe_direct_pclkratechangeok_o	Output	This signal is asserted by the PHY when it is ready for the MAC to change the clock rate.	pipe_direct_pld_tx_clk_out_o
lnX_pipe_direct_rxstatus_o	Output	Reflects the state of the high-speed receiver. A 1 on this bit indicates Rx is detected. *Note:* The only status applicable to the PIPE SerDes architecture mode is "Receiver detected".	pipe_direct_pld_tx_clk_out_o
lnX_pipe_direct_phystatus_o	Output	Indicates the completion of several PHY functions including stable PCLK, after reset deassertion, power management state transitions, rate change and receiver detection.	pipe_direct_pld_tx_clk_out_o
lnX_pipe_direct_cdrlockstatus_o	Output	This is the Receiver CDR lock indicator. • 0 = CDR is not locked and is not at the correct frequency. • 1 = CDR is at the correct frequency. If this signal is deasserted when it is expected to be asserted, it indicates a fault condition and the receiver should be reset.	Async
lnX_pipe_direct_cdrlock2data_o	Output	This is the Receiver CDR data lock indicator. • 0 = CDR is not locked to the data. • 1 = CDR is locked to the data. RX data is valid.	Async

Signal Name	Direction	Descriptions/Notes	Clock Domain
		Note: This signal will go low when the Soft IP Controller instructs the R-Tile Avalon Streaming IP to start an evaluation of the far end transmitter TX EQ settings (by setting the RxEqEval bit in the Rx Control 3 register). Once the evaluation is completed, the R-Tile Avalon Streaming IP will provide a Figure of Merit value and drive this signal high. For additional details on the Equalization sequence, refer to section 9.10 of the PIPE Specification 5.1.1.	

4.4.3. Message Bus Signals

This interface is used for PHY (P)-MAC (M) communication, using the P2M and M2P unidirectional 8-bit signals per lane. This interface serves to reduce the number of dedicated signals for implementations like link equalization, lane margining etc.

Table 83. PIPE Direct EMIB Control Message Channel M2P/P2M Signals

Signal Name	Direction	Descriptions/Notes	Clock Domain
`lnX_pipe_direct_m2p_messagebus_i[7:0]`	Input	Low Pin Count Message Interface from MAC to PHY.	`pipe_direct_pld_tx_clk_out_o`
`lnX_pipe_direct_p2m_messagebus_o[7:0]`	Output	Low Pin Count Message Interface from PHY to MAC.	`pipe_direct_pld_tx_clk_out_o`

Note: The R-Tile in PIPE Direct mode does not support querying the FS/LS/Preset/Coefficient values via the PIPE interface. In addition, EQ feedback direction is not supported. Only the Figure of Merit (FOM) must be implemented using the following values:

- For the OPNs AGIx027R29AxxxxR0, AGIx027R29AxxxxR1:
  - For Gen3/Gen4/Gen5: FS = 48, LF = 16
  - For Gen3/Gen5:

    {c-1, c0, c+1 = preset} = {8/40/0 = P9}
  - For Gen4:

    {c-1, c0, c+1 = preset} = {0/42/6 = P3}
- For the OPNs AGIx027R29AxxxxR2, AGIx027R29AxxxxR3, AGIx027R29BxxxxR3, AGIx023R18AxxxxR0, AGIx041R29DxxxxR0, AGIx041R29DxxxxR1:
  - For Gen3/Gen4/Gen5: FS = 48, LF = 16
  - For Gen5:

    {c-1, c0, c+1 = preset} = {6/36/6 = P8}

    {c-1, c0, c+1 = preset} = {8/40/0 = P9}
  - For Gen4:

    {c-1, c0, c+1 = preset} = {0/42/6 = P3}

    {c-1, c0, c+1 = preset} = {5/33/10 = P7}
  - For Gen3:

    {c-1, c0, c+1 = preset} = {0/48/0 = P4}

For additional details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

4.4.4. Deskew Signals

In PIPE Direct mode, the R-Tile Avalon Streaming Intel FPGA IP for PCI Express removes any lane-to-lane skew introduced while crossing the EMIB. A dedicated deskew marker is used for detecting and compensating for any multi-lane skew introduced by EMIB. The deskew logic can account for a maximum of up to 3 cycles of parallel skew. After a cold/warm/hot reset or CvP update, the deskew process will start.

The user application logic needs to send a deskew marker every 16 clock cycles for the purpose of deskewing the data on the EMIB channels. The R-Tile Avalon Streaming Intel FPGA IP for PCI Express runs the deskew process every time it receives the deskew marker.

Table 84. EMIB TX Deskew Grouping for R-Tile Topologies

PIPE Direct Tx Deskew Bundle	Octet 1								Octet 0							
	Lane 15	Lane 14	Lane 13	Lane 12	Lane 11	Lane 10	Lane 9	Lane 8	Lane 7	Lane 6	Lane 5	Lane 4	Lane 3	Lane 2	Lane 1	Lane 0
1X16	Octet1_Dsk_0								Octet0_Dsk_0							
2X8	Octet1_Dsk_0								Octet0_Dsk_0							
4X4	Octet1_Dsk_2				Octet1_Dsk_0				Octet0_Dsk_2				Octet0_Dsk_0			
8X2	Octet1_Dsk_3		Octet1_Dsk_2		Octet1_Dsk_1		Octet1_Dsk_0		Octet0_Dsk_3		Octet0_Dsk_2		Octet0_Dsk_1		Octet0_Dsk_0	
16X1	No Tx Deskew															
2X4 ; 1X8	Octet1_Dsk_2				Octet1_Dsk_0				Octet0_Dsk_0							
4X2 ; 1X8	Octet1_Dsk_3		Octet1_Dsk_2		Octet1_Dsk_1		Octet1_Dsk_0		Octet0_Dsk_0							
8X1 ; 1X8	No Tx Deskew								Octet0_Dsk_0							
1X8 ; 2X4	Octet1_Dsk_0								Octet0_Dsk_2				Octet0_Dsk_0			
4X2 ; 2X4	Octet1_Dsk_3		Octet1_Dsk_2		Octet1_Dsk_1		Octet1_Dsk_0		Octet0_Dsk_2				Octet0_Dsk_0			
8X1 ; 2X4	No Tx Deskew								Octet0_Dsk_0				Octet0_Dsk_0			
1X8 ; 4X2	Octet1_Dsk_0								Octet0_Dsk_3		Octet0_Dsk_2		Octet0_Dsk_1		Octet0_Dsk_0	
2X4 ; 4X2	Octet1_Dsk_2				Octet1_Dsk_0				Octet0_Dsk_3		Octet0_Dsk_2		Octet0_Dsk_1		Octet0_Dsk_0	
8X1 ; 4X2	No Tx Deskew								Octet0_Dsk_3		Octet0_Dsk_2		Octet0_Dsk_1		Octet0_Dsk_0	
1X8 ; 8X1	Octet1_Dsk_0								No Tx Deskew							
2X4 ; 8X1	Octet1_Dsk_2				Octet1_Dsk_0				No Tx Deskew							
4X2 ; 8X1	Octet1_Dsk_3		Octet1_Dsk_2		Octet1_Dsk_1		Octet1_Dsk_0		No Tx Deskew							

4.4.4.1. MAC to PHY (M2P) Signals

Table 85. PIPE Direct EMIB Control Deskew Channel - M2P Signals

Signal Name	Direction	Descriptions/Notes	Clock Domain
lnX_pipe_direct_deskew_clear_0/1/2/3_i	Input	When asserted, the current tx_dsk_eval_done and tx_dsk_status are cleared. A new deskew evaluation is expected after the current status is cleared. This signal is asserted for two clock cycles.	pipe_direct_pld_tx_clk_out_o

4.4.4.2. PHY to MAC (P2M) Signals

Table 86. PIPE Direct EMIB Control Deskew Channel P2M Signals

Signal Name	Direction	Descriptions/Notes	Clock Domain
lnX_pipe_direct_txdeskewmarker_i	Input	Tx Deskew marker used to deskew EMIB routings per bundle mode. This is a simple repeating pulse that provides a protocol-agnostic mechanism to detect EMIB channel skew and perform alignment. The marker fans out and appears on all bundle channels simultaneously once every 16 clock cycles. The deskew module looks for the deskew marker from each EMIB channel and adds delays on the early channels to compensate for the delays of the late channels.	pipe_direct_pld_tx_clk_out_o
octet#_pipe_direct_phy_dsk_active_chans_o	Output	Indicates which channels received a deskew marker.	pipe_direct_pld_tx_clk_out_o
octet#_pipe_direct_phy_dsk_monitor_err_o	Output	Value is latched upon an error, and held until the state machine is restarted via i_dsk_clear or async reset. Monitor this signal only after 16 pclk (pipe_direct_pld_tx_clk_out_o) cycles following the assertion of octet#_pipe_direct_phy_dsk_eval_done_[3:0]_o.	pipe_direct_pld_tx_clk_out_o
octet#_pipe_direct_phy_dsk_monitor_err_status_[3:0]_o	Output	Indicates a deskew monitor error. Monitor this signal only after 16 pclk (pipe_direct_pld_tx_clk_out_o) cycles following the assertion of octet#_pipe_direct_phy_dsk_eval_done_[3:0]_o.	pipe_direct_pld_tx_clk_out_o
octet#_pipe_direct_phy_dsk_status_[3:0]_o	Output	Indicates the deskew evaluation result. Monitor this signal only after 16 pclk (pipe_direct_pld_tx_clk_out_o) cycles following the assertion of octet#_pipe_direct_phy_dsk_eval_done_[3:0]_o. *Note:* octet#_pipe_direct_phy_dsk_status should be monitored only after octet#_pipe_direct_phy_dsk_valid is asserted.	pipe_direct_pld_tx_clk_out_o
octet#_pipe_direct_phy_dsk_valid_[3:0]_o	Output	Indicates the deskew operation status.	pipe_direct_pld_tx_clk_out_o

continued...

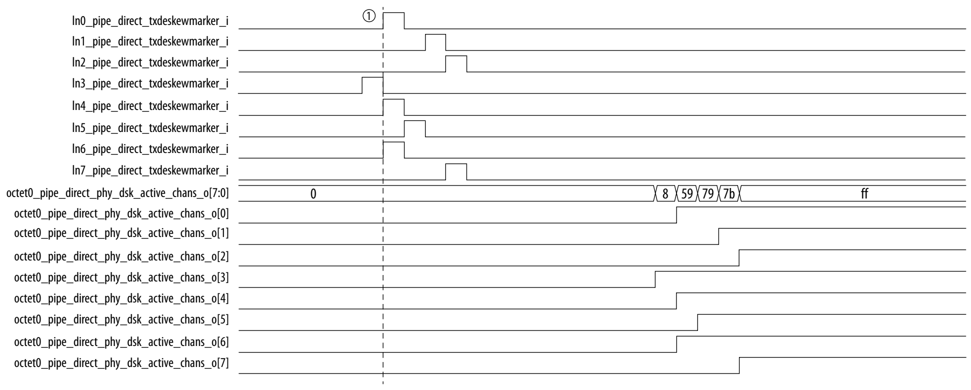
Signal Name	Direction	Descriptions/Notes	Clock Domain
		When using x16, the octet#_pipe_direct_phy_dsk_valid_o from each of the octets must be ANDed together by the user logic.	
octet#_pipe_direct_phy_dsk_eval_done_[3:0]_o	Output	Indicates the deskew process is complete. This signal is for debugging purpose. When using x16, the octet#_pipe_direct_phy_dsk_eval_done_o from each of the octets must be ANDed together by the user logic.	pipe_direct_pld_tx_clk_out_o

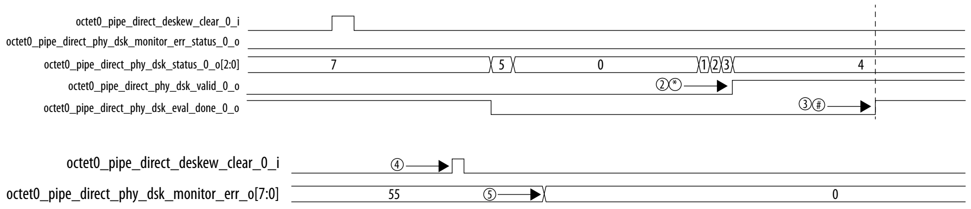
To use the deskew interface, follow these steps:

1. The controller in the application logic sends the deskew marker for each lane of the bundle every 16 pipe_direct_pld_tx_clk_out_o clock cycles using the signal ln*_pipe_direct_txdeskewmarker_i.
2. After the data from the EMIB is deskewed, octet*_pipe_direct_phy_dsk_valid_o is asserted, indicating deskew done status.

   Note: (*) When using x16, the octet*_pipe_direct_phy_dsk_valid_o from each of the octets must be ANDed together.
3. In addition to the octet*_pipe_direct_phy_dsk_valid_o signals, the PIPE interface provides octet*_pipe_direct_phy_dsk_eval_done_o and octet*_pipe_direct_phy_dsk_status_*_o signals to show the details of the deskew status.

   Note: (#) These signals are for debugging purposes only. The user application logic should rely only on the octet*_pipe_direct_phy_dsk_valid_o signals.
4. The octet*_pipe_direct_deskew_clear_i signals on both octets can be used to clear the current deskew state to allow additional deskew evaluations. When using x16, the octet*_pipe_direct_deskew_clear_i for each of the octets must be used.
5. After the pulse on octet*_pipe_direct_deskew_clear_i, the deskew state on octet*_pipe_direct_phy_dsk_monitor_err_o bus is cleared.


ln0_pipe_direct_txdeskewmarker_i
ln1_pipe_direct_txdeskewmarker_i
ln2_pipe_direct_txdeskewmarker_i
ln3_pipe_direct_txdeskewmarker_i
ln4_pipe_direct_txdeskewmarker_i
ln5_pipe_direct_txdeskewmarker_i
ln6_pipe_direct_txdeskewmarker_i
ln7_pipe_direct_txdeskewmarker_i
octet0_pipe_direct_phy_dsk_active_chans_o[7:0]
octet0_pipe_direct_phy_dsk_active_chans_o[0]
octet0_pipe_direct_phy_dsk_active_chans_o[1]
octet0_pipe_direct_phy_dsk_active_chans_o[2]
octet0_pipe_direct_phy_dsk_active_chans_o[3]
octet0_pipe_direct_phy_dsk_active_chans_o[4]
octet0_pipe_direct_phy_dsk_active_chans_o[5]
octet0_pipe_direct_phy_dsk_active_chans_o[6]
octet0_pipe_direct_phy_dsk_active_chans_o[7]
①
0
8
59
79
7b
ff


octet0_pipe_direct_deskew_clear_0_i
octet0_pipe_direct_phy_dsk_monitor_err_status_0_o
octet0_pipe_direct_phy_dsk_status_0_o[2:0]
octet0_pipe_direct_phy_dsk_valid_0_o
octet0_pipe_direct_phy_dsk_eval_done_0_o
7
5
0
1 2 3
4
octet0_pipe_direct_deskew_clear_0_i
octet0_pipe_direct_phy_dsk_monitor_err_o[7:0]
55
0

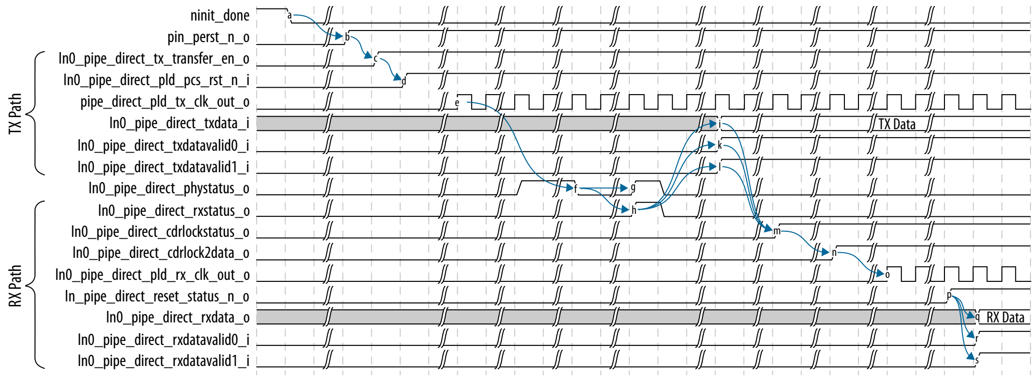
4.4.5. PIPE Direct Reset Sequence

In PIPE Direct mode, your application logic is responsible for managing most of the PHY reset sequence in the FPGA fabric. The following figure describes the required sequence.

Note: In case your Soft IP Controller is only using the lanes 8-15, the following procedure is required for lane 0:

1. Deassert `ln0_pipe_direct_pld_pcs_rst_n_i` as described in Step d of the reset sequence shown in PIPE Direct Reset Sequence.
2. Set `ln0_pipe_direct_powerdown_i` = 2'b00, until any of the 8-15 lanes has reached Step f of the reset sequence shown in PIPE Direct Reset Sequence.
3. After this, set `ln0_pipe_direct_powerdown_i` = 2'b11 as required in Unused Lanes in PIPE Direct Mode.

Figure 46. PIPE Direct Reset Sequence


ninit_done
pin_perst_n_o
ln0_pipe_direct_tx_transfer_en_o
ln0_pipe_direct_pld_pcs_rst_n_i
pipe_direct_pld_tx_clk_out_o
ln0_pipe_direct_txdata_i
ln0_pipe_direct_txdatavalid0_i
ln0_pipe_direct_txdatavalid1_i
ln0_pipe_direct_physstatus_o
ln0_pipe_direct_rxstatus_o
ln0_pipe_direct_cdrlockstatus_o
ln0_pipe_direct_cdrlock2data_o
ln0_pipe_direct_pld_rx_clk_out_o
ln_pipe_direct_reset_status_n_o
ln0_pipe_direct_rxdata_o
ln0_pipe_direct_rxdatavalid0_i
ln0_pipe_direct_rxdatavalid1_i
TX Path
RX Path
TX Data
RX Data
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s

Below are the steps required for the reset sequence and the TX/RX data transfer for lane 0 in the R-Tile Avalon Streaming IP when configured in PIPE-D mode. This behavior applies in the same way for other lanes.

Please note that each of the required steps correlates with the corresponding letter in the waveforms.

For the TX path:

1. Step (a) : ninit_done is driven low by the Reset Release IP indicating the FGPA fabric is configured. The Soft IP controller should be in reset until this signal is low.
2. Step (b) : pin_perst_n_o is driven high by the R-Tile Avalon Streaming IP. This signal reflects the PERTS# signal at the board level.
3. Step (c) : lnX_pipe_direct_tx_transfer_en_o is driven high by the R-Tile Avalon Streaming IP indicating the EMIB bridge between the R-Tile Avalon Streaming IP and the FPGA fabric is ready.
4. Step (d) : lnX_pipe_direct_pld_pcs_rst_n_i is driven high by the Soft IP controller. The Soft IP controller must also drive high the per-lane lnX_pipe_direct_pld_pcs_rst_n_i signal to come out of reset after the per-lane lnX_pipe_direct_tx_transfer_en_o signal is driven high.
5. Step (e) : pipe_direct_pld_tx_clk_out_o becomes active as the TX clock output to be used by the Soft IP controller for the TX path.
6. Step (f) : lnX_pipe_direct_phystatus_o is driven low by the R-Tile Avalon Streaming IP indicating a reset exit.
7. Step (g) : lnX_pipe_direct_phystatus_o is pulsed and in
8. Step (h) : lnX_pipe_direct_rx_status_o is pulsed as well. Both pulses confirm to the Soft IP controller the RX detection.
9. Step (j) : The Soft IP controller starts sending data on the lnX_pipe_direct_txdata_i bus along with its corresponding lnX_pipe_direct_txdatavalid0_i at Step (k) and lnX_pipe_direct_txdatavalid1_i signals at Step (l). Refer to PIPE Direct TX Datapath for additional details.

For the RX path:

1. Step (m) : After TX data is transmitted from the Soft IP controller and once enough RX data has been received from the link partner to recover the clock, the lnX_pipe_direct_cdrlockstatus_o signal is driven high.
2. Step (n) : The lnX_pipe_direct_cdrlock2data_o signal is driven high indicating the CDR has locked to the data being received.
3. Step (o) : The lnX_pipe_direct_rx_clk_out_o signal becomes active as the RX clock output to be used by the Soft IP controller for the RX data path.
4. Step (p) : The ln_pipe_direct_reset_status_n_o signal is driven high by the R-Tile Avalon Streaming IP indicating the RX data path is out of reset.
5. Step (q) : The Soft IP controller starts sampling data on the lnX_pipe_direct_rxdata_o while qualifying the data with its corresponding lnX_pipe_direct_rxdatavalid0_i and lnX_pipe_direct_rxdatavalid1_i signals. Application logic needs to wait for the assertion of the corresponding lane's ln_pipe_direct_reset_status_n_o[15:0] to sample the Rx data. Refer to fPIPE Direct RX Datapathor additional details.

4.4.6. PIPE Direct Speed Change

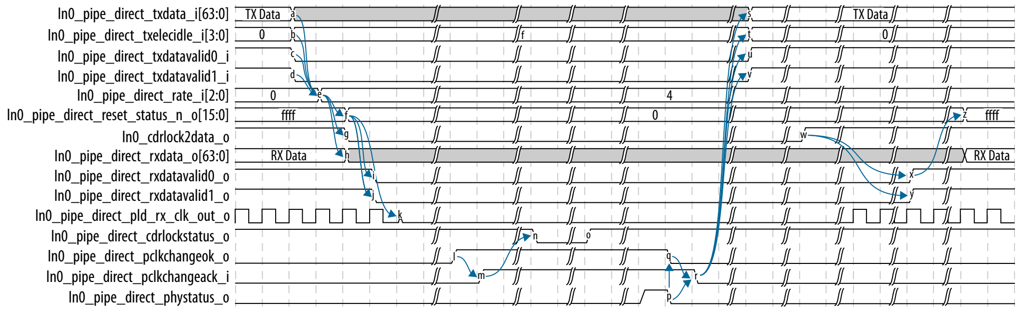
In the PIPE Direct Data mode, the clock for the RX datapath is sourced from the PHY recovered clock (`pipe_direct_pld_rx_clk_out_o`). The PHY recovered clock changes frequency when the PHY trains from Gen1 to Gen5. During the PIPE Direct RX rate change, the following sequence needs to be adhered to.

The soft IP controller first changes the rate or width if required. The R-Tile Avalon Streaming IP only asserts `lnX_pipe_direct_pclkchangeok_o` after the Soft IP controller has made the changes. The Soft IP controller asserts `lnX_pipe_direct_pclkchangeack_i` when the change is complete and stable. After the Soft IP controller asserts `lnX_pipe_direct_pclkchangeack_i`, the R-Tile Avalon Streaming IP responds by asserting `lnX_pipe_direct_phystatus_o` for one cycle and deasserting `lnX_pipe_direct_pclkchangeok_o` at the same time as `lnX_pipe_direct_phystatus_o`. The Soft IP controller deasserts `lnX_pipe_direct_pclkchangeack_i` when `lnX_pipe_direct_pclkchangeok_o` is sampled low.

As a reference, the following diagram illustrates the speed change from Gen1 to Gen5.

Note: Although the diagram below illustrates a speed change from Gen1 to Gen5, the overall sequence applies to all speed changes. However, the final value of `ln0_pipe_direct_rate_i` varies depending on what the final speed is.

Figure 47. PIPE Direct Speed Change


ln0_pipe_direct_txdata_i[63:0]
ln0_pipe_direct_txelecidle_i[3:0]
ln0_pipe_direct_txdatavalid0_i
ln0_pipe_direct_txdatavalid1_i
ln0_pipe_direct_rate_i[2:0]
ln0_pipe_direct_reset_status_n_o[15:0]
ln0_cdrlock2data_o
ln0_pipe_direct_rxdata_o[63:0]
ln0_pipe_direct_rxdatavalid0_o
ln0_pipe_direct_rxdatavalid1_o
ln0_pipe_direct_pld_rx_clk_out_o
ln0_pipe_direct_cdrlockstatus_o
ln0_pipe_direct_pclkchangeok_o
ln0_pipe_direct_pclkchangeack_i
ln0_pipe_direct_phystatus_o
TX Data
0
4
ffff
RX Data

Below are the steps required for the speed change sequence for lane 0 in the R-Tile Avalon Streaming IP when configured in PIPE Direct mode. This behavior also applies to other lanes and other speed rates.

Note that each of the steps required correlates with the corresponding letter in the waveforms.

- Steps (a, b, c, d): The Soft IP controller stops the data transmission on the TX data path signals in preparation for the speed change event.
- Step (e): Once ready, the Soft IP Controller will set the targeted rate on the `ln0_pipe_direct_rate_i` signal.
- Steps (f, g, h): The `ln_pipe_direct_reset_status_n_o` signal goes low, invalidating any further data received on the `ln0_pipe_direct_rxdata_o` bus. Additionally, the `ln0_cdrlock2data_o` signal goes low.

  Note: The Rx data must be qualified with an AND operation between the corresponding `ln_pipe_direct_reset_status_n_o` lane signal, `ln0_pipe_direct_rxdatavalid0_o` and `ln0_pipe_direct_rxdatavalid1_o`.

- Steps (i,j): The `ln0_pipe_direct_rxdatavalid0_o` and `ln0_pipe_direct_rxdatavalid1_o` signal go low.

  Note: The Rx data must be qualified with an AND operation between the corresponding `ln_pipe_direct_reset_status_n_o` lane signal, `ln0_pipe_direct_rxdatavalid0_o` and `ln0_pipe_direct_rxdatavalid1_o`.

- Step (k): The `ln0_pipe_direct_pld_rx_clk_out_o` stops toggling.
- Steps (l, m): The R-Tile Avalon Streaming IP asserts `ln0_pipe_direct_pclkchangeok_o` and the Soft IP Controller confirms by driving high the `ln0_pipe_direct_pclkchangeack_i` signal.
- Step (n): The `ln0_pipe_direct_cdrlockstatus_o` signal goes low until the R-Tile Avalon Streaming IP locks to the new clock frequency.
- Step (o): The `ln0_pipe_direct_cdrlockstatus_o` signal goes high once the R-Tile Avalon Streaming IP locks to the new clock frequency.
- Steps (p, q): The R-Tile Avalon Streaming IP confirms a successful rate change with a single pulse on the `ln0_pipe_direct_phystatus_o` signal and by driving low the `ln0_pipe_direct_pclkchangeok_o` signal.
- Step (r): The Soft IP Controller acknowledges the rate change by driving low the `ln0_pipe_direct_pclkchangeack_i` signal.
- Steps (s, t, u, v): The Soft IP Controller starts the data transmission at the new rate on the TX data path signals.
- Steps (w, x, y): The `ln0_cdrlock2data_o` signal is driven high by the R-Tile Avalon Streaming IP. The `ln0_pipe_direct_rxdatavalid0_o` and the `ln0_pipe_direct_rxdatavalid1_o` signals go high.

  Note: The RX data is not valid until the corresponding `ln_pipe_direct_reset_status_n_o` lane signal goes high.

- Step (z): The corresponding `ln_pipe_direct_reset_status_n_o` lane signal goes high. This qualifies the RX data along with the `ln0_pipe_direct_rxdatavalid0_o` and `ln0_pipe_direct_rxdatavalid1_o` signals.

4.4.7. PIPE Lane Reset Staggering

This section focuses on the staggering for the lane resets of the R-Tile Avalon Streaming IP. These lane resets are driven by a Soft IP controller. The IP controller executes a part of the reset procedure for the R-Tile Avalon Streaming IP. It only performs the lane reset procedure when the Soft IP itself is out of reset. Each lane has its own `lnX_pipe_direct_pld_pcs_rst_n_i` driven by the Soft IP controller. When multiple Soft IP controllers are being implemented, each Soft IP can reset its respective lane independent of the other Soft IPs.

Reset staggering is optional. The staggering is to reduce the power distribution network (PDN) noise during power-up. If implemented, the reset staggering should start after `lnX_pipe_direct_tx_transfer_en` is asserted.

The staggering interval must be set to more than 100ns. The lane resets are staggered in such a way that they start from lane 0 to lane 15 for the PIPE Direct x16 mode and lane0 to lane 7 for any PIPE Direct bundle mode. Both reset assertion and deassertion must be staggered.

The `lnX_pipe_direct_powerdown_i` and `lnX_pipe_direct_rxstandby_i` signals also require lane staggering per bundle. The staggering must be greater or equal to a x4 bundle. For example, in the PIPE Direct 8x2 bundle mode, the lane staggering for the mentioned PIPE signals must be grouped in x4 or greater.

4.4.8. Unused Lanes in PIPE Direct Mode

In PIPE Direct mode, the Soft IP controller is responsible for driving the `lnX_pipe_direct_powerdown_i[1:0]` to 2'`b11` in the FPGA core fabric for those unused lanes. Upon entering user mode followed by the deassertion of `ln0_pipe_direct_pld_pcs_rst_n_i` by the Soft IP controller, the unused lanes will transition to P2 pstate.

5. Parameters

This chapter provides a reference for all the parameters that are configurable in the Intel Quartus Prime IP Parameter Editor for the R-Tile Avalon Streaming Intel FPGA IP for PCIe.

5.1. Top-Level Settings

Table 87. Top-Level Settings

Parameter	Value	Default Value	Description
Hard IP Mode	**Gen5 1x16, Interface - 1024-bit** **Gen4 1x16, Interface - 1024-bit** **Gen3 1x16, Interface - 1024-bit** **Gen4 1x16, Interface - 512-bit** **Gen3 1x16, Interface - 512-bit** **Gen5 2x8, Interface - 512-bit** **Gen4 2x8, Interface - 512-bit** **Gen3 2x8, Interface - 512-bit** **Gen4 2x8, Interface - 256-bit** **Gen3 2x8, Interface - 256-bit** **Gen5 4x4, Interface - 256-bit** **Gen4 4x4, Interface - 256-bit** **Gen3 4x4, Interface - 256-bit** **Gen4 4x4, Interface - 128-bit** **Gen3 4x4, Interface - 128-bit** **PIPE Direct 16-channel**	**Gen5 1x16, Interface - 1024-bit**	Selects the width of the data interface between the transaction layer and the application layer implemented in the FPGA fabric, and the lane rate. Select the following elements: Lane data rate: • Gen3, Gen4 and Gen5 are supported. *Note:* When selecting Gen3 or Gen4 Hard IP modes, the R-Tile Avalon streaming Intel FPGA IP for PCI Express continues to advertise its capabilities as a device compliant with the 5.0 PCI Express Base Specification. Lane width: • x16, x8 and x4 are supported. Some of these configurations are only available in the following OPNs: • AGIx027R29AxxxxR2 • AGIx027R29AxxxxR3 • AGIx027R29BxxxxR3 • AGIx023R18AxxxxR0 • AGIx041R29DxxxxR0 • AGIx041R29DxxxxR1 . For additional details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview. For more details, refer to Avalon Streaming Interface Data and Header Bus Widths per Port.
Port Mode	**Native Endpoint** **Root Port**	**Native Endpoint**	Specifies the port type.

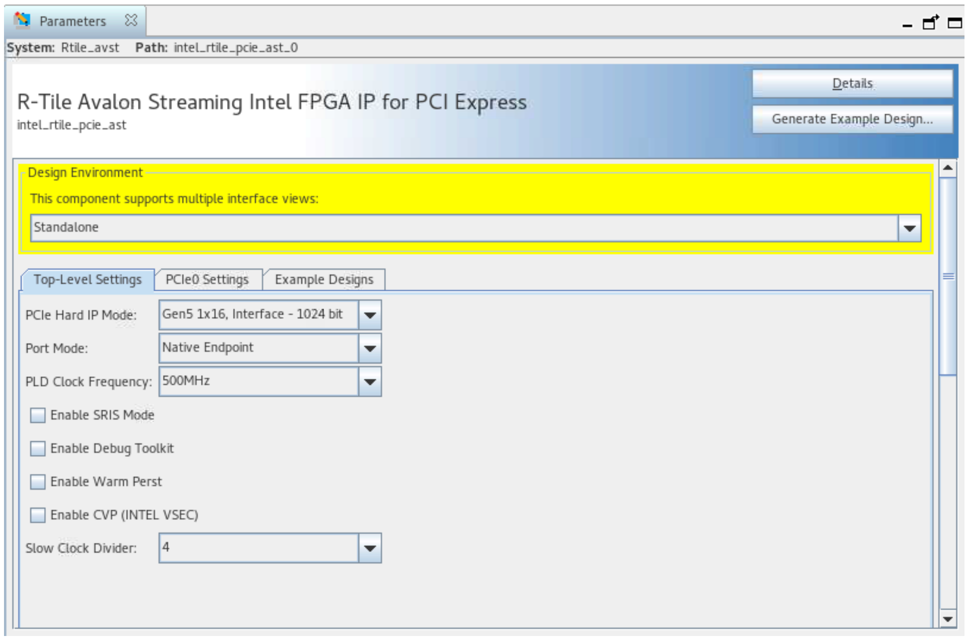
continued...

Parameter	Value	Default Value	Description
	Downstream **Upstream**		*Note:* To operate the IP in TLP Bypass mode, choose either the **Downstream** or **Upstream** Port Mode.
PLD Clock Frequency	**500 MHz** **475 MHz** **450 MHz** **425 MHz** **400 MHz** **300 MHz** **275 MHz** **250 MHz**	**500 MHz** (for Gen5 modes) **500 MHz / 300 MHz** (for Gen4 and Gen3 modes, the default frequency depends on whether the IP is in double-width or single-width mode)	Select the frequency of the Application clock. The options available vary depending on the setting of the **Hard IP Mode** parameter. For Gen5 modes, the available clock frequencies are 500 MHz / 475 MHz / 450 MHz / 425 MHz / 400 MHz. For Gen4 and Gen3 modes, the available clock frequencies depend on whether the IP is in double-width or single-width mode. For more details, refer to IP to FPGA Fabric Interface Summary.
Enable SRIS Mode	**True/False**	**False**	Enable the Separate Reference Clock with Independent Spread Spectrum Clocking (SRIS) feature.
Enable Debug Toolkit	**True/False**	**False**	When set, this parameters enables the Debug Tookit feature. For more information, refer to Debug Toolkit.
Enable Warm Perst	**True/False**	**False**	Only available when the parameter **Hard IP Mode** is set to **1x16**. When set, the additional `pX_cold_perst` and `pX_warm_perst` are exposed. For additional information on the usage of these ports, refer to Reset.
Enable Independent GPIO Perst	**True/False**	**False**	Only available when the parameter **Hard IP Mode** is set to **2x8**. When set, the additional `pX_cold_perst` and `pX_warm_perst` are exposed. For additional information on the usage of these ports, refer to Reset.
Enable Independent Perst Pins	**True/False**	**False**	Only available when the parameter **Hard IP Mode** is set to **2x8** (Endpoint only), and a supported device is selected in the Intel Quartus Prime project. When set, the additional `pX_perst0_n`, `pX_perst1_n`, and `pX_warm_perst` are exposed. For a list of the supported devices and additional information on the usage of these ports, refer to Reset.
Enable CVP (INTEL VSEC)	**True/False**	**False**	When set, this parameter enables CVP for a single tile only.

continued...

Parameter	Value	Default Value	Description
Slow Clock Divider	**2** **4**	**4**	Allows you to set the `slow_clk` to be divided by 2 or 4 from the `coreclkout_hip`.
PIPE Direct Mode	**1x16** **2x8** **4x4** **8x2** **2x4 / 1x8** **4x2 / 1x8** **8x1 / 1x8** **1x8 / 2x4** **4x2 / 2x4** **8x1 / 2x4** **1x8 / 4x2** **2x4 / 4x2** **8x1 / 4x2** **1x8 / 8x1** **2x4 / 8x1** **4x2 / 8x1** **16x1**	**16x1**	This option provides lane to lane TX deskew per grouping based on topologies. For example, if the **2x8** topology is selected, the EMIB is deskewed per every eight EMIB channels. This results in two sets of deskew markers, deskew errors, etc.
Octet 0 Active Lanes	**Off** **1** **2** **3** **4** **5** **6** **7** **8**	**8**	Enables for lanes 0-7. This parameter indicates how many Lower lanes are used by the user application logic.
Octet 1 Active Lanes	**Off** **1** **2** **3** **4** **5** **6** **7** **8**	**8**	Enables for lanes 8-15. This parameter indicates how many Upper lanes are used by the user application logic.

Figure 48. Intel R-Tile Avalon Streaming Top-Level IP Parameter Editor for PCIe Gen5 1x16 Mode



System: Rtile_avst Path: intel_rtile_pcie_ast_0

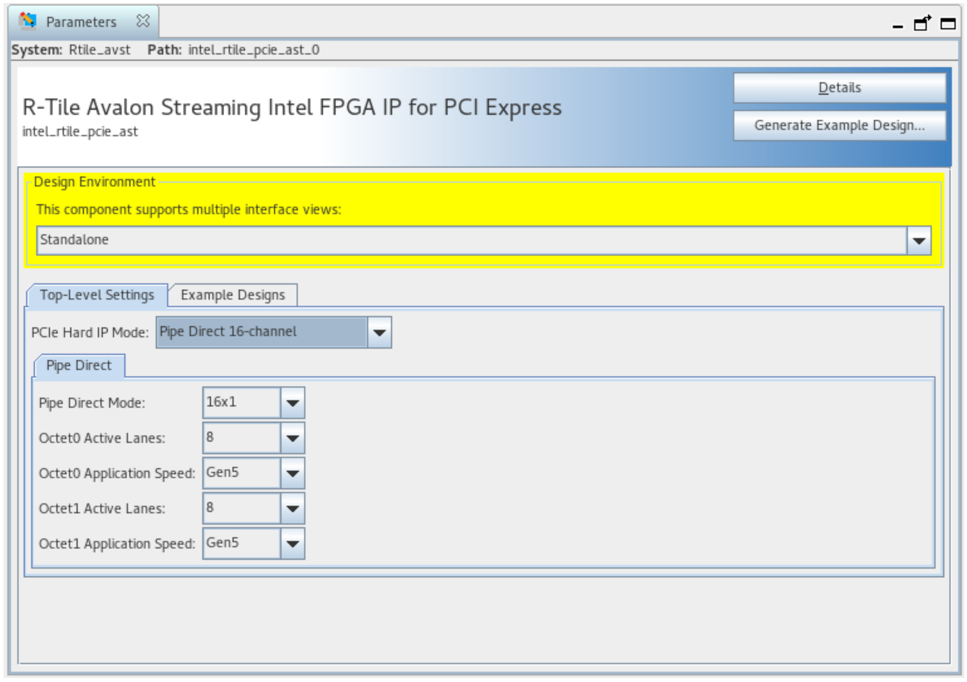
R-Tile Avalon Streaming Intel FPGA IP for PCI Express
intel_rtile_pcie_ast

Design Environment
This component supports multiple interface views:
Standalone

Top-Level Settings | PCIe0 Settings | Example Designs

PCIe Hard IP Mode: Gen5 1x16, Interface - 1024 bit
Port Mode: Native Endpoint
PLD Clock Frequency: 500MHz
☐ Enable SRIS Mode
☐ Enable Debug Toolkit
☐ Enable Warm Perst
☐ Enable CVP (INTEL VSEC)
Slow Clock Divider: 4

Figure 49. Intel R-Tile Avalon Streaming Top-Level IP Parameter Editor for PIPE Direct Mode



System: Rtile_avst Path: intel_rtile_pcie_ast_0

R-Tile Avalon Streaming Intel FPGA IP for PCI Express
intel_rtile_pcie_ast

Design Environment
This component supports multiple interface views:
Standalone

Top-Level Settings | Example Designs

PCIe Hard IP Mode: Pipe Direct 16-channel

Pipe Direct

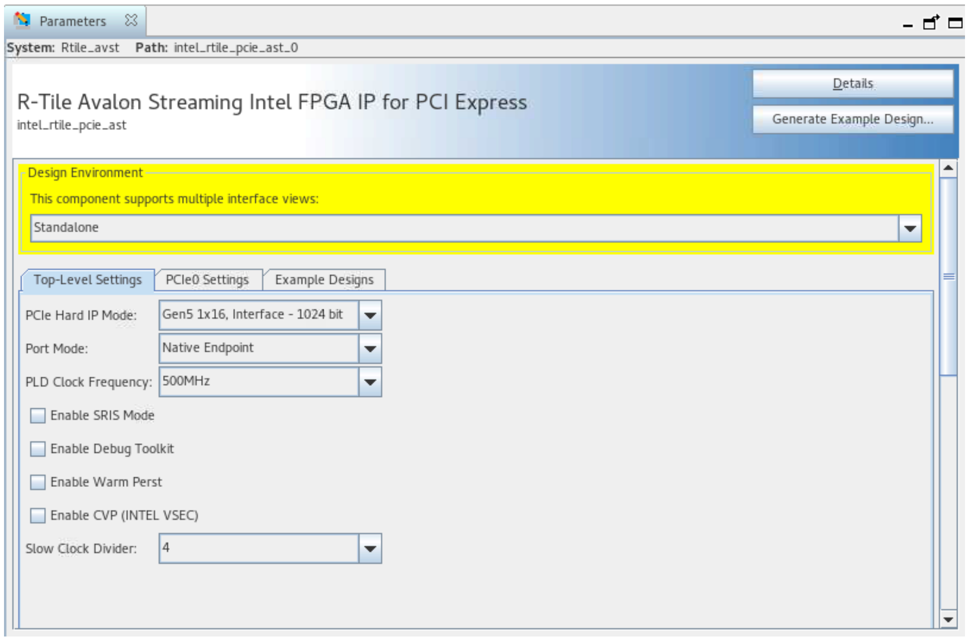
Pipe Direct Mode: 16x1
Octet0 Active Lanes: 8
Octet0 Application Speed: Gen5
Octet1 Active Lanes: 8
Octet1 Application Speed: Gen5

5.2. Core Parameters

Depending on which **Hard IP Mode** you choose in the **Top-Level Settings** tab, you will see different tabs for setting the core parameters.

If you choose a 1x16 mode (Gen3, Gen4 or Gen5), only the **PCIe0 Settings** tab will appear.

Figure 50. Intel R-Tile Avalon Streaming Top-Level IP Parameter Editor for PCIe Gen5 1x16 Mode



System: Rtile_avst Path: intel_rtile_pcie_ast_0

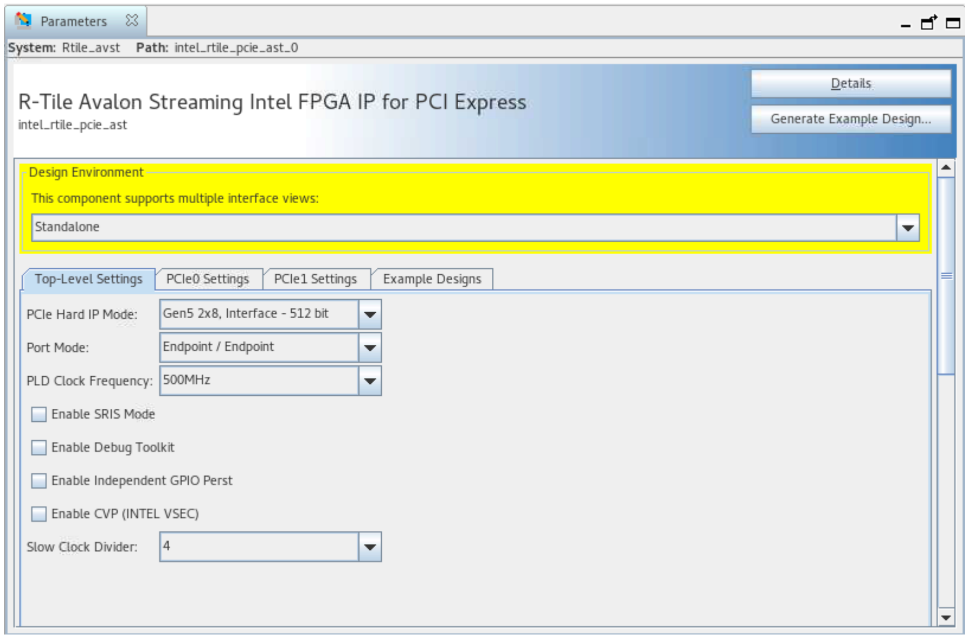
R-Tile Avalon Streaming Intel FPGA IP for PCI Express
intel_rtile_pcie_ast

Design Environment
This component supports multiple interface views:
Standalone

PCIe Hard IP Mode: Gen5 1x16, Interface - 1024 bit
Port Mode: Native Endpoint
PLD Clock Frequency: 500MHz
☐ Enable SRIS Mode
☐ Enable Debug Toolkit
☐ Enable Warm Perst
☐ Enable CVP (INTEL VSEC)
Slow Clock Divider: 4

If you choose a 2x8 mode (Gen3, Gen4 or Gen5), only the **PCIe0 Settings** and **PCIe1 Settings** tabs will appear.

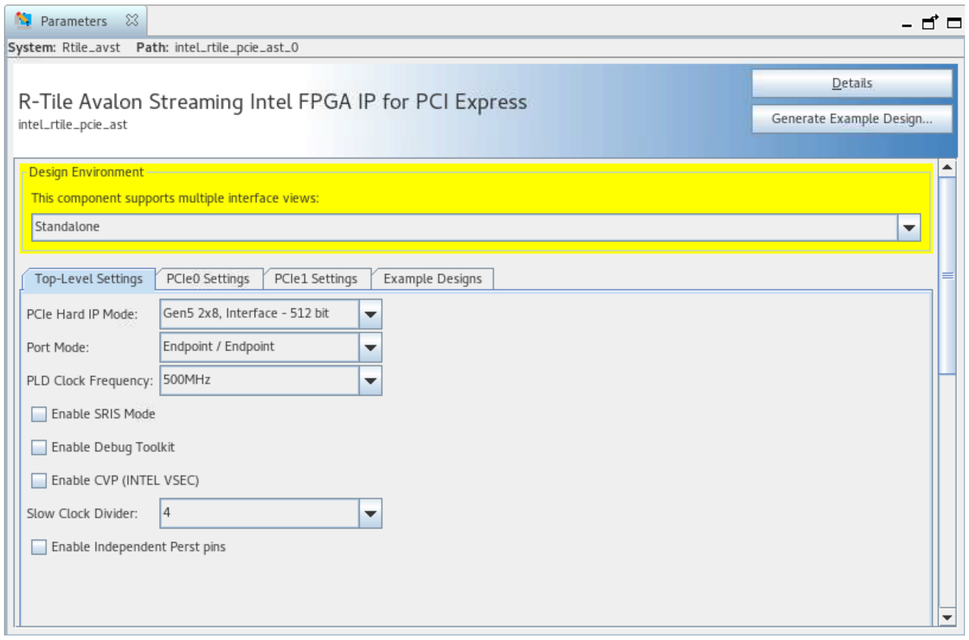
Figure 51. Intel R-Tile Avalon Streaming Top-Level IP Parameter Editor for PCIe Gen5 2x8 Mode


Parameters
System: Rtile_avst Path: intel_rtile_pcie_ast_0
R-Tile Avalon Streaming Intel FPGA IP for PCI Express
intel_rtile_pcie_ast
Details
Generate Example Design...
Design Environment
This component supports multiple interface views:
Top-Level Settings
PCIe0 Settings
PCIe1 Settings
Example Designs

Standalone

PCIe Hard IP Mode: Gen5 2x8, Interface - 512 bit
Port Mode: Endpoint / Endpoint
PLD Clock Frequency: 500MHz
☐ Enable SRIS Mode
☐ Enable Debug Toolkit
☐ Enable Independent GPIO Perst
☐ Enable CVP (INTEL VSEC)
Slow Clock Divider: 4

Figure 52. Intel R-Tile Avalon Streaming Top-Level IP Parameter Editor for PCIe Gen5 2x8 Mode with Support of Independent Perst Pins


Parameters
System: Rtile_avst Path: intel_rtile_pcie_ast_0
R-Tile Avalon Streaming Intel FPGA IP for PCI Express
intel_rtile_pcie_ast
Details
Generate Example Design...
Design Environment
This component supports multiple interface views:
Top-Level Settings
PCIe0 Settings
PCIe1 Settings
Example Designs

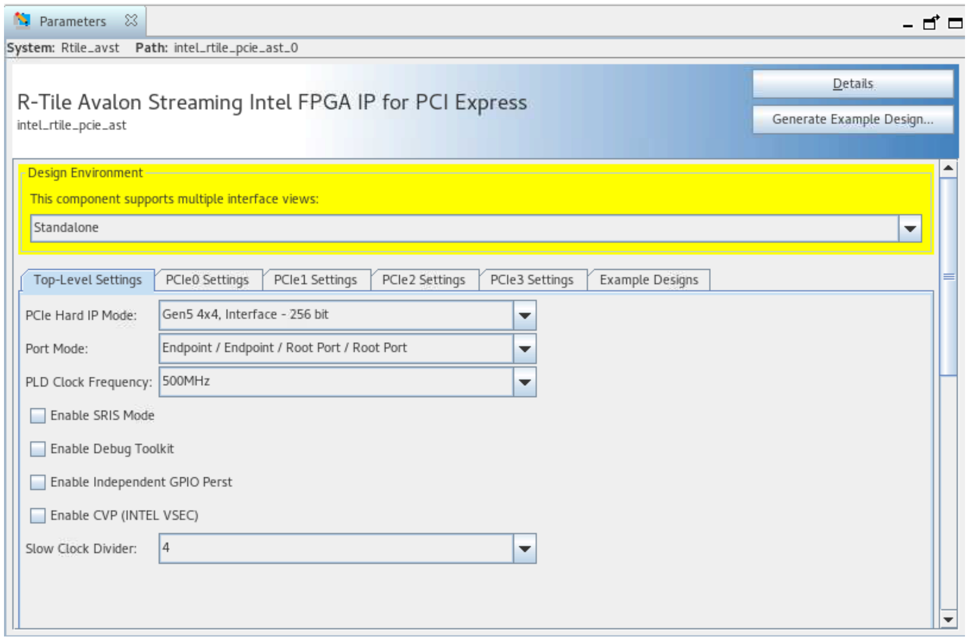
Standalone

PCIe Hard IP Mode: Gen5 2x8, Interface - 512 bit
Port Mode: Endpoint / Endpoint
PLD Clock Frequency: 500MHz
☐ Enable SRIS Mode
☐ Enable Debug Toolkit
☐ Enable CVP (INTEL VSEC)
Slow Clock Divider: 4
☐ Enable Independent Perst pins

Note: For a list of devices that support the **Enable Independent Perst pins** option, refer to Reset.

If you choose a 4x4 mode (Gen3, Gen4 or Gen5), the **PCIe0 Settings**, **PCIe1 Settings**, **PCIe2 Settings** and **PCIe3 Settings** tabs will appear.

Figure 53. Intel R-Tile Avalon Streaming Top-Level IP Parameter Editor for PCIe Gen5 4x4 Mode



System: Rtile_avst Path: intel_rtile_pcie_ast_0

R-Tile Avalon Streaming Intel FPGA IP for PCI Express
intel_rtile_pcie_ast

Design Environment
This component supports multiple interface views:
Standalone

PCIe Hard IP Mode: Gen5 4x4, Interface - 256 bit
Port Mode: Endpoint / Endpoint / Root Port / Root Port
PLD Clock Frequency: 500MHz
Enable SRIS Mode
Enable Debug Toolkit
Enable Independent GPIO Perst
Enable CVP (INTEL VSEC)
Slow Clock Divider: 4

5.2.1. Avalon Parameters

Table 88. Avalon Parameters

Parameter	Value	Default Value	Description
Enable Power Management Interface	**True/False**	**False**	When enabled, the Power Management Interface and Hard IP Status Interface are exported. For more details, refer to section Power Management Interface on page 110.
Enable Legacy Interrupt	**True/False**	**False**	Enable the support for legacy interrupts. For more details, refer to section Legacy Interrupts on page 98.
Enable Completion Timeout Interface	**True/False**	**False**	Enable the Completion Timeout Interface. For more details, refer to section Completion Timeout Interface on page 107.
Enable PRS Event	**True/False**	**False**	Enable the Page Request Service (PRS) Event Interface. For more details, refer to section Page Request Services (PRS) Interface (Endpoint Only). *Note:* This parameter is only available in EP mode.
Enable Error Interface	**True/False**	**False**	Enable the Error Interface. For more details, refer to section Error Interface on page 105.

continued...

Parameter	Value	Default Value	Description
PCIe Header Format	**True/False**	**False**	When this parameter is enabled, the header format is the P-tile header format, else it is the Arria 10 header format.
Enable Configuration Intercept Interface	**True/False**	**False**	Enable the Configuration Intercept Interface. For more details, refer to section Configuration Intercept Interface on page 108. *Note:* This parameter is only available in EP mode.
Power Management State	**True/False**	**False**	When this parameter is enabled and there is a transition to D3cold, the link will transition to L3. When this parameter is disabled and there is transition to D3cold, the link will transition to L2.
Enable Hard IP Reconfiguration Interface	**True/False**	**False**	When selected, this parameter creates an Avalon-MM interface that the Application logic can use to access the internal registers of the Hard IP.
Enable PHY Reconfiguration Interface	**True/False**	**False**	When selected, this parameter enables the PHY Reconfiguration interface.
Enable Parity Ports on Avalon-ST Interface	**True/False**	**False**	When this parameter is enabled, the parity ports appear on the block symbol. These parity ports include: `pX_rx_stN_data_par_o`, `pX_rx_stN_hdr_par_o`, `pX_rx_stN_prefix_par_o`, `pX_tx_stN_data_par_i`, `pX_tx_stN_hdr_par_i`, and `pX_tx_stN_prefix_par_i` ports. When this parameter is enabled, the application layer must provide valid parity in the Avalon-ST TX direction.

5.2.2. Base Address Registers

Table 89. BAR Registers

Parameter	Value	Description
BAR0 Type	**Disabled** **64-bit prefetchable memory** **64-bit non-prefetchable memory** **32-bit non-prefetchable memory**	If you select 64-bit prefetchable memory, 2 contiguous BARs are combined to form a 64-bit prefetchable BAR; you must set the higher numbered BAR to **Disabled**. Defining memory as prefetchable allows contiguous data to be fetched ahead. Prefetching memory is advantageous when the requestor may require more data from the same region than was originally requested. If you specify that a memory is prefetchable, it must have the following 2 attributes: • Reads do not have side effects such as changing the value of the data read. • Write merging is allowed.
BAR1 Type	**Disabled** **32-bit non-prefetchable memory**	For a definition of prefetchable memory, refer to the **BAR0 Type** description.

continued...

Parameter	Value	Description
BAR2 Type	**Disabled** **64-bit prefetchable memory** **64-bit non-prefetchable memory** **32-bit non-prefetchable memory**	For a definition of prefetchable memory and a description of what happens when you select the 64-bit prefetchable memory option, refer to the **BAR0 Type** description.
BAR3 Type	**Disabled** **32-bit non-prefetchable memory**	For a definition of prefetchable memory, refer to the **BAR0 Type** description.
BAR4 Type	**Disabled** **64-bit prefetchable memory** **64-bit non-prefetchable memory** **32-bit non-prefetchable memory**	For a definition of prefetchable memory and a description of what happens when you select the 64-bit prefetchable memory option, refer to the **BAR0 Type** description.
BAR5 Type	**Disabled** **32-bit non-prefetchable memory**	For a definition of prefetchable memory, refer to the **BAR0 Type** description.
BARn Size	**128 Bytes - 16 EBytes**	Specifies the size of the address space accessible to BARn when BARn is enabled. n = 0, 1, 2, 3, 4 or 5
Expansion ROM	**Disabled** **4 KBytes - 12 bits** **8 KBytes - 13 bits** **16 KBytes - 14 bits** **32 KBytes - 15 bits** **64 KBytes - 16 bits** **128 KBytes - 17 bits** **256 KBytes - 18 bits** **512 KBytes - 19 bits** **1 MByte - 20 bits** **2 MBytes - 21 bits** **4 MBytes - 22 bits** **8 MBytes - 23 bits** **16 MBytes - 24 bits**	Specifies the size of the expansion ROM from 4 KBytes to 16 MBytes when enabled.

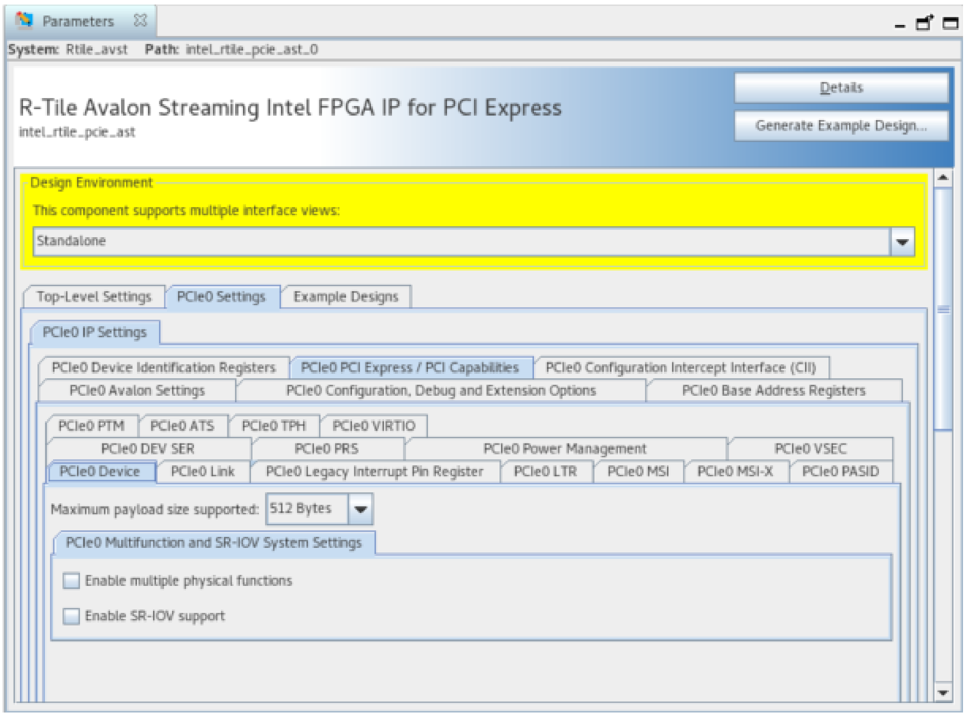
5.2.3. PCI Express and PCI Capabilities Parameters

For each core (PCIe0/PCIe1/PCIe2/PCIe3), the PCI Express / PCI Capabilities tab contains separate tabs for the device, link, device serial number (EP), power management, VSEC, PRS (EP), MSI (EP), MSI-X (EP), PASID (EP), ATS (EP), TPH, PTM (EP), VirtIO (EP), LTR (EP), ACS (RP), slot (RP), and legacy interrupt pin register (EP) parameters.

Note:

- (EP) = the parameter is only available when the IP is in Endpoint mode.
- (RP) = the parameter is only available when the IP is in Root Port mode.
- Otherwise, the parameter is always available.

Figure 54. PCI Express and PCI Capabilities Parameters in Endpoint Mode


Parameters
System: Rtile_avst Path: intel_rtile_pcie_ast_0

R-Tile Avalon Streaming Intel FPGA IP for PCI Express
intel_rtile_pcie_ast

Details
Generate Example Design...

Design Environment
This component supports multiple interface views:
Standalone

Top-Level Settings | PCIe0 Settings | Example Designs

PCIe0 IP Settings

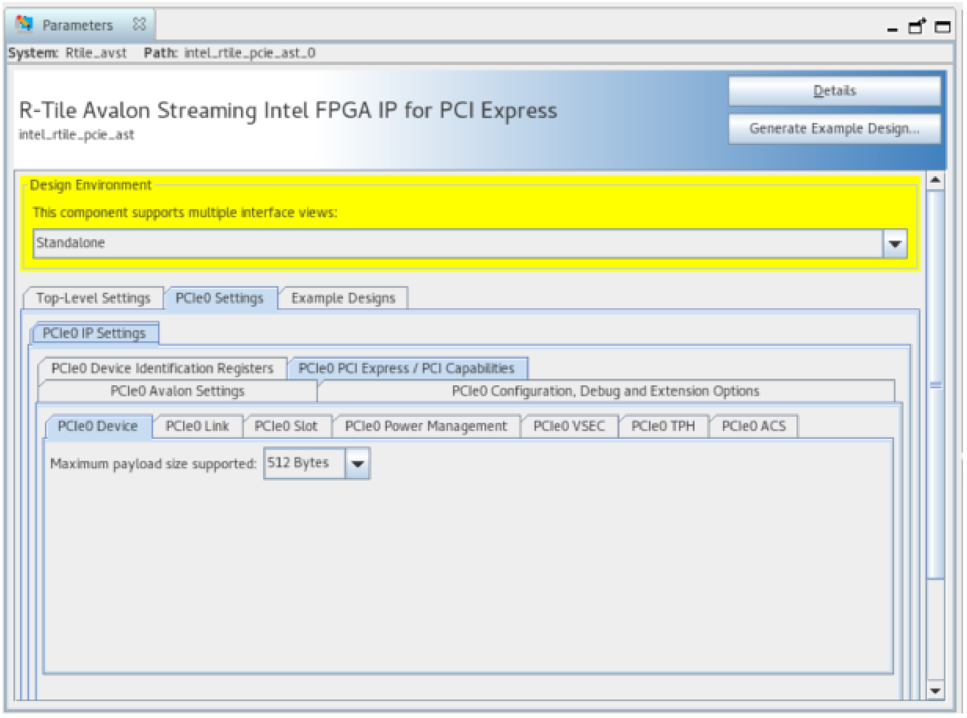
PCIe0 Device Identification Registers | PCIe0 PCI Express / PCI Capabilities | PCIe0 Configuration Intercept Interface (CII)
PCIe0 Avalon Settings | PCIe0 Configuration, Debug and Extension Options | PCIe0 Base Address Registers

PCIe0 PTM | PCIe0 ATS | PCIe0 TPH | PCIe0 VIRTIO
PCIe0 DEV SER | PCIe0 PRS | PCIe0 Power Management | PCIe0 VSEC
PCIe0 Device | PCIe0 Link | PCIe0 Legacy Interrupt Pin Register | PCIe0 LTR | PCIe0 MSI | PCIe0 MSI-X | PCIe0 PASID

Maximum payload size supported: 512 Bytes

PCIe0 Multifunction and SR-IOV System Settings
☐ Enable multiple physical functions
☐ Enable SR-IOV support

Figure 55. PCI Express and PCI Capabilities Parameters in Root Port Mode


Parameters
System: Rtile_avst Path: intel_rtile_pcie_ast_0

R-Tile Avalon Streaming Intel FPGA IP for PCI Express
intel_rtile_pcie_ast

Details
Generate Example Design...

Design Environment
This component supports multiple interface views:
Standalone

Top-Level Settings | PCIe0 Settings | Example Designs

PCIe0 IP Settings

PCIe0 Device Identification Registers | PCIe0 PCI Express / PCI Capabilities
PCIe0 Avalon Settings | PCIe0 Configuration, Debug and Extension Options

PCIe0 Device | PCIe0 Link | PCIe0 Slot | PCIe0 Power Management | PCIe0 VSEC | PCIe0 TPH | PCIe0 ACS

Maximum payload size supported: 512 Bytes

5.2.3.1. Device Capabilities

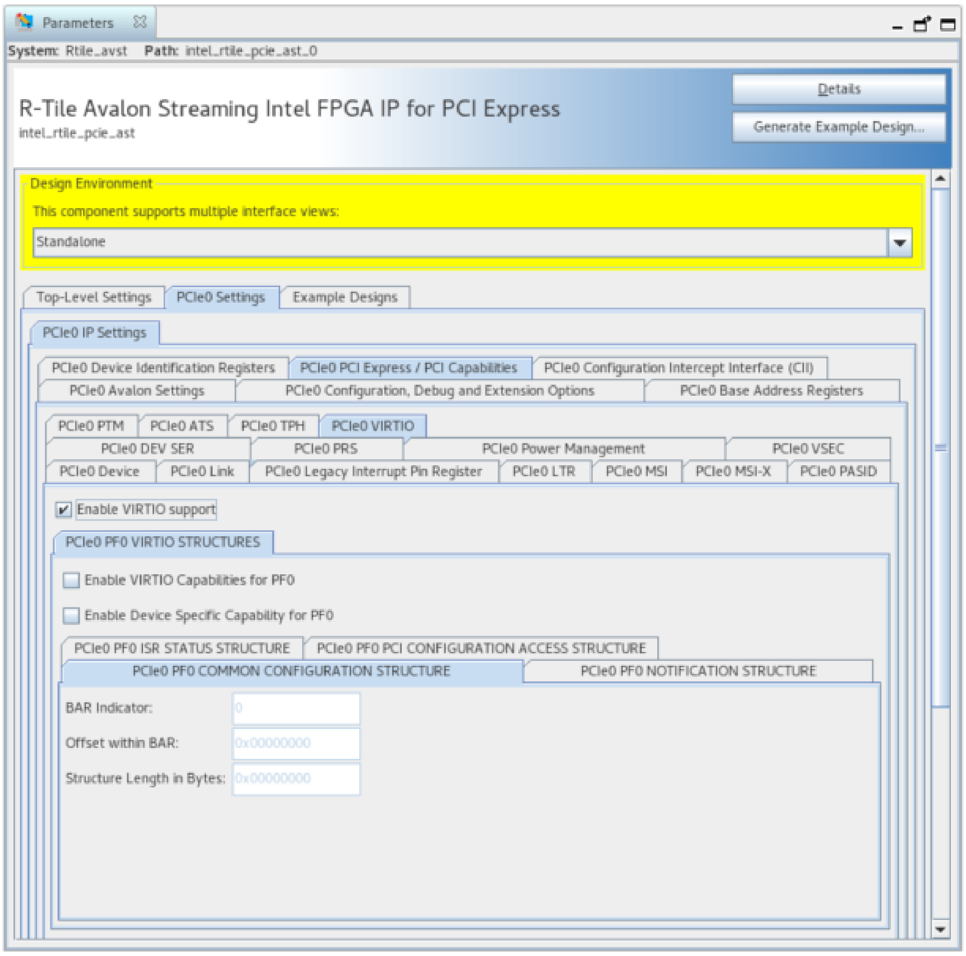
Table 90. Device Capabilities

Parameter	Value	Default Value	Description
Maximum payload sizes supported	**128 bytes** **256 bytes** **512 bytes**	512 bytes	Specifies the maximum payload size supported. This parameter sets the read-only value of the max payload size supported field of the Device Capabilities register.
Enable Multiple Physical Functions	**True/False**	**False**	Enables multiple physical functions.
Enable SR-IOV Support	**True/False**	**False**	Enables support for Single Root I/O Virtualization.

5.2.3.2. VirtIO Parameters

You can enable VirtIO as shown in the screenshot below:

Figure 56. Enable VirtIO Support


Parameters
Details
Generate Example Design...
Top-Level Settings
PCIe0 Settings
Example Designs
PCIe0 IP Settings
PCIe0 Device Identification Registers
PCIe0 PCI Express / PCI Capabilities
PCIe0 Configuration Intercept Interface (CII)
PCIe0 Avalon Settings
PCIe0 Configuration, Debug and Extension Options
PCIe0 Base Address Registers
PCIe0 PTM
PCIe0 ATS
PCIe0 TPH
PCIe0 VIRTIO
PCIe0 DEV SER
PCIe0 PRS
PCIe0 Power Management
PCIe0 VSEC
PCIe0 Device
PCIe0 Link
PCIe0 Legacy Interrupt Pin Register
PCIe0 LTR
PCIe0 MSI
PCIe0 MSI-X
PCIe0 PASID
PCIe0 PF0 VIRTIO STRUCTURES
PCIe0 PF0 ISR STATUS STRUCTURE
PCIe0 PF0 PCI CONFIGURATION ACCESS STRUCTURE
PCIe0 PF0 COMMON CONFIGURATION STRUCTURE
PCIe0 PF0 NOTIFICATION STRUCTURE

System: Rtile_avst **Path:** intel_rtile_pcie_ast_0

R-Tile Avalon Streaming Intel FPGA IP for PCI Express
intel_rtile_pcie_ast

Design Environment
This component supports multiple interface views:
Standalone

[✓] Enable VIRTIO support

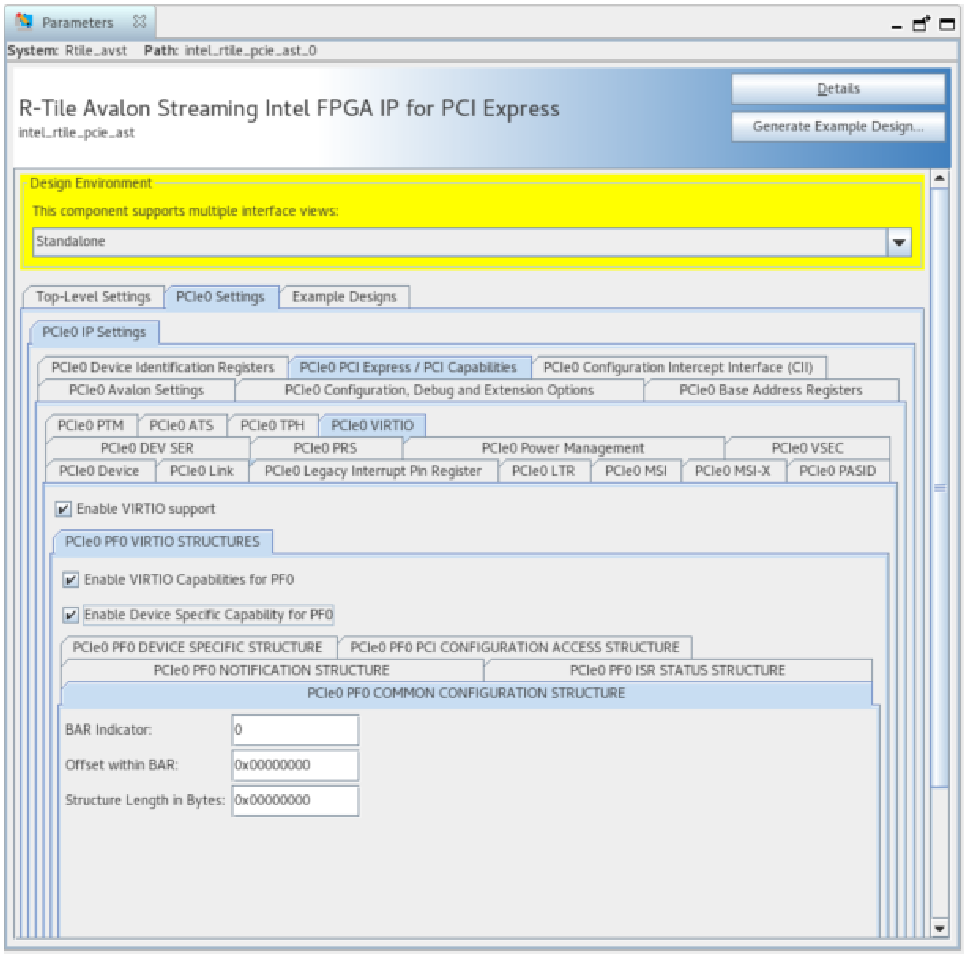
[ ] Enable VIRTIO Capabilities for PF0

[ ] Enable Device Specific Capability for PF0

BAR Indicator: 0
Offset within BAR: 0x00000000
Structure Length in Bytes: 0x00000000

Then, you can configure the appropriate VirtIO capability parameters as shown in the screenshot below:

Figure 57. Configure VirtIO Capability Parameters


Parameters
System: Rtile_avst Path: intel_rtile_pcie_ast_0
R-Tile Avalon Streaming Intel FPGA IP for PCI Express
intel_rtile_pcie_ast
Details
Generate Example Design...
Design Environment
This component supports multiple interface views:
Standalone
Top-Level Settings
PCIe0 Settings
Example Designs
PCIe0 IP Settings
PCIe0 Device Identification Registers
PCIe0 PCI Express / PCI Capabilities
PCIe0 Configuration Intercept Interface (CII)
PCIe0 Avalon Settings
PCIe0 Configuration, Debug and Extension Options
PCIe0 Base Address Registers
PCIe0 PTM
PCIe0 ATS
PCIe0 TPH
PCIe0 VIRTIO
PCIe0 DEV SER
PCIe0 PRS
PCIe0 Power Management
PCIe0 VSEC
PCIe0 Device
PCIe0 Link
PCIe0 Legacy Interrupt Pin Register
PCIe0 LTR
PCIe0 MSI
PCIe0 MSI-X
PCIe0 PASID
PCIe0 PF0 VIRTIO STRUCTURES
PCIe0 PF0 DEVICE SPECIFIC STRUCTURE
PCIe0 PF0 PCI CONFIGURATION ACCESS STRUCTURE
PCIe0 PF0 NOTIFICATION STRUCTURE
PCIe0 PF0 ISR STATUS STRUCTURE
PCIe0 PF0 COMMON CONFIGURATION STRUCTURE

☑ Enable VIRTIO support

☑ Enable VIRTIO Capabilities for PF0

☑ Enable Device Specific Capability for PF0

BAR Indicator: 0

Offset within BAR: 0x00000000

Structure Length in Bytes: 0x00000000

The next table summarizes the parameters associated with the five VirtIO device configuration structures:

Table 91. VirtIO Structure PCI Capabilities Parameters

Parameter	Description	Allowed Range	Default Value
PF/VF VirtIO Common Configuration Structure Capability Parameters			
PFs 0-7 Common Configuration Structure BAR Indicator	Indicates BAR holding the Common Configuration Structure of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-5	0
PFs 0-7 VFs Common Configuration Structure BAR Indicator	Indicates BAR holding the Common Configuration Structure of VFs associated	0-5	0

continued...

Parameter	Description	Allowed Range	Default Value
	with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.		
PFs 0-7 Common Configuration Structure Offset within BAR	Indicates starting position of Common Config Structure in a given BAR of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-536870911	0
PFs 0-7 VFs Common Configuration Structure BAR Indicator	Indicates starting position of Common Config Structure in a given BAR of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-536870911	0
PFs 0-7 Common Configuration Structure Length	Indicates length in bytes of Common Config Structure of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-536870911	0
PFs 0-7 VFs Common Configuration Structure Length	Indicates length in bytes of Common Config Structure of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-536870911	0
PF/VF VirtIO Notifications Structure Capability Parameters			
PFs 0-7 Notifications Structure BAR Indicator	Indicates BAR holding the Notifications Structure of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-5	0
PFs 0-7 VFs Notifications Structure BAR Indicator	Indicates BAR holding the Notifications Structure of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-5	0
PFs 0-7 Notifications Structure Offset within BAR	Indicates starting position of Notifications Structure in given BAR of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-536870911	0
PFs 0-7 VFs Notifications Structure BAR Indicator	Indicates starting position of Notifications Structure in given BAR of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-536870911	0
PFs 0-7 Notifications Structure Length	Indicates length in bytes of Notifications Structure of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-536870911	0

continued...

Parameter	Description	Allowed Range	Default Value
PFs 0-7 VFs Notifications Structure Length	Indicates length in bytes of Notifications Structure of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-536870911	0
PFs 0-7 Notifications Structure Notify Off Multiplier	Indicates multiplier for queue_notify_off in Notifications Structure of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-536870911	0
PFs 0-7 VFs Notifications Structure Notify Off Multiplier	Indicates multiplier for queue_notify_off in Notifications Structure of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-536870911	0
PF/VF VirtIO ISR Status Structure Capability Parameters			
PFs 0-7 ISR Status Structure BAR Indicator	Indicates BAR holding the ISR Status Structure of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-5	0
PFs 0-7 VFs ISR Status Structure BAR Indicator	Indicates BAR holding the ISR Status Structure of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-5	0
PFs 0-7 ISR Status Structure Offset within BAR	Indicates starting position of ISR Status Structure in given BAR of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-536870911	0
PFs 0-7 VFs ISR Status Structure BAR Indicator	Indicates starting position of ISR Status Structure in given BAR of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-536870911	0
PFs 0-7 ISR Status Structure Length	Indicates length in bytes of ISR Status Structure of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-536870911	0
PFs 0-7 VFs ISR Status Structure Length	Indicates length in bytes of ISR Status Structure of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-536870911	0
PF/VF VirtIO Device-Specific Configuration Structure Capability Parameters			
Enable PFs 0-7 VirtIO Device Specific Capability	Enable PFs 0-7 VirtIO Device-Specific Configuration Structure	True / False	False

continued...

Parameter	Description	Allowed Range	Default Value
	Capability. This parameter is located under the **PFn VirtIO Structures** tab.		
Enable PFs 0-7 VFs VirtIO Device-Specific Capability	Enable VirtIO Device-Specific Configuration Structure Capability of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	True / False	False
PFs 0-7 Device-Specific Configuration Structure BAR Indicator	Indicates BAR holding the Device-Specific Configuration Structure of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-5	0
PFs 0-7 VFs Device-Specific Configuration Structure BAR Indicator	Indicates BAR holding the Device-Specific Configuration Structure of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-5	0
PFs 0-7 Device-Specific Configuration Structure Offset within BAR	Indicates starting position of Device-Specific Configuration Structure in given BAR of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-536870911	0
PFs 0-7 VFs Device-Specific Configuration Structure BAR Indicator	Indicates starting position of Device-Specific Configuration Structure in given BAR of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-536870911	0
PFs 0-7 Device-Specific Configuration Structure Length	Indicates length in bytes of Device-Specific Configuration Structure of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-536870911	0
PFs 0-7 VFs Device-Specific Configuration Structure Length	Indicates length in bytes of Device-Specific Configuration Structure of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-536870911	0
PF/VF VirtIO PCI Configuration Access Structure Capability Parameters			
PFs 0-7 PCI Configuration Access Structure BAR Indicator	Indicates BAR holding the PCI Configuration Access Structure of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-5	0

continued...

Parameter	Description	Allowed Range	Default Value
PFs 0-7 VFs PCI Configuration Access Structure BAR Indicator	Indicates BAR holding the PCI Configuration Access Structure of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-5	0
PFs 0-7 PCI Configuration Access Structure Offset within BAR	Indicates Starting position of PCI Configuration Access Structure in given BAR of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-536870911	0
PFs 0-7 VFs PCI Configuration Access Structure BAR Indicator	Indicates Starting position of PCI Configuration Access Structure in given BAR of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-536870911	0
PFs 0-7 PCI Configuration Access Structure Length	Indicates length in bytes of PCI Configuration Access Structure of PFs 0-7. This parameter is located under the **PFn VirtIO Structures** tab.	0-536870911	0
PFs 0-7 VFs PCI Configuration Access Structure Length	Indicates length in bytes of PCI Configuration Access Structure of VFs associated with PFs 0-7. This parameter is located under the **PFn VFs VirtIO Structures** tab.	0-536870911	0

5.2.3.3. Link Capabilities

Table 92. Link Capabilities

Parameter	Value	Default Value	Description
Link port number	0 - 255	1	Sets the read-only value of the port number field in the `Link Capabilities` register.
Slot clock configuration	**True/False**	**True**	When this parameter is **True**, it indicates that the Endpoint uses the same physical reference clock that the system provides on the connector. When it is **False**, the IP core uses an independent clock regardless of the presence of a reference clock on the connector. This parameter sets the Slot Clock Configuration bit (bit 12) in the `PCI Express Link Status` register. You cannot enable this option when the **Enable SRIS Mode** option is enabled.

5.2.3.4. Legacy Interrupt Pin Register

Table 93. Legacy Interrupts Parameters

These parameters are only available in Endpoint mode.

Parameter	Value	Default Value	Description
Set Interrupt Pin for PFn	**NO INT** **INTA** **INTA/INTB/INTC/INTD**	**NO INT**	When Legacy Interrupts are not enabled, the only option available is **NO INT**. When Legacy Interrupts are enabled and multifunction is disabled, the only option available is **INTA**. When Legacy Interrupts are enabled and multifunction is enabled (total PFs > 1), the options available are **INTA**, **INTB**, **INTC** and **INTD**.

5.2.3.5. MSI Capabilities

Table 94. MSI Capabilities

Parameter	Value	Default Value	Description
PF0 Enable MSI	**True/False**	**False**	Enables MSI functionality for PF0. If this parameter is **True**, the **Number of MSI messages requested** parameter will appear allowing you to set the number of MSI messages.
PF0 MSI 64-bit Addressing	**True/False**	**False**	Enables or disables MSI 64-bit addressing for PF0.
PF0 MSI Extended Data Capable	**True/False**	**False**	Enables or disables MSI extended data capability for PF0.
PF0 Number of MSI messages requested	**1** **2** **4** **8** **16** **32**	**1**	Sets the number of messages that the application can request in the multiple message capable field of the `Message Control` register.

5.2.3.6. MSI-X Capabilities

Table 95. MSI-X Capabilities

Parameter	Value	Default Value	Description
Enable MSI-X	**True/False**	**False**	Enables the MSI-X functionality.
Table Size	0x0 - 0x7FF (only values of powers of two minus 1 are valid)	0	System software reads this field to determine the MSI-X table size <**n**>, which is encoded as <**n-1**>.

continued...

Send Feedback

Parameter	Value	Default Value	Description
			For example, a returned value of 2047 indicates a table size of 2048. This field is read-only. Address offset: 0x068[26:16]
Table Offset	0x0 - 0xFFFFFFFF	0	Points to the base of the MSI-X table. The lower 3 bits of the table BAR indicator (BIR) are set to zero by software to form a 64-bit qword-aligned offset. This field is read-only after being programmed.
Table BAR indicator	0x0 - 0x5	0	Specifies which one of a function's BARs, located beginning at 0x10 in Configuration Space, is used to map the MSI-X table into memory space. This field is read-only after being programmed.
Pending bit array (PBA) offset	0x0 - 0xFFFFFFFF	0	Used as an offset from the address contained in one of the function's Base Address registers to point to the base of the MSI-X PBA. The lower 3 bits of the PBA BIR are set to zero by software to form a 32-bit qword-aligned offset. This field is read-only after being programmed.
PBA BAR indicator	0x0 - 0x5	0	Specifies the function's Base Address register, located beginning at 0x10 in Configuration Space, that maps the MSI-X PBA into memory space. This field is read-only after being programmed.
VF Table size	0x0 - 0x7FF (only values of powers of two minus 1 are valid)	0	Sets the number of entries in the MSI-X table for VFs. MSI-X cannot be disabled for VFs. Set to 1 to save resources.

5.2.3.7. Slot Capabilities

Note: This tab is visible in the Parameter Editor only if the **Port Mode** parameter in the **Top-Level Settings** tab is set to **Root Port**.

Table 96. Slot Capabilities

Parameter	Value	Default Value	Description
Use Slot register	**True/False**	**False**	This parameter is only supported in Root Port mode. The slot capability is required for Root Ports if a slot is implemented on the

continued...

Parameter	Value	Default Value	Description
			port. Slot status is recorded in the `PCI Express Capabilities` register.
Slot power scale	0 - 3	0	Specifies the scale used for the slot power limit. The following coefficients are defined: • 0 = 1.0x • 1 = 0.1x • 2 = 0.01x • 3 = 0.001x The default value prior to hardware and firmware initialization is b'00. Writes to this register also cause the port to send the `Set_Slot_Power_Limit` message.
Slot power limit	0 - 255	0	In combination with the **Slot power scale** value, specifies the upper limit in watts for the power supplied by the slot.
Slot number	0 - 8191	0	Specifies the slot number.

5.2.3.8. Latency Tolerance Reporting (LTR)

This capability allows the R-Tile Avalon streaming IP, when operating in Endpoint mode, to report the delay that it can tolerate when requesting service from the Host. This information can help software optimize performance when the Endpoint needs a fast response, or optimize system power when a fast response is not necessary.

Table 97. Latency Tolerance Reporting (LTR) Parameters

Parameter	Value	Default Value	Description
PCIeN Enable LTR	**True/False**	**False**	Enable or disable LTR capability for PCIeN. LTR is a new mechanism that enables Endpoints to send information about their latency requirements for memory read/writes and interrupts.

5.2.3.9. Process Address Space ID (PASID)

PASID is an optional feature that enables sharing of a single Endpoint device across multiple processes while providing each process a complete 64-bit virtual address space. In practice, this feature adds support for a TLP prefix that contains a 20-bit address space that can be added to memory transaction TLPs.

Table 98. Process Address Space ID (PASID) Parameters

Parameter	Value	Default Value	Description
PCIeN PF0 Enable PASID	**True/False**	**False**	Enable or disable PASID capability for PCIeN PF0.
PCIeN PF0 Enable Execute Permission Support	**True/False**	**False**	Enable or disable PASID Execute Permission Support for PCIeN PF0.
PCIeN PF0 Enable Privileged Mode Support	**True/False**	**False**	Enable or disable PASID Privileged Mode Support for PCIeN PF0.
PCIeN PF0 Max PASID Width	**0 - 20**	**0**	Set the Max PASID Width for PCIeN PF0.

5.2.3.10. Device Serial Number Capability

The PCIe device serial number capability is an optional extended capability that can be implemented by any PCIe device. The device serial number is a 64-bit value that is unique for a given PCIe device.

Table 99. Device Serial Number Capability

Parameter	Value	Default Value	Description
Enable Device Serial Number Capability	**True/False**	**False**	Enables the device serial number capability. This is an optional extended capability that provides a unique identifier for the PCIe device.
Device Serial Number (DW1)	32 bits	0x0000000000000000	Sets the lower 32 bits of the IEEE 64-bit Device Serial Number (DW1).
Device Serial Number (DW2)	32 bits	0x0000000000000000	Sets the upper 32 bits of the IEEE 64-bit Device Serial Number (DW2).

5.2.3.11. Page Request Service (PRS)

Note: This capability is available in EP mode only.

Note: Only Port 0 and Port 1 support PRS.

Table 100. Page Request Service (PRS) Capability

Parameter	Value	Default Value	Description
Enable PRS	**True/False**	**False**	Enable or disable Page Request Service (PRS) capability.

5.2.3.12. Access Control Service (ACS)

Note: This feature is available if the IP is in Root Port mode, or if the IP is in Endpoint mode and the **Enable multiple physical functions** in the **PCIeN Device** tab is set to **True**.

Note: ACS support for Ports 2 and 3 in Root Port mode is only available in the following OPNs:

- AGIx027R29AxxxxR2
- AGIx027R29AxxxxR3
- AGIx027R29BxxxxR3
- AGIx023R18AxxxxR0
- AGIx041R29DxxxxR0
- AGIx041R29DxxxxR1

For additional details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

Table 101. ACS Capabilities for Physical Functions

Parameter	Value	Default Value	Description
Enable Access Control Service (ACS)	**True/False**	**False**	ACS defines a set of control points within a PCI Express topology to determine whether a TLP is to be routed normally, blocked, or redirected.
Enable ACS P2P Traffic Support	**True/False**	**False**	Indicates if the component supports Peer to Peer Traffic.
Enable ACS P2P Egress Control	**True/False**	**False**	Indicates if the component implements ACS P2P Egress Control. This parameter is visible only if **Enable ACS P2P Traffic Support** is set to **True**.
Enable ACS P2P Egress Control Vector Size	0 - 255	0	Indicates the number of bits in the ACS P2P Egress Control Vector. This parameter is visible only if **Enable ACS P2P Egress Control** is set to **True**.

Table 102. ACS Capabilities for Virtual Functions

Parameter	Value	Default Value	Description
Enable Access Control Service (ACS)	**True/False**	**False**	ACS defines a set of control points within a PCI Express topology to determine whether a TLP is to be routed normally, blocked, or redirected.

5.2.3.13. Power Management

Table 103. Power Management

Parameter	Value	Default Value	Description
Endpoint L0s acceptable latency	Maximum of 64 ns Maximum of 128 ns Maximum of 256 ns Maximum of 512 ns Maximum of 1 us Maximum of 2 us Maximum of 4 us No limit	Maximum of 64 ns	This design parameter specifies the maximum acceptable latency that the application layer can tolerate for any link between the device and the root complex to exit the L0s state. It sets the read-only value of the Endpoint L0s acceptable latency field of the `Device Capabilities Register` (0x084). This Endpoint does not support the L0s or L1 states. However, in a switched system, there may be links connected to switches that have L0s and L1 enabled. This parameter is set to allow system configuration software to read the acceptable latencies for all devices in the system and the exit latency for each link to determine which links can enable Active State Power Management (ASPM). This setting is disabled for Root Ports. The default value of this parameter is 64 ns. This is the safest setting for most designs.
Endpoint L1 acceptable latency	Maximum of 1 us Maximum of 2 us Maximum of 4 us Maximum of 8 us Maximum of 16 us Maximum of 32 us Maximum of 64 us No limit	Maximum of 1 us	This value indicates the acceptable latency that an Endpoint can withstand in the transition from the L1 state to L0 state. It is an indirect measure of the Endpoint's internal buffering. It sets the read-only value of the Endpoint L1 acceptable latency field of the `Device Capabilities Register`. This Endpoint does not support the L0s or L1 states. However, a switched system may include links connected to switches that have L0s and L1 enabled. This parameter is set to allow system configuration software to read the acceptable latencies for all devices in the system and the exit latency for each link to determine which links can enable Active State Power Management (ASPM). This setting is disabled for Root Ports.

5.2.3.14. Vendor Specific Extended Capability (VSEC) Registers

Table 104. VSEC Register

Parameter	Value	Default Value	Description
Vendor Specific Extended Capability	0/1	0	Enables the Vendor Specific Extended Capability (VSEC).
User ID register from the Vendor Specific Extended Capability	0 - 65534	0	Sets the read-only value of the 16-bit User ID register from the Vendor Specific Extended Capability.
Drops Vendor Type0 Messages	0/1	0	When this parameter is set to **1**, the IP core drops vendor Type 0 messages while treating them as Unsupported Requests (UR). When it is set to **0**, the IP core passes these messages on to the user logic. This option is not applicable to TLP Bypass mode. In TLP Bypass mode, the received Vendor Message Type 0 will always be visible on the RX AVST interface.
Drops Vendor Type1 Messages	0/1	0	When this parameter is set to **1**, the IP core silently drops vendor Type 1 messages. When it is set to **0**, the IP core passes these messages on to the user logic. This option is not applicable to TLP Bypass mode. In TLP Bypass mode, the received Vendor Message Type 1 will always be visible on the RX AVST interface.

5.2.3.15. TLP Processing Hints (TPH)

Table 105. TPH Capabilities

Parameter	Value	Default Value	Description
Enable TLP Processing Hints (TPH)	**True/False**	**False**	Enable or disable TLP Processing Hints (TPH) capability. Using TPH may improve the latency performance and reduce traffic congestion.

5.2.3.16. Address Translation Services (ATS) Capabilities

Table 106. ATS Capabilities

Parameter	Value	Default Value	Description
Enable Address Translation Services (ATS)	**True/False**	**False**	Enable or disable Address Translation Services (ATS) capability.

Parameter	Value	Default Value	Description
			When ATS is enabled, senders can request and cache translated addresses using the RP memory space for later use.

5.2.3.17. Precision Time Measurement (PTM)

Note: Only Port 0 and Port 1 support PTM.

Table 107. PTM Parameters

These parameters are only available in Endpoint mode.

Parameter	Value	Default Value	Description
Enable PTM	**True/False**	**False**	Enables Precision Time Measurement (PTM) for PCIe0.
Period between each automatic update of PTM context (in ms)	**Disable** **1** **2** **3** **4** **5** **6** **7** **8** **9** **10**	**Disable**	Determines the PTM context auto-update interval. Selecting **Disable** prevents PTM context auto-update.

6. Troubleshooting/Debugging

As you bring up your PCI Express system, you may face issues related to FPGA configuration, link training, BIOS enumeration, data transfer, and so on. This chapter suggests some strategies to resolve the common issues that occur during bring-up. You can additionally use the R-Tile Debug Toolkit to identify some of these issues.

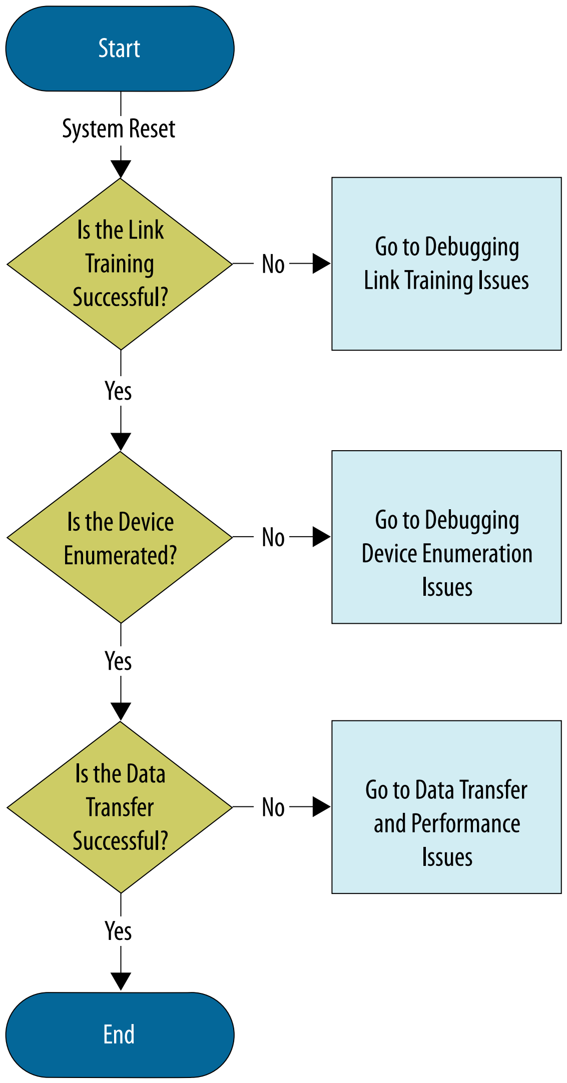
6.1. Hardware Debug

Typically, PCI Express link-up involves the following phases:

1. Link training
2. BIOS enumeration
3. Data transfer

The following sections describe the flow to debug issues during the hardware bring-up. Intel recommends a systematic approach to diagnosing issues as illustrated in the following figure.

Figure 58. PCI Express Debug Flow Chart


Start
System Reset
Is the Link Training Successful?
No
Go to Debugging Link Training Issues
Yes
Is the Device Enumerated?
No
Go to Debugging Device Enumeration Issues
Yes
Is the Data Transfer Successful?
No
Go to Data Transfer and Performance Issues
Yes
End

6.1.1. Debugging Link Training Issues

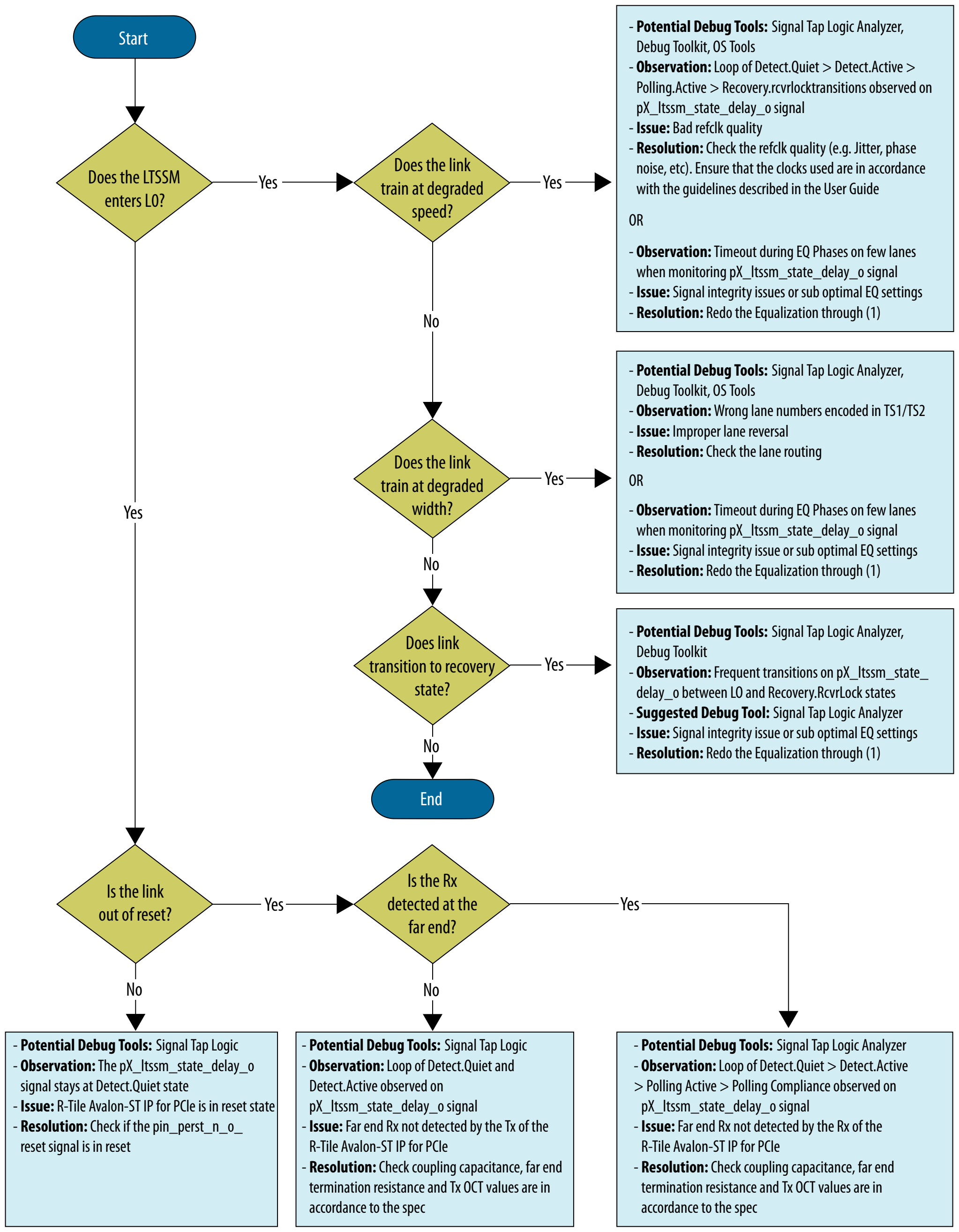
The Physical Layer automatically performs link training and initialization without software intervention. This is a well-defined process to configure and initialize the device's Physical Layer and link so that PCIe packets can be transmitted.

Some examples of link training issues may include:

- Link fails to negotiate to the expected link speed.
- Link fails to negotiate to the expected link width.
- LTSSM fails to reach/stay stable at L0.

Use the flow chart below to identify the potential cause of a link training problem when using the R-Tile Avalon-ST IP for PCI Express.

Figure 59. Link Training Debug Flow


Start
Does the LTSSM enters L0?
Yes
Does the link train at degraded speed?
Yes
No
Does the link train at degraded width?
Yes
No
Does link transition to recovery state?
Yes
No
End
Yes
Is the link out of reset?
Yes
Is the Rx detected at the far end?
Yes
No
No

- **Potential Debug Tools:** Signal Tap Logic Analyzer, Debug Toolkit, OS Tools
- **Observation:** Loop of Detect.Quiet > Detect.Active > Polling.Active > Recovery.rcvrlocktransitions observed on pX_ltssm_state_delay_o signal
- **Issue:** Bad refclk quality
- **Resolution:** Check the refclk quality (e.g. Jitter, phase noise, etc). Ensure that the clocks used are in accordance with the guidelines described in the User Guide

OR

- **Observation:** Timeout during EQ Phases on few lanes when monitoring pX_ltssm_state_delay_o signal
- **Issue:** Signal integrity issues or sub optimal EQ settings
- **Resolution:** Redo the Equalization through (1)

- **Potential Debug Tools:** Signal Tap Logic Analyzer, Debug Toolkit, OS Tools
- **Observation:** Wrong lane numbers encoded in TS1/TS2
- **Issue:** Improper lane reversal
- **Resolution:** Check the lane routing

OR

- **Observation:** Timeout during EQ Phases on few lanes when monitoring pX_ltssm_state_delay_o signal
- **Issue:** Signal integrity issue or sub optimal EQ settings
- **Resolution:** Redo the Equalization through (1)

- **Potential Debug Tools:** Signal Tap Logic Analyzer, Debug Toolkit
- **Observation:** Frequent transitions on pX_ltssm_state_delay_o between L0 and Recovery.RcvrLock states
- **Suggested Debug Tool:** Signal Tap Logic Analyzer
- **Issue:** Signal integrity issue or sub optimal EQ settings
- **Resolution:** Redo the Equalization through (1)

- **Potential Debug Tools:** Signal Tap Logic
- **Observation:** The pX_ltssm_state_delay_o signal stays at Detect.Quiet state
- **Issue:** R-Tile Avalon-ST IP for PCIe is in reset state
- **Resolution:** Check if the pin_perst_n_o_ reset signal is in reset

- **Potential Debug Tools:** Signal Tap Logic
- **Observation:** Loop of Detect.Quiet and Detect.Active observed on pX_ltssm_state_delay_o signal
- **Issue:** Far end Rx not detected by the Tx of the R-Tile Avalon-ST IP for PCIe
- **Resolution:** Check coupling capacitance, far end termination resistance and Tx OCT values are in accordance to the spec

- **Potential Debug Tools:** Signal Tap Logic Analyzer
- **Observation:** Loop of Detect.Quiet > Detect.Active > Polling Active > Polling Compliance observed on pX_ltssm_state_delay_o signal
- **Issue:** Far end Rx not detected by the Rx of the R-Tile Avalon-ST IP for PCIe
- **Resolution:** Check coupling capacitance, far end termination resistance and Tx OCT values are in accordance to the spec

Note: To reinitiate the equalization procedure, write 1b to the Perform Equalization bit [0] in the Link Control 3 register.

6.1.2. Debugging Device Enumeration Issues

Device enumeration is performed at the system level for all the PCIe devices included in the system. The enumeration algorithm is in charge of performing the correct exploration of all the potential PCIe devices that may be included. You can refer to Root Port Enumeration for an example of how this algorithm can be implemented by a Root Port.

During enumeration, the majority of TLPs being transmitted at the link level are Configuration TLPs. In general, if the Link training was performed correctly, the R-Tile Avalon-ST IP responds to any Configuration TLP received. However, if link instability is present, some of these Configuration TLPs may not be received correctly and hence cause enumeration problems. Refer to the section Debugging Link Training Issues to ensure your link is operating correctly. In addition, you can use the Debug Toolkit event counters, described in Event Counters, to ensure Configuration TLPs are being transferred at the link level.

6.1.3. Debugging Data Transfer and Performance Issues

There are many possible reasons causing the PCIe link to stop transmitting data. The PCI Express Base Specification defines three types of errors, outlined in the table below:

Table 108. Error Types Defined by the PCI Express Base Specification

Type	Responsible Agent	Description
Correctable	Hardware	While correctable errors may affect system performance, data integrity is maintained.
Uncorrectable, non-fatal	Device software	Uncorrectable, non-fatal errors are defined as errors in which data is lost, but system integrity is maintained. For example, the fabric may lose a particular TLP, but it still works without problems.
Uncorrectable, fatal	System software	Errors generated by a loss of data and system failure are considered uncorrectable and fatal. Software must determine how to handle such errors: whether to reset the link or implement other means to minimize the problem.

Table 109. Correctable Error Status Register (AER)

Observation	Issue	Resolution
Receiver error bit set	Physical layer error which may be due to a MAC error when a lane is in L0, or a Control symbol being received in the wrong lane, or signal Integrity issues where the link may transition from L0 to the Recovery state.	Use the configuration output interface, or the Hard IP reconfiguration interface and the flow chart in Link Training Debugging Flow to obtain more information about the error.
Bad DLLP bit set	Data link layer error which may occur when a CRC verification fails.	Use the configuration output interface or the Hard IP reconfiguration interface to obtain more information about the error.

continued...

Observation	Issue	Resolution
Bad TLP bit set	Data link layer error which may occur when an LCRC verification fails or when a sequence number error occurs.	Use the configuration output interface or the Hard IP reconfiguration interface to obtain more information about the error.
Replay_num_rollover bit set	Data link layer error which may be due to TLPs sent without success (no ACK) four times in a row.	Use the configuration output interface or the Hard IP reconfiguration interface to obtain more information about the error.
replay timer timeout status bit set	Data link layer error which may occur when no ACK or NAK was received within the timeout period for the TLPs transmitted.	Use the configuration output interface or the Hard IP reconfiguration interface to obtain more information about the error.
Advisory non-fatal	Transaction layer error which may be due to higher priority uncorrectable error detected.	
Corrected internal error bits set	Transaction layer error which may be due to an ECC error in the internal Hard IP RAM.	Use the error interface, configuration output interface, or the Hard IP reconfiguration interface and DBI registers to obtain more information about the error.

Table 110. Uncorrectable Error Status Register (AER)

Observation	Issue	Resolution
Data link protocol error	Data link layer error which may be due to transmitter receiving an ACK/NAK whose Seq ID does not correspond to an unacknowledged TLP or ACK sequence number.	Use the configuration output interface, Hard IP reconfiguration interface to obtain more information about the error.
Surprise down error	Data link layer error which may be due to `link_up_o` getting deasserted during L0, indicating the physical layer link is going down unexpectedly.	Use the error interface, configuration output interface, Hard IP reconfiguration interface and DBI registers to obtain more information about the error.
Flow control protocol error	Transaction layer error which can be due to the receiver reporting more than the allowed credit limit. This error occurs when a component does not receive updated flow control credits with the 200 μs limit.	Use the TX/RX flow control interface, configuration output interface, Hard IP reconfiguration interface to obtain more information about the error.
Poisoned TLP received	Transaction layer error which can be due to a received TLP with the EP bit set.	Use the error interface, configuration output interface, configuration intercept interface, Hard IP reconfiguration interface to obtain more information on the error and determine the appropriate action.
Completion timeout	Transaction layer error which can be due to a completion not received within the required amount of time after a non-posted request was sent.	Use the error interface, completion timeout interface, configuration output interface, Hard IP reconfiguration interface to obtain more information on the error.
Completer abort	Transaction layer error which can be due to a completer being unable to fulfill a request due to a problem with the requester or a failure of the completer.	Use the configuration output interface, error interface, Hard IP reconfiguration interface to obtain more information on the error.

continued...

Observation	Issue	Resolution
Unexpected completion	Transaction layer error which can be due to a requester receiving a completion that doesn't match any request awaiting a completion. The TLP is deleted by the Hard IP and not presented to the Application Layer.	Use the configuration output interface, error interface, Hard IP reconfiguration interface to obtain more information on the error.
Receiver overflow	Transaction layer error which can be due to a receiver receiving more TLPs than the available receive buffer space. The TLP is deleted by the Hard IP and not presented to the Application Layer.	Use the TX/RX flow control interface, error interface, configuration output interface, Hard IP reconfiguration interface to obtain more information on the error.
Malformed TLP	Transaction layer error which can be due to errors in the received TLP header. The TLP is deleted by the Hard IP and not presented to the Application Layer.	Use the error interface, configuration output interface, Hard IP reconfiguration interface to obtain more information on the error.
ECRC error	Transaction layer error which can be due to an ECRC check failure at the receiver despite the fact that the TLP is not malformed and the LCRC check is valid. The Hard IP block handles this TLP automatically. If the TLP is a non-posted request, the Hard IP block generates a completion with a completer abort status. The TLP is deleted by the Hard IP and not presented to the Application Layer.	Use the configuration output interface, Hard IP reconfiguration interface to obtain more information on the error.
Unsupported request	Transaction layer error which can be due to the completer being unable to fulfill the request. The TLP is deleted in the Hard IP block and not presented to the Application Layer. If the TLP is a non-posted request, the Hard IP block generates a completion with Unsupported Request status.	Use the configuration output interface, error interface, Hard IP reconfiguration interface to obtain more information on the error.
ACS violation	Transaction layer error which can be due to access control error in the received posted or non-posted request.	Use the configuration output interface, error interface, Hard IP reconfiguration interface to obtain more information on the error.
Uncorrectable internal error	Transaction layer error which can be due to an internal error that cannot be corrected by the hardware.	Use the error interface, configuration output interface, Hard IP reconfiguration interface and DBI registers to obtain more information on the error.
Atomic egress blocked		Use the error interface, configuration output interface, Hard IP reconfiguration interface to obtain more information on the error.
TLP prefix blocked	EP or RP only	Use the error interface, configuration output interface, Hard IP reconfiguration interface to obtain more information on the error.
Poisoned TLP egress blocked	EP or RP only	Use the error interface, configuration output interface, configuration intercept interface, Hard IP reconfiguration interface to obtain more information on the error.

6.1.3.1. Advanced Error Reporting (AER)

Each PCI Express compliant device must implement a basic level of error management and can optionally implement advanced error management. The PCI Express Advanced Error Reporting Capability is an optional Extended Capability that may be implemented by PCI Express device functions supporting advanced error control and reporting. The R-Tile Avalon-ST IP for PCI Express implements both basic and advanced error reporting. You can use the Hard IP Reconfiguration Interface to access the AER registers and also the PCIe Debug Toolkit.

6.2. Operating System Tools and Utilities

You can use Linux utilities like `lspci` and `setpci` to access the PCIe Configuration space of the device and retrieve information such as the negotiated link speed, negotiated link width, etc.

6.2.1. Using `lspci` Utility to Read Negotiated Link Speed

To read the negotiated link speed of the R-Tile Avalon-ST IP for PCIe from a host system, you can perform the following command:

```
sudo lspci -s $bdf -vvv
```

`-s` refers to "slot" and is used with the bus, device and function (bdf) number.

Figure 60. `lspci` Output Using the Bus, Device and Function Number

```
sudo lspci -s 98:00.0 -vvv
98:00.0 Unassigned class [ff00]: Altera Corporation Device 0000 (rev 01)
        Physical Slot: 28
        Control: I/O- Mem- BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr-
Stepping- SERR+ FastB2B- DisINTx-
        Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort-
<MAbort- >SERR- <PERR- INTx-
        NUMA node: 1
        Region 0: Memory at c9400000 (32-bit, non-prefetchable) [disabled]
[size=64K]
        Capabilities: [40] Power Management version 3
                Flags: PMEClk- DSI- D1- D2- AuxCurrent=0mA PME(D0+,D1-,D2-,D3hot
+,D3cold-)
                Status: D0 NoSoftRst+ PME-Enable- DSel=0 DScale=0 PME-
        Capabilities: [70] Express (v2) Endpoint, MSI 00
                DevCap: MaxPayload 512 bytes, PhantFunc 0, Latency L0s <64ns, L1
<1us
                        ExtTag+ AttnBtn- AttnInd- PwrInd- RBE+ FLReset-
SlotPowerLimit 75.000W
                DevCtl: CorrErr- NonFatalErr- FatalErr- UnsupReq-
                        RlxdOrd+ ExtTag+ PhantFunc- AuxPwr- NoSnoop+
                        MaxPayload 512 bytes, MaxReadReq 4096 bytes
                DevSta: CorrErr+ NonFatalErr- FatalErr- UnsupReq+ AuxPwr+
TransPend-
                LnkCap: Port #1, Speed 32GT/s, Width x16, ASPM not supported
                        ClockPM- Surprise- LLActRep- BwNot- ASPMOptComp+
                LnkCtl: ASPM Disabled; RCB 64 bytes Disabled- CommClk-
                        ExtSynch- ClockPM- AutWidDis- BWInt- AutBWInt-
                LnkSta: Speed 32GT/s (ok), Width x16 (ok)
                        TrErr- Train- SlotClk- DLActive- BWMgmt- ABWMgmt-
                DevCap2: Completion Timeout: Range ABCD, TimeoutDis+, NROPrPrP-,
LTR+
                         10BitTagComp+, 10BitTagReq+, OBFF Not Supported, ExtFmt
+, EETLPPrefix+, MaxEETLPPrefixes 1
                         EmergencyPowerReduction Not Supported,
EmergencyPowerReductionInit-
                         FRS-, TPHComp+, ExtTPHComp-
                         AtomicOpsCap: 32bit+ 64bit+ 128bitCAS+
                DevCtl2: Completion Timeout: 260ms to 900ms, TimeoutDis-, LTR-,
OBFF Disabled
                         AtomicOpsCtl: ReqEn+
                LnkCtl2: Target Link Speed: 32GT/s, EnterCompliance- SpeedDis-
                         Transmit Margin: Normal Operating Range,
EnterModifiedCompliance- ComplianceSOS-
                         Compliance De-emphasis: -6dB
                LnkSta2: Current De-emphasis Level: -6dB, EqualizationComplete-,
EqualizationPhase1-
                         EqualizationPhase2-, EqualizationPhase3-,
LinkEqualizationRequest-
        Capabilities: [100 v2] Advanced Error Reporting
                UESta:  DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF-
MalfTLP- ECRC- UnsupReq- ACSViol-
                UEMsk:  DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF-
MalfTLP- ECRC- UnsupReq- ACSViol-
                UESvrt: DLP+ SDES+ TLP- FCP+ CmpltTO- CmpltAbrt- UnxCmplt- RxOF+
MalfTLP+ ECRC- UnsupReq- ACSViol-
                CESta:  RxErr- BadTLP- BadDLLP- Rollover- Timeout-
AdvNonFatalErr-
                CEMsk:  RxErr- BadTLP- BadDLLP- Rollover- Timeout- AdvNonFatalErr
+
                AERCap: First Error Pointer: 00, ECRCGenCap+ ECRCGenEn-
ECRCChkCap+ ECRCChkEn-
                        MultHdrRecCap- MultHdrRecEn- TLPPfxPres- HdrLogCap-
                HeaderLog: 00000000 00000000 00000000 00000000
```

Figure 61. `lspci` Output Using the Bus, Device and Function Number (continued)

```
        Capabilities: [148 v1] Virtual Channel
                Caps:   LPEVC=0 RefClk=100ns PATEntryBits=1
                Arb:    Fixed- WRR32- WRR64- WRR128-
                Ctrl:   ArbSelect=Fixed
                Status: InProgress-
                VC0:    Caps:   PATOffset=00 MaxTimeSlots=1 RejSnoopTrans-
                        Arb:    Fixed- WRR32- WRR64- WRR128- TWRR128- WRR256-
                        Ctrl:   Enable+ ID=0 ArbSelect=Fixed TC/VC=ff
                        Status: NegoPending- InProgress-
        Capabilities: [174 v1] Alternative Routing-ID Interpretation (ARI)
                ARICap: MFVC- ACS-, Next Function: 0
                ARICtl: MFVC- ACS-, Function Group: 0
        Capabilities: [184 v1] Secondary PCI Express
                LnkCtl3: LnkEquIntrruptEn-, PerformEqu-
                LaneErrStat: 0
        Capabilities: [1b4 v1] Physical Layer 16.0 GT/s <?>
        Capabilities: [1e4 v1] Lane Margining at the Receiver <?>
        Capabilities: [22c v1] Extended Capability ID 0x2a
        Capabilities: [4ac v1] Data Link Feature <?>
        Capabilities: [d00 v1] Vendor Specific Information: ID=1172 Rev=0
Len=05c <?>
        Kernel modules: altera_cvp
```

The **LnkCap** under Capabilities indicates the advertised link speed and width capabilities of the device. The **LnkSta** under Capabilities indicates the negotiated link speed and width of the device.

You can use the following command if you do not know the bdf assigned to the device in the system topology:

```
sudo lspci -d $vid:$did -vvv
```

Figure 62. `lspci` Output Using the Vendor ID

```
$ sudo lspci -d 1172: -v
98:00.0 Unassigned class [ff00]: Altera Corporation Device 0000 (rev 01)
        Physical Slot: 28
        Flags: fast devsel, NUMA node 1
        Memory at c9400000 (32-bit, non-prefetchable) [disabled] [size=64K]
        Capabilities: [40] Power Management version 3
        Capabilities: [70] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
        Capabilities: [148] Virtual Channel
        Capabilities: [174] Alternative Routing-ID Interpretation (ARI)
        Capabilities: [184] Secondary PCI Express
        Capabilities: [1b4] Physical Layer 16.0 GT/s <?>
        Capabilities: [1e4] Lane Margining at the Receiver <?>
        Capabilities: [22c] Extended Capability ID 0x2a
        Capabilities: [4ac] Data Link Feature <?>
        Capabilities: [d00] Vendor Specific Information: ID=1172 Rev=0 Len=05c <?
>
        Kernel modules: altera_cvp
```

`-d` refers to "device" and is used with the Device ID that was configured during the parameterization of the R-Tile Avalon-ST IP for PCIe prior to compilation.

You can rescan the PCIe bus using the following commands. You must have root privileges to perform these commands:

To detach the device from the tree:

```
echo 1 > /sys/bus/pci/devices/0000\:98\:00.0/remove
```

To rescan the bus:

```
echo 1 > /sys/bus/pci/rescan
```

6.3. Signal Tap Logic Analyzer

You can use the Signal Tap Logic Analyzer to monitor the top-level signals below from the R-Tile Avalon-ST IP for PCIe as an additional debug tool for PCIe issues.

Table 111. Top-Level Signals to be Monitored for Debug Purposes

Signal	Description	Expected Value for Successful Link-up
`ninit_done`	A "1" on this active-low signal indicates that the FPGA device is not yet fully configured. A "0" indicates the device has been configured and is in normal operating mode. You need to instantiate the Reset Release IP and connect the output of that IP to `ninit_done`.	1'b0
`pin_perst_n_o`	This output signal to the FPGA fabric indicates if PERST# is asserted.	1'b1
`pX_reset_status_n_o`	This active-low signal is held low until `pin_perst_n` has been deasserted and the PCIe Hard IP has come out of reset. This signal is synchronous to `coreclkout_hip`. Traffic between the user logic in the FPGA core and the IP can start when `pX_reset_status_n_o` is asserted high.	1'b1
`pX_link_up_o`	When asserted, this signal indicates the link is up at the Physical Layer.	1'b1
`pX_dl_up_o`	When asserted, this signal indicates the Data link (DL) Layer is active.	1'b1
`pX_ltssm_state_delay_o[5:0]`	Indicates the LTSSM state. Note that there is a time difference between the actual link state at the physical level and the time it takes to reflect its value on this signal.	6'h11 (L0)

6.4. Using the Hard IP Reconfiguration Interface for Debugging

You can use the Hard IP reconfiguration interface of the R-Tile Avalon Streaming Intel FPGA IP for PCI Express to access configuration registers for debugging purpose. Refer to Hard IP Reconfiguration Interface on page 103 for more details on this interface.

6.4.1. Using the Hard IP Reconfiguration Interface to Enable and Read ECRC and LCRC Error Counts

Offset			Bit Positions	Register
x16	x8	x4		
0x0000_0119	0x0000_0119	0x0000_0119	[0]	Register: AER_CAP/ ADV_ERR_CAP_CTRL_OFF Field: ECRC_CHECK_EN
0x0000_036C	0x0000_032C	0x0000_02D4	[1:0]	Set to 2'b11 to clear the counters.
			[4:2]	Set to 3'b111 to enable the counters.
0x0000_036D	0x0000_032D	0x0000_02D5	[7:0]	Set to 0x00. Reserved.
0x0000_036E	0x0000_032E	0x0000_02D6	[7:0]	Event Number. For LCRC error count, set to 0x01. For ECRC error count, set to 0x02.
0x0000_036F	0x0000_032F	0x0000_02D7	[7:0]	Group Number. For LCRC error count, set to 0x02. For ECRC error count, set to 0x03.
0x0000_0370	0x0000_0330	0x0000_02D8	[7:0]	Error counter data bits [7:0].
0x0000_0371	0x0000_0331	0x0000_02D9	[7:0]	Error counter data bits [15:8].
0x0000_0372	0x0000_0332	0x0000_02DA	[7:0]	Error counter data bits [23:16].
0x0000_0373	0x0000_0333	0x0000_02DB	[7:0]	Error counter data bits [31:24].

Follow the steps below to access registers in the table above using the Hard IP reconfiguration interface:

1. Enable the Hard IP reconfiguration interface using the IP Parameter Editor.
2. Enable CRC checking in the register AER_CAP/ADV_ERR_CAP_CTRL_OFF.
3. Set the group number and event number.
4. Enable the counters.
5. Read the counter data.

Below is an example to enable the counter for the LCRC:

1. Enable the Hard IP reconfiguration interface using the IP Parameter Editor.
2. Enable CRC checking by performing a read-modify-write to the ECRC_CHECK_EN field within the register AER_CAP/ADV_ERR_CAP_CTRL_OFF.
   a. p0_hip_reconfig_write = 1’b1
   b. p0_hip_reconfig_address[31:0] = 0x0000_0119
   c. p0_hip_reconfig_writedata[7:0] = 8'h01
3. Perform read-modify-write to address 0x0000_036F to set Group Number to 0x2.

   a. p0_hip_reconfig_write = 1’b1
   b. p0_hip_reconfig_address[31:0] = 0x0000_036F
   c. p0_hip_reconfig_writedata[7:0] = 8'h02
4. Perform read-modify-write to address 0x0000_036E to set Event Number to 0x1.
   a. p0_hip_reconfig_write = 1’b1
   b. p0_hip_reconfig_address[31:0] = 0x0000_036E
   c. p0_hip_reconfig_writedata[7:0] = 8'h01
5. Perform read-modify-write to address 0x0000_036D with 0x0.
   a. p0_hip_reconfig_write = 1’b1
   b. p0_hip_reconfig_address[31:0] = 0x0000_036D
   c. p0_hip_reconfig_writedata[7:0] = 8'h00
6. Perform read-modify-write to address 0x0000_036C to set Enable Event Counter.
   a. p0_hip_reconfig_write = 1’b1
   b. p0_hip_reconfig_address[31:0] = 0x0000_036C
   c. p0_hip_reconfig_writedata[7:0] = 8'h1C
7. Read the error counter data by performing a read operation from registers 0x0000_0370, 0x0000_0371, 0x0000_0372, and 0x0000_0373.

6.5. Other Interfaces That Can Be Used for Debug Activities

Use the interfaces below as additional debug tools for issues you may observe on the PCIe link when using the R-Tile Avalon-ST IP for PCIe.

- Error Interface: Refer to Error Interface for details on this interface and its address map.
- Configuration Intercept Interface: Refer to Configuration Intercept Interface for details on this interface and its address map.
- Credit Interface: Refer to the sections TX Flow Control Interface and RX Flow Control Interface for details on these interfaces and their address maps.
- Hard IP Reconfiguration Interface: Refer to Hard IP Reconfiguration Interface for details on this interface and its address map.

6.6. Debug Toolkit

6.6.1. Overview

The R-Tile Debug Toolkit (DTK) is a System Console-based tool for R-Tile that provides real-time control, monitoring and debugging of the PCIe links.

The R-Tile Debug Toolkit allows you to perform the following actions on a per port basis:

- Monitor the IP configuration and the link status.
- Monitor the PCIe Configuration space.
- Monitor different counters for errors and event conditions.
- Perform Time margin (horizontal) and Voltage margin (vertical) and compare margin values against recommended masks for each channel.

Note: The 22.2 R-Tile Avalon Streaming IP provides an initial version of the Debug Toolkit. Consider this version a Beta release of this tool.

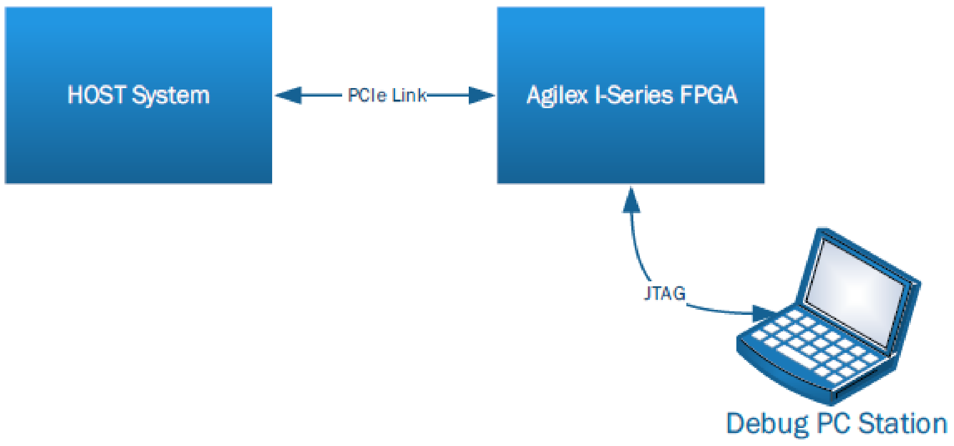
Note: The 22.2 version of Intel Quartus Prime supports enabling the Debug Toolkit for Endpoint mode only.

Note: If implementing temperature monitoring, the Application logic must prevent a temperature readout to the R-Tile temperature sensing diode (TSD) while the Debug Toolkit GUI is open. Refer to the Intel Agilex 7 F-Series and I-Series Power Management User Guide for additional information.

Overview of the R-Tile Debug Toolkit below describes the 3 main components that are active when you are using the Debug Toolkit:

- Host: The link partner component connected to R-Tile through a PCIe link.
- Intel Agilex 7 I-Series FPGA: This component includes the R-Tile PCIe IP under debug.
- Debug PC Station: An additional computer system with the Intel Quartus Prime Pro Edition software installed which is running the Debug Toolkit and is connected to the Intel Agilex 7 I-Series FPGA through a JTAG connection.

Figure 63. Overview of the R-Tile Debug Toolkit



When you enable the R-Tile Debug Toolkit, the `intel_rtile_pcie_ast` module of the generated IP includes the Debug Toolkit modules and related logic as shown in Debug Toolkit Arbitration.

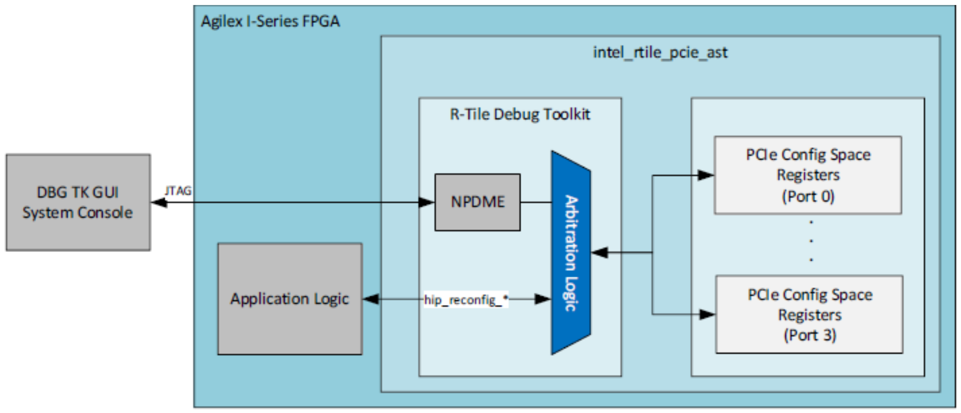
The Debug PC Station that is running the System Console tool connects to the Debug Toolkit via a Native PHY Debug Master Endpoint (NPDME). This connection is made using JTAG through an Intel FPGA Download Cable.

When you enable the R-Tile Debug Toolkit in the IP Parameter Editor, the Hard IP Reconfiguration Interface will be enabled. A multiplexer is implemented to allow a dynamic switching between the Application logic and the Debug Toolkit. The Application logic has the default access. Upon launching the Debug Toolkit in the System Console tool, the arbitration selection happens automatically. While the DTK is open in the System Console, Application logic will not be able to drive signals on the Hard IP Reconfiguration Interface. Once the Debug Toolkit window in the System Console is closed, the Application logic can once again drive the Hard IP Reconfiguration Interface.

To make sure there are no transactions in progress, the Debug Toolkit can be launched successfully only if there is no pending Read or Write transaction made by the Application logic on the Hard IP Reconfiguration Interface.

Note: The System Console message window will show an error message indicating there is an ongoing request, and the Debug Toolkit cannot be launched.

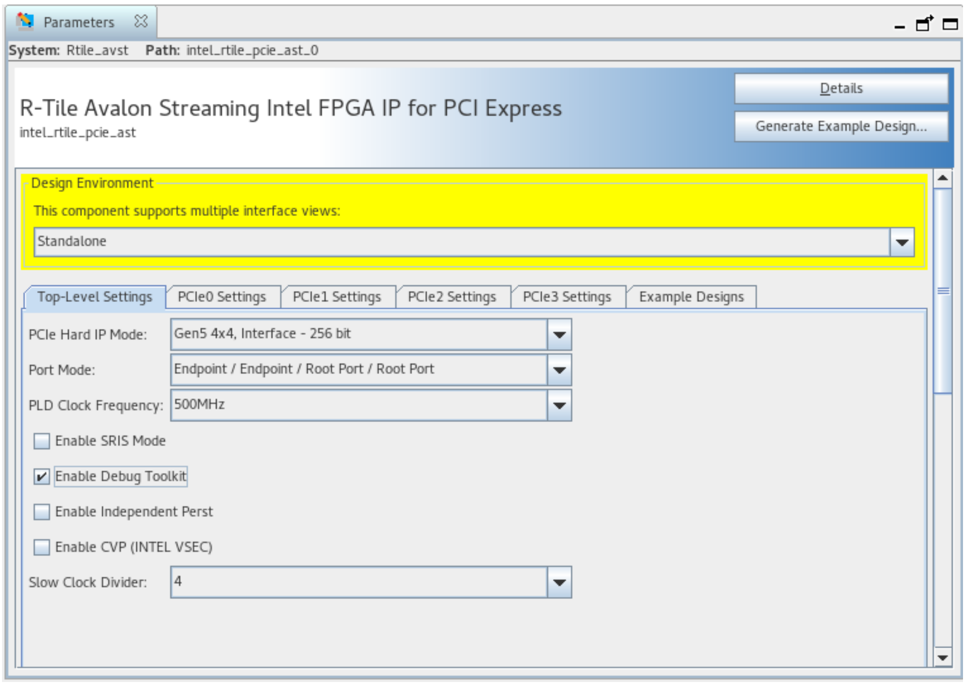
Figure 64. Debug Toolkit Arbitration



6.6.2. Enabling the R-Tile Debug Toolkit

To enable the R-Tile Debug Toolkit in your design, enable the option **Enable Debug Toolkit** in the **Top-Level Settings** tab of the R-Tile Avalon Streaming Intel FPGA IP for PCI Express. When the Debug Toolkit is enabled via the **Enable Debug Toolkit** parameter, the Hard IP Reconfiguration Interface is automatically enabled.

Note: When the Debug Toolkit is enabled via the **Enable Debug Toolkit** parameter, the PHY Reconfiguration interface is automatically enabled. A 100MHz clock must be provided on the `xcvr_reconfig_clk` port for proper functionality of the Debug Toolkit.



System: Rtile_avst Path: intel_rtile_pcie_ast_0

R-Tile Avalon Streaming Intel FPGA IP for PCI Express
intel_rtile_pcie_ast

Design Environment
This component supports multiple interface views: Standalone

PCIe Hard IP Mode: Gen5 4x4, Interface - 256 bit
Port Mode: Endpoint / Endpoint / Root Port / Root Port
PLD Clock Frequency: 500MHz
[ ] Enable SRIS Mode
[✓] Enable Debug Toolkit
[ ] Enable Independent Perst
[ ] Enable CVP (INTEL VSEC)
Slow Clock Divider: 4

6.6.3. Launching the R-Tile Debug Toolkit

You can use the design example provided with the Intel R-Tile Avalon Streaming Intel FPGA IP for PCIe to familiarize yourself with the R-Tile Debug Toolkit. Follow the steps described in the R-Tile Avalon Streaming Intel FPGA IP for PCI Express Design Example User Guide to generate the SRAM Object File (.sof) for this design example.

To use the R-Tile Debug Toolkit, program the .sof in the Intel Agilex 7 I-Series Development Kit. Then, open the System Console and load the design into the System Console as well. Loading the .sof into the System Console allows the System Console to communicate with the design using the NPDME unit.

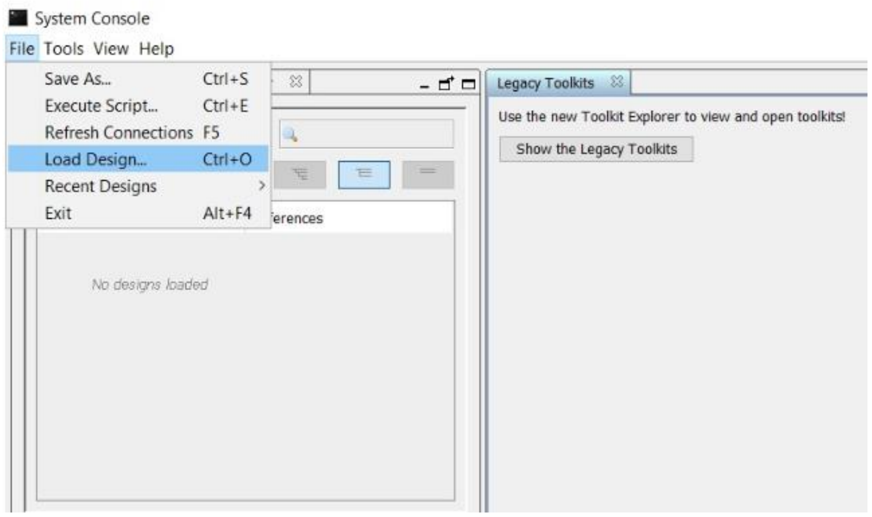
Here are the steps to complete these tasks:

1. Use the Intel Quartus Prime Programmer to download the .sof to the Intel FPGA Development Kit.

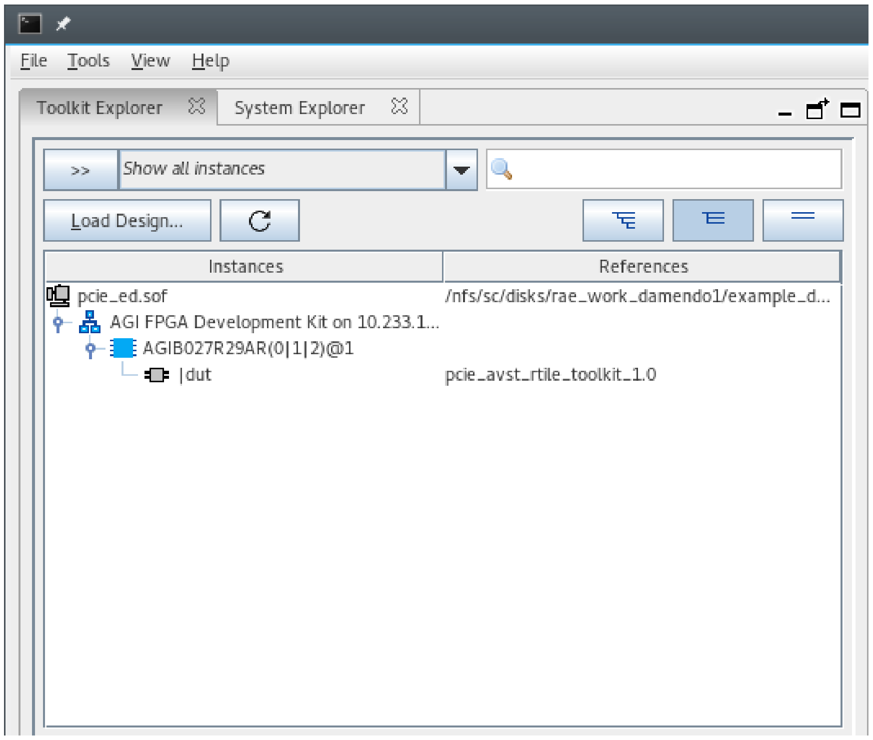
   Note: To ensure correct operation, use a full installation of the Intel Quartus Prime Pro Edition Software and Devices of the same version of the Intel Quartus Prime Programmer and Intel Quartus Prime Pro Edition software that you used to generate the .sof.

   Note: A standalone installation of the Intel Quartus Prime Pro Edition Programmer and Tools does not work.

2. To load the design into System Console:
   a. Launch the Intel Quartus Prime Pro Edition software.
   b. Start System Console by choosing **Tools**, then **System Debugging Tools**, then **System Console**.
   c. On the System Console File menu, select **Load Design** and browse to the .sof file.



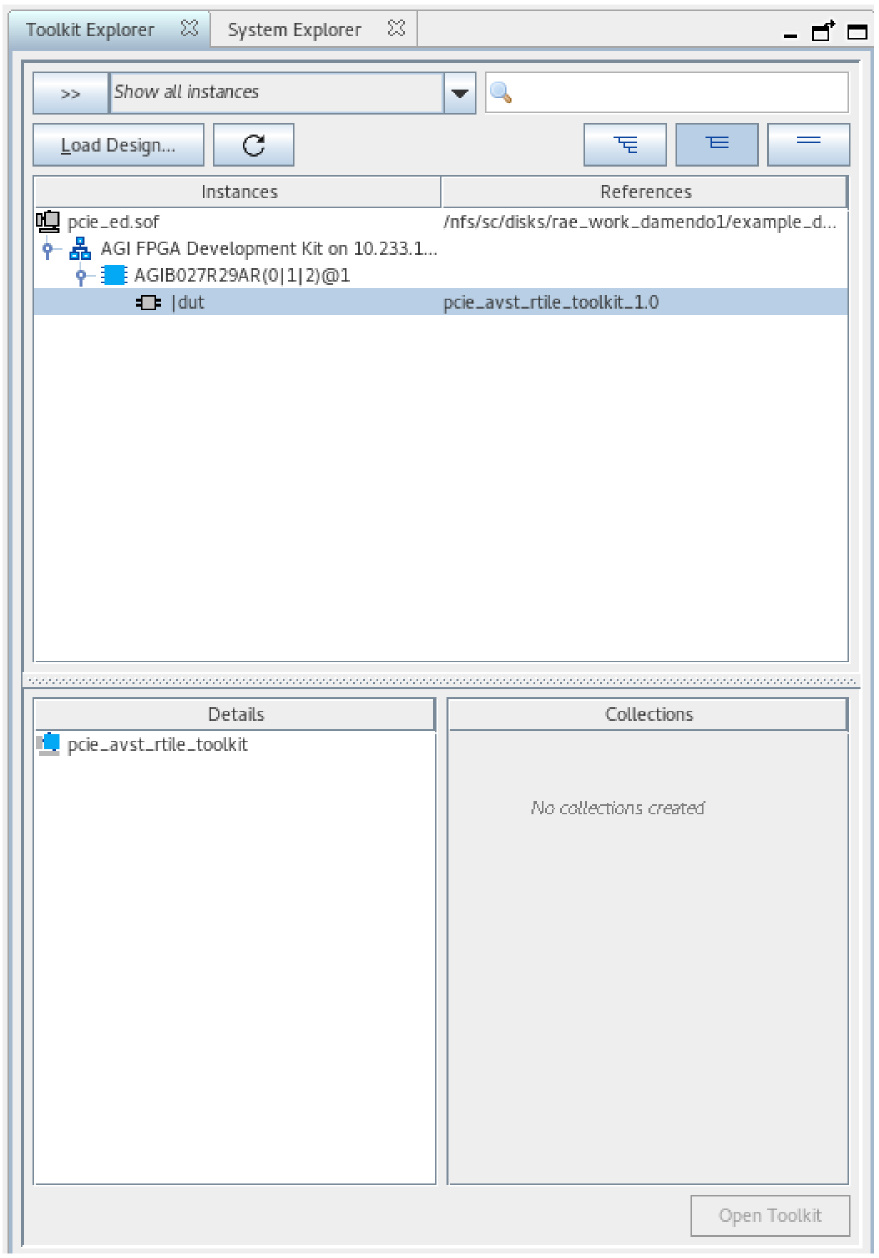
d. Select the .sof and click OK. The .sof loads to the System Console.



Instances	References
pcie_ed.sof	/nfs/sc/disks/rae_work_damendo1/example_d...
AGI FPGA Development Kit on 10.233.1...	
AGIB027R29AR(0|1|2)@1	
|dut	pcie_avst_rtile_toolkit_1.0

3. The System Console Toolkit Explorer window will list all the DUTs in the design that have the R-Tile Debug Toolkit enabled.

   a. Select the DUT with the R-Tile Debug Toolkit under the **Instances** column. This will open the Debug Toolkit instance of that DUT in the **Details** window.



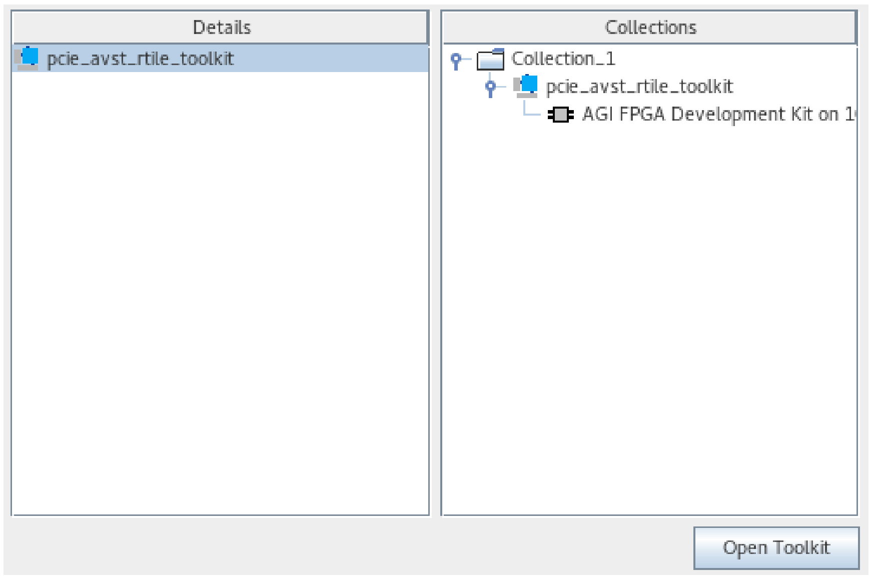
Instances	References
pcie_ed.sof	/nfs/sc/disks/rae_work_damendo1/example_d...
AGI FPGA Development Kit on 10.233.1...	
AGIB027R29AR(0|1|2)@1	
|dut	pcie_avst_rtile_toolkit_1.0

Details
pcie_avst_rtile_toolkit

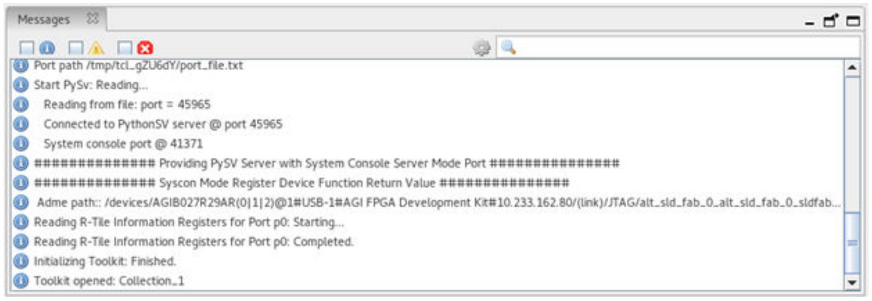
Collections
No collections created

b. Select `pcie_avst_rtile_toolkit` under the **Details** panel.

c. Click **Open Toolkit** to open that instance of the Toolkit.

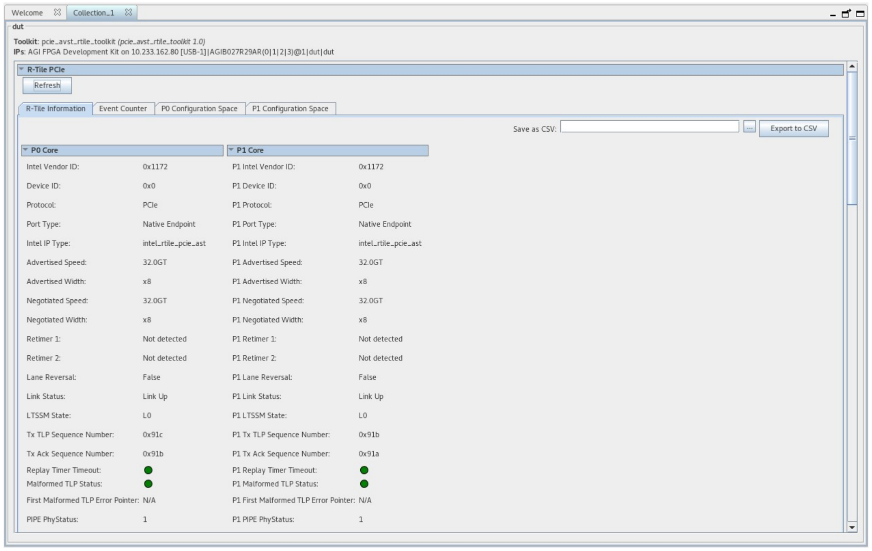


d. Once the Debug Toolkit is initialized and loaded, you will see the following message in the **Messages** window:



```
Port path /tmp/tcl_gZU6dY/port_file.txt
Start PySv: Reading...
   Reading from file: port = 45965
   Connected to PythonSV server @ port 45965
   System console port @ 41371
############## Providing PySV Server with System Console Server Mode Port ###############
############## Syscon Mode Register Device Function Return Value ###############
 Adme path:: /devices/AGIB027R29AR(0|1|2)@1#USB-1#AGI FPGA Development Kit#10.233.162.80/(link)/JTAG/alt_sld_fab_0_alt_sld_fab_0_sldfab...
Reading R-Tile Information Registers for Port p0: Starting...
Reading R-Tile Information Registers for Port p0: Completed.
Initializing Toolkit: Finished.
Toolkit opened: Collection_1
```

e. A new window Main View will open with a view of the R-Tile PCIe Toolkit.



Toolkit: pcie_avst_rtile_toolkit *(pcie_avst_rtile_toolkit 1.0)*
IPs: AGI FPGA Development Kit on 10.233.162.80 [USB-1]|AGIB027R29AR(0|1|2|3)@1|dut|dut

P0 Core		P1 Core	
Intel Vendor ID:	0x1172	P1 Intel Vendor ID:	0x1172
Device ID:	0x0	P1 Device ID:	0x0
Protocol:	PCIe	P1 Protocol:	PCIe
Port Type:	Native Endpoint	P1 Port Type:	Native Endpoint
Intel IP Type:	intel_rtile_pcie_ast	P1 Intel IP Type:	intel_rtile_pcie_ast
Advertised Speed:	32.0GT	P1 Advertised Speed:	32.0GT
Advertised Width:	x8	P1 Advertised Width:	x8
Negotiated Speed:	32.0GT	P1 Negotiated Speed:	32.0GT
Negotiated Width:	x8	P1 Negotiated Width:	x8
Retimer 1:	Not detected	P1 Retimer 1:	Not detected
Retimer 2:	Not detected	P1 Retimer 2:	Not detected
Lane Reversal:	False	P1 Lane Reversal:	False
Link Status:	Link Up	P1 Link Status:	Link Up
LTSSM State:	L0	P1 LTSSM State:	L0
Tx TLP Sequence Number:	0x91c	P1 Tx TLP Sequence Number:	0x91b
Tx Ack Sequence Number:	0x91b	P1 Tx Ack Sequence Number:	0x91a
Replay Timer Timeout:	🟢	P1 Replay Timer Timeout:	🟢
Malformed TLP Status:	🟢	P1 Malformed TLP Status:	🟢
First Malformed TLP Error Pointer:	N/A	P1 First Malformed TLP Error Pointer:	N/A
PIPE PhyStatus:	1	P1 PIPE PhyStatus:	1

6.6.4. Using the R-Tile Debug Toolkit

The following sections describe the different tabs and features available in the Debug Toolkit.

6.6.4.1. R-Tile Information

The following table lists a summary of the R-Tile PCIe IP parameter settings in the PCIe IP Parameter Editor when the IP was generated, as read by the R-Tile Debug Toolkit when initialized. Depending on the Hard IP Mode selected during the IP configuration (for example Gen5 2x8), then this tab will populate the R-Tile information for each core (P0 core, P1 core, etc.). Also, please note that:

- All the information in the **R-Tile information** tab is read-only.
- Use the **Refresh** button to read the settings.

Table 112. R-Tile PCIe IP Parameter Settings

Parameters	Values	Descriptions
Intel Vendor ID	0x1172	Indicates the Vendor ID as set in the IP Parameter Editor.
Device ID	x0	This is a unique identifier for the device that is assigned by the vendor.
Protocol	PCIe	Indicates the protocol.
Port Type	Native Endpoint	Indicates the Hard IP Port type.
Intel IP Type	intel_rtile_pcie_ast	Indicates the IP type used.
Advertised Speed	8.0GT, 16.0GT, 32.0GT	Indicates the advertised speed as configured in the IP Parameter Editor.
Advertised Width	x16, x8, x4	Indicates the advertised width as configured in the IP Parameter Editor.
Negotiated Speed	2.5GT, 5.0GT, 8.0GT, 16.0GT, 32.0GT	Indicates the negotiated speed during link training.
Negotiated Width	x16, x8, x4, x2, x1	Indicates the negotiated link width during link training.
Retimer 1	Detected, Not Detected	Indicates if a retimer was detected between R-Tile and the link partner.
Retimer 2	Detected, Not Detected	Indicates if a retimer was detected between R-Tile and the link partner.
Lane Reversal	True, False	Indicates if lane reversal happens on the link.
Link Status	Link up, Link Down	Indicates if the link (DL) is up or not.
LTSSM State	Refer to Hard IP Status Interface.	Indicates the current state of the link.
Tx TLP Sequence Number	Hexadecimal value	Indicates the next transmit sequence number for the transmit TLP.
Tx Ack Sequence Timeout	Hexadecimal value	Indicates the ACK sequence number which is updated by receiving ACK/NAK DLLP.
Replay Timer Timeout	Green, Red	Green: no timeout Red: timeout
Malformed TLP Status	Green, Red	Green: no malformed TLP

continued...

Parameters	Values	Descriptions
		Red: malformed TLP detected
First Malformed TLP Error Pointer	• AtomicOp Address Alignment • AtomicOp Operand • AtomicOp Byte Enable • TLP Length Mismatch • Max Payload Size • Message TLP Without TC0 • Invalid TC • Unexpected Route Bit in Message TLP • Unexpected CRS Status in Completion TLP • Byte Enable • Memory Address 4KB Boundary • TLP Prefix Rules • Translation Request Rules • Invalid TLP Type • Completion Rules • Application	
PIPE PhyStatus	1, 0	Indicates the PMA and MAC are in reset mode. 1: PMA and MAC are out of reset. 0: PMA and MAC are in reset.

6.6.4.2. Event Counters

This tab allows you to read the events occurring at the link level, such as the number of received errors, framing errors, etc. for each port. You can use the **Clear PX counter** button at the end of each event counter list to reset these counters. Also, please note that:

- All the information is read-only.
- The per-lane related counters correspond to the logical lanes.
- Use the **Refresh Event Counter** button to read the event counters' registers.

Table 113. List of Events with Event Counters

Group	Counter Size	Description
0	4 bits	Elastic Buffer (EBUF) Overflow
		Elastic Buffer (EBUF) Under-run
		Decode Error
		Running Disparity Error
		Gen3 (G3) SKP OS Parity Error
		Gen3 - Gen5 (G3-G5) SYNC Header Error
		Rx Valid deassert w/o EIOS
		CTL SKP OS Parity Error
		1st Retimer Parity Error
		2nd Retimer Parity Error

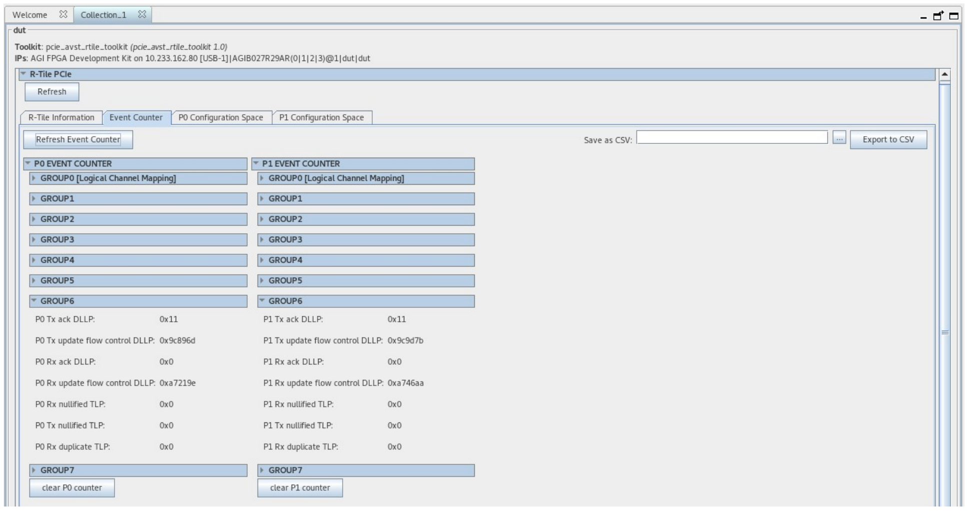
continued...

Group	Counter Size	Description
		Gen4 (G4) Margin CRC & Parity Error
1	8 bits	Detect Electrical Idle (EI) Inferred
		Receiver Error
		Rx Recovery Request
		N_FTS Timeout
		Gen3 (G3) Framing Error
		Deskew Error
2	8 bits	Bad TLP
		LCRC Error
		Bad DLLP
		Replay Number Rollover
		Replay Timeout
		Rx Nak DLLP
		Tx Nak DLLP
		Retry TLP
3	8 bits	Flow Control (FC) Timeout
		Poisoned TLP
		ECRC Error
		Unsupported Request
		Completer Abort
		Completion Timeout
4	4 bits	Elastic Buffer (EBUF) SKP Addition
		Elastic Buffer (EBUF) SKP Deletion
5	32 bits	L0 to Recovery
		L1 to Recovery
		ASPM L1 Reject
		L1 Entry
		L2 Entry
		Speed Change
		Link Width Change
6	32 bits	Tx Ack DLLP
		Tx Update Flow Control (FC) DLLP
		Rx Ack DLLP
		Rx Update Flow Control (FC) DLLP
		Rx Nullified TLP
		Tx Nullified TLP

continued...

Group	Counter Size	Description
		Rx Duplicate TLP
7	32 bits	Tx Memory Write
		Tx Memory Read TLP
		Tx Configuration Write TLP
		Tx Configuration Read TLP
		Tx IO Write TLP
		Tx IO Read TLP
		Tx Completion without Data TLP
		Tx Completion with Data TLP
		Tx Message TLP (VC Only) TLP
		Tx Atomic TLP
		Tx TLP with Prefix
		Rx Memory Write TLP
		Rx Memory Read TLP
		Rx Configuration Write TLP
		Rx Configuration Read TLP
		Rx IO Write TLP
		Rx IO Read TLP
		Rx Completion without Data TLP
		Rx Completion with Data TLP
		Rx Message TLP (VC Only) TLP
		Rx Atomic TLP
		Rx TLP with Prefix

Figure 65. Event Counters Tab



Toolkit: pcie_avst_rtile_toolkit *(pcie_avst_rtile_toolkit 1.0)*
IPs: AGI FPGA Development Kit on 10.233.162.80 [USB-1]|AGIB027R29AR(0|1|2|3)@1|dut|dut

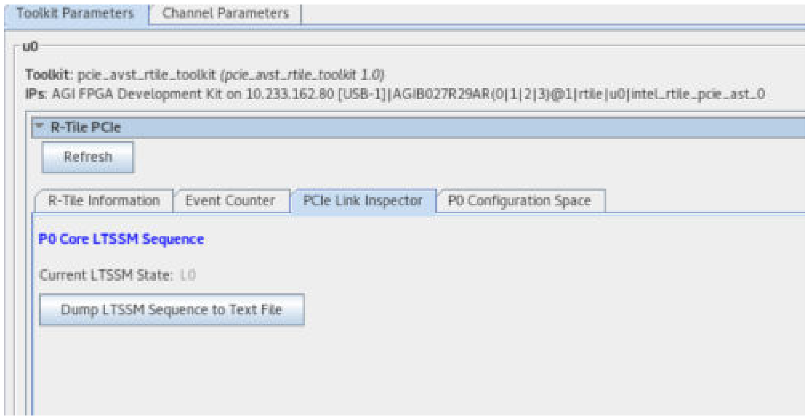
```
P0 Tx ack DLLP:                  0x11        P1 Tx ack DLLP:                  0x11
P0 Tx update flow control DLLP:  0x9c896d    P1 Tx update flow control DLLP:  0x9c9d7b
P0 Rx ack DLLP:                  0x0         P1 Rx ack DLLP:                  0x0
P0 Rx update flow control DLLP:  0xa7219e    P1 Rx update flow control DLLP:  0xa746aa
P0 Rx nullified TLP:             0x0         P1 Rx nullified TLP:             0x0
P0 Tx nullified TLP:             0x0         P1 Tx nullified TLP:             0x0
P0 Rx duplicate TLP:             0x0         P1 Rx duplicate TLP:             0x0
```

6.6.4.3. Link Inspector

The Link Inspector is found under the **PCIe Link Inspector** tab after opening the Debug Toolkit.

The Link Inspector is enabled by default when the **Enable Debug Toolkit** parameter is enabled. It tracks up to 1024 state transitions with the capability to dump them into a file.

Figure 66. View of the Link Inspector in 1x16 Mode



Toolkit: pcie_avst_rtile_toolkit (pcie_avst_rtile_toolkit 1.0)
IPs: AGI FPGA Development Kit on 10.233.162.80 [USB-1]|AGIB027R29AR(0|1|2|3)@1|rtile|u0|intel_rtile_pcie_ast_0

Current LTSSM State: L0

When the **Dump LTSSM Sequence to Text File** button is initially clicked, a text file (ltssm_sequence_dump_p*.txt) with the LTSSM information is created in the location where the System Console window is opened. Depending on the PCIe topology, there can be up to four text files. Subsequent LTSSM sequence dumps will append to the respective files.

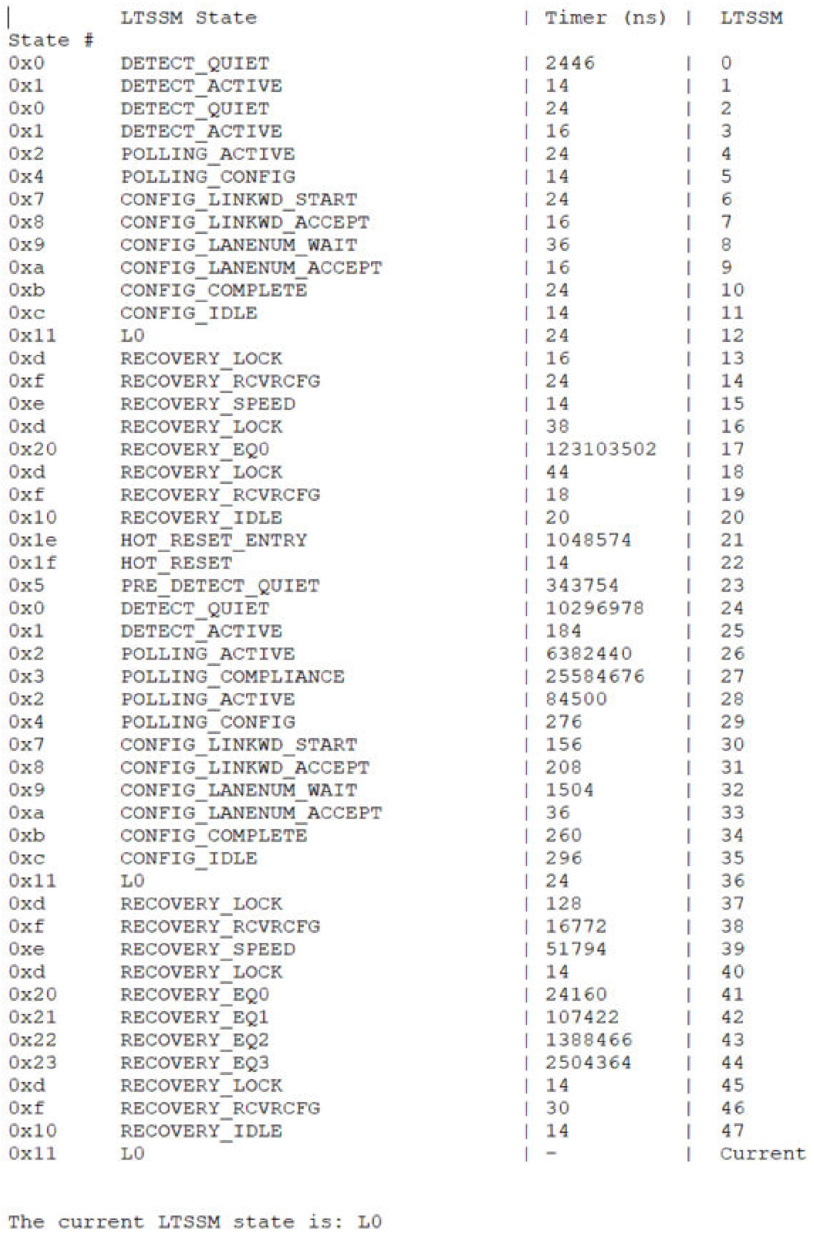
Note: If you open System Console in a directory that is not writable, the text file is not generated. To avoid this issue, open System Console from the Command Prompt window (on a Windows system) or change the directory's permission settings to writable.

Each LTSSM monitor has a FIFO storing the time values and captured LTSSM states. When you choose to dump out the LTSSM states, reads are dependent on the FIFO elements and will empty out the FIFO.

The Link Inspector only writes to its FIFO if there is a state transition. In cases where the link is stable in L0, there is no write and hence no text file is dumped.

When you want to dump the LTSSM sequence, a single read of the FIFO status of the respective core is performed. Depending on the empty status and how many entries are in the FIFO, successive reads are executed.

Figure 67. Example LTSSM Sequence Dump



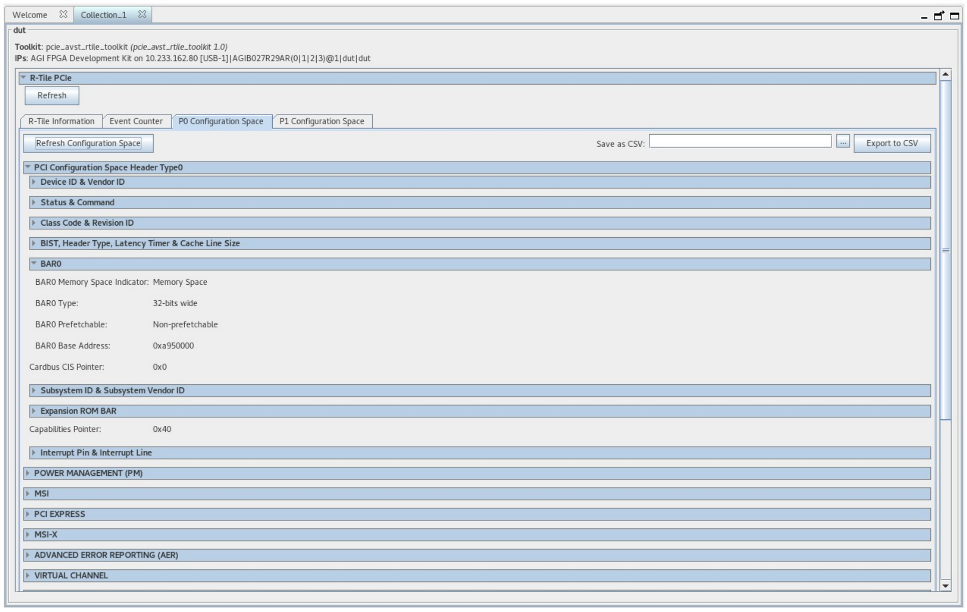
```
|           LTSSM State                          | Timer (ns)  |  LTSSM
State #
0x0       DETECT_QUIET                         | 2446        |  0
0x1       DETECT_ACTIVE                        | 14          |  1
0x0       DETECT_QUIET                         | 24          |  2
0x1       DETECT_ACTIVE                        | 16          |  3
0x2       POLLING_ACTIVE                       | 24          |  4
0x4       POLLING_CONFIG                       | 14          |  5
0x7       CONFIG_LINKWD_START                  | 24          |  6
0x8       CONFIG_LINKWD_ACCEPT                 | 16          |  7
0x9       CONFIG_LANENUM_WAIT                  | 36          |  8
0xa       CONFIG_LANENUM_ACCEPT                | 16          |  9
0xb       CONFIG_COMPLETE                      | 24          |  10
0xc       CONFIG_IDLE                          | 14          |  11
0x11      L0                                   | 24          |  12
0xd       RECOVERY_LOCK                        | 16          |  13
0xf       RECOVERY_RCVRCFG                     | 24          |  14
0xe       RECOVERY_SPEED                       | 14          |  15
0xd       RECOVERY_LOCK                        | 38          |  16
0x20      RECOVERY_EQ0                         | 123103502   |  17
0xd       RECOVERY_LOCK                        | 44          |  18
0xf       RECOVERY_RCVRCFG                     | 18          |  19
0x10      RECOVERY_IDLE                        | 20          |  20
0x1e      HOT_RESET_ENTRY                      | 1048574     |  21
0x1f      HOT_RESET                            | 14          |  22
0x5       PRE_DETECT_QUIET                     | 343754      |  23
0x0       DETECT_QUIET                         | 10296978    |  24
0x1       DETECT_ACTIVE                        | 184         |  25
0x2       POLLING_ACTIVE                       | 6382440     |  26
0x3       POLLING_COMPLIANCE                   | 25584676    |  27
0x2       POLLING_ACTIVE                       | 84500       |  28
0x4       POLLING_CONFIG                       | 276         |  29
0x7       CONFIG_LINKWD_START                  | 156         |  30
0x8       CONFIG_LINKWD_ACCEPT                 | 208         |  31
0x9       CONFIG_LANENUM_WAIT                  | 1504        |  32
0xa       CONFIG_LANENUM_ACCEPT                | 36          |  33
0xb       CONFIG_COMPLETE                      | 260         |  34
0xc       CONFIG_IDLE                          | 296         |  35
0x11      L0                                   | 24          |  36
0xd       RECOVERY_LOCK                        | 128         |  37
0xf       RECOVERY_RCVRCFG                     | 16772       |  38
0xe       RECOVERY_SPEED                       | 51794       |  39
0xd       RECOVERY_LOCK                        | 14          |  40
0x20      RECOVERY_EQ0                         | 24160       |  41
0x21      RECOVERY_EQ1                         | 107422      |  42
0x22      RECOVERY_EQ2                         | 1388466     |  43
0x23      RECOVERY_EQ3                         | 2504364     |  44
0xd       RECOVERY_LOCK                        | 14          |  45
0xf       RECOVERY_RCVRCFG                     | 30          |  46
0x10      RECOVERY_IDLE                        | 14          |  47
0x11      L0                                   | -           |  Current


The current LTSSM state is: L0
```

6.6.4.4. Configuration Space

This tab allows you to read the configuration space registers directly, without the need to issue a Configuration Read from the link partner. Depending on the Hard IP Mode selected during the IP configuration (for example Gen5 2x8), you will see a separate tab with the configuration space for each port. Also, please note that:

- All the information is read-only.
- The per-lane information under the tab **Configuration Space** corresponds to the logical lanes.
- Use the **Refresh Configuration Space** button to read the Configuration Space registers.

Figure 68. Configuration Space Tab



Toolkit: pcie_avst_rtile_toolkit *(pcie_avst_rtile_toolkit 1.0)*
IPs: AGI FPGA Development Kit on 10.233.162.80 [USB-1]|AGIB027R29AR(0|1|2|3)@1|dut|dut

BAR0 Memory Space Indicator:	Memory Space
BAR0 Type:	32-bits wide
BAR0 Prefetchable:	Non-prefetchable
BAR0 Base Address:	0xa950000
Cardbus CIS Pointer:	0x0
Capabilities Pointer:	0x40

6.6.4.5. Channel Parameters

The **Channel Parameters** window allows you to read the transmitter and receiver settings for a given channel. It has the following 4 sub-windows.

- General PHY
- Tx Path
- Rx Path
- Lane Margining

Use the **Lane Refresh** button to read the status of the General PHY, TX Path, and RX Path sub-windows for each channel.

Note: The **Channel Parameters** tab is only available in the following OPNs:

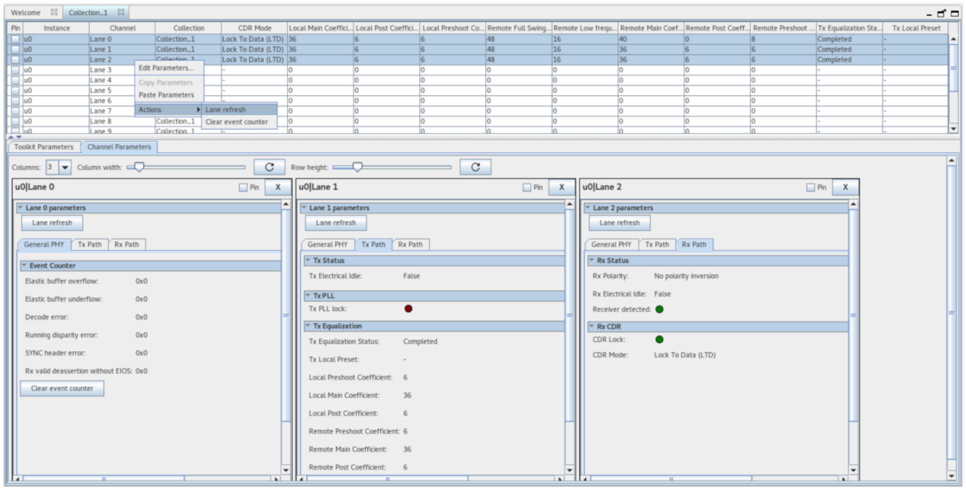
- AGIx027R29AxxxxR2
- AGIx027R29AxxxxR3
- AGIx027R29BxxxxR3
- AGIx023R18AxxxxR0
- AGIx041R29DxxxxR0
- AGIx041R29DxxxxR1

For additional details on OPN decoding, refer to the *Available Options* section of the Intel Agilex 7 FPGAs and SoCs Device Overview.

Note: To refresh channel parameters for more than one lane simultaneously, select the lanes under the **Collection** tab, right click and select **Refresh**.

Note: The per-lane information under the **Channel Parameters** tab corresponds to the physical lanes.

Figure 69. Channel Parameters Tab Showing 3 Lanes Selected with the Menu to be Refreshed



Pin	Instance	Channel	Collection	CDR Mode	Local Main Coeffici...	Local Post Coeffici...	Local Preshoot Co...	Remote Full Swing...	Remote Low frequ...	Remote Main Coef...	Remote Post Coeff...	Remote Preshoot ...	Tx Equalization Sta...	Tx Local Preset
	u0	Lane 0	Collection_1	Lock To Data (LTD)	36	6	6	48	16	40	0	8	Completed	-
	u0	Lane 1	Collection_1	Lock To Data (LTD)	36	6	6	48	16	36	6	6	Completed	-
	u0	Lane 2	Collection_1	Lock To Data (LTD)	36	6	6	48	16	36	6	6	Completed	-
	u0	Lane 3		-	0	0	0	0	0	0	0	0	-	-
	u0	Lane 4		-	0	0	0	0	0	0	0	0	-	-
	u0	Lane 5		-	0	0	0	0	0	0	0	0	-	-
	u0	Lane 6		-	0	0	0	0	0	0	0	0	-	-
	u0	Lane 7			0	0	0	0	0	0	0	0	-	-
	u0	Lane 8	Collection_1		0	0	0	0	0	0	0	0	-	-
	u0	Lane 9	Collection_1	-	0	0	0	0	0	0	0	0	-	-

Columns: 3 Column width: Row height:

u0|Lane 0

Lane 0 parameters

Event Counter

Elastic buffer overflow:	0x0
Elastic buffer underflow:	0x0
Decode error:	0x0
Running disparity error:	0x0
SYNC header error:	0x0
Rx valid deassertion without EIOS:	0x0

u0|Lane 1

Lane 1 parameters

Tx Status

Tx Electrical Idle:	False

Tx PLL

Tx PLL lock:	●

Tx Equalization

Tx Equalization Status:	Completed
Tx Local Preset:	-
Local Preshoot Coefficient:	6
Local Main Coefficient:	36
Local Post Coefficient:	6
Remote Preshoot Coefficient:	6
Remote Main Coefficient:	36
Remote Post Coefficient:	6

u0|Lane 2

Lane 2 parameters

Rx Status

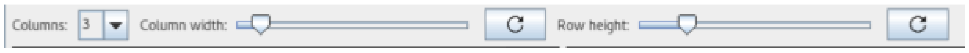
Rx Polarity:	No polarity inversion
Rx Electrical Idle:	False
Receiver detected:	●

Rx CDR

CDR Lock:	●
CDR Mode:	Lock To Data (LTD)

You can use the Columns drop down menu, the Column width and the Row height bars to adjust the graphical interface when monitoring multiple lanes at the same time.

Figure 70. Parameters to Adjust the Graphical Interface

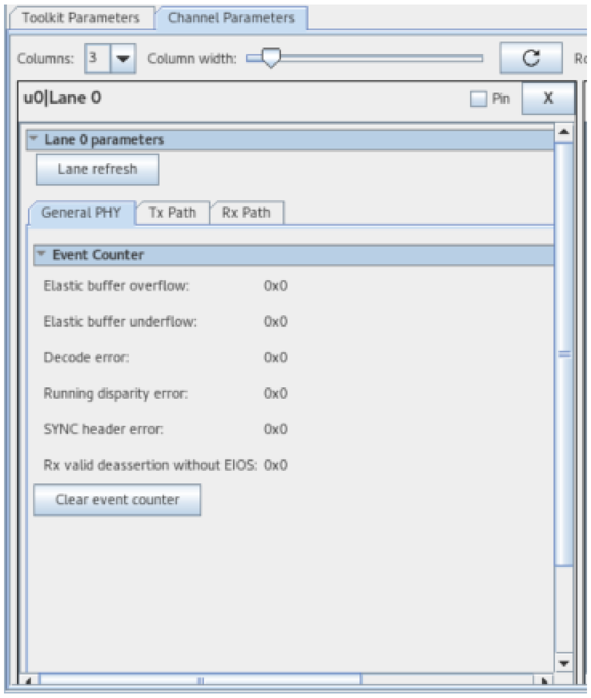


Columns: 3 Column width: Row height:

6.6.4.5.1. General PHY

This tab shows the event counters related to the PHY at each lane level.

Figure 71. General PHY Tab



Elastic buffer overflow: 0x0
Elastic buffer underflow: 0x0
Decode error: 0x0
Running disparity error: 0x0
SYNC header error: 0x0
Rx valid deassertion without EIOS: 0x0

6.6.4.5.2. Tx Path

This tab allows you to monitor the transmitter settings for the channel selected.

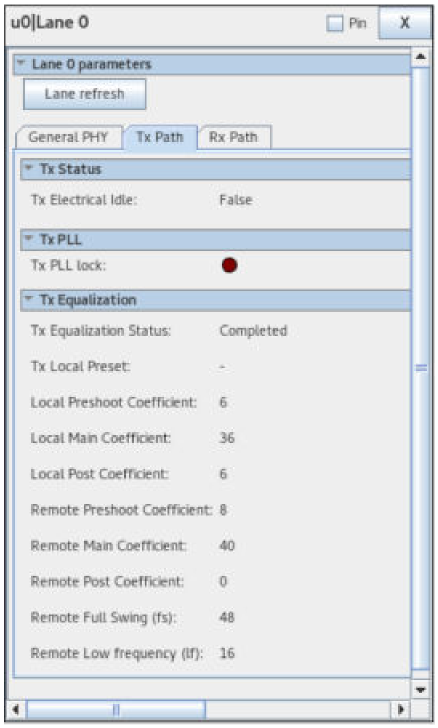
Table 114. Tx Path Parameters

Group	Parameters	Values	Descriptions
Tx Status	Tx Electrical Idle	True False	Indicates if TX is in electrical idle. True: indicates TX is in electrical idle. False: indicates TX is out of electrical idle.
	Tx PLL Lock	Green Red	Indicates if TX PLL is locked. *Note:* The status of the Tx PLL Lock may not be reflected properly by the Debug Toolkit in the Intel Quartus Prime software version 22.4. The fix for this problem is planned for a future release of the Debug Toolkit.
Tx Equalization	Tx Equalization Status	Not attempted Completed Unsuccessful	Indicates transmitter equalization status. The TX local and remote parameters are valid only when the value of the Equalization status is returned as completed, indicating equalization has completed successfully.
	Tx Local Preset	TBD	TBD

continued...

Group	Parameters	Values	Descriptions
	Local Preshoot Coefficient	Depends on the coefficient requested by the link partner.	Indicates transmitter driver output pre-emphasis (pre-cursor coefficient value).
	Local Main Coefficient		Indicates transmitter driver output pre-emphasis (main cursor coefficient value).
	Local Post Coefficient		Indicates transmitter driver output pre-emphasis (post-cursor coefficient value).
	Remote Preshoot Coefficient	Depends on the transmitter driver output of the link partner.	Indicates link partner's transmitter driver's output pre-cursor coefficient value
	Remote Main Coefficient		Indicates link partner's transmitter driver's output main coefficient value
	Remote Post Coefficient		Indicates link partner's transmitter driver's output post-cursor coefficient value
	Remote Full Swing (fs)	Depends on the device capability of the link partner.	Indicates the full swing value used by the link partner during the Equalization phase of link training.
	Remote Low Frequency (lf)		Indicates the low frequency value used by the link partner during the Equalization phase of link training.

Figure 72. Tx Path



Tx Electrical Idle: False

Tx PLL lock: ●

Tx Equalization Status: Completed
Tx Local Preset: -
Local Preshoot Coefficient: 6
Local Main Coefficient: 36
Local Post Coefficient: 6
Remote Preshoot Coefficient: 8
Remote Main Coefficient: 40
Remote Post Coefficient: 0
Remote Full Swing (fs): 48
Remote Low frequency (lf): 16

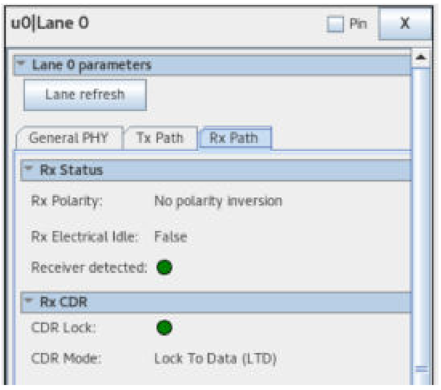
6.6.4.5.3. Rx Path

This tab allows you to monitor the receiver settings for the channel selected.

Table 115. Rx Path Parameters

Group	Parameters	Values	Descriptions
Rx Status	Rx Polarity	No polarity inversion Polarity inversion	Indicates RX polarity inversion for the selected lane.
	Rx Electrical Idle	True False	Indicates if RX is in electrical idle.
	Receiver Detected	True False	Indicates if far end RX is detected.
Rx CDR	CDR Lock	True False	Indicates the CDR lock state.
	CDR Mode	Locked to Reference (LTR) Locked to Data (LTD)	Indicates the CDR mode.

Figure 73. Rx Path



Rx Polarity: No polarity inversion
Rx Electrical Idle: False
Receiver detected: ●
CDR Lock: ●
CDR Mode: Lock To Data (LTD)

6.6.4.5.4. Lane Margining

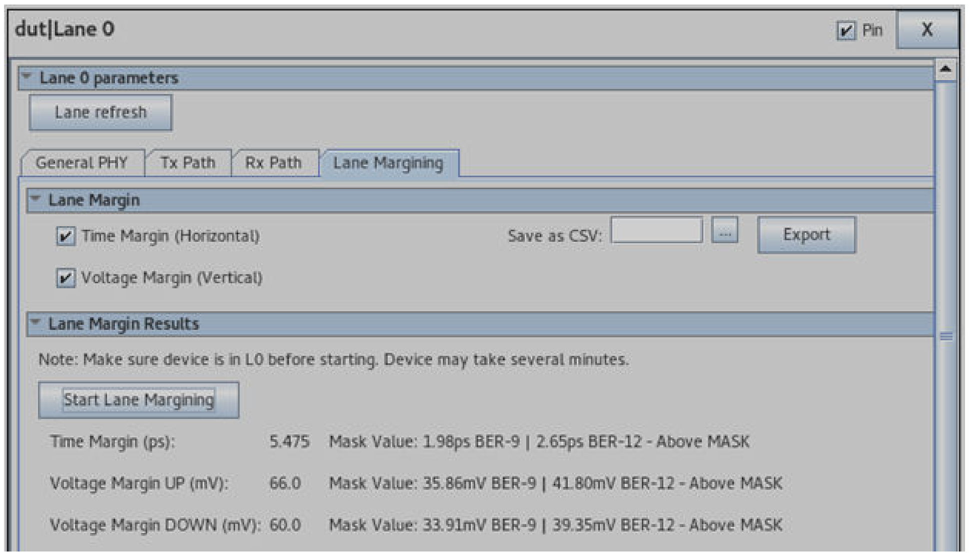
The R-Tile Debug Toolkit supports electrical lane margining that allows you to assess the electrical health for each channel. This feature allows you to:

- Assess the voltage margin (vertical) from sampling point to top and bottom.
- Assess the time margin (horizontal) from sampling point to left and right. However, please note that the R-Tile Debug Toolkit will only report the minor margin measured between the left margin and the right margin.
- Perform lane margining with the following configurations:
  - Configuration Mode 0 (1x16) and Configuration Mode 1 (2x8)
  - 8.0 GT/s (PCIe 3.0) at BER 10e-9
  - 16.0 GT/s (PCIe 3.0) at BER 10e-9
  - 32.0 GT/s (PCIe 3.0) at BER 10e-9
- Perform automatic comparison of lane margining results versus the recommended mask at BER 10e-9 and BER 10e-12. The lane margining exercise is performed at a BER of 10e-9. However, the tool automatically compares the results against the recommended mask for both BER 10e-9 and BER 10e-12.

Note: Intel recommends that the margin for each lane on your board be more than the mask in the horizontal and vertical directions to make sure that the channel is good and meets the PCIe specification. For more information on the Margin Masks, refer to Margin Masks for the R-Tile Avalon Streaming Intel FPGA IP for PCI Express.

Note: The R-Tile Avalon Streaming Intel FPGA IP for PCI Express Lane Margining feature of the Debug Toolkit does not support an independent error sampler for performing the lane margining. The lane margining is performed on the actual data path. As a result, the lane margining may produce uncorrectable errors in the data stream and cause the Link Training and Status State Machine (LTSSM) to go into the Recovery state. You may mask out all errors through the Advanced Error Reporting (AER) registers while performing lane margining, and reset all error counters, error registers, etc. after margining completes.

Figure 74. Lane Margining



Time Margin (Horizontal) ✓

Voltage Margin (Vertical) ✓

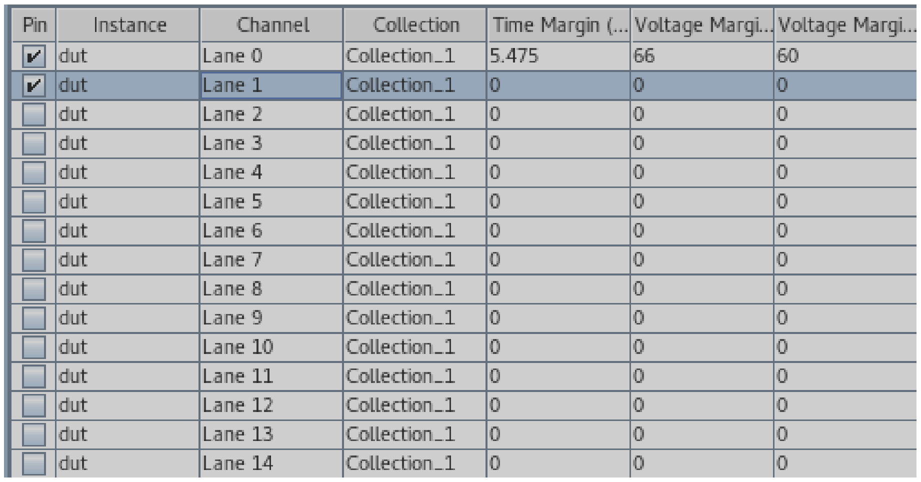
Save as CSV:

Note: Make sure device is in L0 before starting. Device may take several minutes.

```
Time Margin (ps):           5.475   Mask Value: 1.98ps BER-9 | 2.65ps BER-12 - Above MASK
Voltage Margin UP (mV):     66.0    Mask Value: 35.86mV BER-9 | 41.80mV BER-12 - Above MASK
Voltage Margin DOWN (mV):   60.0    Mask Value: 33.91mV BER-9 | 39.35mV BER-12 - Above MASK
```

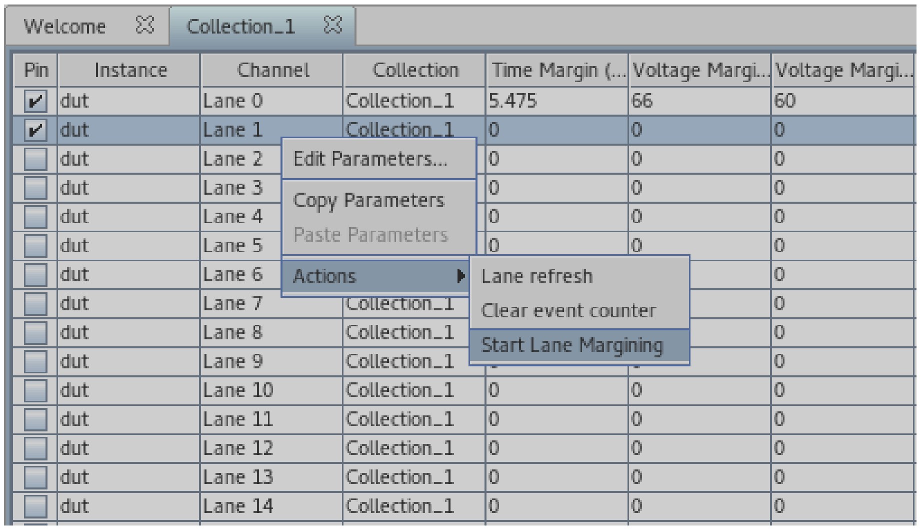
Execute the following procedure to perform lane margining for a given lane:

1. Select the targeted lane on the **Collection** tab.



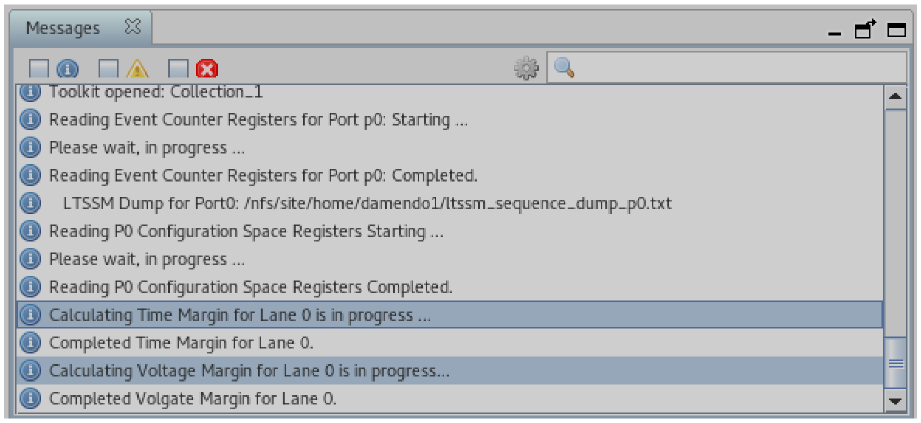
Pin	Instance	Channel	Collection	Time Margin (...	Voltage Margi...	Voltage Margi...
✓	dut	Lane 0	Collection_1	5.475	66	60
✓	dut	Lane 1	Collection_1	0	0	0
	dut	Lane 2	Collection_1	0	0	0
	dut	Lane 3	Collection_1	0	0	0
	dut	Lane 4	Collection_1	0	0	0
	dut	Lane 5	Collection_1	0	0	0
	dut	Lane 6	Collection_1	0	0	0
	dut	Lane 7	Collection_1	0	0	0
	dut	Lane 8	Collection_1	0	0	0
	dut	Lane 9	Collection_1	0	0	0
	dut	Lane 10	Collection_1	0	0	0
	dut	Lane 11	Collection_1	0	0	0
	dut	Lane 12	Collection_1	0	0	0
	dut	Lane 13	Collection_1	0	0	0
	dut	Lane 14	Collection_1	0	0	0

2. A new panel is displayed on the **Channel Parameters** tab. Select the **Lane Margining** sub-tab.
3. Under the **Lane Margin** section, select **Time Margin (Horizontal)** and/or **Voltage Margin (Vertical)**.
4. Under the **Lane Margining Results** section, click the **Start** button. Alternatively, you can right-click on the targeted lane and select **Start Lane Margining**.



Pin	Instance	Channel	Collection	Time Margin (...	Voltage Margi...	Voltage Margi...
✓	dut	Lane 0	Collection_1	5.475	66	60
✓	dut	Lane 1	Collection_1	0	0	0
	dut	Lane 2		0	0	0
	dut	Lane 3		0	0	0
	dut	Lane 4		0	0	0
	dut	Lane 5		0	0	0
	dut	Lane 6				0
	dut	Lane 7	Collection_1			0
	dut	Lane 8	Collection_1			0
	dut	Lane 9	Collection_1			0
	dut	Lane 10	Collection_1	0	0	0
	dut	Lane 11	Collection_1	0	0	0
	dut	Lane 12	Collection_1	0	0	0
	dut	Lane 13	Collection_1	0	0	0
	dut	Lane 14	Collection_1	0	0	0

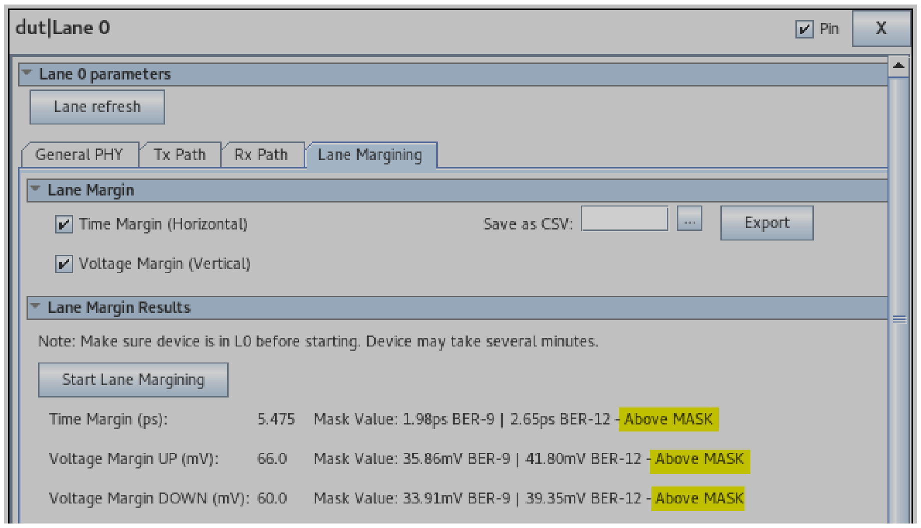
5. The lane margining may take several minutes to complete. Wait until the results are displayed.



```
Toolkit opened: Collection_1
Reading Event Counter Registers for Port p0: Starting ...
Please wait, in progress ...
Reading Event Counter Registers for Port p0: Completed.
   LTSSM Dump for Port0: /nfs/site/home/damendo1/ltssm_sequence_dump_p0.txt
Reading P0 Configuration Space Registers Starting ...
Please wait, in progress ...
Reading P0 Configuration Space Registers Completed.
Calculating Time Margin for Lane 0 is in progress ...
Completed Time Margin for Lane 0.
Calculating Voltage Margin for Lane 0 is in progress...
Completed Volgate Margin for Lane 0.
```

6. Once the lane margining results are available, the Debug Toolkit automatically compares them against the recommended Masks and provides a comparison result by printing **Above Mask** or **Below Mask**. This label appears next to the mask values. Refer to Margin Masks for the R-Tile Avalon Streaming Intel FPGA IP for PCI Express for details on the suggested methodology to evaluate your overall link margins.

   Note: The Debug Toolkit reports **Above Mask** if the obtained margin is above the BER 10e-9 mask but less than the BER 10e-12 mask.



☑ Time Margin (Horizontal)
☑ Voltage Margin (Vertical)

Note: Make sure device is in L0 before starting. Device may take several minutes.

```
Time Margin (ps):            5.475   Mask Value: 1.98ps BER-9 | 2.65ps BER-12 - Above MASK
Voltage Margin UP (mV):      66.0    Mask Value: 35.86mV BER-9 | 41.80mV BER-12 - Above MASK
Voltage Margin DOWN (mV):    60.0    Mask Value: 33.91mV BER-9 | 39.35mV BER-12 - Above MASK
```

7. R-Tile Avalon Streaming Intel FPGA IP for PCI Express User Guide Archives

For the latest and previous versions of these user guides, refer to the R-Tile Avalon Streaming Intel FPGA IP for PCI Express User Guide. If an IP or software version is not listed, the user guide for the previous IP or software version applies.

8. Document Revision History for the R-Tile Avalon Streaming Intel FPGA IP for PCI Express User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2023.06.26	23.2	10.0.0	• Updated the *Clocking* section to include connection requirements for the input clocks `refclk0`, `refclk1`, and `refclk2` for different Configuration Modes and different values of the **Enable Independent Perst Pins** parameter. • Added the *Independent Perst Pins* sub-section to the *Reset* section. • Added the requirement of using the Hard IP Reconfiguration Interface to access TLP Bypass registers to the *Register Settings for the TLP Bypass Mode* section. • Added some implementation considerations to the *Precision Time Measurement (PTM) Interface (Endpoint Only)* section. • Added the **Enable Debug Toolkit**, **Enable Warm Perst**, **Enable Independent GPIO Perst**, and **Enable Independent Perst Pins** parameters to the *Top-Level Settings* section. Updated corresponding screenshots of the IP Parameter Editor. • Updated IP Parameter Editor screenshots in the *Core Parameters* section.
2023.04.03	23.1	9.0.0	• Updated product family name to "Intel Agilex 7". • Updated all references to OPN numbers to differentiate between AGI OPNs and AGM OPNs. • Removed the section *Bias Temperature Instability (BTI) Protection Mode*. • Changed the section *PMA/PCS* to *PMA* since the PCS responsibilities are implemented by the MAC layer. Also changed all references to PCS to MAC, and some instances of PHY to PMA. • Updated the description of the `pX_tx_st_ready_o` signal in the *Avalon Streaming TX Interface* section. • Updated the guidelines in the *Application Logic Guidelines for the Avalon Streaming TX Interface* section. • Added example cases to the *Avalon Streaming TX Interface pX_tx_st_ready_o Behavior* section. • Added a Note on temperature monitoring to the *Hard IP Reconfiguration Interface* section.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Added a Note on temperature monitoring to the *Overview* section under the *Debug Toolkit* section. • Added a Note about the PHY Reconfiguration interface being automatically enabled to the *Enabling the R-Tile Debug Toolkit* section. • Added a Note about the availability of the **Channel Parameters** tab to the *Channel Parameters* section.
2022.12.19	22.4	8.0.0	• Added guidelines for the Application logic to the *Avalon Streaming RX Interface* and *Avalon Streaming TX Interface* sections. • Moved the **Enable the PHY Reconfiguration Interface** from the *Top-Level Settings* section to the *Avalon Parameters* section to match the 22.4 IP Parameter Editor. • Updated screenshots in the *Core Parameters* section to match the 22.4 IP Parameter Editor. • Added the *Link Inspector* and the *Channel Parameters* sections.
2022.10.07	22.3	7.0.0	Updated the links to the configuration space register maps in the *Configuration Space Registers* section.
2022.09.26	22.3	7.0.0	• Updated the information in the table *Intel Agilex Recommended FPGA Fabric Speed Grades for All Avalon Streaming Widths and Frequencies.* • Added guidelines that the Application logic must follow to the *Avalon Streaming RX Interface* section. • Added guidelines that the Application logic must follow to the *Avalon Streaming TX Interface* section. • Updated the *Troubleshooting/Debugging* chapter to: — Add the information regarding hardware debugging, specifically how to debug link training issues, device enumeration issues, and performance and data transfer issues. — Add the list of top-level signals that can be monitored via the Signal Tap Logic Analyzer. — Add the list of interfaces that can be used for debug activities.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
2022.06.20	22.2	6.0.0	• Updated the *Debug Toolkit* section to document the additional capabilities for users to verify in hardware the status of the R-Tile Avalon Streaming Intel FPGA IP for PCI Express. • Documented the new ports `pX_cold_perst_n_i`, `pX_warm_perst_n_i`, and `pX_ip_rst_n_o`. • Updated the *Independent PERST* section with new timing diagrams showing the behaviors of the `pX_cold_perst_n_i` and `pX_warm_perst_n_i` signals. • Updated the *Hard IP Reconfiguration Interface* section with a new timing diagram showing the behavior of the signals on this interface while doing a read of the configuration space registers.
2022.03.28	22.1	5.0.0	• Removed Questa* FPGA Edition from the list of supported simulators in the *Features* section. • Updated the FPGA fabric speed grades table in the *Performance and Resource Utilization* section. • Updated the application clock frequencies table in the *Clocking* section. • Updated the list of PCI Express Capability Structure registers that need to be implemented in the Application logic in the *Hard IP Reconfiguration Interface*. • Updated the tables in the *Completion Buffer Size* section. • Added the subsections *D3Hot Exit Initiated by Host*, *D3Hot Exit Initiated by EP*, *D3Cold Entry* and *D3Cold Exit* to the *Power Management Interface* section. • Modified the list of steps given in the *PIPE Direct Reset Sequence* section. • Modified the description given in the *PIPE Direct Speed Change* section.
2021.12.13	21.4	4.0.0	• Added the new section *BTI Protection Mode*. • Updated the *Completion Buffer Size* section to show the correct buffer sizes. Also added examples showing the amount of Completion buffer entries consumed for Memory Read requests. Finally, added a suggested flow for the Application logic to track the completion buffer entries and based on this, schedule Non-Posted (NP) requests to the R-Tile Avalon Streaming Intel FPGA IP for PCI Express. • Updated the text descriptions and timing waveforms for the *Avalon Streaming RX Interface* and the *Avalon Streaming TX Interface* to show how the user application logic can properly use these interfaces. • Updated the signal descriptions and timing waveforms for the *Deskew Channel* section to show how the user application logic can properly use this interface.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
2021.10.06	21.3	3.0.0	Removed the section *ECRC* due to missing information on the register offsets for the ECRC or LCRC counters.
2021.10.04	21.3	3.0.0	• Updated the block diagrams in the *PCI Express Mode* and *PIPE Direct Mode* sections to match the interface signals on the 21.3 block symbol for the R-Tile Avalon Streaming Intel FPGA IP for PCI Express. • Added a Note to the *Avalon Streaming TX Interface* section stating that for this interface, the SOP can only be sent on segments 0 and 2. • Added the *Root Port Enumeration* Appendix chapter.
2021.07.12	21.2	2.0.0	Initial release.

A. Configuration Space Registers

To access the configuration space registers of the PFs and VFs in the R-Tile Avalon Streaming Intel FPGA IP for PCI Express, drive the base addresses for the appropriate PFs/VFs using the `pX_hip_reconfig_address_i[31:0]` bus in the Hard IP Reconfiguration Interface. For more details, refer to Hard IP Reconfiguration Interface on page 103.

The register maps for the R-Tile Avalon Streaming Intel FPGA IP for PCI Express Revision A in X16, X8 and X4 modes can be found at R-Tile Avalon Streaming Intel FPGA IP for PCI Express Revision A Configuration Register Maps.

The register maps for the R-Tile Avalon Streaming Intel FPGA IP for PCI Express Revision B in X16, X8 and X4 modes can be found at R-Tile Avalon Streaming Intel FPGA IP for PCI Express Revision B Configuration Register Maps.

For more details, refer to PCI Express Base Specification Revision 5.0, Version 1.0.

B. Root Port Enumeration

This chapter provides a flow chart that explains the Root Port enumeration process.

The goal of enumeration is to find all connected devices in the system and for each connected device, set the necessary registers and make address range assignments.

At the end of the enumeration process, the Root Port (RP) must set the following registers:

- Primary Bus, Secondary Bus and Subordinate Bus numbers
- Memory Base and Limit
- IO Base and IO Limit
- Max Payload Size
- Memory Space Enable bit

The Endpoint (EP) must also have the following registers set by the RP:

- Master Enable bit
- BAR Address
- Max Payload Size
- Memory Space Enable bit
- Severity bit

The figure below shows an example tree of connected devices on which the following flow chart will be based.

Figure 75. Tree of Connected Devices in Example System

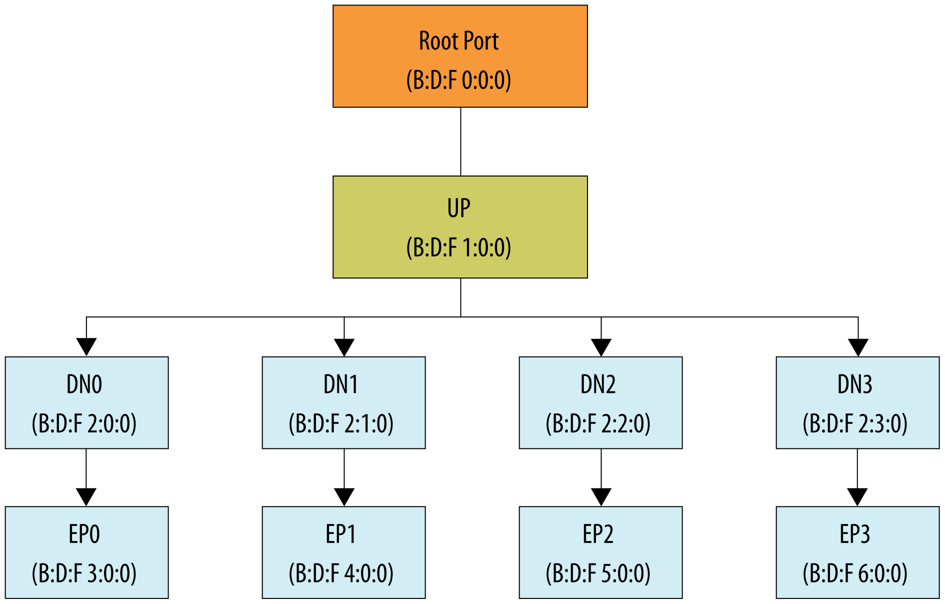


Figure 76. Root Port Enumeration Flow Chart

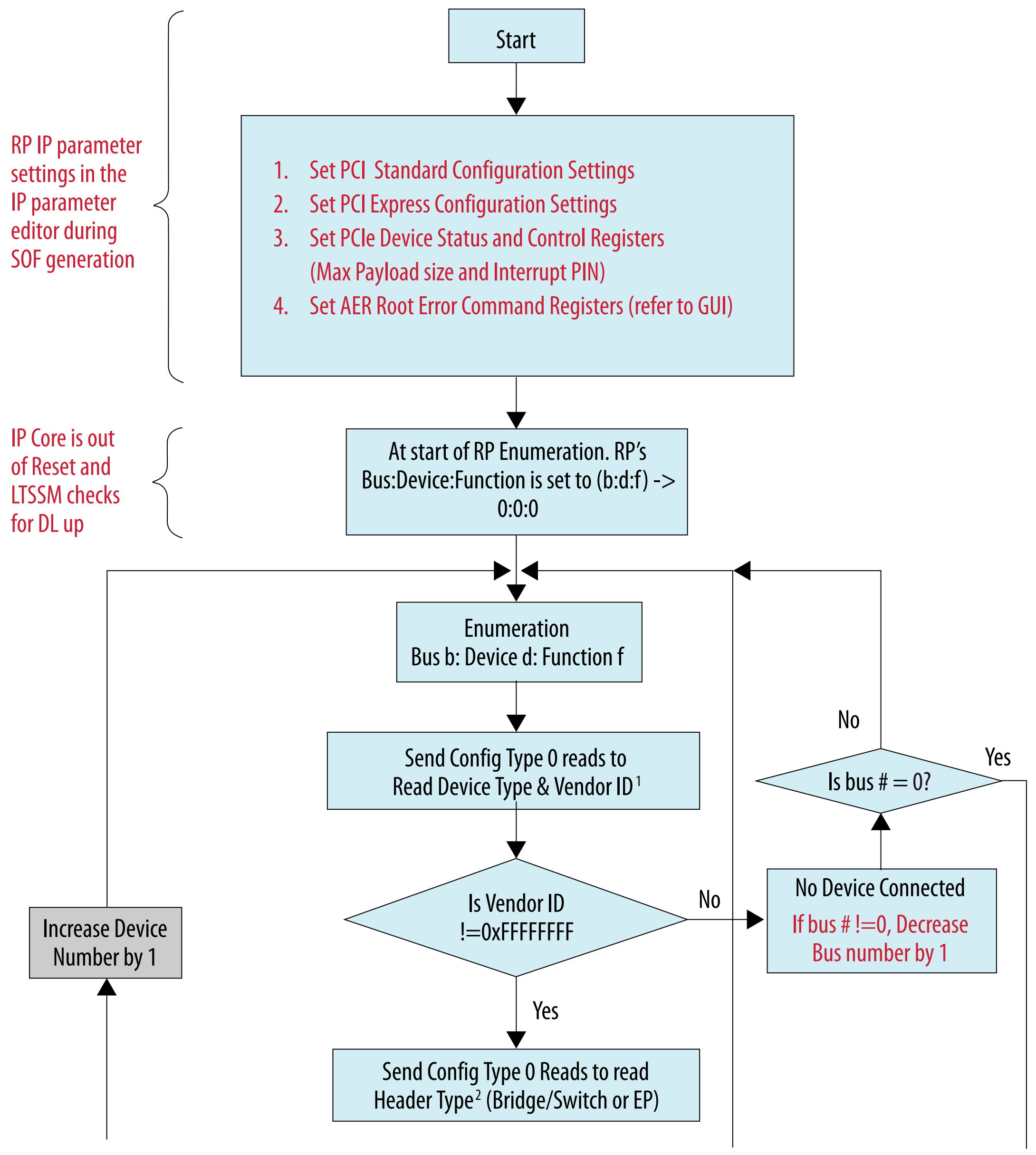

Start
1. Set PCI Standard Configuration Settings
2. Set PCI Express Configuration Settings
3. Set PCIe Device Status and Control Registers (Max Payload size and Interrupt PIN)
4. Set AER Root Error Command Registers (refer to GUI)
RP IP parameter settings in the IP parameter editor during SOF generation
At start of RP Enumeration. RP's Bus:Device:Function is set to (b:d:f) -> 0:0:0
IP Core is out of Reset and LTSSM checks for DL up
Enumeration Bus b: Device d: Function f
Send Config Type 0 reads to Read Device Type & Vendor ID 1
Is Vendor ID !=0xFFFFFFFF
No
Yes
No Device Connected
If bus # !=0, Decrease Bus number by 1
Is bus # = 0?
No
Yes
Increase Device Number by 1
Send Config Type 0 Reads to read Header Type 2 (Bridge/Switch or EP)

Figure 77. Root Port Enumeration Flow Chart (continued)

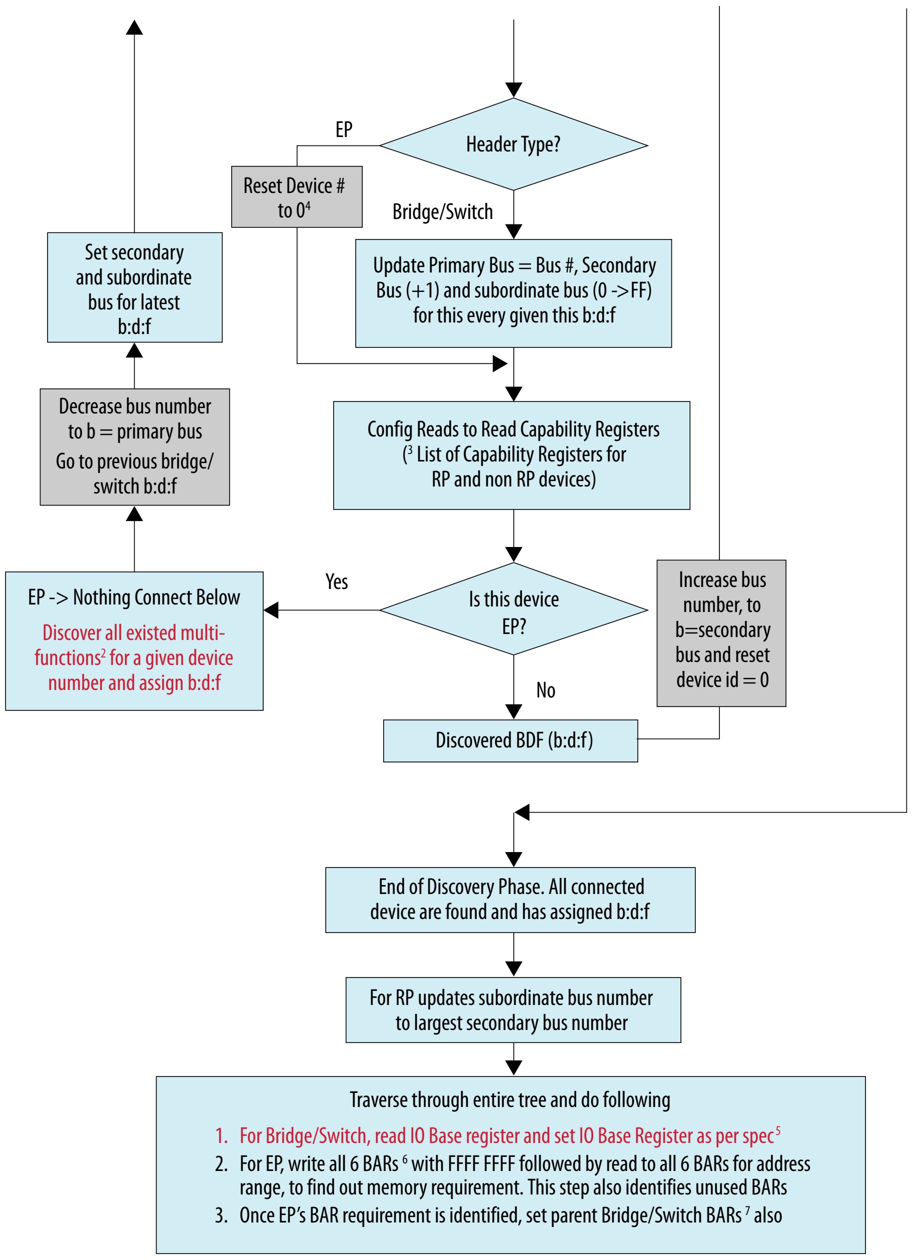
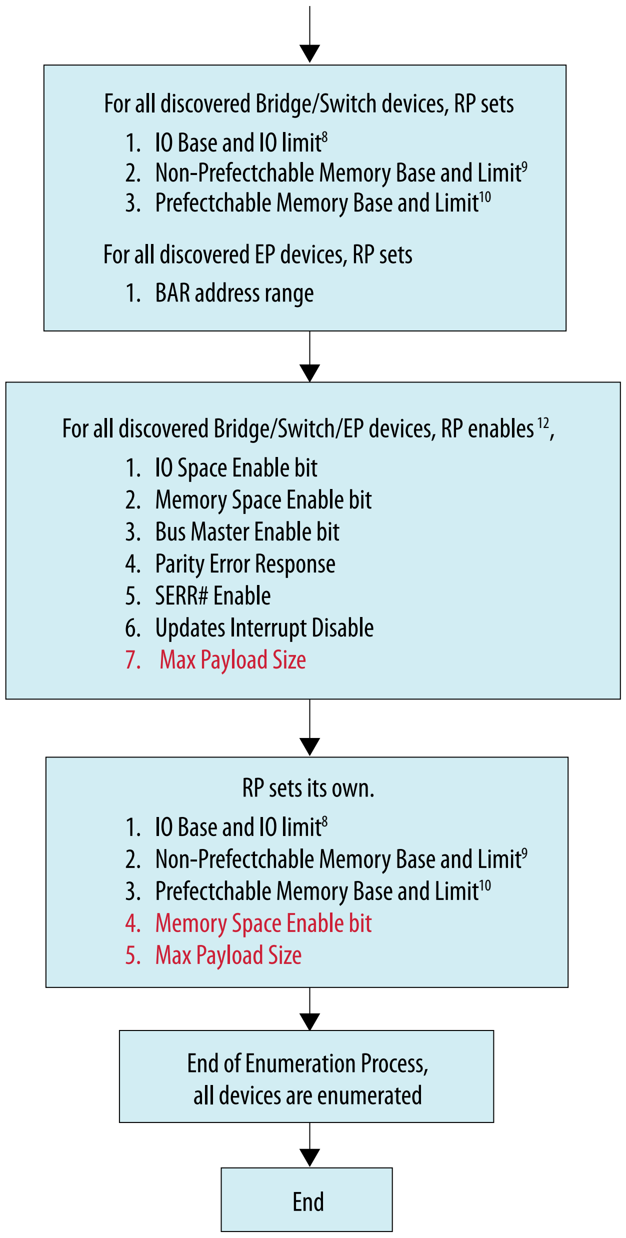

Header Type?
EP
Reset Device # to 0[4]
Bridge/Switch
Update Primary Bus = Bus #, Secondary Bus (+1) and subordinate bus (0 ->FF) for this every given this b:d:f
Set secondary and subordinate bus for latest b:d:f
Decrease bus number to b = primary bus Go to previous bridge/ switch b:d:f
Config Reads to Read Capability Registers ([3] List of Capability Registers for RP and non RP devices)
EP -> Nothing Connect Below
Discover all existed multi-functions[2] for a given device number and assign b:d:f
Yes
Is this device EP?
No
Increase bus number, to b=secondary bus and reset device id = 0
Discovered BDF (b:d:f)
End of Discovery Phase. All connected device are found and has assigned b:d:f
For RP updates subordinate bus number to largest secondary bus number
Traverse through entire tree and do following
1. For Bridge/Switch, read IO Base register and set IO Base Register as per spec[5]
2. For EP, write all 6 BARs[6] with FFFF FFFF followed by read to all 6 BARs for address range, to find out memory requirement. This step also identifies unused BARs
3. Once EP's BAR requirement is identified, set parent Bridge/Switch BARs[7] also

Figure 78. Root Port Enumeration Flow Chart (continued)


For all discovered Bridge/Switch devices, RP sets
1. IO Base and IO limit[8]
2. Non-Prefectchable Memory Base and Limit[9]
3. Prefectchable Memory Base and Limit[10]
For all discovered EP devices, RP sets
1. BAR address range
For all discovered Bridge/Switch/EP devices, RP enables [12],
1. IO Space Enable bit
2. Memory Space Enable bit
3. Bus Master Enable bit
4. Parity Error Response
5. SERR# Enable
6. Updates Interrupt Disable
7. Max Payload Size
RP sets its own.
1. IO Base and IO limit[8]
2. Non-Prefectchable Memory Base and Limit[9]
3. Prefectchable Memory Base and Limit[10]
4. Memory Space Enable bit
5. Max Payload Size
End of Enumeration Process, all devices are enumerated
End

Notes:

1. Vendor ID and Device ID information is located at offset 0x00h for both Header Type 0 and Header Type 1.
2. For PCIe Gen4, the Header Type is located at offset 0x0Eh (2nd DW). If bit 0 is set to 1, it indicates the device is a Bridge; otherwise, it is an EP. If bit 7 is set to 0, it indicates this is a single-function device; otherwise, it is a multi-function device.
3. List of capability registers for RP and non-RP devices:

Send Feedback

- 0x34h – Capabilities Pointers. This register is used to point to a linked list of capabilities implemented by a Function:
    a. Capabilities Pointer for RP
        i. Address 40 - Identifies the Power Management Capability ID
        ii. Address 50 - Identifies MSI Capability ID
        iii. Address 70 - Identifies the PCI Express Capability structure
    b. Capabilities Pointer for non-RP
        i. Address 40 - Identifies Power Management Capability ID
        ii. Address 48 - Identifies the PCI Express Capability structure

4. EP does not have an associated register of Primary, Secondary and Subordinate Bus numbers.
5. Bridge/Switch IO Base and Limit register offset 0x1Ch. These registers are set per the PCIe 4.0 Base Specification. For more accurate information and flow, refer to chapter 7.5.1.3.6 of the Base Specification.
6. For EP Type 0 header, BAR addresses are located at the following offsets:
    a. 0x10h – Base Address 0
    b. 0x14h – Base Address 1
    c. 0x18h – Base Address 2
    d. 0x1ch – Base Address 3
    e. 0x20h – Base Address 4
    f. 0x24h – Base Address 5
7. For Bridge/Switch Type 1 header, BAR addresses are located at the following offsets:
    a. 0x10h – Base Address 0
    b. 0x14h – Base Address 1
8. For Bridge/Switch Type 1 header, IO Base and IO limit registers are located at offset 0x1Ch.
9. For Bridge/Switch Type 1 header, Non-Prefetchable Memory Base and Limit registers are located at offset 0x20h.
10. For Bridge/Switch Type 1 header, Prefetchable Memory Base and Limit registers are located at offset 0x24h.
11. For Bridge/Switch/EP Type 0 & 1 headers, the Bus Master Enable bit is located at offset 0x04h (Command Register) bit 2.
12. For Bridge/Switch/EP Type 0 & 1 headers,
    a. IO Space Enable bit is located at offset 0x04h (Command Register) bit 0.
    b. Memory Space Enable bit is located at offset 0x04h (Command Register) bit 1.
    c. Bus Master Enable bit is located at offset 0x04h (Command Register) bit 2.
    d. Parity Error Response bit is located at offset 0x04h (Command Register) bit 6.
    e. SERR# Enable bit is located at offset 0x04h (Command Register) bit 8.
    f. Interrupt Disable bit is located at offset 0x04h (Command Register) bit 10.

C. Implementation of Address Translation Services (ATS) in Endpoint Mode

With the R-Tile Avalon streaming Intel FPGA IP for PCIe:

- ATS messages/completions are sent and received through the Avalon streaming interface.
- Address translation caches (ATC) need to be implemented in the user logic. There must be a separate ATC for each VF/PF that supports ATS.

Refer to the *Address Translation Services Revision 1.1* specification, section 4.1 *Page Request Message* for more details.

C.1. Sending Translated/Untranslated Requests

The function with an ATC can send Memory Read requests that contain either translated or untranslated addresses. If the Endpoint wants to receive the associated translated address to update the ATC, it will generate a Memory Read with the "Translation Request" field set.

The Endpoint will receive the translated address in the associated Completion with the ATS field's S/N/G/P/E/U/W/R values.

C.2. Sending a Page Request Message from the Endpoint (EP) to the Root Complex (RC)

The user application issues a Page Request Message while using the Avalon-ST interface to send the contents of the ATS message.

The RC will respond with PRG Response message(s).

The EP will update its ATC accordingly.

C.3. Invalidating Requests/Completions

Invalidation is done via the Message mechanism:

- When the EP receives an Invalidate request from the RP, it needs to clear the associated ATC.
- When the ATC is cleared, the user application generates a Completion message.

D. Packets Forwarded to the User Application in TLP Bypass Mode

In TLP Bypass mode, the R-Tile IP for PCIe forwards TLPs to the Avalon-ST RX interface except for malformed TLPs. The following tables describe how the IP handles each TLP type for upstream and downstream.

D.1. EP TLP Bypass Mode (Upstream)

Table 116. Packets Forwarded in EP TLP Bypass Mode

TLP Type	Routing	Direction	TLP Corruption	Forwarded to Avalon-ST Interface
ASSERT/DEASSERT INTx	Local	Upstream	None	No
			Ecrc_err	No
			Malformed	No
VENDOR_MESSAGE_0 /1	Route_to_RC	Upstream	None	No (VENDOR0) Yes (VENDOR1)
			Poisoned	No (VENDOR0) Yes (VENDOR1)
			Ecrc_err	Yes
			Malformed	No
VENDOR_MESSAGE_0 /1	Route_by_ID	Both	None	Yes
			ID_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
VENDOR_MESSAGE_0 /1	Broadcast	Downstream	None	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
VENDOR_MESSAGE_0 /1	Local	Both	None	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
PM_ACTIVE_STATE_NAK	Local	Downstream	None	Yes

continued...

TLP Type	Routing	Direction	TLP Corruption	Forwarded to Avalon-ST Interface
			Ecrc_err	Yes
			Malformed	No
PM_PME	Route_to_RC	Upstream	None	No
			Ecrc_err	Yes
			Malformed	No
PME_TURN_OFF	Broadcast	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
PME_TO_ACK	Gather	Upstream	None	No
			Ecrc_err	Yes
			Malformed	No
ERR_COR	Route_to_RC	Upstream	None	No
			Ecrc_err	Yes
			Malformed	No
ERR_NONFATAL	Route_to_RC	Upstream	None	No
			Ecrc_err	Yes
			Malformed	No
ERR_FATAL	Route_to_RC	Upstream	None	No
			Ecrc_err	Yes
			Malformed	No
UNLOCK	Broadcast	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
SET_SLOT_POWER_LIMIT	Local	Downstream	None	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
LN_MESSAGE	Route_by_ID	Both	None	Yes
			ID_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
LN_MESSAGE	Broadcast	Downstream	None	Yes
			Poisoned	Yes
			Ecrc_err	Yes

continued...

TLP Type	Routing	Direction	TLP Corruption	Forwarded to Avalon-ST Interface
			Malformed	No
DRS_MESSAGE	Local	Upstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
FRS_MESSAGE	Route_to_RC	Upstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
HIERARCHY_ID_MSG	Broadcast	Downstream	None	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
IGNORED_MSG_ATT_ ON	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
IGNORED_MSG_ATT_ BLINK	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
IGNORED_MSG_ATT_ OFF	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
IGNORED_MSG_IND_ ON	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
IGNORED_MSG_IND_ BLINK	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
IGNORED_MSG_IND_ OFF	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
IGNORED_MSG_ATT_ BT_PRESS	Local	Upstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
LTR_MESSAGE	Local	Upstream	None	No
			Poisoned	No
			Ecrc_err	Yes

continued...

TLP Type	Routing	Direction	TLP Corruption	Forwarded to Avalon-ST Interface
			Malformed	No
OBFF_MESSAGE	Local	Downstream	None	No
			Poisoned	No
			Ecrc_err	Yes
			Malformed	No
PTM_REQUEST	Local	Upstream	None	No
			Ecrc_err	Yes
			Malformed	No
PTM_RESPONSE	Local	Downstream	None	No
			Poisoned	No
			Ecrc_err	Yes
			Malformed	No
PTM_RESPONSE_D	Local	Downstream	None	No
			Poisoned	No
			Ecrc_err	Yes
			Malformed	No
INVALIDATE_REQUEST	Route_by_ID	Both	None	Yes
			ID_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
INVALIDATE_COMPLETION	Route_by_ID	Both	None	Yes
			ID_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
CFG_WR_0	Route_by_ID	Downstream	None	Yes
			ID_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
CFG_WR_1	Route_by_ID	Downstream	None	Yes
			ID_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes

continued...

TLP Type	Routing	Direction	TLP Corruption	Forwarded to Avalon-ST Interface
			Malformed	No
CFG_RD_0	Route_by_ID	Downstream	None	Yes
			ID_mismatch	Yes
			Ecrc_err	Yes
			Malformed	No
CFG_RD_1	Route_by_ID	Downstream	None	Yes
			ID_mismatch	Yes
			Ecrc_err	Yes
			Malformed	No
IO_WR	Address	Downstream	None	Yes
			Addr_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
IO_RD	Address	Downstream	None	Yes
			Addr_mismatch	Yes
			Ecrc_err	Yes
			Malformed	No
MEM_WR_32/64	Address	Both	None	Yes
			Addr_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
MEM_RD_32/64	Address	Both	None	Yes
			Addr_mismatch	Yes
			Ecrc_err	Yes
			Malformed	No
MEM_RD_LK_32/64	Address	Both	None	Yes
			Addr_mismatch	Yes
			Ecrc_err	Yes
			Malformed	No
ATOMIC_FETCH_ADD_32/64	Address	Both	None	Yes
			Addr_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes

continued...

TLP Type	Routing	Direction	TLP Corruption	Forwarded to Avalon-ST Interface
			Malformed	No
ATOMIC_SWAP_32/64	Address	Both	None	Yes
			Addr_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
ATOMIC_CAS_32/64/128	Address	Both	None	Yes
			Addr_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	32/64: No 128: No stimulus
CPL	Route_by_ID	Both	None	Yes
			ID_mismatch	Yes
			LUT_mismatch	Yes
			Ecrc_err	Yes
			Malformed	No
			CA_status	Yes
			UR_status	Yes
			CRS_status	Yes
CPLD	Route_by_ID	Both	None	Yes
			ID_mismatch	Yes
			LUT_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No

D.2. RC TLP Bypass Mode (Downstream)

Table 117. Packets Forwarded in RC TLP Bypass Mode

TLP Type	Routing	Direction	TLP Corruption	Forwarded to Avalon-ST Interface
ASSERT/DEASSERT INTx	Local	Upstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
VENDOR_MESSAGE_0/1	Route_to_RC	Upstream	None	Yes
			Poisoned	Yes

continued...

Send Feedback

TLP Type	Routing	Direction	TLP Corruption	Forwarded to Avalon-ST Interface
			Ecrc_err	Yes
			Malformed	No
VENDOR_MESSAGE_0/1	Route_by_ID	Both	None	Yes
			ID_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
VENDOR_MESSAGE_0/1	Broadcast	Downstream	None	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
VENDOR_MESSAGE_0/1	Local	Both	None	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
PM_ACTIVE_STATE_NAK	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
PM_PME	Route_to_RC	Upstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
PME_TURN_OFF	Broadcast	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
PME_TO_ACK	Gather	Upstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
ERR_COR	Route_to_RC	Upstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
ERR_NONFATAL	Route_to_RC	Upstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
ERR_FATAL	Route_to_RC	Upstream	None	Yes
			Ecrc_err	Yes

continued...

TLP Type	Routing	Direction	TLP Corruption	Forwarded to Avalon-ST Interface
			Malformed	No
UNLOCK	Broadcast	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
SET_SLOT_POWER_LIMIT	Local	Downstream	None	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
LN_MESSAGE	Route_by_ID	Both	None	Yes
			ID_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
LN_MESSAGE	Broadcast	Downstream	None	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
DRS_MESSAGE	Local	Upstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
FRS_MESSAGE	Route_to_RC	Upstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
HIERARCHY_ID_MSG	Broadcast	Downstream	None	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
IGNORED_MSG_ATT_ON	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
IGNORED_MSG_ATT_BLINK	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
IGNORED_MSG_ATT_OFF	Local	Downstream	None	Yes
			Ecrc_err	Yes

continued...

TLP Type	Routing	Direction	TLP Corruption	Forwarded to Avalon-ST Interface
			Malformed	No
IGNORED_MSG_IND_ON	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
IGNORED_MSG_IND_BLINK	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
IGNORED_MSG_IND_OFF	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
IGNORED_MSG_ATT_BT_PRESS	Local	Upstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
LTR_MESSAGE	Local	Upstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
OBFF_MESSAGE	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
PTM_REQUEST	Local	Upstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
PTM_RESPONSE	Local	Downstream	None	Yes
			Ecrc_err	Yes
			Malformed	No
PTM_RESPONSE_D	Local	Downstream	None	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
INVALIDATE_REQUEST	Route_by_ID	Both	None	Yes
			ID_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
INVALIDATE_COMPLETION	Route_by_ID	Both	None	Yes

continued...

TLP Type	Routing	Direction	TLP Corruption	Forwarded to Avalon-ST Interface
			ID_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
CFG_WR_0	Route_by_ID	Downstream	None	Yes
			ID_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
CFG_WR_1	Route_by_ID	Downstream	None	Yes
			ID_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
CFG_RD_0	Route_by_ID	Downstream	None	Yes
			ID_mismatch	Yes
			Ecrc_err	Yes
			Malformed	No
CFG_RD_1	Route_by_ID	Downstream	None	Yes
			ID_mismatch	Yes
			Ecrc_err	Yes
			Malformed	No
IO_WR	Address	Downstream	None	Yes
			Addr_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
IO_RD	Address	Downstream	None	Yes
			Addr_mismatch	Yes
			Ecrc_err	Yes
			Malformed	No
MEM_WR_32/64	Address	Both	None	Yes
			Addr_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes

continued...

Send Feedback

TLP Type	Routing	Direction	TLP Corruption	Forwarded to Avalon-ST Interface
			Malformed	No
MEM_RD_32/64	Address	Both	None	Yes
			Addr_mismatch	Yes
			Ecrc_err	Yes
			Malformed	No
ATOMIC_FETCH_ADD_32/64	Address	Both	None	Yes
			Addr_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
ATOMIC_SWAP_32/64	Address	Both	None	Yes
			Addr_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
ATOMIC_CAS_32/64/128	Address	Both	None	Yes
			Addr_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No
CPL	Route_by_ID	Both	None	Yes
			ID_mismatch	Yes
			LUT_mismatch	Yes
			Ecrc_err	Yes
			Malformed	No
			CA_status	Yes
			UR_status	Yes
			CRS_status	Yes
CPLD	Route_by_ID	Both	None	Yes
			ID_mismatch	Yes
			LUT_mismatch	Yes
			Poisoned	Yes
			Ecrc_err	Yes
			Malformed	No

E. Margin Masks for the R-Tile Avalon Streaming Intel FPGA IP for PCI Express

E.1. Margin Masks Overview

The official margin mask is provided in the following table for reference. These margin masks provide a risk assessment for the FPGA PCIe interfaces on Intel R-Tile Avalon Streaming designs.

Table 118. Margin Mask Values for the R-Tile Avalon Streaming Intel FPGA IP for PCI Express

Link Speed	Bit Error Rate	Number of Parts x Number of Repetitions	Minimum Time Margin (ps)	Minimum Voltage Margin Up (mV)	Minimum Voltage Margin Down (mV)
PCIe 3.0	1e-9	5x5	8.44	90.37	89.93
	1e-12	5x5	9.94	91.83	93.22
PCIe 4.0	1e-9	5x5	5.52	47.09	47.15
	1e-12	5x5	7.01	51.34	51.89
PCIe 5.0	1e-9	5x5	1.98	35.86	33.91
	1e-12	5x5	2.65	41.8	39.35

Note: For the **Time Margin** (horizontal), the R-Tile Debug Toolkit only reports the minor margin measured between the left margin and the right margin.

E.2. 5x5 Testing Efficiency

The purpose of repeat testing is to quantify Device Under Test (DUT)-to-DUT board copy variation and run-to-run variation. Choosing two or more random boards and sets of silicon can give a good statistical assessment of DUT-to-DUT variation. Performing five power-cycled margin runs gives a good average of the boot-to-boot variation to fit to the statistical model. The boot-to-boot variation is why it is important to force the Physical (PHY) Layer to retrain between tests. System cold reboot is used to force the PHY Layer to retrain and is recommended as part of a margining flow. Intel recommends a 5x5 (five test boards with five power-cycled margin runs on each) data sample for validating PCIe links.

E.2.1. First Time Testing

When the board is being validated for the first time, Intel recommends testing all PCIe links with a 5x5 sample set. A 2x5 sample set is an acceptable minimum sample size if a five-board set is not available, but Intel further suggests additional testing as time permits and as other boards become available. All data from this testing should be saved for future comparison.

E.2.2. Spot Check Testing

Intel suggests you to also perform a spot check of board performance as new silicon or a new Quartus build is released. Once the board has been completely tested, when new silicon or a new Intel Quartus Prime build becomes available, Intel suggests performing a 1x3 test of the PCIe links and comparing the average results against previous results. Margin results are expected to be better or similar to previous results. If margins degrade, additional testing is recommended.

Ultimately, use your engineering judgment to determine how often spot checking should be done.

E.3. Debug Toolkit Margin Methodology

E.3.1. Debug Toolkit Testing Process

As previously mentioned, Intel recommends you to perform the procedure detailed in 5x5 Testing Efficiency, where you test five different boards with five power-cycled runs on each part or board. This is the recommended process:

1. Power on the board with the link under test at the L0 LTSSM state and at the desired PCIe Speed with no recoveries on the PCIe link.
2. Load the Debug Toolkit and run the Lane Margining tool to get the margins on all PCIe lanes and store the data samples.
3. Apply a power cycle to the board and repeat steps 1 and 2 five times once the PCIe link under test is properly trained.
4. Perform steps 1 to 3 for each test board (in this case five different boards or parts).
5. Finally, you must get an average of the collected Time Margin, Voltage Margin Up and Voltage Margin Down values to compare with Margin Mask values provided in Margin Mask Values for the R-Tile Avalon Streaming Intel FPGA IP for PCI Express. If any average value of the lane margins is greater than or equal to the Margin Mask value, it means that the test values are good and meet the minimum values required. Otherwise, if the value of the average lane margins is less than the Margin Mask value, it means that there are more CRC errors than expected and the link has not been trained correctly.