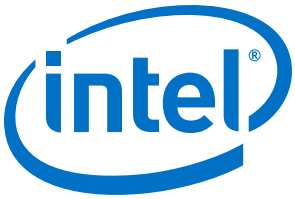
Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



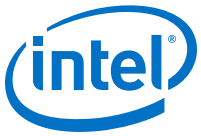
Intel® Arria® 10 Avalon®-ST Interface with SR-IOV PCIe* Solutions User Guide

Updated for Intel® Quartus® Prime Design Suite: **17.1**

Subscribe
Send Feedback

UG-01161 | 2018.08.13
Latest document on the web: **PDF** | **HTML**

Contents



1. Datasheet

1.1. Intel® Intel® Arria® 10 Avalon-ST Interface with SR-IOV for PCIe* Datasheet

Intel® Intel Arria® 10 FPGAs include a configurable, hardened protocol stack for PCI Express* that is compliant with *PCI Express Base Specification 2.1 or 3.0*. The Intel Arria 10 Hard IP for PCI Express with Single Root I/O Virtualization (SR-IOV) IP core consists of this hardened protocol stack and the SR-IOV soft logic. The SR-IOV soft logic uses the Configuration Space Bypass mode of the Hard IP to bypass the internal configuration block and BAR matching logic. These functions are implemented in external soft logic. Soft logic in the SR-IOV Bridge also implements interrupts and error reporting.

In 16.0, SR-IOV Bridge was redesigned to support up to 4 Physical Functions (PFs) and 2048 Virtual Functions (VFs). The SR-IOV bridge also supports the Address Translation Services (ATS) and TLP Processing Hints (TPH) capabilities.

Figure 1. Intel Arria 10 PCIe Variant with SR-IOV

The following figure shows the high-level modules and connecting interfaces for this variant.

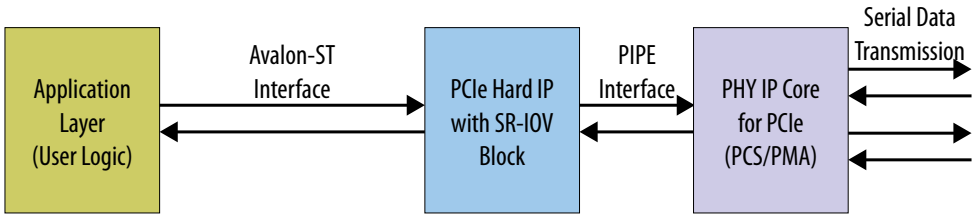


Figure 2. Intel Arria 10 PCIe Variant with SR-IOV

The following figure provides the next level of detail for the modules that comprise the SR-IOV Bridge.

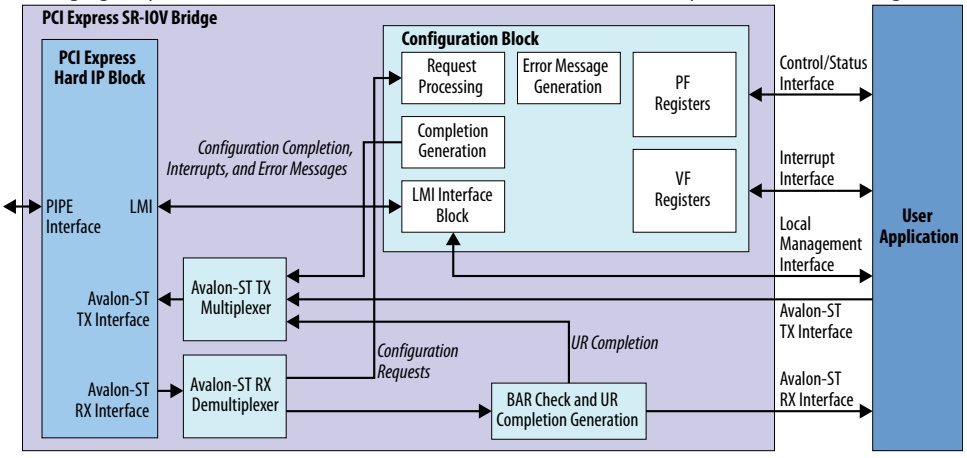


Table 2. PCI Express Data Throughput

The following table shows the aggregate bandwidth of a PCI Express link for Gen2 x4. The protocol specifies 2.5 giga-transfers per second for Gen1, 5.0 giga-transfers per second for Gen2. This table provides bandwidths for a single transmit (TX) or receive (RX) channel. The numbers double for duplex operation. Gen1 and Gen2 use 8B/10B encoding which introduces a 20% overhead.

	Link Width
	×4
PCI Express Gen2 (5.0 Gbps) - 256-bit interface	N/A

Table 3. PCI Express Data Throughput

The following table shows the aggregate bandwidth of a PCI Express link for Gen2 and Gen3 for supported link widths. The protocol specifies 2.5 giga-transfers per second for Gen1, 5.0 giga-transfers per second for Gen2, and 8.0 giga-transfers per second for Gen3. This table provides bandwidths for a single transmit (TX) or receive (RX) channel. The numbers double for duplex operation. Gen1 and Gen2 use 8B/10B encoding which introduces a 20% overhead. In contrast, Gen3 uses 128b/130b encoding which reduces the data throughput lost to encoding to about 1.5%.

	Link Width	
	×4	×8
PCI Express Gen2 (5.0 Gbps) - 256-bit interface	N/A	32
PCI Express Gen3 (8.0 Gbps) - 256-bit interface	31.51	63

Related Information

- Introduction to Intel FPGA IP Cores
  Provides general information about all Intel FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.
- Creating Version-Independent IP and Platform Designer Simulation Scripts
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- Project Management Best Practices
  Guidelines for efficient management and portability of your project and IP files.
- Intel Arria 10 Avalon-ST with SR-IOV Interface for PCIe Solutions User Guide Archive on page 134
- Avalon Interface Specifications
  For information about the Avalon-ST interface protocol.
- Arria 10 Avalon-ST Interface for PCIe Solutions User Guide
  For the Avalon-ST interface to the application without SR-IOV.
- PCI Express Base Specification 3.0

1.1.1. SR-IOV Features

New features in the Intel Quartus Prime 17.1 release:

- Added parameter to invert TX polarity.

The Intel Arria 10 Hard IP for PCI Express with SR-IOV supports the following features:

- Support for ×4, and ×8 configurations with Gen2 or Gen3 lane rates for Endpoints
- Configuration Spaces for up to four PCIe Physical Functions (PFs) and a maximum of 2048 Virtual Functions (VFs) for the PFs
- Base address register (BAR) checking logic
- Dedicated 16 kilobyte (KB) receive buffer
- Platform Designer example designs demonstrating parameterization, design modules, and connectivity
- Extended credit allocation settings to better optimize the RX buffer space based on application type
- Support for Advanced Error Reporting (AER) for PFs
- Support for Address Translation Services (ATS) and TLP Processing Hints (TPH) capabilities
- Support for a Control Shadow Interface to read the current settings for some of the VF Control Register fields in the PCI and PCI Express Configuration Spaces
- Support for Configuration Space Bypass Mode, allowing you to design a custom Configuration Space and support multiple functions
- Support for Function Level Reset (FLR) for PFs and VFs
- Support for Gen3 PIPE simulation
- Support for the following interrupt types:
  - Message signaled interrupts (MSI) for PFs
  - MSI-X for PFs and VFs
  - Legacy interrupts for PFs
- Easy to use:
  - Flexible configuration.
  - Example designs to get started.

The *Intel Arria 10 Avalon-ST Interface with SR-IOV PCIe Solutions User Guide* explains how to use this IP core and not the PCI Express protocol. Although there is inevitable overlap between these two purposes, use this document only in conjunction with an understanding of the *PCI Express Base Specification*.

Note: This release provides separate user guides for the different variants.

Related Information

- Arria 10 Avalon-MM DMA Interface for PCIe Solutions User Guide
  For the Avalon-MM interface and DMA functionality.
- Arria 10 Avalon-MM Interface for PCIe Solutions User Guide
  For the Avalon-MM interface with no DMA.
- Arria 10 Avalon-ST Interface for PCIe Solutions User Guide
  For the Avalon-ST interface.

1.2. Release Information

Table 4. Hard IP for PCI Express Release Information

Item	Description
Version	17.1
Release Date	November 2017
Ordering Codes	Primary: IP-PCIE/SRIOV Renewal: IPR-PCIE/SRIOV
Product IDs	00FB
Vendor ID	6AF7

Intel verifies that the current version of the Quartus Prime software compiles the previous version of each IP core, if this IP core was included in the previous release. Intel reports any exceptions to this verification in the *Intel IP Release Notes* or clarifies them in the Quartus Prime IP Update tool. Intel does not verify compilation with IP core versions older than the previous release.

Related Information

- Intel FPGA IP Release Notes
  Provides release notes for the current and past versions Intel FPGA IP cores.
- Errata for the Intel Arria 10 Hard IP for PCI Express IP Core in the Knowledge Base

1.3. 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 Arria 10	Final.
Other device families	Refer to the *Intel's PCI Express IP Solutions* web page for support information on other device families.

Related Information

PCI Express Solutions Web Page

1.4. Debug Features

Debug features allow observation and control of the Hard IP for faster debugging of system-level problems.

Related Information

Debugging on page 125

1.5. IP Core Verification

To ensure compliance with the PCI Express specification, Intel performs extensive verification. The simulation environment uses multiple testbenches that consist of industry-standard bus functional models (BFMs) driving the PCI Express link interface. Intel performs the following tests in the simulation environment:

- Directed and pseudorandom stimuli test the Application Layer interface, Configuration Space, and all types and sizes of TLPs
- Error injection tests inject errors in the link, TLPs, and Data Link Layer Packets (DLLPs), and check for the proper responses
- PCI-SIG® Compliance Checklist tests that specifically test the items in the checklist
- Random tests that test a wide range of traffic patterns

Intel provides example designs that you can leverage to test your PCBs and complete compliance base board testing (CBB testing) at PCI-SIG, upon request.

1.5.1. Compatibility Testing Environment

Intel has performed significant hardware testing to ensure a reliable solution. In addition, Intel internally tests every release with motherboards and PCI Express switches from a variety of manufacturers. All PCI-SIG compliance tests are run with each IP core release.

1.6. Performance and Resource Utilization

Because the PCIe protocol stack is implemented in hardened logic, it uses no core device resources (no ALMs and no embedded memory).

The SR-IOV Bridge is implemented is soft logic, requiring FPGA fabric resources. The following table shows the typical device resource utilization for selected configurations using the current version of the Quartus Prime software. With the exception of M20K memory blocks, the numbers of ALMs and logic registers are rounded up to the nearest 50.

Table 6. Performance and Resource Utilization Intel Arria 10 Avalon-ST with SR-IOV

Number of PFs and VFs	ALMs	M20K Memory Blocks	Logic Registers
1 PF, 4 VFs	2350	0	5200
2 PFs, 4 VFs	3600	0	6500

continued...

Number of PFs and VFs	ALMs	M20K Memory Blocks	Logic Registers
4 PFs, 4 VFs	4650	0	7700
1 PF, 2048 VFs	10350	0	5700
2 PFs, 2048 VFs	11750	0	7500
4 PFs 2048 VFs	14150	0	10650
2 PFs	2300	0	5100
4 PFs	3450	0	6300

Related Information

Running the Fitter
For information on Fitter constraints.

1.7. Recommended Speed Grades for SR-IOV Interface

Table 7. Intel Arria 10 Recommended Speed Grades for All SR-IOV Configurations

Intel recommends setting the Quartus Prime Analysis & Synthesis Settings **Optimization Technique** to **Speed** when the Application Layer clock frequency is 250 MHz. For information about optimizing synthesis, refer to "*Setting Up and Running Analysis and Synthesis* in Quartus Prime Help. For more information about how to effect the **Optimization Technique** settings, refer to *Area and Timing Optimization* in volume 2 of the *Quartus Prime Handbook*. Refer to the *Related Links* below.

Link Rate	Link Width	Interface Width	Application Clock Frequency (MHz)	Recommended Speed Grades
Gen2	×8	256 bits	125	–1, –2, –3
Gen3	×4	256 bits	125	–1, –2, –3
	×8	256 bits	250	–1, –2

Table 8. Intel Arria 10 Recommended Speed Grades for All SR-IOV Configurations

Intel recommends setting the Quartus Prime Analysis & Synthesis Settings **Optimization Technique** to **Speed** when the Application Layer clock frequency is 250 MHz. For information about optimizing synthesis, refer to "*Setting Up and Running Analysis and Synthesis* in Quartus Prime Help. For more information about how to effect the **Optimization Technique** settings, refer to *Area and Timing Optimization* in volume 2 of the *Quartus Prime Handbook*. Refer to the *Related Links* below.

Link Rate	Link Width	Interface Width	Application Clock Frequency (MHz)	Recommended Speed Grades
Gen2	×4	64 bits 128	125	–1, –2, –3

Related Information

- Running Synthesis
  For settings that affect timing closure.
- Intel FPGA Software Installation and Licensing Manual
  For comprehensive information for installing and licensing Intel FPGA software.

2. Getting Started with the SR-IOV Design Example

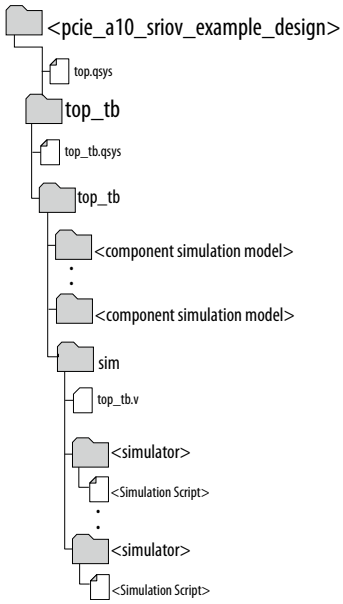
The SR-IOV example design consists of a PCIe Endpoint that includes an SR-IOV bridge configured for one PF and four VFs. The example design also includes a basic application to facilitate host accesses to a target memory. This design example supports simulation. In simulation, the testbench issues downstream memory accesses to the virtual function BAR. The testbench then reads the data written and compares it to the expected result. The test passes if all the comparisons pass.

When you install the Intel Quartus Prime software you also install the IP Library. This installation includes design examples for Hard IP for PCI Express under the `<install_dir>/ip/altera/altera_pcie/` directory. You can copy the design examples from the `<install_dir>/ip/altera/ altera_pcie/ altera_pcie_a10_ed/example_design/a10` directory. This walkthrough uses the `sriov2_target_g3x8_1pf_4vf.qsys` design example.

Note: Starting in the Quartus Prime 16.0 software release, you cannot simulate or compile SR-IOV designs without a license. Contact your local sales representative or email pcie@altera.com to obtain a license.

2.1. Directory Structure for Intel Arria 10 SR-IOV Design Example

Figure 3. Directory Structure for the Generated Example Design



2.2. Design Components for the SR-IOV Design Example

Figure 4. Platform Designer Testbench for Intel Arria 10 Gen1 x8 128-bit SR-IOV Design Example

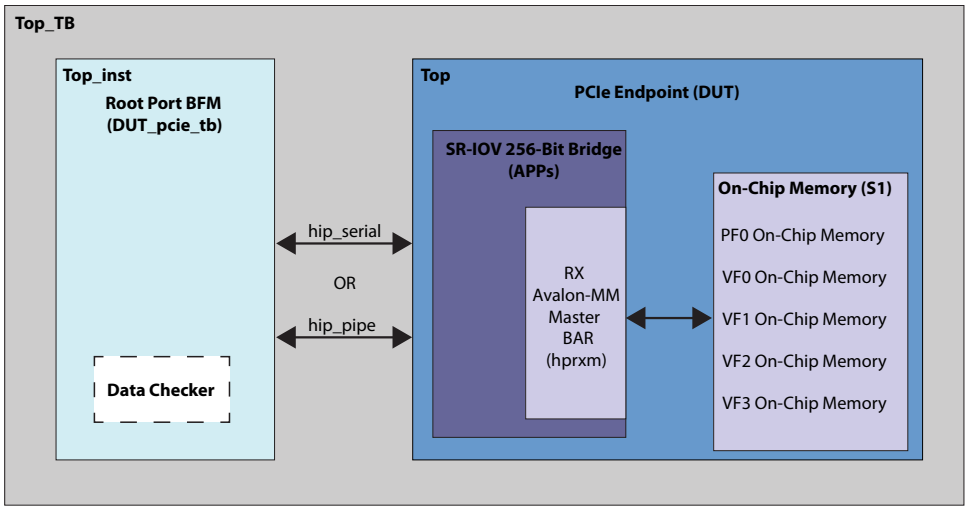
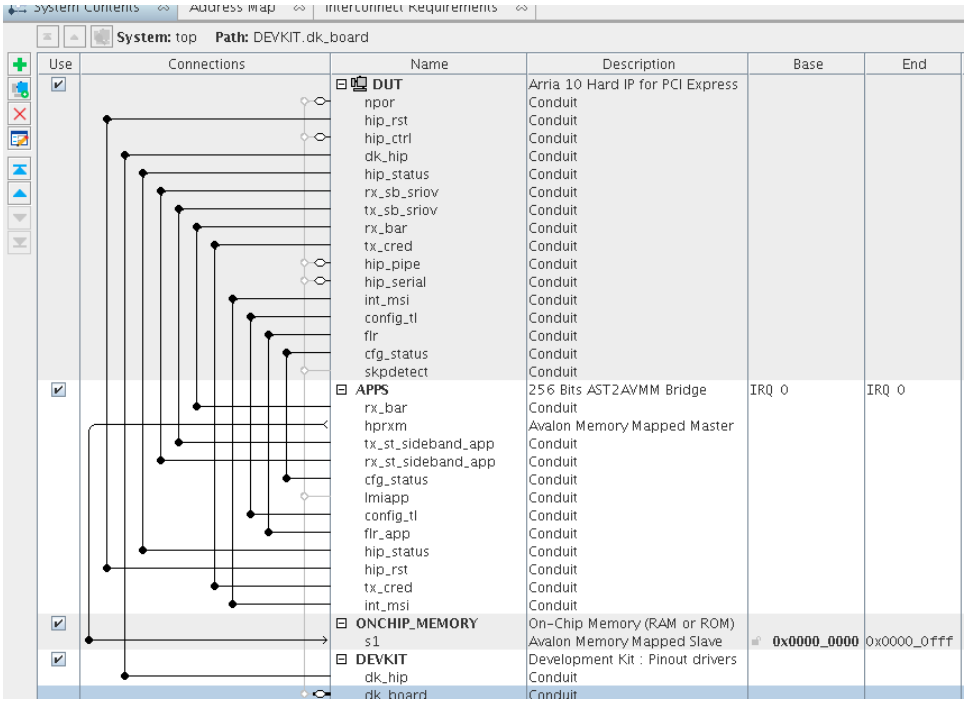


Figure 5. Platform Designer Schematic for Top

This image of the Intel Arria 10 PCI Express DMA Design Example shows only the Avalon-ST, clock, and reset interfaces.



The testbench includes a PCIe Root Port BFM and a PCIe Gen3 x8 Endpoint implemented in hard logic. The SR-IOV bridge, implemented in soft logic, drives memory writes and reads to the four VFs. The simulation includes the following stages:

- Link Training
- Configuration
- Memory writes to each VF
- Memory reads and compares to the expected data

2.3. Generating the SR-IOV Design Example

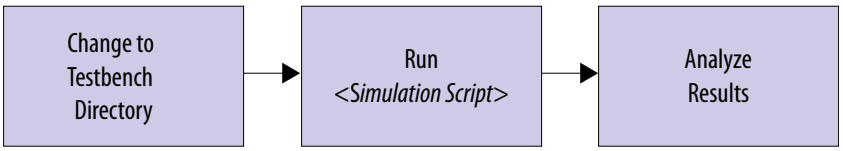
After installing the Quartus Prime software, copy the design examples from the `<install_dir>/ip/altera/ altera_pcie/altera_pcie_a10_ed/example_design/a10` directory. This walkthrough uses the `sriov2_top_target_gen3x8_1pf_4vf.qsys` design example. To run the simulation, you must rename the design example `top.qsys`

1. Launch Platform Designer and open `top.qsys`.
2. On the Generate menu, select **Generate Testbench System**.
3. For **Create testbench Platform Designer system**, select **Standard, BFMs for stand Platform Designer interfaces**.
4. For **Create testbench simulation model**, select either **Verilog** or **VHDL**.
5. For **Output Directory ➤ Testbench**, you can accept the default directory or modify it.
6. Click **Generate**.

   Note: Intel Arria 10 devices do not support the **Create timing and resource estimates for third-party EDA synthesis tools** option on the **Generate ➤ Generate HDL** menu. You can select this menu item, but generation fails.

2.4. Compiling and Simulating the Design for SR-IOV

Figure 6. Procedure



Follow these steps to compile and simulate the design:

1. Change the simulation directory.
2. Run the simulation script for the simulator of your choice. Refer to the table below.
3. Analyze the results.

Table 9. Steps to Run Simulation

Simulator	Working Directory	Instructions
Mentor ModelSim*	*<example_design>*/top_tb/top_tb/sim/mentor/	a. Invoke vsim b. do msim_setup.tcl c. ld_debug d. run -all e. A successful simulation ends with the following message, "Simulation stopped due to successful completion! Simulation passed."
Mentor VCS*	<example_design>/top_tb/top_tb/sim/synopsys/vcs	a. sh vcs_setup.sh USER_DEFINED_SIM_OPTIONS="" b. A successful simulation ends with the following message, "Simulation stopped due to successful completion! Simulation passed."
Cadence NCSim*	*<example_design>*top_tb/top_tb/sim/cadence	a. Create a shell script, my_setup.sh. This script allows you to add additional commands and override the defaults included in ncsim_setup.sh. b. Include the following command in my_setup.sh: source ncsim_setup.sh USER_DEFINED_SIM_OPTIONS="" c. chmod +x *.sh d. ./my_setup.sh e. A successful simulation ends with the following message, "Simulation stopped due to successful completion! Simulation passed."

3. Parameter Settings

3.1. Parameters

This chapter provides a reference for all the parameters of the Intel Arria 10 Hard IP for PCI Express IP core.

Table 10. Design Environment Parameter

Starting in Intel Quartus Prime 18.0, there is a new parameter **Design Environment** in the parameters editor window.

Parameter	Value	Description
Design Environment	**Standalone** **System**	Identifies the environment that the IP is in. • The **Standalone** environment refers to the IP being in a standalone state where all its interfaces are exported. • The **System** environment refers to the IP being instantiated in a Platform Designer system.

Table 11. System Settings

Parameter	Value	Description
Application Interface Type	**Avalon-ST** **Avalon-MM** **Avalon-MM with DMA** **Avalon-ST with SR-IOV**	Selects the interface to the Application Layer. *Note:* When the **Design Environment** parameter is set to **System**, all four **Application Interface Types** are available. However, when **Design Environment** is set to **Standalone**, only **Avalon-ST** and **Avalon-ST with SR-IOV** are available.
Hard IP mode	**Gen3x8, Interface: 256-bit, 250 MHz** **Gen3x4, Interface: 256-bit, 125 MHz** **Gen3x4, Interface: 128-bit, 250 MHz** **Gen3x2, Interface: 128-bit, 125 MHz** **Gen3x2, Interface: 64-bit, 250 MHz** **Gen3x1, Interface: 64-bit, 125 MHz** **Gen2x8, Interface: 256-bit, 125 MHz** **Gen2x8, Interface: 128-bit, 250 MHz** **Gen2x4, Interface: 128-bit, 125 MHz** **Gen2x2, Interface: 64-bit, 125 MHz** **Gen2x4, Interface: 64-bit, 250 MHz** **Gen2x1, Interface: 64-bit, 125 MHz** **Gen1x8, Interface: 128-bit, 125 MHz** **Gen1x8, Interface: 64-bit, 250 MHz** **Gen1x4, Interface: 64-bit, 125 MHz** **Gen1x2, Interface: 64-bit, 125 MHz** **Gen1x1, Interface: 64-bit, 125 MHz** **Gen1x1, Interface: 64-bit, 62.5 MHz**	Selects the following elements: • The lane data rate. Gen1, Gen2, and Gen3 are supported • The width of the data interface between the hard IP Transaction Layer and the Application Layer implemented in the FPGA fabric • The Application Layer interface frequency Intel Cyclone® 10 GX devices support up to Gen2 x4 configurations.
Port type	**Native Endpoint**	Specifies the port type.

continued...

Parameter	Value	Description
	Root Port	The Endpoint stores parameters in the Type 0 Configuration Space. The Root Port stores parameters in the Type 1 Configuration Space. The **Avalon-ST with SR-IOV** interface supports only **Native Endpoint** operation. You can enable the Root Port in the current release. Root Port mode only supports the Avalon®-MM interface type, and it only supports basic simulation and compilation. However, the Root Port mode is not fully verified.
RX Buffer credit allocation - performance for received requests	**Minimum** **Low** **Balanced** **High** **Maximum**	Determines the allocation of posted header credits, posted data credits, non-posted header credits, completion header credits, and completion data credits in the 16 KB RX buffer. The settings allow you to adjust the credit allocation to optimize your system. The credit allocation for the selected setting displays in the **Message** pane. The **Message** pane dynamically updates the number of credits for Posted, Non-Posted Headers and Data, and Completion Headers and Data as you change this selection. Refer to the *Throughput Optimization* chapter for more information about optimizing your design. Refer to the *RX Buffer Allocation Selections Available by Interface Type* below for the availability of these settings by interface type. **Minimum**—configures the minimum PCIe specification allowed for non-posted and posted request credits, leaving most of the RX Buffer space for received completion header and data. Select this option for variations where application logic generates many read requests and only infrequently receives single requests from the PCIe link. **Low**—configures a slightly larger amount of RX Buffer space for non-posted and posted request credits, but still dedicates most of the space for received completion header and data. Select this option for variations where application logic generates many read requests and infrequently receives small bursts of requests from the PCIe link. This option is recommended for typical endpoint applications where most of the PCIe traffic is generated by a DMA engine that is located in the endpoint application layer logic. **Balanced**—configures approximately half the RX Buffer space to received requests and the other half of the RX Buffer space to received completions. Select this option for variations where the received requests and received completions are roughly equal. **High**—configures most of the RX Buffer space for received requests and allocates a slightly larger than minimum amount of space for received completions. Select this option where most of the PCIe requests are generated by the other end of the PCIe link and the local application layer logic only infrequently generates a small burst of read requests. This option is recommended for typical Root Port applications where most of the PCIe traffic is generated by DMA engines located in the endpoints. **Maximum**—configures the minimum PCIe specification allowed amount of completion space, leaving most of the RX Buffer space for received requests. Select this option when most of the PCIe requests are generated by the other end of the PCIe link and the local application layer logic never or only infrequently generates single read requests. This

continued...

Parameter	Value	Description
		option is recommended for control and status endpoint applications that do not generate any PCIe requests of their own and only are the target of write and read requests from the root complex.
RX Buffer completion credits	**Header credits** **Data credits**	Displays the number of completion credits in the 16 KB RX buffer resulting from the credit allocation parameter. Each header credit is 16 bytes. Each data credit is 20 bytes.

3.2. Intel Arria 10 Avalon-ST Settings

Table 12. System Settings for PCI Express

Parameter	Value	Description
Enable Avalon-ST reset output port	**On/Off**	When **On**, the generated reset output port has the same functionality that the `reset_status` port included in the Reset and Link Status interface.
Enable byte parity ports on Avalon-ST interface	**On/Off**	When **On**, the RX and TX datapaths are parity protected. Parity is odd. The Application Layer must provide valid byte parity in the Avalon-ST TX direction. This parameter is only available for the Avalon-ST Intel Arria 10 Hard IP for PCI Express.
Enable multiple packets per cycle for the 256-bit interface	**On/Off**	When **On**, the 256-bit Avalon-ST interface supports the transmission of TLPs starting at any 128-bit address boundary, allowing support for multiple packets in a single cycle. To support multiple packets per cycle, the Avalon-ST interface includes 2 start of packet and end of packet signals for the 256-bit Avalon-ST interfaces. This is not supported for the Avalon-ST with SR-IOV interface.
Enable credit consumed selection port	**On/Off**	When you turn on this option, the core includes the `tx_cons_cred_sel` port. This parameter does not apply to the Avalon-MM interface.
Enable Configuration bypass (CfgBP)	**On/Off**	When **On**, the Intel Arria 10 Hard IP for PCI Express bypasses the Transaction Layer Configuration Space registers included as part of the Hard IP, allowing you to substitute a custom Configuration Space implemented in soft logic. This parameter is not available for the Avalon-MM IP Cores.
Enable local management interface (LMI)	**On/Off**	When **On**, your variant includes the optional LMI interface. This interface is used to log error descriptor information in the TLP header log registers. The LMI interface provides the same access to Configuration Space registers as Configuration TLP requests.

Related Information

PCI Express Base Specification 3.0

3.3. Intel Arria 10 SR-IOV System Settings

Parameter	Value	Description
Total Physical Functions (PFs) :	`1–4`	This core supports 1-4 Physical Functions.
Total Virtual Functions of Physical Function0 (PF0 VFs) - Total Virtual Functions of Physical Function3 (PF3 VFs):	`0 –2048`	Total number of VFs assigned to a PF. You can assign VFs in the following granularities:

continued...

Parameter	Value	Description
		• Granularity of 1 for 1-8 VFs • Granularity of 4 for 8-256 VFs • Granularity of 64 for 256-1024 VFs • Granularity of 512 for 1024-2048 VFs The sum of VFs assigned to PF0, PF1, PF2 and PF3 cannot exceed the 2048 VF Total.
System Supported Page Size:	**4KB - 4MB**	Specifies the pages sizes supported. Sets the `Supported Page Sizes` register of the SR-IOV Capability structure.
Enable SR-IOV Support	**On/Off**	When **On**, the variant supports multiple PFs and VFs. When **Off**, .supports PFs only.
Enable Alternative Routing-ID (ARI) support	**On/Off**	When **On**, ARI supports up to 256 functions. Refer to *Section 6.1.3 Alternative Routing-ID Interpretation (ARI)* of the *PCI Express Base Specification* for more information about ARI.
Enable Functional Level Reset (FLR)	**On/Off**	When **On**, each function has its own, individual reset.
Enable TLP Processing Hints (TPH) support for PFs	**On/Off**	When **On**, the variant includes the TPH registers to help you improve latency and traffic congestion.
Enable TLP Processing Hints (TPH) support for VFs		
Enable Address Translation Services (ATS) support for PFs	**On/Off**	When **On**, the variant includes the ATS registers.
Enable Address Translation Services (ATS) support for VFs		

Related Information

PCI Express Base Specification 2.1 or 3.0

3.4. Base Address Register (BAR) Settings

Each function can implement up to six BARs. You can configure up to six 32-bit BARs or three 64-bit BARs for both PFs and VFs. The BAR settings are the same for all VFs associated with a PF.

Table 13. BAR Registers

Parameter	Value	Description
Present (BAR0-BAR5)	**Enabled/Disabled**	Indicates whether or not this BAR is instantiated.
Type	**32-bit address** **64-bit address**	Specifies 32- or 64-bit addressing.
Prefetchable	**Prefetchable** **Non-prefetchable**	Defining memory as **Prefetchable** allows data in the region to be fetched ahead anticipating that 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 • Write merging is allowed If you select **64-bit address**, 2 contiguous BARs are combined to form a 64-bit BAR. You must set the higher numbered BAR to **Disabled**.

continued...

Parameter	Value	Description
		If the BAR TYPE of any even BAR is set to 64-bit memory, the next higher BAR supplies the upper address bits. The supported combinations for 64-bit BARs are {BAR1, BAR0}, {BAR3, BAR2}, {BAR4, BAR5}.
Size	**16 Bytes–2 GB**	Specifies the memory size.

3.5. SR-IOV Device Identification Registers

Table 14. Device ID Registers

The following table lists the default values of the read-only Device ID registers. You can use the parameter editor to change the values of these registers. At run time, you can change the values of these registers using the optional reconfiguration block signals. You can specify Device ID registers for each Physical Function.

Register Name	Default Value	Description
Vendor ID	0x00001172	Sets the read-only value of the `Vendor ID` register. This parameter can not be set to 0xFFFF per the PCI Express Specification. Address offset: 0x000.
Device ID	0x00000000	Sets the read-only value of the `Device ID` register. Address offset: 0x000.
VF Device ID	0x00000000	Sets the read-only value of the VF Device ID register.
Revision ID	0x00000000	Sets the read-only value of the `Revision ID` register. Address offset: 0x008.
Class code	0x00000000	Sets the read-only value of the `Class Code` register. Address offset: 0x008.
Subclass code	0x00000000	Sets the read-only value of the `Subclass Code` register. Address offset: 0x008.
Subsystem Vendor ID	0x00000000	Sets the read-only value of the register in the PCI Type 0 Configuration Space. This parameter cannot be set to 0xFFFF per the *PCI Express Base Specification*. This value is assigned by PCI-SIG to the device manufacturer. Address offset: 0x02C.
Subsystem Device ID	0x00000000	Sets the read-only value of the `Subsystem Device ID` register in the PCI Type 0 Configuration Space. Address offset: 0x02C

Related Information

PCI Express Base Specification 2.1 or 3.0

3.6. Intel Arria 10 Interrupt Capabilities

Table 15. MSI and MSI-X Interrupt Settings

Each Physical Function defines its own MSI-X table settings. The VF MSI-X table settings are the same for all the Virtual Functions associated with each Physical Function.

Parameter	Value	Description
MSI Interrupt Settings		
PF0 MSI Requests - PF3 MSI Requests	**1,2,4,8,16,32**	Specifies the maximum number of MSI messages the Application Layer can request. This value is reflected in Multiple Message Capable field of the `Message Control` register, 0x050[31:16]. For MSI Interrupt Settings, if the **PF MSI** option is enabled, all PFs support MSI capability.
MSI-X PF0 - MSI-X PF3 Interrupt Settings		
PF MSI-X	**On/Off**	When **On**, enables the MSI-X functionality. For PF and VF MSI-X Interrupt Settings, if **PF MSI-X** is enabled, all PFs supports MSI-X capability.
VF MSI-X	**On/Off**	
	Bit Range	
MSI-X Table size	[10:0]	System software reads this field to determine the MSI-X Table size <*n*>, which is encoded as <*n–1*>. For example, a returned value of 2047 indicates a table size of 2048. This field is read-only. Legal range is 0–2047 (2^{11}). Address offset: 0x068[26:16]
MSI-X Table Offset	[31:0]	Specifies the offset from the BAR indicated in the**MSI-X Table BAR Indicator**. The lower 3 bits of the table BAR indicator (BIR) are set to zero by software to form a 32-bit qword-aligned offset (1). This field is read-only.
MSI-X Table BAR Indicator	[2:0]	Specifies which one of a function's BAR number. This field is read-only. For 32-bit BARs, the legal range is 0–5. For 64-bit BARs, the legal range is 0, 2, or 4.
MSI-X Pending Bit Array (PBA) Offset	[31:0]	Points to the MSI-X Pending Bit Array table. It is offset from the BAR value indicated in **MSI-X Table BAR Indicator**. 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.
MSI-X PBA BAR Indicator	[2:0]	Specifies which BAR number contains the MSI-X PBA. For 32-bit BARs, the legal range is 0–5. For 64-bit BARs, the legal range is 0, 2, or 4. This field is read-only.
Legacy Interrupts		
PF0 - PF3 Interrupt Pin	**inta–intd**	Applicable for PFs only to support legacy interrupts. When enabled, the core receives interrupt indications from the Application Layer on its `INTA_IN`, `INTB_IN`, `INTC_IN` and `INTD_IN` inputs, and sends out `Assert_INTx` or `Deassert_INTx` messages on the link in response to their activation or deactivation, respectively. You can configure the Physical Functions with separate interrupt pins. Or, both functions can share a common interrupt pin.
PF0 - PF3 Interrupt Line	**0-255**	Defines the input to the interrupt controller (IRQ0 - IRQ15) in the Root Port that is activated by each `Assert_INTx` message.

(1) Throughout this user guide, the terms word, dword and qword have the same meaning that they have in the *PCI Express Base Specification*. A word is 16 bits, a dword is 32 bits, and a qword is 64 bits.

Related Information

PCI Express Base Specification Revision 2.1 or 3.0

3.7. Physical Function TLP Processing Hints (TPH)

TPH support PFs that target a TLP towards a specific processing resource such as a host processor or cache hierarchy. Steering Tags (ST) provide design-specific information about the host or cache structure.

Software programs the Steering Tag values that are stored in an ST table. You can store the ST Table in the TPH Requestor Capability structure or combine it with the MSI-X Table. For more information about Steering Tags, refer to *Section 6.17.2 Steering Tags* of the *PCI Express Base Specification, Rev. 3.0.* After analyzing the traffic of your system, you may be able to use TPH hints to improve latency or reduce traffic congestion.

Table 16. TPH Capabilities

The values specified here are common for all PFs.

Parameter	Value	Description
Interrupt Mode	**On/Off**	When **On**, the Steering Tag is selected by an MSI/MSI-X interrupt vector number.
Device Specific Mode	**On/Off**	When **On**, the Steering Tag is selected from Steering Tag Table entry stored in the TPH Requestor Capability structure.
Steering Tag Table location	**0, 1, 2**	When non-zero, specifies the location of the Steering Tag Table. The following encodings are defined: • 0: Steering Tag table not present • 1: Steering Tag table is stored in the TPH Requestor Capability structure • 2: Steering Tag table is located in the MSI-X table.
Steering Tag Table size	**0-2047**	Specifies the number of 2-byte Steering Table entries.

Related Information

PCI Express Base Specification Revision 3.0

3.8. Address Translation Services (ATS)

ATS extends the PCIe protocol to support an address translation agent (TA) that translates DMA addresses to cached addresses in the device. The translation agent can be located in or above the Root Port. Locating translated addresses in the device minimizes latency and provides a scalable, distributed caching system that improves I/O performance. The Address Translation Cache (ATC) located in the device reduces the processing load on the translation agent, enhancing system performance. For more information about ATS, refer to *Address Translation Services Revision 1.1*.

Table 17. ATS Capabilities

ATS must maintain cache coherence between the addresses in the TA and ATC. The TA and associated software ensure that the addresses caches in the ATC are not stale by issuing Invalidate Requests. An Invalidate Request clears a specific subset of the address range from the ATC. For more information about ATS, refer to *Address Translation Services Revision 1.1* The values specified here are common for all PFs.

Parameter	Value	Description
PF0 - PF3 Maximum outstanding Invalidate Requests	0–32	Specifies the maximum number outstanding Invalidate Requests for each PF before putting backpressure on the upstream connection.

Related Information

- Address Translation Services Revision 1.1
- PCI Express Base Specification Revision 3.0

3.9. PCI Express and PCI Capabilities Parameters

This group of parameters defines various capability properties of the IP core. Some of these parameters are stored in the PCI Configuration Space - PCI Compatible Configuration Space. The byte offset indicates the parameter address.

3.9.1. PCI Express and PCI Capabilities

Table 18. Capabilities Registers

Parameter	Possible Values	Default Value	Description
Maximum payload size	**128 bytes** **256 bytes** **512 bytes** **1024 bytes** **2048 bytes**	128 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 (0x084[2:0]). Address: 0x084. *Note:* The SR-IOV bridge supports a single value for the **Maximum payload size** parameter. When the configuration includes 2 or more PFs, you must program all 4 PFs for the SR-IOV bridge to specify a value larger than 128 bytes.
Number of Tags supported	**32** **64**	32	Indicates the number of tags supported for non-posted requests transmitted by the Application Layer. This parameter sets the values in the Device Control register (0x088) of the PCI Express capability structure described in Table 9–9 on page 9–5. The Transaction Layer tracks all outstanding completions for non-posted requests made by the Application Layer. This parameter configures the Transaction Layer for the maximum number of **Tags supported** to track. The Application Layer must set the tag values in all non-posted PCI Express headers to be less than this value. Values greater than 32 also set the extended tag field supported bit in the Configuration Space Device Capabilities register. The Application Layer can only use tag numbers greater than 31 if configuration software sets the Extended Tag Field Enable bit of the Device Control register.
Completion timeout range	**ABCD** **BCD** **ABC** **AB** **B** **A** **None**	ABCD	Indicates device function support for the optional completion timeout programmability mechanism. This mechanism allows system software to modify the completion timeout value. This field is applicable only to Root Ports and Endpoints that issue requests on their own behalf. Completion timeouts are specified and enabled in the Device Control 2 register (0x0A8) of the *PCI*

continued...

Parameter	Possible Values	Default Value	Description
			Express Capability Structure Version. For all other functions this field is reserved and must be hardwired to 0x0000b. Four time value ranges are defined: • Range A: 50 us to 10 ms • Range B: 10 ms to 250 ms • Range C: 250 ms to 4 s • Range D: 4 s to 64 s Bits are set to show timeout value ranges supported. The function must implement a timeout value in the range 50 s to 50 ms. The following values specify the range: • None—Completion timeout programming is not supported • 0001 Range A • 0010 Range B • 0011 Ranges A and B • 0110 Ranges B and C • 0111 Ranges A, B, and C • 1110 Ranges B, C and D • 1111 Ranges A, B, C, and D All other values are reserved. Intel recommends that the completion timeout mechanism expire in no less than 10 ms.
Disable completion timeout	**On/Off**	On	Disables the completion timeout mechanism. When **On**, the core supports the completion timeout disable mechanism via the PCI Express `Device Control Register 2`. The Application Layer logic must implement the actual completion timeout mechanism for the required ranges.

3.9.2. Error Reporting

Table 19. Error Reporting

Parameter	Value	Default Value	Description
Enable Advanced Error Reporting (AER)	**On/Off**	Off	When **On**, enables the Advanced Error Reporting (AER) capability.
Enable ECRC checking	**On/Off**	Off	When **On**, enables ECRC checking. Sets the read-only value of the ECRC check capable bit in the `Advanced Error Capabilities and Control Register`. This parameter requires you to enable the AER capability.
Enable ECRC generation	**On/Off**	Off	When **On**, enables ECRC generation capability. Sets the read-only value of the ECRC generation capable bit in the `Advanced Error Capabilities and Control Register`. This parameter requires you to enable the AER capability.
Enable ECRC forwarding on the Avalon-ST interface	**On/Off**	Off	When **On**, enables ECRC forwarding to the Application Layer. On the Avalon-ST RX path, the incoming TLP contains the ECRC dword(1) and the `TD` bit is set if an ECRC exists. On the transmit the TLP from the Application Layer must contain the ECRC dword and have the `TD` bit set.
Track RX completion buffer	**On/Off**	Off	When **On**, the core includes the `rxfc_cplbuf_ovf` output status signal to track the RX posted completion buffer overflow status.

continued...

Parameter	Value	Default Value	Description
overflow on the Avalon-ST interface			

Note:
1. Throughout this user guide, the terms word, dword and qword have the same meaning that they have in the *PCI Express Base Specification*. A word is 16 bits, a dword is 32 bits, and a qword is 64 bits.

3.9.3. Link Capabilities

Table 20. Link Capabilities

Parameter	Value	Description
Link port number (Root Port only)	**0x01**	Sets the read-only value of the port number field in the `Link Capabilities` register. This parameter is for Root Ports only. It should not be changed.
Data link layer active reporting (Root Port only)	**On/Off**	Turn **On** this parameter for a Root Port, if the attached Endpoint supports the optional capability of reporting the DL_Active state of the Data Link Control and Management State Machine. For a hot-plug capable Endpoint (as indicated by the `Hot Plug Capable` field of the `Slot Capabilities` register), this parameter must be turned **On**. For Root Port components that do not support this optional capability, turn **Off** this option.
Surprise down reporting (Root Port only)	**On/Off**	When your turn this option **On**, an Endpoint supports the optional capability of detecting and reporting the surprise down error condition. The error condition is read from the Root Port.
Slot clock configuration	**On/Off**	When you turn this option **On**, indicates that the Endpoint uses the same physical reference clock that the system provides on the connector. When **Off**, 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.

3.9.4. Slot Capabilities

Table 21. Slot Capabilities

Parameter	Value	Description
Use Slot register	**On/Off**	This parameter is only supported in Root Port mode. The slot capability is required for Root Ports if a slot is implemented on the port. Slot status is recorded in the `PCI Express Capabilities` register. Defines the characteristics of the slot. You turn on this option by selecting **Enable slot capability**. Refer to the figure below for bit definitions.
Slot power scale	`0–3`	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.

continued...