IP CORE MANUAL

NAVIGATOR
FPGA Design Kit

PCI Express to AXI4–Lite Bridge
512–Bit IP

px_pcie2axil_512

PENTEK

Pentek, Inc.
One Park Way
Upper Saddle River, NJ 07458
(201) 818–5900
http://www.pentek.com/

Manual Part Number:807.48437 Rev: 1.0 – October 4, 2019

Manual Revision History

Date	Version	Comments
10/4/19	1.0	Initial Release

Legal Notices

Copyright

Trademarks

Printed in the United States of America.

Table of Contents

Page

Table of Contents

Page

IP Facts

Description

Pentek's Navigator™ PCI Express (PCIe®) to AXI4–Lite Bridge 512–Bit Core serves as a bridge between the Completer Interfaces of the Xilinx® Gen3 Integrated Block for PCI Express IP Core and an AXI4–Lite Interface. This version of the core only supports 512–bit data aligned mode, which is only available in the Ultrascale+ family of Xilinx FPGAs.

This core complies with the ARM® AMBA® AXI4 Specification. This user manual defines the hardware interface, software interface, and parameterization options for the PCI Express to AXI4–Lite Bridge 512–Bit Core.

Features

- One dword memory read and write requests
- Six 32–bit Base Address Registers (BARs)
- Address Translation from Transaction Layer Packet (TLP) header address to AXI4 addressing
- Supports 512–Bit data aligned mode only
- Supports shutdown of completer completion interface and ensures data transfer is completed before shutdown

Table 1–1: IP Facts Table

Core Specifics	
Supported Design Family[a]	Ultrascale+
Supported User Interfaces	AXI4–Lite and AXI4–Stream
Resources	See Table 2–1
Provided with the Core	
Design Files	VHDL
Example Design	Not Provided
Test Bench	VHDL
Constraints File	Not Provided[b]
Simulation Model	VHDL
Supported S/W Driver	N/A
Tested Design Flows	
Design Entry	Vivado® Design Suite 2019.1 or later
Simulation	Vivado VSim
Synthesis	Vivado Synthesis
Support	
Provided by Pentek fpgasupport@pentek.com	

a.For a complete list of supported devices, see the *Vivado Design Suite Release Notes.*
b.Clock constraints can be applied at the top level module of the user design.

This page is intentionally blank

Chapter 1: Overview

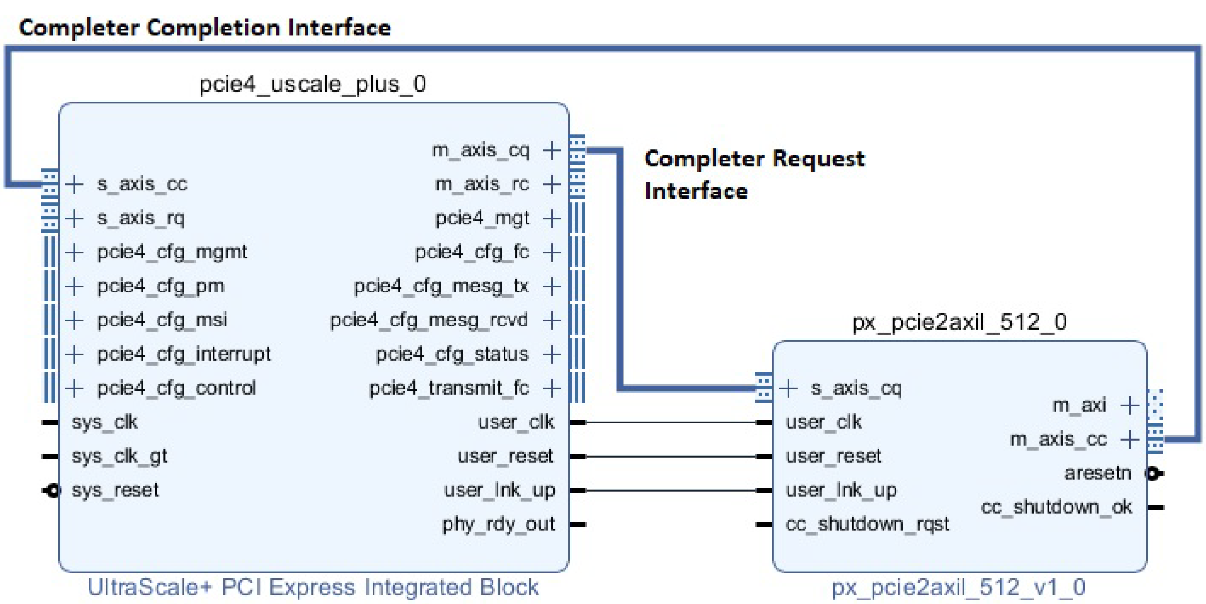
1.1 Functional Description

The Xilinx Gen3 Integrated Block for PCI Express IP Core provides only AXI4–Stream user interfaces, and requires a bridge to access the AXI4–Lite Interface compliant cores. The PCIe to AXI4–Lite Bridge 512–BIt Core is used to serve this purpose, and uses only the Completer Request (CQ) and Completer Completion (CC) Interfaces of the Xilinx PCIe Core.

The Completer Request (**`m_axis_cq`**) and Completer Completion (**`s_axis_cc`**) Interfaces of the Xilinx PCIe Core are connected to the Completer Request (**`s_axis_cq`**) and Completer Completion (**`m_axis_cc`**) Interfaces of the PCIe to AXI4–Lite Bridge 512–BIt Core, respectively. The CQ interface (**`m_axis_cq`**) of the Xilinx PCIe Core provides memory read and write requests from the PCIe host, which are decoded by the PCIe to AXI4–Lite Bridge 512–BIt Core, and translated to the AXI4–Lite Master Interface. Memory write requests from the host are translated to the AXI4–Lite Write Address Channel and Write Data Channel transactions. Memory read requests from the host are translated to the AXI4–Lite Read Address Channel and Read Data channel transactions. A completion TLP is created on the CC Interface (**`m_axis_cc`**) of the core with the payload received from the target AXI4–Lite Slave on the Read Data channel.

Figure 1–1 shows the CC and CQ interface connections from the Xilinx PCIe Core to the PCIe to AXI4–Lite Bridge 512–BIt Core, using Vivado IP Integrator.

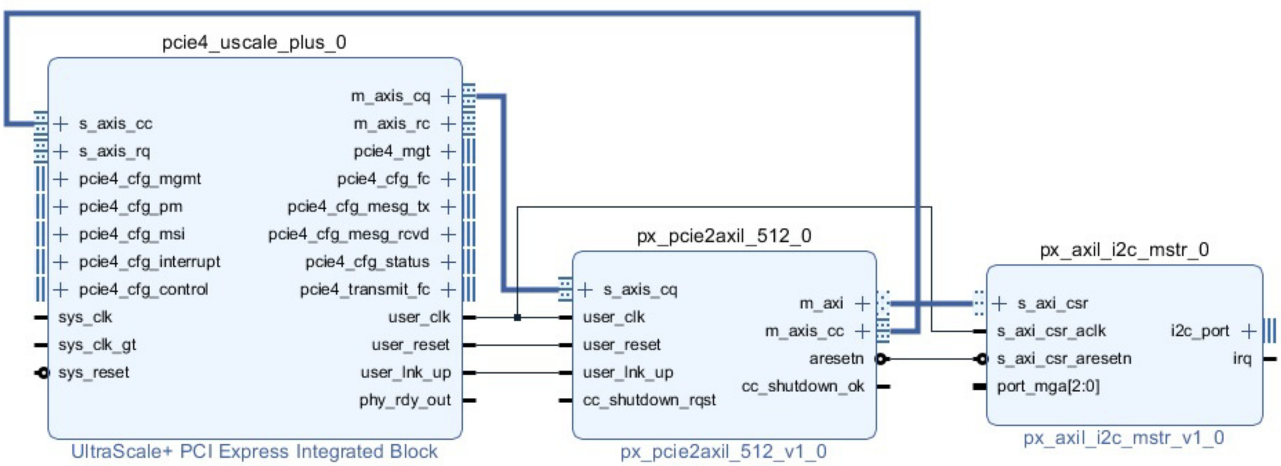
Figure 1–1: PCI Express to AXI4–Lite Bridge 512–Bit Core Interfaces



1.1 Functional Description (continued)

Figure 1–1 shows an example application for the PCIe to AXI4–Lite Bridge 512–BIt Core. The Completer Interfaces of the Xilinx PCIe Core are connected to the Pentek AXI I2C Bus Interface Core through the PCIe to AXI4–Lite Bridge 512–BIt Core, for access–ing the I2C Bus from the PCIe Link.

Figure 1–2: PCI Express to AXI4–Lite Bridge 512–Bit Core Interface Example



1.2 Applications

The PCIe to AXI4–Lite Bridge 512–BIt Core can be used to connect the Completer Interfaces of the Xilinx Gen3 Integrated Block for PCI Express IP Core to any AXI4–Lite Interface compliant core.

1.3 System Requirements

For a list of system requirements, see the Vivado Design Suite Release Notes.

1.4 Licensing and Ordering Information

This core is included with all Pentek Navigator FPGA Design Kits for Pentek Quartz and Jade series board products. Contact Pentek for Licensing and Ordering Information (www.pentek.com).

1.5 Contacting Technical Support

Technical Support for Pentek's Navigator FPGA Design Kits is available via e–mail (fpgasupport@pentek.com) or by phone (201–818–5900 ext. 238, 9 am to 5 pm EST).

1.6 Documentation

This user manual is the main document for this IP core. The following documents provide supplemental material:

1) *Vivado Design Suite User Guide: Designing with IP*

2) *Vivado Design Suite User Guide: Programming and Debugging*

3) *ARM AMBA AXI4 Protocol Version 2.0 Specification*
http://www.arm.com/products/system–ip/amba–specifications.php

4) Xilinx Gen 3 Integrated Block for PCI Express IP Core

This page is intentionally blank

Chapter 2: General Product Specifications

2.1 Standards

The PCI Express to AXI4–Lite Bridge 512–Bit Core has bus interfaces that comply with the *ARM AMBA AXI4–Lite Protocol Specification* and the *AMBA AXI4–Stream Protocol Specification*.

2.2 Performance

The performance of the PCIe to AXI4–Lite Bridge 512–BIt Core is limited only by the FPGA logic speed. The values presented in this section should be used as an estimation guideline. Actual performance can vary.

2.2.1 Maximum Frequencies

The PCIe to AXI4–Lite Bridge 512–BIt Core is designed to meet a target frequency of 250 MHz on an Ultrascale+ –1 speed grade FPGA. 250 MHz is typically the PCI Express (PCIe®) AXI bus clock frequency.

2.3 Resource Utilization

The resource utilization of the PCI Express to AXI4–Lite Bridge 512–Bit Core is shown in Table 2–1. Resources have been estimated for the Ultrascale+ XCZU27DR –1 speed grade device. These values were generated using the Vivado Design Suite.

Table 2–1: Resource Usage and Availability

Resource	# Used
LUTs	267
Flip–Flops	587

NOTE: Actual utilization may vary based on the user design in which the PCIe to AXI4–Lite Bridge 512–BIt Core is incorporated.

2.4 Limitations and Unsupported Features

- The PCIe to AXI4–Lite Bridge 512–BIt Core does not support the Requester Interfaces of the Xilinx PCIe Core.
- This core does not support 64–bit AXI4–Lite Interfaces.
- This core supports only 512–bit data aligned mode of the Xilinx PCIe Core.

2.5 Generic Parameters

The generic parameters of the PCI Express to AXI4–Lite Bridge 512–Bit Core are described in Table 2–2. These parameters can be set as required by the user application while customizing the core.

Table 2-2: Generic Parameters

Port/Signal Name	Type	Description
num_addr_bits	Integers	**Number of Address Bits:** This parameter defines the address width of the AXI4–Lite Slave Interface for both read and write channels. It can range from 3 to 32.
bar0_size_bits		**BARx Address Size Bits:** The number of bits required to access the BARx address space where x = 0,1,2,3,4,5.
bar1_size_bits		
bar2_size_bits		
bar3_size_bits		
bar4_size_bits		
bar5_size_bits		
bar0_addr_translation	std_logic_ vector	**BARx Address Translation:** Base address from where the BARx address from the TLP will translate to an AXI4 Address where x = 0,1,2,3,4,5.
bar1_addr_translation		
bar2_addr_translation		
bar3_addr_translation		
bar4_addr_translation		
bar5_addr_translation		

Chapter 3: Port Descriptions

This chapter provides details about the port descriptions for the following interface types:

- AXI4–Lite Core Interfaces
- AXI4–Stream Core Interfaces
- I/O Signals

3.1 AXI4–Lite Core Interfaces

The PCI Express to AXI4–Lite Bridge 512–Bit Core has a 32–bit AXI4–Lite Master Inter–face, used to transfer data from the Completer Interfaces of the Xilinx PCIe Core to target AXI4–Lite Slave Interfaces.

Table 3–1 defines the ports in the AXI4–Lite Master Interface of the PCIe to AXI4–Lite Bridge 512–BIt Core. See the *AMBA AXI4–Lite Specification* for more details on opera–tion of the AXI4–Lite interfaces.

Table 3-1: AXI4-Lite Interface Port Descriptions

Port	Direction	Width	Description
m_axi_awaddr	Output	num_addr_ bits	**Write Address:** Address used for write operations. It must be valid when **m_axi_awvalid** is asserted and must be held until **m_axi_awready** is asserted by the slave. The address width is equal to the generic parameter (**num_addr_bits**) defined by the user while customizing the core.
m_axi_awprot		3	**Protection:** The PCIe to AXI4–Lite Bridge 512–BIt Core always outputs “000”.
m_axi_awvalid		1	**Write Address Valid:** This input must be asserted to indicate that a valid write address is available on **m_axi_awaddr**. The **m_axi_awvalid** signal must remain asserted until the rising clock edge after the assertion of **m_axi_awready**.
m_axi_awready	Input		**Write Address Ready:** This input from the slave indicates the readiness of the slave to accept the write address.The address is latched when **m_axi_awvalid** and **m_axi_awready** are high on the same cycle.

Table 3-1: AXI4-Lite Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
m_axi_wdata	Output	32	**Write Data:** This data will be written to the address specified by **m_axi_awaddr** when **m_axi_wvalid** and **m_axi_wready** are both asserted.
m_axi_wstrb		4	**Write Strobes:** This signal, when asserted, indicates the number of bytes of valid data on the **m_axi_wdata** signal. Each of these bits, when asserted, indicate that the corresponding byte of **m_axi_wdata** contains valid data. Bit 0 corresponds to the least significant byte, and bit 3 to the most significant.
m_axi_wvalid		1	**Write Valid:** This signal must be asserted to indicate that the write data is valid during the write operation. The value on **m_axi_wdata** is written into the register at address **m_axi_awaddr** when **m_axi_wready** and **m_axi_wvalid** are high on the same cycle.
m_axi_wready	Input		**Write Ready:** This signal is asserted by the slave when it is ready to accept data. The value on **m_axi_wdata** is written into the register at address **m_axi_awaddr** by the core when **m_axi_wready** and **m_axi_wvalid** are high on the same cycle, assuming that the address has already or simultaneously been submitted.
m_axi_bresp	Input	2	**Write Response:** The slave indicates success or failure of a write transaction through this signal, which is valid when **m_axi_bvalid** is asserted; 00 = Success of normal access 01 = Success of exclusive access 10 = Slave error 11 = Decode error Note: For more details about this signal refer to the *AMBA AXI Specification*.
m_axi_bready	Output	1	**Write Response Ready:** This signal is asserted by the PCIe to AXI4–Lite Bridge 512–BIt Core when it is ready to accept the Write Response.
m_axi_bvalid	Input		**Write Response Valid:** This signal is asserted by the slave when the write operation is complete and the Write Response is valid. It is held until **m_axi_bready** is asserted by the PCIe to AXI4–Lite Bridge 512–BIt Core.

Port	Direction	Width	Description
m_axi_araddr	Output	6	**Read Address:** Address used for read operations. It must be valid when **m_axi_arvalid** is asserted and must be held until **m_axi_arready** is asserted by the slave.
m_axi_arprot		3	**Protection:** These bits are always set to “000” by the PCIe to AXI4–Lite Bridge 512–BIt Core.
m_axi_arvalid		1	**Read Address Valid:** This input is asserted to indicate that a valid read address is available on **m_axi_araddr**. The slave asserts **m_axi_arready** when it is ready to accept the Read Address. The **m_axi_arvalid** signal must remain asserted until the rising clock edge after the assertion of **m_axi_arready.**
m_axi_arready	Input		**Read Address Ready:** This output is asserted by the slave when it is ready to accept the read address. The address is latched when **m_axi_arvalid** and **m_axi_arready** are high on the same cycle.
m_axi_rdata		32	**Read Data:** This value is the data read from the address specified by the **m_axi_araddr** when **m_axi_arvalid** and **m_axi_arready** are high on the same cycle.
m_axi_rresp		2	**Read Response:** The slave indicates success or failure of a read transaction through this signal, which is valid when **m_axi_rvalid** is asserted; 00 = Success of normal access 01 = Success of exclusive access 10 = Slave error 11 = Decode error Note: For more details about this signal refer to the *AMBA AXI Specification*.
m_axi_rvalid		1	**Read Data Valid:** This signal is asserted by the slave when the read is complete and the read data is available on **m_axi_rdata**. It is held until **m_axi_rready** is asserted by the PCIe to AXI4–Lite Bridge 512–BIt Core.
m_axi_rready	Output		**Read Data Ready:** This signal is asserted by the PCIe to AXI4–Lite Bridge 512–BIt Core when it is ready to accept the Read Data.

Table 3-1: AXI4-Lite Interface Port Descriptions (Continued)

3.2 AXI4–Stream Core Interfaces

The PCIe to AXI4–Lite Bridge 512–BIt Core has the following types of AXI4–Stream Interfaces, used to transfer and receive data streams.

- PCIe Completer Request (CQ) Interface: The interface through which all the memory read and write requests from the PCIe host are received. This interface is directly compatible with the Xilinx PCIe Core's Completer Request Interface in address–aligned mode.

- PCIe Completer Completion (CC) Interface: The interface through which completion TLPs for the memory read requests are transferred from the user design to the Xilinx PCIe Core. This interface is directly compatible with the Xilinx PCIe Core's Completer Completion Interface in address–aligned mode.

3.2.1 PCIe Completer Request (CQ) Interface

The PCIe Completer Request Interface is an AXI4–Stream Slave interface that is used to receive memory read and write requests from the Xilinx PCIe Core.

Table 3–2 defines the ports in the PCIe Completer Request Interface of the PCIe to AXI4–Lite Bridge 512–BIt Core. See the Completer Request section of the *Xilinx Gen3 Integrated Block for PCI Express product guide* for more details.

Table 3-2: PCIe Completer Request Interface Port Descriptions

Port	Direction	Width	Description
s_axis_cq_tdata	Input	512	**Completer Request Data Bus:** This input contains the completer request data from the Xilinx PCIe Core. It has a fixed width of 512 bits and is therefore only compatible with the 512–bit wide version of the Xilinx PCIe Core. This core assumes the data to be in the address–aligned mode.
s_axis_cq_tlast		1	**TLAST Indication for the Completer Request Data:** The Xilinx PCIe Core asserts this signal in the last cycle of a TLP to indicate the end of a packet.
s_axis_cq_tvalid			**Completer Request Data Valid:** The Xilinx PCIe Core asserts this signal to indicate valid data on the **s_axis_cq_tdata** bus. The Xilinx PCIe Core keeps this signal High during the transfer of a packet.
s_axis_cq_tuser		183	**Completer Request User Data:** This signal contains the sideband information for the TLP being received. This signal is valid when **s_axis_cq_tvalid** is High.

Table 3-2: PCIe Completer Request Interface Port Descriptions (Continued)

Port	Direction	Width	Description
s_axis_cq_tkeep	Input	16	**TKEEP Indication for the Completer Request Data:** The assertion of bit *i* of this bus during a transfer indicates that dword *i* (in this case a dword is 32 bits) of the **s_axis_cq_tdata** bus contains valid data. This bit is set to 1 contiguously for all dwords, starting from the first dword of the descriptor to the last dword of the payload. Thus, **s_axis_cq _tdata** is set to all 1s in all beats of a packet, except in the final beat when the total size of the packet is not a multiple of the width of data bus.
s_axis_cq_tready	Output	1	**Completer Request Ready:** This signal is asserted by the PCIe to AXI4–Lite Bridge 512–BIt Core to indicate that it is ready to accept the data from the Xilinx PCIe Core. Data is received across the interface when both **s_axis_cq_tready** and **s_axis_cq _tvalid** are High on the same cycle. If the PCIe to AXI4–Lite Bridge 512–BIt Core deasserts the ready signal when **s_axis_cq_tvalid** is High, the Xilinx PCIe Core completer request interface maintains the data on the bus and keeps the valid signal asserted until this core has asserted the ready signal.

3.2.2 PCIe Completer Completion (CC) Interface

The PCIe Completer Completion Interface is an AXI4–Stream Master Interface that is used to send completer memory read completions to the Xilinx PCIe Core.

Table 3–3 defines the ports in the PCIe Completer Completion Interface of the PCIe to AXI4–Lite Bridge 512–BIt Core. See the Completer Completion section of the *Xilinx Gen3 Integrated Block for PCI Express product guide* for more details.

Table 3-3: PCIe Completer Completion Interface Port Descriptions

Port	Direction	Width	Description
m_axis_cc_tdata	Output	512	**Completer Completion Data Bus:** This is the Completer completion data from the PCIe to AXI4–Lite Bridge 512–BIt Core to the Xilinx PCIe Core. It has a fixed width of 512 bits and therefore it is compatible only with the 512–bit wide version of the Xilinx PCIe Core and assumes address–aligned mode.
m_axis_cc_tlast		1	**TLAST Indication for the Completer Completion Data:** The PCIe to AXI4–Lite Bridge 512–BIt Core asserts this signal in the last cycle of a TLP to indicate the end of a packet.

Port	Direction	Width	Description
m_axis_cc_tvalid	Output	1	**Completer Completion Data Valid:** The PCIe to AXI4–Lite Bridge 512–BIt Core asserts this signal to indicate valid data on the **m_axis_cc_tdata** bus. This signal is kept High during the transfer of a packet.
m_axis_cc_tuser		81	**Completer Completion User Data:** This signal contains the sideband information for the TLP being transferred. This signal is valid when **m_axis_cc_tvalid** is High.
m_axis_cc_tkeep		16	**TKEEP Indication for the Completer Completion Data:** The assertion of bit *i* of this bus during a transfer indicates that dword *i* (in this case a dword is 32 bits) of the **m_axis_cc_ tdata** bus contains valid data. This bit is set to 1 contiguously for all dwords, starting from the first dword of the descriptor to the last dword of the payload. Thus, **m_axis_cc _tdata** is set to all 1s in all beats of a packet, except in the final beat when the total size of the packet is not a multiple of the width of the data bus.
m_axis_cc_tready	Input	4	**Completer Completion Ready:** This signal is asserted by the Xilinx PCIe Core to indicate that it is ready to accept the data. Data is transferred across the interface when both **m_axis_cc _tready** and **m_axis_cc_tvalid** are High on the same cycle. If the Xilinx PCIe Core deasserts the ready signal when **m_axis_cc_tvalid** is High, the PCIe to AXI4–Lite Bridge 512–BIt Core maintains the data on the bus and keeps the valid signal asserted, but data transfers are held off until the Xilinx PCIe core has re–asserted the ready signal, and transfers may resume.

Table 3-3: PCIe Completer Completion Interface Port Descriptions (Continued)

3.3 I/O Signals

The I/O port/signal descriptions of the top level module of the PCI Express to AXI4–Lite Bridge 512–Bit Core are discussed in Table 3–4.

Table 3-4: I/O Signals

Port/ Signal Name	Type	Direction	Description
user_clk	std_logic	Input	**User Clock (250 MHz):** Connect to the **user_clk** of the Xilinx PCIe Core.
user_reset			**Reset:** Active High. Synchronous with **user_clk.**
user_lnk_up			**Link Status:** Active High. Indicates whether the Xilinx PCIe Core is linked up with a host device.
aresetn		Output	**Reset:** Active low. This reset is defined based on **user_reset** and **user_lnk_up**. It is held Low until **user_reset** is deasserted, and **user_lnk_up** is asserted.
cc_shutdown_rqst		Input	**Completer Completion Interface Shutdown Request:** Active High.
cc_shutdown_ok		Output	**Completer Completion Interface Shutdown OK:** Active High. Indicates the response for a shutdown request. The PCIe to AXI4–Lite Bridge 512–BIt Core asserts this signal when all the current transactions in progress are complete.

This page is intentionally blank

Chapter 4: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the PCI Express to AXI4–Lite Bridge 512–Bit Core.

4.1 General Design Guidelines

The PCIe to AXI4–Lite Bridge 512–BIt Core provides the required logic to bridge the CC and CQ Interfaces of the Xilinx PCIe Core to the AXI4–Lite Interface. The user can customize the PCIe to AXI4–Lite Bridge 512–BIt Core by defining the generic parameters as described in Section 2.5. The data width of the AXI4–Lite Master Interface is fixed to 32 bits, and the data width of the CC and CQ interfaces is fixed to 512 bits. The data width of the AXI4–Stream interfaces of the PCIe to AXI4–Lite Bridge 512–BIt Core must match the CC and CQ interfaces' data widths of the Xilinx PCIe Core.

The PCIe to AXI4–Lite Bridge 512–BIt Core responds with Unsupported Request (UR) status for an unsupported request received on the Completer Request Interface from the Xilinx PCIe Core. It responds with a three–dword completion descriptor followed by the request that generated the completion. For more details on unsupported requests please refer to the *Xilinx Gen3 Integrated Block for PCI Express product guide*.

4.1.1 Address Translation

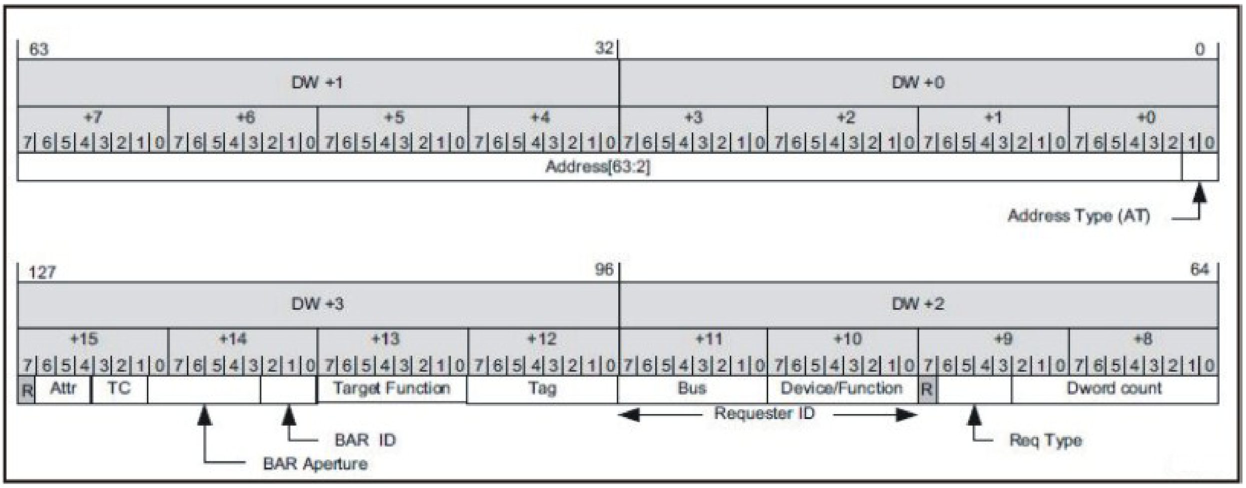
The address from the TLP is provided in the CQ descriptor from the Xilinx PCIe Core. The address is provided in the dwords 0 and 1 as shown in the Figure 4–1. The address provided in the descriptor comes directly from the PCI Express TLP.

The address translation takes place based on the generic parameters defined by the user. The BAR address translation value provides the Base Address from where a BAR hit will translate, in the AXI4–Lite address space. The address size defined by the **`num_addr_bits`** generic parameter determines the address width of the AXI4–Lite Interface. The BAR address size is used to mask the additional bits of the TLP address to match the address width defined by the **`num_addr_bits`** parameter. The resultant address is thus determined using the masked address and the BAR address translation.

4.1 General Design Guidelines (continued)

4.1.1 Address Translation (continued)

Figure 4–1: PCI Express to AXI4–Lite Bridge 512–Bit Core Interface Example



4.2 Clocking

Main Clock: **`user_clk`**

The 250 MHz user clock is used to clock all the ports on the PCIe to AXI4–Lite Bridge 512–BIt Core.

4.3 Resets

Main reset: **`user_reset`**

This is a synchronous reset associated with the **`user_clk`**. This reset along with the **`user_lnk_up`** is used to derive the reset for the PCIe to AXI4–Lite Bridge 512–BIt Core.

4.4 Interrupts

This section is not applicable to this IP core.

4.5 Interface Operation

AXI4–Lite Interface: This core includes an AXI4–Lite Master interface which is described in **Section 3.1**.

PCIe Completer Request Interface: This interface is associated with the `user_clk`. It is an AXI4–Stream Slave Interface directly compatible with the Xilinx PCIe Core's Completer Request Bus when the Xilinx PCIe Core is set up in the data–aligned mode with a 512–bit wide data bus. For details about this interface's bit definitions refer to Section 3.2.1.

PCIe Completer Completion Interface: This interface is associated with the `user_clk`. It is an AXI4–Stream Master Interface directly compatible with the Xilinx PCIe core's Completer Completion Bus when the Xilinx PCIe Core is set up in the data–aligned mode with a 512–bit wide data bus. For details about this interface's bit definitions refer to Section 3.2.2.

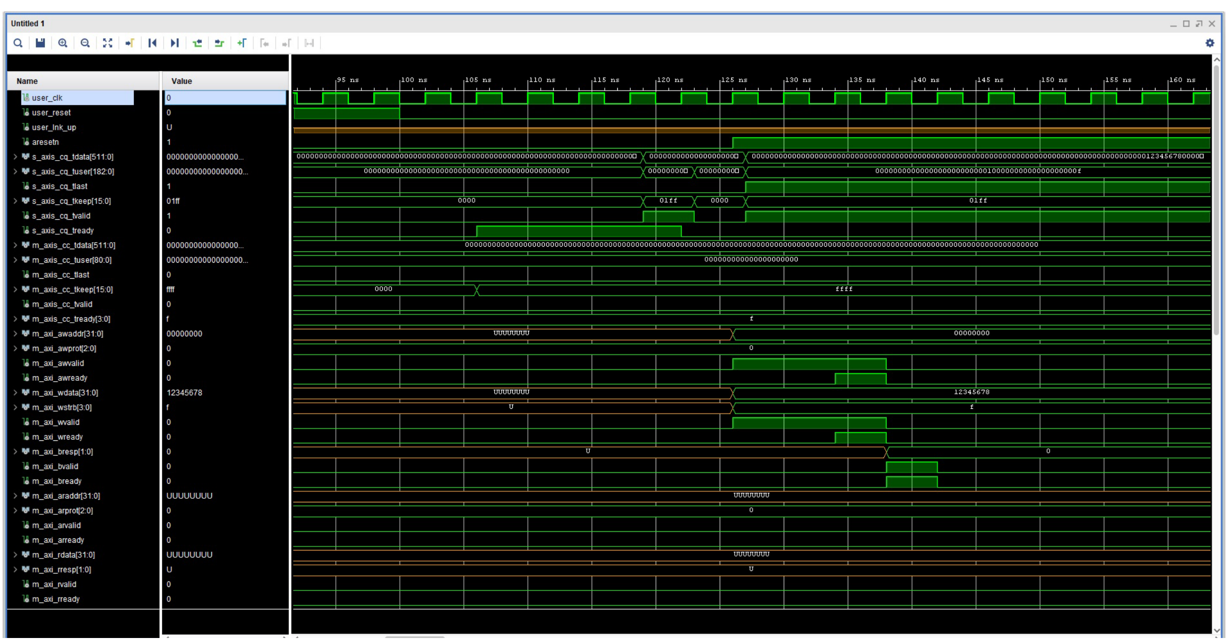
4.6 Programming Sequence

This section is not applicable to this IP core.

4.7 Timing Diagrams

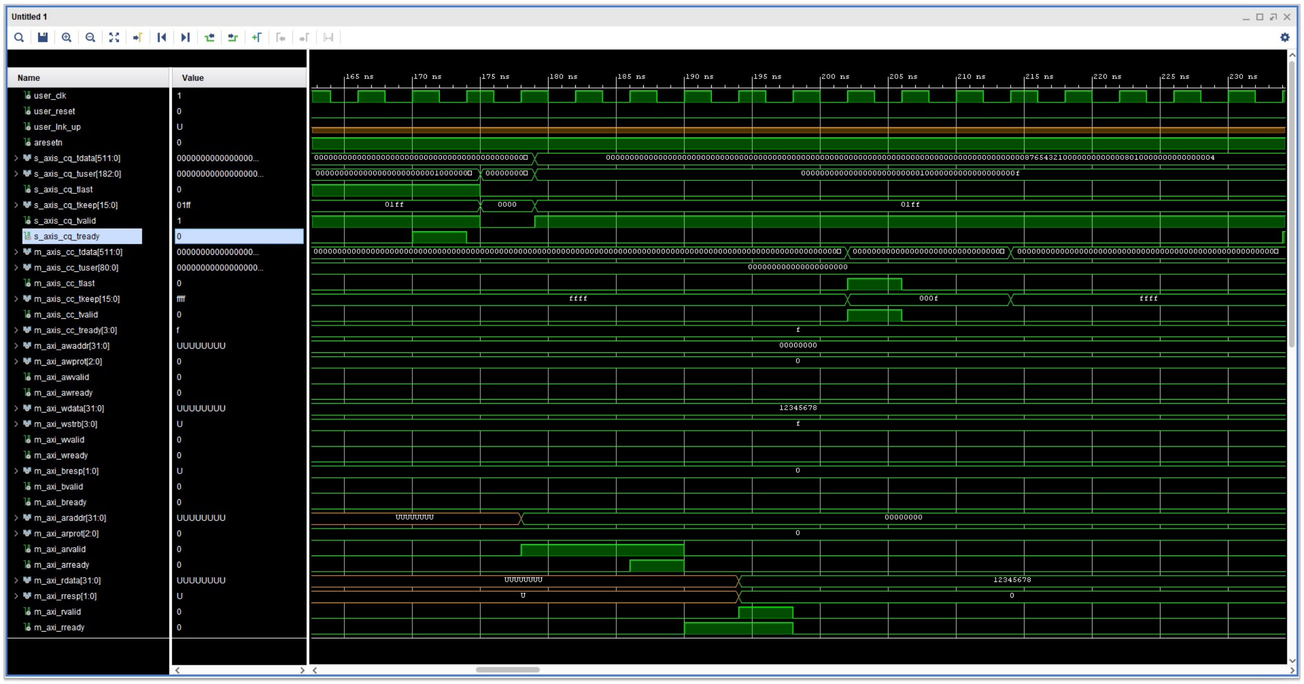
The timing diagrams for the PCI Express to AXI4–Lite Bridge 512–Bit Core are obtained by running the simulation of the test bench for the core in Vivado VSim environment. These timing diagrams depict the functionality of the core for memory read and write operations. A detailed explanation of the test bench is in Section 5.5 of this user manual. The Figures 4–2 and 4–3 show the write and read operation timing respectively.

Figure 4–2: PCI Express to AXI4–Lite Bridge 512–Bit Core – Write Operation



4.7 Timing Diagrams (continued)

Figure 4–3: PCI Express to AXI4–Lite Bridge 512–Bit Core – Read Operation

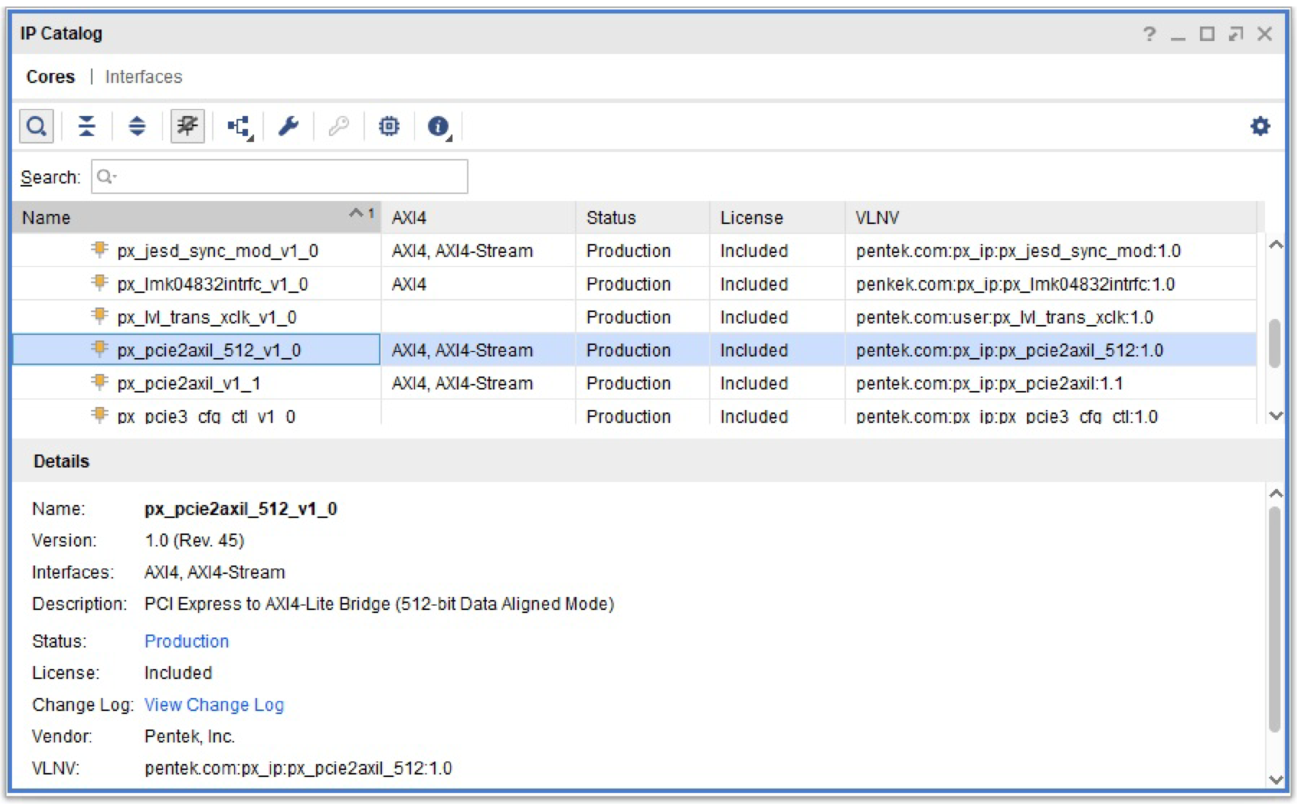


Chapter 5: Design Flow Steps

5.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek PCI Express to AXI4–Lite Bridge 512–Bit Core. It also includes simulation, synthesis, and implementation steps that are specific to this IP core. This core can be generated from the Vivado IP Catalog when the Pentek IP Repository has been installed. It will appear in the IP Catalog list as **px_pcie2axil_512_v1_0** as shown in Figure 5–1.

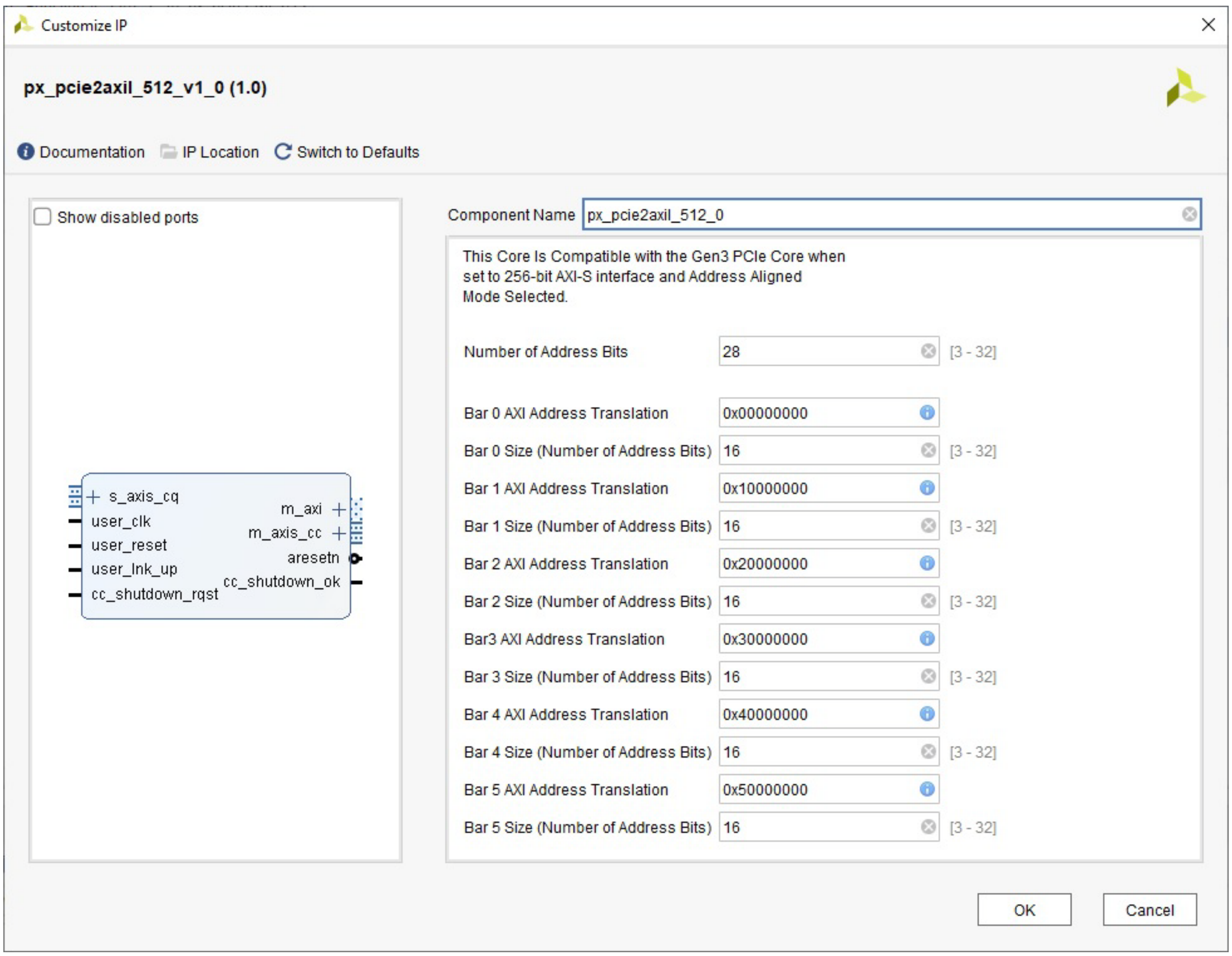
Figure 5–1: PCI Express to AXI4–Lite Bridge 512–Bit Core in Pentek IP Catalog



5.1 Pentek IP Catalog (continued)

When you select the **px_pcie2axil_512_v1_0** core, a screen appears that shows the core's symbol and the core's parameters (see Figure 5–2). The core's symbol is the box on the left side.

Figure 5–2: PCI Express to AXI4–Lite Bridge 512–Bit Core IP Symbol



5.2 User Parameters

The user parameters of this core are described in Section 2.5 of this user manual.

5.3 Generating Output

For more details about generating and using IP in the Vivado Design Suite, refer to the *Vivado Design Suite User Guide – Designing with IP*.

5.4 Constraining the Core

This section contains information about constraining the PCI Express to AXI4–Lite Bridge 512–Bit Core in Vivado Design Suite.

Required Constraints

The XDC constraints are not provided with the PCI Express to AXI4–Lite Bridge 512–Bit Core. Clock constraints can be applied in the top–level module of the user design.

Device, Package, and Speed Grade Selections

This IP works for Ultrascale+ FPGAs.

Clock Frequencies

The clock frequency (**`user_clk`**) for this IP core is 250 MHz.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking and Placement

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

5.5 Simulation

The PCI Express to AXI4–Lite Bridge 512–Bit Core has a test bench which generates the output waveforms using the Vivado VSim environment.

The test bench is designed to run at 250 MHz input clock frequency. It performs both memory read and memory write operations. The read/write requests from the PCIe host are received across the Completer Request Interface. During write operations, this core writes data to the specified address by translating the write request into AXI4–Lite Interface format. During read operations, this core reads data from the target AXI4–Lite Slave through the AXI4–Lite Master Interface and generates completion packets which are transferred through the Completer Completion Interface to the PCIe host. This testbench performs three write, and three read operations. When run, the simulation produces the results shown in Figure 5–3.

Figure 5–3: PCI Express to AXI4–Lite Bridge 512–Bit Core Test Bench Simulation Output

5.6 Synthesis and Implementation

For details about synthesis and implementation see the *Vivado Design Suite User Guide – Designing with IP*.

This page is intentionally blank