# IP CORE MANUAL


## AXI4-Stream Multiplexer IP

**px_axis_pdti_mux**

PENTEK

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



Manual Part Number: 807.48312 Rev: 1.0 - December 09, 2016

## Manual Revision History

| Date | Version | Comments |
|---|---|---|
| 12/09/16 | 1.0 | Initial Release |

## Legal Notices



## Copyright



## Trademarks



Printed in the United States of America.

# *Table of Contents*

***Page***

# Table of Contents

# *IP Facts*

## Description

Pentek Navigator™ AXI4-Stream Multiplexer Core generates an AXI4-Stream output from upto 8 incoming AXI4-Stream inputs by implementing a n:1 Multiplexer (where n = number of input data streams). The selection of the AXI Stream input to be mapped to the output is based on the register bits of the Selection Control Register within the core.

This core complies with the ARM® AMBA® AXI4 Specification. This user manual defines the hardware interface, software interface, and parameterization options for the AXI4-Stream Multiplexer Core.

## Features

- Software programmable number of input data streams
- Software programmable width of input data stream
- Supports up to 16 bytes wide input data stream
- AXI4-Streams across input and output ports follow a format that combines sample data with its time-aligned timestamp and data information
- Supports addition of input registers

**Table 1-1: IP Facts Table**

| | |
|---|---|
| **Core Specifics** | |
| Supported Design Family[a] | Kintex® 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 | HAL Software Support |
| **Tested Design Flows** | |
| Design Entry | Vivado® Design Suite 2016.3 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

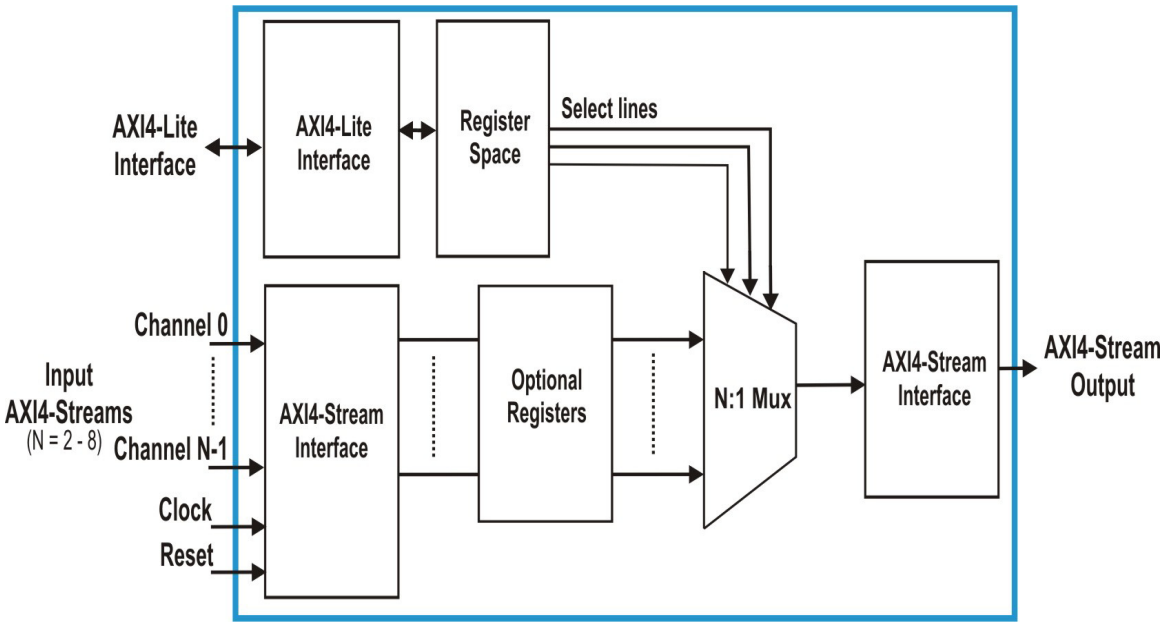
The AXI4-Stream Multiplexer Core accepts upto 8 input AXI4-Streams through an AXI4-Stream Slave Interface and delivers an output AXI4-Stream through an AXI4-Stream Master Interface. This core implements a n:1 multiplexer, where 'n' is the number of input data streams, which is used to map one of the input streams to the output.

The input AXI4-Stream to be selected is based on the **select bits** of the **Selection Control Register** within the Register Space of this core. The Register Space of the core can be accessed through an AXI4-Lite Interface.

The width of the input data stream can be defined using the generic parameter **data_byte_width** (See Section 2.5). The input and output AXI4-Streams for this core are combined Sample Data/ Timestamp/ Information Streams. This type of data streams combine sample data with its time-aligned timestamp and data information (See Section 2.6.2).

Figure 1-1 is a top-level block diagram of the Pentek AXI4-Stream Multiplexer Core. The modules within the block diagram are explained in the later sections of this manual.

**Figure 1-1: AXI4-Stream Multiplexer Core Block Diagram**



## 1.1 Functional Description (continued)

- **AXI4-Stream Interface:** The AXI4-Stream Multiplexer Core has three AXI4-Stream Interfaces. At the input, two AXI4-Stream Slave Interfaces are used to receive AXI4-Streams and at the output an AXI4-Stream Master Interface is used to transfer AXI4-Streams through the output ports. For more details about the AXI4-Stream Interfaces refer to Section 3.2 AXI4-Stream Core Interfaces.

- **AXI4-Lite Interface:** This core implements a 32-bit AXI4-Lite Slave Interface to access the Register Space. For more details about the AXI4-Lite Interface, refer to Section 3.1 AXI4-Lite Core Interfaces.

- **Registers:** This is an optional module present after the AXI4-Stream Slave Interface which can be enabled (or disabled) by the generic parameter **add_input_regs** (See Section 2.5). It is used to add input registers within the core.

- **Register Space:** This module contains the Selection Control Register which is used to control the select bits of the multiplexer. It can be accessed through the AXI4-Lite interface.

- **Multiplexer:** This is a n:1 multiplexer (where n = number of input data streams) within the core, which is used to generate the output AXI4-Stream from input AXI4-Streams based on the select bits.

## 1.2 Applications

The AXI4-Stream Multiplexer Core can be incorporated into any Kintex Ultrascale FPGA where one output AXI4-Stream is to be selected from 2 to 8 input AXI4-Streams.

## 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 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

***This page is intentionally blank***

# *Chapter 2: General Product Specifications*

## 2.1 Standards

The AXI4-Stream Multiplexer Core has bus interfaces that comply with the A*RM AMBA AXI4-Lite Protocol Specification* and the *AMBA AXI4-Stream Protocol Specification*.

## 2.2 Performance

The performance of the AXI4-Stream Multiplexer 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 AXI4-Stream Multiplexer Core has two incoming clock signals. The AXI4-Stream clock has a maximum frequency of 500 MHz while the clock across the AXI4-Lite interface has a maximum frequency of 250 MHz on a Kintex Ultrascale -2 speed grade FPGA. 250 MHz is typically the PCI Express® (PCIe®) AXI bus clock frequency.

## 2.3 Resource Utilization

The resource utilization of the AXI4-Stream Multiplexer Core is shown in Table 2-1. Resources have been estimated for the Kintex Ultrascale XCKU060 -2 speed grade device. These values were generated using the Vivado Design Suite.

**Table 2-1: Resource Usage and Availability**

| Resource | # Used |
|---|---|
| LUTs | 100 |
| Flip-Flops | 176 |

**NOTE:** Actual utilization may vary based on the user design in which the AXI4-Stream Multiplexer Core is incorporated.

## 2.4 Limitations and Unsupported Features

This section is not applicable to this IP core.

## 2.5 Generic Parameters

The generic parameters of the AXI4-Stream Multiplexer 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 |
|---|---|---|
| **data_byte_width** | Integer | **Data Byte Width:** This parameter indicates the width of the input data streams across the AXI4-Stream Slave Interface in bytes. It can range from 1 to 16 bytes. |
| **num_input_streams** | | **Number of Input Streams:** This parameter indicates the number of input AXI4-Streams to the core. It can range from 2 to 8. |
| **add_input_regs** | Boolean | **Add Input Registers:** This parameter is used enable or disable the addition of input registers to the core to store the incoming AXI4-Stream data. |

# 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

## 3.1 AXI4-Lite Core Interfaces

The AXI4-Stream Multiplexer Core uses the Control/Status Register (CSR) interface to control, and receive status from, the user design.

### 3.1.1 Control/Status Register (CSR) Interface

The CSR Interface is an AXI4-Lite Slave Interface that can be used to access the control and status registers in the AXI4-Stream Multiplexer Core. Table 3-1 defines the ports in the CSR interface. See the *AMBA AXI4-Lite Specification* for more details on operation of the AXI4-Lite interfaces.

**Table 3-1: Control/Status Register (CSR) Interface Port Descriptions**

| Port | Direction | Width | Description |
|---|---|---|---|
| **s_axi_csr_aclk** | Input | 1 | **Clock** |
| **s_axi_csr_aresetn** | Input | 1 | **Reset:** Active low. This signal will reset the control register to it's initial state. |
| **s_axi_csr_awaddr** | Input | 3 | **Write Address:** Address used for write operations. It must be valid when **s_axi_csr_awvalid** is asserted and must be held until **s_axi_csr_awready** is asserted by the AXI4-Stream Multiplexer Core. |
| **s_axi_csr_awprot** | Input | 3 | **Protection:** The AXI4-Stream Multiplexer Core ignores these bits. |
| **s_axi_csr_awvalid** | Input | 1 | **Write Address Valid:** This input must be asserted to indicate that a valid write address is available on **s_axi_csr_awaddr**. The AXI4-Stream Multiplexer Core asserts **s_axi_csr_awready** when it is ready to accept the address. The **s_axi_csr_awvalid** must remain asserted until the rising clock edge after the assertion of **s_axi_csr_awready**. |

**Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)**

| Port | Direction | Width | Description |
|---|---|---|---|
| **s_axi_csr_awready** | Output | 1 | **Write Address Ready:** This output is asserted by the AXI4-Stream Multiplexer Core when it is ready to accept the write address.The address is latched when **s_axi_csr_awvalid** and **s_axi_csr_awready** are high on the same cycle. |
| **s_axi_csr_wdata** | Input | 32 | **Write Data:** This data will be written to the address specified by **s_axi_csr_awaddr** when **s_axi_csr_wvalid** and **s_axi_csr _wready** are both asserted. The value must be valid when **s_axi_csr_wvalid** is asserted and held until **s_axi_csr_wready** is also asserted. |
| **s_axi_csr_wstrb** | Input | 4 | **Write Strobes:** This signal, when asserted, indicates the number of bytes of valid data on the **s_axi_csr_wdata** signal. Each of these bits, when asserted, indicate that the corresponding byte of **s_axi_csr_wdata** contains valid data. Bit 0 corresponds to the least significant byte, and bit 3 to the most significant. |
| **s_axi_csr_wvalid** | Input | 1 | **Write Valid:** This signal must be asserted to indicate that the write data is valid for a write operation. The value on **s_axi_csr _wdata** is written into the register at address **s_axi_csr_awaddr** when **s_axi_csr_wready** and **s_axi_csr_wvalid** are high on the same cycle. |
| **s_axi_csr_wready** | Output | 1 | **Write Ready:** This signal is asserted by the AXI4-Stream Multiplexer Core when it is ready to accept data. The value on **s_axi_csr_wdata** is written into the register at address **s_axi_csr_awaddr** when **s_axi_csr_wready** and **s_axi_csr_wvalid** are high on the same cycle, assuming that the address has already or simultaneously been submitted. |
| **s_axi_csr_bresp** | Output | 2 | **Write Response:** The AXI4-Stream Multiplexer Core indicates success or failure of a write transaction through this signal, which is valid when **s_axi_csr_bvalid** is asserted;<br>00 = Success of normal access<br>01 = Success of exclusive access<br>10 = Slave Error<br>11 = Decode Error<br>Note: For more details about this signal refer to the *AMBA AXI Specification*. |
| **s_axi_csr_bready** | Input | 1 | **Write Response Ready:** This signal must be asserted by the user logic when it is ready to accept the Write Response. |
| **s_axi_csr_bvalid** | Output | 1 | **Write Response Valid:** This signal is asserted by the AXI4-Stream Multiplexer Core when the write operation is complete and the Write Response is valid. It is held until **s_axi_csr_bready** is asserted by the user logic. |

**Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)**

| Port | Direction | Width | Description |
|---|---|---|---|
| **s_axi_csr_araddr** | Input | 3 | **Read Address:** Address used for read operations. It must be valid when **s_axi_csr_arvalid** is asserted and must be held until **s_axi_csr_arready** is asserted by the AXI4-Stream Multiplexer Core. |
| **s_axi_csr_arprot** | Input | 3 | **Protection:** These bits are ignored by the AXI4-Stream Multiplexer Core. |
| **s_axi_csr_arvalid** | Input | 1 | **Read Address Valid:** This input must be asserted to indicate that a valid read address is available on **s_axi_csr_araddr**. The core asserts **s_axi_csr_arready** when it ready to accept the Read Address. This input must remain asserted until the rising clock edge after the assertion of **s_axi_csr_arready**. |
| **s_axi_csr_arready** | Output | 1 | **Read Address Ready:** This output is asserted by the AXI4-Stream Multiplexer Core when it is ready to accept the read address. The address is latched when **s_axi_csr_arvalid** and **s_axi_csr_arready** are high on the same cycle. |
| **s_axi_csr_rdata** | Output | 32 | **Read Data:** This value is the data read from the address specified by the **s_axi_csr_araddr** when **s_axi_csr_arvalid** and **s_axi_csr_arready** are high on the same cycle. |
| **s_axi_csr_rresp** | Output | 2 | **Read Response:** The AXI4-Stream Multiplexer Core indicates success or failure of a read transaction through this signal, which is valid when **s_axi_csr_rvalid** is asserted;<br>00 = Success of normal access<br>01 = Success of exclusive access<br>10 = Slave Error<br>11 = Decode Error<br>Note: For more details about this signal refer to the *AMBA AXI Specification*. |
| **s_axi_csr_rvalid** | Output | 1 | **Read Data Valid:** This signal is asserted by the AXI4-Stream Multiplexer Core when the read is complete and the read data is available on **s_axi_csr_rdata**. It is held until **s_axi_csr_rready** is asserted by the user logic. |
| **s_axi_csr_rready** | Input | 1 | **Read Data Ready:** This signal is asserted by the user logic when it is ready to accept the Read Data. |

## 3.2 AXI4-Stream Core Interfaces

The AXI4-Stream Multiplexer Core has the following AXI4-Stream Interfaces, used to receive and transfer data streams.

- Combined Sample Data/ Timestamp/ Information Streams (PDTI) Interface**:** This core implements two of these AXI4-Stream interfaces across the input and output to receive and transfer data streams**.**

### 3.2.1 Combined Sample Data/ Timestamp/ Information Streams (PDTI) Interface

Pentek's Jade series board products have AXI4-Streams that follow a combined Sample Data/ Timestamp/ Information Stream format. This type of data stream combines sample data with its time aligned timestamp and data information. There is an AXI4-Stream Slave Interface across the input to receive AXI4-Streams, and an AXI4-Stream Master Interface across the output to transfer data streams.

Table 3-2, defines the ports in the AXI4-Stream Sample Data/ Timestamp/ Information Stream Interfaces. See the *AMBA AXI4-Stream Specification* for more details on the operation of the AXI4-Stream Interfaces.

**Table 3-2: Combined Sample Data/ Timestamp/ Information Streams Interface Port Descriptions**

| Port | Direction | Width | Description |
|---|---|---|---|
| **AXI4-Stream Slave Interface** | | | |
| **aclk** | Input | 1 | **AXI4-Stream Clock** |
| **aresetn** | | 1 | **Reset:** Active Low. |
| **s_axis_i0_pdti_tdata** | | depends on the generic parameter **data_byte_width** | **Channel X Input Data:** This is the input data across channel X where X = 0,1,2,3,4,5,6,7. |
| **s_axis_i1_pdti_tdata** | | | |
| **s_axis_i2_pdti_tdata** | | | |
| **s_axis_i3_pdti_tdata** | | | |
| **s_axis_i4_pdti_tdata** | | | |
| **s_axis_i5_pdti_tdata** | | | |
| **s_axis_i6_pdti_tdata** | | | |
| **s_axis_i7_pdti_tdata** | | | |

**Table 3-2: Combined Sample Data/ Timestamp/ Information Streams Interface Port Descriptions (Continued)**

| Port | Direction | Width | Description |
|---|---|---|---|
| **AXI4-Stream Slave Interface (continued)** | | | |
| **s_axis_i0_pdti_tvalid** | Input | 1 | **Channel X Input Data Valid:** Asserted when data is valid on **s_axis_ix_pdti_tdata** where X = 0,1,2,3,4,5,6,7. |
| **s_axis_i1_pdti_tvalid** | | | |
| **s_axis_i2_pdti_tvalid** | | | |
| **s_axis_i3_pdti_tvalid** | | | |
| **s_axis_i4_pdti_tvalid** | | | |
| **s_axis_i5_pdti_tvalid** | | | |
| **s_axis_i6_pdti_tvalid** | | | |
| **s_axis_i7_pdti_tvalid** | | | |
| **s_axis_i0_pdti_tuser** | Input | 128 | **Channel X Sideband Information:** This is the user defined sideband information transmitted alongside the data stream where X= 0,1,2,3,4,5,6,7.<br>**tuser [63:0] - Timestamp[63:0]**<br>**tuser [71:64] - Gate Positions**<br>**tuser [79:72] - Sync Positions**<br>**tuser [87:80] - PPS Positions**<br>**tuser [91:88] - Samples per clock cycle**<br>**tuser[92] - I/Q data of the sample => 0 = I; 1 = Q**<br>**tuser [94:93] - Data Format => 0 = 8-bit; 1 = 16-bit; 2 = 24-bit; 3 = 32-bit**<br>**tuser [95] - Data Type => 0 = Real; 1 = I/Q**<br>**tuser [103:96] - channel [7:0]**<br>**tuser [127:104] - Reserved**<br>Note: The bits [103:96] define the channel number in the user design from where the data is being received. |
| **s_axis_i1_pdti_tuser** | | | |
| **s_axis_i2_pdti_tuser** | | | |
| **s_axis_i3_pdti_tuser** | | | |
| **s_axis_i4_pdti_tuser** | | | |
| **s_axis_i5_pdti_tuser** | | | |
| **s_axis_i6_pdti_tuser** | | | |
| **s_axis_i7_pdti_tuser** | | | |

**Table 3-2: Combined Sample Data/ Timestamp/ Information Streams Interface Port Descriptions (Continued)**

| Port | Direction | Width | Description |
|---|---|---|---|
| **AXI4-Stream Master Interface** | | | |
| **m_axis_pdti_tdata** | Output | depends on the generic parameter **data_byte_width** | **Output Data:** This is the output data from the AXI4-Stream Multiplexer Core. |
| **m_axis_pdti_tvalid** | | 1 | **Output Data Valid:** Asserted when data is valid on **m_axis_pdti_tdata**. |
| **m_axis_pdti_tuser** | | 128 | **Output Sideband Information:** This is the user defined sideband information transmitted alongside the data stream.<br>**tuser [63:0] - Timestamp[63:0]**<br>**tuser [71:64] - Gate Positions**<br>**tuser [79:72] - Sync Positions**<br>**tuser [87:80] - PPS Positions**<br>**tuser [91:88] - Samples per clock cycle**<br>**tuser[92] - I/Q data of the sample => 0 = I; 1 = Q**<br>**tuser [94:93] - Data Format => 0 = 8-bit; 1 = 16-bit; 2 = 24-bit; 3 = 32-bit**<br>**tuser [95] - Data Type => 0 = Real; 1 = I/Q**<br>**tuser [103:96] - channel [7:0]**<br>**tuser [127:104] - Reserved**<br>Note: The bits [103:96] define the channel number in the user design from where the data is being received. |

# Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the register space of the AXI4-Stream Multiplexer Core. The memory map is provided in Table 4-1.

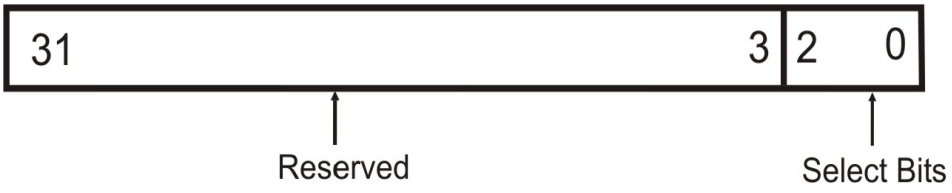
**Table 4-1: Register Space Memory Map**

| Register Name | Address (Base Address +) | Access | Description |
|---|---|---|---|
| **Selection Control Register** | 0x00 | R/W | Controls the selection bits of the multiplexer. |

## 4.1 Selection Control Register

This register controls the select bits of the multiplexer which determine the output of the core. The Selection Control Register is illustrated in Figure 4-1 and described in Table 4-2.

**Figure 4-1: Selection Control Register**



## 4.1 Selection Control Register (continued)

**Table 4-2: Selection Control Register (Base Address + 0x00)**

| Bits | Field Name | Default Value | Access Type | Description |
|---|---|---|---|---|
| **31:3** | Reserved | N/A | N/A | **Reserved** |
| **2:0** | select_bits | 000 | R/W | **Select Bits:** These bits are the select bits of the multiplexer and determine which input channel data of the core that is to be mapped to the output.<br>000 = Channel 0<br>001 = Channel 1<br>010 = Channel 2<br>011 = Channel 3<br>100 = Channel 4<br>101 = Channel 5<br>110 = Channel 6<br>111 = Channel 7 |

# *Chapter 5: Designing with the Core*

This chapter includes guidelines and additional information to facilitate designing with the AXI4-Stream Multiplexer Core.

## 5.1 General Design Guidelines

The AXI4-Stream Multiplexer Core provides the required logic to generate an AXI Stream output from incoming AXI Streams. This IP core supports AXI4-Lite and AXI4-Stream user interfaces. The user can customize the core by setting the generic parameters based on the application requirement as described in Section 2.5. By adding input registers to the core, latency of the core is increased to 2 clock cycles of the AXI4-Stream clock (**s_axis_aclk**).

## 5.2 Clocking

AXI4-Stream Clock: **s_axis_aclk**

This clock is used to clock the input and output ports across the AXI4-Stream Interfaces of the core.

CSR Clock: **s_axi_csr_aclk**

This clock is used to clock the AXI4-Lite interface of the core.

## 5.3 Resets

Main reset: **s_axis_aresetn**

This is an active low reset synchronous with **s_axis_aclk**.

CSR Reset: **s_axi_csr_aresetn**

This is an active low reset synchronous with **s_axi_csr_clk**.

## 5.4 Interrupts

This section is not applicable to this IP core.

## 5.5 Interface Operation

**CSR Interface:** This is the Control/Status Register Interface and is associated with **s_axi_csr_aclk**. It is a standard AXI4-Lite Slave interface.

**Combined Sample Data/ Timestamp/ Information Streams (PDTI) Interfaces:** This core implements two of these AXI4-Stream interfaces across the input and output to receive and transfer data streams and is associated with **s_axis_aclk**. For more details about this interface refer to Section 3.2.1.

## 5.6 Programming Sequence

This section briefly describes the programming sequence for the AXI4-Stream Multiplexer Core.

1) Assign desired values to the generic parameter.

2) Set the Selection Control Register with the required value.

3) Observe the outputs across the outputs ports.

## 5.7 Timing Diagrams

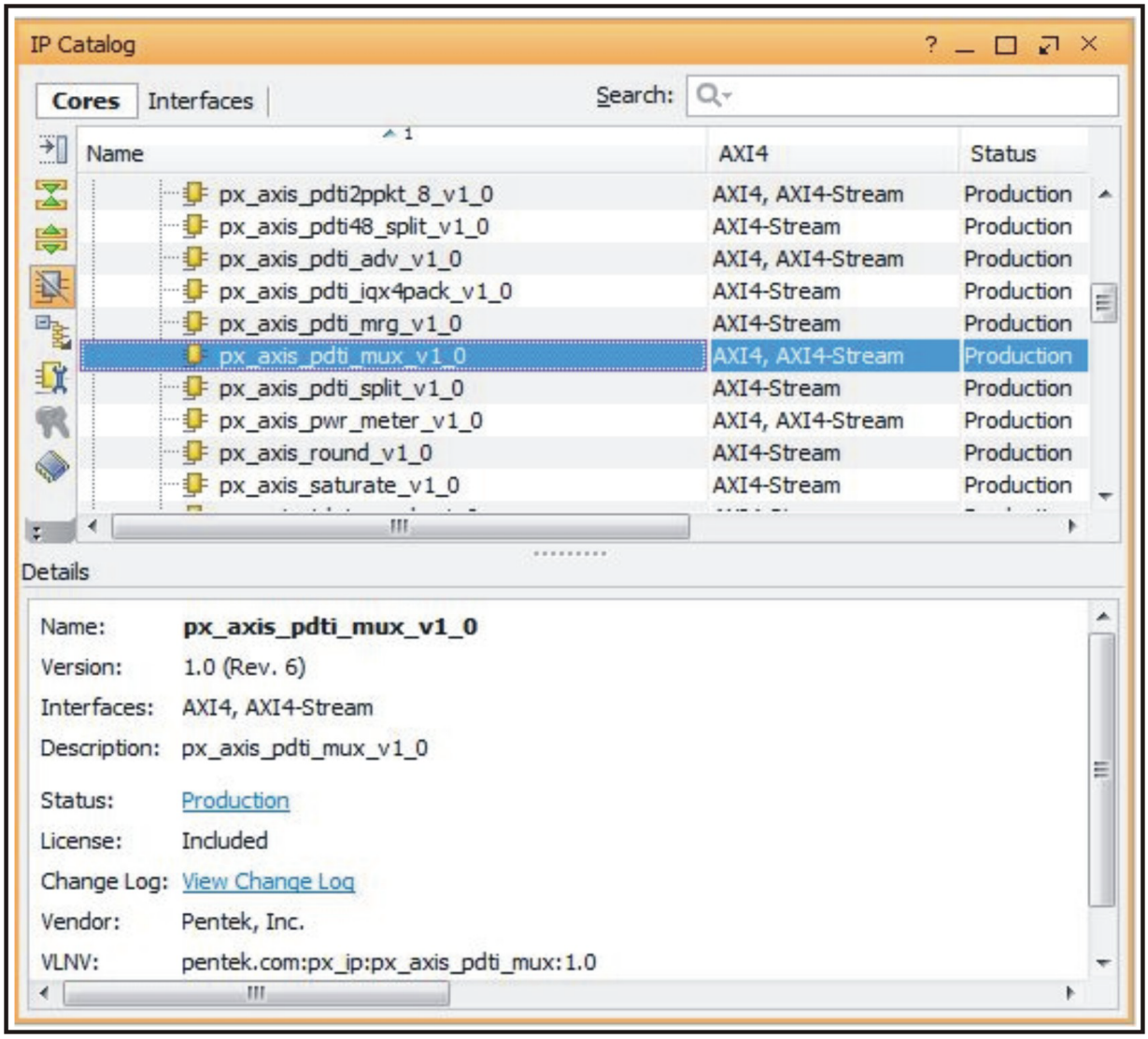
This section is not applicable to this IP core.

# Chapter 6: Design Flow Steps

## 6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek AXI4-Stream Multiplexer 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_axis_pdti_mux_v1_0** as shown in Figure 6-1.

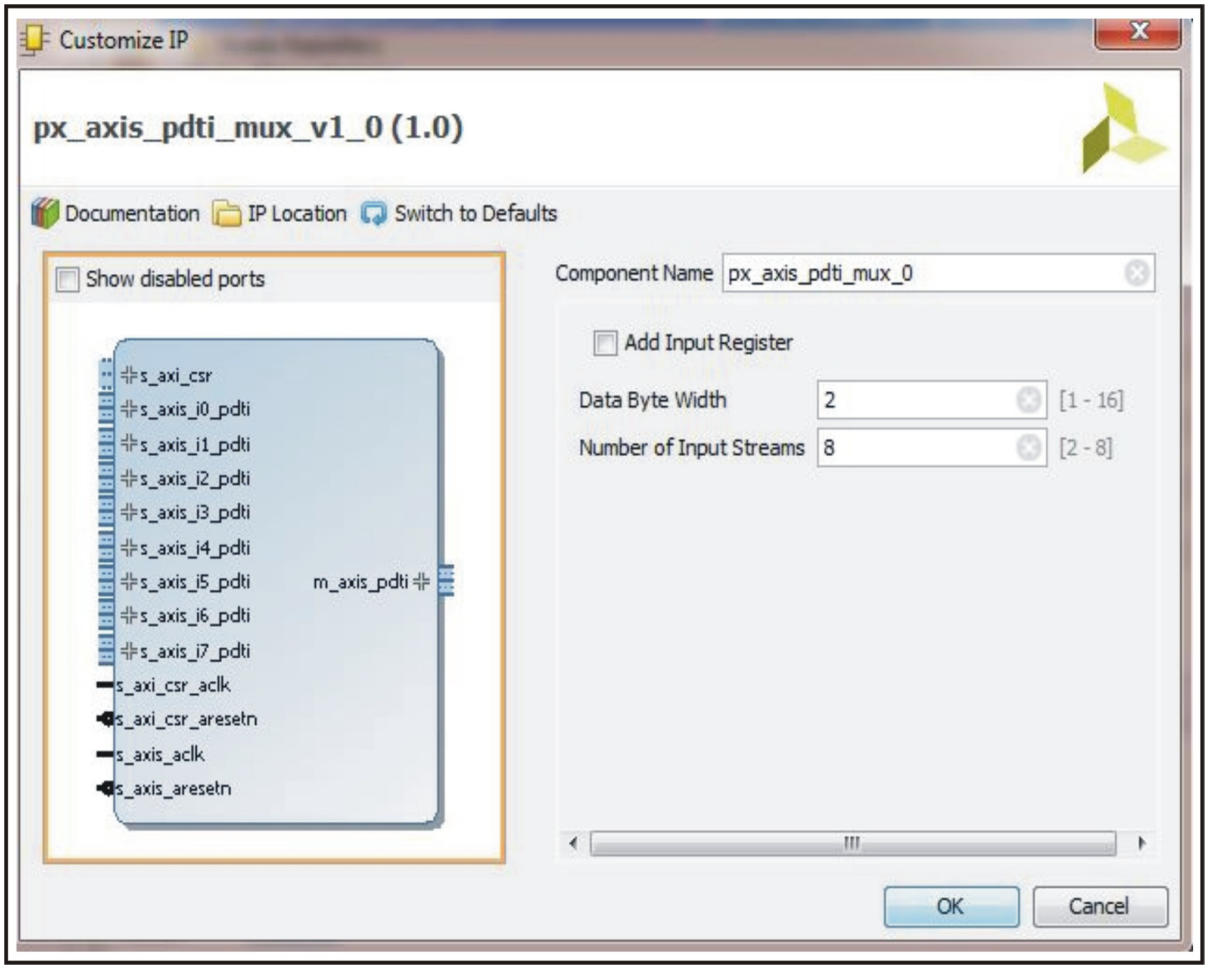
**Figure 6-1: AXI4-Stream Multiplexer Core in Pentek IP Catalog**



## 6.1 Pentek IP Catalog (continued)

When you select the **px_axis_pdti_mux_v1_0** core, a screen appears that shows the core's symbol and the core's parameters (see Figure 6-2). The core's symbol is the box on the left side.

**Figure 6-2: AXI4-Stream Multiplexer Core IP Symbol**




## 6.2 User Parameters

The user parameter of this core is described in Section 2.5 of this user manual.

## 6.3 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.*

## 6.4 Constraining the Core

This section contains information about constraining the AXI4-Stream Multiplexer Core in Vivado Design Suite.

**Required Constraints**

The XDC constraints are not provided with the AXI4-Stream Multiplexer 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 the Kintex Ultrascale FPGAs.

**Clock Frequencies**

The clock (**s_axi_csr_aclk**) can take frequencies up to 250 MHz. The AXI4-Stream clock (**s_axis_aclk**) has a maximum frequency of 500 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.

## 6.5 Simulation

This section is not applicable to this IP core.

## 6.6 Synthesis and Implementation

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

***This page is intentionally blank***