IP CORE MANUAL

AXI4–Stream Packet to Contiguous Stream 512–Bit IP

px_axis_pkt2contigstream_512

PENTEK

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

Manual Part Number:807.48441 Rev: 1.0 – October 14, 2019

Manual Revision History

Date	Version	Comments
10/14/19	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's Navigator™ AXI4–Stream Packet to Contiguous Stream 512–Bit Core repacks data from 512–bit AXI4–Stream packets containing a "**tkeep**" field into a contiguous stream of full 64–byte words containing only valid data.

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 Packet to Contiguous Stream 512–Bit Core.

Features

- Fully AXI4–compliant interfaces
- User accessible reset and packet builder flush
- Control access through AXI4–Lite interface
- The core assumes 32–bit granularity for the data.

Table 1–1: IP Facts Table

Core Specifics	
Supported Design Family[a]	Kintex® Ultrascale and 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 AXI4–Stream Packet to Contiguous Stream 512–Bit Core takes an incoming packetized AXI4–Stream, possibly containing some invalid data, and converts it to a contiguous AXI4–Stream of valid 64–byte words.

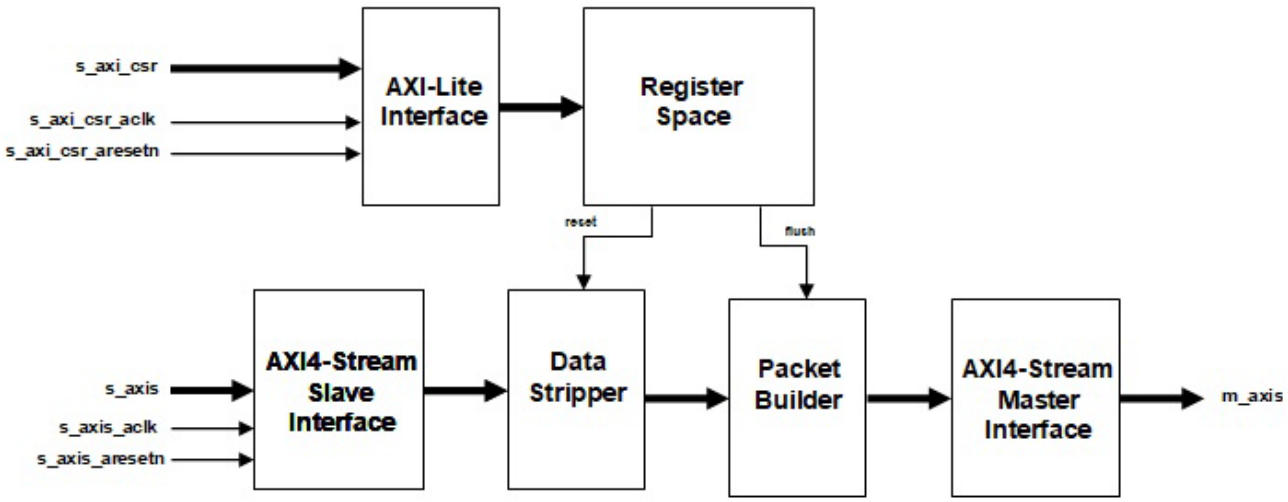
Incoming data is received in the form of 512–bit wide AXI4–stream packets by the AXI4–Stream Slave interface. This AXI4–Stream contains a 64–bit "**tkeep**" field, wherein each bit represents an 8–bit byte of the data bus. When asserted, each bit in this field indicates that its corresponding byte in the data bus is valid.

Note, however, that to save significant FPGA resources, the AXI4–Stream Packet to Contiguous Stream, 512–Bit Core will only look at every 4th bit of the "**tkeep**" field to determine which data words are to be stripped–out. This results in a granularity of 32 bits.

The AXI4–Stream Packet to Contiguous Stream, 512–Bit Core takes the incoming packets and discards the 32–bit data words that are not to be passed (as described above), and continues to accumulate valid data until it can build a full 64–byte packet. The core then passes the packet via the Master AXI4–Stream interface to the downstream device.

Figure 1–1 is a top–level block diagram of the PentekAXI4–Stream Packet to Contiguous Stream 512–Bit Core. The modules within the block diagram are explained in the later sections of this manual.

Figure 1–1: AXI4–Stream Packet to Contiguous Stream 512–Bit Core Block Diagram



1.1 Functional Description (continued)

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

- **Register Space:** This module contains the control register. The register is accessed through the AXI4−Lite interface.

- **AXI4−Stream Interfaces:** The AXI4−Stream Packet to Contiguous Stream, 512−Bit Core has two AXI4−Stream Interfaces. At the input, a 512−bit AXI4−Stream Slave Interface is used to receive AXI4−Streams, and at the output a 512−bit AXI4−Stream Master Interface is used to transfer contiguous, packed AXI4−Streams through the output ports. For more details about the AXI4−Stream Interfaces refer to Section 3.2 AXI4−Stream Core Interfaces.

- **Address FIFO:** This module contains the FIFO which captures the addresses from the burst as well as the packet length.

- **Data Stripper:** This module implements the logic to remove data flagged as invalid by the "**tkeep**" field.

- **Packet Builder:** This module contains the logic to build complete 64−byte packets from the data provided by the Data Stripper.

1.2 Applications

The AXI4−Stream Packet to Contiguous Stream, 512−Bit Core can be incorporated into an Ultrascale or Ultrascale+ FPGA to repack data from incoming 512−bit AXI4−Stream packets containing a "**tkeep**" field into a contiguous output AXI4−Stream of fully valid 64−byte words.

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

4) Pentek IP Core Conventions Guide and Example Labs Guide (807.48111)

This page is intentionally blank

Chapter 2: General Product Specifications

2.1 Standards

The AXI4–Stream Packet to Contiguous Stream 512–Bit Core has interfaces that comply with the *ARM AMBA AXI4–Lite Protocol Specification* and the *ARM AMBA AXI4–Stream Protocol Specification*.

2.2 Performance

The performance of theAXI4–Stream Packet to Contiguous Stream 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 AXI4–Stream Packet to Contiguous Stream 512–Bit Core has two incoming clock signals. The AXI4–Stream clock (**`s_axis_aclk`**) has a maximum frequency of 400 MHz while the CSR clock for the AXI4–Lite interface (**`s_axi_csr_aclk`**) has a maximum frequency of 250 MHz on a Kintex Ultrascale –2 speed grade FPGA. Note that 250 MHz is typically the PCI Express (PCIe®) AXI bus clock frequency.

2.3 Resource Utilization

The resource utilization of theAXI4–Stream Packet to Contiguous Stream 512–Bit 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	1,839
Flip–Flops	2,221

NOTE: Actual utilization may vary based on the user design in which theAXI4–Stream Packet to Contiguous Stream 512–Bit Core is incorporated.

2.4 Limitations and Unsupported Features

- To keep the resource utilization of the core at a reasonable level, only bits 60, 56, 52, 48, 44, 40, 36, 32, 28, 24, 20, 16, 12, 8, 4 and 0 (i.e. every 4th bit) of the "**`tkeep`**" field are evaluated by the data stripper module. This means the granularity of the data stripper module is fixed at 32−bits.

- Input and output data busses are fixed at 512−bits.

2.5 Generic Parameters

This section is not applicable to this IP core.

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
- AXI4–Stream Output Data Bus

3.1 AXI4–Lite Core Interfaces

The AXI4–Stream Packet to Contiguous Stream 512–Bit Core uses the Control/Status Register (CSR) interface to control the reset and flush functions 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 register in the AXI4–Stream Packet to Contiguous Stream 512–Bit Core. Table 3–1 defines the ports in the CSR interface. See Table 4 for a Control Register memory map and bit definitions. 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 reset. This signal is associated with **aclk**. It will reset the control register to its initial state.
s_axi_csr_awaddr	Input	7	**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 Packet to Contiguous Stream 512–Bit Core.
s_axi_csr_awprot	Input	3	**Protection:** The AXI4–Stream Packet to Contiguous Stream 512–Bit 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 Packet to Contiguous Stream 512–Bit 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 Packet to Contiguous Stream 512–Bit 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 Packet to Contiguous Stream 512–Bit 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 Packet to Contiguous Stream 512–Bit Core indicates success or failure of a write transaction through this signal, which is valid when **s_axi_csr_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*.
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.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axi_csr_bvalid	Output	1	**Write Response Valid:** This signal is asserted by theAXI4–Stream Packet to Contiguous Stream 512–Bit 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.
s_axi_csr_araddr	Input	7	**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 Packet to Contiguous Stream 512–Bit Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the AXI4–Stream Packet to Contiguous Stream 512–Bit 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 the **s_axi_csr_araddr**. The AXI4–Stream Packet to Contiguous Stream 512–Bit 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 Packet to Contiguous Stream 512–Bit 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 Packet to Contiguous Stream 512–Bit Core indicates success or failure of a read transaction through this signal, which is valid when **s_axi_csr_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*.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axi_csr_rvalid	Output	1	**Read Data Valid:** This signal is asserted by the AXI4–Stream Packet to Contiguous Stream 512–Bit 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 Packet to Contiguous Stream 512–Bit Core has the following AXI4–Stream Interface, which is used to transfer data streams.

- At the input, a 512–bit AXI4–Stream Slave Interface is used to receive a packetized AXI4–Stream.
- At the output a 512–bit AXI4–Stream Master Interface is used to transfer a contiguous, packed AXI4–Stream with all bits carrying valid data.

3.2.1 AXI4–Stream Input Data Bus

Table 3–2 defines the ports in the AXI4–Stream Input Data Bus Interface. This interface is an AXI4–Stream Slave Interface that is associated with **`s_axis_aclk`**. See the *AMBA AXI4 Specification* for more details on operation of the AXI4–Stream interfaces.

Table 3-2: AXI4–Stream Input Data Bus Interface Port Descriptions			
Port	**Direction**	**Width**	**Description**
s_axis_aclk	Input	1	**Clock:** This is the clock for both the input and the output AXI4–Stream interfaces.
s_axis_aresetn	Input	1	**Reset:** Active LOW reset for both the input and the output AXI4–Stream interfaces.
s_axis_tvalid	Input	1	**Data Valid:** The user design asserts this signal whenever there is valid data on **`s_axis_ tdata`**.
s_axis_tready	Output	1	**Ready for Data:** This signal is asserted by the AXI4–Stream Packet to Contiguous Stream 512–Bit Core to indicate that it is ready to accept data.

Table 3-2: AXI4–Stream Input Data Bus Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axis_tdata	Input	512	**AXI4–Stream Input Data Bus:** This is the input data bus for the core.
s_axis_tkeep	Input	64	**TKEEP Indication for the AXI4–Stream Input Data:** The assertion of bit i of this bus during a transfer indicates that dword i (in this case a dword is 8 bits) of the s_axis_tdata bus contains valid data. **NOTE:** While each of the 64 bits of this input represent 8 of the 512 bits of the incoming data bus, only every 4th bit is actually evaluated by the core. See Section 2.4 for more details.
s_axis_tlast	Input	1	**TLAST Indication AXI4–Stream Input Data:** The user design asserts this signal in the last cycle of a data transfer to indicate the end of the packet.

3.2.2 AXI4–Stream Output Data Bus

Table 3–3 defines the ports in the AXI4–Stream Output Data Bus Interface. This interface is an AXI4–Stream Master Interface that is associated with **`s_axis_aclk`**. See the *AMBA AXI4 Specification* for more details on operation of the AXI4–Stream interfaces.

Table 3-3: AXI4–Stream Output Data Bus Interface Port Descriptions			
Port	**Direction**	**Width**	**Description**
m_axis_tvalid	Output	1	**Data Valid:** The AXI4–Stream Packet to Contiguous Stream 512–Bit Core asserts this signal whenever there is valid data on **`m_axis_tdata`**.
m_axis_tdata	Output	512	**AXI4–Stream Output Data Bus:** This is the output data bus for the AXI4–Stream Packet to Contiguous Stream 512–Bit Core.

This page is intentionally blank

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the AXI4–Stream Packet to Contiguous Stream 512–Bit Core.

Table 4-1: Register Space Memory Map

Register Name	Base Address (Base Address +)	Access	Description
Control Register	0x00	R/W	**Control Register:** Controls the Data Stripper Reset and the Packet Builder Flush functions of the core.

4.1 Control Register

This register controls the reset for the data stripper and the flush control for the data packet builder. The Control Register is illustrated in Figure 4–1 and described in Table 4–2.

Figure 4–1: Control Register

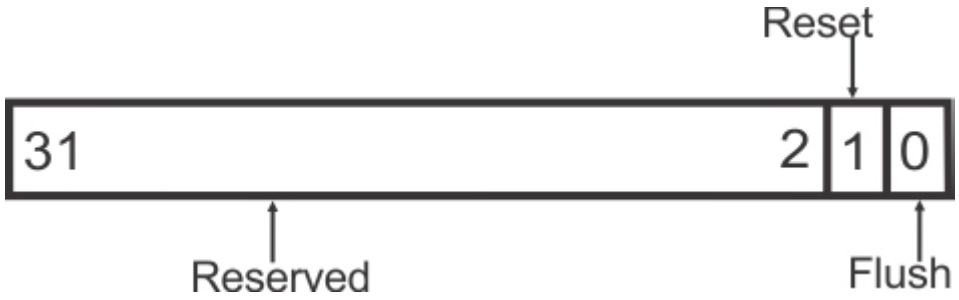


Table 4-2: Control Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:2	**Reserved**	–	–	**Reserved**
1	**Reset**	0	R/W	**Reset:** Active HIGH reset will reset the Data Stripper.
0	**Flush**	0	R/W	**Flush:** Active HIGH – when asserted, any left–over partial data in the data packet builder will be flushed–out as valid data.

This page is intentionally blank

Chapter 5: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the AXI4−Stream Packet to Contiguous Stream 512−Bit Core.

5.1 General Design Guidelines

The AXI4−Stream Packet to Contiguous Stream 512−Bit Core provides the required logic to repack data from the incoming 512−bit AXI4−Stream packets containing a "**tkeep**" field into a contiguous stream of fully−valid 64−byte words.

5.2 Clocking

Main Clock: s_axis_aclk

This clock is used to clock all the ports and logic in the AXI4−Stream Packet to Contiguous Stream 512−Bit Core except the CSR interface and its associated logic.

CSR Clock: s_axi_csr_aclk

This clock is used to clock the CSR interface and its associated logic.

5.3 Resets

Main reset: s_axis_aresetn

This is an active low synchronous reset associated with **s_axis_aclk**. When this reset is asserted, all logic in the s_axis_aclk clock domain of the AXI4−Stream Packet to Contiguous Stream 512−Bit Core is reset.

CSR reset: s_axi_csr_aresetn

This is an active low synchronous reset associated with **s_axi_csr_aclk**. When this reset is asserted, all logic in the **s_axi_csr_aclk** clock domain of the AXI4−Stream Packet to Contiguous Stream 512−Bit Core is reset.

5.4 Interrupts

This core does not have interrupts.

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. See Chapter 4 for the control register memory map, which provides more details on the register that can be accessed through this interface.

- **Input Data Bus:** This is the AXI4−Stream Slave Interface which is associated with **`s_axis_aclk`**. Data from this interface is in the form of packets which may contain invalid data, as defined by the "**`tkeep`**" field. This incoming data is repacked by the AXI4−Stream Packet to Contiguous Stream 512−Bit Core into a contiguous stream of fully valid 64−byte words. For more details about this interface, refer to Section 3.2.1.

- **Output Data Bus:** This is the AXI4−Stream Master Interface which is associated with **`s_axis_aclk`**. Data for this interface is in the form a contiguous stream of fully valid 64−byte words. For more details about this interface, refer to Section 3.2.2.

5.6 Programming Sequence

This section briefly describes the programming sequence for the AXI4−Stream Packet to Contiguous Stream, 512−Bit Core.

1) Set the Control Register with the desired values.

1) Look for a contiguous stream of fully valid 64−byte data on the AXI4−Stream Master output bus as data enters the core on the AXI4−Stream Slave input bus.

5.7 Timing Diagrams

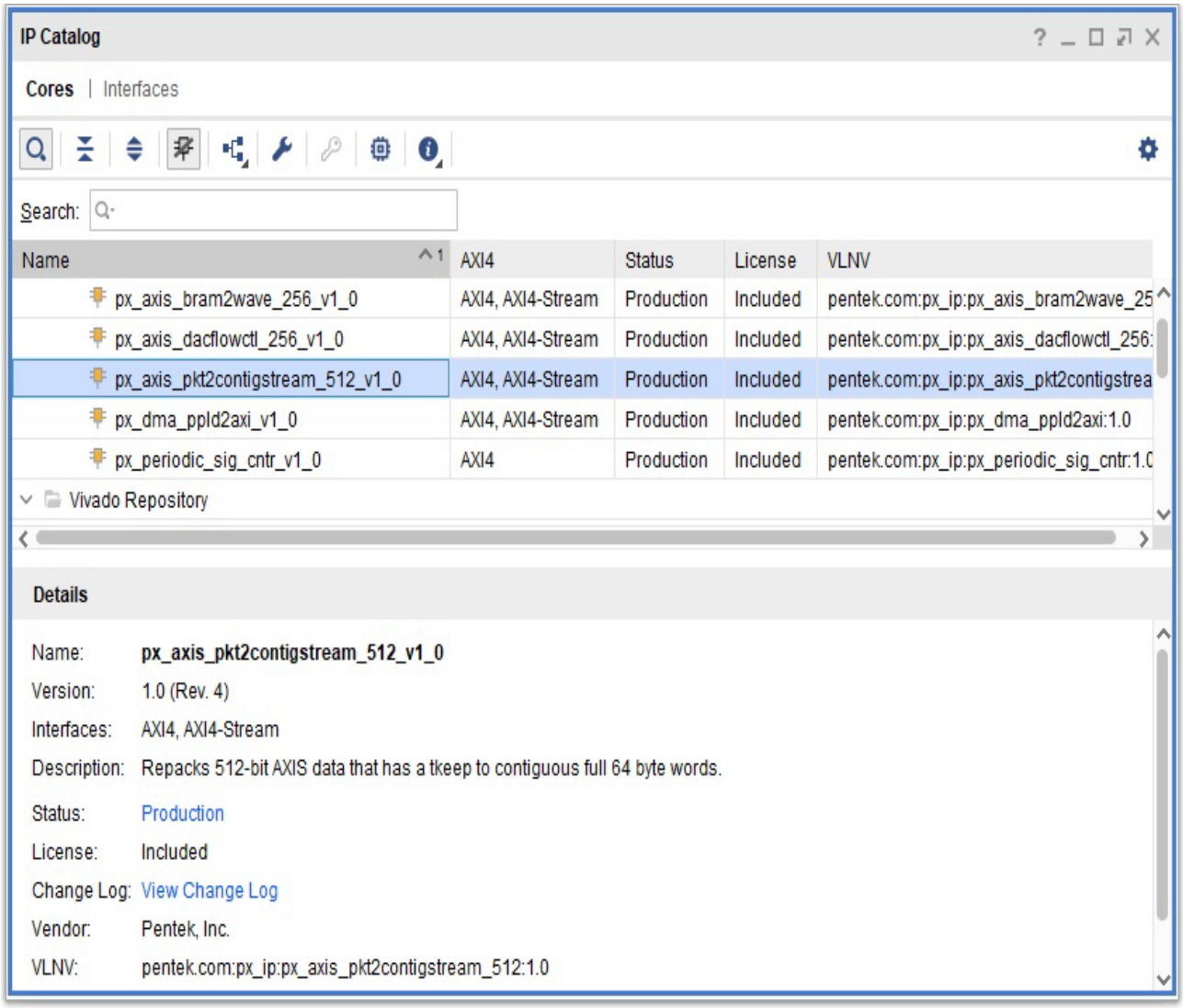
The timing diagram for the AXI4−Stream Packet to Contiguous Stream 512−Bit Core is shown in Section 6−3. This timing diagram is obtained by running the simulation of the test bench for the core in Vivado's VSim environment. For more details about the sim−ulation, refer to Section 6.5.

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek AXI4–Stream Packet to Contiguous Stream 512–Bit Core. It also includes simulation, synthesis, and imple–mentation 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_pkt2contigstream_512_v1_0** as shown in Figure 6–1.

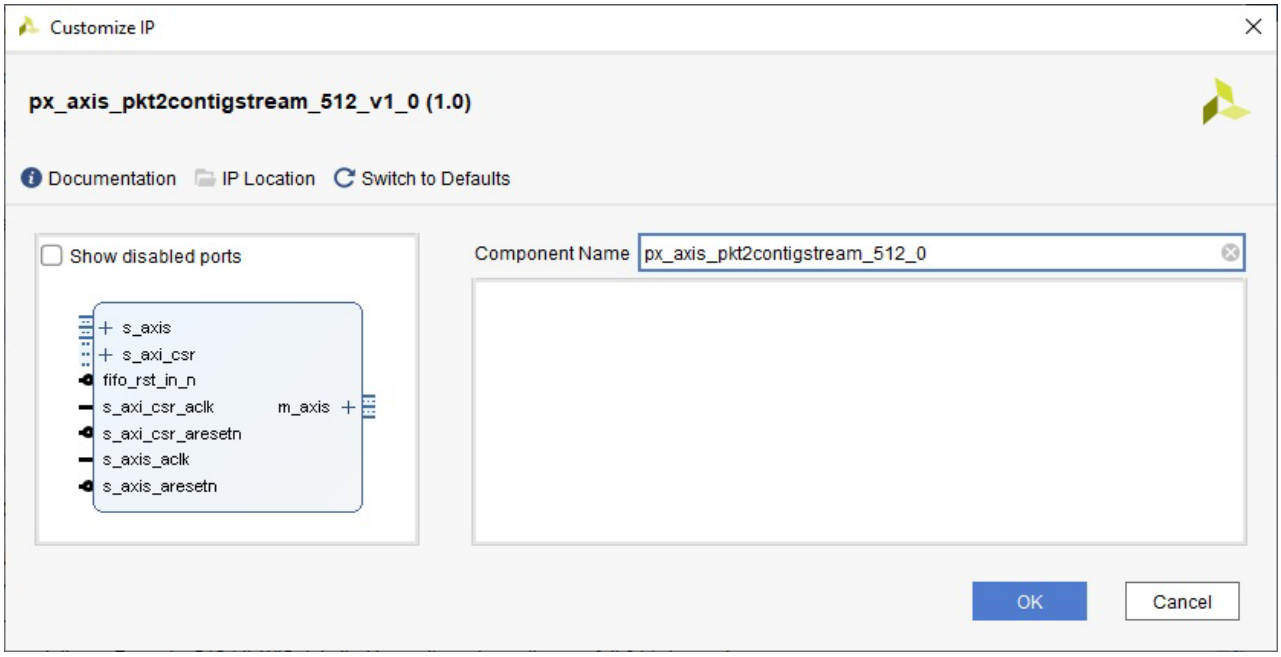
Figure 6–1: AXI4–Stream Packet to Contiguous Stream 512–Bit Core in Pentek IP Catalog



6.1 Pentek IP Catalog (continued)

When you select the **px_axis_pkt2contigstream_512_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 Packet to Contiguous Stream 512–Bit



6.2 User Parameters

This section is not applicable to this IP core.

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 core in Vivado Design Suite.

Required Constraints

The XDC constraints are not provided with this core. The necessary constraints can be applied in the top level module of the user design.

Device, Package, and Speed Grade Selections

This IP works for the Ultrascale and Ultrascale+ families of FPGAs.

Clock Frequencies

The AXI4−Stream input clock (**`s_axis_aclk`**) of the AXI4−Stream Packet to Contiguous Stream 512−Bit Core can take clock frequencies up to 400MHz.

The AXI4−Lite input clock (**`s_axi_csr_aclk`**) of the AXI4−Stream Packet to Contiguous Stream 512−Bit Core can take clock frequencies up to 250MHz.

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.

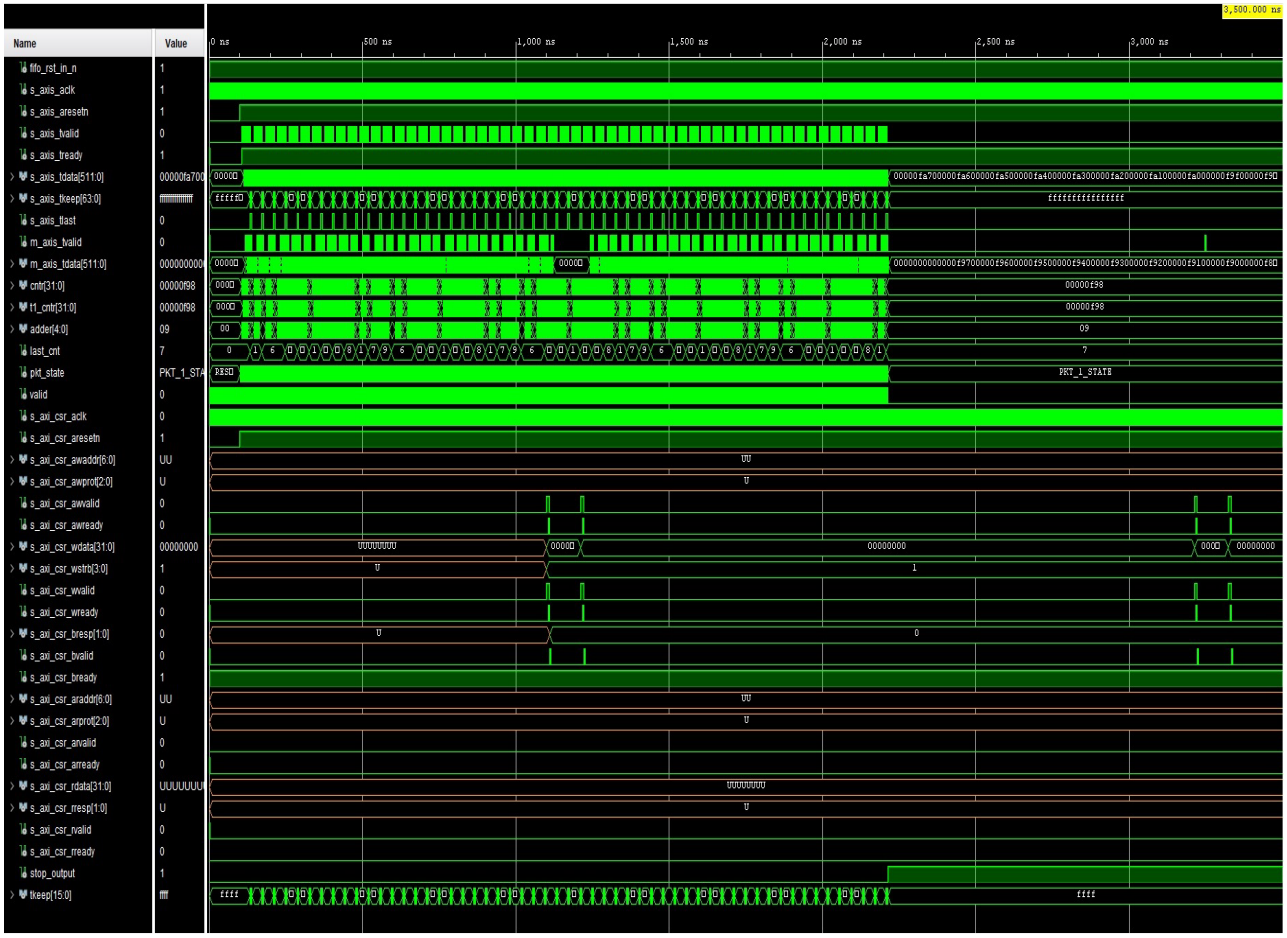
6.5 Simulation

The AXI4–Stream Packet to Contiguous Stream 512–Bit Core has a test bench which generates output waveforms using the Vivado VSim environment. This test bench is designed to with a 250MHz clock for the AXI4–Lite clock (**s_axi_csr_aclk**), and a 312.5MHz clock for the AXI4–Stream clock (**s_axis_aclk**).

The test bench generates packets for the input of the core in groups of five data words. For four of the words, all of the bits in the "**tkeep**" field are set HIGH (set to keep all bits), however for the last word the "**tkeep**" field has some bits that are set LOW (indicating invalid data).

The data stripper state machine and the data fragment flush (in the packet builder) are also tested in the simulation. The resulting waveforms can be seen in Figure 6–3 below.

Figure 6–3: AXI4–Stream Packet to Contiguous Stream 512–Bit Core Simulation Waveform



6.6 Synthesis and Implementation

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