IP CORE MANUAL

NAVIGATOR

FPGA Design Kit

AXI4–Stream PPKT to AXI4–Stream PPLD IP

px_axis_ppkt2ppld

PENTEK

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

Manual Part Number: 807.48425

Rev: 1.2 – November 13, 2018

Manual Revision History

Date	Version	Comments
10/9/18	1.0	Initial Release
10/22/18	1.1	Minor Revisions. Revised Table 3–2, Table 4–1, Table 4–2, and enlarged Figure 6–8.
11/13/18	1.2	Revised Table 2–2. Added bits to Sect 4.1.

Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Pentek products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Pentek hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON–INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Pentek shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in conjunction with, the Materials (including your use of Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage and loss was reasonably foreseeable or Pentek had been advised of the possibility of the same. Pentek assumes no obligation to correct any error contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the materials without prior written consent. Certain products are subject to the terms and conditions of Pentek's limited warranty, please refer to Pentek's Ordering and Warranty information which can be viewed at http://www.pentek.com/contact/customerinfo.cfm; IP cores may be subject to warranty and support terms contained in a license issued to you by Pentek. Pentek products are not designed or intended to be fail–safe or for use in any application requiring fail–safe performance; you assume sole risk and liability for the use of Pentek products in such critical applications.

Copyright

Trademarks

Pentek, Jade, and Navigator are trademarks or registered trademarks of Pentek, Inc.

ARM and AMBA are registered trademarks of ARM Limited. PCI, PCI Express, PCIe, and PCI–SIG are trademarks or registered trademarks of PCI–SIG. Xilinx, Kintex UltraScale, Vivado, and Platform Cable USB are registered trademarks of Xilinx Inc., of San Jose, CA.

Printed in the United States of America.

Table of Contents

Page

IP Facts

Chapter 1: Overview

Chapter 2: General Product Specifications

Chapter 3: Port Descriptions

Table of Contents

Page

Chapter 4: Register Space

Chapter 5: Designing with the Core

Table of Contents

Page

Chapter 6: Design Flow Steps

Table of Contents

Page

This page is intentionally blank

IP Facts

Description

Pentek's Navigator™ AXI4–Stream PPKT to AXI4–Stream PPLD Core provides a method of converting a Packetized Sample Data/Time–stamp/Information (PPKT) data stream to a payload style data stream with programmable packet lengths. The packet length is configurable via a register setting which is accessible to the user via an AXI4–Lite CSR interface, or via a dedicated input bus.

The AXI4–Stream PPKT to AXI4–STREAM PPLD Core requires an External Packet FIFO to buffer the stream. This FIFO is a Xilinx core that must be generated and added to the user design. Details on generating the core can be found in Section 6.3.

This core complies with the ARM® AMBA® *AXI4 Specification* and also provides a control/status register interface. This manual defines the hardware interface, software interface, and parameterization options for the AXI4–Stream PPKT to AXI4–Stream PPLD Core.

Features

- Supports packets up to 64K–Bytes long
- User–programmable packet length via the AXI4–Lite CSR interface
- Override feature allows packet length to be assigned directly from user design
- PPLD packets are made available at the output only when an entire packet is assembled

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 2018.2 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

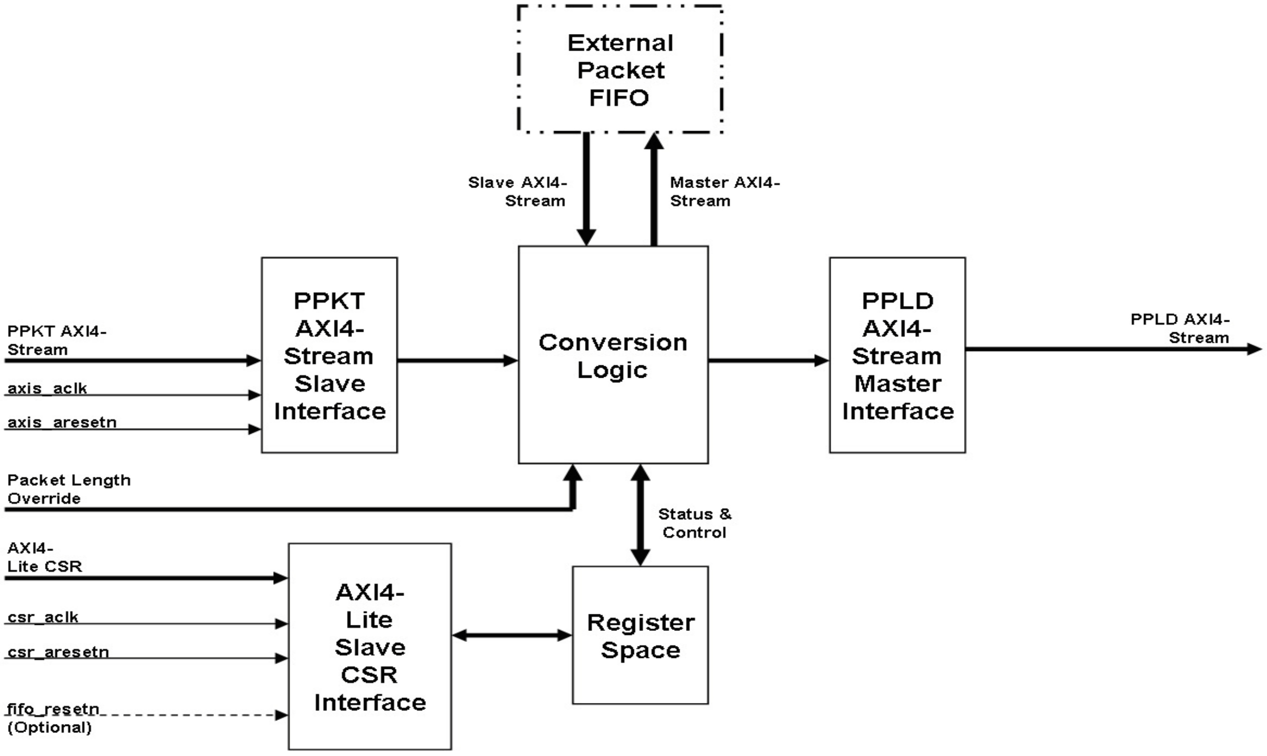
1.1 Functional Description

The AXI4–Stream PPKT to AXI4–Stream PPLD Core assembles payload packets from an input PPKT–style AXI4–Stream. The maximum payload packet size can be defined by the user either via a CSR register or directly from the user's design via an optional override port.

The packet boundary is defined either by the incoming stream's "`tlast`" signal or by the maximum payload packet size set by the user.

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

Figure 1–1: AXI4–Stream PPKT to AXI4–Stream PPLD Core Block Diagram



1.1 Functional Description (continued)

- ❑ **External Packet FIFO:** This module is a Xilinx FIFO core with two AXI4_Stream interfaces: a master from the AXI4−Stream PPKT to AXI4−STREAM PPLD Core to the External Packet FIFO core, and a slave for the reverse direction. For additional details about the AXI4−Stream Interfaces, refer to Section 3.2: AXI4−Stream Core Interfaces. For additional details on generating the core, refer to Section 6.3.

- ❑ **PPKT AXI4−Stream Slave Interface:** This module implements an AXI4−Stream Slave PPKT−style interface for the input data stream to the Core. The data width for this interface, which is the same as the data width for the AXI4−Stream Master Interface, is user configurable when the core is generated. For additional details about the AXI4−Stream Slave Interface, refer to Section 3.2: AXI4−Stream Core Interfaces.

- ❑ **Conversion Logic:** This module buffers incoming data in the External Packet FIFO, and generates packets whose lengths are defined by the user either via a CSR register or the Packet Length Override input to pass to the AXI4−Stream Master Interface.

- ❑ **AXI4−Stream Master Interface:** This module implements an AXI4−Stream Master PPLD−style interface for the output data stream for the Core. The data width for this interface, which is the same as the data width for the PPKT AXI4−Stream Slave Interface, is user configurable when the core is generated. For additional details about the AXI4−Stream Slave Interface, refer to Section 3.2: AXI4−Stream Core Interfaces.

- ❑ **AXI4−Lite Slave CSR 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 the Section 3.1: AXI4−Lite Core Interfaces.

- ❑ **Register Space:** This module contains the control register for the core. The register space is accessed through the AXI4−Lite CSR interface. See Chapter 4 for the register memory map.

1.2 Applications

This core is useful when variable−length, payload−style packets must be generated from a PPKT−style AXI4−Stream.

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) *Vivado Design Suite: FIFO Generator LogiCore IP Product Guide*

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

5) 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 PPKT to AXI4–Stream PPLD 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 AXI4–Stream PPKT to AXI4–Stream PPLD Core is limited 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 PPKT to AXI4–Stream PPLD Core has two incoming clock signals, the AXI4–Stream clock (**axis_aclk**) and AXI4–Lite Interface CSR clock (**s_axi_csr_aclk**). The AXI4–Lite Interface CSR clock has a maximum frequency of 250 MHz, and the AXI4–Stream clock has a maximum frequency of 500 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 the AXI4–Stream PPKT to AXI4–Stream PPLD Core is shown in Table 2–1. Resources have been estimated for a core with a Word Width of 16 (default), Default Packet Size of 256 Bytes and the Packet Length Override option is set to FALSE (default). The target device is a 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	158
Flip–Flops	344

NOTE: Actual utilization may vary based on the user design in which the AXI4–Stream PPKT to AXI4–Stream PPLD Core is incorporated.

NOTE: Resources required for the External Packet FIFO **are not** included in the estimate.

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 PPKT to AXI4–Stream PPLD 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
word_width	Integer	**Word Width:** This parameter defines the width of the data bus of the input and output AXI4–Streams in 16–bit words. Acceptable range is 1 to 64, the default is 16.
default_packet _size	Integer	**Default Packet Size:** This parameter defines the packet size that the core will assume upon reset. Acceptable range is 2 to 65536, the default is 256.
has_override	Boolean	**Has Override:** When set to TRUE the Packet Length Override input will be incorporated into the core. Default setting is FALSE.
has_fifo_rst_in_n	Boolean	**Has FIFO Reset Input:** When set to TRUE, an active–low reset input for the packet FIFO will be incorporated into the core. Default setting is TRUE.
tdest_width	Integer	**Width of the tdest bus:** This parameter defines the width of the tdest AXI4–Stream field. This width setting determines how many AXI4–Stream Destination Control bits in the Control Register are used. The acceptable range is 1 to 4, and the default is 1. See Section 4.1.

Chapter 3: Port Descriptions

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

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

3.1 AXI4–Lite Core Interfaces

The AXI4–Stream PPKT to AXI4–Stream PPLD Core uses the Control/Status Register (CSR) interface to access the control, status and interrupt registers 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 PPKT to AXI4–Stream PPLD Core. Table 3–1 defines the ports in the CSR Interface. See Chapter 4 for a Control/Status Register memory map and bit definitions. See the *AMBA AXI4–Lite Specification* for more details on operation of the AXI4–Lite inter–faces.

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 value will reset all control/status registers to their initial states.
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 PPKT to AXI4–Stream PPLD Core.
s_axi_csr_awprot	Input	3	**Protection:** The AXI4–Stream PPKT to AXI4–Stream PPLD Core ignores these bits.

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

Port	Direction	Width	Description
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 PPKT to AXI4–Stream PPLD 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**.
s_axi_csr_awready	Output	1	**Write Address Ready:** This output is asserted by the AXI4–Stream PPKT to AXI4–Stream PPLD 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 PPKT to AXI4–Stream PPLD 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 PPKT to AXI4–Stream PPLD 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*.

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

Port	Direction	Width	Description
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 PPKT to AXI4–Stream PPLD 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 PPKT to AXI4–Stream PPLD Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the AXI4–Stream PPKT to AXI4–Stream PPLD 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 PPKT to AXI4–Stream PPLD 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 PPKT to AXI4–Stream PPLD 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 PPKT to AXI4–Stream PPLD 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 Interface

The AXI4–Stream PPKT to AXI4–Stream PPLD Core has the following AXI4–Stream Interface as the input data interface to the core:

3.2.1 Stream Data Interface

These interfaces are used to transfer data from the slave input port, through the conversion logic and packet FIFO to the master data output port of the AXI4–Stream PPKT to AXI4–Stream PPLD Core. Table 3–2 defines the ports in the Stream Data Interface. See the *AMBA AXI4–Stream Specification* for more details on the operation of the AXI4–Stream Interface. Refer to the Pentek IP Core Conventions Guide and Example Labs Guide (807.48111) for more information on the PPKT– and PPLD–Style formats.

.

Table 3-2: Stream Data Interface Port Descriptions			
Port	**Direction**	**Width**	**Description**
Input AXI4–Stream Slave Interface (PPKT–Style Stream)			
axis_aclk	Input	1	**AXI4–Stream Clock**
axis_aresetn	Input	1	**Reset:** Active low
s_axis_ppkt_tvalid	Input	1	**Input Data Valid:** This signal is asserted by the user logic when data is valid on **`s_axis_ppkt_tdata`** bus. A data transfer takes place when both **`s_axis_ppkt_tvalid`** and **`s_axis_ppkt_tready`** are High in the same cycle.
s_axis_ppkt_tready	Output	1	**Output Data Ready:** This signal is asserted by the AXI4–Stream PPKT to AXI4–Stream PPLD Core when it is ready to accept data from the user logic.

Table 3-2: Stream Data Interface Port Descriptions (Continued)

Port	Direction	Width	Description
s_axis_ppkt_tdata	Input	word _width *16	**Input Data**

s_axis_ppkt_tuser	Input	80	**Sideband Information:** This is user defined sideband information received alongside the data stream. Data is mapped as follows:
			`tuser[63:0] = timestamp[63:0]`
			`tuser[64] = SOP`
			`tuser[66:65] = Data Format: 0 = 8-bit, 1= 16 bit, 2 = 24-bit, 3 = 32-bit`
			`tuser[67] = Data Type: 0 = Real, 1 = I/Q`
			`tuser[75:68] = channel[7:0]`
			`tuser[79:76] = user[3:0]`
			NOTE: These bits are only valid with `tlast`.

s_axis_ppkt_tkeep	Input	word _width	**Data Keep:** This is a word qualifier signal. Each bit of this signal corresponds to a 16–bit word in `s_axis_ppkt_tdata` i.e., bit 0 corresponds to the least significant 16–bit word of `s_axis_ppkt_tdata` and the most significant bit to the most significant word. Note that this differs from the "standard" (ie non–PPKT–style) usage of `tkeep` in that each bit would qualify an 8–bit byte in a "standard" usage. For PPKT–style packets each bit qualifies an entire 16–bit word. When a bit is asserted, the data on `s_axis_ppkt_tdata` is considered valid. All `s_axis_ppkt_tkeep` bits must be '1' contiguously until `s_axis_ppkt_tlast` is asserted. When `s_axis_ppkt_tlast` is asserted and the number of data samples is not a multiple of `tdata` width, `tkeep` bits are set to '0' to indicate which data bytes are to be ignored.
s_axis_ppkt_tlast	Input	1	**Data Last:** When asserted, `s_axis_ppkt_tlast` marks the last data in the current data frame.
External FIFO AXI4–Stream Master Interface (Data Stream TO the FIFO)			
m_axis_pktfifo_aresetn	Output	1	**FIFO Reset:** Active low
m_axis_pktfifo_tvalid	Output	1	**Output Data Valid:** This signal is asserted when data is valid on `m_axis_pktfifo_tdata` bus.

Table 3-2: Stream Data Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
m_axis_pktfifo_tready	Input	1	**Data Ready:** Active High. This is an input tready signal from the FIFO's AXI Stream slave, indicating that it is ready to accept data. Data is transferred to the External Packet FIFO when both **m_axis_pktfifo_tvalid** and **m_axis_pktfifo_tready** are High on the same cycle. If the FIFO's AXI Stream slave deasserts the ready signal when **m_axis_pktfifo_tvalid** is High, the core maintains the data on the bus and keeps the valid signal asserted until the slave has asserted the ready signal.
m_axis_pktfifo_tdata	Output	word _width*16	**Output data**
m_axis_pktfifo_tuser	Output	104	**Sideband Data:** This is user defined sideband output information transmitted alongside the data stream. Data is mapped as follows: **tuser[63:0] = timestamp[63:0]** **tuser[64] = Start of Payload Packet** **tuser[66:65] = Data Format 0 = 8-bit, 1= 16 bit, 2 = 24-bit, 3 = 32-bit** **tuser[67] = Data Type 0 = Real, 1 = I/Q** **tuser[75:68] = channel[7:0]** **tuser[78:76] = Reserved** **tuser[79] = Acq. End (Input PPKT had tlast)** **tuser[96:80] = Payload Size (in bytes)** **tuser[103:97] = Number Valid Bytes in cycle**
m_axis_pktfifo_tkeep	Output	word _width*2	**Data Keep:** This is a byte qualifier signal. Each bit of this signal corresponds to a byte in **m_axis_pktfifo_tdata** i.e., bit 0 corresponds to the least significant byte of **m_axis_pktfifo_tdata** and the most significant bit to the most significant byte. When a bit is asserted, the data on **m_axis_pktfifo_tdata** is considered valid. All **m_axis_pktfifo_tkeep** bits must be '1' contiguously until **m_axis_pktfifo_tlast** is asserted. When **m_axis_pktfifo_tlast** is asserted, and the number of data samples is not a multiple of tdata width, **tkeep** bits are set to '0' to indicate which data bytes are to be ignored.
m_axis_pktfifo_tlast	Output	1	**Data Last:** When asserted, **m_axis_pktfifo_tlast** marks the last data in the current data frame.
External FIFO AXI4–Stream Slave Interface (Data Stream FROM the FIFO)			

Table 3-2: Stream Data Interface Port Descriptions (Continued)

Port	Direction	Width	Description
s_axis_pktfifo_tvalid	Input	1	**Input Data Valid:** This signal is asserted by the user logic when data is valid on **s_axis_pktfifo_tdata** bus. A data transfer takes place when both **s_axis_ppkt_tvalid** and **s_axis_ppkt_tready** are High in the same cycle.
s_axis_pktfifo_tready	Output	1	**Output Data Ready:** This signal is asserted by the AXI4–Stream PPKT to AXI4–STREAM PPLD Core when it is ready to accept data from the FIFO.
s_axis_pktfifo_tdata	Input	word _width*16	**Input Data**
s_axis_pktfifo_tuser	Input	104	**Sideband Data:** This is user defined sideband information received alongside the data stream. Data is mapped as follows: **tuser[63:0] = timestamp[63:0]** **tuser[64] = Start of Payload Packet** **tuser[66:65] = Data Format 0 = 8-bit, 1= 16 bit, 2 = 24-bit, 3 = 32-bit** **tuser[67] = Data Type 0 = Real, 1 = I/Q** **tuser[75:68] = channel[7:0]** **tuser[78:76] = Reserved** **tuser[79] = Acq. End (Input PPKT had tlast)** **tuser[96:80] = Payload Size (in bytes)** **tuser[103:97] = Number Valid Bytes in cycle**
s_axis_pktfifo_tkeep	Input	word _width*2	**Data Keep:** This is a byte qualifier signal. Each bit of this signal corresponds to a 16–bit word in **s_axis_pktfifo_tdata** i.e., bit 0 corresponds to the least significant 16–bit word of **s_axis_pktfifo_tdata** and the most significant bit to the most significant byte. When a bit is asserted, the data on **s_axis_pktfifo_tdata** is considered valid. All **s_axis_pktfifo_tkeep** bits must be '1' contiguously until s_axis_ppkt_tlast is asserted. When **s_axis_pktfifo_tlast** is asserted and the number of data samples is not a multiple of tdata width, **tkeep** bits are set to '0' to indicate which data bytes are to be ignored.
s_axis_pktfifo_tlast	Input	1	**Data Last:** When asserted, **s_axis_pktfifo_tlast** marks the last data in the current data frame.
Output AXI4–Stream Master Interface (PPLD–Style Stream)			
m_axis_ppld_tvalid	Output	1	**Output Data Valid:** This signal is asserted when data is valid on **m_axis_ ppld_tdata** bus.

Table 3-2: Stream Data Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
m_axis_ppld_tready	Input	1	**Data Ready:** Active High. This is an input tready signal from the user logic's slave, indicating that it is ready to accept data. Data is transferred to the External Packet FIFO when both **m_axis_ppld_tvalid** and **m_axis_ppld_tready** are High on the same cycle. If the user logic's slave deasserts the ready signal when **m_axis_ppld_tvalid** is High, the core maintains the data on the bus and keeps the valid signal asserted until the slave has asserted the ready signal.
m_axis_ppld_tdata	Output	word _width*16	**Output Data**
m_axis_ppld_tuser	Output	104	**Sideband Data:** This is user defined sideband output information transmitted alongside the data stream. Data is mapped as follows: **tuser[63:0] = timestamp[63:0]** **tuser[64] = Start of Payload Packet** **tuser[66:65] = Data Format 0 = 8-bit, 1= 16 bit, 2 = 24-bit, 3 = 32-bit** **tuser[67] = Data Type 0 = Real, 1 = I/Q** **tuser[75:68] = channel[7:0]** **tuser[78:76] = Reserved** **tuser[79] = Acq. End (Input PPKT had tlast)** **tuser[96:80] = Payload Size (in bytes)** **tuser[103:97] = Number Valid Bytes in cycle**
m_axis_ppld_tkeep	Output	word _width*2	**Data Keep:** This is a byte qualifier signal. Each bit of this signal corresponds to a byte in **m_axis_ppld_tdata** i.e., bit 0 corresponds to the least significant byte of **m_axis_ppld_tdata** and the most significant bit to the most significant byte. When a bit is asserted, the data on **m_axis_ppld_tdata** is considered valid. All **m_axis_ppld_tkeep** bits must be '1' contiguously until **m_axis_ppld_tlast** is asserted. When **m_axis_ppld_tlast** is asserted, and the number of data samples is not a multiple of **tdata** width, **tkeep** bits are set to '0' to indicate which data bytes are to be ignored.
m_axis_ppld_tlast	Output	1	**Data Last:** When asserted, **m_axis_pktfifo_tlast** marks the last data in the current data frame.

3.3 I/O Signals

The I/O port/signal descriptions of the top level module of the AXI4–Stream to AXI4 DMA Core are provided in Table 3–3.

Table 3–3: I/O Signal Descriptions

Port/Signal Name	Type	Direction	Description
fifo_rst_in_n	std_logic	Input	**Optional FIFO Reset:** This active LOW input is associated with **`axis_aclk`**. This input is only available when parameter **`has_fifo_rst_in_n`** is set to TRUE.
override_pkt _size_en	std_logic	Input	**Optional Packet Size Override Enable:** This active HIGH input is associated with **`axis_aclk`**. This input is only available when parameter **`has_override`** is set to TRUE. When available, asserting this input directs the core to use the current value on the **`override_pkt_size`** input as the packet size rather than using the CSR register setting.
override_pkt_s ize	std_logic	Input	**Optional Packet Size Override:** This input is associated with **`axis_aclk`**, and is only available when parameter **`has_override`** is set to TRUE. When available and **`override_pkt_size_en`** is asserted, the value on this input overrides the value in the CSR register, and is used as the packet size.

This page is intentionally blank

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the register space of the AXI4–Stream PPKT to AXI4–Stream PPLD Core. The memory map is provided in Table 4–1.

Table 4–1: Register Space Memory Map

Register Name	Address (Base Address +)	Access	Description
Control Register	0x00	R/W	Control word

4.1 Control Register

This register provides the control settings for the core. It is illustrated in Figure 4–1 and described in Table 4–2.

Figure 4–1: Control Register

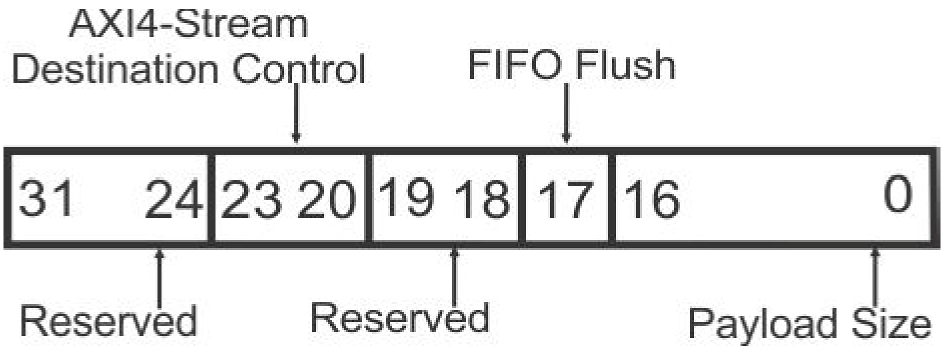


Table 4–2: Control Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:24	Reserved	N/A	N/A	**Reserved**
23:20	AXI4–Stream Destination Control	0	R/W	**AXI4–Stream Destination Control:** These bits set the `tdest` field of the output AXI4–Stream. This can be used to steer the output stream to a destination through an AXI4–Stream switch. 0 = Routes ADC/DDC data to DMA 1 = Routes ADC/DDC data to 100GE
19:18	Reserved	N/A	N/A	**Reserved**
17	FIFO Flush	0	R/W	**FIFO Flush:** Active HIGH Reset for the External Packet FIFO.
16:0	Payload Size	0x0000	R/W	**Payload Size:** When not overridden by the optional override inputs, this sets the number of bytes to be assigned to each packet.

Chapter 5: Designing with the Core

This chapter provides guidelines and additional information to facilitate designing with the AXI4–Stream PPKT to AXI4–Stream PPLD Core.

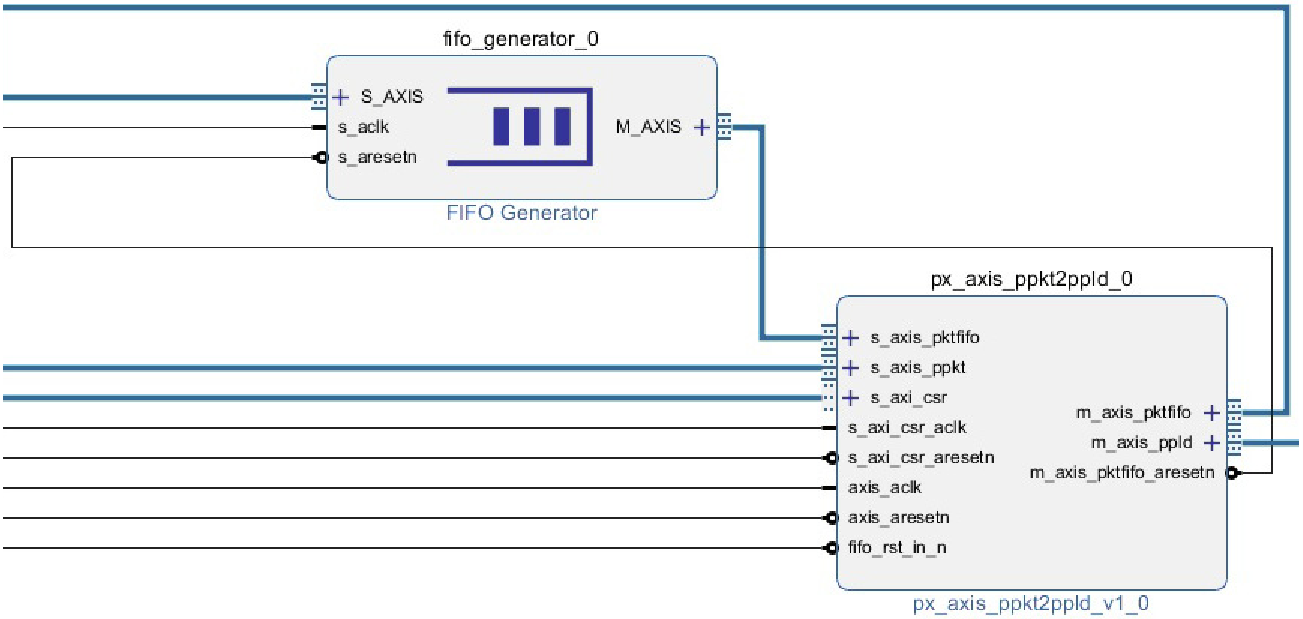
5.1 General Design Guidelines

The AXI4–Stream PPKT to AXI4–Stream PPLD Core, when combined with the required External Packet FIFO, provides the required logic to convert a PPKT–Style AXI4–Stream to a variable packet–length, PPLD–Style AXI4–Stream. The user can customize the core by setting the generic parameters based on the application requirement as described in Section 2.5.

5.2 Generating the External Packet FIFO

The External Packet FIFO is a Xilinx core which must be generated as part of the design process and added to the top–level design alongside the AXI4–Stream PPKT to AXI4–STREAM PPLD Core as shown in Figure 5–1.

Figure 5–1: AXI4–Stream PPKT to AXI4–Stream PPLD Core with External Packet FIFO



Details on generating the core can be found in Section 6.3: Generating the Xilinx Packet Fifo.

5.3 Clocking

AXI4–Lite Clock: **`s_axi_csr_aclk`**

This clock is used to clock the AXI4–Lite Control/Status Register (s_axi_csr) interface of the core and its associated logic.

AXI4–Stream Clock: **`axis_aclk`**

This clock provides clocking for all of the AXI4–Stream interfaces and logic as well as the External Packet FIFO.

5.4 Resets

CSR Reset: **`s_axi_csr_aresetn`**

This is an active–low synchronous reset associated with the **`s_axi_csr_aclk`**. When asserted, all CSR state machines in the core are reset.

AXI4–Stream Reset: **`axis_aresetn`**

This is an active–low synchronous reset associated with the **`axis_aclk`**. When asserted the AXI4–stream interfaces are reset.

FIFO Reset (Optional): **`fifo_resetn`**

This active–low synchronous reset is associated with the **`axis_aclk`**. When asserted, the External Packet FIFO is reset.

5.5 Interrupts

This core does not have interrupts.

5.6 Interface Operation

- ❑ **Control/Status Register Interface:** This is the control register interface. It is associated with the **`s_axi_csr_aclk`**, and is a standard AXI4–Lite type interface. See Chapter 4 for the control register memory map and for more details on the register that can be accessed through this interface.
- ❑ **Stream Data Interface (input and output):** This is the interface which will be converted from PPKT–Style AXI4–Stream to a variable packet–length, PPLD–Style AXI4–Stream. It is a standard AXI4–Stream Interface – slave at the input and master at the output. For more details about this interface refer to Section 3.2.
- ❑ **External Packet FIFO Stream Data Interface (input and output):** This is the interface to and from the External Packet FIFO. It is also a standard AXI4–Stream Interface – slave at the input (FROM the FIFO) and master at the output (TO the FIFO). For more details about this interface refer to Section 3.2.

5.7 Programming Sequence

This section briefly describes the programming sequence for the AXI4−Stream PPKT to AXI4−Stream PPLD Core.

1) Set the Maximum Payload Size, either via the CSR register or the optional override interface, to the user application requirements.

2) Observe output data when valid input data streams are available at the input ports.

5.8 Timing Diagrams

The timing diagram for the AXI4−Stream PPKT to AXI4−Stream PPLD Core is shown in Figure 6−8. This timing diagram is obtained by running the simulation of the test bench of the core in Vivado VSim environment. For more details about the test bench, refer to Section 6.6.

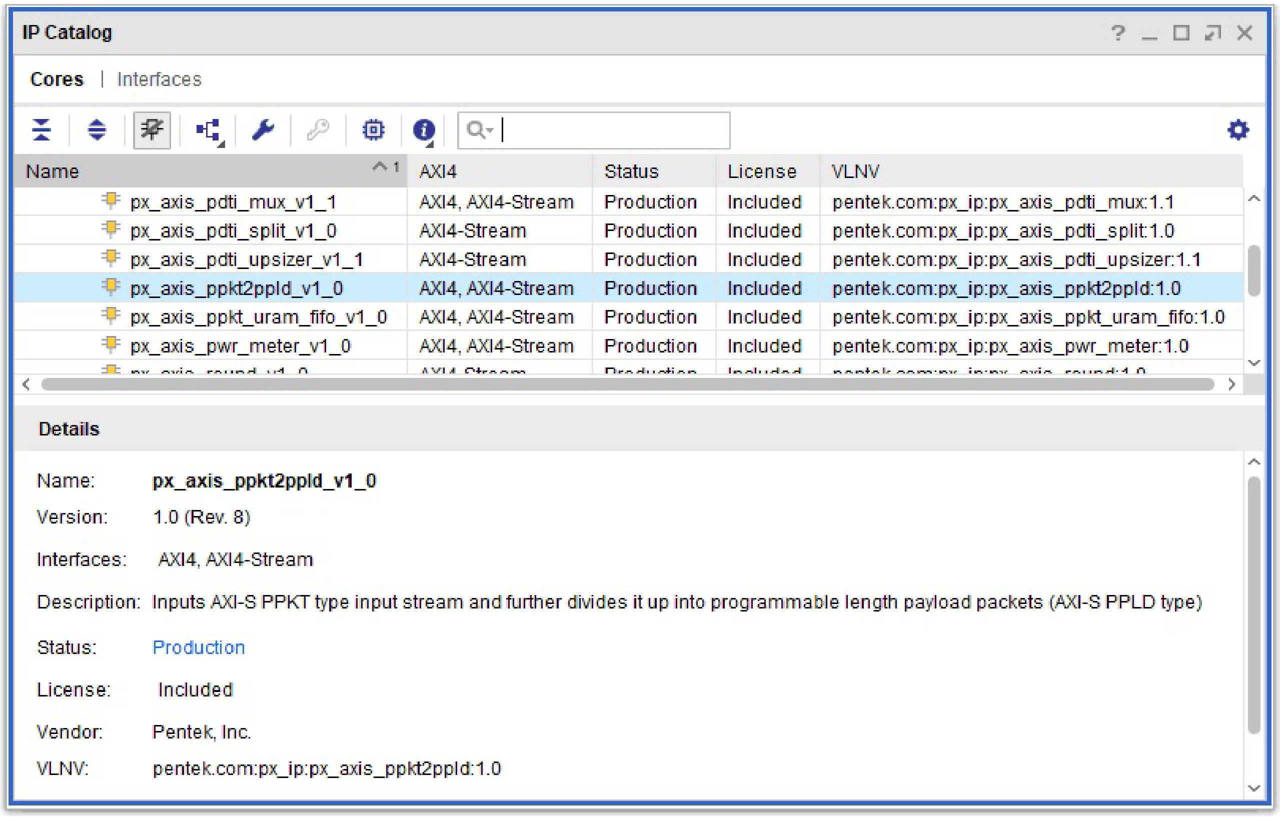
This page is intentionally blank

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek AXI4–Stream PPKT to AXI4–Stream PPLD 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_ppkt2ppld_v1_0`** as shown in Figure 6–1.

Figure 6–1: AXI4–Stream PPKT to AXI4–Stream PPLD Core in Pentek



Name	AXI4	Status	License	VLNV
px_axis_pdti_mux_v1_1	AXI4, AXI4-Stream	Production	Included	pentek.com:px_ip:px_axis_pdti_mux:1.1
px_axis_pdti_split_v1_0	AXI4-Stream	Production	Included	pentek.com:px_ip:px_axis_pdti_split:1.0
px_axis_pdti_upsizer_v1_1	AXI4-Stream	Production	Included	pentek.com:px_ip:px_axis_pdti_upsizer:1.1
px_axis_ppkt2ppld_v1_0	AXI4, AXI4-Stream	Production	Included	pentek.com:px_ip:px_axis_ppkt2ppld:1.0
px_axis_ppkt_uram_fifo_v1_0	AXI4, AXI4-Stream	Production	Included	pentek.com:px_ip:px_axis_ppkt_uram_fifo:1.0
px_axis_pwr_meter_v1_0	AXI4, AXI4-Stream	Production	Included	pentek.com:px_ip:px_axis_pwr_meter:1.0
px_axis_round_v1_0	AXI4-Stream	Production	Included	pentek.com:px_ip:px_axis_round:1.0

Name: **px_axis_ppkt2ppld_v1_0**

Version: 1.0 (Rev. 8)

Interfaces: AXI4, AXI4-Stream

Description: Inputs AXI-S PPKT type input stream and further divides it up into programmable length payload packets (AXI-S PPLD type)

Status: Production

License: Included

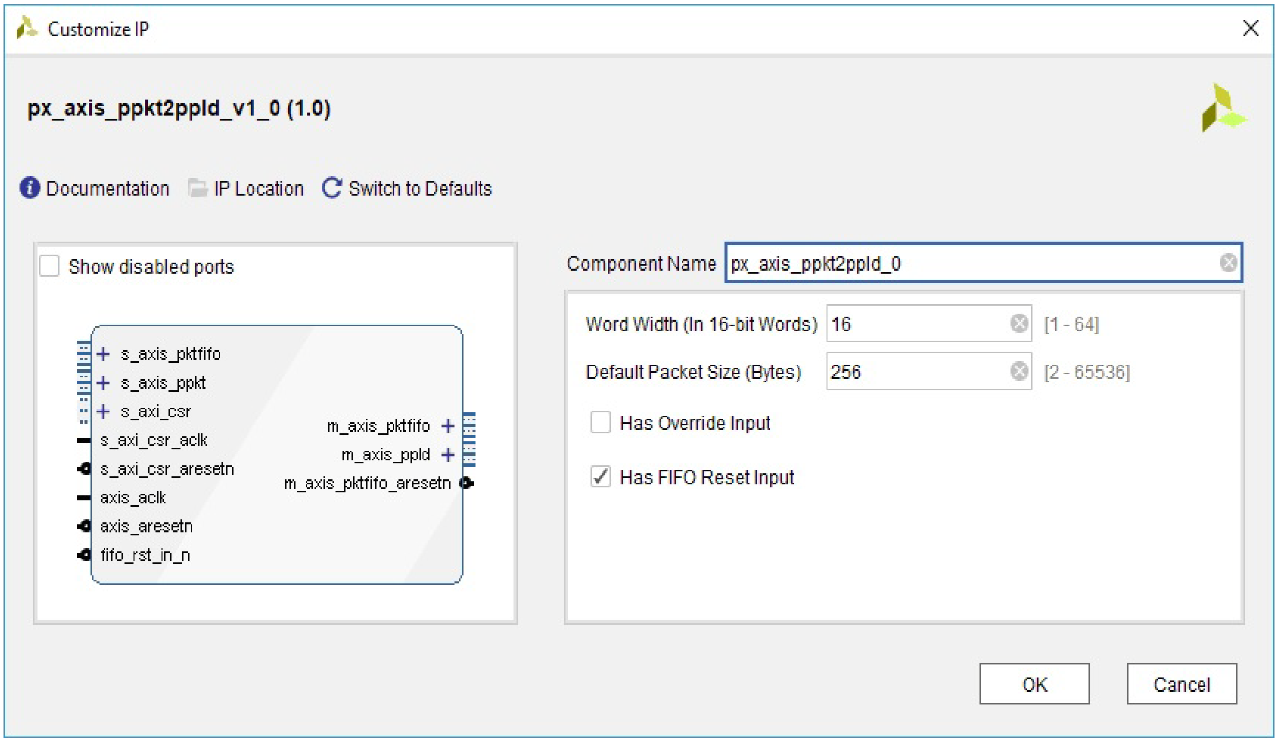
Vendor: Pentek, Inc.

VLNV: pentek.com:px_ip:px_axis_ppkt2ppld:1.0

6.1 Pentek IP Catalog (continued)

When you select the **`px_axis_ppkt2ppld_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 PPKT to AXI4–Stream PPLD Core IP



Customize IP

px_axis_ppkt2ppld_v1_0 (1.0)

□ Show disabled ports

Component Name: px_axis_ppkt2ppld_0

Word Width (In 16-bit Words)	16	[1 - 64]
Default Packet Size (Bytes)	256	[2 - 65536]

□ Has Override Input

☑ Has FIFO Reset Input

6.2 User Parameters

The user parameters of this AXI4–Stream PPKT to AXI4–Stream PPLD Core are explained in Section 2.5 of this user manual.

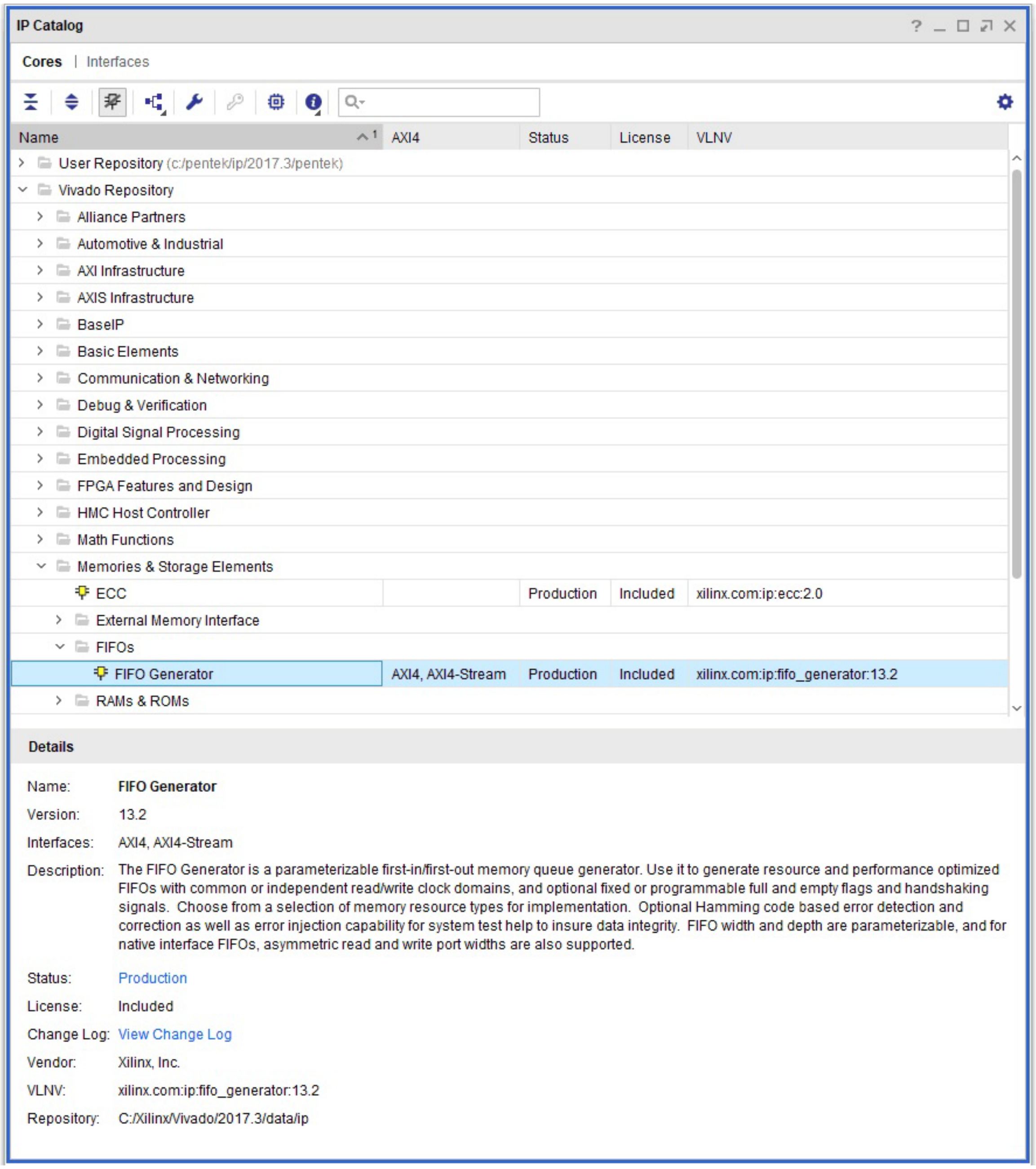
6.3 Generating the Xilinx Packet FIFO

NOTE: The parameters shown in this section are for demonstrative purposes only, and may not be optimal for the user's design. The user should choose settings based on the requirements of the system being designed.

6.3.1 Step 1

Add the core to the top level block diagram by selecting Vivado Repository => Memories & Storage Elements => FIFOs => FIFO Generator (see Figure 6–3).

Figure 6–3: FIFO Generator in the Vivado IP Catalog



Name	AXI4	Status	License	VLNV
User Repository (c:/pentek/ip/2017.3/pentek)				
Vivado Repository				
Alliance Partners				
Automotive & Industrial				
AXI Infrastructure				
AXIS Infrastructure				
BaseIP				
Basic Elements				
Communication & Networking				
Debug & Verification				
Digital Signal Processing				
Embedded Processing				
FPGA Features and Design				
HMC Host Controller				
Math Functions				
Memories & Storage Elements				
ECC		Production	Included	xilinx.com:ip:ecc:2.0
External Memory Interface				
FIFOs				
FIFO Generator	AXI4, AXI4-Stream	Production	Included	xilinx.com:ip:fifo_generator:13.2
RAMs & ROMs				

Details

Name: **FIFO Generator**

Version: 13.2

Interfaces: AXI4, AXI4-Stream

Description: The FIFO Generator is a parameterizable first-in/first-out memory queue generator. Use it to generate resource and performance optimized FIFOs with common or independent read/write clock domains, and optional fixed or programmable full and empty flags and handshaking signals. Choose from a selection of memory resource types for implementation. Optional Hamming code based error detection and correction as well as error injection capability for system test help to insure data integrity. FIFO width and depth are parameterizable, and for native interface FIFOs, asymmetric read and write port widths are also supported.

Status: Production

License: Included

Change Log: View Change Log

Vendor: Xilinx, Inc.

VLNV: xilinx.com:ip:fifo_generator:13.2

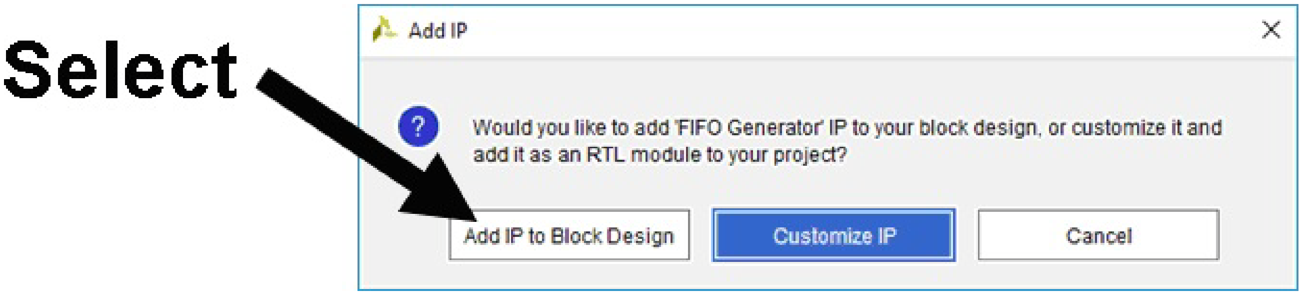
Repository: C:/Xilinx/Vivado/2017.3/data/ip

6.3 Generating the Xilinx Packet FIFO (continued)

6.3.2 Step 2

In the "Add IP" dialog box, select "Add IP to Block Design" (see Figure 6–4).

Figure 6–4: "Add IP" Dialog Box for Xilinx FIFO Generator



Select

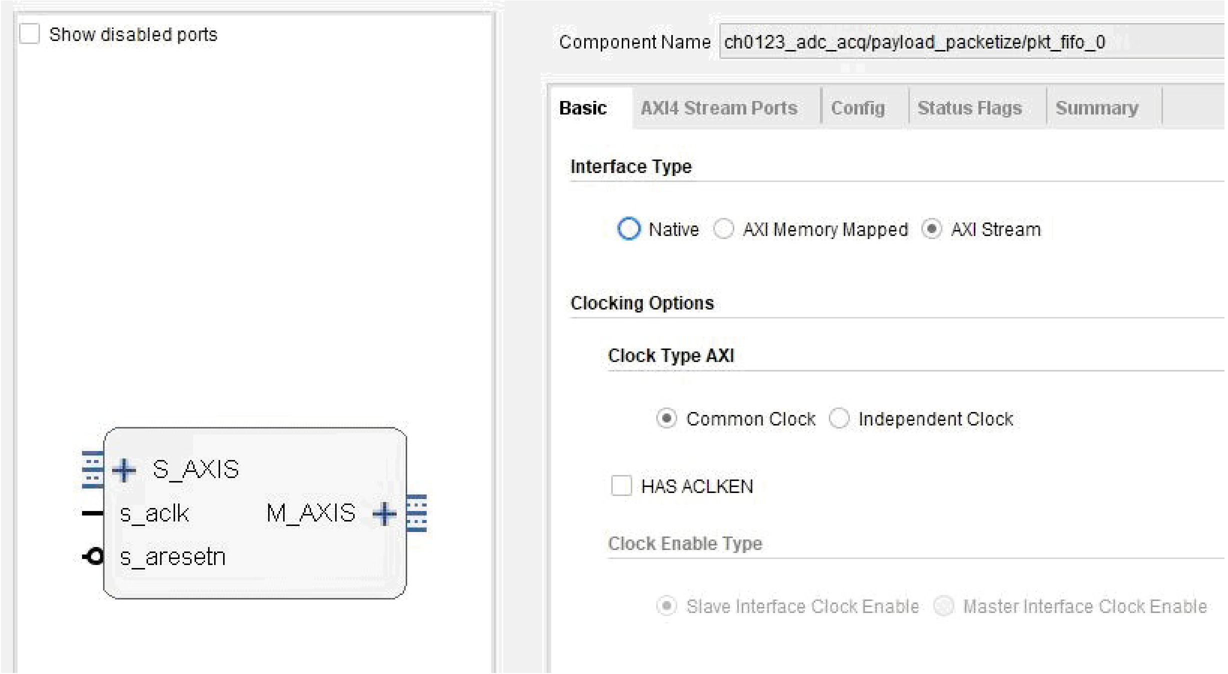
Add IP

Would you like to add 'FIFO Generator' IP to your block design, or customize it and add it as an RTL module to your project?

6.3.3 Step 3

Double–click on the FIFO Generator block in the Block Diagram to open the customization dialog box. In the "Basic" tab, set the parameters as required. See example settings in Figure 6–5.

Figure 6–5: "Basic" Tab Setup for Xilinx FIFO Generator



☐ Show disabled ports

Component Name ch0123_adc_acq/payload_packetize/pkt_fifo_0

Interface Type

○ Native ○ AXI Memory Mapped ● AXI Stream

Clocking Options

Clock Type AXI

● Common Clock ○ Independent Clock

☐ HAS ACLKEN

Clock Enable Type

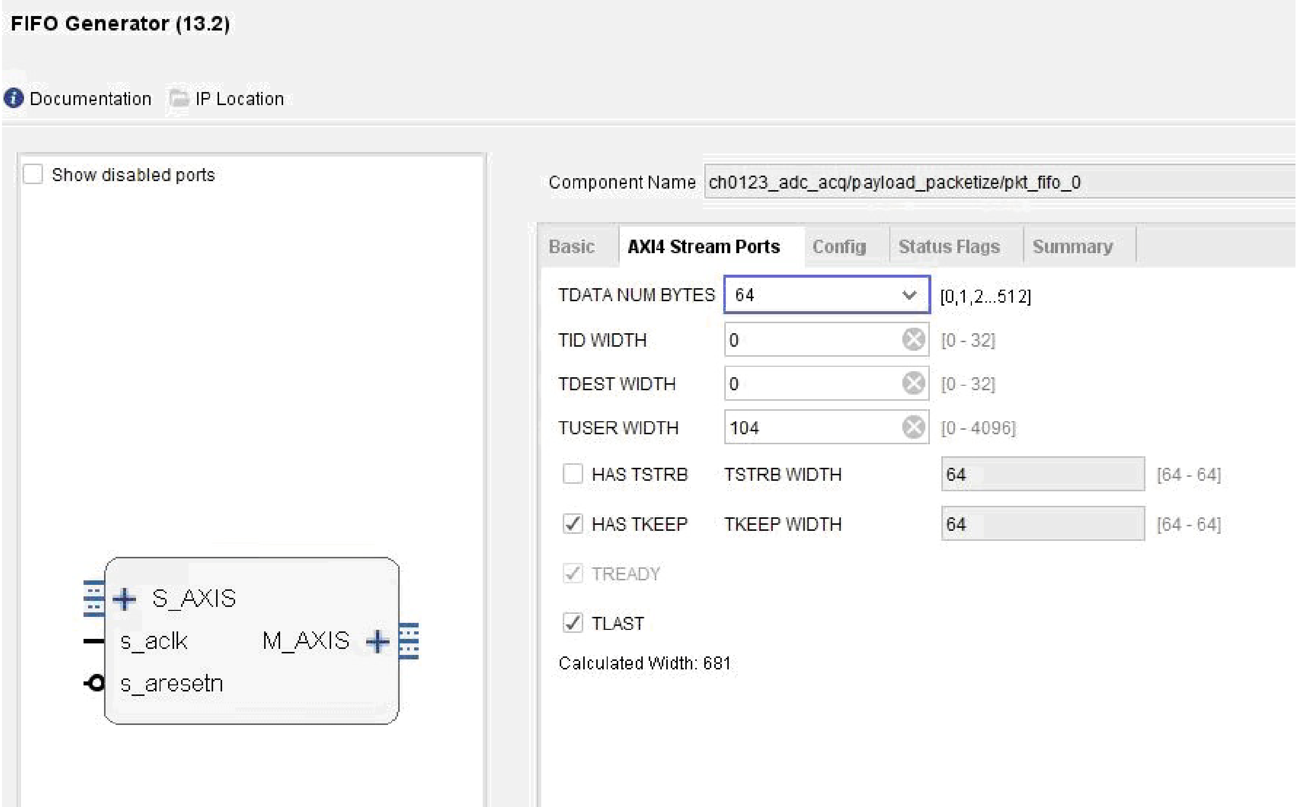
● Slave Interface Clock Enable ○ Master Interface Clock Enable

6.3 Generating the Xilinx Packet FIFO (continued)

6.3.4 Step 4

In the "AXI4 Stream Ports" tab, set the parameters as required. See example settings in Figure 6–6.

Figure 6–6: "AXI4 Stream Ports" Tab Setup for Xilinx FIFO Generator



FIFO Generator (13.2)

Documentation IP Location

□ Show disabled ports

Component Name ch0123_adc_acq/payload_packetize/pkt_fifo_0

Basic | AXI4 Stream Ports | Config | Status Flags | Summary

TDATA NUM BYTES 64 [0,1,2...512]

TID WIDTH 0 [0 - 32]

TDEST WIDTH 0 [0 - 32]

TUSER WIDTH 104 [0 - 4096]

□ HAS TSTRB TSTRB WIDTH 64 [64 - 64]

☑ HAS TKEEP TKEEP WIDTH 64 [64 - 64]

☑ TREADY

☑ TLAST

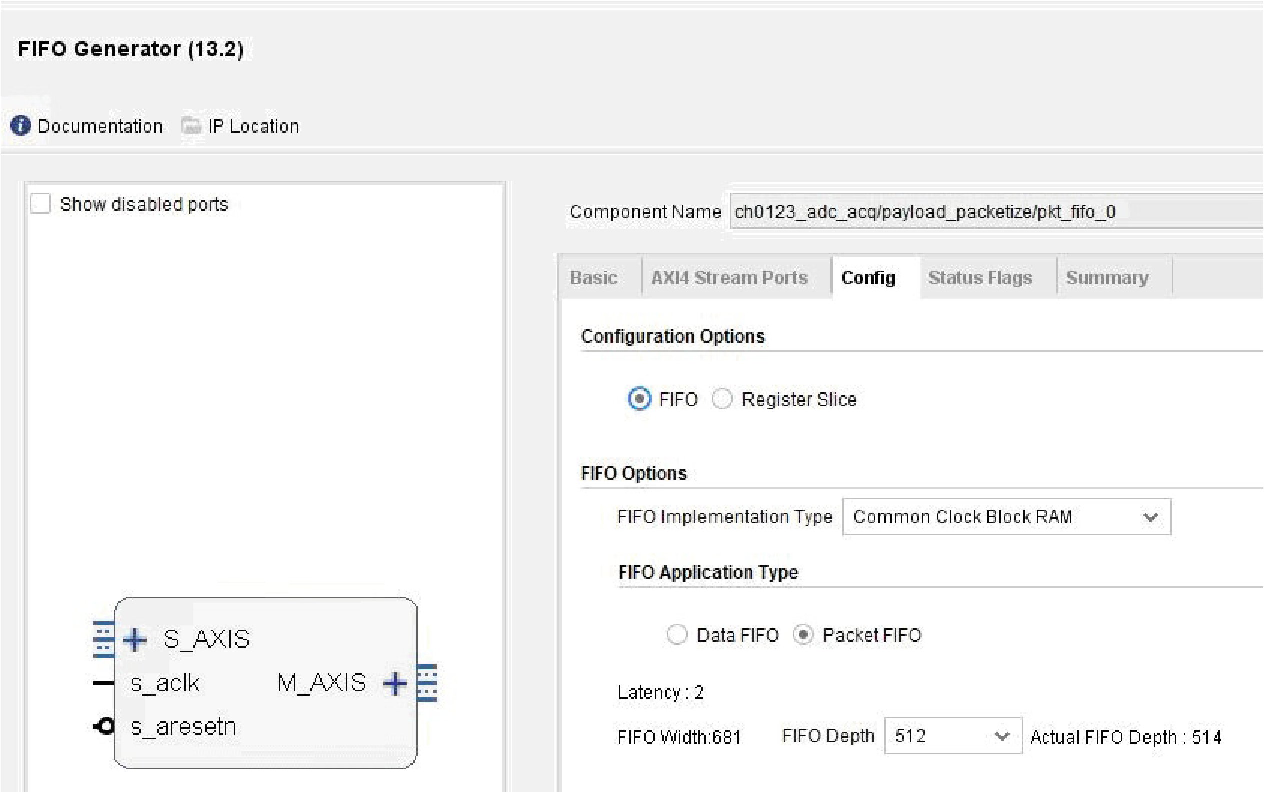
Calculated Width: 681

6.3 Generating the Xilinx Packet FIFO (continued)

6.3.5 Step 5

In the "Config" tab, set the parameters as required. See example settings in Figure 6–7.

Figure 6–7: "Config" Tab Setup for Xilinx FIFO Generator



FIFO Generator (13.2)

□ Show disabled ports

Component Name ch0123_adc_acq/payload_packetize/pkt_fifo_0

Configuration Options

● FIFO ○ Register Slice

FIFO Options

FIFO Implementation Type Common Clock Block RAM

FIFO Application Type

○ Data FIFO ● Packet FIFO

Latency : 2

FIFO Width:681 FIFO Depth 512 Actual FIFO Depth : 514

Since there are no status flags, no changes in the "Status Flags" tab are necessary. Select the "OK" button in the lower right corner of the dialog box to complete the setup.

6.4 Output Generation

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

6.5 Constraining the Core

This section contains information about constraining the AXI4−Stream PPKT to AXI4−Stream PPLD Core in the Vivado Design Suite.

Required Constraints

The XDC constraints are not provided with the AXI4−Stream PPKT to AXI4−Stream PPLD 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 Kintex Ultrascale FPGAs.

Clock Frequencies

For the streaming data path clock (**`axis_aclk`**), the limiting factor is the BRAM in the External Packet FIFO, which has a maximum clock frequency of 500 MHz.

The AXI4−Lite interface clock (**`s_axi_csr_aclk`**) frequency 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.

6.6 Simulation

The AXI4–Stream PPKT to AXI4–Stream PPLD Core has a test bench which generates output waveforms using the Vivado VSim environment. The test bench is designed to run with the following parameters:

1) AXI4–Stream clock (**axis_aclk**) frequency: 200 MHz

2) AXI4–Lite CSR clock (**s_axis_csr_aclk**) frequency: 250 MHz

3) Parameter "**word_width**" is set to 2

4) Parameter "**default_packet_size**" is set to 256 (default)

5) Parameter "**has_override**" is set to FALSE (default)

6) Parameter "**has_fifo_rst_in_n**" is set to TRUE (default)

Once the reset (**axis_aresetn**) is released, the testbench begins providing data in the form of a counter to the AXI4–Stream slave. The data is divided into 4 64–word (4 bytes/ word) packets (distinguished by **s_axis_ppkt_tready** going LOW at the end of each packet) and continues until 4 packets (256 words total) have been provided. The **s_axis_ppkt_tlast** input is driven HIGH during the last (256th) word of the 4th packet to indicate the end of the stream.

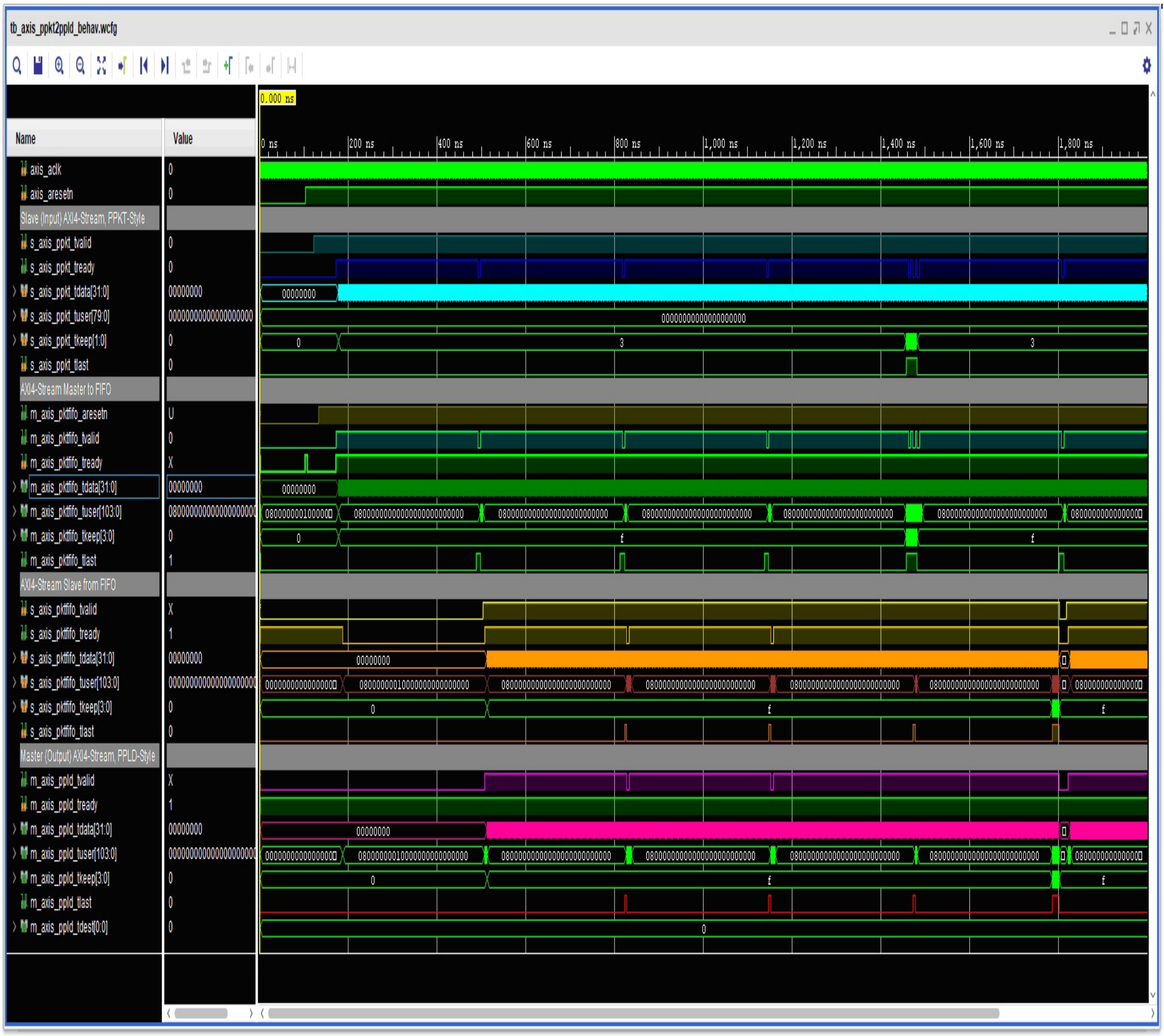
As the data arrives at the AXI4–Slave input, it is passed to the packet FIFO in the form of 256–byte packets that are distinguished by **m_axis_pktfifo_tlast** going HIGH at the end of each packet.

Once the first complete 256–byte packet is received by the FIFO, it asserts **s_axis_pktfifo_tready** to indicate that a packet is ready to be transferred. The core responds by transferring the data from the FIFO to the AXI4–Stream master.

The signals presented in the waveform window demonstrate this to the user as shown in Figure 6–8 on the next page.

6.6 **Simulation** (continued)

Figure 6–8: AXI4–Stream PPKT to AXI4–Stream PPLD Core Test Bench Simulation Output



6.7 Synthesis and Implementation

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

This page is intentionally blank