IP CORE MANUAL

NAVIGATOR

FPGA Design Kit

AXI4–Stream PPKT UltraRam FIFO IP

px_axis_ppkt_uram_fifo

PENTEK

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

Manual Part Number: 807.48421 Rev: 1.0 – September 20, 2018

Manual Revision History

Date	Version	Comments
9/20/18	1.0	Initial Release

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 UltraRam FIFO Core is a FIFO for a PPKT AXI4–Stream. The AXI4–Stream PPKT data width and the FIFO depth are configurable, and control, status and interrupts are accessible via the CSR AXI4–Lite bus. An optional FIFO Overflow LED output is also available.

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 PPKT UltraRam FIFO Core.

Features

- User–configurable width of input and output PPKT–style AXI4–Streams
- Supports input ready signal from a PPKT–style AXI4–Stream Slave in the user design
- User–configurable FIFO depth
- Register access through AXI4–Lite CSR interface
- Optional FIFO reset input for user design control
- Optional Overflow LED output to user design

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

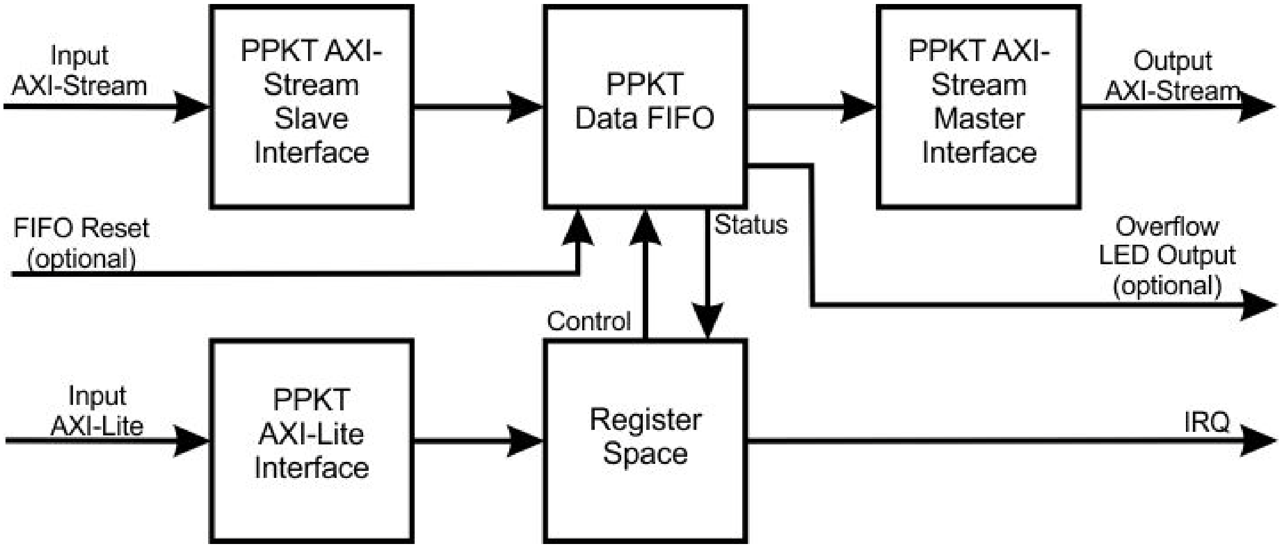
Chapter 1: Overview

1.1 Functional Description

The AXI4–STREAM PPKT UltraRam FIFO Core is a fifo for a PPKT–style AXI4–Stream which utilizes Xilinx UltraRam blocks as the storage medium. The AXI4–Stream PPKT data width and the FIFO depth are user configurable. Control, status, and interrupts are accessible via the CSR AXI4–Lite bus. An optional FIFO Overflow LED output and an optional reset input are also available.

Figure 1–1 is a top–level block diagram of the AXI4–Stream PPKT UltraRam FIFO Core. The modules within the block diagram are explained in other sections of this manual.

Figure 1–1: AXI4–Stream PPKT UltraRam FIFO Core Block Diagram



- ❑ **AXI4–Stream Slave Interface:** This module implements a PPKT–style AXI4–Stream Slave interface for the input data stream to the Core. For additional details about the AXI4–Stream Slave Interface, refer to **Section 3.1 AXI4–Stream Core Interfaces**.

- ❑ **PPKT Data FIFO:** This module comprises the Xilinx UltraRam–based FIFO. Parameters for data width and FIFO depth are user configurable.

1.1 Functional Description (continued)

- ❑ **AXI4–Stream Master Interface:** This module implements a PPKT–style AXI4–Stream Master interface for the output data stream for the Core. For additional details about the AXI4–Stream Slave Interface, refer to **Section 3.1 AXI4–Stream Core Interfaces**.
- ❑ **AXI4–Lite Interface:** This module implements a 32–bit AXI4–Lite Slave Interface to receive the memory read/write requests and also access the register space. For more details about the AXI4–Lite Interface, refer to **Section 3.1 AXI4–Lite Core Interfaces**.
- ❑ **Register Space:** This module contains the control, status and interrupt registers for the core. The registers are accessed through the AXI4–Lite interface.

1.2 Applications

This core is useful when the user needs a deep buffer for 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) *ARM AMBA AXI4 Protocol Version 2.0 Specification*
http://www.arm.com/products/system–ip/amba–specifications.php

Chapter 2: General Product Specifications

2.1 Standards

The AXI4–Stream PPKT UltraRam FIFO Core has interfaces that comply with the *AMBA AXI4–Lite Protocol Specification* and the *AMBA AXI4–Stream Protocol Specification*.

2.2 Performance

The performance of the AXI4–Stream PPKT UltraRam FIFO 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 –STREAM PPKT UltraRam FIFO 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 Zynq Ultrascale+ RFSOC –1 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 UltraRam FIFO Core is shown in Table 2–1. Resources have been estimated for the Zynq Ultrascale+ RFSOC xczu27dr–ffvg1517–1–e speed grade device, utilizing a 32–byte wide data bus and a fifo depth of 4096. These values were generated using the Vivado Design Suite. The resource usage will vary with the selected bus width and selection of optional signals.

Table 2–1: Resource Usage and Availability

Resource	# Used
LUTs	258
Flip–Flops	1024
URAMs	6

NOTE: Actual utilization may vary based on the user design in which the AXI4–Stream PPKT UltraRam FIFO 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 PPKT UltraRam FIFO 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_width_bytes	Integer	**Data Width (in Bytes):** This parameter defines the width of the data bus of the input and output AXI4-Stream in bytes. Valid widths are 2, 4, 8, 16, 32, 64 and 128 bytes. The default setting for this parameter is 32.
fifo_depth	Integer	**FIFO Depth:** This parameter defines the depth of the fifo. Valid depths are any integer value from 512 to 65536 words. The default setting for this parameter is 4096 words.
has_in_ready	Boolean	**Has Input Data Ready:** This parameter is used to indicate whether the incoming AXI Stream has the Data Ready input signal. It also enables an output data ready signal from the AXI4-Stream PPKT UltraRam FIFO Core to the user design. **NOTE:** Overrun (overflow) is disabled when this parameter is set to TRUE. The default setting for this parameter is FALSE**.**
has_in_reset	Boolean	**Has Reset Input:** When set to true, this parameter indicates that an active-low reset input is included in the design. The default setting for this parameter is TRUE.
has_fifo_ovfl_led	Boolean	**Has Fifo Overflow LED:** When set to true, this parameter indicates that an active-low overflow output to drive an LED is included in the design. **NOTE:** Overflow is disabled when "**has_in_tready**" is set to TRUE. The default setting for this parameter is TRUE.

Chapter 3: Port Descriptions

This chapter provides port descriptions for the following interface types:

- AXI4–Lite Core Interfaces
- AXI4–Stream Core Interfaces

3.1 AXI4–Lite Core Interfaces

The AXI4–Stream PPKT UltraRam FIFO 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, status and interrupt registers in the AXI4–STREAM PPKT Ultra–Ram FIFO Core. Table 3–1 defines the ports in the CSR Interface. See Chapter 4 for a Register memory map and bit definitions. See the *AMBA AXI4–Lite Specification* for more details on the AXI4–Lite interface.

Table 3-1: AXI4-Lite Slave Interface Port Descriptions

Port	Direction	Width	Description
s_axi_csr_aclk	Input	1	**Clock**
s_axi_csr_aresetn	Input	1	**Reset:** Active low. This will reset the state machine within the core.
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 DAC Waveform Generator Core.
s_axi_csr_awprot	Input	3	**Protection:** The DAC Waveform Generator 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 DAC Waveform Generator 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 DAC Waveform Generator 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.

Table 3-1: AXI4-Lite Slave Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
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 DAC Waveform Generator 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 DAC Waveform Generator 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.
s_axi_csr_bvalid	Output	1	**Write Response Valid:** This signal is asserted by the DAC Waveform Generator 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 DAC Waveform Generator Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the DAC Waveform Generator Core.

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

Port	Direction	Width	Description
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 DAC Waveform Generator 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 DAC Waveform Generator 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*.
s_axi_csr_rvalid	Output	1	**Read Data Valid:** This signal is asserted by the DAC Waveform Generator 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.
irq	Output	1	**Interrupt:** This is an active high, edge–type interrupt output.

3.1 AXI4–Stream Core Interfaces (continued)

The AXI4–Stream PPKT UltraRam FIFO Core has the following AXI4–Stream Interface, which is used to transfer data streams.

3.1.2 Stream Data Interface

This interface is used to transfer data from the slave input port, through the PPKT FIFO to the master data output port of the AXI4–Stream PPKT Ultra–Ram FIFO 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.

Table 3-2: Stream Data Interface Port Descriptions

Port	Direction	Width	Description
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 _tdata	Input	See description	**Input Data** Width = **data_width_bytes***8
s_axis_ppkt _tuser	Input	80	**Sideband Information:** This is user defined sideband information received alongside the data stream. It is formatted 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] (these bits only valid only with tlast)
s_axis_ppkt _tkeep	Input	**data _width _bytes**/2	**Data Keep:** This is a byte qualifier signal. Each bit of this signal corresponds to a byte in **s_axis_ppkt_tdata** i.e., bit 0 corresponds to the least significant byte of **s_axis_ppkt_tdata** and the most significant bit to the most significant byte. 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.

Table 3-2: Stream Data Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axis_ppkt _tlast	Input	1	**Data Last:** When asserted, the `s_axis_ppkt_tlast` input marks the last data in the current data frame.
s_axis_ppkt _tready	Output	1	**Output Data Ready:** This signal is asserted by the AXI4–Stream PPKT UltraRam FIFO Core when it is ready to accept data from the user logic. This output can be enabled by setting the generic parameter `has_in_tready` to True.
AXI4–Stream Master Interface			
m_axis_ppkt _tready	Input	1	**Data Ready:** Active High. This is an input tready signal from an AXI Stream slave in the user design indicating that it is ready to accept data. This input can be enabled by setting the generic parameter `has_in_tready` to True. When `has_in_tready` is True, data is transferred across the AXI Stream Master interface when both `m_axis_ppkt_tvalid` and `m_axis_ppkt_tready` are High on the same cycle. If the slave in the user design deasserts the ready signal when `m_axis_ppkt_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_ppkt _tdata	Output	See description	**Output Data** Width = data_width_bytes*8
m_axis_ppkt _tvalid	Output	1	**Output Data Valid:** This signal is asserted when data is valid on `m_axis_ppkt_tdata` bus.
m_axis_ppkt _tuser	Output	80	**Sideband Data:** This is user defined sideband output information transmitted alongside the data stream. It is formatted 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] (these bits only valid only with tlast)

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

Port	Direction	Width	Description
m_axis_ppkt_tkeep	Output	data_width_bytes/2	**Data Keep:** This is a byte qualifier signal. Each bit of this signal corresponds to a byte in m_axis_ppkt_tdata i.e., bit 0 corresponds to the least significant byte of m_axis_ppkt_tdata and the most significant bit to the most significant byte. When a bit is asserted, the data on m_axis_ppkt_tdata is considered valid. All m_axis_ppkt_tkeep bits must be '1' contiguously until m_axis_ppkt_tlast is asserted. When m_axis_ppkt_tlast is asserted, and the number of data samples is not a multiple of tdata width, tkeep its are set to '0' to indicate which data bytes are to be ignored.
m_axis_tlast	Output	1	**Data Last:** When asserted, m_axis_ppkt_tlast marks the last data in the current data frame.

3.2 I/O Signals

The I/O port/signal descriptions of the top level module of the AXI4–Stream PPKT UltraRam FIFO Core are discussed in Table.

Table 3-3: I/O Signals

Port/ Signal Name	Type	Direction	Description
fifo_resetn	std_logic	Input	**Optional FIFO Reset:** This active LOW input is associated with axis_aclk, and is only available when "has_in_reset" is set to TRUE.
fifo_reset_out_n	std_logic	Output	**FIFO Reset Output:** This active LOW output is associated with axis_aclk, and is a copy of the internally generated FIFO reset.
fifo_ovfl_led	std_logic	Output	**Optional Overflow LED Output:** This active HIGH output is associated with axis_aclk, and is only available when "has_fifo_ovfl_led" is set to TRUE.

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the Register Space of the AXI4–Stream PPKT UltraRam FIFO 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	Controls FIFO flush as well as read and write disables.
FIFO Count Register	0x04	RO	Indicates the current FIFO data count (in bytes)
Peak FIFO Count Register	0x08	RO	Indicates the FIFO peak data count (in bytes) since the last reset
FIFO Flags Register	0x0C	RO	Indicates FIFO Status – empty and full flags
Interrupt Enable Register	0x10	R/W	Interrupt enable bits
Interrupt Status Register	0x14	RO	Interrupt status bits
Interrupt Flag Register	0x18	R/CLR	Interrupt Flag bits

4.1 Control Register

This register controls the FIFO flush, read disable and write disable functions for the FIFO. The Mode Control Register is illustrated in Figure 4–1 and described in Table 4–2.

Figure 4–1: Control Register

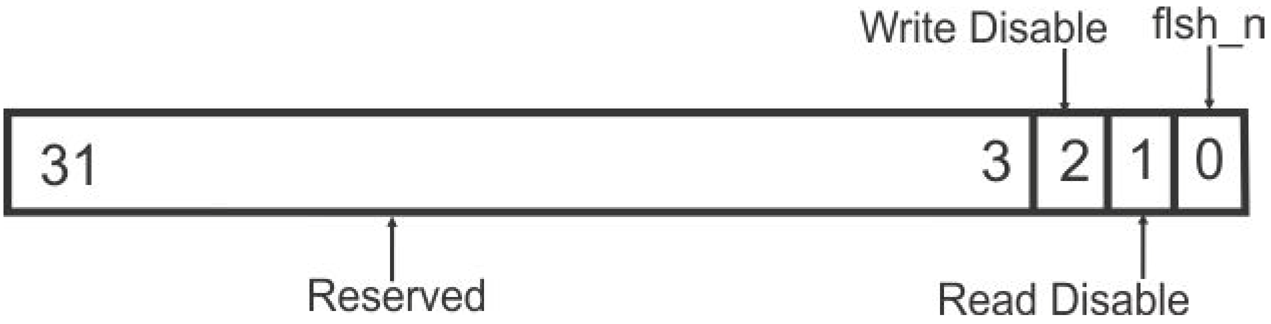


Table 4–2: Control Register (Base Address + 0x00)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:3	Reserved	N/A	N/A	**Reserved**
2	Write Disable	0	R/W	**Write Disable:** When set to '1' this bit disables writes to the FIFO.
1	Read Disable	0	R/W	**Read Disable:** When set to '1' this bit disables reads from the FIFO.
0	flsh_n	0	R/W	**Flush Fifo:** When set to '1' this bit forces the contents of the FIFO to be cleared and the FIFO is reset.

4.2 FIFO Count Register

This status register provides the current value of the data byte counter. This register is illustrated in Figure 4–2 and described in Table 4–3.

Figure 4–2: FIFO Count Register

Table 4–3: FIFO Count Register (Base Address + 0x04)

Bits	Field Name	Default Value	Access Type	Description
31:0	FIFO Count Status	0x00000000	RO	**FIFO Count:** Provides the current value of the FIFO data byte counter.

4.3 FIFO Peak Count Register

This status register provides the peak value of the data byte counter since the last reset. This register is illustrated in Figure 4–3 and described in Table 4–4.

Figure 4–3: FIFO Peak Count Register

Table 4–4: FIFO Peak Count Register (Base Address + 0x08)

Bits	Field Name	Default Value	Access Type	Description
31:0	Peak FIFO Count Status	0x00000000	RO	**Peak FIFO Count:** Provides the peak value of the FIFO data byte counter since the last reset.

4.4 FIFO Flags Register

This status register provides the current values of the FIFO's full and empty flags. This register is illustrated in Figure 4–4 and described in Table 4–5.

Figure 4–4: FIFO Flags Register

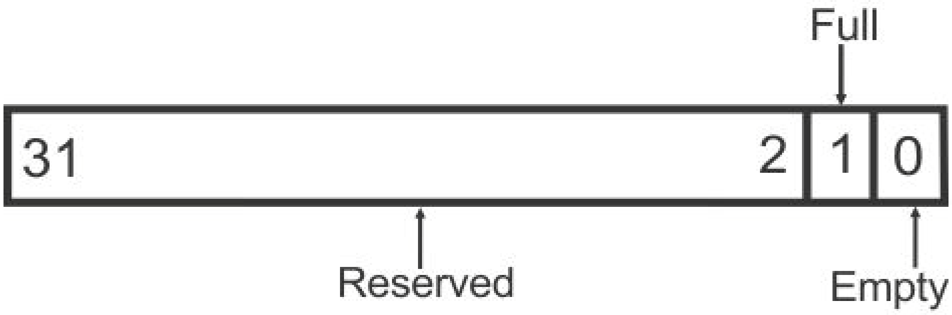


Table 4–5: FIFO Flags Register (Base Address + 0x0C)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:2	Reserved	N/A	N/A	**Reserved**
1	Full	0	RO	**FIFO Full:** Provides the status of the FIFO full flag
0	Empty	0	RO	**FIFO Empty:** Provides the status of the FIFO empty flag

4.5 Interrupt Enable Register

The bits in the Interrupt Enable Register are used to enable (or disable) the generation of interrupts based on the condition of certain circuit elements, known as interrupt sources. When a bit in this register associated with a given interrupt source is High, an interrupt will be generated by the rising edge of that source's Interrupt Status Register bit (See Section 4.6). This register is illustrated in Figure 4–5 and described in Table 4–6.

Figure 4–5: Interrupt Enable Register

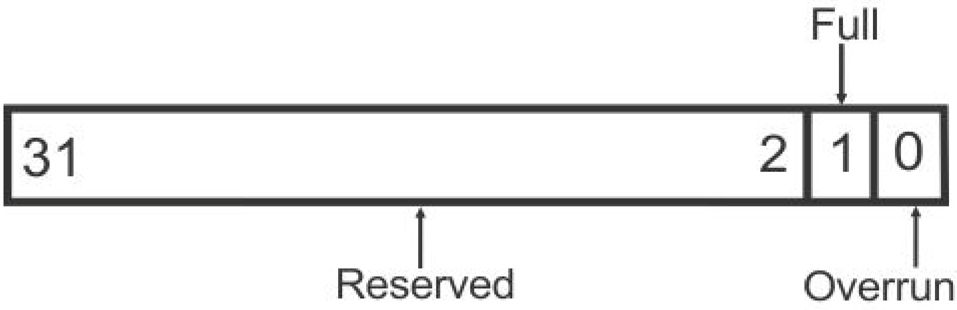


Table 4–6: Interrupt Enable Register (Base Address + 0x10)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:2	Reserved	N/A	N/A	**Reserved**
1	Full	0	R/W	**FIFO Full:** This bit enables or disables the FIFO full interrupt source. 0 = Disable interrupt 1 = Enable interrupt
0	Overrun	0	R/W	**FIFO Overrun:** This bit enables or disables the FIFO overrun interrupt source. 0 = Disable interrupt 1 = Enable interrupt **NOTE:** Overrun is disabled when flow control is present (parameter "**`has_in_tready`**" is set to TRUE).

4.6 Interrupt Status Register

The Interrupt Status Register has read–only access associated with each interrupt condition. A status bit changes to ‘1’ when the source interrupt occurs. When a status bit in this register changes to ‘1’ the corresponding flag bit in the Interrupt Flag Register is set to ‘1’. A status bit in this register clears to ‘0‘ when that interrupt condition clears, whereas the associated flag bit in the Interrupt Flag Register remains at logic ‘1’ until it is explicitly cleared by the user.

Some of the interrupt sources are transient and so may not appear in the Interrupt Status Register at the time it is read. In such cases, use the Interrupt Flag Register to see the interrupt conditions that have occurred. The Interrupt Status Register is illustrated in Figure 4–6 and described in Table 4–7.

Figure 4–6: Interrupt Status Register

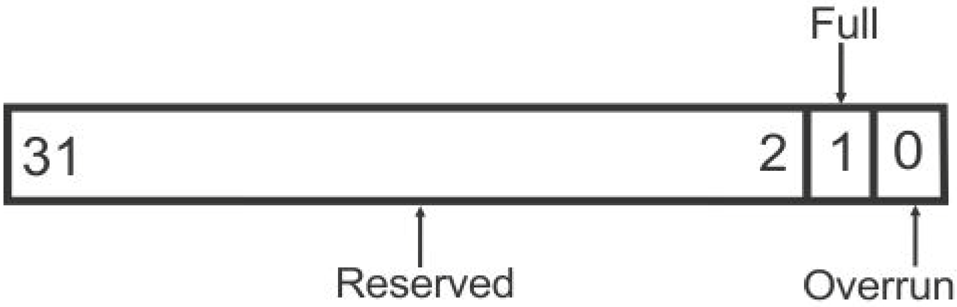


Table 4–7: Interrupt Status Register (Base Address + 0x14)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:2	Reserved	N/A	N/A	**Reserved**
1	Full	0	RO	**FIFO Full:** This bit indicates the status of the FIFO full interrupt source. 0 = No interrupt 1 = Interrupt condition is asserted
0	Overrun	0	RO	**FIFO Overrun:** This bit indicates the status of the FIFO overrun interrupt source. 0 = No interrupt 1 = Interrupt condition is asserted **NOTE:** Overrun is disabled when flow control is present (parameter "**`has_in_tready`**" is set to TRUE).

4.7 Interrupt Flag Register

The Interrupt Flag Register has read/clear access associated with each interrupt condition. When reset, this register has all bits set to '0' (cleared). Each flag bit in this register latches an interrupt occurrence. A '1' in any flag bit in this register indicates that an interrupt has occurred.

Note that when any status bit in the Interrupt Status Register, changes to '1' the corresponding flag bit in this register will also be set to '1'. However, when a status bit in the Interrupt Status Register clears from '1' to '0', the corresponding latched flag bit in this register does not clear, but remains at '1'. To clear the flag bits, write '1's to the desired bits. The flags are not affected by the Interrupt Enable Register. The Interrupt Flag Register is illustrated in Figure 4–7 and described in Table 4–8.

Figure 4–7: Interrupt Flag Register

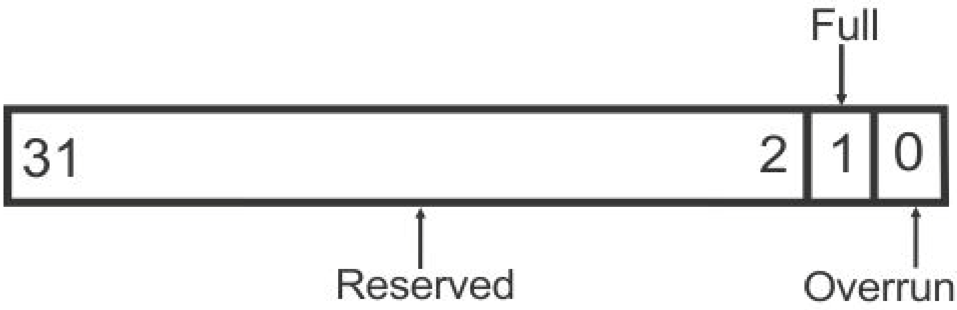


Table 4–8: Interrupt Flag Register (Base Address + 0x18)

Bits	Field Name	Default Value	Access Type	Description
31:2	Reserved	N/A	N/A	**Reserved**
1	Full	0	R/CLR	**FIFO Full:** This bit indicates status of the FIFO full interrupt flag. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
0	Overrun	0	R/CLR	**FIFO Overrun:** This bit indicates the status of the FIFO overrun interrupt flag. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch **NOTE:** Overrun is disabled when flow control is present (parameter "**`has_in_tready`**" is set to TRUE).

Chapter 5: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the AXI4–Stream PPKT UltraRam FIFO Core.

5.1 General Design Guidelines

The AXI4–STREAM PPKT UltraRam FIFO Core provides the required logic to implement a user–configurable packet data FIFO for a PPKT–style AXI4–Stream data 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 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 both the slave and master AXI4–stream interfaces, as well as the FIFO.

5.3 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 and the FIFO are reset.

Optional FIFO Reset Input: **`fifo_resetn`**.

This is an active–low synchronous reset associated with the **`axis_aclk`**. When asserted, the FIFO is reset and its contents are cleared. Setting the "**`has_in_reset`**" to TRUE (default setting) enables this function.

A reset output (**`fifo_reset_out_n`**), which is an active–low synchronous reset associated with **`axis_aclk`**, is available. This output is a copy of the internally generated FIFO reset.

5.4 Interrupts

This core has an edge–type (rising edge–triggered) interrupt output. It is synchronous with the **`s_axi_csr_aclk`**. On the rising edge of any interrupt signal, a one clock cycle wide pulse is output from the core on its **`irq`** output. Each interrupt event is stored in two registers, accessible on the **`s_axi_csr`** bus. The Interrupt Status Register always reflects the current state of the interrupt condition, which may have changed since the generation of the interrupt. The Interrupt Flag Register latches the occurrence of each interrupt, in a bit that retains its state until explicitly cleared.The interrupt flags can be cleared by writing '1' to the associated bit's location. All interrupt sources that are enabled (via the Interrupt Enable Register) are "OR ed" onto the **`irq`** output.

NOTE: All interrupt sources are latched in the Interrupt Flag Register, even when an interrupt source is not enabled to create an interrupt.

NOTE: Because this core uses edge–triggered interrupts, the fact that an interrupt condition may remain active after servicing will not cause the generation of a new interrupt. A new interrupt will only be generated by another rising edge on an interrupt source.

5.5 Interface Operation

Control/Status Register Interface: The control register interface 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 registers that can be accessed through this interface.

Stream Data Interface (input and output): This is a PPKT–style AXI4–Stream Interface – slave at the input and master at the output, and is associated with the **`axis_aclk`**. For more details about this interface refer to Section 3.1.

5.6 Programming Sequence

This section briefly describes the programming sequence for the AXI4–Stream PPKT UltraRam FIFO Core.

1) Ensure that the Interrupt Flag Register is cleared.

2) Enable the Interrupt Enable Register bits based on the user design requirement.

3) Write the desired values to the Control Register.

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

5.7 Timing Diagrams

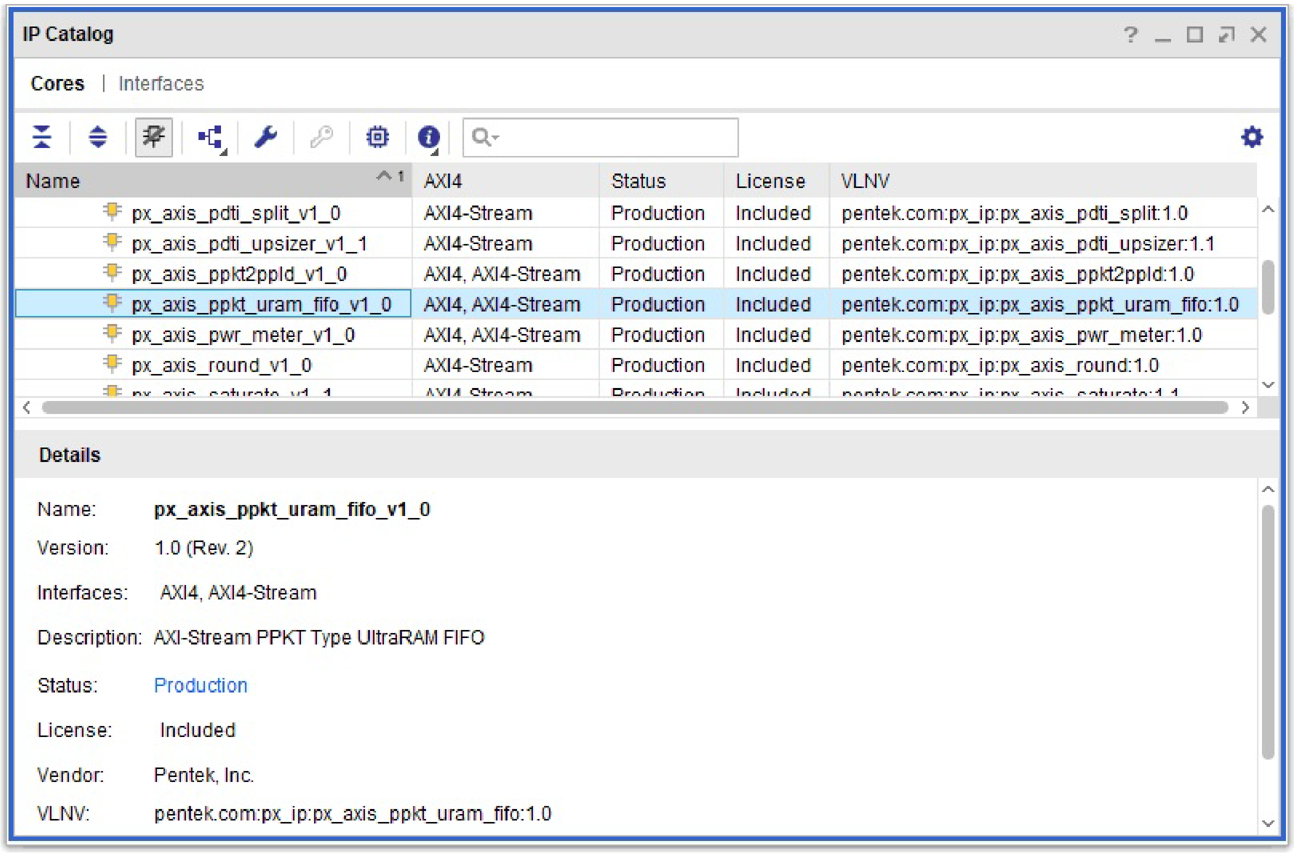
The timing diagram for the AXI4–Stream PPKT UltraRam FIFO Core, shown in Figure 6–3, is obtained by running the simulation of the test bench of the core in Vivado VSim environment. For more details about the test bench, refer to Section 6.5.

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek AXI4–Stream PPKT UltraRam FIFO 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_ppkt_uram_fifo_v1_0** as shown in Figure 6–1.

Figure 6–1: AXI4–Stream PPKT UltraRam FIFO Core in Pentek IP Catalog



Name	AXI4	Status	License	VLNV
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
px_axis_saturate_v1_1	AXI4-Stream	Production	Included	pentek.com:px_ip:px_axis_saturate:1.1

Name: **px_axis_ppkt_uram_fifo_v1_0**

Version: 1.0 (Rev. 2)

Interfaces: AXI4, AXI4-Stream

Description: AXI-Stream PPKT Type UltraRAM FIFO

Status: Production

License: Included

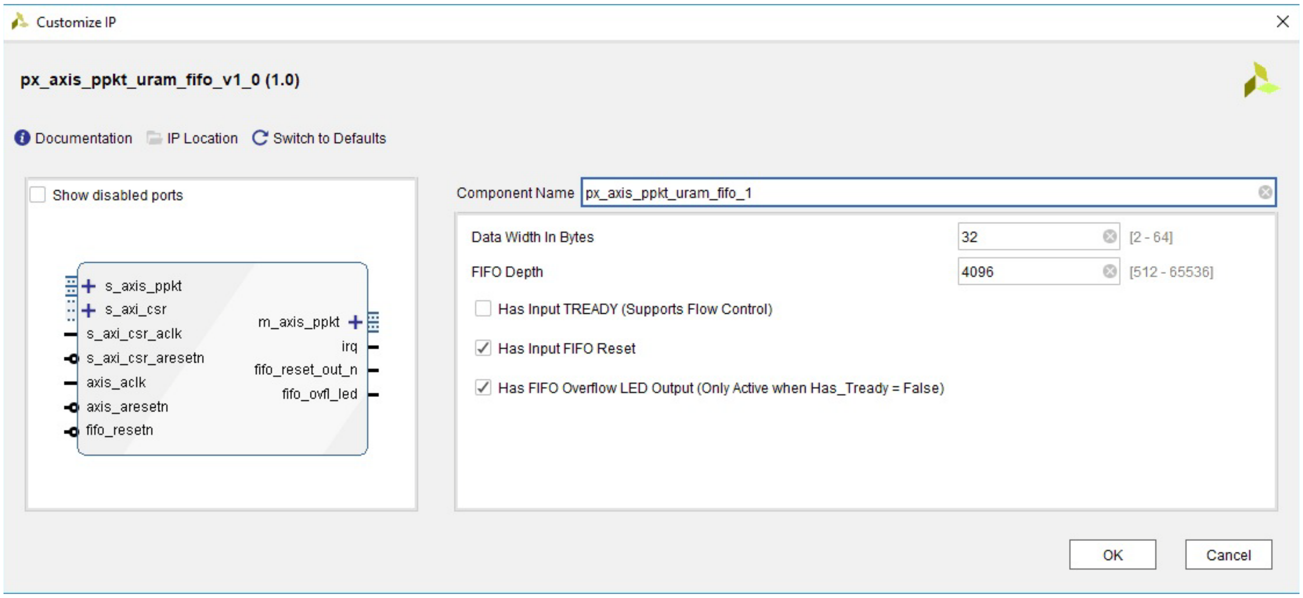
Vendor: Pentek, Inc.

VLNV: pentek.com:px_ip:px_axis_ppkt_uram_fifo:1.0

6.1 Pentek IP Catalog (continued)

When you select the **px_axis_ppkt_uram_fifo_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 UltraRam FIFO IP Core Symbol



px_axis_ppkt_uram_fifo_v1_0 (1.0)

□ Show disabled ports

Component Name: px_axis_ppkt_uram_fifo_1

Data Width In Bytes: 32 [2 - 64]

FIFO Depth: 4096 [512 - 65536]

□ Has Input TREADY (Supports Flow Control)

☑ Has Input FIFO Reset

☑ Has FIFO Overflow LED Output (Only Active when Has_Tready = False)

6.2 User Parameters

For a detailed explanation of the user parameters, refer to Section 2.5.

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 environment.

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 all of the Xilinx Ultrascale+ FPGA families that contain UltraRam.

Clock Frequencies

The AXI4−Stream clock maximum clock frequency is 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.

6.5 Simulation

The AXI4-Stream PPKT UltraRam FIFO 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: 500 MHz

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

3) Parameter "**has_in_tready**" is set to TRUE

4) Parameter "**has_in_reset**" is set to TRUE (default)

5) Parameter "**data_width_bytes**" is set to 2

6) Parameter "**fifo_depth**" is set to 2048

7) Parameter "**has_fifo_ovfl_led**" is set to TRUE (default)

8) Control Register bits 0, 1 and 2 are all set to '0' (default)

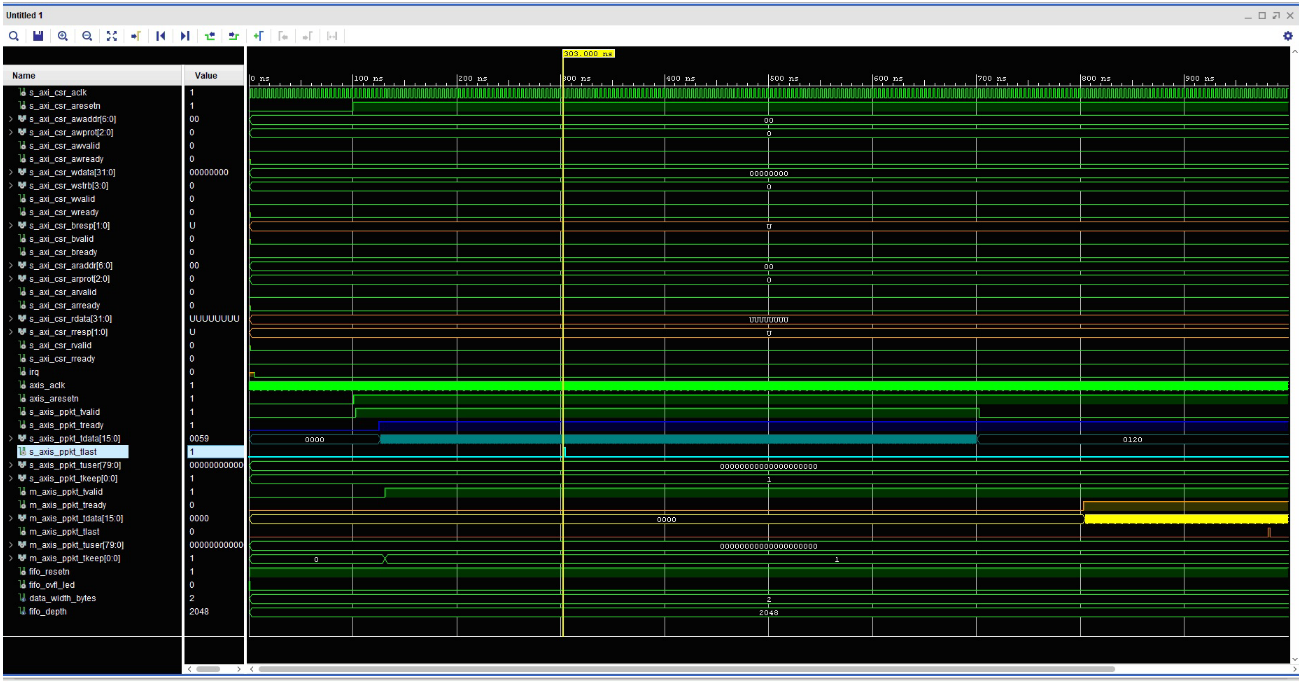
Once the reset is released, the testbench waits for the core to bring "**s_axis_ppkt_tready**" HIGH, then it begins providing data in the form of a counter to the AXI4-Stream slave port. The testbench will drive the "**s_axis_ppkt_last**" signal HIGH when the data reaches 0x0059, but the data stream (counter) will continue until it reaches 0x0120.

~100ns after the counter stops, the testbench will drive the "**m_axis_ppkt_tready**" signal high and the counter data that was stored in the FIFO will be streamed-out on the AXI4-Stream master port. The "**m_axis_ppkt_tlast**" signal will go HIGH when the output data = 0x0059.

When run, the simulation produces the results shown in Figure 6-3.

6.5 Simulation (continued)

Figure 6-3: AXI4-Stream PPKT UltraRam FIFO Core Test Bench Simulation Output



Name	Value
s_axi_csr_aclk	1
s_axi_csr_aresetn	1
s_axi_csr_awaddr[6:0]	00
s_axi_csr_awprot[2:0]	0
s_axi_csr_awvalid	0
s_axi_csr_awready	0
s_axi_csr_wdata[31:0]	00000000
s_axi_csr_wstrb[3:0]	0
s_axi_csr_wvalid	0
s_axi_csr_wready	0
s_axi_csr_bresp[1:0]	U
s_axi_csr_bvalid	0
s_axi_csr_bready	0
s_axi_csr_araddr[6:0]	00
s_axi_csr_arprot[2:0]	0
s_axi_csr_arvalid	0
s_axi_csr_arready	0
s_axi_csr_rdata[31:0]	UUUUUUUU
s_axi_csr_rresp[1:0]	U
s_axi_csr_rvalid	0
s_axi_csr_rready	0
irq	0
axis_aclk	1
axis_aresetn	1
s_axis_ppkt_tvalid	1
s_axis_ppkt_tready	1
s_axis_ppkt_tdata[15:0]	0059
s_axis_ppkt_tlast	1
s_axis_ppkt_tuser[79:0]	00000000000
s_axis_ppkt_tkeep[0:0]	1
m_axis_ppkt_tvalid	1
m_axis_ppkt_tready	0
m_axis_ppkt_tdata[15:0]	0000
m_axis_ppkt_tlast	0
m_axis_ppkt_tuser[79:0]	00000000000
m_axis_ppkt_tkeep[0:0]	1
fifo_resetn	1
fifo_ovfl_led	0
data_width_bytes	2
fifo_depth	2048

6.6 Synthesis and Implementation

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