Changes on page 21-35

IP CORE MANUAL

NAVIGATOR

FPGA Design Kit

10 Gigabit Ethernet UDP Transmit Core

px_10ge_udp_tx

PENTEK

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

Manual Part Number: 807.48401 Rev: 1.2 – June 7, 2018

Manual Revision History

Date	Version	Comments
6/16/17	1.0	Initial Release
10/20/17	1.1	Revised Sect 3.2, Sect 3.3, Sect 4.1, Sect 4.9, Sect 4.12, Sect 4.13, Sect 4.14, Sect 5.5, Sect 6.1, and Sect 6.2.
6/7/18	1.2	Revised Sect 1.6, Table 4–1, and Sect 5.6.

Legal Notices

Copyright

Trademarks

Printed in the United States of America.

Table of Contents

Page

Table of Contents

Page

Chapter 4: Register Space

Table of Contents

Page

Table of Contents

Page

This page is intentionally blank

IP Facts

Description

Pentek's Navigator™ 10 Gigabit Ethernet UDP Transmit IP Core takes AXI4–Lite data and packages it into UDP packets to be sent by the Xilinx PCS/PMA IP Core, or the Xilinx MAC IP Core. This core also handles setup for the PCS/PMA Core, or Xaui IP Core.

This core complies with the ARM® AMBA® AXI4 specification. This manual defines the hardware interface, software interface, and parameterization options for the 10 Gigabit Ethernet UDP Transmit IP Core.

Features

- Programmable MAC, IP, and UDP source and destination addresses.
- Selectable maximum packet sizes of 1K or 8K.
- Custom packet size enable.
- All controls and registers are accessible via AXI4–Lite.

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	encrypted 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 2017.2 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

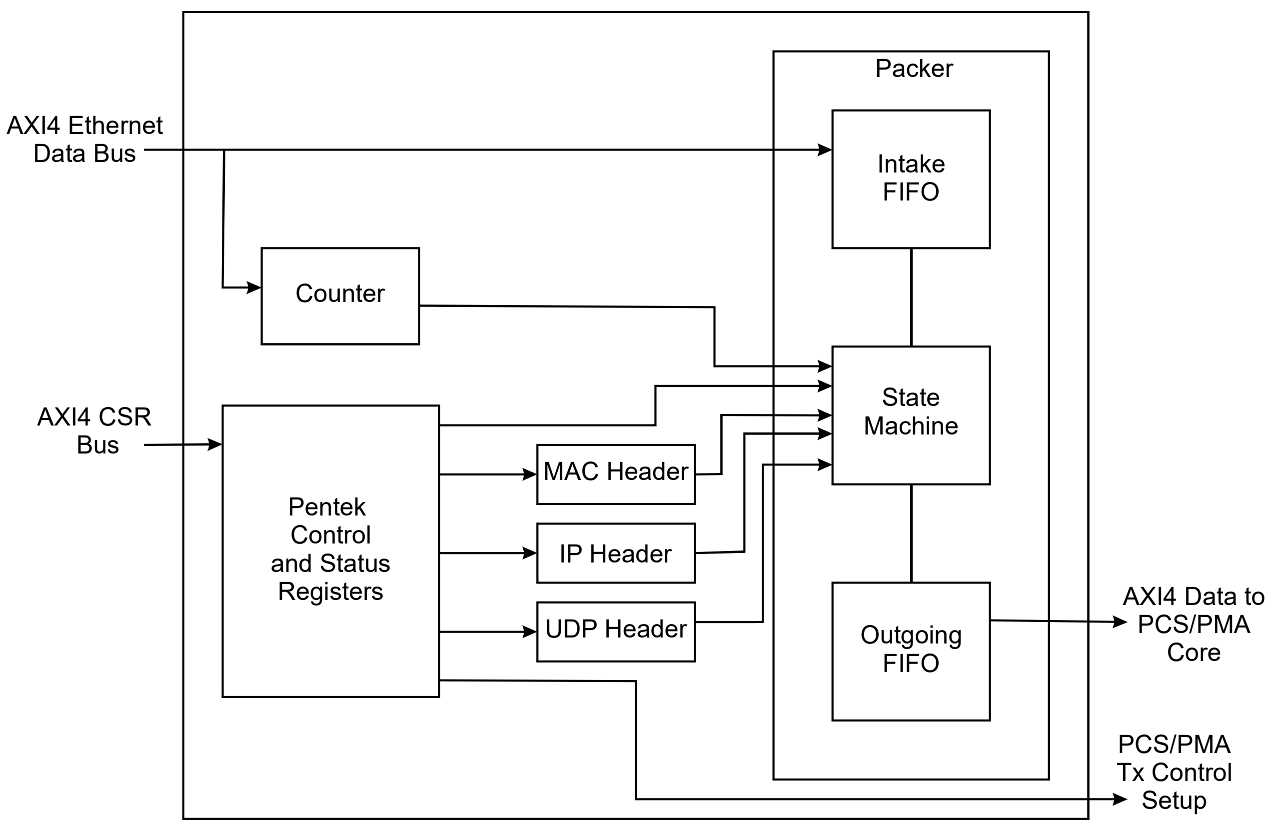
Chapter 1: Overview

1.1 Functional Description

Pentek's Navigator 10 Gigabit Ethernet UDP Transmit IP Core takes AXI4–Lite data and packages it into UDP packets to be sent by the Xilinx PCS/PMA IP Core, or Xilinx MAC IP core. This core also handles setup for the PCS/PMA core, or Xaui IP core.

Figure 1–1 is the top level block diagram of the 10 Gigabit Ethernet UDP Transmit Core. The modules within the block diagram are explained in the later sections of this manual.

Figure 1–1: 10 Gigabit Ethernet UDP Transmit Core Block Diagram



1.1 Functional Description (continued)

- **Pentek Control & Status Registers:** This module implements a 32–bit AXI4 Slave interface to access the Register Space.
- **Counter:** This module counts the incoming data. If custom packet size is enabled, this count is used in the state machine to count how much data there is to send. This count is also used to compute the IP header checksum.
- **MAC Header:** Reads the Control block to get the Destination and Source MAC address. This is used in the State machine when packaging the data.
- **IP Header:** Reads the Control block to get the Destination and Source IP address. This is used in the State machine when packaging the data. The IP header checksum is also calculated.
- **UDP Header:** Reads the Control block to get the Destination and Source ports. This is used in the State machine when packaging the data.
- **Intake FIFO:** Takes data in and waits for a whole packet if custom packet size is enabled or waits for 1K or 8K if custom size is disabled.
- **State Machine:** Builds the Ethernet packet 64 bits at a time. Sends the MAC layer, IP header, UDP header, then data payload. There are no pauses once the state machine starts building the packet. Calculates the correct tkeep on the last cycle of sending each packet.
- **Outgoing FIFO:** Stores a whole packet ready for the PCS/PMA core. Packet fifo handles the AXI control signals to the PCS/PMA core.

1.2 Applications

The Pentek Navigator 10 Gigabit Ethernet UDP Transmit IP Core can be used to package data for the following Xilinx IP Cores:

10G Ethernet Subsystem
10G Ethernet MAC
Tri Mode Ethernet MAC

1.3 System Requirements

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

1.4 Licensing and Ordering Information

The 10 Gigabit Ethernet UDP Transmit IP Core is included with all Pentek Navigator FPGA Design Kits for Pentek Jade series board products. Contact Pentek for licensing and ordering Information
(www.pentek.com).

1.5 Contacting Technical Support

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

1.6 Documentation

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

1) *Vivado Design Suite User Guide: Designing with IP*
2) *Vivado Design Suite User Guide: Programming and Debugging*
3) *ARM AMBA AXI4 Protocol Version 2.0 Specification*
http://www.arm.com/products/system–ip/amba–specifications.php

This page is intentionally blank

Chapter 2: General Product Specifications

2.1 Standards

The 10 Gigabit Ethernet UDP core has bus interfaces that comply with the *ARM AMBA AXI4–Lite Protocol Specification* and the *AMBA AXI4–Stream Protocol Specification*.

This core also complies with the following:

- 802.1Q–2011 – IEEE Standard for Local and metropolitan area networks––Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks
- IPv4 (Internet Protocol version 4)
- UDP (User Datagram Protocol)

2.2 Performance, Maximum Frequencies

- 10 Gigabit Ethernet capable core
- Ethernet IP runs at 156.25 MHz

2.3 Resource Utilization

The resource utilization of the 10 Gigabit Ethernet UDP 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
LUT	~~447~~ 384
Flip-Flops	~~398~~ 431

NOTE: Actual utilization may vary based on the user design in which the 10 Gigabit Ethernet UDP core is incorporated.

2.4 Limitations and Unsupported Features

This core cannot send data packets larger than 8192 bytes.

This page is intentionally blank

Chapter 3: Port Descriptions

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

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

3.1 AXI4–Lite Core Interfaces

The 10 Gigabit Ethernet UDP Transmit IP Core uses the Control/Status Register (CSR) interface to control, and receive status from, the user design.

3.1.1 Control/Status Register (CSR) Interface

The CSR interface is an AXI4–Lite Slave Interface that can be used to access the control and status registers in the 10 Gigabit Ethernet UDP Transmit 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 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 10 GbE UDP Transmit Core.
s_axi_csr_awprot	Input	3	**Protection:** The 10 GbE UDP Transmit 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 indi–cate that a valid write address is available on **s_axi_csr_ awaddr**. The 10 GbE UDP Transmit Core asserts **s_axi_csr_awready** when it is ready to accept the address. The **s_axi_csr_awvalid** must remain asserted until the ris–ing 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 10 GbE UDP Transmit 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 **s_axi_csr_wdata** signal. Each of these bits, when asserted indicate that the corre–sponding 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		**Write Ready:** This signal is asserted by the 10 GbE UDP Transmit 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 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 the 10 GbE UDP Transmit 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 10 GbE UDP Transmit Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the 10 GbE UDP Transmit 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 10 GbE UDP Transmit 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 10 GbE UDP Transmit 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 10 GbE UDP Transmit 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 10 GbE UDP Transmit Core when the read is complete and the read data is available on the **s_axi_csr_rdata**. It is held until **s_axi_csr_rready** is asserted by the user logic.
s_axi_csr_rready	Input		**Read Data Ready:** This signal is asserted by the user logic when it is ready to accept the Read Data.
irq	Output		**Interrupt:** This is an active High, edge type interrupt request output.

3.2 AXI4–Stream Core Interfaces

The 10 Gigabit Ethernet UDP Transmit IP Core has the following AXI4–Stream Interface, which is used to transfer data streams.

3.2.1 Stream Data (DATAIO_PD) Interface

This interface is used to transfer the Ethernet stream through the output ports of the 10 Gigabit Ethernet UDP Transmit Core. Table 3–2 defines the ports in the Stream Data Interface. This interface is an AXI4–Stream Master and Slave Interface that is used to output the data to the Ethernet core. This AXI4–Stream bus is synchronous with Ethernet Clock (**coreclk_out**) input of the core. See the *AMBA AXI4–Stream Specification* for more details on the operation of the AXI4–Stream Interface.

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

Port	Direction	Width	Description
axis_aclk	Input	1	Clock for core. Must come from Xilinx IP coreclk_out
axis_aresetn	Input	1	Reset for core. Must come from resetdone_out from PCS/PMA IP Core
s_axis_eth_tvalid	Input	1	Input data valid
s_axis_eth_tready	Output	1	‘1’ when the core is ready to accept data
s_axis_eth_tdata	Input	64	Data to be unpacked
s_axis_eth_tkeep	Input	8	Tkeep for tdata. Must be FF till tlast= ‘1’
s_axis_eth_tlast	Input	1	Tlast for tdata
m_axis_tx_tvalid	Output	1	Tvalid going to Xilinx IP core.
m_axis_tx_tready	Input	1	Tready going from Xilinx IP core to this one.
m_axis_tx_tdata	Output	64	Tdata going to Xilinx IP core.
m_axis_tx_tkeep	Output	8	Tkeep going to Xilinx IP core.
m_axis_tx_tlast	Output	1	Tlast going to Xilinx IP core.
m_axis_tx_tuser	Output	1	Tuser going to Xilinx core. Used to abort packets.

3.3 I/O Signals

The I/O port/signal descriptions of the top level module of the 10 Gigabit Ethernet UDP Transmit Core are provided in Table 3–3..

Table 3–3: I/O SIgnals

Port/Signal Name	Type	Direction	Description
Data Signals			
xilinx_core_rdy	std_logic	Input	Signal from Xilinx IP core notifying user that core is ready to be used.

This page is intentionally blank

Chapter 4: Register Space

This chapter provides the memory maps and register descriptions for the register space of the **px_10ge_udp_tx**. The memory maps are provided in Table 4–1 through Table 4–3. Table 4–4 through Table 4–16 provide further details.

4.1 Memory Maps

Upper Source MAC 0x00 R/W Upper Bits of MAC source address
Lower Source MAC 0x04 R/W Lower Bits of MAC source address

Table 4–1: Memory Map: Control Registers			
Register Name	**Address (Base Address +)**	**Access**	**Description**
Upper Dest MAC	~~0x00~~ 0x08	R/W	Upper bits of MAC destination address
Lower Dest MAC	~~0x04~~ 0x0C	R/W	Lower bits of MAC destination address
IP Source	~~0x08~~ 0x10	R/W	IP Source address
IP Destination	~~0x0C~~ 0x14	R/W	IP Destination address
UDP Source Port	~~0x10~~ 0x18	R/W	Bits 15:0 – UDP Source address
UDP Destination Port	~~0x14~~ 0x18	R/W	Bits ~~15:0~~ 31:16 – UDP Destination address
Core Function Control	~~0x18~~ 0x1C	R/W	Bit 0: Resetn Control Bit 1: Intake FIFO Enable/Reset Bit 2: Packet Creation Enable Bit 3: Packet Size Sel. 0=1k 1=8k Bit 4: Custom Packet Size Enable

combined into one register

Table 4–2: Memory Map: Status Registers			
Register Name	**Address (Base Address +)**	**Access**	**Description**
TX Status	~~0x1C~~ 0x20	R	Bit 0: xilinx_core_ready Bit 1: Counter Error Bit 2: State Machine Error
Intake FIFO Status	~~0x20~~ 0x24	R	Bit 0: Intake FIFO Empty Bit 1: Intake FIFO Full Bits 16:2 – Intake FIFO Count
Outgoing FIFO Status	~~0x24~~ 0x28	R	Bit 0: Intake FIFO Empty Bit 1: Intake FIFO Full Bits 16:2 – Intake FIFO Count

Table 4–3: Memory Map: Interrupt Enable/Status/Flag Registers			
Register Name	**Address (Base Address +)**	**Access**	**Description**
Interrupt Enable Register	~~0x28~~ 0x2C	R/W	Bit 0: xilinx_core_ready Bit 1: Intake FIFO Empty Bit 2: Intake FIFO Full Bit 3: Outgoing FIFO Empty Bit 4: Outgoing FIFO Full
Interrupt Status Register	~~0x2C~~ 0x30	R	Bit 0: xilinx_core_ready Bit 1: Intake FIFO Empty Bit 2: Intake FIFO Full Bit 3: Outgoing FIFO Empty Bit 4: Outgoing FIFO Full
Interrupt Flag Register	~~0x30~~ 0x3C	R/CLR	Bit 0: xilinx_core_ready Bit 1: Intake FIFO Empty Bit 2: Intake FIFO Full Bit 3: Outgoing FIFO Empty Bit 4: Outgoing FIFO Full

Add section for
Upper Source Mac 0x00
Lower Source Mac 0x04
Similar to Upper Dest MAC
and Lower Dest MAC

4.2 Upper Dest MAC

This register is used to control the upper bits of the MAC destination address. This register is illustrated in Figure 4–1 and described in Table 4–4.

Figure 4–1: Upper Dest MAC

31	Ethernet MAC Address bits 47 → 16	0

Table 4–4: Upper Dest MAC (Base Address + ~~0x00~~) 0x08				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:0	MAC Address	0xFFFFFFFF	R/W	**MAC Address**

4.3 Lower Dest MAC

This register is used to control the lower bits of the MAC destination address. This register is illustrated in Figure 4–2 and described in Table 4–5.

Figure 4–2: Lower Dest MAC

31 Reserved 16	15 Ethernet MAC Address bits 15 → 0 0

Table 4–5: Lower Dest MAC (Base Address + ~~0x04~~) 0x0C				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:16	Reserved	N/A	N/A	**Reserved**
15:0	MAC Address	0xFFFFFFFF	R/W	**MAC Address**

4.4 IP Source

This register is used to control the IP source address. This register is illustrated in Figure 4–3 and described in Table 4–6.

Figure 4–3: IP Source

31	IP Source Address	0

Table 4–6: IP Source (Base Address + 0~~x08~~) 0x10				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:0	IP Source	0x00000000	R/W	**IP Source Address**

4.5 IP Destination

This register is used to control the upper bits of the IP destination address. This register is illustrated in Figure 4–4 and described in Table 4–7.

Figure 4–4: IP Destination

31	IP Destination Address	0

Table 4–7: IP Destination (Base Address + ~~0x0C~~) 0x14				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:0	IP Destination	0x00000000	R/W	**IP Destination Address**

4.6 ~~UDP Source Port~~ UDP Destination and Source Port

This register is used to control the UDP source address. This register is illustrated in Figure 4–5 and described in Table 4–8.

Figure 4–5: UDP Source Port

31 ~~Reserved~~ 16	15 Source UDP Port 0

Destination UDP Port

Table 4–8: UDP Source (Base Address + ~~0x10~~) 0x18				
Bits	**Field Name** UDP Destination	**Default Value**	**Access Type**	**Description**
31:16	~~Reserved~~	~~N/A~~ 0x0000	~~N/A~~ R/W	~~**Reserved**~~ Destination UDP Port
15:0	UDP Source	0x0000	R/W	**UDP Source Port**

4.7 ~~UDP Destination Port~~

Combined in UDP Port Register

~~This register is used to control the UDP destination port. This register is illustrated in Figure 4–6 and described in Table 4–9.~~

~~Figure 4–6: UDP Destination Port~~

~~31 Reserved 16~~	~~15 Source UDP Port 0~~

~~Table 4–9: UDP Destination (Base Address + 0x14)~~				
~~Bits~~	**~~Field Name~~**	**~~Default Value~~**	**~~Access Type~~**	**~~Description~~**
~~31:16~~	~~Reserved~~	~~N/A~~	~~N/A~~	**~~Reserved~~**
~~15:0~~	~~UDP Destination~~	~~0x0000~~	~~R/W~~	**~~UDP Destination Port~~**

4.8 Core Function Control

This register is used to control the functions of this core. This register is illustrated in Figure 4–7 and described in Table 4–10.

Figure 4–7: Core Function Control

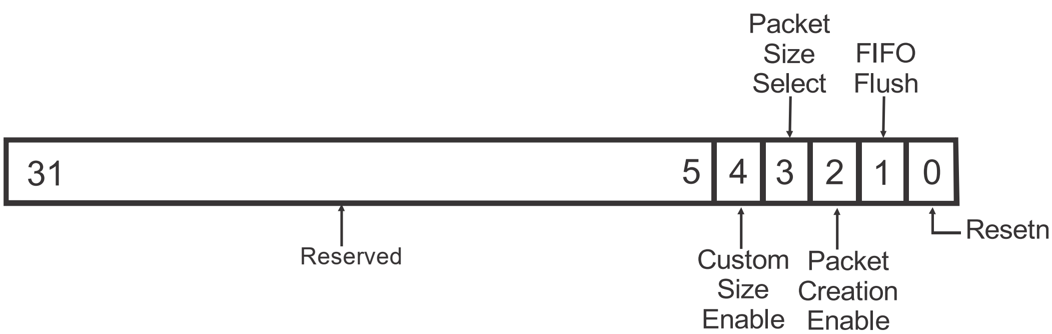


Table 4–10: Core Function Control (Base Address + ~~0x18~~) 0x1C				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:5	Reserved	N/A	N/A	**Reserved**
4	Custom Size Enable	0	R/W	**When enabled, packet size will be according to tlast of corresponding data or up to the maximum packet size: 1K or 8K.**
3	Packet Size Select	0	R/W	**0 = 1K packet** **1 = 8K packet**
2	Packet Creation Enable	0	R/W	**Starts packet creation. Intake FIFO will continue to function.**
1	FIFO Enable	0	R/W	**User intake FIFO Enable/Reset.**
0	Resetn Control	0	R/W	**User–controllable core reset.**

4.9 TX Status

This register is used to show the transmit statistics. **`xilinx_core_rdy`** indicates that the Xilinx core is ready. Bit 1 shows any errors involving counting incoming data. This bit could indicate a counter overflow. Bit 2 indicates any error involving the state machine reaching an invalid state or condition. This register is illustrated in Figure 4–8 and described in Table 4–11.

Figure 4–8: TX Status

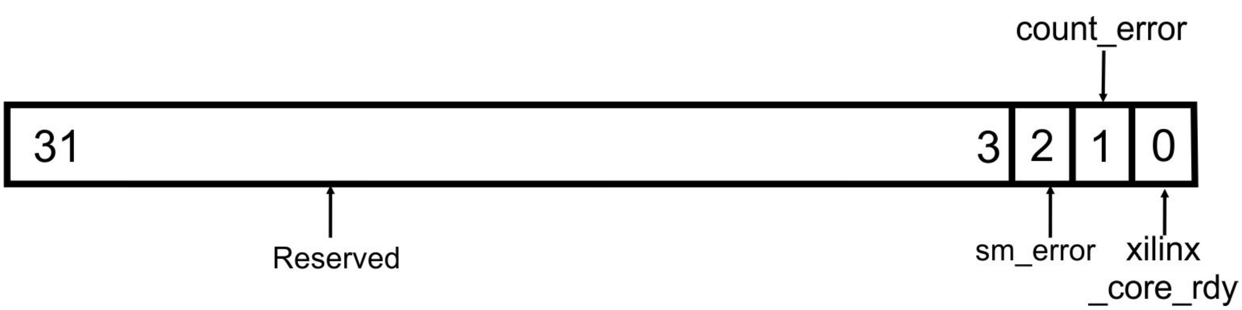


Table 4–11: TX Status (Base Address + ~~0x1C~~) 0x20

Bits	Field Name	Default Value	Access Type	Description
31:3	Reserved	N/A	N/A	**Reserved**
2	sm_error	0x0000	R	**State Machine Error:** An error occured in the state machine logic.
1	count_error	0	R	**Count Error:** An error occured with the counter.
0	xilinx_core_rdy	0	R	**TX_ready from Xilinx core.**

4.10 Intake FIFO Status

This register is used to show the status of the input FIFO. This register is illustrated in Figure 4–9 and described in Table 4–12.

Figure 4–9: Intake FIFO Status

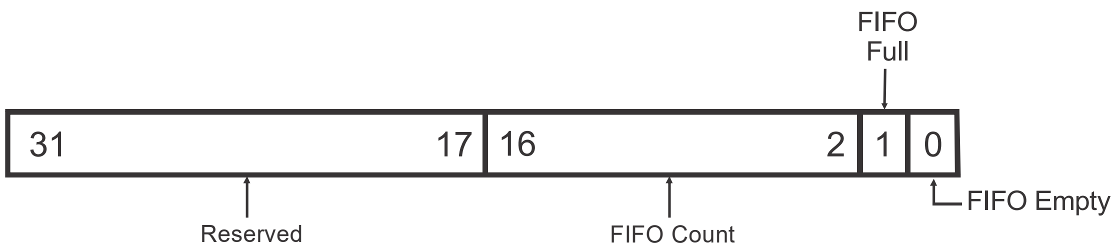


Table 4–12: Intake FIFO Status (Base Address + ~~0x20~~) 0x24				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:17	Reserved	N/A	N/A	**Reserved**
16:2	Intake FIFO Count	0x0000	R	**Intake FIFO count in bytes.**
1	Intake FIFO Full	0	R	**Intake FIFO full.**
0	Intake FIFO Empty	0	R	**Intake FIFO empty.**

4.11 Outgoing FIFO Status

This register is used to show the status of the outgoing FIFO. This register is illustrated in Figure 4–10 and described in Table 4–13.

Figure 4–10: Outgoing FIFO Status

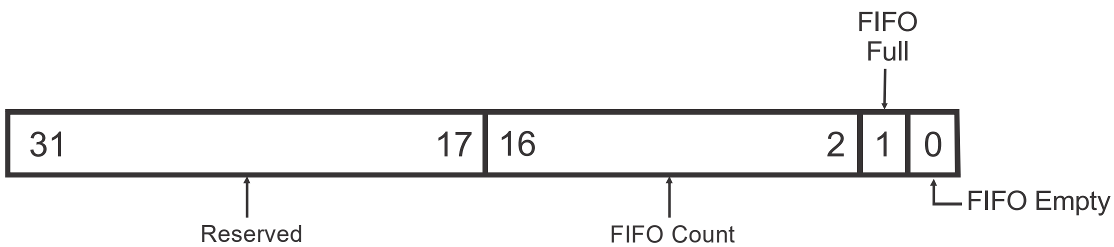


Table 4–13: Outgoing FIFO Status (Base Address + ~~0x24~~) 0x28				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:17	Reserved	N/A	N/A	**Reserved**
16:2	Outgoing FIFO Count	0x0000	R	**Outgoing FIFO count in bytes.**
1	Outgoing FIFO Full	0	R	**Outgoing FIFO full.**
0	Outgoing FIFO Empty	0	R	**Outgoing FIFO empty.**

4.12 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 4.13). This register is illustrated in Figure 4–11 and described in Table 4–14.

Figure 4–11: Interrupt Enable Register

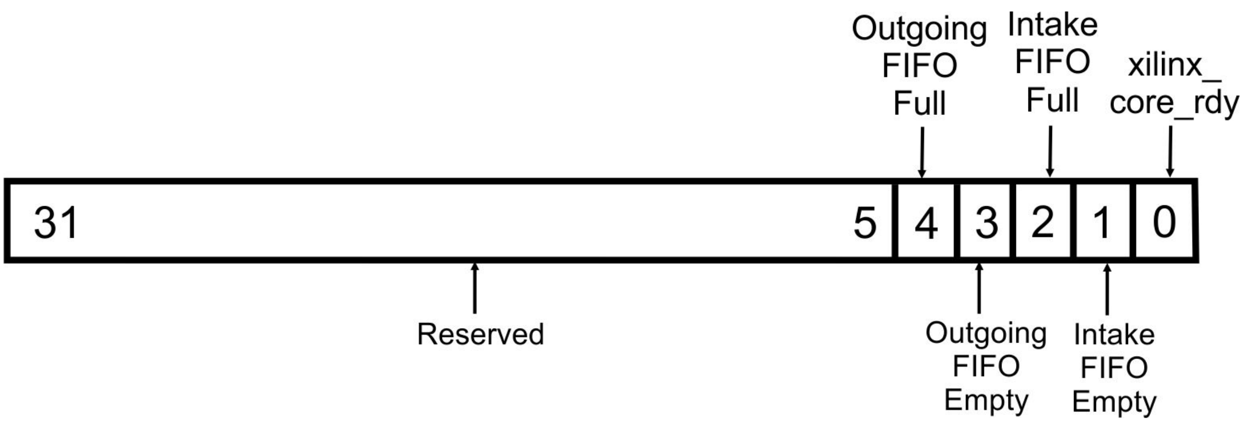


Table 4–14: Interrupt Enable Register (Base Address + ~~0x28~~) 0x2C				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:5	Reserved	N/A	N/A	**Reserved**
4	Outgoing FIFO Full	0	R/W	**Interrupt enable for outgoing FIFO going full.**
3	Outgoing FIFO Empty	0	R/W	**Interrupt enable for outgoing FIFO going empty.**
2	Intake FIFO Full	0	R/W	**Interrupt enable for intake FIFO going full.**
1	Intake FIFO Empty	0	R/W	**Interrupt enable for intake FIFO going empty.**
0	xilinx_core_rdy	0	R/W	**Interrupt enable for xilinx_core_rdy.**

4.13 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. This register is illustrated in Figure 4–12 and described in Table 4–15.

Figure 4–12: Interrupt Status Register

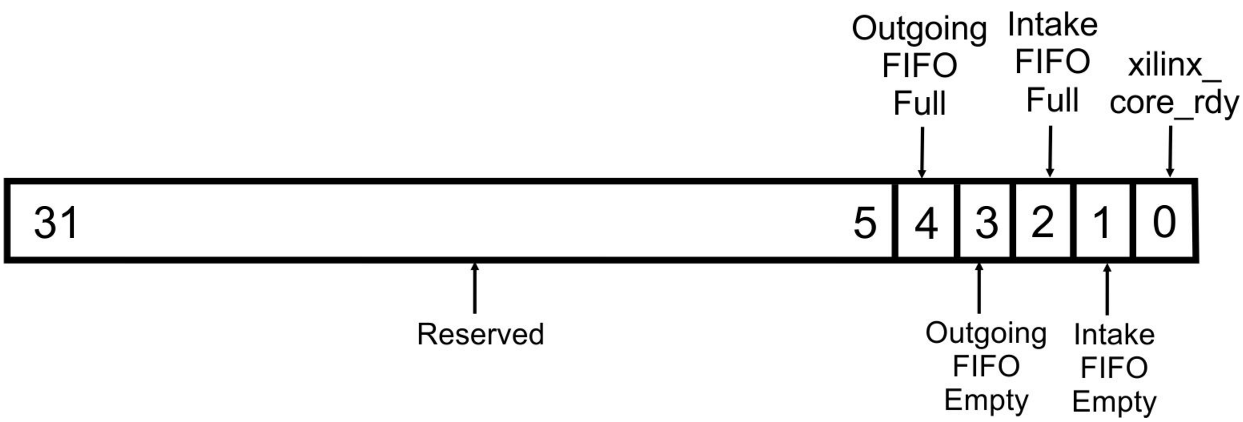


Table 4–15: Interrupt Status Register (Base Address + ~~0x2C~~) 0x30

Bits	Field Name	Default Value	Access Type	Description
31:5	Reserved	N/A	N/A	**Reserved**
4	Outgoing FIFO Full	0	R	**Interrupt status for outgoing FIFO going full.**
3	Outgoing FIFO Empty	0	R	**Interrupt status for outgoing FIFO going empty.**
2	Intake FIFO Full	0	R	**Interrupt status for intake FIFO going full.**
1	Intake FIFO Empty	0	R	**Interrupt status for intake FIFO going empty.**
0	xilinx_core_rdy	0	R	**Interrupt status for xilinx_core_rdy going low.**

4.14 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. This register is illustrated in Figure 4–13 and described in Table 4–16.

Figure 4–13: Interrupt Flag Register

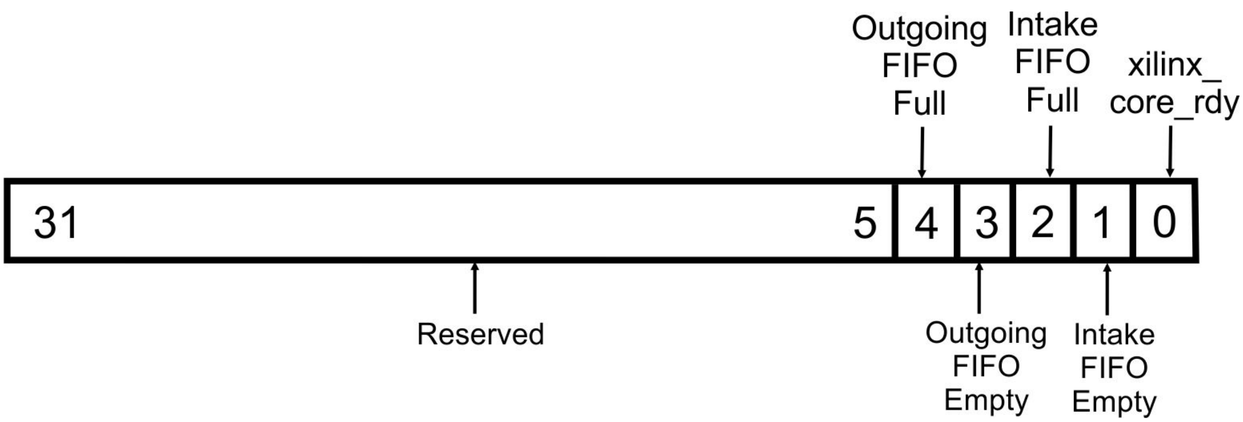


Table 4–16: Interrupt Flag Register (Base Address + ~~0x30~~) 0x34				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:5	Reserved	N/A	N/A	**Reserved**
4	Outgoing FIFO Full	0	R/W	**Interrupt flag for outgoing FIFO going full.**
3	Outgoing FIFO Empty	0	R/W	**Interrupt flag for outgoing FIFO going empty.**
2	Intake FIFO Full	0	R/W	**Interrupt flag for intake FIFO going full.**
1	Intake FIFO Empty	0	R/W	**Interrupt flag for intake FIFO going empty.**
0	xilinx_core_rdy	0	R/W	**Interrupt flag for xilinx_core_rdy going low.**

This page is intentionally blank

Chapter 5: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the 10 Gigabit Ethernet UDP Transmit Core.

5.1 General Design Guidelines

The 10 Gigabit Ethernet UDP Transmit Core packetizes data for the Xilinx MAC Core.

5.2 Clocking

AXI4–Lite Clock: **s_axi_csr_aclk**

The **s_axi_csr_aclk** is used to clock the AXI4–Lite Control/Status Register (**s_axi_csr**) interface of the core.

AXI4–Stream Interface Clock: **axis_aclk**

This clock is used to clock the AXI4–Stream inputs and outputs of the core as well as clocking all the logic in the core.

5.3 Resets

Main resets: **axis_aresetn**, **s_axi_csr_aresetn**

This is an active low synchronous reset associated with the **axis_aresetn**. When asserted, all state machines in the core are reset, all FIFOs are flushed. All the control registers are cleared back to their initial default states using **s_axi_csr_aresetn**.

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 it's **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.

5.4 Interrupts (continued)

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: This is the control/status register Interface. It is associated with the **s_axi_csr_aclk**. It is a standard AXI4–Lite type interface. See Chapter 4 for the control register memory map, for more details on the registers that can be accessed through this interface.

Stream Data (axis_eth) Interface: This interface is used to transfer input data streams. It is a standard AXI4–Stream Slave and Master Interface. For more details about this interface refer to Table 3–2.

5.6 Programming Sequence

The programming sequence for this core is as follows:

1) Take core out of reset by writing 0x1 to CSR reg 0x18.

2) Enable State machine by writing 0x05 to CSR reg 0x18.

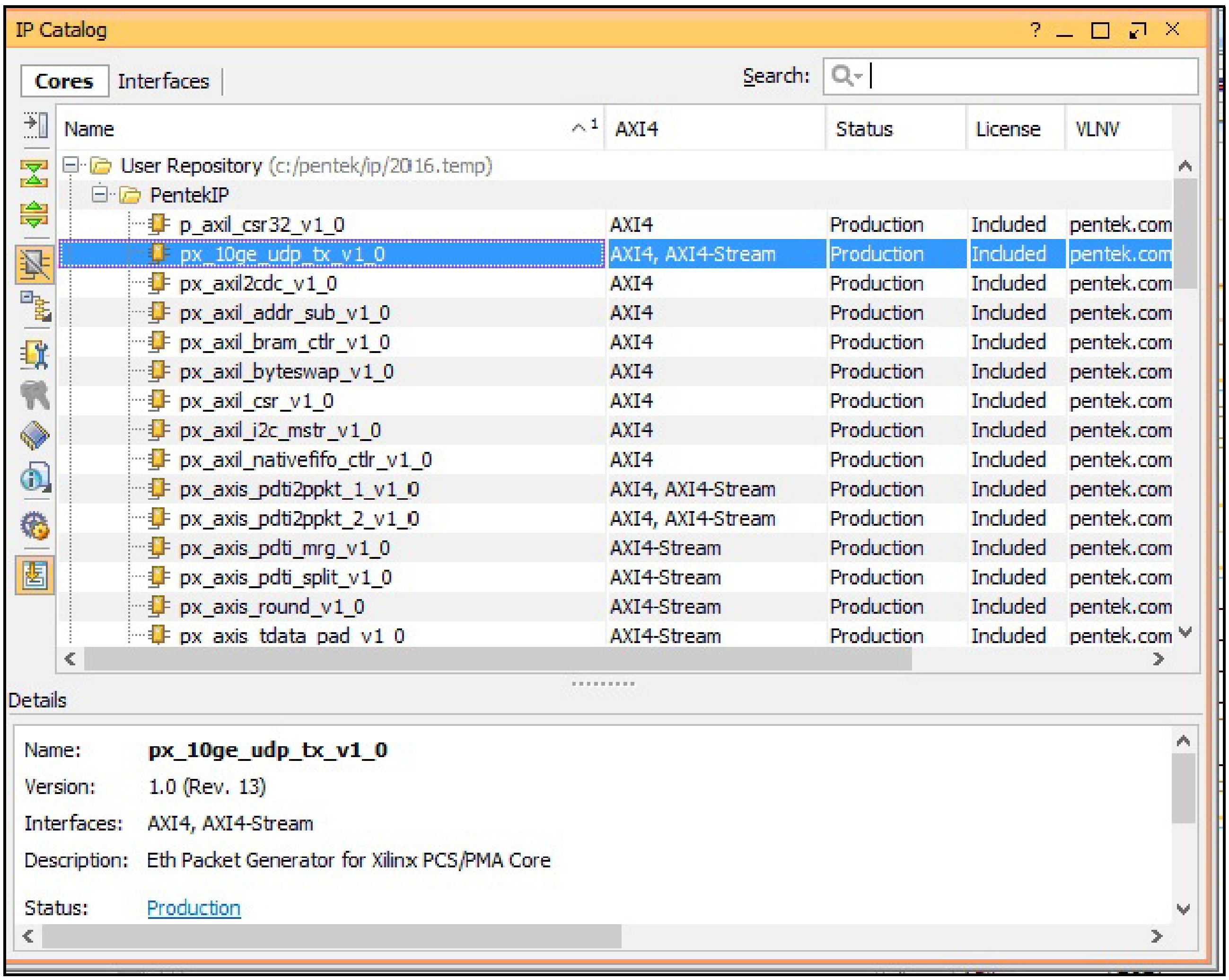
3) Enable FIFO by writing 0x7 to CSR reg 0x18

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the 10 Gigabit Ethernet UDP Transmit Core. It also includes simulation, synthesis, and implementation steps that are specific to this core. This IP 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_10ge_udp_tx_v1_0** as shown in Figure 6–1.

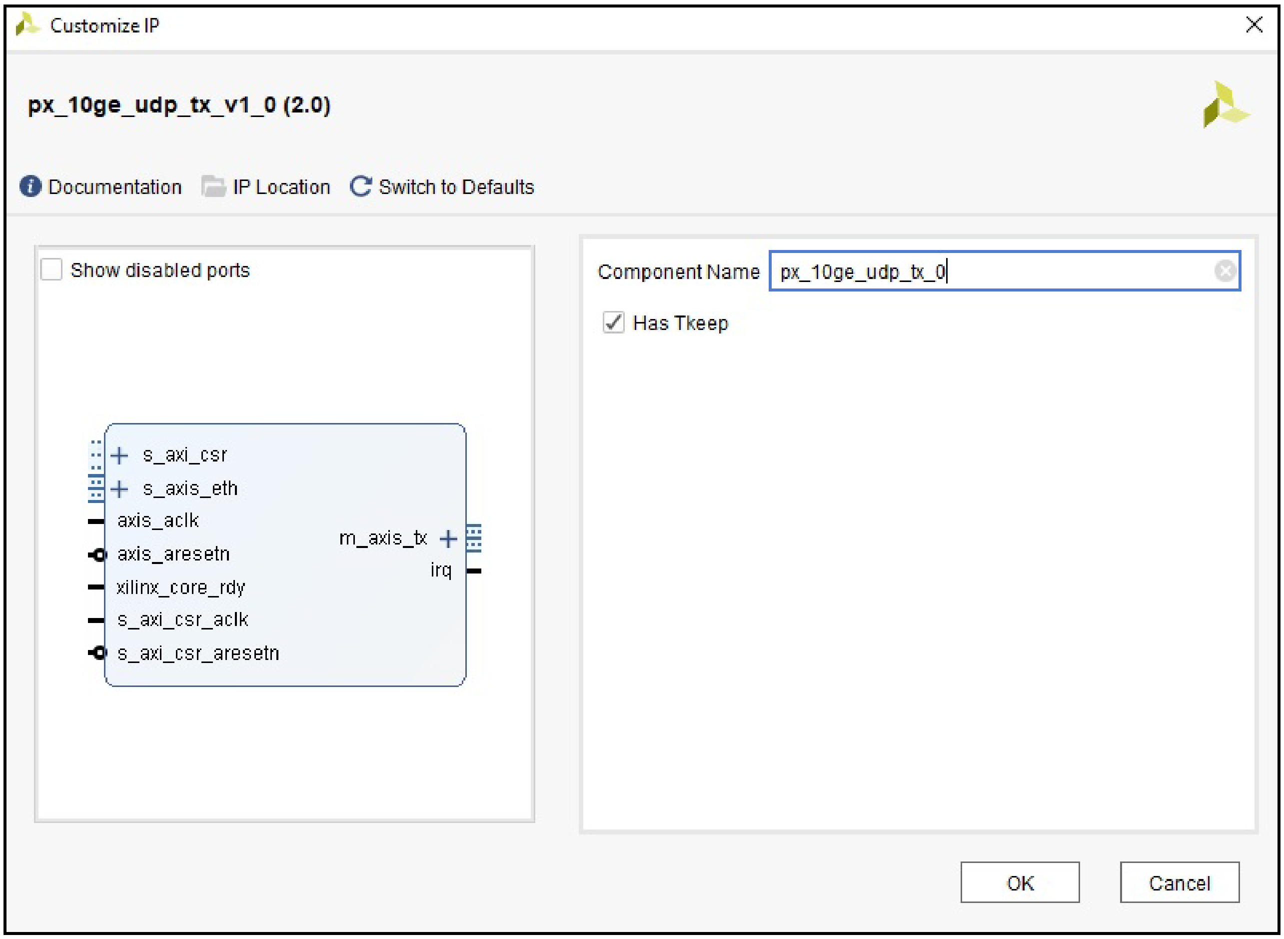
Figure 6–1: 10 Gigabit Ethernet UDP Transmit Core in Pentek IP Catalog



6.1 Pentek IP Catalog (continued)

When you select the **px_10ge_udp_tx_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: 10 Gigabit Ethernet UDP Transmit Core IP Symbol



6.2 User Parameters

`has_tkeep`: If it is enabled the user can specify **`tkeep`**. If it is disabled the internal **`tkeep`** is set to all 1's.

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 Kintex Ultrascale FPGAs.

Clock Frequencies

The maximum **`axis_aclk`** frequency for this IP core is **156.25 MHz** while 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

Figure 6–3: Setup Registers

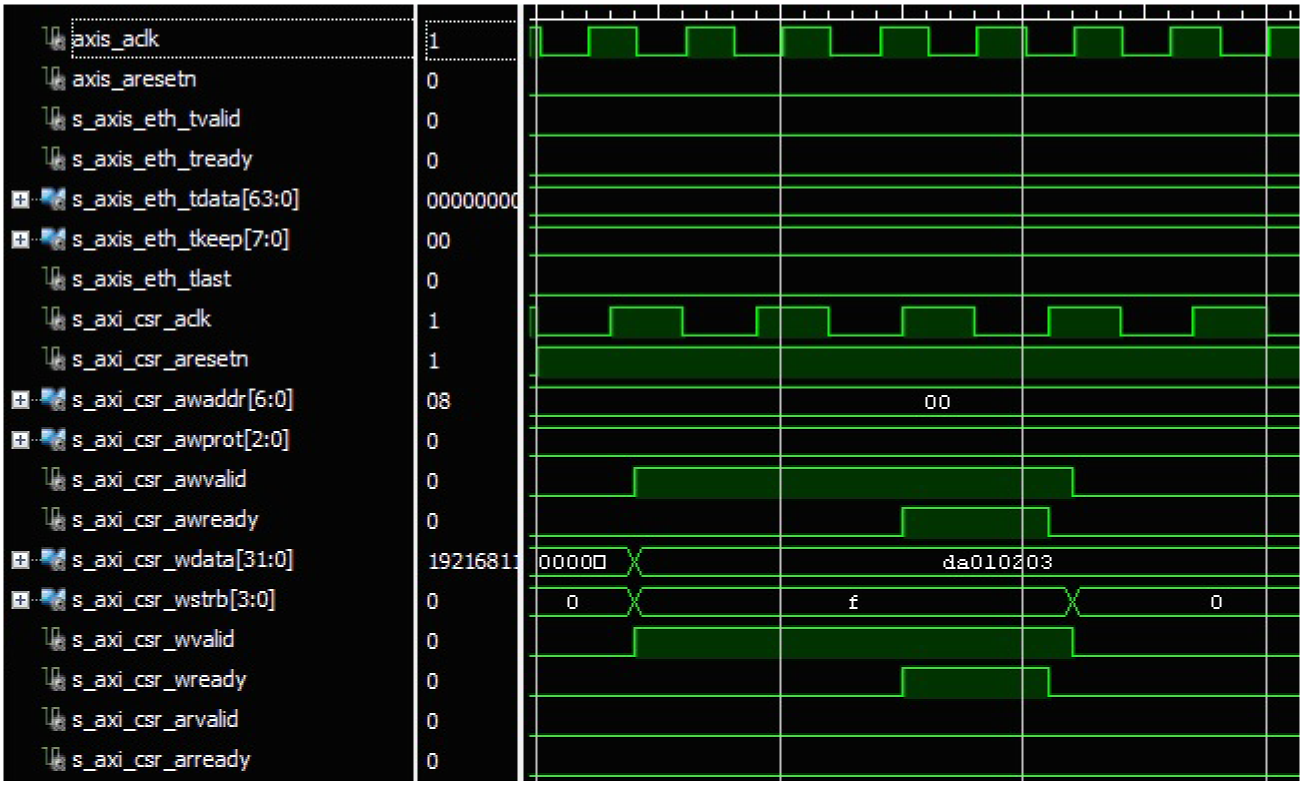
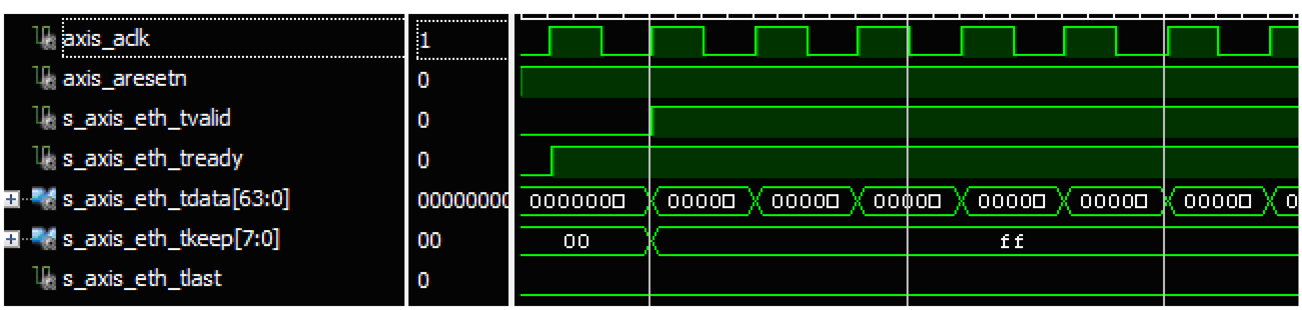


Figure 6–4: Valid Data to Core



6.6 Synthesis and Implementation

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