IP CORE MANUAL

100G Ethernet RX UDP to AXI4–Stream Core IP

px_100ge_rx_udp2axis

PENTEK

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

Copyright © 2019

Manual Part Number:807.48445 Rev: 1.0 – October 31, 2019

Manual Revision History

Date	Version	Comments
10/31/19	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

Copyright © 2019, Pentek, Inc. All Rights Reserved. Contents of this publication may not be reproduced in any form without written permission.

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

Table of Contents

IP Facts

Description

Pentek's Navigator™ 100G Ethernet RX UDP to AXI4–Stream Core accepts the UDP packet–based AXI4–Stream from the RX interface of a Xilinx 100G Ethernet MAC. The core strips–off the UDP headers, calculates the destination for the outgoing stream, discards any padding that was added to the data by the MAC, reformats the data, and generates a generic AXI4–Stream with all of the UDP characteristics removed.

This core complies with the ARM® AMBA® AXI4 Specification. This user manual defines the hardware interface, software interface, and parameterization options for the 100GE RX UDP to AXI4–Stream Core.

Features

- UDP Destination Port, EOP and SOP are retained in the "**tuser**" field of the new AXI4–Stream
- Destination port routing for the outgoing stream is calculated based on the UDP Destination Port specified in the incoming stream and a table of up to 8 different port number values
- The port address table is populated by the user via an AXI–Lite Control–Status Register (CSR) interface
- Any payload padding that was added to the data stream by the MAC will automatically be discarded
- This core is only for use with the AXI4–Stream–based 100G Ethernet MAC cores (Vivado 2019.1 and later)

Table 1–1: IP Facts Table

Core Specifics	
Supported Design Family[a]	Ultrascale+
Supported User Interfaces	AXI4–Lite and AXI4–Stream
Resources	See Table 2–1
Provided with the Core	
Design Files	VHDL
Example Design	Not Provided
Test Bench	VHDL
Constraints File	Not Provided[b]
Simulation Model	VHDL
Supported S/W Driver	N/A
Tested Design Flows	
Design Entry	Vivado® Design Suite 2019.1 or later
Simulation	Vivado VSim
Synthesis	Vivado Synthesis
Support	
Provided by Pentek fpgasupport@pentek.com	

a.For a complete list of supported devices, see the *Vivado Design Suite Release Notes*.
b.Clock constraints can be applied at the top level module of the user design.

This page is intentionally blank

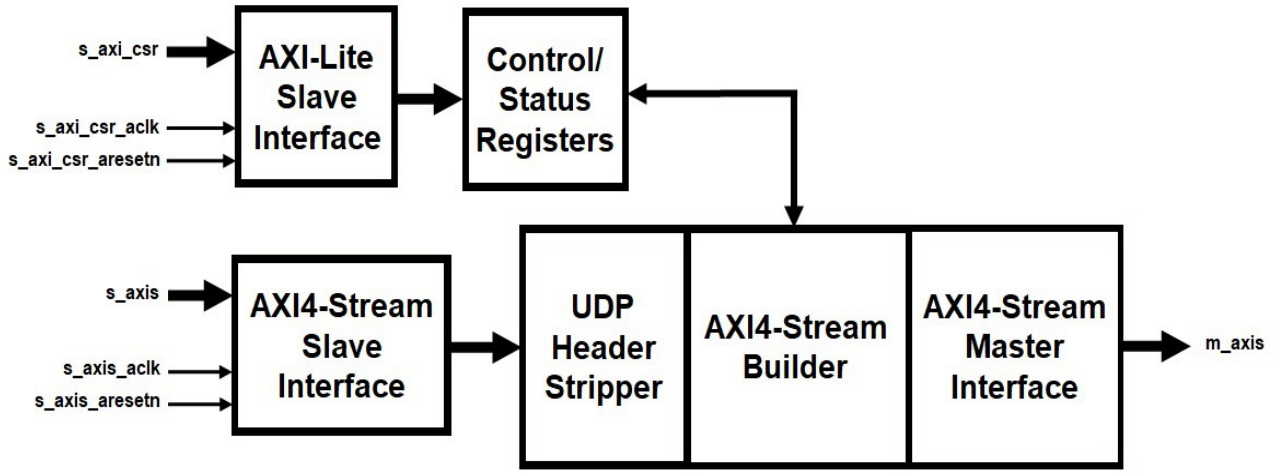
Chapter 1: Overview

1.1 Functional Description

The 100GE RX UDP to AXI4–Stream Core accepts data as a packetized AXI4–Stream from the RX interface of a Xilinx 100G Ethernet MAC. It strips–off the UDP packet headers, calculates the channel number and destination port based on a user–provided table and the port defined in the UDP header, and builds a generic AXI4–Stream with all of the UDP characteristics removed.

Figure 1–1 is a top–level block diagram of the Pentek100GE RX UDP to AXI4–Stream Core. The modules within the block diagram are explained in the later sections of this manual.

Figure 1–1: 100GE RX UDP to AXI4–Stream Core Block Diagram



- **AXI4–Stream Slave Interface:** This module contains the logic for interfacing to the Xilinx 100G Ethernet MAC's RX interface.

- **AXI4–Lite Slave Interface:** This module implements a 32–bit AXI4–Lite Control/ Status Register (CSR) Slave Interface to access the control/status registers within the core. For additional details about the AXI4–Lite Interface, refer to Section 3.1.

- **UDP Header Stripper:** This module removes the headers from the incoming UDP packets and extracts the necessary parameters from the incoming data stream that are needed to generate the Start of Packet (SOP) and End of Packet (EOP) flags as well as to determine the channel and destination ports for the outgoing stream.

1.1 Functional Description (continued)

- **AXI4–Stream Builder:** This module builds the outgoing AXI4–Stream by removing any data padding which the MAC added to the incoming stream, reformatting the data and adding a "**tuser**" field containing the Start of Packet (SOP) and End of Packet (EOP) flags, channel identification and destination port.

- **AXI4–Stream Master Interface:** This module contains the logic for the output AXI4–Stream.

1.2 Applications

The 100GE RX UDP to AXI4–Stream Core can be incorporated into a Xilinx Ultrascale+ FPGA to provide a means of converting the UDP packet–based AXI4–Stream from a Xilinx 100G Ethernet MAC's RX interface to a generic 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 and Quartz series board products. Contact Pentek for Licensing and Ordering Information (www.pentek.com).

1.5 Contacting Technical Support

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

1.6 Documentation

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

1) *Vivado Design Suite User Guide: Designing with IP*

2) *Vivado Design Suite User Guide: Programming and Debugging*

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

4) Xilinx Zynq UltraScale+ RFSoC Data Sheet, DS926

5) Ultrascale+ Devices Integrated 100G Ethernet Subsystem, PG203

Chapter 2: General Product Specifications

2.1 Standards

The 100GE RX UDP to AXI4–Stream Core has interfaces that comply with the *ARM AMBA AXI4–Lite Protocol Specification* and the *ARM AMBA AXI4–Stream Protocol Specification*.

2.2 Performance

The performance of the 100GE RX UDP to AXI4–Stream 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 100GE RX UDP to AXI4–Stream Core has two incoming clock signals. The AXI4–Stream Interface Clock (**s_axis_aclk**) has a maximum frequency of 500 MHz while the AXI4–Lite interface Clock (**s_axi_csr_aclk**) has a maximum frequency of 250 MHz. Note that 250 MHz is typically the PCI Express (PCIe®) AXI bus clock frequency.

2.3 Resource Utilization

The resource utilization of the100GE RX UDP to AXI4–Stream Core is shown in Table 2–1. Resources have been estimated for a Ultrascale+ RFSoC XCZU27 –1 speed grade device. These values were generated using the Vivado Design Suite.

Table 2–1: Resource Usage and Availability

Resource	# Used
LUTs	408
Flip–Flops	2,041

NOTE: Actual utilization may vary based on the user design in which the100GE RX UDP to AXI4–Stream Core is incorporated.

2.4 Limitations and Unsupported Features

- This core is only for use with AXI4–Stream based 100G Ethernet MAC cores (Vivado 2019.1 and later)
- Backpressure is not supported (no "**`tready`**")

2.5 Generic Parameters

The generic parameter for the 100GE RX UDP to AXI4–Stream Core is described in Table 2–2. This parameter can be set as required by the user application while customizing the core.

Table 2–2: Generic Parameters

Port/Signal Name	Type	Description
tdest_width	integer	**Destination Address Width:** This parameter defines the width (in bits) of the destination port's address bus. Allowable range is 1 – 3, the default is 3.

Chapter 3: Port Descriptions

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

- Control/Status Register (CSR) Interface
- Packetized AXI4–Stream Slave Interface
- Output AXI4–Stream Master Interface

3.1 AXI4–Lite Core Interfaces

The 100GE RX UDP to AXI4–Stream Core uses the AXI4–Lite interface to access the control/ status registers from the user design.

3.1.1 Control/Status Register (CSR) Interface

The AXI4–Lite Slave Interface can be used to access the control/status regis–ters in the 100GE RX UDP to AXI4–Stream Core. Table 3–1 defines the ports in the CSR interface. See Table 4 for a Control Register memory map and bit definitions. See the *AMBA AXI4–Lite Specification* for more details on opera–tion of the AXI4–Lite interfaces.

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

Port	Direction	Width	Description
s_axi_csr_aclk	Input	1	**CSR Clock:** 250 MHz
s_axi_csr_aresetn	Input	1	**Reset:** Active low. Asserting this input will reset the control registers to their initial state.
s_axi_csr_awaddr	Input	7	**Write Address:** Address used for write operations. It must be valid when **s_axi_csr_awvalid** is asserted and must be held until **s_axi_csr_awready** is asserted by the 100GE RX UDP to AXI4–Stream Core.
s_axi_csr_awprot	Input	3	**Protection:** The 100GE RX UDP to AXI4–Stream 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 100GE RX UDP to AXI4–Stream Core asserts **s_axi_csr_awready** when it is ready to accept the address. The **s_axi_csr_awvalid** must remain asserted until the rising clock edge after the assertion of **s_axi_csr_awready**.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axi_csr_awready	Output	1	**Write Address Ready:** This output is asserted by the 100GE RX UDP to AXI4–Stream 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 input 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 100GE RX UDP to AXI4–Stream 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:** This 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 the100GE RX UDP to AXI4–Stream Core when the write operation is complete and the Write Response is valid. It is held until **s_axi_csr_bready** is asserted by the user logic.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axi_csr_araddr	Input	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 100GE RX UDP to AXI4–Stream Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the 100GE RX UDP to AXI4–Stream 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 100GE RX UDP to AXI4–Stream 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 100GE RX UDP to AXI4–Stream 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 output 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 100GE RX UDP to AXI4–Stream 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 100GE RX UDP to AXI4–Stream Core when the read is complete and the read data is available on **s_axi_csr_rdata**. It is held until **s_axi_csr_rready** is asserted by the user logic.
s_axi_csr_rready	Input	1	**Read Data Ready:** This signal is asserted by the user logic when it is ready to accept the Read Data.

3.2 AXI4–Stream Core Interfaces

The 100GE RX UDP to AXI4–Stream Core has two AXI4–Stream Interfaces which are used to receive and to transfer data streams. The input (slave) data stream is comprised of an AXI4–Stream containing UDP packets, and the output (master) data stream is a generic AXI4–Stream.

3.2.1 Packetized AXI4–Stream Slave Interface

The incoming data stream is in the form of a packetized AXI4–Stream from the RX interface of a Xilinx 100G Ethernet MAC, and consists of UDP packets. It is associated with **`s_axis_aclk`**. Table 3–2 defines the ports in the AXI4–Stream Slave Interface. See the *AMBA AXI4 Specification* for more details on operation of the AXI4–Stream interfaces.

Table 3-2: Packetized AXI4–Stream Slave Interface Port Descriptions

Port	Direction	Width	Description
s_axis_aclk	Input	1	**AXI4–Stream Clock**
s_axis_tvalid	Input	1	**Input Data Valid:** Asserted when data is valid on **s_axis_tdata** and **s_axis_tuser**.
s_axis_tdata	Input	512	**Input Data:** This is the input data stream.
s_axis_tuser	Input	1	**Sideband Data:** This is the sideband signal from the 100GE MAC. It is equivalent to the 100GE MAC's **tx_errin** signal. 1 = indicates a bad packet 0 = indicates a good packet **NOTE:** The 100G Ethernet RX UDP to AXI4–Stream Core ignores this signal.
s_axis_tkeep	Input	64	**TKEEP Indication for the AXI4–Stream Input Data:** The assertion of bit i of this bus during a transfer indicates that dword i (in this case a dword is 8 bits) of the **s_axis_tdata** bus contains valid data.
s_axis_tlast	Input	1	**TLAST Indication AXI4–Stream Input Data:** The 100GE MAC asserts this signal in the last cycle of a data transfer to indicate the end of the packet.

3.2.2 Output AXI4–Stream Master Interface

The outgoing data stream is a generic AXI4–Stream with all of the UDP characteristics removed. This stream is also associated with **s_axis_aclk**. Table 3–3 defines the ports in the Output AXI4–Stream Master Interface. See the *AMBA AXI4 Specification* for more details on operation of the AXI4–Stream interfaces.

Table 3-3: Output AXI4–Stream Master Interface Port Descriptions

Port	Direction	Width	Description
m_axis_aresetn	Input	1	**Master AXI4–Stream Reset:** Active Low
m_axis_tvalid	Output	1	**Output Data Valid:** The 100GE RX UDP to AXI4–Stream Core asserts this output when data is valid on **m_axis_tdata** and **m_axis_tuser**.
m_axis_tdata	Output	512	**Output Data:** This is the output data stream.
m_axis_tuser	Output	24	**Sideband Data:** This is the sideband signal which contains the packet information extracted from the incoming stream. It is formatted as follows: [2:0] = Channel [3] = EOP [4] = SOP [7:5] = Reserved [23:8] = UDP Destination Port #
m_axis_tkeep	Output	64	**TKEEP Indication for the AXI4–Stream output Data:** The 100GE RX UDP to AXI4–Stream Core asserts bit i of this bus during a transfer to indicate that dword i (in this case a dword is 8 bits) of the **m_axis_tdata** bus contains valid data.
m_axis_tlast	Output	1	**TLAST Indication AXI4–Stream Output Data:** The 100GE RX UDP to AXI4–Stream Core asserts this signal in the last cycle of a data transfer to indicate that this is the last data word to be transferred.
m_axis_tdest	Output	tdest_ width	**Destination Port Address:** This is the address of the destination port for the data stream. For more information, see **Chapter 4**.

This page is intentionally blank

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the 100GE RX UDP to AXI4–Stream Core. The registers in this core comprise the table of addresses which the core will use to compare with the port defined in the UDP packet to deter–mine the destination port. The memory map is provided in Table 4–1.

NOTE: Note that if the port defined in the UDP packet does not match any of the table values, the data is routed to the default destination port.

Table 4-1: Register Space Memory Map

Register Name	Base Address (Base Address +)	Access	Description
Address values for Port 1 and Port 0	0x00	R/W	Bits [31:16] = Port 1, maps to **tdest** = "001" Bits [15:0] = Port 0, maps to **tdest** = "000"
Address values for Port 3 and Port 2	0x01	R/W	Bits [31:16] = Port 3, maps to **tdest** = "011" Bits [15:0] = Port 2, maps to **tdest** = "010"
Address values for Port 5 and Port 4	0x02	R/W	Bits [31:16] = Port 5, maps to **tdest** = "101" Bits [15:0] = Port 4, maps to **tdest** = "100"
Address values for Port 7 and Port 6	0x03	R/W	Bits [31:16] = Port 7, maps to **tdest** = "111" Bits [15:0] = Port 6, maps to **tdest** = "110" **Note:** This is the default destination port. See Table 4–5.

4.1 Address Values for Port 1 and Port 0 Register

This register contains two of the eight user–configurable address values to be compared to the destination port address specified in the UDP packet. The Address Values for Port 1 and Port 0 Register is illustrated in Figure 4–1 and described in Table 4–2.

Figure 4–1: Address Values for Port 1 and Port 0 Register

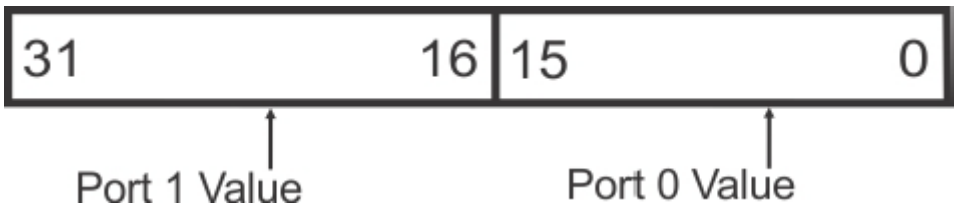


Table 4-2: Address Values for Port 1 and Port 0 Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:16	**Port 1 Value**	0x00	R/W	**Port 1 Value:** If the value in this register matches the current value in the UDP packet's destination port field, the Master AXI4–Stream's "**tdest**" field will be set to "001"
15:0	**Port 0 Value**	0x00	R/W	**Port 0 Value:** If the value in this register matches the current value in the UDP packet's destination port field, the Master AXI4–Stream's "**tdest**" field will be set to "000"

4.2 Address Values for Port 3 and Port 2 Register

This register contains two of the eight user–configurable address values to be compared to the destination port address specified in the UDP packet. The Address Values for Port 3 and Port 2 Register is illustrated in Figure 4–2 and described in Table 4–3.

Figure 4–2: Address Values for Port 3 and Port 2 Register

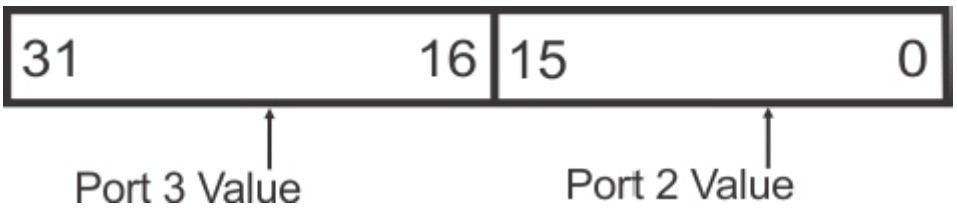


Table 4-3: Address Values for Port 3 and Port 2 Register (Base Address + 0x01)

Bits	Field Name	Default Value	Access Type	Description
31:16	**Port 3 Value**	0x00	R/W	**Port 3 Value:** If the value in this register matches the current value in the UDP packet's destination port field, the Master AXI4–Stream's "**tdest**" field will be set to "011"
15:0	**Port 2 Value**	0x00	R/W	**Port 2 Value:** If the value in this register matches the current value in the UDP packet's destination port field, the Master AXI4–Stream's "**tdest**" field will be set to "010"

4.3 Address Values for Port 5 and Port 4 Register

This register contains two of the eight user–configurable address values to be compared to the destination port address specified in the UDP packet. The Address Values for Port 5 and Port 4 Register is illustrated in Figure 4–3 and described in Table 4–4.

Figure 4–3: Address Values for Port 5 and Port 4 Register

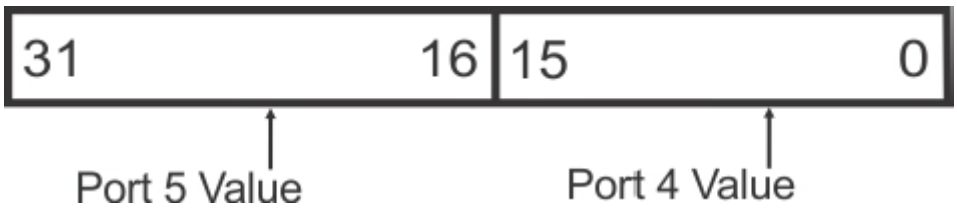


Table 4-4: Address Values for Port 5 and Port 4 Register (Base Address + 0x02)

Bits	Field Name	Default Value	Access Type	Description
31:16	**Port 5 Value**	0x00	R/W	**Port 5 Value:** If the value in this register matches the current value in the UDP packet's destination port field, the Master AXI4–Stream's "**tdest**" field will be set to "100"
15:0	**Port 4 Value**	0x00	R/W	**Port 4 Value:** If the value in this register matches the current value in the UDP packet's destination port field, the Master AXI4–Stream's "**tdest**" field will be set to "101"

4.4 Address Values for Port 7 and Port 6 Register

This register contains two of the eight user–configurable address values to be compared to the destination port address specified in the UDP packet. The Address Values for Port 7 and Port 6 Register is illustrated in Figure 4–4 and described in Table 4–5.

NOTE: Port 7 is the default destination port.

Figure 4–4: Address Values for Port 7 and Port 6 Register

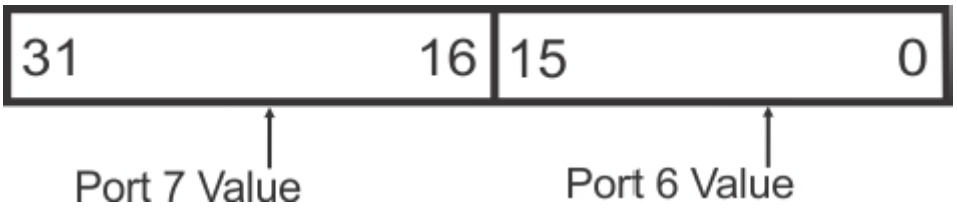


Table 4-5: Address Values for Port 7 and Port 6 Register (Base Address + 0x03)

Bits	Field Name	Default Value	Access Type	Description
31:16	**Port 7 Value**	0x00	R/W	**Port 7 Value:** If the value in this register matches the current value in the UDP packet's destination port field, OR if the current value in the UDP packet's destination port field ***does not match any*** of the values stored in any of the table registers, the Master AXI4–Stream's "**`tdest`**" field will be set to "111"
15:0	**Port 6 Value**	0x00	R/W	**Port 6 Value:** If the value in this register matches the current value in the UDP packet's destination port field, the Master AXI4–Stream's "**`tdest`**" field will be set to "110"

This page is intentionally blank

Chapter 5: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the 100GE RX UDP to AXI4−Stream Core.

5.1 General Design Guidelines

The 100GE RX UDP to AXI4−Stream Core provides the required logic to accept an incoming packetized AXI4−Stream from the RX interface of a Xilinx 100G Ethernet MAC, strip−off the UDP Ethernet headers, generate the channel number and destination port and build a generic AXI4−Stream with all of the UDP characteristics removed.

5.2 Clocking

AXI4−Stream Clock: `s_axis_aclk`

This clock is used to clock all of the ports and logic in the 100GE RX UDP to AXI4−Stream Core that are in the data path.

CSR Clock: `s_axi_csr_aclk`

This clock is used to clock the control/status register logic.

5.3 Resets

AXI4−Stream Reset: `m_axis_aresetn`

This is an active low synchronous reset associated with **`s_axis_aclk`**. When asserted, this signal will reset all of the logic within the **`s_axis_aclk`** clock domain.

AXI4−Lite CSR Reset: `s_axi_csr_aresetn`

This is an active low synchronous reset associated with **`s_axi_csr_aclk`**. When asserted, this signal will reset all of the logic within the **`s_axi_csr_aclk`** clock domain, and will set the registers to their initialization values.

5.4 Interrupts

This core does not have interrupts.

5.5 Interface Operation

- **CSR Interface:** This is the control/status register interface. It is associated with **`s_axi_csr_aclk`** and is a standard AXI4–Lite Slave interface.
- **AXI4–Stream Slave Input Interface:** This is the input AXI4–Stream from the RX interface of the Xilinx 100G Ethernet MAC core containing UDP packets to be converted to a generic AXI4–Stream.
- **AXI4–Stream Master Output Interface:** This is the output AXI4–Stream which has all of the UDP characteristics removed.

5.6 Programming Sequence

This section briefly describes the programming sequence for the 100GE RX UDP to AXI4–Stream Core:

1) Set the control registers with the desired target address values.

1) Observe the output data stream across the output ports when valid data is available at the input ports.

5.7 Timing Diagrams

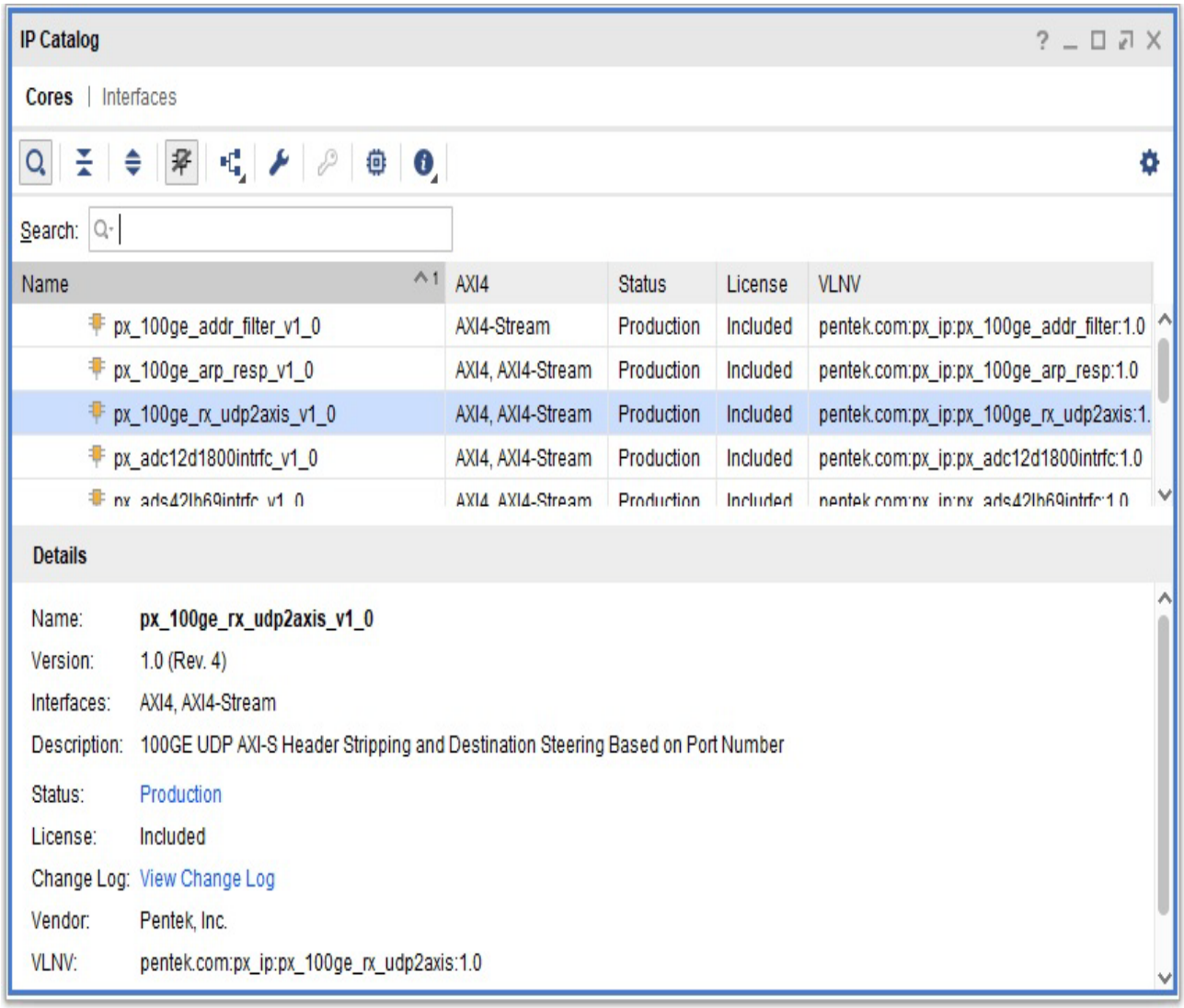
This section is not applicable for this IP core.

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the 100G Ethernet RX UDP to AXI4–Stream Core. It also includes synthesis and implementation steps that are specific to this 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_100ge_rx_udp2axis_v1_0** as shown in Figure 6–1.

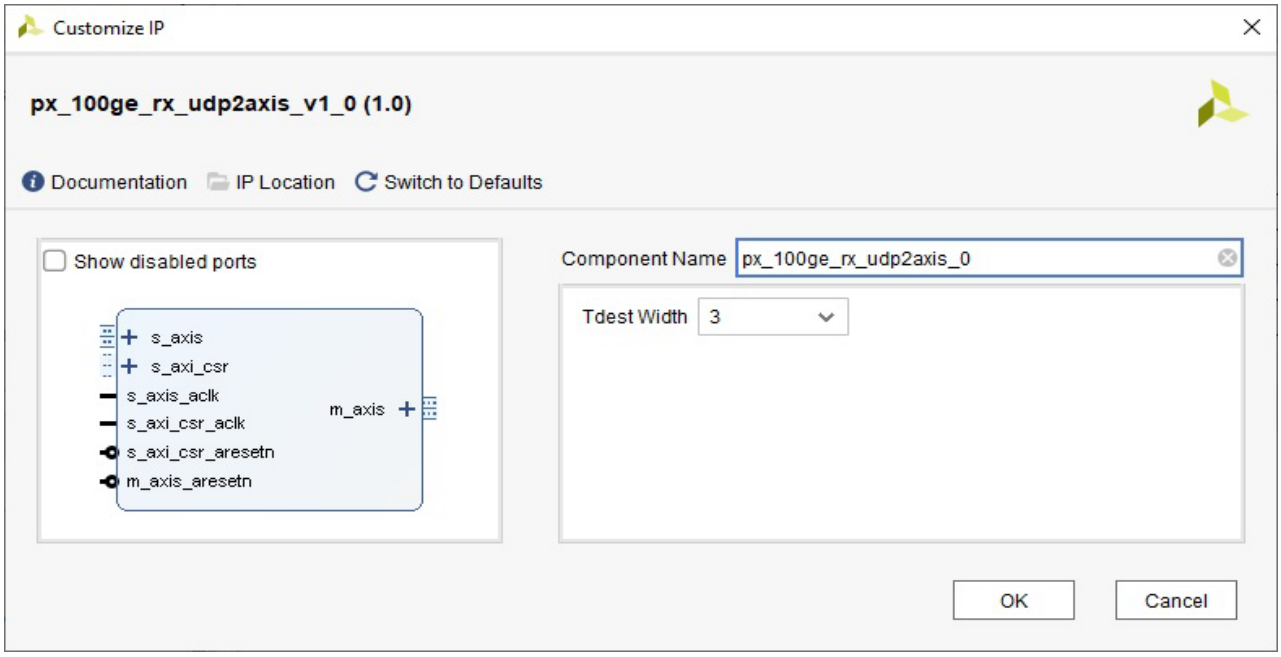
Figure 6–1: 100GE RX UDP to AXI4–Stream Core in Pentek IP Catalog



6.1 Pentek IP Catalog (continued)

When you select the **px_100ge_rx_udp2axis_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: 100GE RX UDP to AXI4–Stream Core IP Symbol



6.2 User Parameters

The user parameters of the 100GE RX UDP to AXI4–Stream Core are explained in Section 2.5 of this user manual.

6.3 Generating Output

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

6.4 Constraining the Core

This section contains information about constraining the 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 Xilinx Ultrascale+ FPGAs.

Clock Frequencies

The CSR clock (**`s_axi_csr_aclk`**) of the 100GE RX UDP to AXI4–Stream Core can take frequencies up to 250 MHz.

The AXI4–Stream clock (**`s_axis_aclk`**) can take frequencies up to 500MHz.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking and Placement

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

6.5 Simulation

This section is not applicable for this IP core.

6.6 Synthesis and Implementation

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

This page is intentionally blank