IP CORE MANUAL

AXI4–Stream PPLD to AXI4 Direct Memory Access (DMA) IP

px_dma_ppld2axi

PENTEK

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

Manual Part Number: 807.48424 Rev: 1.1 – October 22, 2018

Manual Revision History

Date	Version	Comments
10/5/18	1.0	Initial Release
10/22/18	1.1	Minor Revisions. Revised Sect 3.2.1, Table 4–6, Table 4–16, Table 4–17, Sect 6.3, and enlarged Figure 7–3.

Legal Notices

Copyright

Trademarks

Printed in the United States of America.

Table of Contents

Table of Contents

Page

Chapter 4: Register Space

Table of Contents

Page

Chapter 4: Register Space (continued)

Chapter 5: Linked List Descriptor RAM Memory Maps

Chapter 6: Designing with the Core

Table of Contents

Page

Chapter 7: Design Flow Steps

IP Facts

Description

Pentek's Navigator™ AXI4–Stream PPLD to AXI4 DMA Core is a DMA engine that facilitates high bandwidth and flexible movement of data from a PPLD–style AXI4–Stream source to an AXI4 hosted memory via DMA. This core operates using a linked list methodology to write the incoming data to the memory of the AXI4 host.

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

Features

- Fully AXI4–compliant interfaces
- Linked list DMA operation using external memory provides multiple link storage options
- Supports early DMA termination (optional) when an end of packet is reached in the input AXI4–Stream
- Interrupts available for DMA status and alarms
- Register access through AXI4–Lite CSR interface for control and status

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 PPLD to AXI4 DMA Core is a high bandwidth, reliable DMA engine used to move data from a PPLD−style AXI4−Stream source to the memory of an AXI4 host. The DMA core operates using a linked list methodology which provides the flex−ibility and ability to create complex data movement scenarios.

This core also transmits sideband metadata, which includes information about the data being moved such as timestamp, transfer length, and start and end of data acquisition markers. This core has the ability to auto−increment and loop through a number of data frames with one DMA link descriptor, using AXI4−Stream end−of−packet mark−ers to govern the DMA transfer lengths.

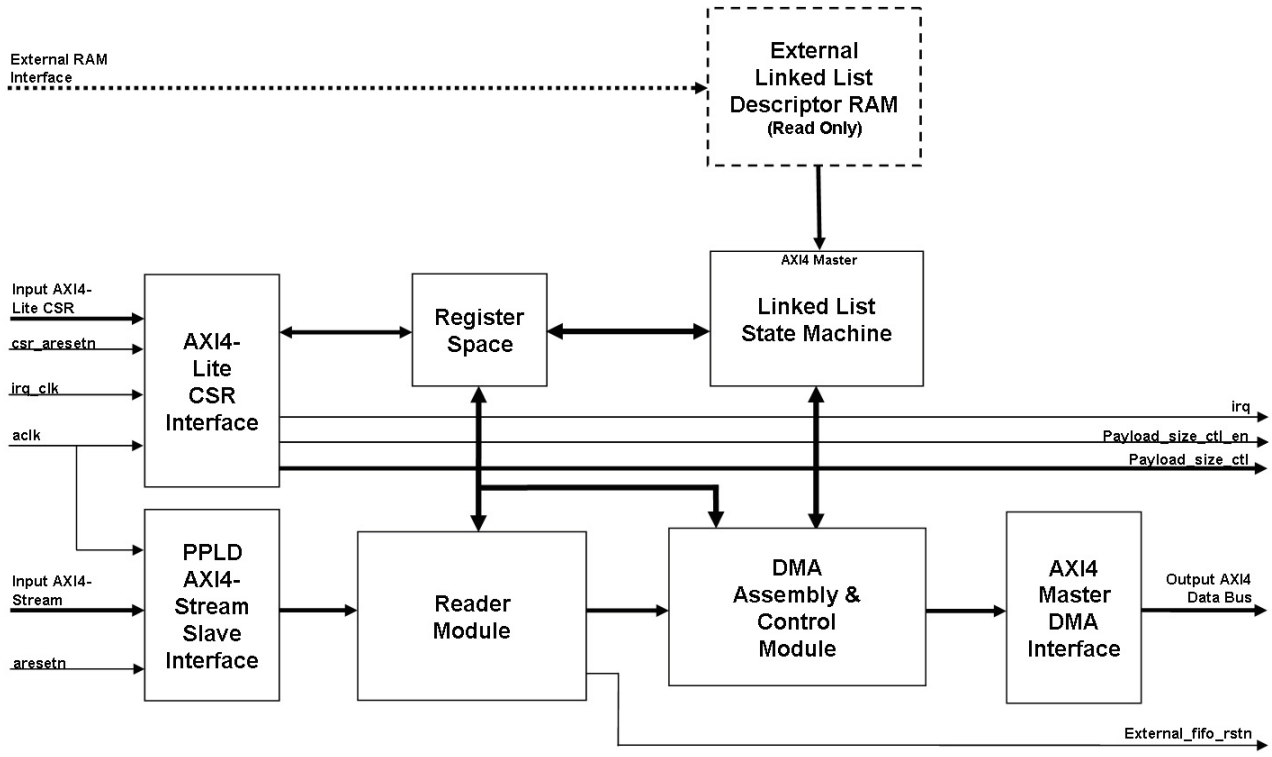
This core accepts input AXI4−Streams in the Pentek Payload (PPLD) AXI4−Stream Format (see Section 3.3 for more details) and generates AXI4 DMA write accesses which are transferred through the Output AXI4 Data Bus. This core also includes a Linked List State Machine which contains the required logic to generate the DMA write requests based on the linked list descriptors defined by the user. Link descriptors are accessed from an external RAM via a separate AXI4 interface. Note that the core's access to the Link Descriptor memory is read−only, thus link descriptors must be applied to the memory via another external access link.

An AXI4−Lite Control/Status Register (CSR) Bus accesses the control/status registers within the Register Space of the core (see Chapter 4 for a register memory map and bit definitions).

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

1.1 Functional Description (continued)

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



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

- **Register Space:** This module contains control and status registers including Interrupt Enable, Interrupt Flag, and Interrupt Status registers. Registers are accessed by the user through the AXI4–Lite CSR Interface.

- **Linked List State Machine:** This module contains the AXI4 Master Interface and control logic for retrieving link descriptors from the External Linked List Descriptor RAM. The links are used by the Linked List State Machine to generate write requests to the DMA Assembly and Control Module.

1.1 Functional Description (continued)

- **External Linked List Descriptor RAM:** This is the memory in the user design that contains the link descriptors. The AXI4 Master interface in the Linked List State Machine is used for READ–ONLY access to this memory. Because it is a READ–ONLY interface, linked list descriptors must be applied to the RAM via another (external) link. For additional details about the AXI4 Master Linked List Descriptor RAM Interface, refer to Section 3.2, AXI4 Master Core Interfaces.

- **AXI4–Stream Slave Interface:** The AXI4–Stream Slave Interface receives the packetized sample data/ timestamp/ data information AXI4–Streams in PPLD format to be passed to the target AXI4 memory host. For more details about the AXI4–Stream Interface please refer to Section 3.3, AXI4–Stream Core Interface.

- **Reader Module:** The Reader module buffers the incoming data and manages data transfers between the input and the DMA Assembly and Control Module.

- **DMA Assembly and Control Module:** This module converts the PPLD–style packets from the incoming data stream to AXI4 write commands based on the linked list descriptors provided by the Linked List Descriptor RAM.

- **AXI4 Master DMA Interface:** The AXI4 Master DMA Interface provides the DMA write commands to the target AXI4 memory host. For more details about the AXI4 Master Interface, refer to Section 3.2, AXI4 Master Core Interfaces.

1.2 Applications

The AXI4–Stream PPLD to AXI4 DMA Core can be incorporated into a Kintex Ultra–scale FPGA to perform DMA transfers of data from an AXI4–Stream PPLD source to an AXI4 memory host.

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 pro−vide supplemental material:

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

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

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

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

5) *Xilinx AXI DataMover Product Specification, PG022*

Chapter 2: General Product Specifications

2.1 Standards

The AXI4–Stream PPLD to AXI4 DMA Core has bus interfaces that comply with the *ARM AMBA AXI4–Lite Protocol Specification* and the *AMBA AXI4–Stream Protocol Specification*.

2.2 Performance

The performance of the AXI4–Stream PPLD to AXI4 DMA Core is limited by the FPGA logic speed. The values presented in this section should be used as an estimation guideline. Actual performance can vary.

2.2.1 Maximum Frequencies

The AXI4–Stream PPLD to AXI4 DMA Core has a single incoming clock signal. The main input clock (**aclk**) and the interrupt clock (**irq_clk**) have maximum frequency of 250MHz on a Kintex Ultrascale –2 speed grade FPGA. Note that 250MHz is typically the PCIe AXI bus clock frequency.

2.3 Resource Utilization

The resource utilization of the AXI4–Stream PPLD to AXI4 DMA Core is shown in Table 2–1. Resources have been estimated for the Kintex Ultrascale XCKU060 –2 speed grade device. These values were generated using the Vivado Design Suite.

Table 2–1: Resource Usage and Availability

Resource	# Used
LUTs	4181
Flip–Flops	7830
BRAMs	8.5

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

2.4 Limitations and Unsupported Features

- This core does not support DMA transfer length less than 1 dword (8 bytes).
- Descriptor addresses MUST start on 64 byte address boundaries.

2.5 Generic Parameters

The generic parameters of the AXI4–Stream PPLD to AXI4 DMA 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
channel_id	Integer	**Channel ID:** This parameter defines the destination AXI4 channel to which the data will be transferred. Allowable range is 0 – 7, the default is 0.
channel_id_width	Integer	**Channel ID Width:** This parameter defines the width (in bits) of the destination AXI4 channel ID. Allowable range is 0 – 16, the default is 5.
desc_axi_addr _width	Integer	**Descriptor RAM Address Width:** This parameter defines width (in bits) of the address bus for the descriptor RAM. Allowable range is 7 – 64, the default is 64.
data_axi_addr _width	Integer	**Target RAM DMA Address Width:** This parameter defines width (in bits) of the address bus for the destination AXI4 hosted RAM. Allowable range is 7 – 64, the default is 64.

Chapter 3: Port Descriptions

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

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

3.1 AXI4–Lite Core Interfaces

The AXI4–Stream PPLD to AXI4 DMA Core uses the Control/Status Register (CSR) interface to access the control, status and interrupt registers from the user design.

3.1.1 Control/Status Register (CSR) Interface

The CSR interface is an AXI4–Lite Slave Interface that can be used to access the control and status registers in the AXI4–Stream PPLD to AXI4 DMA 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 interfaces.

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

Port	Direction	Width	Description
aclk	Input	1	**Clock** (250 MHz)
aresetn	Input	1	**Core Reset**
s_axi_csr_aresetn	Input	1	**Reset:** Active low. This value will reset all control/status registers (only) to their initial states.
s_axi_csr_awaddr	Input	6	**Write Address:** Address used for write operations. It must be valid when **s_axi_csr_awvalid** is asserted and must be held until **s_axi_csr_awready** is asserted by the AXI4–Stream PPLD to AXI4 DMA Core.
s_axi_csr_awprot	Input	3	**Protection:** The AXI4–Stream PPLD to AXI4 DMA Core ignores these bits.

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

Port	Direction	Width	Description
s_axi_csr_awvalid	Input	1	**Write Address Valid:** This input must be asserted to indicate that a valid write address is available on **s_axi_csr_awaddr**. The AXI4–Stream PPLD to AXI4 DMA Core asserts **s_axi_csr_awready** when it is ready to accept the address. The **s_axi_csr_awvalid** must remain asserted until the rising clock edge after the assertion of **s_axi_csr_awready**.
s_axi_csr_awready	Output	1	**Write Address Ready:** This output is asserted by the AXI4–Stream PPLD to AXI4 DMA Core when it is ready to accept the write address.The address is latched when **s_axi_csr_awvalid** and **s_axi_csr_awready** are high on the same cycle.
s_axi_csr_wdata	Input	32	**Write Data:** This data will be written to the address specified by **s_axi_csr_awaddr** when **s_axi_csr_wvalid** and **s_axi_csr_wready** are both asserted. The value must be valid when **s_axi_csr_wvalid** is asserted and held until **s_axi_csr_wready** is also asserted.
s_axi_csr_wstrb	Input	4	**Write Strobes:** This signal, when asserted, indicates the number of bytes of valid data on the **s_axi_csr_wdata** signal. Each of these bits, when asserted, indicate that the corresponding byte of **s_axi_csr_wdata** contains valid data. Bit 0 corresponds to the least significant byte, and bit 3 to the most significant.
s_axi_csr_wvalid	Input	1	**Write Valid:** This signal must be asserted to indicate that the write data is valid for a write operation. The value on **s_axi_csr_wdata** is written into the register at address **s_axi_csr_awaddr** when **s_axi_csr_wready** and **s_axi_csr_wvalid** are high on the same cycle.
s_axi_csr_wready	Output	1	**Write Ready:** This signal is asserted by the AXI4–Stream PPLD to AXI4 DMA Core when it is ready to accept data. The value on **s_axi_csr_wdata** is written into the register at address **s_axi_csr_awaddr** when **s_axi_csr_wready** and **s_axi_csr_wvalid** are high on the same cycle, assuming that the address has already or simultaneously been submitted.
s_axi_csr_bresp	Output	2	**Write Response:** The AXI4–Stream PPLD to AXI4 DMA Core indicates success or failure of a write transaction through this signal, which is valid when **s_axi_csr_bvalid** is asserted; 00 = Success of normal access 01 = Success of exclusive access 10 = Slave Error 11 = Decode Error Note: For more details about this signal refer to the *AMBA AXI Specification*.

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

Port	Direction	Width	Description
s_axi_csr_bready	Input	1	**Write Response Ready:** This signal must be asserted by the user logic when it is ready to accept the Write Response.
s_axi_csr_bvalid	Output	1	**Write Response Valid:** This signal is asserted by the AXI4–Stream PPLD to AXI4 DMA 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	6	**Read Address:** Address used for read operations. It must be valid when **s_axi_csr_arvalid** is asserted and must be held until **s_axi_csr_arready** is asserted by the AXI4–Stream PPLD to AXI4 DMA Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the AXI4–Stream PPLD to AXI4 DMA Core.
s_axi_csr_arvalid	Input	1	**Read Address Valid:** This input must be asserted to indicate that a valid read address is available on **s_axi_csr_araddr**. The core asserts **s_axi_csr_arready** when it ready to accept the Read Address. This input must remain asserted until the rising clock edge after the assertion of **s_axi_csr_arready**.
s_axi_csr_arready	Output	1	**Read Address Ready:** This output is asserted by the AXI4–Stream PPLD to AXI4 DMA Core when it is ready to accept the read address. The address is latched when **s_axi_csr_arvalid** and **s_axi_csr_arready** are high on the same cycle.
s_axi_csr_rdata	Output	32	**Read Data:** This value is the data read from the address specified by the **s_axi_csr_araddr** when **s_axi_csr_arvalid** and **s_axi_csr_arready** are high on the same cycle.
s_axi_csr_rresp	Output	2	**Read Response:** The AXI4–Stream PPLD to AXI4 DMA Core indicates success or failure of a read transaction through this signal, which is valid when **s_axi_csr_rvalid** is asserted; 00 = Success of normal access 01 = Success of exclusive access 10 = Slave Error 11 = Decode Error Note: For more details about this signal refer to the *AMBA AXI Specification*.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axi_csr_rvalid	Output	1	**Read Data Valid:** This signal is asserted by the AXI4–Stream PPLD to AXI4 DMA 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_clk	Input	1	**Interrupt Clock:** This clock input provides the clock domain for the **irq** interrupt output.
irq	Output	1	**Interrupt:** This is an active high, edge–type interrupt output representing all of the enabled interrupt sources. It is synchronized to the **irq_clk** clock.

3.2 AXI4 Master Core Interfaces

3.2.1 Linked List Descriptor RAM (DESCR) Interface

Table 3–2 defines the ports in the DESCR Interface. This interface is an AXI4 Master READ–ONLY Interface that can be used to access the Linked List Descriptor RAM from the AXI4–Stream PPLD to AXI4 DMA Core. See Chapter 5 for the Descriptor RAM memory map and bit definitions. This interface is associated with **aclk**. See the *AMBA AXI4 Specification* for more details on operation of the AXI4–Lite interfaces.

Table 3-2: Linked List Descriptor RAM (DESCR) Interface Port Descriptions			
Port	**Direction**	**Width**	**Description**
m_axi_descr_arid	Output	channel_id_width	**Read Data Channel ID:** AXI4 channel ID for the Link Descriptor RAM.
m_axi_descr_araddr	Output	desc_axi_addr_width	**Read Address:** Address used for read operations. It must be valid when **m_axi_descr_arvalid** is asserted and must be held until **m_axi_descr_arready** is asserted by the External Linked List Descriptor RAM.
m_axi_descr_arprot	Output	3	**Protection:** The AXI4–Stream PPLD to AXI4 DMA Core ignores these bits.
m_axi_descr_arsize	Output	3	**Size:** The width of AXI data bus in bytes

Table 3-2: Linked List Descriptor RAM (DESCR) Interface Port Descriptions (Continued)

Port	Direction	Width	Description
m_axi_descr_arburst	Output	2	**Burst Type:** The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated: "00" = FIXED: Fixed–address burst "01" = INCR Incrementing–address burst "10" = WRAP Wrap–around, Incrementing–address burst "11" = Reserved
m_axi_descr_arlen	Output	8	**Burst Length:** The burst length gives the exact number of transfers in a burst.
m_axi_descr_arvalid	Output	1	**Read Address Valid:** This output must be asserted to indicate that a valid read address is available on **m_axi_descr_araddr**. The External Linked List Descriptor RAM asserts **m_axi_descr_arready** when it is ready to accept the Read Address. This output must remain asserted until the rising clock edge after the assertion of **m_axi_descr_arready**.
m_axi_descr_arready	Input	1	**Read Address Ready:** This input is asserted by the External Linked List Descriptor RAM when it is ready to accept the read address. The address is latched when **m_axi_descr_arvalid** and **m_axi_descr_arready** are high on the same cycle.
m_axi_descr_rdata	Input	256	**Read Data:** The value on this input is the data read from the address specified by the **m_axi_descr_araddr** when **m_axi_descr_arvalid** and **m_axi_descr_arready** are high on the same cycle.
m_axi_descr_rresp	Input	2	**Read Response:** The External Linked List Descriptor RAM indicates success or failure of a read transaction through this input, which is valid when **s_axi_descr_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.
m_axi_descr_rvalid	Input	1	**Read Data Valid:** This input is asserted by the External Linked List Descriptor RAM when the read is complete and the read data is available on **m_axi_descr_rdata**. It is held until **m_axi_descr_rready** is asserted by the AXI4–Stream PPLD to AXI4 DMA Core.
m_axi_descr_rlast	Input	1	**Read Last:** This input indicates the last transfer in a read burst.

Table 3-2: Linked List Descriptor RAM (DESCR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
m_axi_descr_rready	Output	1	**Read Data Ready:** This output is asserted by the AXI4–Stream PPLD to AXI4 DMA Core when it is ready to accept the Read Data.
m_axi_descr_rid	Input	channel_id_width	**Read ID Tag:** This input is not used by the AXI4–Stream PPLD to AXI4 DMA Core.

3.2.2 Output AXI4 Data Bus Interface

Table 3–3 defines the ports in the Output AXI4 Data Bus Interface. This interface is an AXI4 Master WRITE–ONLY Interface that is used to access the target host's RAM from the AXI4–Stream PPLD to AXI4 DMA Core. This interface is associated with aclk. See the AMBA AXI4 Specification for more details on operation of the AXI4 interfaces.

Table 3-3: Output AXI4 Data Bus Port Descriptions			
Port	**Direction**	**Width**	**Description**
m_axi_s2mm_awid	Output	channel_id_width	**Write Data Channel ID:** AXI4 channel ID for the target RAM host.
m_axi_s2mm_awaddr	Output	data_axi_addr_width	**Write Address:** Address used for write operations. It must be valid when **m_axi_s2mm_awvalid** is asserted and must be held until **m_axi_s2mm_awready** is asserted by the target RAM host.
m_axi_s2mm_awlen	Output	8	**Burst Length:** These bits define the exact burst length, i.e. number of transfers, of the AXI4 transaction.
m_axi_s2mm_awsize	Output	3	**Burst Size:** These bits define the maximum number of bytes to transfer in each data transfer as follows: "000" = 1 "001" = 2 "010" = 4 "011" = 8 "100" = 16 "101" = 32 "110" = 64 "111" = 128 **Note:** For more details about this signal refer to the AMBA AXI4 Specification.

Table 3-3: Output AXI4 Data Bus Port Descriptions (Continued)

Port	Direction	Width	Description
m_axi_s2mm _awburst	Output	2	**Burst Type:** These bits define the burst type of the transaction as follows: "00" = FIXED "01" = INCR "10" = WRAP "11" = RESERVED **Note:** For more details about this signal refer to the AMBA AXI4 Specification.
m_axi_s2mm_awprot	Output	3	**Protection:** These bits are ignored by the AXI4–Stream PPLD to AXI4 DMA Core.
m_axi_s2mm _awcache	Output	4	**Memory Type:** This is the memory type for the transaction as defined by the AXI4 spec: "0000" = Device Non–bufferable "0001" = Device Bufferable "0010" = Normal Non–cahceable, Non–bufferable "0011" = Normal Non–cahceable, Bufferable "0110" = Write–through, No–allocate "1110" = Write–through, Write–allocate "0111" = Write–back, No–allocate "1111" = Write–back, Write–allocate All values not shown are reserved. **Note:** For more details about this signal refer to the AMBA AXI4 Specification.
m_axi_s2mm_awuser	Output	4	**User Data:** These bits represent the xUSER field in the Command Interface of the Xilinx DataMover Core inside the DMA Assembly and Control Module. For more details about this signal refer to the Xilinx AXI DataMover Product Specification, PG022.
m_axi_s2mm_awvalid	Output	1	**Write Address Valid:** This output is asserted to indicate that a valid write address is available on **m_axi_s2mm_awaddr**. The **m_axi_s2mm_awvalid** will remain asserted until the rising clock edge after the assertion of **m_axi_s2mm_awready**.
m_axi_s2mm _awready	Input	1	**Write Address Ready:** This input is asserted by the target RAM host when it is ready to accept the write address. The address is latched when **m_axi_s2mmr_awvalid** and **m_axi_s2mm_awready** are high on the same cycle.
m_axi_s2mm_wdata	Output	512	**Write Data:** The value on this output is the data to be written to the address specified by the **m_axi_s2mm_awaddr** when **m_axi_s2mm_awvalid** and **m_axi_s2mm_awready** are high on the same cycle.

Table 3-3: Output AXI4 Data Bus Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
m_axi_s2mm_wstrb	Output	64	**Write Strobes:** This signal, when asserted, indicates the number of bytes of valid data on the **m_axi_s2mm_wdata** signal. Each of these bits, when asserted, indicate that the corresponding byte of the **m_axi_s2mm_wdata** bus contains valid data. Bit 0 corresponds to the least significant byte, and bit 63 to the most significant.
m_axi_s2mm_wlast	Output	1	**Write Last:** This signal indicates the last transfer in a write burst.
m_axi_s2mm_wvalid	Input	1	**Write Data Valid:** This output is asserted when the write data is available on **m_axi_s2mm_wdata** is valid. It is held until **m_axi_s2mm_wready** is asserted by the target RAM host.
m_axi_s2mm_wready	Input	1	**Write Ready:** This signal is asserted by the by the target RAM host when it is ready to accept data.
m_axi_s2mm_bresp	Input	2	**Write Response:** The target RAM host indicates success or failure of a write transaction through this input, which is valid when **m_axi_s2mm_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.
m_axi_s2mm_bvalid	Input	1	**Write Response Valid:** This signal is asserted by the target RAM host when the write operation is complete and the Write Response is valid. It is held until **m_axi_s2mm_bready** is asserted by the AXI4–Stream PPLD to AXI4 DMA Core.
m_axi_s2mm_bready	Output	1	**Write Response Ready:** This signal is asserted by the AXI4–Stream PPLD to AXI4 DMA Core when it is ready to accept the Write Response.
m_axi_s2mm_bid	Input	channel _id _width	**Response ID tag:** This signal is the ID tag of the write response.

3.3 AXI4–Stream Core Interface

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

3.3.1 PPLD Stream Data Interface

This interface is used to transfer data from the slave input port to a target AXI4 memory host. Table 3–4 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 3-4: PPLD Stream Data Interface Port Descriptions

Port	Direction	Width	Description
s_axis_ppld_tvalid	Input	1	**Input Data Valid:** This input is asserted by the user design when data is valid on **s_axis_ppld_tdata** bus. A data transfer takes place when both the **s_axis_ppld_tvalid** and **s_axis_ppld_tready** signals are High in the same cycle.
s_axis_ppld_tready	Output	1	**Output Data Ready:** This output is asserted by the AXI4–Stream PPLD to AXI4 DMA Core when it is ready to accept data from the user design.
s_axis_ppld_tdata	Input	512	**Input Data**
s_axis_ppld_tuser	Input	104	**Sideband Information:** This is the sideband information received alongside the data stream. Data is mapped as follows: [63:0] = timestamp[63:0] [64] = Start of Payload Packet [66:65] = Data Format: "00" = 8–bit "01" = 16 bit "10" = 24–bit "11" = 32–bit [67] = Data Type: '0' = Real '1' = I/Q [75:68] = channel[7:0] [77:76] = Reserved [78] = Acq. Start [79] = Acq. End (Input PPKT had tlast) [96:80] = Payload Size (bytes) [103:97] = Number Valid Bytes in cycle

Table 3-4: PPLD Stream Data Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axis_ppld_tkeep	Input	64	**Data Keep:** This input indicates the valid data sample bytes on **s_axis_ppkt_tdata**. Each bit corresponds to an 8–bit word in the **s_axis_ppkt_tdata** bus, i.e. bit 0 corresponds to the least significant word and bit 1 to the most significant. When a bit is asserted, the data corresponding to that bit is considered valid. All **tkeep** bits must be '1' contiguously until the tlast signal is asserted.
s_axis_ppld_tlast	Input	1	**Data Last:** When asserted, this input marks the last data in the current data frame.

3.4 I/O Signals

The I/O port/signal descriptions of the top level module of the AXI4–Stream to AXI4 DMA Core are provided in Table 3–5.

Table 3–5: I/O Signal Descriptions			
Port/Signal Name	**Type**	**Direction**	**Description**
payload_size_ctl	std_logic_vector [13:0]	Output	**Payload Size Control:** This output represents the current setting of the Payload Size Control Register. It indicates the number of bytes assigned to each packet.
payload_size_ctl_en	std_logic	Output	**Payload Size Control Enable:** This output indicates that the value in the Output Size Control Register (see Section 4.7) is valid and can be written to the target AXI4 memory host.
external_fifo_rstn	std_logic	Output	**External Fifo Reset:** Reset for an external FIFO. This output is synchronized with the input fifo reset in the Reader Module.

Chapter 4: Register Space

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

NOTE: There are two seperate memory map addresses for this IP Core: one for the core control registers and one for the Linked List Descriptor RAM (Chapter 5). Base addresses mentioned in this chapter are Register Space Base Addresses.

Table 4–1: Register Space Memory Map

Register Name	Address (Base Address +)	Access	Description
DMA Restart	0x00	R/W	Controls the reset of the DMA engine.
DMA Advance	0x04		Controls the start of link chain execution after a DMA restart.
DMA Abort	0x08		Controls the abort operation of the DMA.
DMA Start Link Address LS	0x0C		Controls the DMA Start Link Descriptor Address, lower–order bits
DMA Start Link Address MS	0x10		Controls the DMA Start Link Descriptor Address, higher–order bits
RQST FIFO Flush	0x14		Request for reset of the FIFO
Out Size CTL	0x18		Sets the payload size (in bytes) for the upstream PPLD device
DMA Status	0x1C	R	Indicates the DMA Status
Current Link Address LS	0x20		Indicates the lower–order bits the current link address.
Current Link Address MS	0x24		Indicates the higher–order bits the current link address.
Last Link Address LS	0x28		Indicates lower–order bits of the last link address
Last Link Address MS	0x2C		Indicates lower–order bits of the last link address
Bytes Last Transferred	0x30		Indicates the number of bytes transferred in the last executed transaction.
Interrupt Enable	0x34	R/W	Interrupt enable bits
Interrupt Status	0x38	R	Interrupt status bits
Interrupt Flag	0x3C	R/Clr	Interrupt flag bits

4.1 DMA Restart Register

This register is a control register that controls the reset of the DMA engine. The DMA Restart Register is illustrated in Figure 4–1 and described in Table 4–2. The control bit of this register must be toggled once to start execution of a link chain after power–up and DMA configuration, or after a DMA abort sequence, or after a "chain end" (when end of link chain is reached).

Figure 4–1: DMA Restart Register

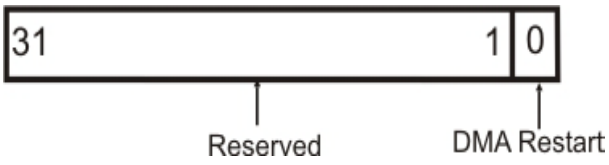


Table 4–2: DMA Restart Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	DMA Restart	0	R/W	**DMA Restart:** When this bit is toggled '1' then '0', the DMA is reset.

4.2 DMA Advance Register

The DMA Advance Register is used to advance the DMA linked list. The DMA Advance Register control bit must be toggled to start execution of the first DMA link descriptor in a chain regardless of whether that link is set to auto or manual mode of operation. It is then used to start execution of any link in the chain that is set to manual start mode. A link that is set to manual start mode will indicate it is waiting to start with the "waiting for advance" status bit of the DMA Status register (see Table 4–9). This register is illustrated in Table 4–2 and described in Table 4–3.

Figure 4–2: DMA Advance Register

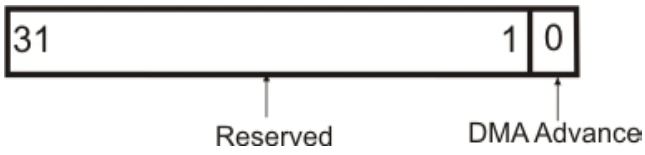


Table 4–3: DMA Advance Register (Base Address + 0x04)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	DMA Advance	0	R/W	**DMA Advance:** When toggled '1' then '0', the Linked List State Machine starts the execution of the first link descriptor after reset. The address of the first link descriptor in the linked list RAM is given by the DMA Start Link Address Control Register (see **Section 4.4**). This bit must also be toggled to start any link in a chain if that link descriptor is set to manual execution mode.

4.3 DMA Abort Register

The DMA Abort Register is used to control the generation of a DMA engine abort sequence. When an abort sequence is enabled, the DMA packets that have already been assembled by the core will be allowed to be transmitted while new DMA activity will be inhibited. When all assembled packets have been transmitted, a DMA reset will automatically be generated. After a DMA abort, the DMA Restart Register control bit must be toggled to start execution of a new link chain. This register is illustrated in Figure 4–3 and described in Table 4–4.

Figure 4–3: DMA Abort Register

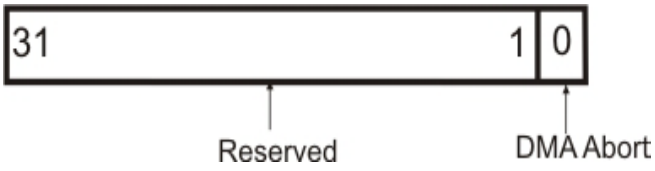


Table 4–4: DMA Abort Register (Base Address + 0x08)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	DMA Abort	0	R/W	**DMA Abort:** When toggled '1' then '0' a DMA abort sequence will commence.

4.4 DMA Start Link Address, Least Significant Bits Register

This DMA Start Link Address Register is used to control the address of the Linked list Descriptor RAM from where the DMA engine starts execution after reset. This register holds the 32 LSB's of the address, and is illustrated in Figure 4–4 and described in Table 4–5.

Figure 4–4: DMA Start Link Address, Least Significant Bits Register

Table 4–5: DMA Start Link Address, Least Significant Bits Register (Base Address + 0x0C)

Bits	Field Name	Default Value	Access Type	Description
31:0	DMA Start Link Address LS	0x0000	R/W	**DMA Start Link Descriptor – Least Significant:** These are the least–significant bits of the start address of the Link List Descriptor RAM whose link definition is to be executed by the DMA after a reset.

4.5 DMA Start Link Address, Most Significant Bits Register

This DMA Start Link Address Register is used to control the address of the Linked list Descriptor RAM from where the DMA engine starts execution after reset. This register holds the 32 MSB's of the address, and is illustrated in Figure 4–5 and described in Table 4–6.

Figure 4–5: DMA Start Link Address, Most Significant Bits Register

Table 4–6: DMA Start Link Address, Most Significant Bits Register (Base Address + 0x10)

Bits	Field Name	Default Value	Access Type	Description
31:0	DMA Start Link Address LS	0x0000	R/W	**DMA Start Link Descriptor – Least Significant:** These are the most–significant bits of the start address of the Link List Descriptor RAM whose link definition is to be executed by the DMA after a reset.

4.6 RQST FIFO Flush Register

The FIFO Flush Request Register is used to issue a FIFO flush request to the core. The core will respond by resetting the input data FIFO inside the Read Module when the current transaction is completed. This register is illustrated in Figure 4–6 and described in Table 4–7.

Figure 4–6: RQST FIFO Flush Register

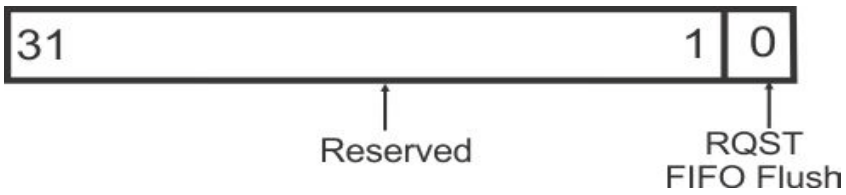


Table 4–7: RQST FIFO Flush Register (Base Address + 0x14)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	RQST FIFO Flush	0	R/W	**FIFO Flush Request:** When this bit is set, the core will issue a reset of the input data fifo once the current transaction is complete.

4.7 Output Size Control Register

The Output Size Control Register sets the Payload Size Control output (**payload_size_ctl**) for the target AXI4 memory host. This register is illustrated in Figure 4–6 and described in Table 4–7.

Figure 4–7: Output Size Control Register

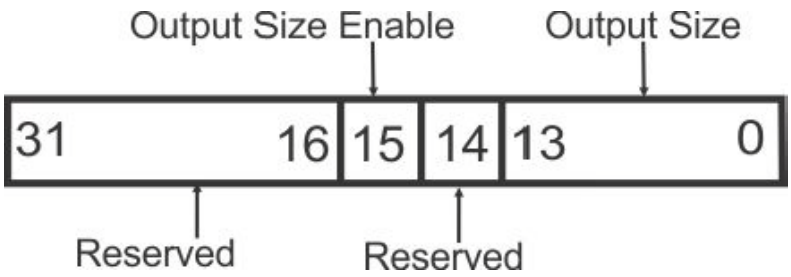


Table 4–8: Output Size Control Register (Base Address + 0x18)

Bits	Field Name	Default Value	Access Type	Description
31:16	Reserved	N/A	N/A	**Reserved**
15	Output SIze Enable	0	R/W	**Output Size Enable:** When HIGH this bit indicates that the Output Size in the lower bits is valid.
14	Reserved	–	–	**Reserved**
13:0	Output Size	00000000000000	R/W	**Output Size:** These bits provide the packet size setting for the target AXI4 memory host.

4.8 DMA Status Register

The DMA Status Register indicates the status of the DMA engine. This register is illustrated in Figure 4–8 and described in Table 4–9.

Figure 4–8: DMA Status Register

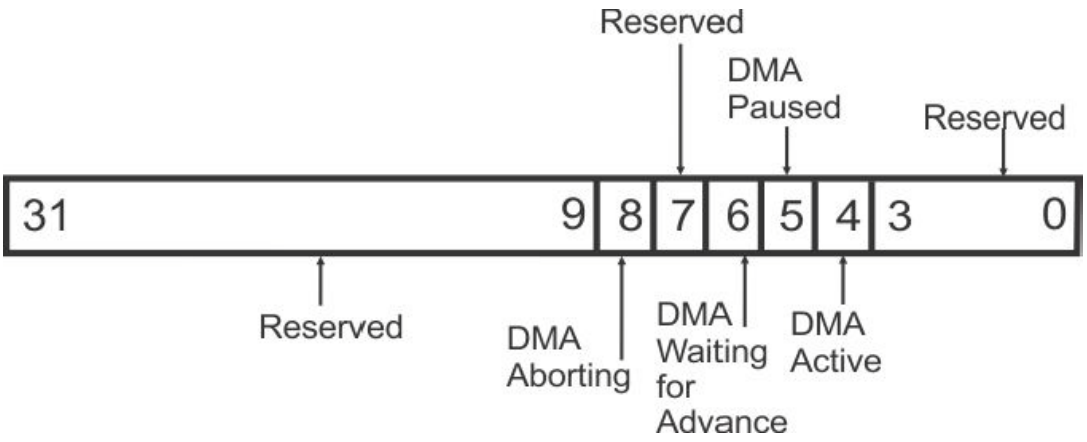


Table 4-9: DMA Status Register (Base Address + 0x1C)

Bits	Field Name	Default Value	Access Type	Description
31:9	Reserved	–	–	**Reserved**
8	DMA Aborting	0	RO	**DMA Aborting:** When this bit '0', the DMA is not in an abort cycle. When this bit '1', a DMA abort is in progress. 0 = Normal 1 = DMA abort
7	Reserved	N/A	N/A	**Reserved**
6	DMA Waiting For Advance	0	RO	**DMA Waiting for Advance:** This bit indicates that the DMA is waiting for the DMA advance bit of the DMA Advance control register to be toggled. See **Section 4.2**. 0 = Not waiting 1 = Waiting for advance
5	DMA Paused			**DMA Paused:** This bit indicates that the DMA is paused. 0 = DMA running 1 = DMA paused
4	DMA Active			**DMA Active:** This bit indicates that a DMA transaction is in progress.
3:0	Reserved	–	–	**Reserved**

4.9 Current Link Address, Least Significant Bits Register

This register is a status register which contains the LSB's of the address provided by the current link descriptor. This register is illustrated in Figure 4–9, below, and described in Table 4–10.

Figure 4–9: Current Link Address, Least Significant Bits Register

Table 4–10: Current Link Address, Least Significant Bits Register (Base Address + 0x20)

Bits	Field Name	Default Value	Access Type	Description
31:0	Current Link Address LS	0x00000000	RO	**Current Link Address LS:** These bits are the LSB's of the link address provided by the current link descriptor.

4.10 Current Link Address, Most Significant Bits Register

This register is a status register which contains the MSB's of the address provided by the current link descriptor. This register is illustrated in Figure 4–10, below, and described in Table 4–11.

Figure 4–10: Current Link Address, Most Significant Bits Register

Table 4–11: Current Link Address, Most Significant Bits Register (Base Address + 0x24)

Bits	Field Name	Default Value	Access Type	Description
31:0	Current Link Address MS	0x00000000	RO	**Current Link Address MS:** These bits are the MSB's of the link address provided by the current link descriptor.

4.11 Last Link Address, Least Significant Bits Register

The Last Link Address LS Register contains the LSB's of the last address available in the link descriptor list. This register is illustrated in Figure 4–11 and described in Table 4–12.

Figure 4–11: Last Link Address, Least Significant Bits Register

Table 4–12: Last Link Address, Least Significant Bits Register (Base Address + 0x28)

Bits	Field Name	Default Value	Access Type	Description
31:0	Last Link Address LS	0x0000 0000	RO	**Last Link Address LS:** These bits are the LSB's of the last link address available in the link descriptor list.

4.12 Last Link Address, Most Significant Bits Register

The Last Link Address MS Register contains the MSB's of the last address available in the link descriptor list. This register is illustrated in Figure 4–12 and described in Table 4–13.

Figure 4–12: Last Link Address, Most Significant Bits Register

Table 4–13: Last Link Address, Most Significant Bits Register (Base Address + 0x2C)

Bits	Field Name	Default Value	Access Type	Description
31:0	Last Link Address MS	0x0000 0000	RO	**Last Link Address MS:** These bits are the MSB's of the last link address available in the link descriptor list.

4.13 Bytes Last Transferred Register

The Bytes Last Transferred Register indicates the number of bytes transferred in the last transaction. This register is illustrated in Figure 4–13 and described in Table 4–14.

Figure 4–13: Bytes Last Transferred Register

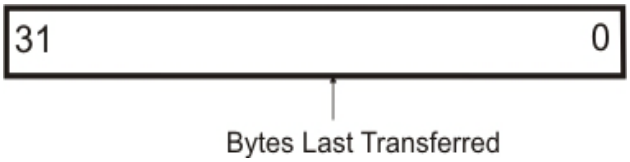


Table 4–14: Bytes Last Transferred Register (Base Address + 0x30)

Bits	Field Name	Default Value	Access Type	Description
31:0	Bytes Last Transferred	0x00000000	RO	**Bytes Last Transferred:** This register indicates the total number of bytes transferred in the last transaction.

4.14 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 4.15). This register is illustrated in Figure 4–14 with the bits described in Table 4–15.

Figure 4–14: Interrupt Enable Register

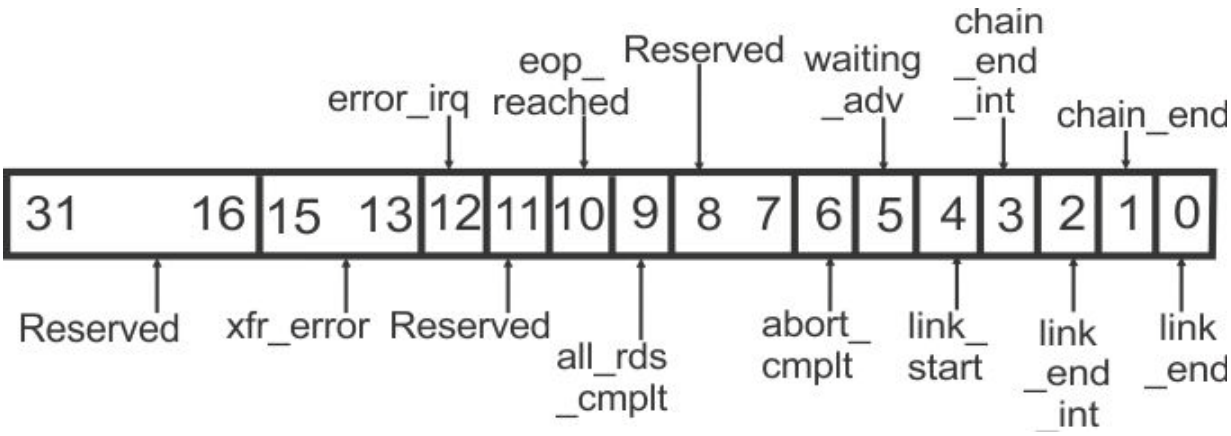


Table 4–15: Interrupt Enable Register (Base Address + 0x34)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:16	Reserved	N/A	N/A	**Reserved**
15:13	xfr_error	000	R/W	**Transfer Error:** These bits enable/disable the **xfr_error** interrupt sources from the Xilinx DataMover Core inside the DMA Assembly and Control Module. For more details about this core refer to the Xilinx AXI DataMover Product Specification, PG022. Source mapping for these bits is as follows: **15: Slave Error:** Indicates the DataMover encountered a slave reported error condition for the associated command. This is received by the response inputs from the AXI4 interface. **14: Decode Error:** Indicates the DataMover encountered an address decode error condition for the associated command. This is received by the response inputs from the AXI4 interface and indicates an address decode timeout occurred on an address generated by the DataMover element while executing the corresponding command. **13: Internal Error:** Indicates the DataMover encountered an internal error condition for the associated command. A BTT (Bytes to Transfer) value of 0 (zero) in the command word can cause this assertion. This error is also flagged by the S2MM function when the Indeterminate BTT mode is not enabled and the number of bytes received on the AXIS is not the same as what was programmed in the BTT field of the command. In other words, this error is flagged if **TLAST** comes early or late or never. For all of the above, '0' = Disable Interrupt '1' = Enable Interrupt
12	error_irq	0	R/W	**Error Interrupt:** This bit enables/disables the **error_irq** interrupt source. When enabled, the error_irq interrupt indicates that an error was detected in the last transfer. '0' = Disable interrupt '1' = Enable interrupt
11	Reserved	–	–	**Reserved**

Table 4–15: Interrupt Enable Register (Base Address + 0x34) (Continued)

Bits	Field Name	Default Value	Access Type	Description
10	eop_reached	0	R/W	**End of Packet reached:** This bit enables/ disables the end of packet reached interrupt source. The end of packet reached interrupt source indicates that the end of packet of the input data stream has been reached. '0' = Disable interrupt '1' = Enable interrupt
9	all_rds_cmplt	0	R/W	**All Reads Complete:** This bit enables/ disables the all reads complete interrupt source. The all reads complete interrupt source indicates that all reads from the FIFO that are required for the Buffer Read Request have been completed. '0' = Disable interrupt '1' = Enable interrupt
8:7	Reserved	N/A	N/A	**Reserved**
6	abort_cmplt	0	R/W	**Abort Complete:** This bit enables/ disables the DMA abort complete interrupt source. 0 = Disable interrupt 1 = Enable interrupt
5	waiting_adv	0	R/W	**Waiting for DMA Advance:** This bit enables/ disables the waiting for DMA advance interrupt source. The waiting for DMA advance interrupt source indicates that the DMA is waiting for the DMA advance signal, after a reset. 0 = Disable interrupt 1 = Enable interrupt
4	link_start	0	R/W	**Link Start:** This bit enables/ disables the link start interrupt source. The link start interrupt source indicates the start of execution of a link descriptor by the DMA after reset. 0 = Disable interrupt 1 = Enable interrupt
3	chain_end_int	0	R/W	**Chain End Interrupt:** This bit enables/ disables the chain end interrupt source. The chain end interrupt source indicates that the end of link descriptor chain has been reached. This interrupt source will be set only when the Chain End Interrupt Enable bit of the Link Descriptor Control Word is set to '1' (see **Table 5–3**). 0 = Disable interrupt 1 = Enable interrupt

Table 4–15: Interrupt Enable Register (Base Address + 0x34) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
2	link_end_int	0	R/W	**Link End Interrupt:** This bit enables/ disables the link end interrupt source. The link end interrupt source indicates that the current link execution is complete. This interrupt source will be set only when the Link End Interrupt Enable bit of the Link Descriptor Control Word is set to '1' (see Table 5–3). 0 = Disable interrupt 1 = Enable interrupt
1	chain_end	0	R/W	**Chain End:** Reserved for test purposes.
0	link_end	0	R/W	**Link End:** Reserved for test purposes.

4.15 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 4–14 with the bits described in Table 4–15.

Figure 4–15: Interrupt Status Register

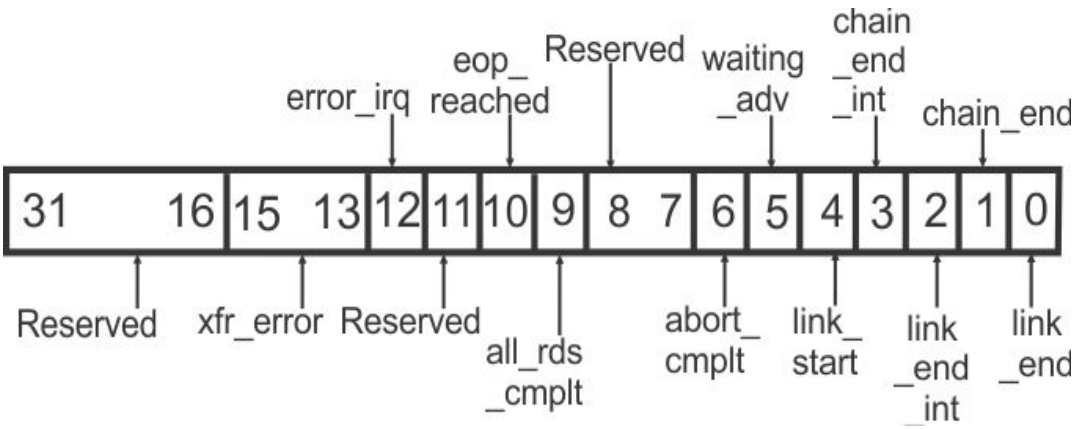


Table 4–16: Interrupt Status Register (Base Address + 0x38)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:16	Reserved	N/A	N/A	**Reserved**
15:13	xfr_error	000	RO	**Transfer Error:** These bits indicate the status of the xfr_error interrupts from the Xilinx DataMover Core inside the DMA Assembly and Control Module. For more details about this core refer to the Xilinx AXI DataMover Product Specification, PG022. Source mapping for these bits is as follows: **15: Slave Error:** Indicates the DataMover encountered a slave reported error condition for the associated command. This is received by the response inputs from the AXI4 interface. **14: Decode Error:** Indicates the DataMover encountered an address decode error condition for the associated command. This is received by the response inputs from the AXI4 interface and indicates an address decode timeout occurred on an address generated by the DataMover element while executing the corresponding command. **13: Internal Error:** Indicates the DataMover encountered an internal error condition for the associated command. A BTT (Bytes to Transfer) value of 0 (zero) in the command word can cause this assertion. This error is also flagged by the S2MM function when the Indeterminate BTT mode is not enabled and the number of bytes received on the AXIS is not the same as what was programmed in the BTT field of the command. In other words, this error is flagged if TLAST comes early or late or never. For all of the above, '0' = No interrupt '1' = Interrupt condition asserted
12	error_irq	0	RO	**Error Interrupt:** This bit indicates the status of the error_irq interrupt source. When enabled, the **error_irq** interrupt indicates that an error was detected in the last transfer. '0' = No interrupt '1' = Interrupt condition asserted
11	Reserved	–	–	**Reserved**

Table 4–16: Interrupt Status Register (Base Address + 0x38) (Continued)

Bits	Field Name	Default Value	Access Type	Description
10	eop_reached	0	RO	**End of Packet reached:** This bit indicates the status of the end of packet reached interrupt source. The end of packet reached interrupt source indicates that the end of packet of the input data stream has been reached. '0' = No interrupt '1' = Interrupt condition asserted
9	all_rds_cmplt	0	RO	**All Reads Complete:** This bit indicates the status of the all reads complete interrupt source. The all reads complete interrupt source indicates that all reads from the FIFO that are required for the Buffer Read Request have been completed. '0' = No interrupt '1' = Interrupt condition asserted
8:7	Reserved	N/A	N/A	**Reserved**
6	abort_cmplt	0	RO	**Abort Complete:** This bit indicates the status of the DMA abort complete interrupt source. '0' = No interrupt '1' = Interrupt condition asserted
5	waiting_adv	0	RO	**Waiting for DMA Advance:** This bit indicates the status of the waiting for DMA advance interrupt source. The waiting for DMA advance interrupt source indicates that the DMA is waiting for the DMA advance signal, after a reset. '0' = No interrupt '1' = Interrupt condition asserted
4	link_start	0	RO	**Link Start:** This bit indicates the status of the link start interrupt source. The link start interrupt source indicates the start of execution of a link descriptor by the DMA after reset. '0' = No interrupt '1' = Interrupt condition asserted
3	chain_end_int	0	RO	**Chain End Interrupt:** This bit indicates the status of the chain end interrupt source. The chain end interrupt source indicates that the end of link descriptor chain has been reached. This interrupt source will be set only when the Chain End Interrupt Enable bit of the Link Descriptor Control Word is set to '1' (see **Table 5–3**). '0' = No interrupt '1' = Interrupt condition asserted

Table 4–16: Interrupt Status Register (Base Address + 0x38) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
2	link_end_int	0	RO	**Link End Interrupt:** This bit indicates the status of the link end interrupt source. The link end interrupt source indicates that the current link execution is complete. This interrupt source will be set only when the Link End Interrupt Enable bit of the Link Descriptor Control Word is set to '1' (see **Table 5–3**). '0' = No interrupt '1' = Interrupt condition asserted
1	chain_end	0	RO	**Chain End:** Reserved for test purposes.
0	link_end	0	RO	**Link End:** Reserved for test purposes.

4.16 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 4–16 with the bits described in Table 4–17.

Figure 4–16: Interrupt Flag Register

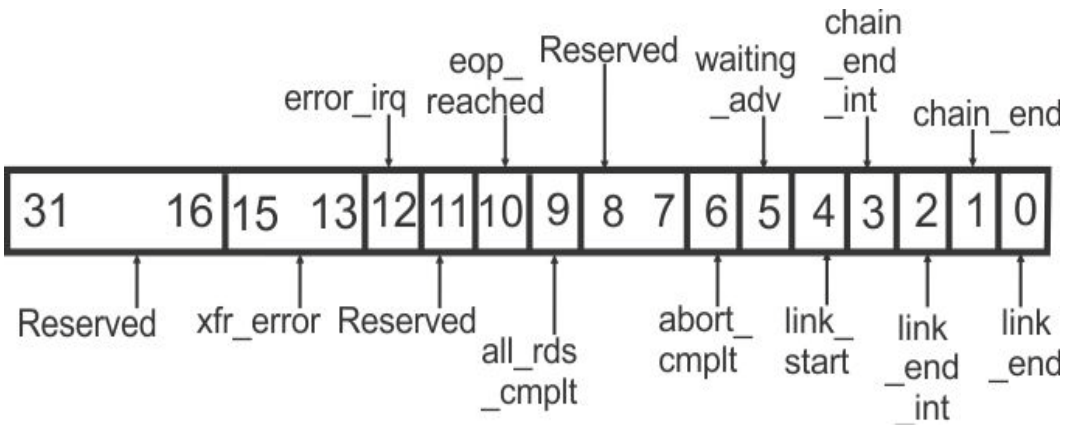


Table 4–17: Interrupt Flag Register (Base Address + 0x3C)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:16	Reserved	N/A	N/A	**Reserved**
15:13	xfr_error	000	R/CLR	**Transfer Error:** These bits indicate the status of the xfr_error interrupt flags from the Xilinx DataMover Core inside the DMA Assembly and Control Module. For more details about this core refer to the Xilinx AXI DataMover Product Specification, PG022. Source mapping for these bits is as follows: **15: Slave Error:** Indicates the DataMover encountered a slave reported error condition for the associated command. This is received by the response inputs from the AXI4 interface. **14: Decode Error: I**ndicates the DataMover encountered an address decode error condition for the associated command. This is received by the response inputs from the AXI4 interface and indicates an address decode timeout occurred on an address generated by the DataMover element while executing the corresponding command. **13: Internal Error:** Indicates the DataMover encountered an internal error condition for the associated command. A BTT (Bytes to Transfer) value of 0 (zero) in the command word can cause this assertion. This error is also flagged by the S2MM function when the Indeterminate BTT mode is not enabled and the number of bytes received on the AXIS is not the same as what was programmed in the BTT field of the command. In other words, this error is flagged if TLAST comes early or late or never. For all of the above, '0' = No interrupt '1' = Interrupt latched **Clear:** '1' = Clear latch
12	error_irq	0	R/CLR	**Error Interrupt:** This bit indicates the status of the error_irq interrupt flag. When enabled, the **`error_irq`** interrupt indicates that an error was detected in the last transfer. '0' = No interrupt '1' = Interrupt latched **Clear:** '1' = Clear latch
11	Reserved	–	–	**Reserved**

Table 4–17: Interrupt Flag Register (Base Address + 0x3C) (Continued)

Bits	Field Name	Default Value	Access Type	Description
10	eop_reached	0	R/CLR	**End of Packet reached:** This bit indicates the status of the end of packet reached interrupt flag. The end of packet reached interrupt source indicates that the end of packet of the input data stream has been reached. '0' = No interrupt '1' = Interrupt latched **Clear:** '1' = Clear latch
9	all_rds_cmplt	0	R/CLR	**All Reads Complete:** This bit indicates the status of the all reads complete interrupt flag. The all reads complete interrupt source indicates that all reads from the FIFO that are required for the Buffer Read Request have been completed. '0' = No interrupt '1' = Interrupt latched **Clear:** '1' = Clear latch
8:7	Reserved	N/A	N/A	**Reserved**
6	abort_cmplt	0	R/CLR	**Abort Complete:** This bit indicates the status of the DMA abort complete interrupt flag. '0' = No interrupt '1' = Interrupt latched **Clear:** '1' = Clear latch
5	waiting_adv	0	R/CLR	**Waiting for DMA Advance:** This bit indicates the status of the waiting for DMA advance interrupt flag. The waiting for DMA advance interrupt source indicates that the DMA is waiting for the DMA advance signal, after a reset. '0' = No interrupt '1' = Interrupt latched **Clear:** '1' = Clear latch
4	link_start	0	R/CLR	**Link Start:** This bit indicates the status of the link start interrupt flag. The link start interrupt source indicates the start of execution of a link descriptor by the DMA after reset. '0' = No interrupt '1' = Interrupt latched **Clear:** '1' = Clear latch
3	chain_end_int	0	R/CLR	**Chain End Interrupt:** This bit indicates the status of the chain end interrupt flag. The chain end interrupt source indicates that the end of link descriptor chain has been reached. This interrupt source will be set only when the Chain End Interrupt Enable bit of the Link Descriptor Control Word is set to '1' (see **Table 5–3**). '0' = No interrupt '1' = Interrupt latched Clear: '1' = Clear latch

Table 4–17: Interrupt Flag Register (Base Address + 0x3C) (Continued)

Bits	Field Name	Default Value	Access Type	Description
2	link_end_int	0	R/CLR	**Link End Interrupt:** This bit indicates the status of the link end interrupt flag. The link end interrupt source indicates that the current link execution is complete. This interrupt source will be set only when the Link End Interrupt Enable bit of the Link Descriptor Control Word is set to '1' (see **Table 5–3**). '0' = No interrupt '1' = Interrupt latched Clear: '1' = Clear latch
1	chain_end	0	R/CLR	**Chain End:** Reserved for test purposes.
0	link_end	0	R/CLR	**Link End:** Reserved for test purposes.

Chapter 5: Linked List Descriptor RAM Memory Maps

Table 5–1 defines the AXI4–Stream PPLD to AXI4 DMA Core's Linked List Descriptor RAM memory maps.

NOTE: There are two seperate memory map addresses for this IP Core: one for the core control registers (Chapter 4) and one for the Linked List Descriptor RAM

Table 5–1: Linked List Descriptor RAM Memory Map

Descriptor	Address (Descriptor RAM Base Address +)	Access	Description
Descriptor 0	0x0000	R/W	Control Word [31:0]
	0x0004		Number of Bytes to transfer [31:0]
	0x0008		AXI4 Memory Destination Address [31:0]
	0x000C		AXI4 Memory Destination Address [63:32]
	0x0010		AXI4 Memory Destination Metadata Address [31:0]
	0x0014		AXI4 Memory Destination Metadata Address [63:32]
	0x0018		Next Link Address [31:0]
	0x001C		Next Link Address [63:32]
Descriptor 1	0x0020 – 0x003C		
Descriptor 2	0x0040 – 0x005C		
.....			
Descriptor Y **Where Y = the last address as defined by desc_ram_addr_width**	Y – 0x1F	R/W	Control Word [31:0]
	Y – 0x1B		Number of Bytes to Transfer [31:0]
	Y – 0x17		AXI4 Memory Destination Address [31:0]
	Y – 0x13		AXI4 Memory Destination Address [63:32]
	Y – 0x0F		AXI4 Memory Destination Metadata Address [31:0]
	Y – 0x0B		AXI4 Memory Destination Metadata Address [63:32]
	Y – 0x07		Next Link Address [31:0]
	Y – 0x03		Next Link Address [63:32]

Table 5–2 defines the fields in each link descriptor stored in the Linked List Descriptor RAM.

Table 5-2: Link Descriptor Field Definitions

Address (Descriptor RAM Base address+)	Bits	Field Name	Access Type	Description
0x0000	**31:0**	control_word	R/W	**Control Word:** These bits define the control bits for the link descriptor. The individual bits of the control word are explained in the **Table 5–3**.
0x0004	**63:32**	bytes_to_transfer	R/W	**Bytes to Transfer:** These bits indicate the number of bytes of data to be written to the AXI4 Memory.
0x0008	**95:64**	dest_addr	R/W	**AXI4 Memory Destination Address:** These bits hold the value of the AXI4 Memory address where the AXI4 Memory write request data is to be written.
0x000C	**127:96**			
0x0010	**159:128**	dest_meta_ data_addr	R/W	**AXI4 Memory Destination Metadata Address:** These bits hold the value of the AXI4 Memory address where the AXI4 Memory write request metadata is to be written.
0x0014	**191:160**			
0x0018	**223:192**	next_link_addr	R/W	**Next Link Address:** These bits indicate the link descriptor RAM address of the next link descriptor to be executed.
0x001C	**255:224**			

Table 5–3 defines the bits in the Control Word of the link descriptor stored in the Linked List Descriptor RAM.

Table 5-3: Link Descriptor Control Word Bit Definitions

Bits	Field Name	Default Value	Access Type	Description
31:12	Reserved	–	–	**Reserved**
11	Output Meta Data	0	R/W	**Output Meta Data:** This bit is used to enable (or disable) writing of metadata packets into the AXI4 Memory.Metadata packet bit definitions can be found in **Table 5–4**. '0' = Disable '1' = Enable
10	Chain End	0	R/W	**Chain End:** This bit indicates whether the link descriptor executed is the end of the link chain. 0 = This link is not the end of the chain 1 = This link is the end of the chain
9	Chain End INT EN	0	R/W	**Chain End Interrupt Enable:** This bit enables (or disables) the generation of an interrupt when the link chain ends. 0 = No interrupt is generated 1 = Generates interrupt when the link chain is completely executed
8	Link End INT EN	0	R/W	**Link End Interrupt Enable:** This bit enables (or disables) the generation of an interrupt when the link has been executed. 0 – No interrupt is generated 1 – Generates an interrupt when a link is executed
7	EOP End	0	R/W	**End on End of Packet Mode:** This bit is used to enable (or disable) the end of DMA transfer when the end of a packet is reached. '0' = Does not end the DMA transfer when an end of packet is reached and continues until the DMA transfer length is achieved. '1' = Ends DMA transfer when an end of packet is reached.
6:4	Reserved	N/A	N/A	**Reserved**
3	INCR AXI Mode	0	R/W	**AXI Burst Increment Mode:** This bit enables (or disables) the auto increment (burst) mode of the target AXI4 Memory. '0' = Disable '1' = Enable

Table 5-3: Link Descriptor Control Word Bit Definitions (Continued)

Bits	Field Name	Default Value	Access Type	Description
2:1	Reserved	N/A	N/A	**Reserved**
0	AUTO	0	R/W	**Start Mode:** This bit indicates the mode of operation of the DMA engine. 0 = Manual mode: DMA advance bit in the DMA Advance Register must be toggled by the user to start execution of the link descriptor 1 = Auto mode: Execution of the link descriptor starts automatically **Note:** The first link in a link descriptor chain requires the DMA Advance Register's advance bit to be toggled to start execution regardless of the setting of this bit.

When metadata packets are written to the AXI4 Memory by setting the **`write_meta_data`** bit to '1' in the Control Word of the Link Descriptor (see Table 5–3), the output data packet is 128 bits wide with the bit definitions as shown in Table 5–4..

Table 5-4: Metadata Packet Bit Definitions

Bit Index	Name	Width	Description
127:126	Reserved	2	**Reserved**
125	last_is_eop	1	**End of Packet:** This bit indicates the end of the packet. Active High.
124	first_is_sop	1	**Start of Packet:** This bit indicates the start of the packet. Active High.
123	type	1	**Type of First Sample:** This bit indicates the type of the first sample of data. '0' = I '1' = Q
122	data_type	1	**Data Type:** '0' = Real '1' = I/Q
121:120	data_format	2	**Data Format:** "00" = 8–bit "01" = 16–bit "10" = 32–bit "11" = Reserved
119:112	chan_num	8	**Channel Number:** These bits indicate the channel number of the ADC channel in the user design that the data is being transferred from.
111:100	counter	12	**Metadata Packet Counter:** These bits indicate the value of a counter implemented to increment each time a metadata packet is generated. The counter resets to 0x000 when the DMA resets or a DMA abort is initiated.
99:96	user_bits	4	**User–defined Bits:** These bits are tied to “0000”.
95:64	valid_bytes	32	**Number of Valid Bytes:** These bits indicate the number of valid bytes being transferred in the data packet.
63:0	timestamp	64	**Timestamp:** These bits indicate the timestamp of the data. The timestamp information is received in the sideband user data of the AXI4–Stream Slave Interface.

This page is intentionally blank

Chapter 6: Designing with the Core

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

6.1 General Design Guidelines

The AXI4–Stream PPLD to AXI4 DMA Core provides the required logic to transfer data from the input PPLD AXI data stream to an AXI4 memory host. The user can control the DMA operation by setting the control registers in the Register Space (as described in Chapter 4) to the desired values.

6.2 Clocking

Main Clock: **aclk**

This clock is used to clock all the ports and logic in the AXI4–Stream PPLD to AXI4 DMA Core.

Interrupt Clock: **irq_clk**

This clock provides the synchronization for the **irq** output.

6.3 Resets

Main reset: **aresetn**

This is an active low synchronous reset associated with **aclk**. When this reset is asserted, all state machines in the core are reset and FIFOs are flushed. Caution should be exercised that this reset is not asserted while the DMA is running in order to avoid generation of incomplete or malformed packets at the AXI4 memory host interface. When possible, an abort sequence should be used instead when the user design needs to stop the DMA operation.

NOTE: This reset (**aresetn**) does not affect the control/status registers.

CSR Reset: **s_axi_csr_aresetn**

This is an active low synchronous reset associated with **aclk**. When asserted, this reset will return all control registers back to their default states, and will clear the interrupt enables and flags. Except for the CSR logic, it does not reset any of the other state machines, nor flush the FIFO in the core.

NOTE: Both of these resets do not clear the contents of the descriptor RAM.

6.4 Interrupts

This core has an edge type (rising edge−triggered) interrupt output, which is synchronous with the **irq_clk**. On the rising edge of any ***enabled*** (see Section 4.14) interrupt signal, a single−clock−cycle wide pulse is output from the core on the **irq** output.

Each interrupt event is also stored in two registers, accessible via 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.

NOTE: All interrupt sources are latched in the Interrupt Flag Register, even when an interrupt source is not enabled (via the Interrupt Enable Register).

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.

6.5 Interface Operation

- **CSR Interface:** This is the control/status register interface. It is associated with **aclk**. It is a standard AXI4−Lite Slave interface. Typically, this interface is connected along with other cores' AXI4−Lite interfaces through an AXI−Lite Crossbar core or a series of AXI−Lite Crossbar cores that route AXI−Lite accesses through to the desired core based on the address range.
- **Linked List Descriptor (DESCR) RAM Interface:** This is the DMA Linked List Descriptor RAM interface which is used by the core to configure the DMA commands. It is associated with **aclk** and is a standard AXI4 Master interface.
- **AXI4 Memory Interface:** This is the AXI4 Memory Interface which is associated with **aclk** and is used to transfer write requests from the AXI4−Stream PPLD to AXI4 DMA Core to the target AXI4 Memory host. This is a standard AXI4−Stream Master interface.
- **Packetized Sample Data/ Timestamp/ Information Streams (PPLD) Interface:** This core implements an AXI4−Stream Slave interface across the input to receive AXI4 PPLD−style streams and is associated with **aclk**. For more details about this interface, refer to Section 3.2.

6.6 Programming Sequence

This section briefly describes the programming sequence for the AXI4–Stream PPLD to AXI4 DMA Core.

1) Set up the linked list RAM.

2) Set the control registers with the required values.

3) Enable DMA interrupts.

4) Reset the DMA.

5) Start the DMA.

6) When done, check the interrupt flag register and clear the interrupts.

6.7 Timing Diagrams

The timing diagram for the AXI4–Stream PPLD to AXI4 DMA Core is shown in Figure 7–3. This timing diagram is obtained by running the simulation of the test bench of the core in Vivado VSim environment. For more details about the test bench, refer to Section 7.5.

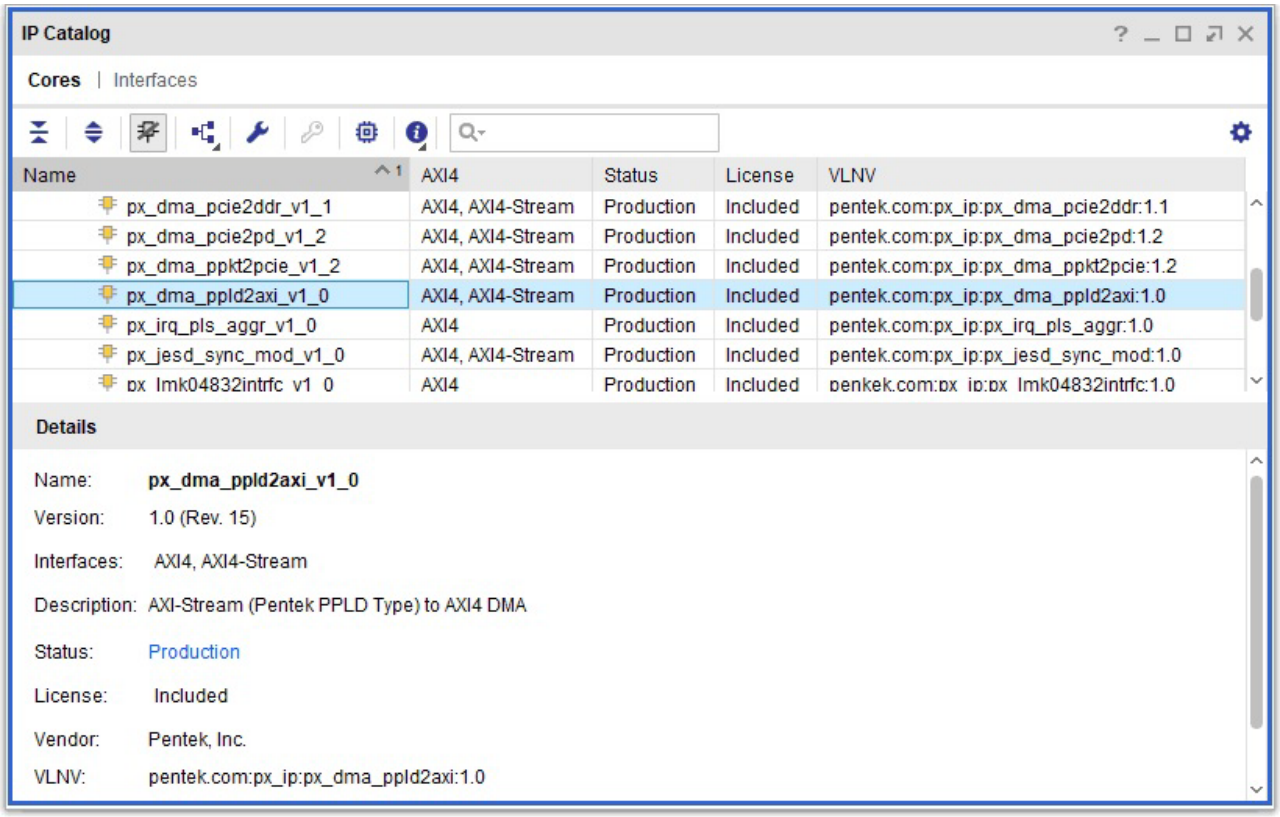
This page is intentionally blank

Chapter 7: Design Flow Steps

7.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek AXI4–Stream PPLD to AXI4 DMA 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_dma_ppld2axi_v1_0** as shown in Figure 7–1.

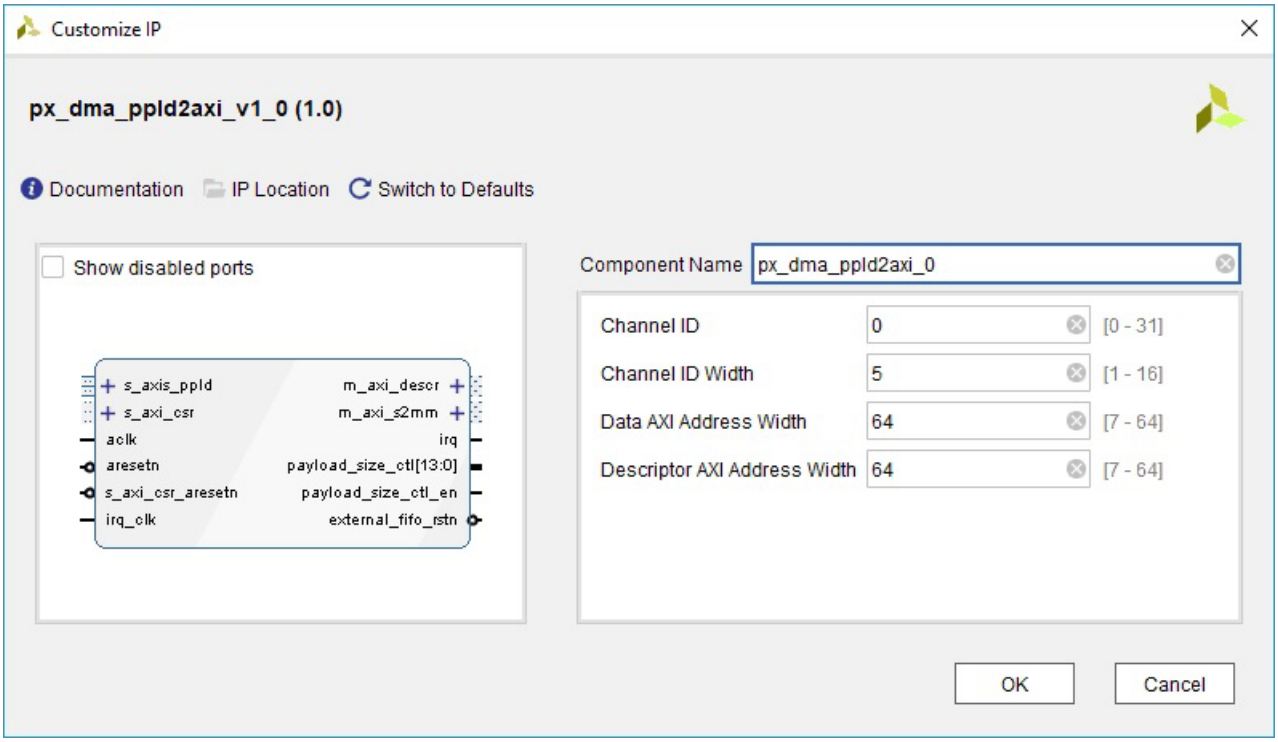
Figure 7–1: AXI4–Stream PPLD to AXI4 DMA Core in Pentek IP



7.1 Pentek IP Catalog (continued)

When you select the **px_dma_ppld2axi_v1_0** core, a screen appears that shows the core's symbol and the core's parameters (see Figure 7–2). The core's symbol is the box on the left side.

Figure 7–2: AXI4–Stream PPLD to AXI4 DMA Core IP Symbol



7.2 User Parameters

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

7.3 Output Generation

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

7.4 Constraining the Core

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

Required Constraints

The XDC constraints are not provided with the AXI4−Stream PPLD to AXI4 DMA 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 main clock (**aclk**) of the AXI4−Stream PPLD to AXI4 DMA Core has maximum frequency of 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.

7.5 Simulation

The AXI4–Stream PPLD to AXI4 DMA Core has a test bench which generates output waveforms using the Vivado VSim environment. This test bench is designed to run at 250MHz for both the main clock (**aclk**) and the interrupt clock (**irq_clk**).

The DMA writes to two buffer spaces whose link descriptors are defined in a **test_parameters.txt** file. This file also contains the required values for the control registers within the Register Space of the AXI4–Stream PPLD to AXI4 DMA Core. The Linked List Descriptor RAM is set up by the test bench by reading from the test_parameters.txt file. This file can be found in the top–level of the project directory.

The DMA control registers are also set with desired values as defined in the test parameters file. The DMA is made operational by toggling the control bit of the DMA Advance control register. The programming sequence of the DMA is the same as described in Section 6.6.

The contents of the **test_parameters.txt** file along with descriptions of the parameters are provided in Table 7–1..

Table 7-1: Test Parameters File Contents and Parameter Descriptions

Parameter	Type	Value	Description
indata_gate_length	Integer	1,000	**Gate Length:** This parameter indicates the length in 16 bit samples of active gate time for the input data stream.
indata_gate _inactivetime	Integer	256	**Gate Inactive Time:** This parameter indicates the length in 16 bit samples of inactive gate time between active gates. This value must be a multiple of input data stream word width (16–bit words).
indata_gate_repeat	Integer	2	**Gate Repetitions:** This indicates the number of times the active gate period is to be repeated.
max_payload_size	Integer	128	**Maximum Payload Size:** Controls maximum payload size allowed. Allowable values are: 1,2,4,8,16,32,64,128,256,512,1024,2048,4096,8192
link_start_address	std_logic _vector [63:0]	0x00000000 00000000	**First Linked List Descriptor Address:** Address of the first linked list descriptor
dest_a_address	std_logic _vector [63:0]	0x00000000 10000000	**Destination Start Address:** Address to program in first descriptor

Table 7-1: Test Parameters File Contents and Parameter Descriptions (Continued)

Parameter	Type	Value	Description
dest_a_metaaddress	std_logic _vector [63:0]	0x00000000 20000000	**Meta Data Destination Start Address:** Address to program in first descriptor
buffera_bytesize	Integer	2,048	**Buffer A Bytes to Transfer:** (limited in size for test) to program in first descriptor
dest_b_address	std_logic _vector [63:0]	0x00000000 30000000	**Destination Start Address:** Address to program in second descriptor
dest_b_metaadress	std_logic _vector [63:0]	0x00000000 40000000	**Meta Data Destination Start Address:** Address to program in second descriptor
bufferb_bytesize	Integer	2,048	**Buffer B bytes to transfer:** (limited in size for test) to program in second descriptor
Link Descriptor #1 (Refer to Table 5–2 and Table 5–3)			
a_auto	Boolean	True	**Start Mode:** This parameter indicates the mode of operation of the DMA engine. When set to True, the DMA works in auto mode, and when set to False the DMA runs in manual mode.
a_write_meta	Boolean	True	**Write metadata:** When set to False, this parameter disables writing of metadata packets to the DMA.
a_incr_mode	Boolean	True	**AXI Burst Increment Mode:** This enables the auto increment (burst) mode of the target AXI4 Memory.
a_eop_end	Boolean	True	**End on End of Packet Mode:** When set to True, the DMA transfer ends after an "end of packet" marker is reached in the input data stream.
a_link_end_int_en	Boolean	True	**Link End Interrupt Enable:** When set to True, this parameter enables the generation of an interrupt when a link is executed.
a_chain_end_int_en	Boolean	False	**Chain End Interrupt Enable:** When set to True, this parameter enables generation of an interrupt when a link chain is completely executed.
a_chain_end	Boolean	False	**Chain End:** When set to False, this parameter indicates that this link descriptor is not the end of the link chain.
Link Descriptor #2 (Refer to Table 5–2 and Table 5–3)			

Table 7-1: Test Parameters File Contents and Parameter Descriptions (Continued)

Parameter	Type	Value	Description
b_auto	Boolean	True	**Start Mode:** This parameter indicates the mode of operation of the DMA engine. When set to True, the DMA works in auto mode, and when set to False the DMA runs in manual mode.
b_write_meta	Boolean	True	**Write Metadata:** When set to False, this parameter disables writing of metadata packets to the DMA.
b_incr_mode	Boolean	True	**AXI Burst Increment Mode:** This enables the auto increment (burst) mode of the target AXI4 Memory.
b_eop_end	Boolean	True	**End on End of Packet Mode:** When set to True, the DMA transfer ends after an "end of packet" marker is reached in the input data stream.
b_link_end_int_en	Boolean	True	**Link End Interrupt Enable:** When set to True, this parameter enables the generation of an interrupt when a link is executed.
b_chain_end_int_en	Boolean	False	**Chain End Interrupt Enable:** When set to True, this parameter enables generation of an interrupt when a link chain is completely executed.
b_chain_end	Boolean	False	**Chain End:** When set to False, this parameter indicates that this link descriptor is not the end of the link chain.

The input data stream to the DMA is generated by the test bench for a gate length of 1,000 16–bit samples, with the gate repeated twice.

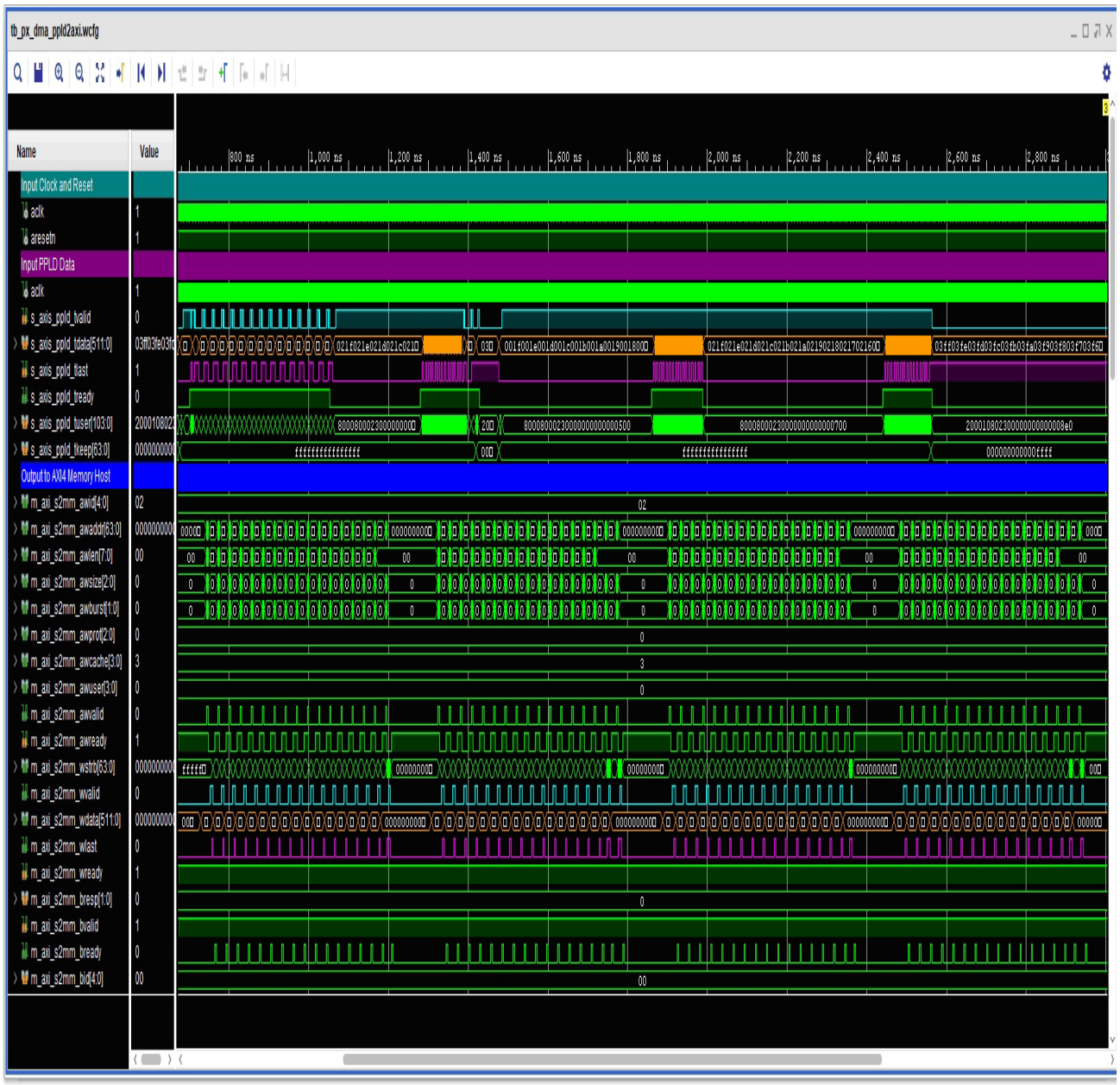
This test bench has Link Descriptors 1 and 2 defined to transfer 2048 bytes of data from the input to the target AXI4 memory, with a maximum payload size of 128 bytes.

When the test bench is run, the simulation produces the resulting waveforms for the input and output data paths as shown in Table 7–3. The control signals and other useful signals can be seen in Table 7–4.

The generated AXI4 memory write request data streams for the DMA can be found in the **`test_results.txt`** file, which will be in the top–level directory of the project.

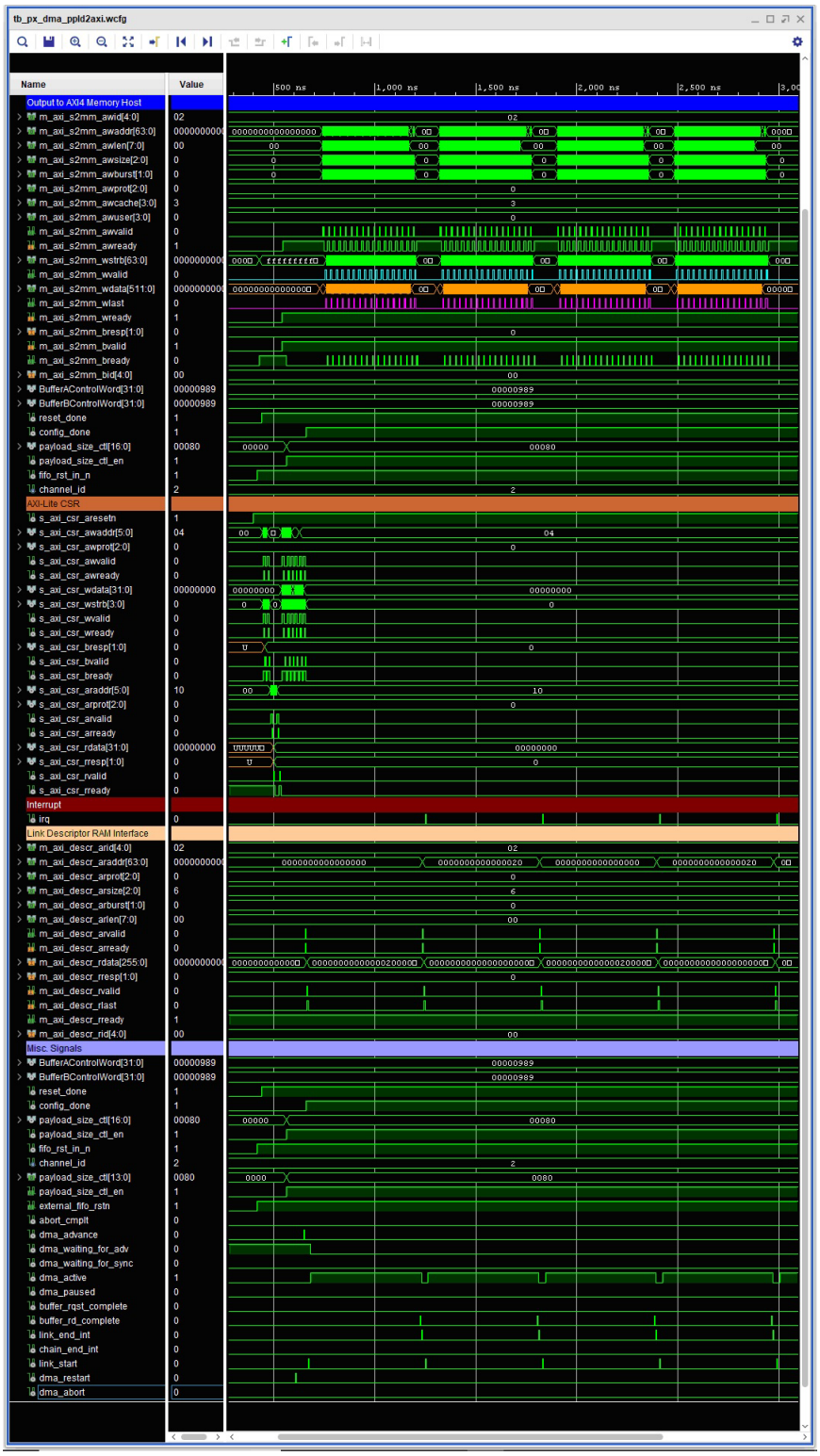
7.5 **Simulation** (continued)

Figure 7–3: AXI4–Stream PPLD to AXI4 DMA Core Test Bench Simulation Input and Output Data Streams



7.5 Simulation (continued)

Figure 7–4: AXI4–Stream PPLD to AXI4 DMA Core Test Bench Simulation Control and Other Signals



7.6 Synthesis and Implementation

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

This page is intentionally blank