IP CORE MANUAL

AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator IP

px_axis_ddr2wave_256

PENTEK

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

Manual Part Number: 807.48410 Rev: 1.2 – April 11, 2018

Manual Revision History

Date	Version	Comments
2/14/18	1.0	Initial Release
3/16/18	1.1	Updated Sect 4.1 and Sect 4.7.
4/11/18	1.2	Revised Table 4–2, Table 4–7, Table 4–9, Table 5–2, and Table 7–1 to add 8–bit real and trig–ger settings.

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

Table of Contents

Page

This page is intentionally blank

IP Facts

Description

Pentek's Navigator™ AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core acts as a bridge between the Xilinx® DDR4 SDRAM memory and a Digital to Analog Converter in the user design. This core converts the DDR4 SDRAM data, received from read requests, into AXI4–Streams to be used by the DAC in the user design. This core operates using a linked–list methodology with up to 128 link descriptors to define trigger length, trigger delay, and DDR4 address range.

This core complies with the ARM® AMBA® AXI4 Specification. This user manual defines the hardware interface, software interface, and parameterization options for the AXI4–Stream DDR4 SDRAM to DAC Waveform Generator Core.

Features

- Linked–list descriptors allow programming of trigger behavior and DDR4 address range to retrieve the DAC waveform from.
- Operates in trigger or gate mode
- 128 link descriptors can be modified during operation
- Interrupts notify of start and end of acquisition and also start and end of link descriptor execution.
- Compatible with Pentek's DDR4 AXI4–Stream request and response bus definitions
- Status of present link descriptor number is accessible over the AXI4–Lite Control/ Status Register Bus.
- Use for 256–bit wide DAC interfaces

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 2017.2 or later
Simulation	Vivado VSim
Synthesis	Vivado Synthesis
Support	
Provided by Pentek fpgasupport@pentek.com	

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

This page is intentionally blank

Chapter 1: Overview

1.1 Functional Description

The AXI4–Stream DDR4 SDRAM to DAC Waveform Generator Core provides a trans–action interface between the DDR4 SDRAM Memory and a DAC in the user design. This core generates DDR4 read requests and accepts the DDR4 read responses which are converted into AXI4–Streams to be used by the DAC. The output AXI4–Streams generated by the core include sample data with time–aligned copy of timing events (gate, sync, and PPS), and data information.

A unique requester ID identifying the DDR4 request data streams can be defined by the user through the generic parameter **id** (see Section 2.5). This allows multiple requestors to access the same DDR4 memory. This core uses linked list methodology with up to 128 link descriptors to define the trigger length, trigger delay, and DDR4 memory address range. This core stores link descriptors in an internal RAM. This Link List Descriptor RAM can be accessed and set up by an AXI Master in the user design through an AXI4–Lite Interface.

The DAC Waveform Generator Core receives timing event data streams from the user design which are used to generate the data acquisition gate signal to control the data flow. The gate signal is generated based on the mode select bits defined in the Mode Control Register of the core (see Section 4.1). The DAC Waveform Generator Core can operate in three modes, where each mode defines the source of the data acquisition gate signal.

- **Gate mode**: The data acquisition gate signal is generated from the input gate signal from the timing event data streams. Users can define a gate signal by enabling the local gate mode in the Mode Control Register, and then defining the local gate bit (see Table 4–2).

- **Trigger mode:** The gate input signal is used as a trigger to generate a data acquisition gate signal which has a trigger delay and trigger length defined by the control registers in the Register Space (see Chapter 4).

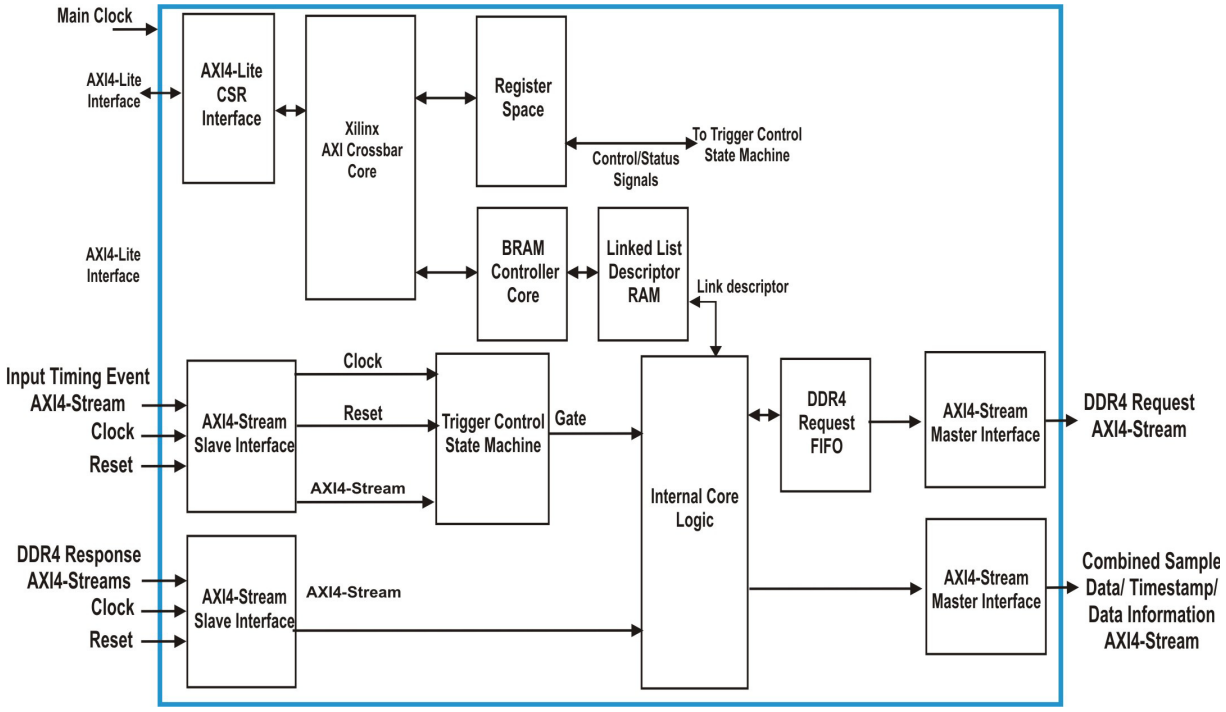
- **Trigger Hold mode:** The gate input signal is used as a trigger to generate a data acquisition gate signal which remains active until the Trigger Control State Machine is reset.

The DAC Waveform Generator Core has a Trigger Control State Machine which is used to generate the packetized data output for the acquisition gate period. The Mode Con–trol Register is used to control the Trigger Control State Machine (see Table 4–2). This core also supports auto increment of the DDR4 memory address, after being set once, for contiguous read requests.

1.1 Functional Description (continued)

Figure 1–1 is a top–level block diagram of the AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core. The modules within the block diagram are explained in other sections of this manual.

Figure 1–1: AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core Block Diagram



- **AXI4–Lite Interface:** This module implements a 32–bit AXI4–Lite Slave Interface to receive the memory read/write requests and also access the register space. For more details about the AXI4–Lite Interface, refer to Section 3.1 AXI4–Lite Core Interfaces.

- **AXI4–Stream Interfaces:** The AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core has AXI4–Stream Interfaces to transfer requests and receive responses to/from the DDR4 SDRAM Memory Controller IP Core. This core also has AXI4–Stream Interfaces to receive timing event data streams and also to transfer the generated output AXI data streams. For more details about the AXI4–Stream Interfaces, refer to **Section 3.2 AXI4–Stream Core Interfaces**.

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

1.1 Functional Description (continued)

- **AXI4–Lite BRAM Controller Core:** This is the Pentek AXI4–Lite BRAM Controller Core which is connected to the AXI4–Lite Interface to communicate with the Linked–List RAM within the DAC Waveform Generator Core.

- **Linked List Descriptor RAM:** This is a Xilinx Dual Port Block RAM generated to store up to 128 link descriptors.

- **DDR4 Request FIFO:** This is a clock crossing AXI4–Stream FIFO used to convert the DDR4 request AXI4–Streams from the AXI4–Stream clock domain to the DDR4 clock domain.

- **Trigger Control State Machine:** This state machine is used generate packed output data streams based on the acquisition gate, and the values defined in the Mode Control Register. This state machine has three states:

  - **Reset** – The Reset state resets the state machine based on the input reset signal (**`s_axis_aresetn`**) from the input AXI4–Stream Slave Interface.

  - **Wait for Trigger Arm** – When the state machine is in the Wait for Arm state, it waits for the trigger arm signal, from the Mode Control Register, to go High.

  - **Armed** – Once in the Armed State, the core waits for the data acquisition gate signal to go High when a valid input is available on the input AXI4–Stream Slave Interface. When the acquisition gate signal goes High, output packed data streams are generated based on the data mode selected.

1.2 Applications

This core can be incorporated into any Kintex Ultrascale FPGA to generate AXI4–Stream waveform data from the DDR4 SDRAM data, to be transferred to a DAC in the user design.

1.3 System Requirements

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

1.4 Licensing and Ordering Information

This core is included with all Pentek Navigator FPGA Design Kits for Pentek Jade series board products. Contact Pentek for Licensing and Ordering Information (www.pentek.com).

1.5 Contacting Technical Support

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

1.6 Documentation

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

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

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

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

Chapter 2: General Product Specifications

2.1 Standards

The AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core has bus interfaces that comply with the *AMBA AXI4–Lite Protocol Specification* and the *AMBA AXI4–Stream Protocol Specification*.

2.2 Performance

The performance of the AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core is limited only by the FPGA logic speed. The values presented in this section should be used as an estimation guideline. Actual performance can vary.

2.2.1 Maximum Frequencies

This core is designed to meet a target frequency of 250 MHz on a Kintex Ultrascale –2 speed grade FPGA. 250 MHz is typically the PCI Express® (PCIe®) AXI Bus clock frequency.

2.3 Resource Utilization

The resource utilization of the AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator 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	288
Memory LUTs	3
Flip–Flops	1586
Block RAM	8
DSP	2

NOTE: Actual utilization may vary based on the user design in which the AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core is incorporated.

2.4 Limitations and Unsupported Features

This section is not applicable to this IP core.

2.5 Generic Parameters

The generic parameters of the AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator 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
id	Integer	**Requester ID:** The user can define a unique ID to identify the data streams through this generic parameter. It can range from 0 - 255. This requester ID is used by the Pentek DDR4 memory interface to identify the source of the memory access requests. It must be unique in a system for all cores that have access to the DDR4 memory. It is the only way that the requested data can be routed back to the correct requester.

Chapter 3: Port Descriptions

This chapter provides port descriptions for the following interface types:

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

3.1 AXI4–Lite Core Interfaces

The AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core has the following AXI4–Lite interface which is used to receive memory read/ write requests from the user design.

3.1.1 AXI4–Lite Slave Interface

Table 3–1 defines the ports in the AXI4–Lite Slave Interface. The AXI4–Lite Interface is connected to a Xilinx AXI Crossbar Core within the core which creates two master interfaces from the single slave interface. One of these master interfaces is used to access the Register Space while the other is used to access the Linked List Descriptor RAM. See the *AMBA AXI4–Lite Specification* for more details on the AXI4–Lite interface.

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

Port	Direction	Width	Description
s_axi_aclk	Input	1	**Clock**
s_axi_aresetn	Input	1	**Reset:** Active low. This will reset the state machine within the core.
s_axi_csr_awaddr	Input	12	**Write Address:** Address used for write operations. It must be valid when **s_axi_csr_awvalid** is asserted and must be held until **s_axi_csr_awready** is asserted by the DAC Waveform Generator Core.
s_axi_csr_awprot	Input	3	**Protection:** The DAC Waveform Generator Core ignores these bits.
s_axi_csr_awvalid	Input	1	**Write Address Valid:** This input must be asserted to indicate that a valid write address is available on **s_axi_csr_awaddr**. The DAC Waveform Generator Core asserts **s_axi_csr_awready** when it is ready to accept the address. The **s_axi_csr_awvalid** must remain asserted until the rising clock edge after the assertion of **s_axi_csr_awready**.

Table 3-1: AXI4-Lite Slave Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axi_csr_awready	Output	1	**Write Address Ready:** This output is asserted by the DAC Waveform Generator Core when it is ready to accept the write address.The address is latched when **s_axi_csr_awvalid** and **s_axi_csr_awready** are high on the same cycle.
s_axi_csr_wdata	Input	32	**Write Data:** This data will be written to the address specified by **s_axi_csr_awaddr** when **s_axi_csr_wvalid** and **s_axi_csr _wready** are both asserted. The value must be valid when **s_axi_csr_wvalid** is asserted and held until **s_axi_csr_wready** is also asserted.
s_axi_csr_wstrb	Input	4	**Write Strobes:** This signal, when asserted, indicates the number of bytes of valid data on the **s_axi_csr_wdata** signal. Each of these bits, when asserted, indicate that the corresponding byte of **s_axi_csr_wdata** contains valid data. Bit 0 corresponds to the least significant byte, and bit 3 to the most significant.
s_axi_csr_wvalid	Input	1	**Write Valid:** This signal must be asserted to indicate that the write data is valid for a write operation. The value on **s_axi_csr _wdata** is written into the register at address **s_axi_csr_awaddr** when **s_axi_csr_wready** and **s_axi_csr_wvalid** are high on the same cycle.
s_axi_csr_wready	Output	1	**Write Ready:** This signal is asserted by the DAC Waveform Generator Core when it is ready to accept data. The value on **s_axi_csr_wdata** is written into the register at address **s_axi_csr_awaddr** when **s_axi_csr_wready** and **s_axi_csr_wvalid** are high on the same cycle, assuming that the address has already or simultaneously been submitted.
s_axi_csr_bresp	Output	2	**Write Response:** The DAC Waveform Generator Core indicates success or failure of a write transaction through this signal, which is valid when **s_axi_csr_bvalid** is asserted; 00 = Success of normal access 01 = Success of exclusive access 10 = Slave Error 11 = Decode Error Note: For more details about this signal refer to the *AMBA AXI Specification*.
s_axi_csr_bready	Input	1	**Write Response Ready:** This signal must be asserted by the user logic when it is ready to accept the Write Response.
s_axi_csr_bvalid	Output	1	**Write Response Valid:** This signal is asserted by the DAC Waveform Generator Core when the write operation is complete and the Write Response is valid. It is held until **s_axi_csr_bready** is asserted by the user logic.

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

Port	Direction	Width	Description
s_axi_csr_araddr	Input	12	**Read Address:** Address used for read operations. It must be valid when **s_axi_csr_arvalid** is asserted and must be held until **s_axi_csr_arready** is asserted by the DAC Waveform Generator Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the DAC Waveform Generator Core.
s_axi_csr_arvalid	Input	1	**Read Address Valid:** This input must be asserted to indicate that a valid read address is available on **s_axi_csr_araddr**. The core asserts **s_axi_csr_arready** when it ready to accept the Read Address. This input must remain asserted until the rising clock edge after the assertion of **s_axi_csr_arready**.
s_axi_csr_arready	Output	1	**Read Address Ready:** This output is asserted by the DAC Waveform Generator Core when it is ready to accept the read address. The address is latched when **s_axi_csr_arvalid** and **s_axi_csr_arready** are high on the same cycle.
s_axi_csr_rdata	Output	32	**Read Data:** This value is the data read from the address specified by the **s_axi_csr_araddr** when **s_axi_csr_arvalid** and **s_axi_csr_arready** are high on the same cycle.
s_axi_csr_rresp	Output	2	**Read Response:** The DAC Waveform Generator Core indicates success or failure of a read transaction through this signal, which is valid when **s_axi_csr_rvalid** is asserted; 00 = Success of normal access 01 = Success of exclusive access 10 = Slave Error 11 = Decode Error Note: For more details about this signal refer to the *AMBA AXI Specification*.
s_axi_csr_rvalid	Output	1	**Read Data Valid:** This signal is asserted by the DAC Waveform Generator Core when the read is complete and the read data is available on **s_axi_csr_rdata**. It is held until **s_axi_csr_rready** is asserted by the user logic.
s_axi_csr_rready	Input	1	**Read Data Ready:** This signal is asserted by the user logic when it is ready to accept the Read Data.
irq	Output	1	**Interrupt:** This is an active high, edge–type interrupt output.

3.2 AXI4–Stream Core Interfaces

The AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core has the following AXI4–Stream Interfaces, used to receive and transfer data streams:

- DDR4 Request (RQST) Interface**:** This core has an DDR4 Memory Request AXI4–Stream Interface at the output of the core to transfer the generated memory read requests.
- DDR4 Response (RSP) Interface**:** This core has an DDR4 Memory Response AXI4–Stream Interface at the input of the core to receive response data streams from the DDR4 Memory Controller core.
- Timing Events (PTCTL) Interface: This is an AXI4–Stream Slave Interface through which timing event data streams are received by the core from the user design.
- Combined Sample Data/ Timestamp/ Information Stream (PDTI) Interface: This is an AXI4–Stream Master Interface of the core used to transfer AXI4–Streams in the PDTI format.

3.2.1 DDR4 Request (RQST) Interface

The AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core implements an DDR4 Request Interface across the output to transfer DDR4 SDRAM memory read request data streams. This is an AXI4–Stream Master Interface. Table 3–2 defines the ports in the AXI4–Stream Master Request Interface. See the *AMBA AXI4–Stream Protocol Specification* for more details on the operation of the AXI4–Stream Interface.

Table 3-2: DDR4 Request Interface Port Descriptions

Port/Signal Name	Direction	Width	Description
m_axis_ddr_aclk	Input	1	**DDR4 SDRAM Clock**
m_axis_ddr_rqst_tdata	Output	512	**Request Data Bus:** This is the DDR4 Memory Request data. For read requests it is not used.
m_axis_ddr_rqst_tvalid		1	**Request Data Valid:** Asserted when data is valid on **m_axis_ddr_rqst_tdata**. The user application can pace the data transfer using the **m_axis_ddr_rqst_tready** signal.
m_axis_ddr_rqst_tuser		256	**Request User Data:** This is the sideband user information data which contains the DDR4 memory request packet header. **Table 3-5** includes the bit definitions of the bits **tuser[256:0].**
m_axis_ddr_rqst_tlast		1	**Request Data Last:** Since all request frames are single cycle in length, this **tlast** signal is asserted on every valid data cycle. It is provided mainly to allow arbitration in AXI Stream switches to use it to arbitrate on **tlast**.

Table 3-2: DDR4 Request Interface Port Descriptions (Continued)			
Port/Signal Name	**Direction**	**Width**	**Description**
m_axis_ddr_rqst_tid	Output	8	**Request Data Stream Identifier:** This is the unique data stream identifier specified by the user using the generic parameter **id**.
m_axis_ddr_rqst_ tready	Input	1	**Request Data Ready:** Activation of this signal by the user logic indicates that the user logic is ready to accept data. Data is transferred across the interface when both **m_axis_ddr_rqst _tvalid** and **m_axis_ddr_rqst_tready** are asserted in the same cycle. If the user application deasserts the ready signal when **m_axis_ddr_rqst_tvalid** is asserted, the core maintains the data on the bus and keeps the valid signal asserted until the user application has asserted the ready signal.

Table 3–3 shows the bit definitions of the DDR4 Request User Data (**m_axis_ddr_rqst_tuser**).

Table 3-3: DDR4 Request User Data Bit Definitions			
Bit Index	**Name**	**Width**	**Description**
255:128	**DATA**	128	**RAM Data:** These bits carry the upper 128 bits of RAM data to be written to the address location in the DDR4 SDRAM. **Note:** The DDR4 memory bus is 640 bit wide. These 128 bits are the upper bit [639:512].
127:120	**RES**	8	**Reserved**
119:40	**MSK**	80	**Byte Mask:** These bits indicate the byte masks of the data which indicate the data on the data bus to be masked. During write requests, bytes with mask bits set to ‘1’ will not be written.
39:36	**RES**	4	**Reserved**
35	**OP**	1	**Type of Request:** This bit indicates the type of Memory request. 0 = Write; 1 = Read
34:32	**ADDRE**	3	**Future Address Expansion:** These bits indicate the address expansion of the DDR4 SDRAM memory location.
31:0	**ADDR**	32	**DDR4 Memory Address:** This is the address location in the DDR4 SDRAM where the read/ write operation is to be performed. The address must be aligned to request size boundaries of 64 bytes.

3.2 AXI4–Stream Core Interfaces (continued)

3.2.2 DDR4 Response (RSP) Interface

The AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core implements an DDR4 Response Interface across the input to receive response data streams. This is an AXI4–Stream Slave Interface. Table 3–4 defines the ports in the AXI4–Stream Slave DDR4 Response Interface. See the *AMBA AXI4–Stream Protocol Specification* for more details on the operation of the AXI4–Stream Interface.

Table 3-4: DDR4 Response Interface Port Descriptions

Port	Direction	Width	Description
s_axis_ddr_rsp_ tdata	Input	512	**Response Data Bus:** This is the Response data received from the DDR memory in response to a DDR4 memory read request.
s_axis_ddr_rsp_ tvalid		1	**Response Data Valid:** Asserted when data is valid on **s_axis_rsp_tdata**. This bus cannot be throttled and there is no **tready** output from the core. The core must be ready to accept data if it requested it.
s_axis_ddr_rsp_ tuser		256	**Request User Data:** This is the sideband user information data which contains the DDR4 response packet header. **Table 3-5** defines the bit definitions of the bits **tuser[255:0].**
s_axis_ddr_rsp_tlast		1	**Request Data Last:** Since all request frames are single cycle in length, this **tlast** signal is asserted on every valid data cycle. It is provided mainly to allow arbitration in AXI Stream switches to use it to arbitrate on **tlast**.
s_axis_ddr_rsp_tid		8	**Response Data Stream Identifier:** This is the unique data stream identifier specified by the user in the request which indicates the request corresponding to the response.

3.2 AXI4–Stream Core Interfaces (continued)

3.2.2 DDR4 Response (RSP) Interface (continued)

Table 3–5 shows the bit definitions of the DDR4 Response User Data (**m_axis_ddr_rsp_tuser**).

Table 3-5: DDR4 Request User Data Bit Definitions

Bit Index	Name	Width	Description
255:128	**DATA**	128	**RAM Data:** These bits carry the upper 128 bits of RAM data being received from the address location in the DDR4 SDRAM. Bits: [639:512]
127:120	**RES**	8	**Reserved**
119:40	**MSK**	80	**Byte Masks:** These bits indicate the byte masks of the data on the data bus to be masked. This information is echoed back from the request.
39:35	**RES**	5	**Reserved**
34:32	**ADDRE**	3	**Future Address Expansion:** These bits indicate the address expansion of the DDR4 SDRAM memory location.
31:0	**ADDR**	32	**DDR4 Memory Address:** This is the address location in the DDR4 SDRAM where the read/ write operation was performed. The address must be aligned to request size boundaries. This information is echoed back from the request.

3.2.3 Timing Events (PTCTL) Interface

Table defines the ports in the Timing Events Interface. This interface is an AXI4–Stream Slave Interface that is used to receive timing events data streams from the user design. See the *AMBA AXI4–Stream Specification* for more details on the operation of the AXI4–Stream Interface.

Table 3-6: Timing Events Interface Port Descriptions

Port	Direction	Width	Description
s_axis_pctl_ tdata	Input	8	**Timing Event Input Data:** This is the timing event data which indicates the gate, sync and PPS signal positions. tdata[0] - Gate tdata[1] - Sync tdata[2] - PPS
s_axis_pctl_ tvalid		1	**Data Valid:** Asserted when data is valid on **s_axis_pctl_tdata.**

3.2 AXI4–Stream Core Interfaces (continued)

3.2.4 Combined Sample Data/ Timestamp/ Information Streams (PDTI) Interface

The Pentek Jade series board products have AXI4–Streams that follow a combined Sample Data/ Timestamp/ Information Stream (PDTI) format. This type of data stream combines sample data with its time–aligned timing events (Gate, Sync, and PPS) and data information. There is an AXI4–Stream Master Interface across the output to transfer AXI4–Streams in the PDTI format. Table 3–2 defines the ports in the AXI4–Stream Slave Combined Sample Data/ Timestamp/ Information Stream Interface. See the *AMBA AXI4–Stream Specification* for more details on the operation of the AXI4–Stream Interface.

Table 3-7: Combined Sample Data/ Timestamp/ Information Streams Interface Port Descriptions

Port	Direction	Width	Description
s_axis_aclk	Input	1	**AXI4–Stream Clock**
s_axis_aresetn	Input	1	**Reset:** Active Low.
m_axis_pdti_tdata	Output	256	**Output DAC Waveform Data:** This core generates output data having 256–bit wide–sample–per–cycle data streams with either sixteen 16–bit real data values, or eight 16–bit I/Q packed data.
m_axis_pdti_tvalid		1	**Input Data Valid:** Asserted when data is valid on **s_axis_pdti_tdata**.
m_axis_pdti_tuser		128	**Sideband Information:** This is the user–defined sideband information received alongside the data stream. **tuser [63:0] – Timestamp[63:0]** **tuser [71:64] – Gate Positions** **tuser [79:72] – Sync Positions** **tuser [87:80] – PPS Positions** **tuser [91:88] – Samples per clock cycle** **tuser[92] – I/Q data of the sample => 0 = I ; 1 = Q** **tuser [94:93] – Data Format => 0 = 8–bit; 1 = 16–bit; 2 = 24–bit; 3 = 32–bit** **tuser [95] – Data Type => 0 = Real; 1 = I/Q** **tuser [103:96] – channel [7:0]** **tuser [127:104] – Reserved** Note: The bits [103:96] define the channel number in the user design from where the data is being received.

3.3 I/O Signals

The I/O port/signal descriptions of the top level module of the AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core are discussed in Table..

Table 3-8: I/O Signals

Port/ Signal Name	Type	Direction	Description
realtime_lost	std_logic	Output	**Real Time Lost:** This output signal is asserted (set to '1') when the gate signal is active and the output DDR4 request FIFO is empty. When this occurs the DAC Waveform Generator Core will have discontinuous output data, and will result in losing real time.

This page is intentionally blank

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the Register Space of the AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core. The memory map is provided in Table 4–1.

Table 4–1: Register Space Memory Map

Register Name	Address (Base Address +)	Access	Description
Mode Control	0x00	R/W	Controls the mode select, data mode select, and trigger state machine arm signals
Trigger Clear	0x04		Controls the clear and disarm signals of the Trigger Control State Machine
Linked List Start	0x08		Controls the loading of the linked list descriptor start index
Linked List Start Index	0x0C		Holds the index of the linked list descriptor to start execution
FIFO Flush	0x10		Controls the reset of the output FIFO and the DDR4 request FIFO
Output Rate Divider	0x14		Control the output data rate
Output Gate Delay	0x18		Control the output gate delay
Reserved	0x1C	N/A	Reserved
Status Register	0x20	R	Indicates the status of the Trigger Control State Machine, and the mode of operation of the core
Current Link Index	0x24	R	Indicates the index of the current link descriptor being executed and the next link descriptor
Reserved	0x28	N/A	Reserved
Reserved	0x2C	N/A	Reserved
Reserved	0x30	N/A	Reserved
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 Mode Control Register

This register controls the trigger arm and stay armed control signals of the Trigger Control State Machine. It is also used to control the mode of operation of the DAC Waveform Generator Core, and the data mode of the input data streams. The Mode Control Register is illustrated in Figure 4–1 and described in Table 4–2.

Figure 4–1: Mode Control Register

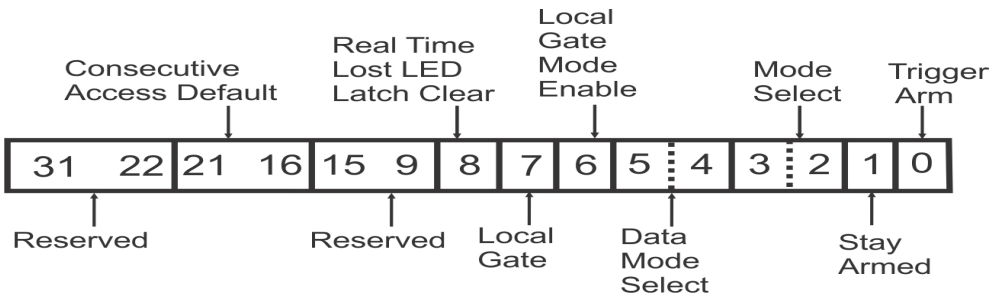


Table 4–2: Mode Control Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:22	Reserved	N/A	N/A	**Reserved**
21:16	consecutive_ access_default	2F	R/W	**Number of Consecutive Access Reads:** This is the number of consecutive memory requests before **`tlast`**. Having an optimal number of memory requests helps maintain maximum memory bandwidth efficency. This will guarantee that this core will access the DDR4 memory for the programmed number of consecutive accesses before it would lose arbitration.
15:9	Reserved	N/A	N/A	**Reserved**
8	rt_clr	0	R/W	**Real Time Lost LED Latch Clear:** This bit is used to clear the real time lost output I/O signal of the DAC Waveform Generator Core. Once the loss of real time is detected the output signal will remain set until cleared. Toggle to ‘1’ then to ‘0’ to clear the latch and get ready to detect a new real time lost event. 0 = Remain unchanged 1 = Clear to ‘0’
7	local_gate	0	R/W	**Local Gate:** This is the user–defined local gate, which is used as the data acquisition gate signal source when the local gate mode is enabled. 0 = Inactive 1 = Active

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

Bits	Field Name	Default Value	Access Type	Description
6	local_gate_mode	0	R/W	**Local Gate Mode Enable:** This bit is used to enable/ disable the local gate mode. When the DAC Waveform Generator Core is operating in Gate mode (**mode_sel** = 00) with local gate mode enabled, the user–defined local gate (bit 7) becomes the source of the data acquisition gate signal generated by the core. 0 = Disable 1 = Enable
5:4	data_mode_sel	0	R/W	**Data Mode Select:** These bits are used to select the data type of the input data to the DAC Waveform Generator Core. For detailed description of output data in each mode, refer to **Table 3–3**. **00 = Single sample real data** **01 = Real 2–channel data or** **Packed I/Q Data** –In a 32–bit word – I(15:0), Q(31:16) **10 = 8–bit Real** **11 = Reserved**
3:2	mode_sel	00	R/W	**Mode Select:** These bits are used to select the mode of operation of the DAC Waveform Generator Core. They define the source of the data acquisition gate signal generated by the core. **00 = Gate mode** => Input gate signal is the source **01 = Trigger mode** => Input gate signal as trigger to generate acquisition gate of user–defined trigger length and trigger delay **10 = Trigger Hold mode** => Input gate signal as trigger to generate acquisition gate which remains active until the Trigger Control State Machine is reset **11 = Reserved**
1	stay_armed	0	R/W	**Stay Armed:** This bit is used to keep the Trigger Control State Machine in the armed state. If this bit is not set, the state machine will go back to the disarmed state after one trigger. 0 = No constraint 1 = Remain in armed state
0	trig_arm	0	R/W	**Trigger Arm:** This bit is used to arm the Trigger Control State Machine. This bit should be toggled to '1' then '0'. 0 = Remains in wait for trigger arm state 1 = Changes from wait for trigger arm to armed state

NOTE: The data_mode_sel, mode_sel, and stay armed controls are latched into the state machine only when the trig–arm bit is toggled. Changing their state, after and while the state machine is armed, will have no effect.

4.2 Trigger Clear Register

This register is used to enable (or disable) a clear (reset) of the Trigger Control State Machine from any state to the Reset state. It is also used to control disarming of the state machine to the Reset State after the acquisition gate ends. This register is illustrated in Figure 4–2 and described in Table 4–3.

Figure 4–2: Trigger Clear Register

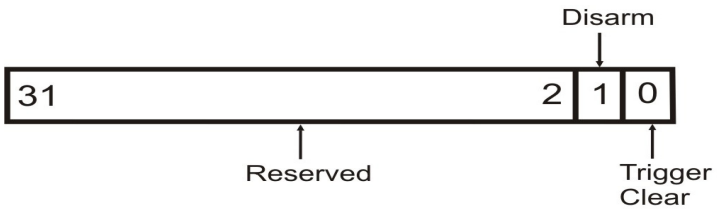


Table 4–3: Trigger Clear Register (Base Address + 0x04)

Bits	Field Name	Default Value	Access Type	Description
31:2	Reserved	N/A	N/A	**Reserved**
1	disarm	0	R/W	**Disarm:** This bit is used to disarm the state machine from armed state to the reset state after the data acquisition gate ends. Toggle to '1' then '0' to disarm. 0 = Disarm disabled 1 = Disarm enabled
0	trig_clear	0	R/W	**Trigger Clear:** This bit used to clear the Trigger Control State Machine from any state to the reset state. 0 = Trigger clear disabled 1 = Trigger is cleared

4.3 Linked List Start Register

This register is used to start the linked list execution of the descriptor at the index specified by the Start Index Register. This register is illustrated in Figure 4–3 and described in Table 4–4.

Figure 4–3: Linked List Start Register

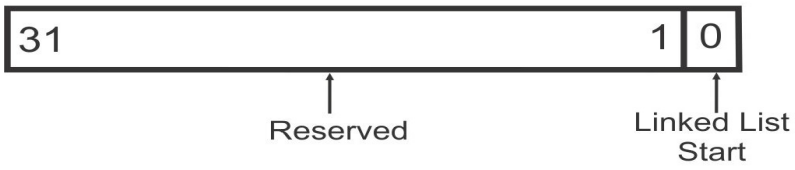


Table 4–4: Linked List Start Register (Base Address + 0x08)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	ll_start	0	R/W	**Linked List Start:** This bit will start link list execution at the linked list index specified in the Linked List Start Index Register, when toggled to '1' then '0'. 0 = No action 1 = Link list start

4.4 Linked List Start Index Register

This register controls the index of the link descriptor to be executed after a reset by the DAC Waveform Generator Core. The value in this register is loaded into the Linked List Descriptor RAM to read the corresponding link descriptor, when the Linked List Start Register is toggled. This register is illustrated in Figure 4–4 and described in Table 4–5.

Figure 4–4: Linked List Start Index Register

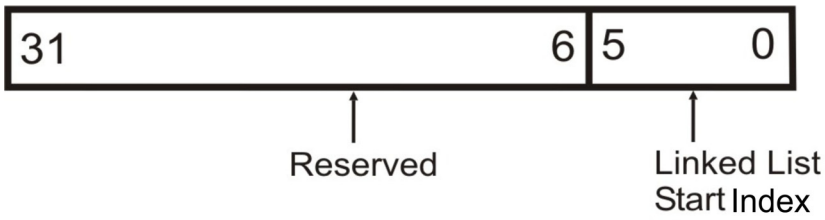


Table 4–5: Linked List Start Index Register (Base Address + 0x0C)

Bits	Field Name	Default Value	Access Type	Description
31:6	Reserved	N/A	N/A	**Reserved**
5:0	ll_start_address	0x00	R/W	**Linked List Start Index:** These bits define the start index of the link descriptor in Linked List Descriptor RAM to be executed. There are up to 64 descriptors in the Linked List Descriptor RAM. Valid values are 0 to 63.

4.5 FIFO Flush Register

The FIFO Flush Register is used to reset the output FIFO and the DDR4 Request FIFO within the DAC Waveform Generator Core. This register is illustrated in Figure 4–5 and described in Table 4–6.

Figure 4–5: FIFO Flush Register

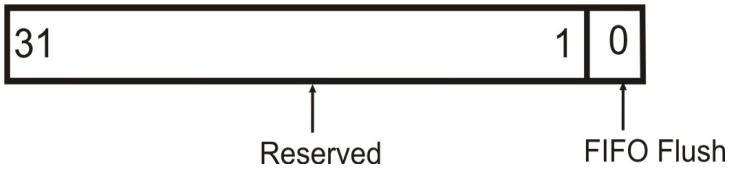


Table 4–6: FIFO Flush Register (Base Address + 0x10)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	fifo_flush	0	R/W	**FIFO Flush:** This bit is used to reset the output FIFO and the DDR4 Request FIFO. 0 = Run 1 = Reset

4.6 Output Rate Divider Register

This register controls the output data rate divider value. The rate divider is used to space out the output data by a programmable (N) number of clock cycles so that if the output data passes through an Interpolation Core in the user design, it can be interpo–lated by N. This register is illustrated in Figure 4–6 and described in Table 4–7.

Figure 4–6: Output Rate Divider Register

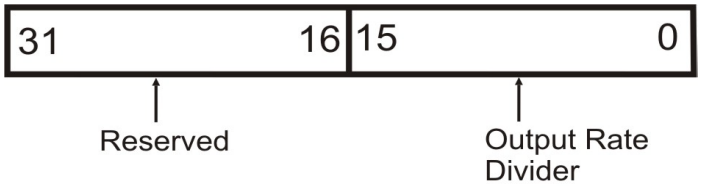


Table 4–7: Output Rate Divider Register (Base Address + 0x14)

Bits	Field Name	Default Value	Access Type	Description
31:16	Reserved	N/A	N/A	**Reserved**
15:0	rate_div	0x0000	R/W	**Output Rate Divider:** These bits control the rate divider of the output data. This value must be set to the desired value N minus one.

4.7 Output Gate Delay Register

This register controls the delay to be introduced to the gate signal and other control signals long enough for data to be received from the DDR4 SDRAM for the generated read request. This register is illustrated in Figure 4–7 and described in Table 4–8.

Figure 4–7: Output Gate Delay Register

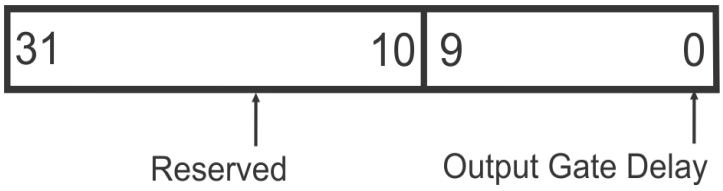


Table 4–8: Output Gate Delay Register (Base Address + 0x18)

Bits	Field Name	Default Value	Access Type	Description
31:10	Reserved	N/A	N/A	**Reserved**
9:0	output_gate_delay	0x00	R/W	**Output Gate Delay:** This is the output delay to be introduced to the gate signal and other control signals.

NOTE: The minimum value required for proper operation will vary based on DAC sample clock frequency. It must be long enough to absorb not only the latency from request to receipt of data from the DDR4 SDRAM, but it also must absorb longer latency periods caused by refresh cycles. If this value is too small you will get a "Real Time Lost" condition, where the data FIFO is empty but data is needed for output.

4.8 Status Register

The Status Register indicates the mode of operation of the core, data mode selected, and the status of the Trigger Control State Machine. This register also indicates the index of the current link descriptor being executed and the next link descriptor to be executed. This register is illustrated in Figure 4–8 and described in Table 4–9.

Figure 4–8: Status Register

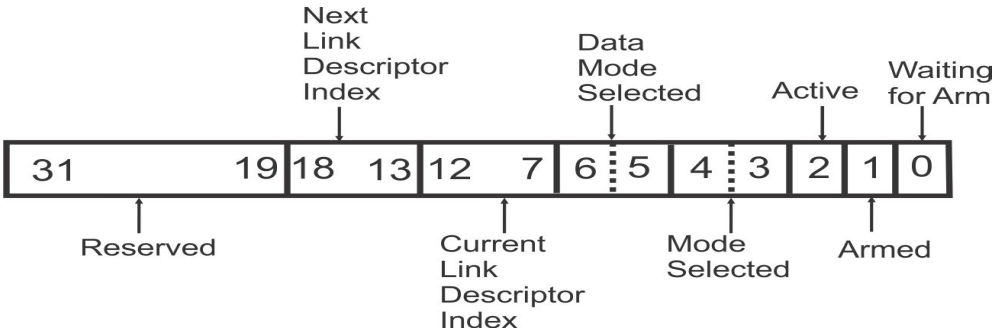


Table 4–9: Status Register (Base Address + 0x20)

Bits	Field Name	Default Value	Access Type	Description
31:19	Reserved	N/A	N/A	**Reserved**
18:13	next_link	0x00	R	**Next Link Descriptor Index:** These bits indicate the index of the next link descriptor to be executed.
12:7	current_link	0x00	R	**Current Link Descriptor Index:** These bits indicate the index of the link descriptor currently being executed.
6:5	data_mode_sel	00	R	**Data Mode Selected:** These bits indicates the data mode selected. **00 = Single Sample Real Data** **01 = Packed I/Q Data or two–channel Real data** **10 = 8–bit real** **11 = Reserved**

Table 4–9: Status Register (Base Address + 0x20) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
4:3	mode_sel	00	R	**Mode Selected:** These bits indicate the selected mode of operation of the core. **00 = Gate mode** **01 = Trigger mode** **10 = Trigger Hold mode** **11 = Reserved**
2	active	0	R	**Active:** This bit indicates that data acquisition and packing is in progress in the Trigger Control State Machine. It is set to '0' for the last packet of data. 0 = data acquisition end 1 = data acquisition in progress
1	armed	0	R	**Armed:** This bit indicates that the Trigger Control State Machine is in the armed state. 0 = state machine not in armed state 1 = state machine in armed state
0	waiting_arm	0	R	**Waiting for Arm:** This bit indicates that the Trigger Control State Machine is in the wait for arm state. 0 = state machine not in wait for arm state 1 = state machine in wait for arm state

4.9 Current Link Index Register

This register indicates the index of the current link descriptor being executed and the next link descriptor to be executed. This register is illustrated in Figure 4–9 and described in Table 4–10.

Figure 4–9: Current Link Index Register

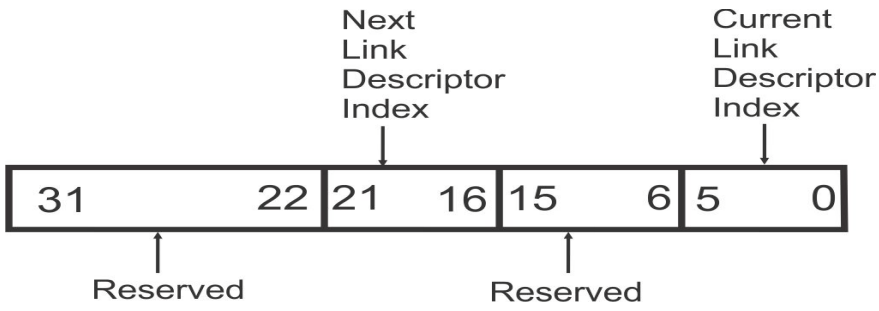


Table 4–10: Current Link Index Register (Base Address + 0x24)

Bits	Field Name	Default Value	Access Type	Description
31:22	Reserved	N/A	N/A	**Reserved**
21:16	next_link	0x00	R	**Next Link Descriptor Index:** These bits indicate the index of the next link descriptor to be executed.
15:6	Reserved	N/A	N/A	**Reserved**
5:0	current_link	0x00	R	**Current Link Descriptor Index:** These bits indicate the index of the link descriptor currently being executed.

4.10 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 4.11). This register is illustrated in Figure 4–10 and described in Table 4–11.

Figure 4–10: Interrupt Enable Register

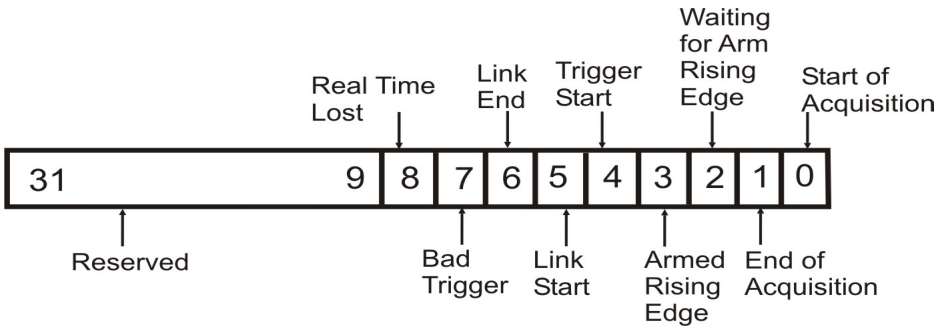


Table 4–11: Interrupt Enable Register (Base Address + 0x34)

Bits	Field Name	Default Value	Access Type	Description
31:9	Reserved	N/A	N/A	**Reserved**
8	realtime_lost	0	R/W	**Real Time Lost:** This bit enables or disables the real time lost interrupt source. 0 = Disable interrupt 1 = Enable interrupt
7	bad_trig	0	R/W	**Bad Trigger:** This bit enables or disables the bad trigger interrupt source. The bad trigger interrupt source is asserted when the trigger occurs before the completion of the last acquisition gate period. 0 = Disable interrupt 1 = Enable interrupt
6	le_int	0	R/W	**Link End:** This bit enables or disables the link end interrupt source. The link end interrupt source indicates that the current link execution is complete. 0 = Disable interrupt 1 = Enable interrupt

Table 4–11: Interrupt Enable Register (Base Address + 0x34) (Continued)

Bits	Field Name	Default Value	Access Type	Description
5	ls_int	0	R/W	**Link Start:** This bit enables or disables the link start interrupt source. The link start interrupt source indicates start of execution of the link descriptor by the core. 0 = Disable interrupt 1 = Enable interrupt
4	trig_start	0	R/W	**Trigger Start:** This bit enables or disables the trigger start interrupt source. The trigger start interrupt source indicates a rising edge on the gate trigger signal from the input timing event data. 0 = Disable interrupt 1 = Enable interrupt
3	armed_re	0	R/W	**Armed Rising Edge:** This bit enables or disables the armed rising edge interrupt source. The armed rising edge interrupt source indicates a rising edge on the **armed** status signal of the status register. 0 = Disable interrupt 1 = Enable interrupt
2	waiting_arm_re	0	R/W	**Waiting for Arm Rising Edge:** This bit enables or disables the wait for arm rising edge interrupt source. The wait for arm rising edge interrupt source indicates a rising edge on the **waiting_arm** status signal of the status register. 0 = Disable interrupt 1 = Enable interrupt
1	acq_end	0	R/W	**End of Acquisition:** This bit enables or disables the end of acquisition interrupt source. The end of acquisition interrupt source indicates the end of data acquisition in the Trigger Control State Machine. 0 = Disable interrupt 1 = Enable interrupt
0	acq_start	0	R/W	**Start of Acquisition:** This bit enables or disables the start of acquisition interrupt source. The start of acquisition interrupt source indicates the start of data acquisition in the Trigger Control State Machine. 0 = Disable interrupt 1 = Enable interrupt

4.11 Interrupt Status Register

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

Some of the interrupt sources are transient and so may not appear in the Interrupt Status Register at the time it is read. In such cases, use the Interrupt Flag Register to see the interrupt conditions that have occurred. The Interrupt Status Register is illustrated in Figure 4–11 and described in Table 4–12.

Figure 4–11: Interrupt Status Register

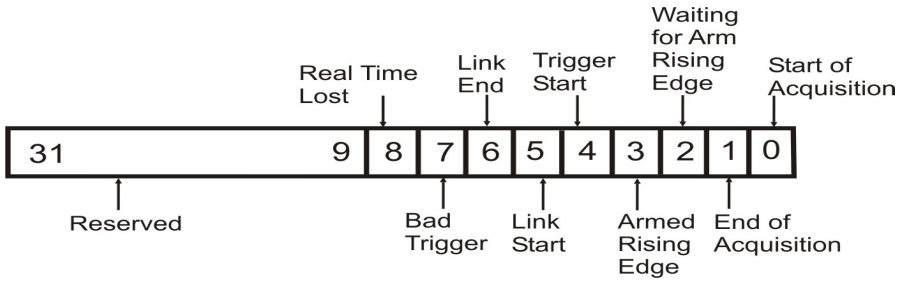


Table 4–12: Interrupt Status Register (Base Address + 0x38)

Bits	Field Name	Default Value	Access Type	Description
31:9	Reserved	N/A	N/A	**Reserved**
8	realtime_lost	0	R	**Real Time Lost:** This bit indicates the status of the real time lost interrupt source. 0 = No interrupt 1 = Interrupt condition asserted
7	bad_trig	0	R	**Bad Trigger:** This bit indicates the status of the bad trigger interrupt source. The bad trigger interrupt source is asserted when the trigger occurs before the completion of the last acquisition gate period. 0 = No interrupt 1 = Interrupt condition asserted

Table 4–12: Interrupt Status Register (Base Address + 0x38) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
6	le_int	0	R	**Link End:** This bit indicates the status of the link end interrupt source. The link end interrupt source indicates that the current link execution is complete. 0 = No interrupt 1 = Interrupt condition asserted
5	ls_int	0	R	**Link Start:** This bit indicates the status of the link start interrupt source. The link start interrupt source indicates the start of execution of the link descriptor by the core. 0 = No interrupt 1 = Interrupt condition asserted
4	trig_start	0	R	**Trigger Start:** This bit indicates the status of the trigger start interrupt source. The trigger start interrupt source indicates a rising edge on the gate trigger signal from the input timing event data. 0 = No interrupt 1 = Interrupt condition asserted
3	armed_re	0	R	**Armed Rising Edge:** This bit indicates the status of the armed rising edge interrupt source. The armed rising edge interrupt source indicates a rising edge on the **armed** status signal of the status register. 0 = No interrupt 1 = Interrupt condition asserted
2	waiting_arm_re	0	R	**Waiting for Arm Rising Edge:** This bit indicates the status of the wait for arm rising edge interrupt source. The wait for arm rising edge interrupt source indicates a rising edge on the **waiting_arm** status signal of the status register. 0 = No interrupt 1 = Interrupt condition asserted
1	acq_end	0	R	**End of Acquisition:** This bit indicates the status of the end of acquisition interrupt source. The end of acquisition interrupt source indicates the end of data acquisition in the Trigger Control State Machine. 0 = No interrupt 1 = Interrupt condition asserted
0	acq_start	0	R	**Start of Acquisition:** This bit indicates the status of the start of acquisition interrupt source. The start of acquisition interrupt source indicates the start of data acquisition in the Trigger Control State Machine. 0 = No interrupt 1 = Interrupt condition asserted

4.12 Interrupt Flag Register

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

Note that when any status bit in the Interrupt Status Register, changes to '1' the corresponding flag bit in this register will also be set to '1'. However, when a status bit in the Interrupt Status Register clears from '1' to '0', the corresponding latched flag bit in this register does not clear, but remains at '1'. To clear the flag bits, write '1's to the desired bits. The flags are not affected by the Interrupt Enable Register. The Interrupt Flag Register is illustrated in Figure 4–12 and described in Table 4–13.

Figure 4–12: Interrupt Flag Register

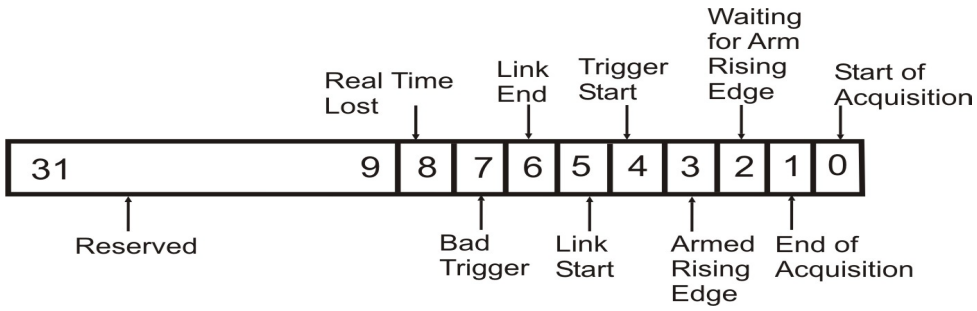


Table 4–13: Interrupt Flag Register (Base Address + 0x3C)

Bits	Field Name	Default Value	Access Type	Description
31:9	Reserved	N/A	N/A	**Reserved**
8	realtime_lost	0	R/Clr	**Real Time Lost:** This bit indicates the status of the real time lost interrupt flag. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch

Table 4–13: Interrupt Flag Register (Base Address + 0x3C) (Continued)

Bits	Field Name	Default Value	Access Type	Description
7	bad_trig	0	R/Clr	**Bad Trigger:** This bit indicates the status of the bad trigger interrupt flag. The bad trigger interrupt source is asserted when the trigger occurs before the completion of the last acquisition gate period. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
6	le_int	0	R/Clr	**Link End:** This bit indicates the status of the link end interrupt flag. The link end interrupt source indicates that the current link execution is complete. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
5	ls_int	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 the link descriptor by the core. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
4	trig_start	0	R/Clr	**Trigger Start:** This bit indicates the status of the trigger start interrupt flag. The trigger start interrupt source indicates a rising edge on the gate trigger signal from the input timing event data. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
3	armed_re	0	R/Clr	**Armed Rising Edge:** This bit indicates the armed rising edge interrupt flag. The armed rising edge interrupt source indicates a rising edge on the **armed** status signal of the status register. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch

Table 4–13: Interrupt Flag Register (Base Address + 0x3C) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
2	waiting_arm_re	0	R/Clr	**Waiting for Arm Rising Edge:** This bit indicates the wait for arm rising edge interrupt flag. The wait for arm rising edge interrupt source indicates a rising edge on the **waiting_arm** status signal of the status register. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
1	acq_end	0	R/Clr	**End of Acquisition:** This bit indicates the end of acquisition interrupt flag. The end of acquisition interrupt source indicates the end of data acquisition in the Trigger Control State Machine. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
0	acq_start	0	R/Clr	**Start of Acquisition:** This bit indicates the start of acquisition interrupt flag. The start of acquisition interrupt source indicates the start of data acquisition in the Trigger Control State Machine. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch

This page is intentionally blank

Chapter 5: Linked List Descriptor RAM Memory Maps

Table 5–1 defines the AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core's Linked List Descriptor RAM memory maps. The RAM can store up to 64 link descriptors.

Table 5–1: Linked List Descriptor RAM Memory Map

Descriptor Index	Address (Base Address +)	Access	Description
Descriptor 0	0x0800	R/W	Trigger Delay
	0x0804		Trigger Length
	0x0808		DDR4 SDRAM Start Byte Address
	0x080C		DDR4 SDRAM End Byte Address
	0x0810		Link Control
	0x0814		Reserved
	0x0818		Reserved
	0x081C		Reserved
Descriptor 1	0x0820 – 0x083C		
Descriptor 2	0x0840 – 0x085C		
.....			
Descriptor 63	0x0FE0	R/W	Trigger Delay
	0x0FE4		Trigger Length
	0x0FE8		DDR4 SDRAM Start Byte Address
	0x0FEC		DDR4 SDRAM End Byte Address
	0x08FF0		Link Control
	0x0FF4		Reserved
	0x0FF8		Reserved
	0x0FFC		Reserved

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 (Base address+)	Bits	Field Name	Access Type	Description
0x0000	**31:0**	trigger_delay	R/W	**Trigger Delay: T**his is the delay to be introduced to the data acquisition gate signal after a trigger event has occurred when the DAC Waveform Generator Core is operating in Trigger mode.
0x0004	**63:32**	trigger_length	R/W	**Trigger Length:** This is the length of the data acquisition gate signal generated by the DAC Waveform Generator Core when it is operating in Trigger mode. This is in terms of samples. **Minimum Setting is based on data mode:** 16–bit real = min. is 16 and must be increments of 16 16–bit I/Q = min. is 8 and must be increments of 8 8–bit real = min. is 32 and must be increments of 32
0x0008	**95:64**	ddr_strt_addr	R/W	**DDR4 SDRAM Start Byte Address:** These bits hold the value of the DDR4 SDRAM start address where a read operation is to be performed. The Start Byte and End Byte Address specify the address range to read.
0x000C	**127:96**	ddr_end_addr	R/W	**DDR4 SDRAM End Byte Address:** These bits hold the value of the DDR4 SDRAM end address. If the trigger length is longer than this, it wraps back around to the start address and continues.
0x0010	**134:128**	next_link	R/W	**Next Link Index:** These bits indicate the index of the next link descriptor to be executed. Valid values are 0 to 63.
	143:135	Reserved	N/A	Reserved
	144	disarm_at_end	R/W	**Disarm:** This bit is used to disarm the Trigger Control State Machine from armed state to the reset state after the data acquisition gate ends. 0 = Disarm disabled 1 = Disarm enabled
	145	link_end_int	R/W	**Link End Interrupt:** This bit generates of an interrupt when the current link descriptor has been executed. Active High.
	146	link_strt_int	R/W	**Link Start Interrupt:** This bit generates of an interrupt when the link descriptor starts execution. Active High.
	159:147	Reserved	N/A	Reserved

Chapter 6: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core.

6.1 General Design Guidelines

The AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core is used to generate read requests to the DDR4 SDRAM and also generate output DAC data streams from the DDR4 responses.

6.2 Clocking

Main Clock: **s_axis_aclk**

This clock is used to clock all the AXI4–Stream ports on the core.

CSR Clock: **s_axi_csr_aclk**

This clock is used to clock CSR Interface ports of the core.

DDR4 SDRAM Clock: **m_axis_ddr_aclk**

This is used to clock the DDR4 SDRAM Interface of the DAC Waveform Generator Core and should be the same clock used by the DDR4 interface.

6.3 Resets

CSR Reset: **s_axi_csr_aresetn**

This is active low synchronous reset associated with the **s_axi_csr_aclk**.

Reset: **s_axis_aresetn**

This is active low synchronous reset associated with the **s_axis_aclk**.

6.4 Interrupts

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

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

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

6.5 Interface Operation

AXI4−Lite Interface: This is an AXI4−Lite Slave Interface and is associated with **s_axi_csr_aclk**. This interfaces to both the Register Space and Linked List Descriptor RAM within the core.

DDR4 Request (RQST) Interface: This is an AXI4−Stream Master Interface used to transfer DDR4 request data streams and is associated with **m_axis_ddr_aclk**. For more details about this interface, refer to Section 3.2.1.

DDR4 Response (RSP) Interface: This is an AXI4−Stream Slave Interface used to receive DDR4 response data streams for the requests transferred by the core and is associated with **m_axis_ddr_aclk**. For more details about this interface, refer to Section 3.2.2.

Timing Events (PTCTL) Interface: This is the interface through which timing events such as the gate signal are input. It is associated with **s_axis_aclk**. For more details about this interface, refer to Section 3.2.3.

Combined Sample Data/ Timestamp/ Information Streams (PDTI) Interface: This core implements an AXI4−Stream Slave interface across the input to receive AXI PDTI streams and is associated with **s_axis_aclk**. For more details about this interface, refer to Section 3.2.4.

6.6 Programming Sequence

This section briefly describes the programming sequence for the AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core.

1) Set the desired values to the control registers.

2) Load the linked list descriptor RAM

3) Write/waveform into the DDR4 SDRAM.

4) Toggle the link start bit.

5) Apply gate or trigger to the start execution.

6.7 Timing Diagrams

The timing diagram for the AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core, shown in Figure 7–3, is obtained by running the simulation of the test bench of the core in Vivado VSim environment. For more details about the test bench, refer to Section 7.5.

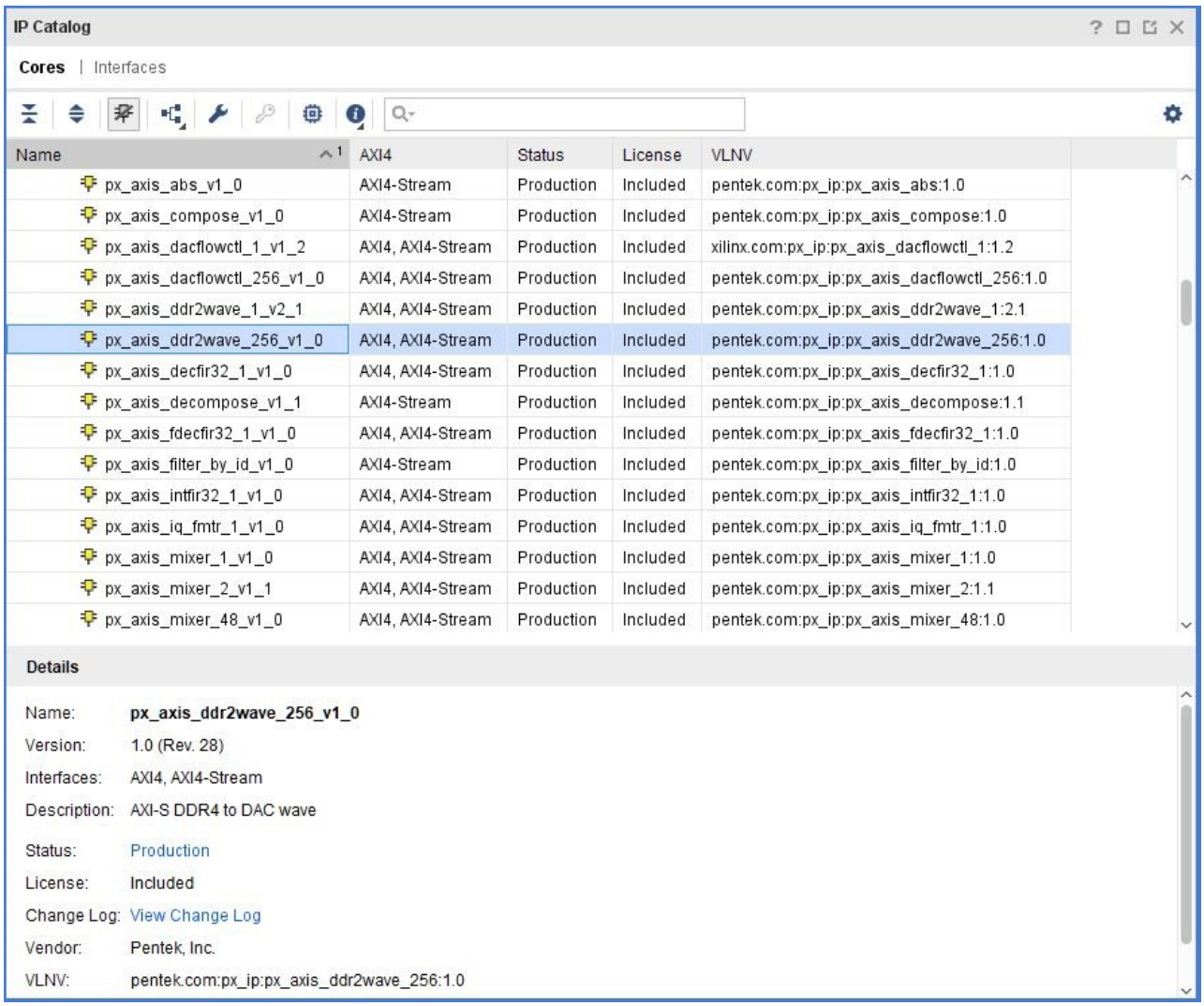
This page is intentionally blank

Chapter 7: Design Flow Steps

7.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core. It also includes simulation, synthesis, and implementation steps that are specific to this IP core. This core can be generated from the Vivado IP Catalog when the Pentek IP Repository has been installed. It will appear in the IP Catalog list as **px_axis_ddr2wave_256_v1_0** as shown in Figure 7–1.

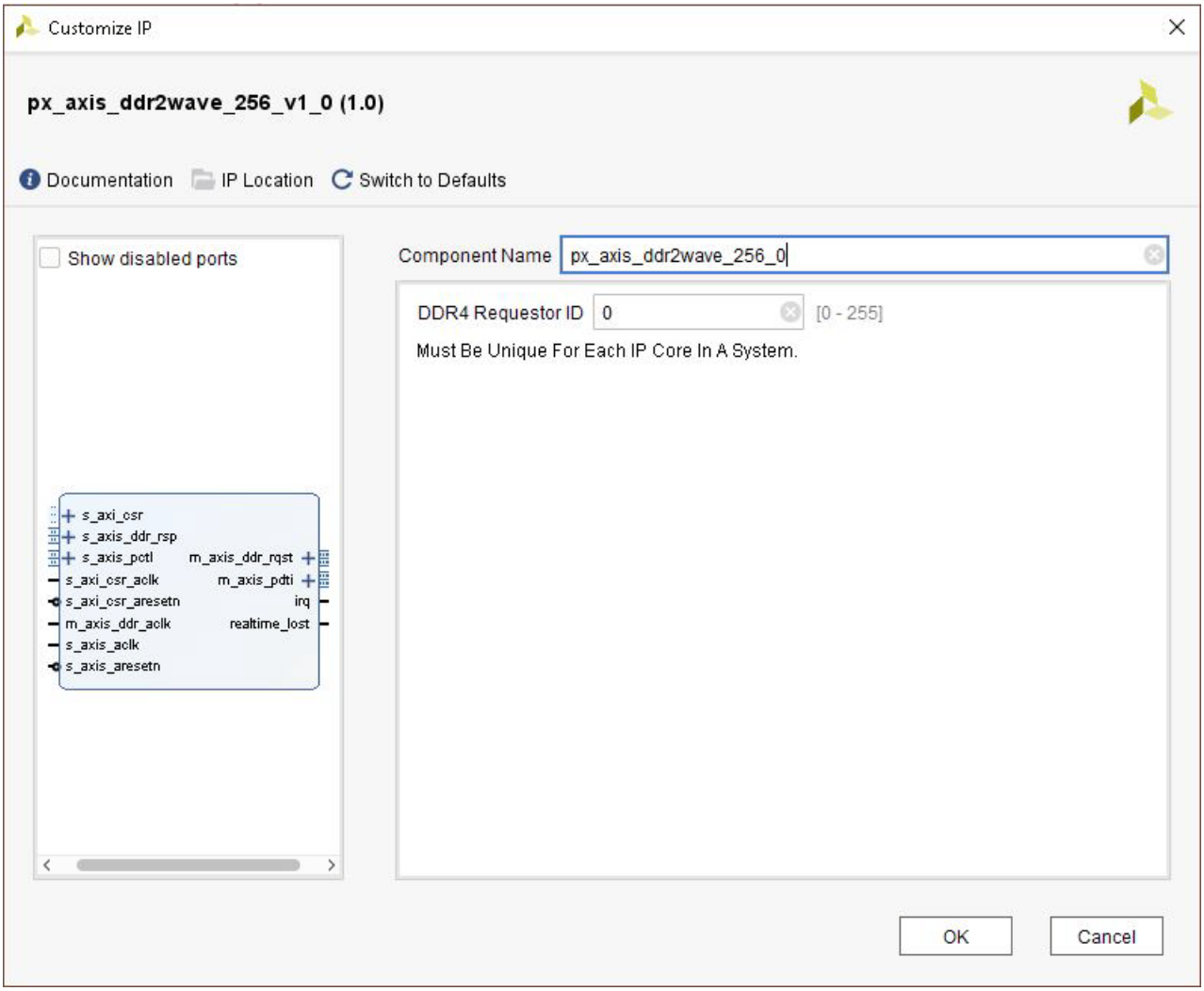
Figure 7–1: AXI4–Stream DDR4 SDRAM to DAC Waveform Generator Core in Pentek IP



7.1 Pentek IP Catalog (continued)

When you select the **px_axis_ddr2wave_256_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 DDR4 SDRAM to DAC Waveform Generator Core Symbol



7.2 User Parameters

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

7.3 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*.

7.4 Constraining the Core

This section contains information about constraining the core in Vivado Design Suite environment.

Required Constraints

The XDC constraints for this core are not included in the Package IP. Clock constraints can be applied at the top level of the user design which includes this IP core.

Device, Package, and Speed Grade Selections

This IP works for the Kintex Ultrascale and Virtex–7 FPGAs.

Clock Frequencies

The clock frequency (**s_axi_csr_aclk**) for this IP core is 250 MHz.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking and Placement

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

7.5 Simulation

The AXI4–Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator Core has a test bench which generates the output waveforms using the Vivado VSim environment. The test bench is designed to run at 250 MHz CSR clock frequency, 200 MHz AXI4–Stream clock frequency and 333 MHz DDR4 SDRAM clock frequency.

The test bench sets the requester ID to 0. The control registers in the Register Space of the DAC Waveform Generator Core are written to based on the values defined in a **`test_parameters.txt`** file. The test parameters file also includes the link descriptors to be loaded into the DDR4 SDRAM, which are written to the core through the AXI4–Lite Interface. 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
Parameter	**Type**	**Value**	**Description**
mode_sel	std_logic_vector	0x1	**Mode Select:** This parameter is used to define the mode of operation of the core. It defines the source of the data acquisition gate signal generated by the core. (See **Table 4–2**) **0x0 = Gate mode** **0x1 = Trigger mode** **0x2 = Trigger Hold mode** **0x3 = Reseved**
data_mode_sel		0x1	**Data Mode Select:** This parameter defines the data type of the input data to the DAC Waveform Generator Core. **0 = Real Data** **1 = Packed I/Q Data** **2 = 8–bit real**
stay_armed	Boolean	True	**Stay Armed:** When set to True, the Trigger Control State Machine is held in the armed state.

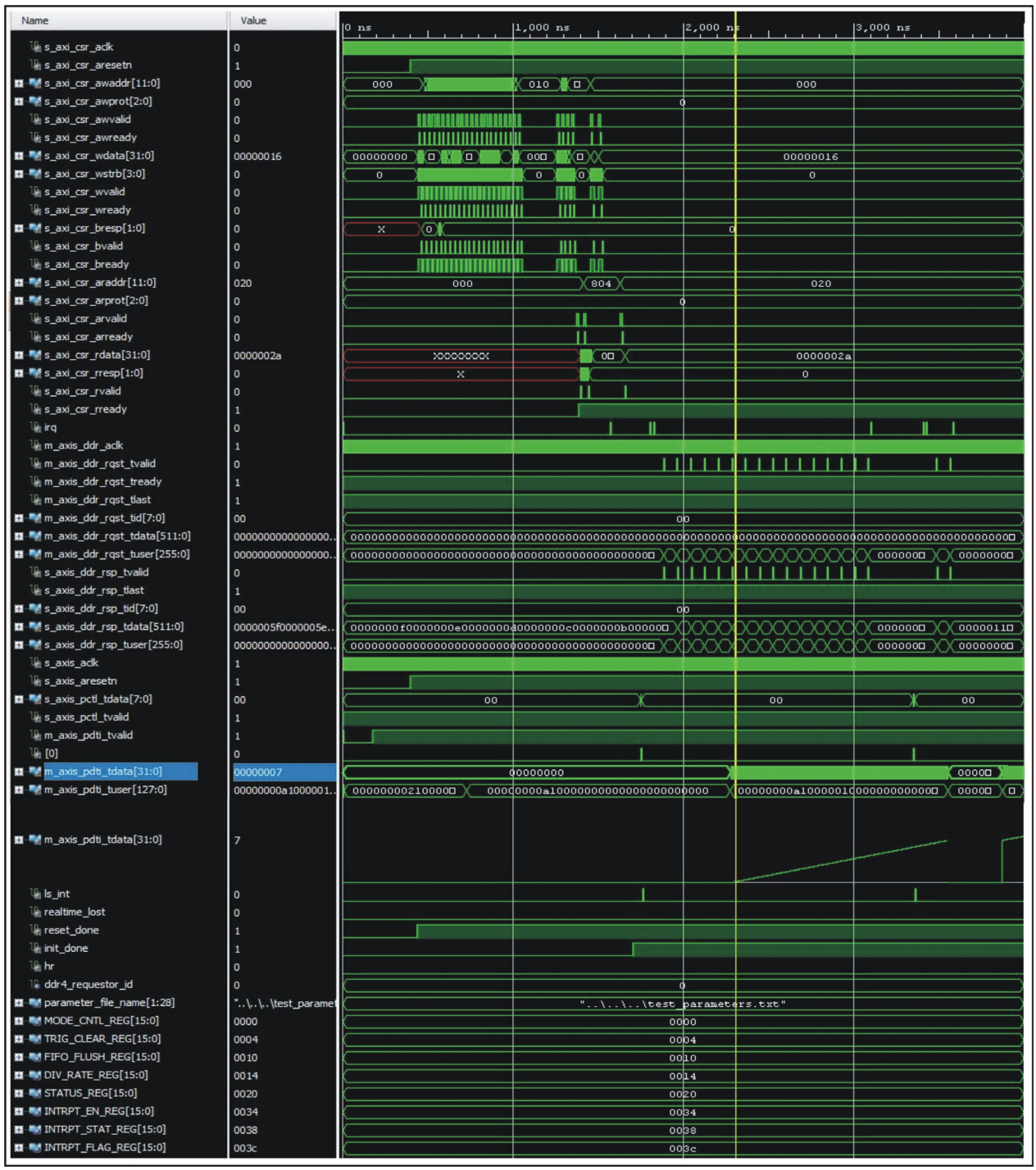
Table 7-1: Test Parameters File Contents and Parameter Descriptions (Continued)

Parameter	Type	Value	Description
output_rate_div	std_logic_ vector	0x00000000	**Output Rate Divider:** This parameter indicates the rate divider value of the output data.
gate_out_dly		0x40	**Output Gate Delay:** This is the output delay to be introduced to the gate signal and other control signals.
trig_space		0x00000140	**Trigger Space:** This parameter indicates the space between two triggers in number of clock cycles.
ll_start_addr		0x00000000	**Linked List Start Address**
interrupt_enable	std_logic_ vector	0x000001ff	**Interrupt Enable:** This parameter indicates the interrupt enables for the interrupt sources of the Interrupt Enable register.
link1_descr_word0		0x00000000	**Link Descriptor 1 Trigger Delay**
link1_descr_word1		0x00000100	**Link Descriptor 1 Trigger Length**
link1_descr_word2		0x80000000	**Link Descriptor 1 DDR4 Start Address**
link1_descr_word3		0x80000000	**Link Descriptor 1 DDR4 End Address**
link1_descr_word4		0x00040001	**Link Descriptor 1 Next Link Address**
link2_descr_word0		0x00000000	**Link Descriptor 2 Trigger Delay**
link2_descr_word1		0x00000020	**Link Descriptor 2 Trigger Length**
link2_descr_word2		0x80000400	**Link Descriptor 2 DDR4 Start Address**
link2_descr_word3		0x90000200	**Link Descriptor 2 DDR4 End Address**
link2_descr_word4		0x00040000	**Link Descriptor 2 Next Link Address**

7.5 Simulation (continued)

The programming procedure is the same as described in Section 6.6. When run, the simulation produces the results shown in Figure 7–3.

Figure 7-3: AXI4-Stream DDR4 SDRAM to DAC 256 Wide Waveform Generator IP Test Bench



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