IP CORE MANUAL

AXI4–Stream DAC Data Flow Control IP

px_axis_dacflowctl_1

PENTEK

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

Manual Part Number: 807.48375 Rev: 1.1 – July 10, 2017

Manual Revision History

Date	Version	Comments
03/29/17	1.0	Initial Release
7/10/17	1.1	Revised Table 4–5

Legal Notices

Copyright

Trademarks

Printed in the United States of America.

Table of Contents

Page

Table of Contents

Page

Chapter 3: Port Descriptions (continued)

Chapter 4: Register Space

Table of Contents

Page

Chapter 5: Designing with the Core

Chapter 6: Design Flow Steps

Table of Contents

Page

This page is intentionally blank

IP Facts

Description

Pentek's Navigator™ AXI4–Stream DAC Data Flow Control Type–1 Core controls the flow of input data AXI4–Streams and timing event AXI4–Streams, and generates a packed Data/ Timestamp/ Data Information output AXI4–Streams, where the start and end of packet are controlled either by a gate signal or trigger signal. This core is a Data Flow Control Type–1 Core which accepts input data with only single sample–per–clock–cycle. This core generates a data acquisition gate signal based on the selected source (gate/ trigger) to control the data flow, and generates output data synchronized to the acquisition gate.

This core complies with the ARM® AMBA® AXI4 Specification and also provides a control/status register interface. This user manual defines the hardware interface, software interface, and parameterization options for the AXI4–Stream DAC Data Flow Control Core.

Features

- Supports 16–bit I/Q packed, or 16–bit real input data streams
- Supports only single–sample–per–clock–cycle input data streams
- Register access through AXI4–Lite interface
- Start and end of packet can be controlled by gate or trigger which is user–defined
- User–programmable trigger length, trigger delay, and timestamp range
- Generates interrupts for start and end of acquisition gate
- Supports user–defined gate signal

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 2016.4 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 Data Flow Control Core (Pentek AXI4–Stream DAC Data Flow Control Core) accepts input data AXI4–Streams from a DMA Core, and Timing Event data streams from a Pentek Sync Bus Interface Core, in the user design. The core generates combined Data/ Timestamp/ Information AXI4–Streams, which include sample data with a time–aligned copy of the timing events (gate, sync, PPS), and data information (see Section 3.2). This core does not support timestamp information. This core has a Register Space which includes the control, status, and interrupt registers, which can be accessed using an AXI4–Lite Interface as shown in Figure 1–1.

The Data Flow Control Core generates a data acquisition gate signal to control the data flow based on the mode select bits defined in the Mode Control Register of the core (see Section 4.1). The Data Flow Control 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 or a user–defined gate signal. 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).

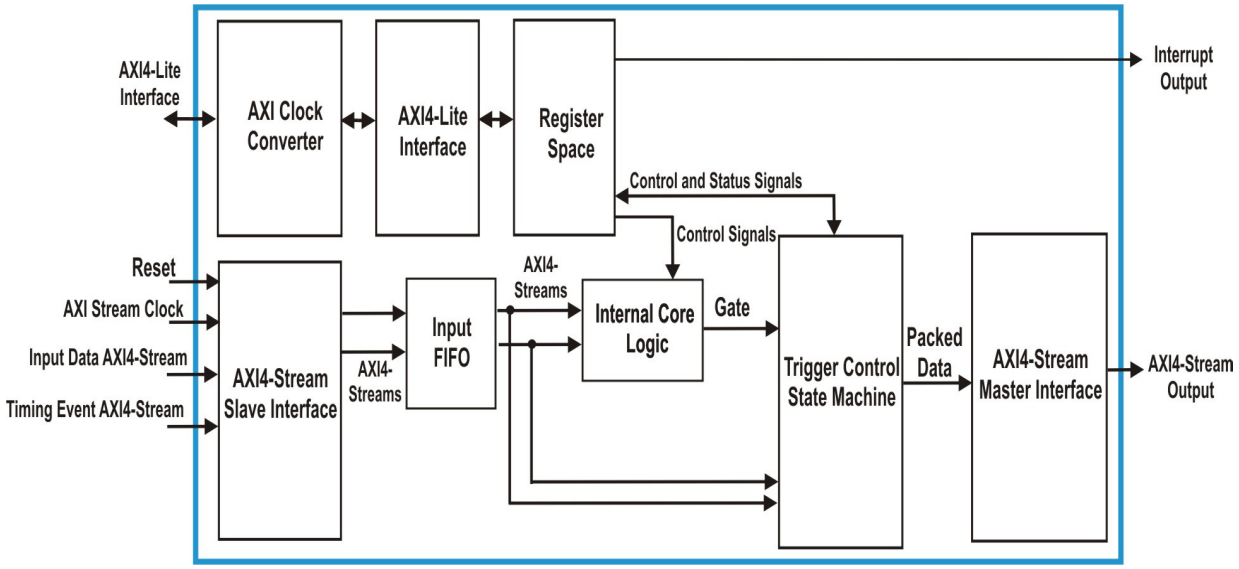
- **Trigger Hold mode:** The gate input signal is used as a trigger to generate a data acquisition gate signal which remains active until the Trigger Control State Machine is reset.

The Data Flow Control Core has a Trigger Control State Machine which is used to gen–erate the packetized data output for the acquisition gate period. The Mode Control Register is used to control the Trigger Control State Machine (see Table 4–2). The Data Flow Control Core also provides edge detection of the active data acquisition gate period for use in creating gate event interrupts.

Figure 1–1 is a top–level block diagram of the Pentek AXI4–Stream DAC Data Flow Control Core. The modules within the block diagram are explained in the later sections of this user manual.

1.1 Functional Description (continued)

Figure 1–1: AXI4–Stream DAC Data Flow Control Core Block Diagram



- **AXI4–Lite Interface:** This module implements a 32–bit AXI4–Lite Slave Interface to access the Register Space. For additional details about the AXI4–Lite Interface, refer to Section 3.1 AXI4–Lite Core Interfaces.

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

- **AXI4–Stream Interfaces:** The Data Flow Control Core has three AXI4–Stream Interfaces. At the input, two AXI4–Stream Slave Interfaces are used to receive input AXI4–Streams and at the output an AXI4–Stream Master Interface is used to transfer packed AXI4–Streams through the output ports. For more details about the AXI4–Stream Interfaces refer to **Section 3.2 AXI4–Stream Core Interfaces**.

- **Input FIFO:** This FIFO is used as a clock crossing FIFO for the input data AXI4–Stream to convert from the PCIe clock domain to the DAC clock domain. The incoming AXI4–Stream data from the DMA will be in the PCIe clock domain.

1.1 Functional Description (continued)

- **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

The AXI4–Stream DAC Data Flow Control Core can be incorporated into any Kintex Ultrascale FPGA where data flow control and packetization of the input AXI4–Stream is required.

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 DAC Data Flow Control 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 Data Flow Control Core is limited only by the FPGA logic speed. The values presented in this section should be used as an estimation guideline. Actual performance can vary.

2.2.1 Maximum Frequencies

The Data Flow Control Core has two incoming clock signals. The AXI4–Stream clock and AXI4–Lite Interface CSR clock, have maximum frequencies 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 Data Flow Control 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	747
Flip–Flops	1938
Memory LUTs	3
DSP	2
Block RAM	8.5

NOTE: Actual utilization may vary based on the user design in which the Data Flow Control Core is incorporated.

2.4 Limitations and Unsupported Features

- This core supports only single–sample–per–clock–cycle data streams.
- The input data streams must be either 16–bit real or 16–bit I/Q packed values.
- This core does not provide timestamp information in the output data streams.

2.5 Generic Parameters

This section is not applicable to this IP core.

Chapter 3: Port Descriptions

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

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

3.1 AXI4–Lite Core Interfaces

The AXI4–Stream DAC Data Flow Control Core uses the Control/Status Register (CSR) interface to control, and receive status from, the user design.

3.1.1 Control/Status Register (CSR) Interface

The CSR interface is an AXI4–Lite Slave Interface that can be used to access the control and status registers in the Data Flow Control 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
s_axi_csr_aclk	Input	1	**Clock**
s_axi_csr_aresetn	Input	1	**Reset:** Active low. This signal will reset all control registers 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 Data Flow Control Core.
s_axi_csr_awprot	Input	3	**Protection:** The Data Flow Control 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 Data Flow Control Core asserts **s_axi_csr_awready** when it is ready to accept the address. The **s_axi_csr_awvalid** must remain asserted until the rising clock edge after the assertion of **s_axi_csr_awready**.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axi_csr_awready	Output	1	**Write Address Ready:** This output is asserted by the Data Flow Control 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 Data Flow Control 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 Data Flow Control 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 Data Flow Control Core when the write operation is complete and the Write Response is valid. It is held until **s_axi_csr_bready** is asserted by the user logic.

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

Port	Direction	Width	Description
s_axi_csr_araddr	Input	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 Data Flow Control Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the Data Flow Control Core
s_axi_csr_arvalid	Input	1	**Read Address Valid:** This input must be asserted to indicate that a valid read address is available on the **s_axi_csr_araddr**. The Data Flow Control 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 Data Flow Control Core when it is ready to accept the read address. The address is latched when **s_axi_csr_arvalid** and **s_axi_csr_ aready** 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 Data Flow Control 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 Data Flow Control 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 Data Flow Control Core has the following AXI4–Stream Interfaces, used to receive and transfer data streams.

- **Data Stream (PD) Interface:** This is an AXI4–Stream Slave Interface used to receive data AXI4–Streams from a DMA Core in the user design.
- **Timing Events (PTCTL) Interface:** This is an AXI4–Stream Slave Interface through which timing event data streams are received from the user design.
- **Combined Data/ Timestamp/ Information Stream (PDTI) Interface:** This is an AXI4–Stream Master Interface of the core used to transfer combined data/ timestamp/ data information AXI4–Streams in the PDTI format.

3.2.1 Data Stream (PD) Interface

The AXI4–Stream DAC Data Flow Control Core implements an Data Stream Interface across the input to receive AXI data streams from a DMA core in the user design. This is an AXI4–Stream Slave Interface. Table 3–2 defines the ports in the Data Stream Interface of the Data Flow Control Core. See the *AMBA AXI4–Stream Specification* for more details on the operation of the AXI4–Stream Interface..

Table 3-2: Data Stream Interface Port Descriptions

Port	Direction	Width	Description
s_axis_aclk	Input	1	**Clock**
s_axis_aresetn			**Reset:** Active Low.
s_axis_pd_tdata		256	**Input Data**
s_axis_pd_tvalid		1	**Input Data Valid:** Asserted when data is valid on **s_axis_pd_tdata**.
s_axis_pd_tkeep		32	**Data Keep:** The **tkeep** signal indicates the valid data sample bytes on **s_axis_pd_tdata**. Each bit corresponds to a 8–bit word in **s_axis_pd_tdata** i.e., bit 0 corresponds to the least significant 8 bits and bit 1 to the most significant. When it is asserted, the data is considered valid. All **tkeep** bits must be '1' contiguously until the **tlast**.
s_axis_pd_tlast		1	**Data Last:** When asserted, this marks the last data in the current data frame.
s_axis_pd_tready	Output		**Data Ready:** The Data Flow Control Core asserts this signal when it is ready to receive data.

3.2 AXI4–Stream Core Interfaces (continued)

3.2.2 Timing Events (PTCTL) Interface

The AXI4–Stream DAC Data Flow Control Core implements an Timing Events AXI4–Stream Slave Interface across the input to receive timing event data streams from the user design. Table defined the ports in the Timing Events Interface of the Data Flow Control Core. See the *AMBA AXI4–Stream Specification* for more details on the operation of the AXI4–Stream Interface..

Table 3-3: Timing Events Interface Port Descriptions

Port	Direction	Width	Description
s_axis_aclk	Input	1	**Clock**
s_axis_aresetn			**Reset:** Active Low.
s_axis_pctl_tdata		8	**Input Data:** This is the timing event data and indicates the gate, sync and PPS positions. tdata[0] – Gate positions tdata[1] – Sync positions tdata[2] – PPS positions
s_axis_pctl_tvalid		1	**Input Data Valid:** Asserted when data is valid on **s_axis_pctl_tdata**.

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

The Pentek Jade series board products have AXI4–Streams that follow a combined Data/ Timestamp/ Information Stream (PDTI) format. This type of data stream combines sample data with its time–aligned timestamp and data information. There is an AXI4–Stream Master Interface across the output to transfer the generated output AXI4–Streams in the PDTI format. Table 3–4 defines the ports in the AXI4–Stream Master 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 3-4: 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.

Table 3-4: Combined Sample Data/ Timestamp/ Information Streams Interface Port Descriptions (Continued)

Port	Direction	Width	Description
m_axis_pdti_tdata	Output	32	**Output Data:** This is the output data stream. **Packed Real Data:** Real data is packed with two consecutive real samples stored in each 32−bit output data word. First sample is placed in bits 15:0 Second sample is placed in bits 31:16 **I/Q Packed Data or two−channel Real Data:** I/Q packed data is packed with each sample from the input data placed in each 32−bit word or dual−channel read data is packed in each 32−bit word. m_axis_pdti_tdata (15:0) – I m_axis_pdti_tdata (31:16) – Q
m_axis_pdti_tvalid		1	**Output Data Valid:** Asserted when data is valid on **m_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] – not supported** **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 DAC Data Flow Control Core are discussed in Table 3–5.

Table 3-5: 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 input FIFO of the core is empty. When this occurs the Data Flow Control 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 DAC Data Flow Control Core. The memory map is provided in Table 4–1.

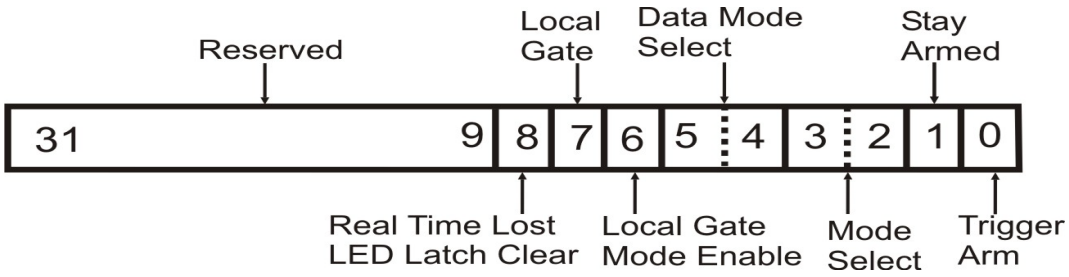
Table 4–1: Register Space Memory Map

Register Name	Address (Base Address +)	Access	Description
Mode Control	0x00	R/W	Controls the mode select, data mode select, and state machine trigger arm signals.
Trigger Clear	0x04	R/W	Controls the clear and disarm signals of the Trigger Control State Machine.
Trigger Delay Value	0x08	R/W	Controls the trigger delay value in Trigger mode.
Trigger Length Value	0x0C	R/W	Controls the trigger length value in Trigger mode.
FIFO Flush	0x10	R/W	Controls reset of the Input FIFO.
Output Rate Divider	0x14	R/W	Controls the output data rate.
Reserved	0x18	N/A	Reserved
Reserved	0x1C	N/A	Reserved
Status	0x20	R	Indicates status of the Trigger Control State Machine, the input FIFO, and the mode of operation of the core.
FIFO Count	0x24	R	Indicates the input FIFO dword count.
Interrupt Enable	0x34	R/W	Interrupt enable bits
Interrupt Status	0x38	R	Interrupt source 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 Data Flow Control Core, and the data mode of the input. 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
31:9	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 Data Flow Control Core. 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
6	local_gate_mode	0	R/W	**Local Gate Mode Enable:** This bit is used to enable/disable local gate mode. When the Data Flow Control 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

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

Bits	Field Name	Default Value	Access Type	Description
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 Data Flow Control Core. For detailed description of output data in each mode, refer to **Table 3–4**. **00 = Packed Real Data** – Two 16–bits of real data packed into a 32–bit word **01 = Packed I/Q Data or two–channel Read data** – In a 32–bit word =>I(15:0), Q(31:16) **10 = Reserved** **11 = Reserved**
3:2	mode_sel	00	R/W	**Mode Select:** These bits are used to select the mode of operation of the Data Flow Control 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 theTrigger Control State Machine in the armed state. 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. 0 = Remains in wait for trigger arm state 1 = Changes from wait for trigger arm to armed state

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

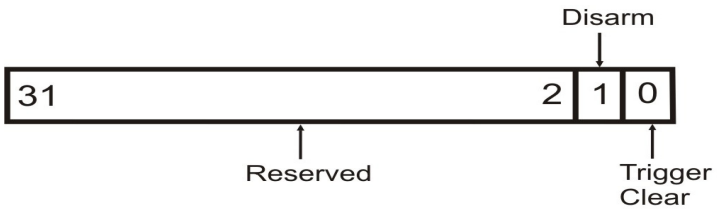


Table 4–3: Trigger Clear Register (Base Address + 0x04)

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. 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 clear enabled

4.3 Trigger Delay Value Register

When the Data Flow Control Core is operating in Trigger mode, the input gate signal is treated as a trigger to generate the data acquisition gate signal. The acquisition gate signal is delayed from the trigger event by a delay value defined by the Trigger Delay Value Register. This register is illustrated in Figure 4–3 and described in Table 4–4.

Figure 4–3: Trigger Delay Value Register

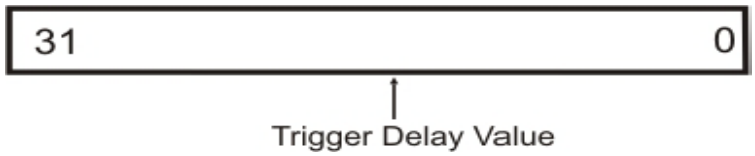


Table 4–4: Trigger Delay Value Register (Base Address + 0x08)

Bits	Field Name	Default Value	Access Type	Description
31:0	trig_dly	0x00000000	R/W	**Trigger Delay Value:** This is the delay to be introduced to the data acquisition gate signal after a trigger event has occurred when the Data Flow Control Core is operating in Trigger mode.

4.4 Trigger Length Value Register

When the Data Flow Control Core is operating in the Trigger mode, the input gate signal is treated as a trigger to generate the data acquisition gate signal. The Trigger Length Value Register is used to control the active gate length of the data acquisition gate signal. This register is illustrated in Figure 4–4 and described in Table 4–5.

Figure 4–4: Trigger Length Value Register

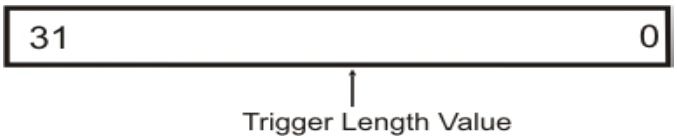


Table 4–5: Trigger Length Value Register (Base Address + 0x0C)

Bits	Field Name	Default Value	Access Type	Description
31:0	trig_len	0x00000010	R/W	**Trigger Length Value:** This is the length of the data acquisition gate signal generated by the gate/trigger generator module when the Packetizer Core is operating in Trigger mode. This value is in terms of data samples. For real data, there are two data samples per clock cycle. For complex data, there is one sample per clock cycle. The value entered for trig–len should be equal to the desired number of samples.

4.5 FIFO Flush Register

This register is used to reset the Input FIFO and clear its contents. This register is illus–trated in Figure 4–5 and described in Table 4–6.

Figure 4–5: FIFO Flush Register

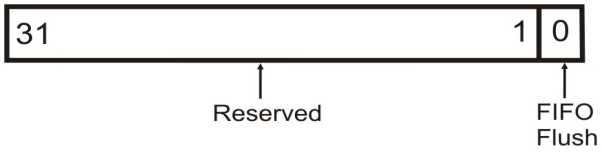


Table 4–6: FIFO Flush Register (Base Address + 0x10)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	fifo_flush	0	R/W	**Input FIFO Flush:** This bit is used to reset the input 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 the output data by programmable (N) number of clock cycles so that if the output data passes through an Interpolation Core, it can be interpolated by N. This register is illus–trated in Figure 4–6 and described in Table 4–7.

Figure 4–6: Output Rate Divider Register

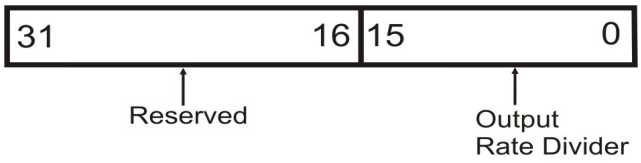


Table 4–7: Output Rate Divider Register (Base Address + 0x14)

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.

4.7 Status Register

The Status Register indicates the mode of operation of the core, data mode selected, status of the Input FIFO, and the status of the Trigger Control State Machine. This register is illustrated in Figure 4–7 and described in Table 4–8.

Figure 4–7: Status Register

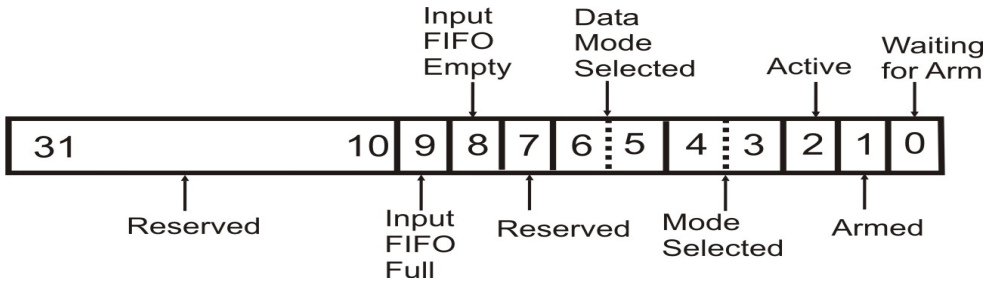


Table 4–8: Status Register (Base Address + 0x20)

Bits	Field Name	Default Value	Access Type	Description
31:10	Reserved	N/A	N/A	**Reserved**
9	fifo_full	0	R	**Input FIFO Full:** This bit indicates that the input FIFO of the Data Flow Control Core is full. 0 = FIFO not full 1 = FIFO full
8	fifo_empty	0	R	**Input FIFO Empty:** This bit indicates that the input FIFO of the AXI4–Stream DAC Data Flow Control Core is empty. 0 = FIFO not empty 1 = FIFO empty
7	Reserved	N/A	N/A	**Reserved**
6:5	data_mode_sel	00	R	**Data Mode Selected:** These bits indicates the data mode selected. **00 = Packed Real Data** **01 = Packed I/Q Data or two–channel Real data** **10 = Reserved** **11 = Reserved**

Table 4–8: 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

4.8 FIFO Count Register

This is a status register that indicates the dword (16–bit) count of the number of dwords in the input FIFO of the Data Flow Control Core. This register is illustrated in Figure 4–7 and described in Table 4–8.

Figure 4–8: FIFO Count Register

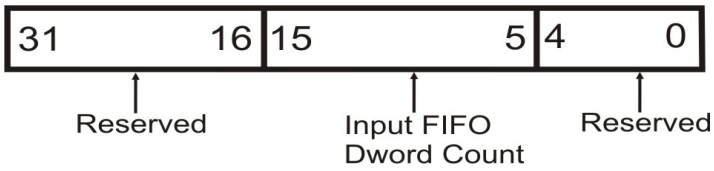


Table 4–9: FIFO Count Register (Base Address + 0x24)

Bits	Field Name	Default Value	Access Type	Description
31:16	Reserved	N/A	N/A	**Reserved**
15:5	fifo_count	0x0000	R	**Input FIFO Dword Count:** These bits indicate the number of dwords (16–bit words) of data currently present in the input FIFO of the Data Flow Control Core.
4:0	Reserved	N/A	N/A	**Reserved**

4.9 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 4.10). This register is illustrated in Figure 4–9 and described in Table 4–10.

Figure 4–9: Interrupt Enable Register

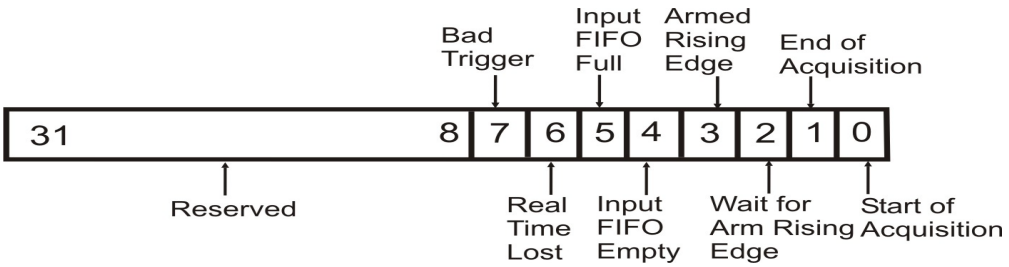


Table 4–10: Interrupt Enable Register (Base Address + 0x34)

Bits	Field Name	Default Value	Access Type	Description
31:8	Reserved	N/A	N/A	**Reserved**
7	bad_trigger	0	R/W	**Bad Trigger:** This bit enables or disables the bad trigger interrupt source. The bad trigger interrupt source indicates that a trigger has been generated before the completion of the last trigger period. 0 = Disable interrupt 1 = Enable interrupt
6	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
5	fifo_full	0	R/W	**Input FIFO Full:** This bit enables or disables the input FIFO full interrupt source. 0 = Disable interrupt 1 = Enable interrupt
4	fifo_empty	0	R/W	**Input FIFO Empty:** This bit enables or disables the input FIFO empty interrupt source. 0 = Disable interrupt 1 = Enable interrupt

Table 4–10: Interrupt Enable Register (Base Address + 0x34) (Continued)

Bits	Field Name	Default Value	Access Type	Description
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

4.10 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 4–10 and described in Table 4–11.

Figure 4–10: Interrupt Status Register

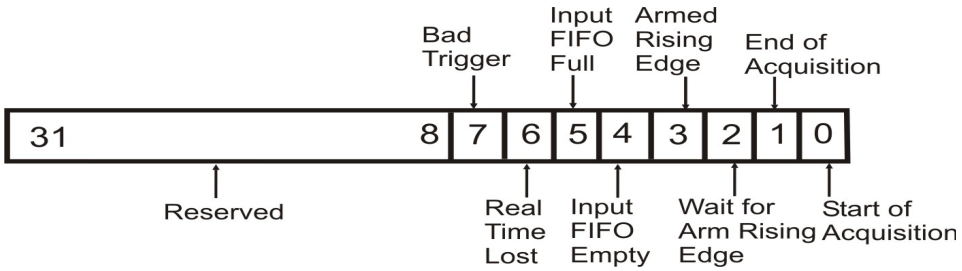


Table 4–11: Interrupt Status Register (Base Address + 0x38)

Bits	Field Name	Default Value	Access Type	Description
31:8	Reserved	N/A	N/A	**Reserved**
7	bad_trigger	0	R	**Bad Trigger:** This bit indicates the status of the bad trigger interrupt source. The bad trigger interrupt source indicates that a trigger has been generated before the completion of the last trigger period. 0 = No interrupt 1 = Interrupt condition asserted
6	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

Table 4–11: Interrupt Status Register (Base Address + 0x38) (Continued)

Bits	Field Name	Default Value	Access Type	Description
5	fifo_full	0	R	**Input FIFO Full:** This bit indicates the status of the input FIFO full interrupt source. 0 = No interrupt 1 = Interrupt condition asserted
4	fifo_empty	0	R	**Input FIFO Empty:** This bit indicates the status of the input FIFO empty interrupt source. 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

4.11 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 4–11 and described in Table 4–12.

Figure 4–11: Interrupt Flag Register

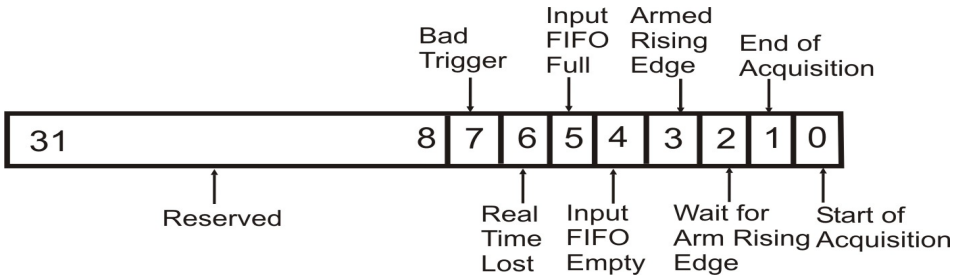


Table 4–12: Interrupt Flag Register (Base Address + 0x3C)

Bits	Field Name	Default Value	Access Type	Description
31:8	Reserved	N/A	N/A	**Reserved**
7	bad_trigger	0	R	**Bad Trigger:** This bit indicates the bad trigger interrupt flag. The bad trigger interrupt source indicates that a trigger has been generated before the completion of the last trigger period. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
6	realtime_lost	0	R	**Real Time Lost:** This bit indicates the real time lost interrupt flag. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch

Table 4–12: Interrupt Flag Register (Base Address + 0x3C) (Continued)

Bits	Field Name	Default Value	Access Type	Description
5	fifo_full	0	RClr	**Input FIFO Full:** This bit indicates the input FIFO full interrupt flag. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
4	fifo_empty	0	R/Clr	**Input FIFO Empty:** This bit indicates the input FIFO empty interrupt flag. **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
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: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the AXI4–Stream DAC Data Flow Control Core.

5.1 General Design Guidelines

The AXI4–Stream DAC Data Flow Control Core provides the required logic to control the input DAC data flow and generate packed output data streams. This IP core supports AXI4–Lite and AXI4–Stream user interfaces. The user can control the Trigger Control State Machine to generate the desired output by setting the required values of the control registers as described in Chapter 4.

5.2 Clocking

AXI4–Stream Clock: **s_axis_aclk**

This clock is used to clock all ports in the Data Flow Control Core.

CSR Clock: **s_axi_csr_aclk**

This clock is used to clock the input AXI4–Lite Interface of the core.

5.3 Resets

Main reset: **s_axis_aresetn**

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

CSR Reset: **s_axi_csr_aresetn**

This is an active low reset synchronous with **s_axi_csr_clk**. When asserted, the control/status registers and the interrupt registers are reset.

5.4 Interrupts

This core has an edge–type (rising edge–triggered) interrupt output. It is synchronous with the **s_axi_csr_aclk**. On the rising edge of any interrupt signal, a one–clock–cycle–wide pulse is output from the core on it **irq** output. Each interrupt event is stored in two registers, accessible on the **s_axi_csr** bus.

5.4 Interrupts (continued)

The Interrupt Status Register always reflects the current state of the interrupt condition, which may have changed since the generation of the interrupt. The Interrupt Flag Register latches the occurrence of each interrupt, in a bit that retains its state until explicitly cleared. The Interrupt flags can be cleared by writing '1' to the associated bit's location. All interrupt sources that are enabled (via the Interrupt Enable Register) are "OR ed" onto the **irq** output.

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

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

5.5 Interface Operation

CSR Interface: This is the Control/Status Register Interface and is associated with **s_axi_csr_aclk**. It is a standard AXI4–Lite Slave Interface. See Chapter 4 for the control register memory map, which provides more details on the registers that can be accessed through this interface.

Data Stream (PD) Interface: This core implements an AXI4–Stream Slave Interface across the input to receive AXI data streams and is associated with **s_axis_aclk**. For more details about this interface, refer to Section 3.2.1.

Timing Events (PTCTL) Interface: This is the interface through which timing events data is received across the input ports and is associated with **s_axis_aclk**. For more details about this interface, refer to Section 3.2.2

Combined Sample Data/ Timestamp/ Information Streams (PDTI) Interface: This core implements an AXI4–Stream Master Interface across the output to transfer AXI PDTI streams and is associated with **s_axis_aclk**. For more details about this interface, refer to Section 3.2.3.

5.6 Programming Sequence

This section briefly describes the programming sequence of registers in the Data Flow Control Core.

1) Ensure that the Interrupt Flag Register is cleared.

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

3) Write the desired values to the control registers.

4) Observe the outputs across the outputs ports.

5) When done, check the Interrupt Flag Register and clear the interrupts.

5.7 Timing Diagrams

The timing diagram for the Data Flow Control Core is shown in Figure 6–3. 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 6.5.

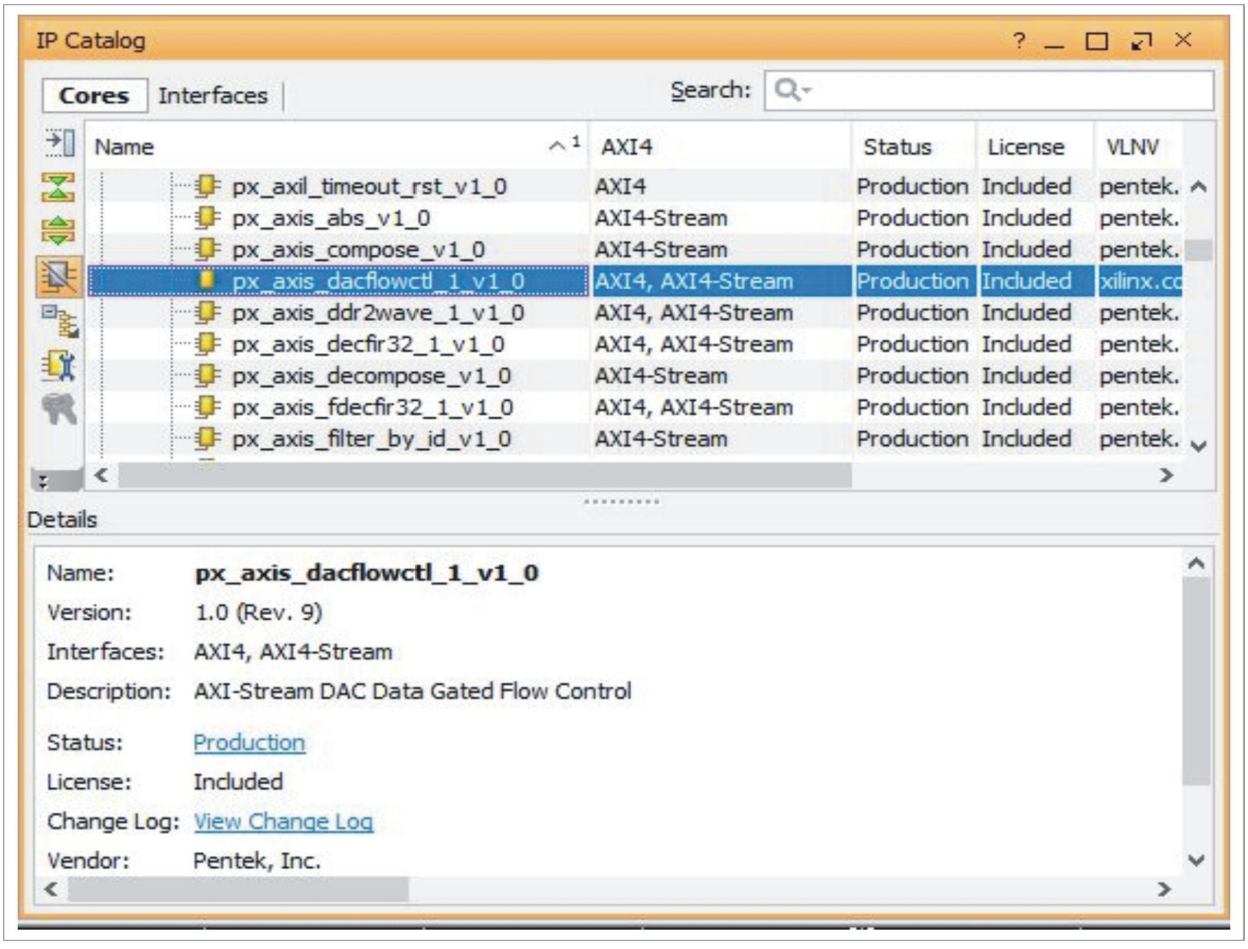
This page is intentionally blank

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek AXI4–Stream DAC Data Flow Control 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_dacflowctl_1_v1_0** as shown in Figure 6–1.

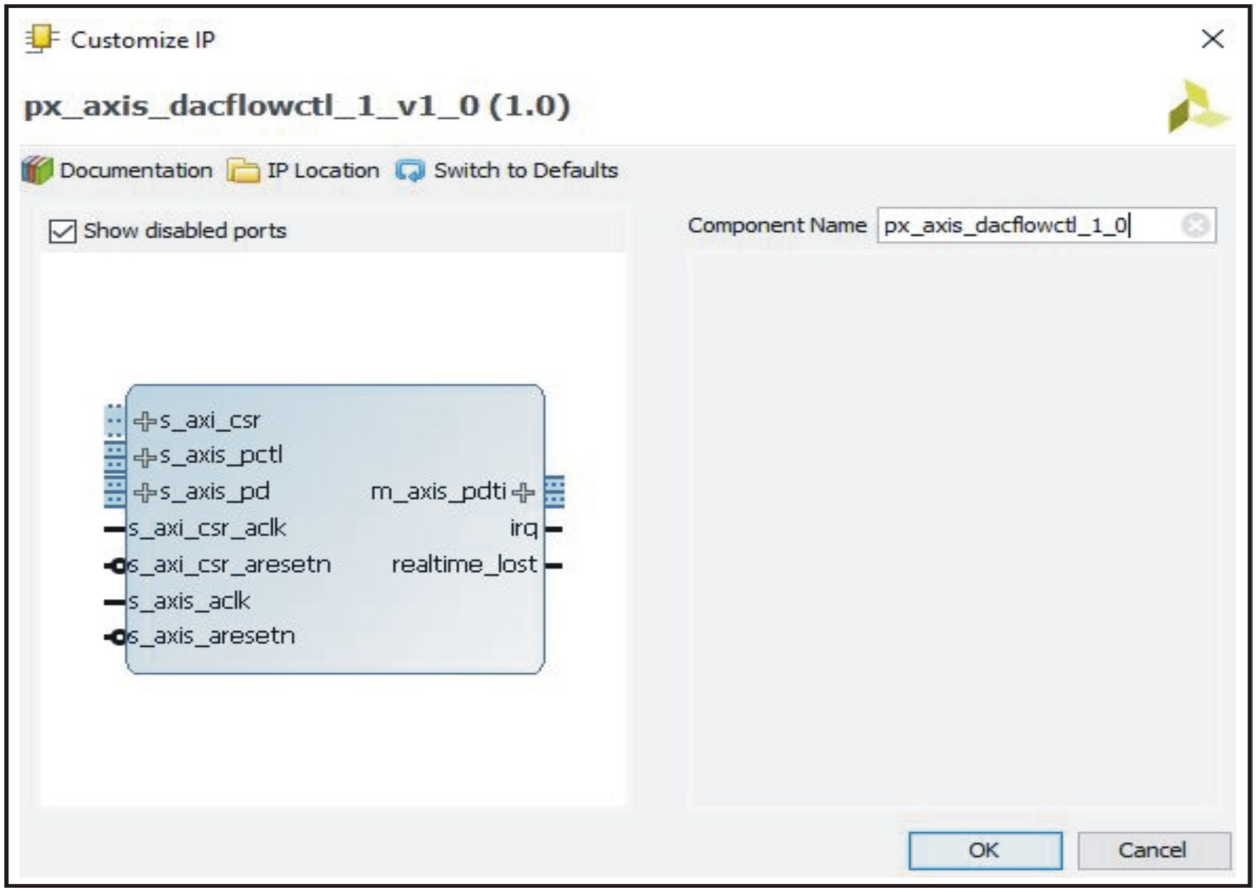
Figure 6–1: AXI4–Stream DAC Data Flow Control Core in Pentek IP Catalog



6.1 Pentek IP Catalog (continued)

When you select the **px_axis_dacflowctl_1_v1_0** core, a screen appears that shows the core's symbol and the core's parameters (see Figure 6–2). The core's symbol is the box on the left side.

Figure 6–2: AXI4–Stream DAC Data Flow Control Core IP Symbol



6.2 User Parameters

This section is not applicable to this IP core.

6.3 Generating Output

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

6.4 Constraining the Core

This section contains information about constraining the Data Flow Control Core in Vivado Design Suite.

Required Constraints

The XDC constraints are not provided with the Data Flow Control Core. Clock 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 CSR clock (**s_axi_csr_aclk**) and AXI4−Stream clock (**s_axis_aclk**) can both take frequencies up to 250 MHz.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking and Placement

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

6.5 Simulation

The Data Flow Control Core has a test bench which generates output waveforms using the Vivado VSim environment. The test bench is designed to run at 200 MHz AXI4–Stream clock frequency and 250 MHz CSR clock frequency.

The test bench provides control register values through a **`test_parameters.txt`** file. The parameter defined in the **`test_parameters.txt`** file is described in Table 6–1. The control registers within the core are written with the values from the text file and verified by reading from them.

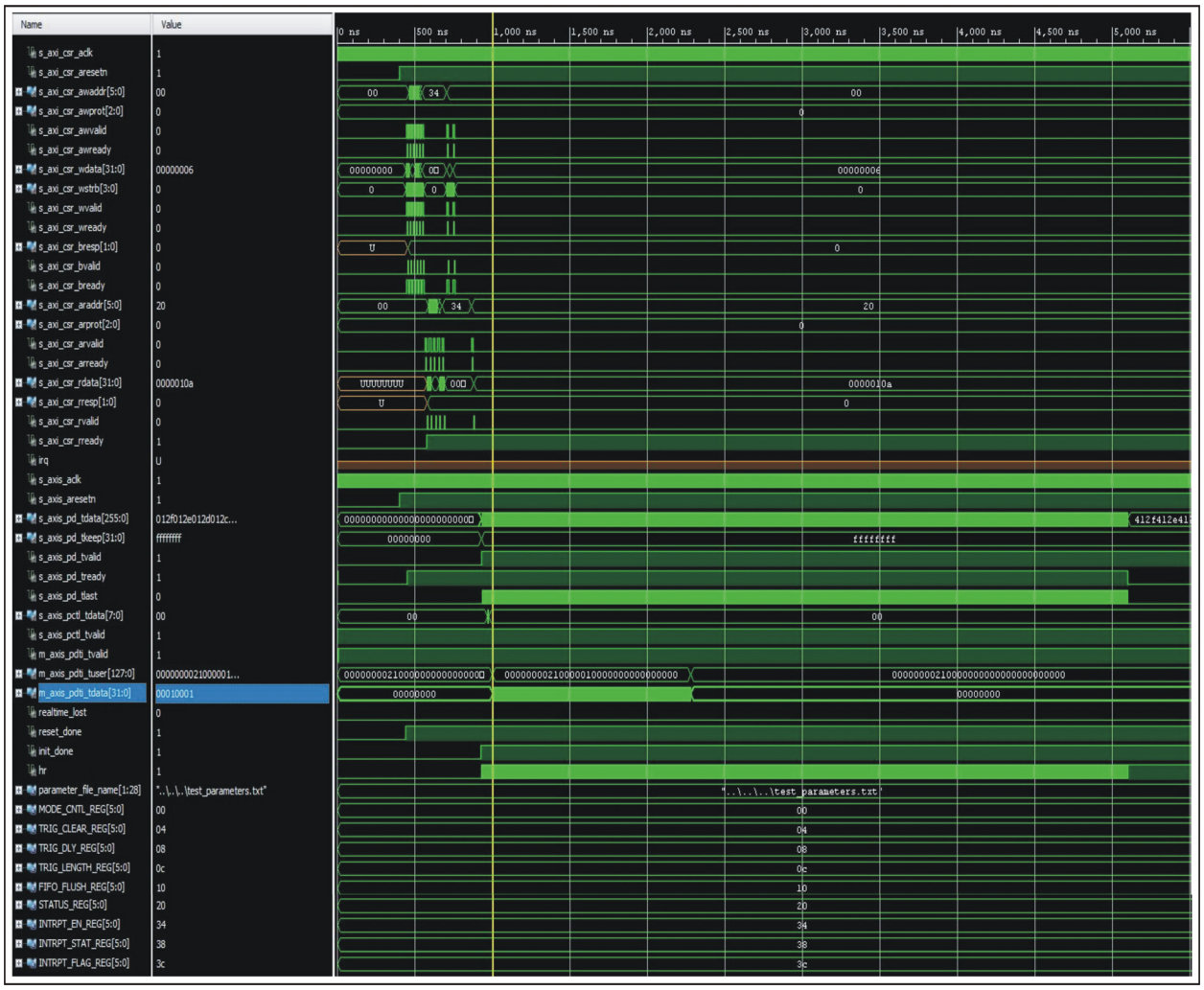
Table 6–1: Test Parameters File Contents and Parameter Descriptions

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 = Reserved**
data_mode_sel		0x0	**Data Mode Select:** This parameter defines the data type of the input data to the Data Flow Control Core. **0 = Real Data** **1 = Packed I/Q Data or two channel real data**
stay_armed	Boolean	True	**Stay Armed:** When set to True, the trigger control state machine is held in the armed state.
trigger_dly_value	std_logic _vector	0x00000000	**Trigger Delay Value:** This is the delay to be introduced to the data acquisition gate signal after a trigger event has occurred.
trigger_len_value		0x00000100	**Trigger Length Value:** This is the length of the data acquisition gate signal generated by the gate/trigger generator module.
output_rate_div		0x00000000	**Output Rate Divider:** This parameter defines the rate divider value of the output data.

6.5 Simulation (continued)

The test bench has the core operating in the Trigger mode and data mode set to Single Sample Real data. Once the trigger control state machine is armed, the input data to the core is generated using an up counter starting from 0x0000. The acquisition gate trigger length is set to 256 AXI4–Stream clock cycles and has no trigger delay. When run, the simulation produces the results shown in Figure 6–3.

Figure 6–3: AXI4–Stream DAC Data Flow Control Core Test Bench Simulation Output



6.6 Synthesis and Implementation

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

This page is intentionally blank