IP CORE MANUAL

DAC38RF89 DAC Interface Core

px_dac38rf89intrfc

PENTEK

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

Manual Part Number: 807.48411 Rev: 1.1 – November 14, 2018

Manual Revision History

Date	Version	Comments
4/13/18	1.0	Initial Release
11/14/18	1.1	Updated Table 2–1.

Legal Notices

Copyright

Trademarks

Printed in the United States of America.

Table of Contents

Table of Contents

Page

Chapter 4: Register Space

Chapter 5: Designing with the Core

Table of Contents

Page

Chapter 6: Design Flow Steps

Table of Contents

Page

This page is intentionally blank

IP Facts

Description

Pentek's Navigator™ DAC38RF89 DAC Interface Core serves as an interface to the Texas Instruments™ DAC38RF89 Digital to Analog Converter.

This core complies with the ARM® AMBA® AXI4 specification. This manual defines the hardware interface, software interface, and parameterization options for the DAC38RF89 DAC Interface Core.

Features

- Register access through AXI4–Lite Interface
- Input/output data through AXI4–Stream Interface
- Requires separate JESD Interface backend for JESD interconnection
- Provides SPI access to DAC38RF89
- Supports DAC38RF89 L–M–F–S–Hd modes:
  82121–1TX,
  84111–2TX,
  81180–1TX,
  82380–2TX,
  82380–2TX–Wide DUC,
  82121–2TX,
  and 82121–2TX–Wide DUC

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	Not Provided[b]
Constraints File	Not Provided[c]
Simulation Model	N/A
Supported S/W Driver	HAL Software Support
Tested Design Flows	
Design Entry	Vivado® Design Suite 2018.3 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.The test bench will be available in the next revision of this core.
c.Clock constraints can be applied at the top level module of the user design.

This page is intentionally blank

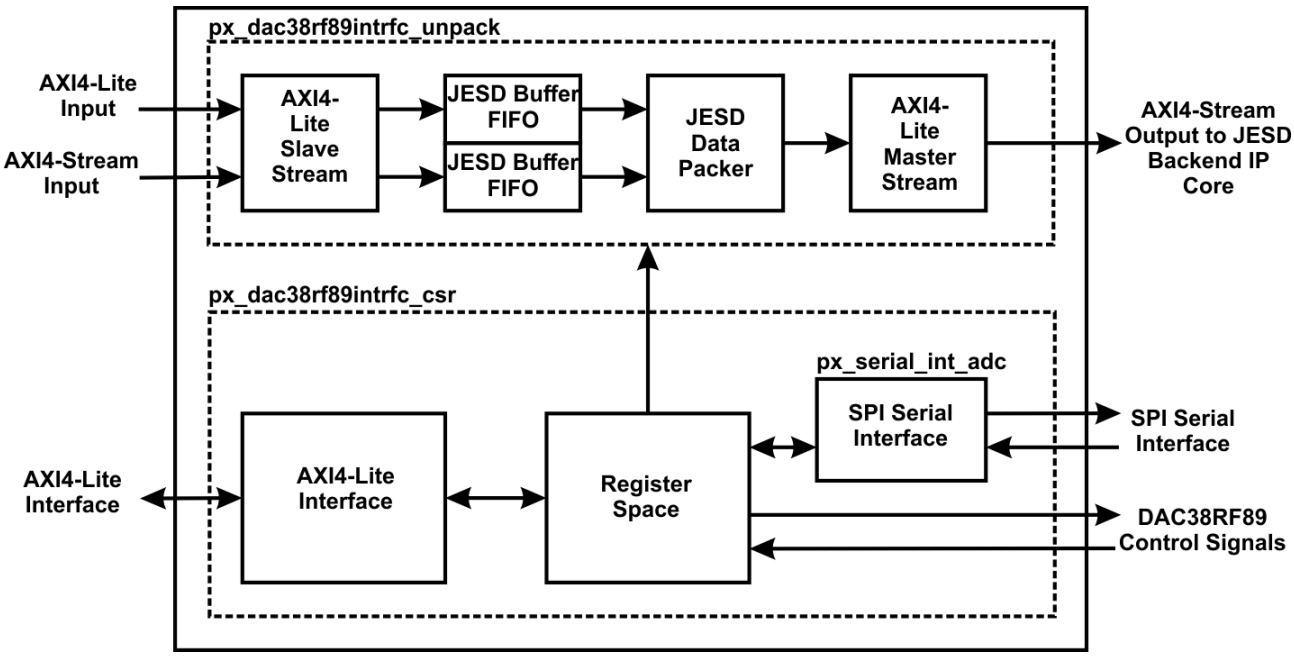
Chapter 1: Overview

1.1 Functional Description

The DAC38RF89 DAC Interface Core is a software controlled interface between Pentek backend acquisition IP Cores to a Xilinx JESD Core to support the Texas Instrument DAC38RF89 Digital to Analog converter device. The core receives DAC data via an input AXI4−Stream slave bus from Pentek IP cores, re−packs and sends the data via an output AXI4−Stream master bus to a Xilinx JESD Core. DAC data is received in accor−dance to Pentek data packing standards and repacked in accordance to the DAC38RF89 L−M−F−S−Hd packing modes. This core supports 8−lane JESD L−M−F−S−Hd modes: 82121−1TX, 84111−2TX, 81180−1TX, 82380−2TX, 82380−2TX−Wide DUC, 82121−2TX, and 82121−2TX−Wide DUC. Refer to Texas Instrument DAC38RF89 datasheet for more details.

Figure 1−1 is the top level block diagram of the DAC38RF89 DAC Interface Core. The modules within the block diagram are explained in the later sections of this manual.

Figure 1−1: DAC38RF89 DAC Interface Core Block Diagram



1.1 Functional Description (continued)

- **PX_DAC38RF89INTRFC_CSR**
  - **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 the Section 3.1.
  - **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.
  - **SPI Interface:** Serial interface to access registers in the external device DAC38RF89.
- **PX_DAC38RF89INTRFC_UNPACK**
  - **AXI4−Stream Slave Interface:** This module receives a 256−bit AXI4−Stream slave bus from Pentek backend acquisition IP Core interfaces which contain DAC data. For additional details about the AXI4−Stream Interface, refer to the Section 3.2.
  - **JESD Buffer FIFO:** FIFO to buffer DAC data before packetizing to JESD backend AXI4−Stream master interface
  - **JESD Data Packer:** Packs data into the appropriate order based on the DAC38RF89 **`L-M-F-S-Hd`** mode of operation
  - **AXI4−Stream Master Interface:** This module implements a 256−bit AXI4−Stream Master interface for DAC data output. For additional details about the AXI4−Stream Interface, refer to the Section 3.2.

1.2 Applications

This core can be used as an interface to the Texas Instruments DAC38RF89 Digital to Analog Converter via a Xilinx JESD interface core.

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
4) *Texas Instruments DAC38RF89 Digital to Analog Converter Datasheet*
5) Xilinx JESD204 Core PG066
6) Xilinx JESD204 PHY PG198

This page is intentionally blank

Chapter 2: General Product Specifications

2.1 Standards

The DAC38RF89 DAC Interface 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 DAC Core is limited by the maximum operating frequency of the DAC38RF89 Digital to Analog Converter. The values presented in this section should be used as an estimation guideline. Actual performance can vary.

2.2.1 Maximum Frequencies

The DAC38RF89 DAC has a maximum operating frequency of 6.66 GHz. However, the maximum operation frequency the Core can operate at is 6.4 GHz. Thus, the maximum JESD line rate at 6.4 GHz is 12.0 Gbps (See the DAC38RF89 datasheet for line rate calculations). The DAC Core is therefore designed to run at a maximum clock frequency of 300 MHz in the JESD interface clock domain and 400 Mhz in the sample clock domain.

2.3 Resource Utilization

The resource utilization of the DAC38RF89 DAC Interface 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	2,784
Flip-Flops	6,667
LUTRAM	2

NOTE: Actual utilization may vary based on the user design in which the DAC38RF89 DAC Interface Core is incorporated.

2.4 Limitations and Unsupported Features

The DAC38RF89 DAC Interface Core only supports 8−lane JESD L−M−F−S−Hd modes 82121−1TX, 84111−2TX, 81180−1TX, 82380−2TX, 82380−2TX Wide DUC, 82121−2TX and 82121−2TX Wide DUC. See the DAC38RF89 data sheet for more details.

2.5 Generic Parameters

There are no generic parameters for 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
- I/O Signals

3.1 AXI4–Lite Core Interfaces

The DAC38RF89 DAC Interface 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 DAC 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 value will reset all control registers to their initial states.
s_axi_csr_awaddr	Input	7	**Write Address:** Address used for write operations. It must be valid when **s_axi_csr_awvalid** is asserted and must be held until **s_axi_csr_awready** is asserted by the DAC Core.
s_axi_csr_awprot	Input	3	**Protection:** The DAC interface 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 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 DAC Core when it is ready to accept the write address.The address is latched when **s_axi_csr_awvalid** and **s_axi_csr_awready** are high on the same cycle.
s_axi_csr_wdata	Input	32	**Write Data:** This data will be written to the address specified by **s_axi_csr_awaddr** when **s_axi_csr_wvalid** and **s_axi_csr_ wready** are both asserted. The value must be valid when **s_axi_csr_wvalid** is asserted and held until **s_axi_csr_wready** is also asserted.
s_axi_csr_wstrb	Input	4	**Write Strobes:** This signal when asserted indicates the number of bytes of valid data on **s_axi_csr_wdata** signal. Each of these bits, when asserted indicate that the corre–sponding byte of **s_axi_csr_wdata** contains valid data. Bit 0 corresponds to the least significant byte, and bit 3 to the most significant.
s_axi_csr_wvalid	Input	1	**Write Valid:** This signal must be asserted to indicate that the write data is valid for a write operation. The value on **s_axi_csr_wdata** is written into the register at address **s_axi_ csr_awaddr** when **s_axi_csr_wready** and **s_axi_csr_wvalid** are high on the same cycle.
s_axi_csr_wready	Output		**Write Ready:** This signal is asserted by the DAC Core when it is ready to accept data. The value on **s_axi_csr_wdata** is written into the register at address **s_axi_csr_awaddr** when **s_axi_csr_wready** and **s_axi_csr_wvalid** are high on the same cycle, assuming that the address has already or simultaneously been submitted.
s_axi_csr_bresp	Output	2	**Write Response:** The core indicates success or failure of a write transaction through this signal, which is valid when **s_axi_csr_ bvalid** is asserted; 00 = Success of normal access 01 = Success of exclusive access 10 = Slave Error 11 = Decode Error Note: For more details about this signal refer to the *AMBA AXI Specification*.
s_axi_csr_bready	Input	1	**Write Response Ready:** This signal must be asserted by the user logic when it is ready to accept the Write Response.
s_axi_csr_bvalid	Output	1	**Write Response Valid:** This signal is asserted by the DAC Core when the write operation is complete and the Write Response is valid. It is held until **s_axi_csr_bready** is asserted by the user logic.

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

Port	Direction	Width	Description
s_axi_csr_araddr	Input	7	**Read Address:** Address used for read operations. It must be valid when **s_axi_csr_arvalid** is asserted and must be held until **s_axi_csr_arready** is asserted by the DAC Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the DAC 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 DAC 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 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 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 Core when the read is complete and the read data is available on the **s_axi_csr_rdata**. It is held until **s_axi_csr_rready** is asserted by the user logic.
s_axi_csr_rready	Input		**Read Data Ready:** This signal is asserted by the user logic when it is ready to accept the Read Data.
irq	Output		**Interrupt:** This is an active High, edge–type interrupt output.

3.2 AXI4–Stream Core Interfaces

The DAC38RF89 DAC Interface Core has the following AXI4–Stream Interface, which is used to transfer data streams.

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

Pentek's Jade series board products have AXI4–Streams that follow a combined Sample Data/ Timestamp/ Information Stream format. This type of data stream combines sample data with its time aligned timestamp and data information. This core has an AXI4–Stream Slave Interface across the input to receive AXI4–Streams.

Table 3–2 defines the ports in the AXI4–Stream Combined Sample Data/ Timestamp/ Information Interface. See the *AMBA AXI4–Stream Specification* for more details on the operation of the AXI4–Stream Interface.

Table 3–2: Combined Sample Data/ Timestamp/ Information Stream Interface Port Descriptions

Port	Direction	Width	Description
s_axis_pdti_a_tdata	Input	256	**Input Data:** From Data Acquisition Channel A. Channel A is used for both single and dual channel DAC operation. See section 5.5 for more details.
s_axis_pdti_a_tvalid	Input	1	**Input Data Valid:** Asserted when data is valid on the **s_axis_pdti_a_tdata**. Data is also qualified when **s_axis_pdti_a_tuser[64]**, which corresponds to gate, is active. Note: For Single Channel operation: **s_axis_pdti_a_tvalid** must operate at full rate For Dual Channel operation: **s_axis_pdti_a_tvalid** must operate at half rate See section 5.5 for more details.

Table 3–2: Combined Sample Data/ Timestamp/ Information Stream Interface Port Descriptions (Continued)

Port	Direction	Width	Description
s_axis_pdti_a_tuser	Input	128	**Sideband User Information:** This is the user defined sideband information transmitted 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 [92:88] – Samples per clock cycle** **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.
s_axis_pdti_b_pd_tdata	Input	256	**Input Data:** From Data Acquisition Channel B. Channel B only applies for dual channel DAC operation. See section 5.5 for more details.
s_axis_pdti_b_tvalid	Input	1	**Input Data Valid:** Asserted when data is valid on the **s_axis_pdti_b_tdata**. Data is also qualified when **s_axis_pdti_b_tuser[64]**, which corresponds to gate, is active. Note: For Single Channel operation: **s_axis_pdti_b_tvalid** is ignored For Dual Channel operation: **s_axis_pdti_b_tvalid** must operate at half rate See section 5.5 for further details.

Table 3–2: Combined Sample Data/ Timestamp/ Information Stream Interface Port Descriptions (Continued)

Port	Direction	Width	Description
s_axis_pdti_b_tuser	Input	128	**Sideband User Information:** This is the user defined sideband information transmitted 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 [92:88] – Samples per clock cycle** **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.
m_axis_jesd_tx_tdata	Output	256	**Output Data:** Output data to JESD interface formatted to the Texas Instruments DAC38RF89 Digital to Analog.
m_axis_jesd_tx_tvalid	Output	1	**Data Valid:** Data valid signal to JESD core
m_axis_jesd_tx_tready	Input	1	**Ready to Recieve Data:** Ready for data signal from JESD core

3.3 I/O Signals

The I/O port/signal descriptions of the top level module of the DAC38RF89 DAC Interface Core are provided in Table 3–3..

Table 3–3: I/O Signals

Port/Signal Name	Type	Direction	Description
Clock Signals			
tx_core_clk	std_logic	Input	**JESD Transmitter Core Clock:** This clock operates at a frequency equal to (JESD Serdes line rate)/40. Refer to Xilinx program guide PG066 - JESD204 V7.1 and PG198 - JESD204 PHY V3.3 for details. See section 5.2 for clocking details.
dac_sample_clk	std_logic	Input	**Sample Clock:** FPGA sample clock rate See section 5.2 for clocking details.
tx_start_of_frame	std_logic_ vector	Input	**JESD Frame Boundary Indication:** The signal is 4 bits to indicate the byte position of the first byte of a frame in **tdata** in the following Clock cycle. This mode is used by the DAC38RF89 DAC Interface Core in 12-bit mode.
Control Signals For DAC38RF89			
dac_reset_n	std_logic	Output	**Reset:** Active low input for chip **RESET**, which resets all the programming registers to their default state. Internal pull-up.
dac_txena	std_logic	Output	**Transmit Enable:** This signal must be active High
dac_sleep	std_logic	Output	**Sleep:** Active high asynchronous hardware power-down input. Internal pull-down.
dac_alarm	std_logic	Input	**Alarm Condition**
Control Signals For SPI Interface to DAC38RF89			
dac_sout	std_logic	Input	**SPI Interface output line from ADC:** Readback data is read from this line.
dac_sdi	std_logic	Output	**SPI Interface input line to DAC38RF89 device:** Address and write data are provided on this line to ADC.
dac_scs_n	std_logic	Output	**Chip Select:** Active Low.
dac_sclk	std_logic	Output	**Serial Clock Line:** Low Frequency

This page is intentionally blank

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the register space of the DAC38RF89 DAC Interface Core. The memory map is provided in Table 4–1.

Table 4–1: Register Space Memory Map

Register Name	Address (Base Address +) (0x0000 +)	Access	Description
Control Registers			
DAC Control Register	0x00	R/W	Control Registers for the DAC38RF89 DAC Interface IP Core and DAC38RF89 external signals
DAC SPI Interface Control Register	0x04	R/W	DAC SPI Interface Control Register
Status Registers			
DAC Control Status Register	0x08	RO	Status Register for DAC Interface.
DAC SPI Interface Status Register	0x0C	RO	Readback of DAC SPI Interface
Interrupt Enable/Status/Flag Registers			
Interrupt Enable Register	0x10	R/W	Interrupt enable bits
Interrupt Status Register	0x14	R	Interrupt status bits
Interrupt Flag Register	0x18	R/Clr	Interrupt flag bits

4.1 DAC Control Register

This register is used to control the DAC38RF89 DAC Interface Core modes of operation and the external control signals of the DAC38RF89 external device. This register is illustrated in Figure 4–1 and described in Table 4–2.

Figure 4–1: DAC Control Register

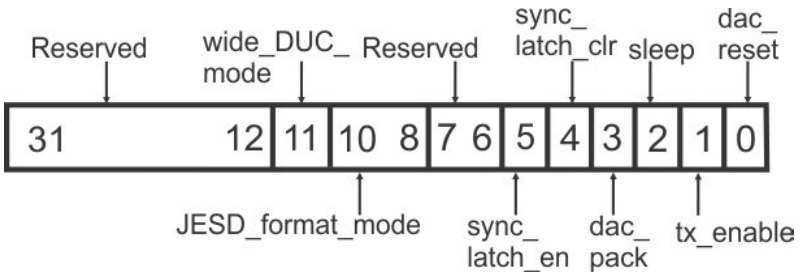


Table 4–2: DAC Control Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:12	Reserved	N/A	N/A	**Reserved**
11	Wide_DUC_ mode	0	R/W	**Wide DUC Mode:** Set '1' to enable the DAC38RF89 DAC Interface Core to support Wide DUC mode for 82380–2TX Wide DUC or the 82121–2TX Wide DUC.
10:8	JESD_format _model	000	R/W	**JESD Format Mode:** This register sets the pack mode for the DAC38RF89 DAC Interface Core based on the L–M–F–S–Hd Mode of operation of the DAC38RF89 device. See the data sheet for details. 000 – LMFSHd 82121 1–TX 001 – LMFSHd 84111 2–TX 010 – Reserved 011 – LMFSHd 81180 1–TX 100 – LMFSHd 82380 1–TX 101 – LMFSHd 82121 2–TX **Note:** The DAC38RF89 device requires configuration.
7:6	Reserved	N/A	N/A	**Reserved**
5	sync_latch_en	0	R/W	**Sync Latch Enable:** Enable the Sync latch to be set when sync is received.
4	sync_latch_clr	0	R/W	**Sync Latch Clear:** Clear Sync Latch in the Pack interface.
3	dac_pack_rst	0	R/W	**DAC Pack Reset:** resets the JESD data packer interface and FIFO of the DAC38RF89 DAC Interface IP core.
2	Sleep	0	R/W	**Sleep:** Active high asynchronous hardware power–down input
1	tx_enable	0	R/W	**Transmit Enable:** Active High input
0	dac_reset	0	R/W	**DAC Chip RESET**

4.2 DAC SPI Interface Control Register

This register is used to control the SPI interface to the DAC38RF89 Digital to Analog device. This register is illustrated in Figure 4–2 and described in Table 4–3.

NOTE: Verify SPI interface is ready at DAC SPI Status register (Base + 0x0C) bit−16; is set to '1'.

Figure 4–2: DAC SPI Interface Control Register

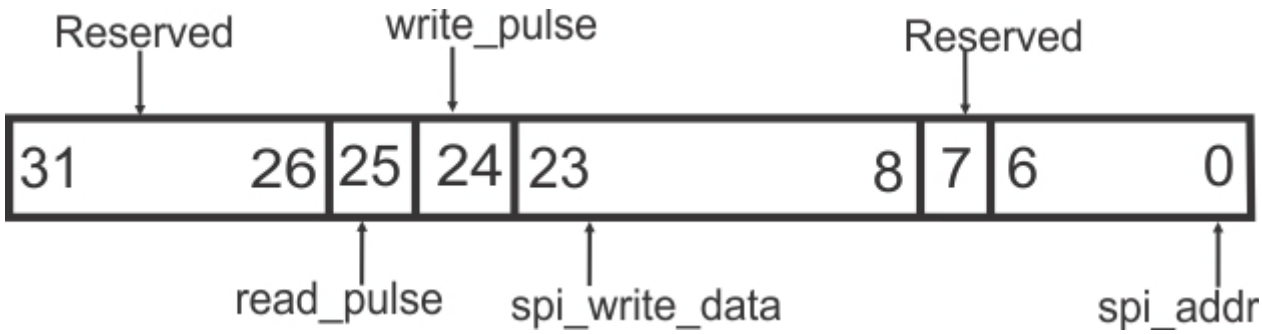


Table 4–3: DAC SPI Interface Control Register (Base Address + 0x04)

Bits	Field Name	Default Value	Access Type	Description
31:26	Reserved	N/A	N/A	**Reserved**
25	read_pulse	0	R/W	**Read Pulse:** Toggle Register to initiate a SPI interface read. **See note above.**
24	write_pulse	0	R/W	**Write Pulse:** Toggle Register to initiate a SPI interface write. **See note above.**
23:8	spi_write_data	0	R/W	**SPI Write Data**
7	Reserved	N/A	N/A	**Reserved**
6:0	spi_addr	0	R/W	**SPI Address**

4.3 DAC Control Status Register

This register provides the status of the DAC38RF89 DAC. This register is illustrated in Figure 4–3 and described in Table 4–4.

Figure 4–3: DAC Control Status Register

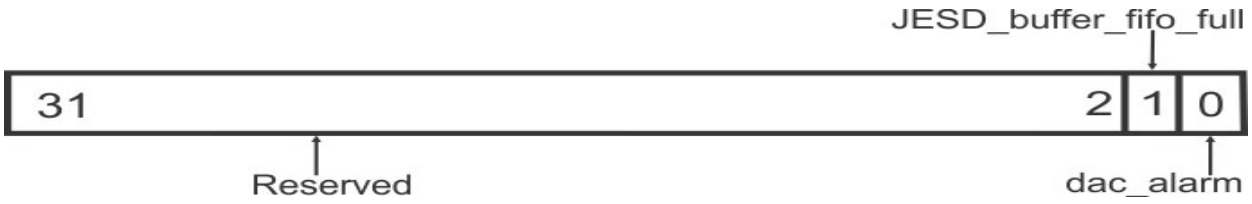


Table 4–4: DAC Control Status Register (Base Address + 0x08)

Bits	Field Name	Default Value	Access Type	Description
31:2	Reserved	N/A	N/A	**Reserved**
1	JESD_buffer_fifo_full	0	RO	**JESD Buffer FIFO Full**
0	dac_alarm	0	RO	**DAC Alarm Signal:** from external DAC38RF89

4.4 DAC SPI Interface Status Register

This register provides the readback from the DAC38RF89 SPI Interface. See the DAC38RF89 Datasheet for register definitions. This register is illustrated in Figure 4–4 and described in Table 4–5.

NOTE: Verify SPI Ready has returned a '1' after initiating a SPI Read pulse before accessing SPI Read Data register

Figure 4–4: DAC SPI Interface Status Register

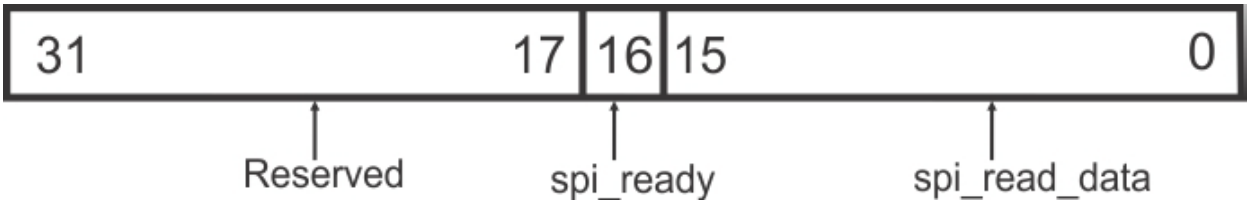


Table 4–5: DAC SPI Interface Status Register (Base Address + 0x0C)

Bits	Field Name	Default Value	Access Type	Description
31:17	Reserved	–	–	**Reserved**
16	spi_ready	1	RO	**SPI Interface Ready:** When '1', interface is ready for a read or write.
15:0	spi_read_data	0	RO	**SPI Read Data:** Readback data from Address is provided at (Base + 0x04).

4.5 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 4.6). This register is illustrated in Figure 4–5 and the bits are described in Table 4–6.

Figure 4–5: Interrupt Enable Register

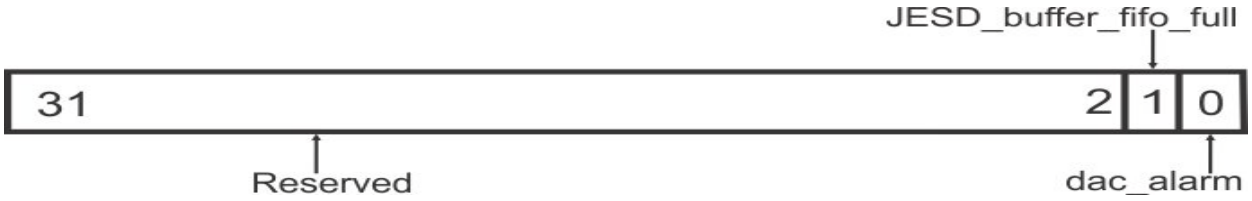


Table 4–6: Interrupt Enable Register (Base Address + 0x10)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
1	JESD_buffer_ fifo_full	0	RO	**JESD Buffer Packing FIFO Full**
0	dac_alarm	0	R/W	**Alarm Signal:** from external DAC38RF89

4.6 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 4−6 and the bits are described in Table 4−7.

Figure 4−6: Interrupt Status Register

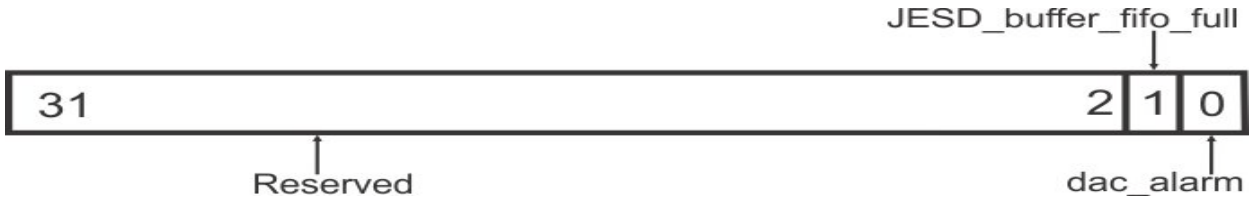


Table 4−7: Interrupt Status Register (Base Address + 0x14)

Bits	Field Name	Default Value	Access Type	Description
31:2	Reserved	N/A	N/A	**Reserved**
1	JESD_buffer_ fifo_full	0	RO	**JESD Buffer Packing FIFO Full**
0	dac_alarm	–	RO	**Alarm Signal:** from external DAC38RF89

4.7 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 4–7 and the bits are described in Table 4–8.

Figure 4–7: Interrupt Flag Register

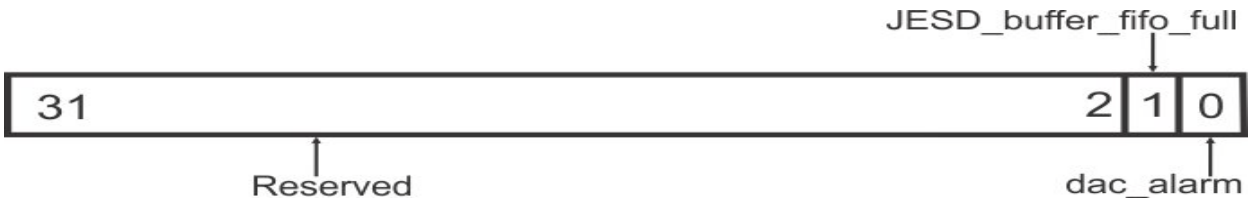


Table 4–8: Interrupt Flag Register (Base Address + 0x18)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
1	JESD_buffer_ fifo_full	0	RO	**JESD Buffer Packing FIFO Full**
0	dac_alarm	–	R/CLR	**Reserved**

Chapter 5: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the DAC38RF89 DAC Interface Core.

5.1 General Design Guidelines

The DAC38RF89 DAC Interface Core is is designed to interface to a Xilinx JESD Back−end interface to transmit data to the Texas Instrument DAC38RF89 digital to analog converter device. This core also provides a Serial Interface to access the registers within the DAC.

5.2 Clocking

AXI4−Lite Clock: `s_axi_csr_aclk`

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

Core Clock: `tx_core_clk`

This is the core clock for the JESD Transmitter interface. The frequency is dependant on the operating line rate of the DAC38RF89 device based on the **`L-M-F-S-Hd`** mode. According to the Xilinx Product Guide PG066, **`tx_core_clk`** frequency = **`line_rate/40`**. Refer to DAC38RF89 datasheet to calculate serdes line rate.

DAC Sample Clock: `dac_sample_clk`

Sample Clock Domain: For Jmode "100", corresponding to LMFSHd 82380, **`dac_sample_clk`** = **`tx_core_clk`** * (4/3).

For all other Jmodes, **`dac_sample_clk`** = **`tx_core_clk`**

5.2 Clocking (continued)

The table below indentifies is a summary of the required clock ratio of **dac_sample_clk** to **tx_core_clk** based on the **L-M-F-S-Hd** mode.

Table 5-1: Required Ratio of the DAC Sample Clock to the TX Core Clock

JMode Addr 0x00 Bits 10:8	LMFSHd	# of Channels	Input Resolution	Interpolation	Line Rate	tx_core_clk	dac_sample_clk
"000"	82121-1TX	Single	16	6	7500	187.5	187.5
				8	5625	140.625	140.625
				12	3750	93.75	93.75
				16	2812.5	70.3125	70.3125
"001"	84111-2TX	Dual	16	6	12500	312.5	312.5
				8	11250	281.25	281.25
				10	9000	225	225
				12	7500	187.5	187.5
				16	5625	140.625	140.625
				18	5000	125	125
"010"	Reserved	-	-	-	-	-	-
"011"	81180-1TX	Single	8	1	11250	281.25	281.25
"100"	82380-2TX	Dual	12	1	12000	300	400
				2			
"100" - Wide DUC[a]	82380-2TX	Single	12	1	12000	300	400
				2			
"101"	82121-2TX	Dual	16	1	12500	312.5	312.5
				2			
				4	11250	281.25	281.25
"100" - Wide DUC[a]	82121-2TX	Single	16	1	12500	312.5	312.5
				2			
				3	11250	281.25	281.25

a. To enable Wide DUC mode, bit–11 at address 0x00 must be set to '1'

5.3 Resets

Main reset: s_axi_csr_aresetn

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

Pack Reset: dac_pack_rst at addr 0x00 bit–3

Reset for the JESD packer interface. This reset should be applied anytime the JESD interface is initialized.

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 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 "ORed" onto the **irq** output.

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

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

5.5 Interface Operation

Control/Status Register Interface: This is the control/status register Interface. It is associated with the **s_axi_csr_aclk**. It is a standard AXI4−Lite type interface. See Chapter 5 for the control register memory map, for more details on the registers that can be accessed through this interface.

Combined Sample Data/ Timestamp/ Information Streams (PDTI) Interface: The DAC38RF89 DAC Interface Core receives DAC data from two separate DAC acquisition channels as input from AXI4−Streams. It is a standard AXI4−Stream Slave Interface. For more details about this interface refer to Section 3.2.1.

In order to ensure proper data flow through the DAC38RF89 DAC Interface Core, the data valids must operate at full rate or half rate based on Single channel or dual channel operation.

Single channel modes: Data is received on **s_axis_pdti_a_tdata**. **s_axis_pdti_a_tvalid** must operate at full rate.

Dual Channel modes: Data is received on **s_axis_pdti_a_tdata** and **s_axis_pdti_b_tdata**. **s_axis_pdti_a_tvalid** and **s_axis_pdti_b_tvalid** must operate at half rate.

The table below summarizes the data valid rates.

Table 5-2: Required Ratio of the DAC Sample Clock to the TX Core Clock

JMode Addr 0x00 Bits 10:8	LMFSHd	# of Channels	s_axis_pdti_a_tvalid Rate	s_axis_pdti_b_tvalid Rate
"000"	82121-1TX	Single	Full	N/A
"001"	84111-2TX	Dual	Half	Half
"010"	Reserved	-	-	-
"011"	81180-1TX	Single	Full	N/A
"100"	82380-2TX	Dual	Half	Half
"100" - Wide DUC	82380-2TX	Single	Full	N/A
"101"	82121-2TX	Dual	Half	Half
"100" - Wide DUC	82121-2TX	Single	Full	N/A

JESD Backend Data Bus: The DAC38RF89 DAC Interface Core outputs DAC data via a standard AXI4−Stream master interface to a backend interface of a Xilinx JESD Core. The DAC38RF89 DAC Interface Core packs DAC data into the master AXI4−Stream depending on the L−M−F−S−Hd mode of the DAC38RF89. L−M−F−S−Hd modes that are supported by this core are :

- 82121−1TX − Single Channel 16−bit I/Q
- 84111−2TX − Dual Channel 16−bit I/Q
- 81180−1TX − Single Channel 8−Bit Real
- 82380−2TX − Dual Channel 12−Bit I/Q
- 82380−2TX Wide DUC − Single Channel 12−Bit I/Q
- 82121−2TX − Dual Channel 16−bit Real
- 82121−2TX Wide DUC − Single Channel 16−bit I/Q

See the datasheet for more details.

SPI Interface: This interface provides read/write access capability to the DAC38RF89 external device via a Serial Peripheral Interface. See section 4.1.2 and 4.1.4 for SPI programming details. See DAC38RF89 datasheet for register details.

5.6 Programming Sequence

This section briefly describes the programming sequence of registers to initiate and complete a transaction on the DAC38RF89 DAC Interface Core. The programming sequence for this core is as follows:

Reads:

1) Load the address register.

2) Verify ready bit is set.

3) Toggle read pulse.

4) Wait for ready bit to be set.

5) Read readback register.

Writes:

1) Load the address register.

2) Load write data register.

3) Verify ready bit is set.

4) Toggle write pulse.

5) Wait for ready bit to be set.

NOTE: For more information about SPI Interface access and register details see the DAC38RF89 datasheet.

5.7 Timing Diagrams

Timing diagrams are 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.

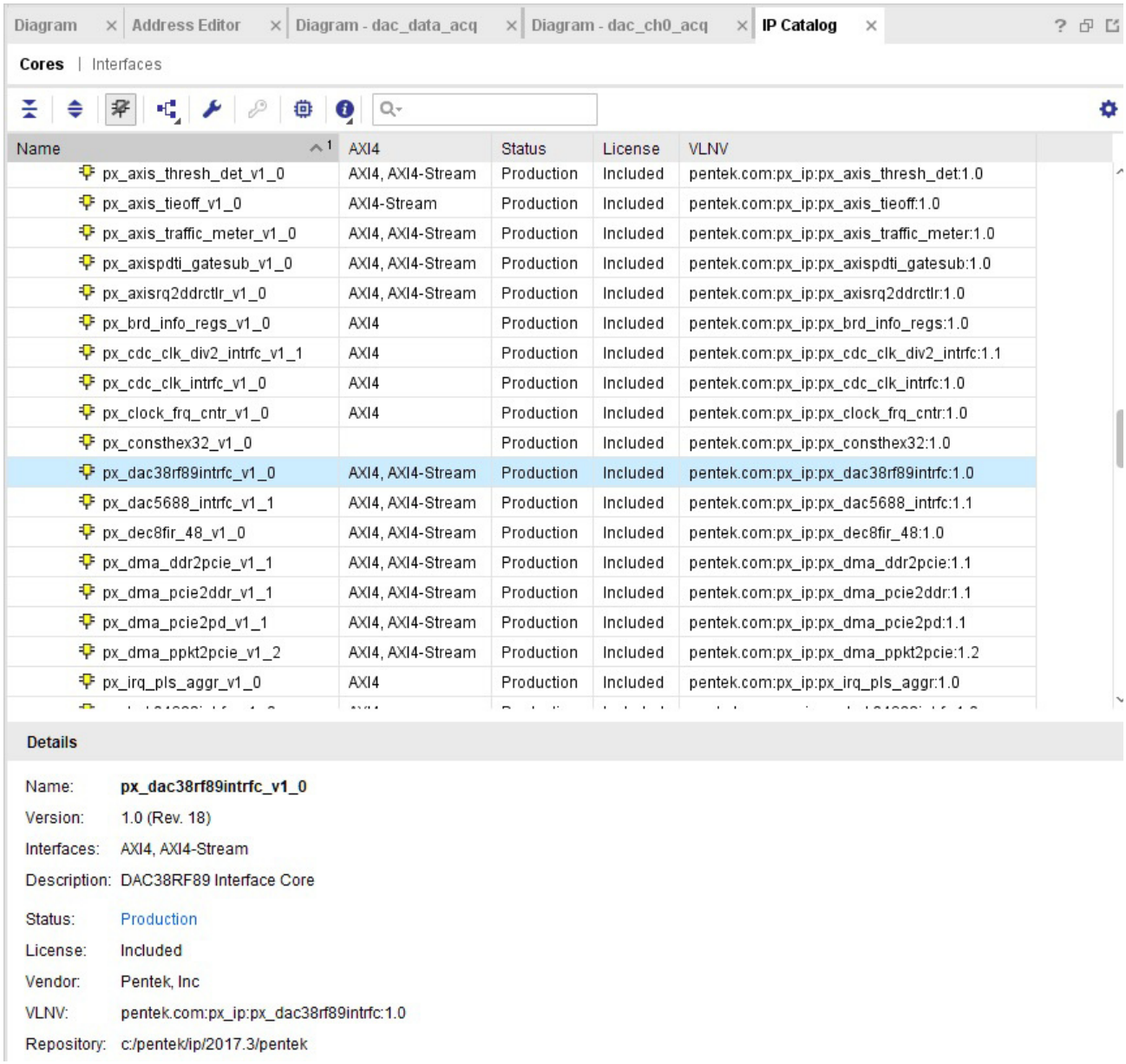
This page is intentionally blank

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek DAC38RF89 DAC Interface Core. It also includes simulation, synthesis and implementation steps that are specific to this core. This IP core can be generated from the Vivado IP Catalog when the Pentek IP Repository has been installed. It will appear in the IP Catalog list as **px_dac38rf89intrfc_v1_0** as shown in Figure 6–1.

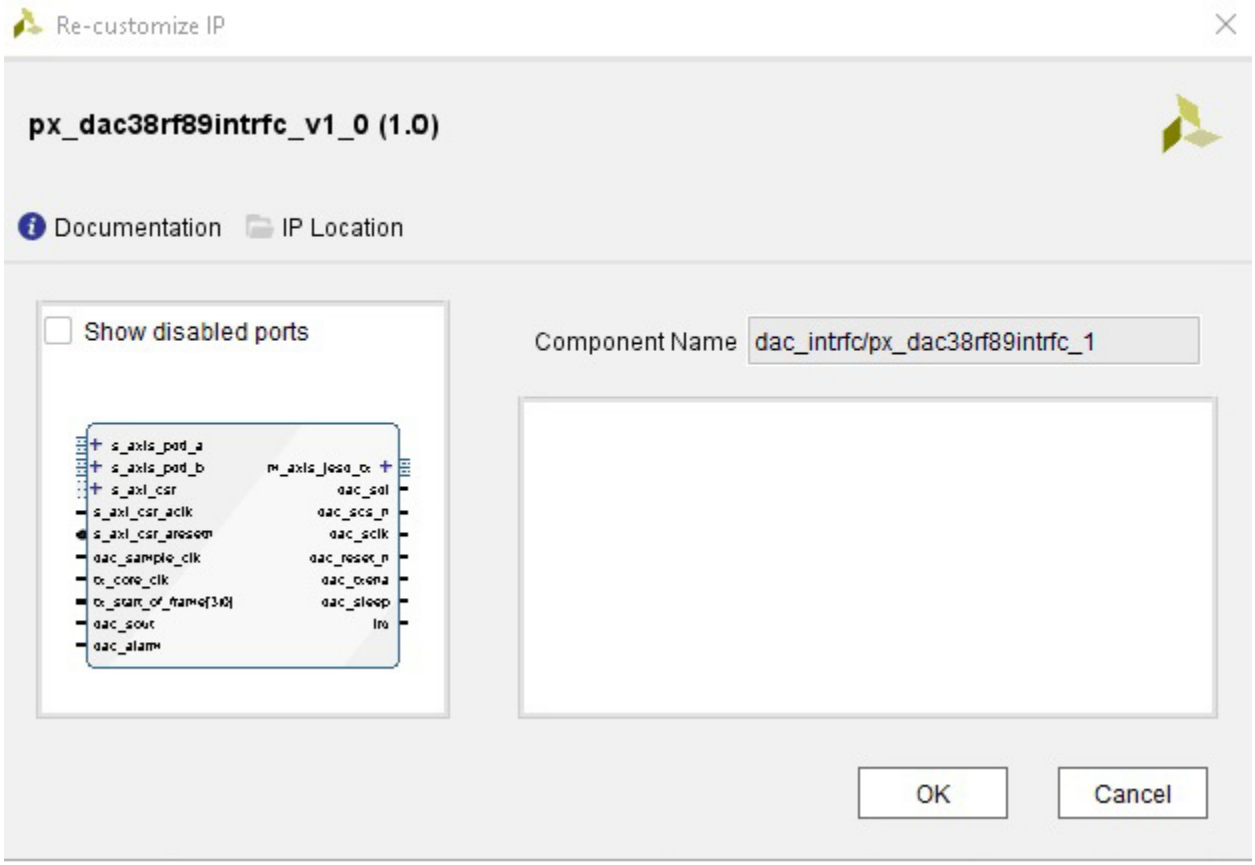
Figure 6–1: DAC38RF89 DAC Interface Core in Pentek IP Catalog



6.1 Pentek IP Catalog (continued)

When you select the **px_dac38rf89intrfc_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: DAC38RF89 DAC Interface Core IP Symbol



6.2 User Parameters

There are no user parameters.

6.3 Generating Output

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

6.4 Constraining the Core

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

Required Constraints

The XDC constraints are not provided with this core. The necessary constraints can be applied in the top level module of the user design.

Device, Package, and Speed Grade Selections

This IP works for the Kintex Ultrascale FPGAs.

Clock Frequencies

The maximum **`dac_sample_clk`** frequency for this IP core is 400 Mhz while the AXI–Lite interface clock (**`s_axis_aclk`**) frequency is 250 MHz.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking and Placement

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

6.5 Simulation

The test bench file, **tb_dac38rf89intrfc_pack.vhd**, has been provided to simulate DAC38RF89 L–M–F–S–Hd modes. The following generics can be modified to run various simulations.

```
DAC_JESD_FORMAT_MODE   : std_logic_vector(2 downto 0) := "100";
```

-- For Dual channel mode. Select which channels to run

-- bit 0 – run channel A

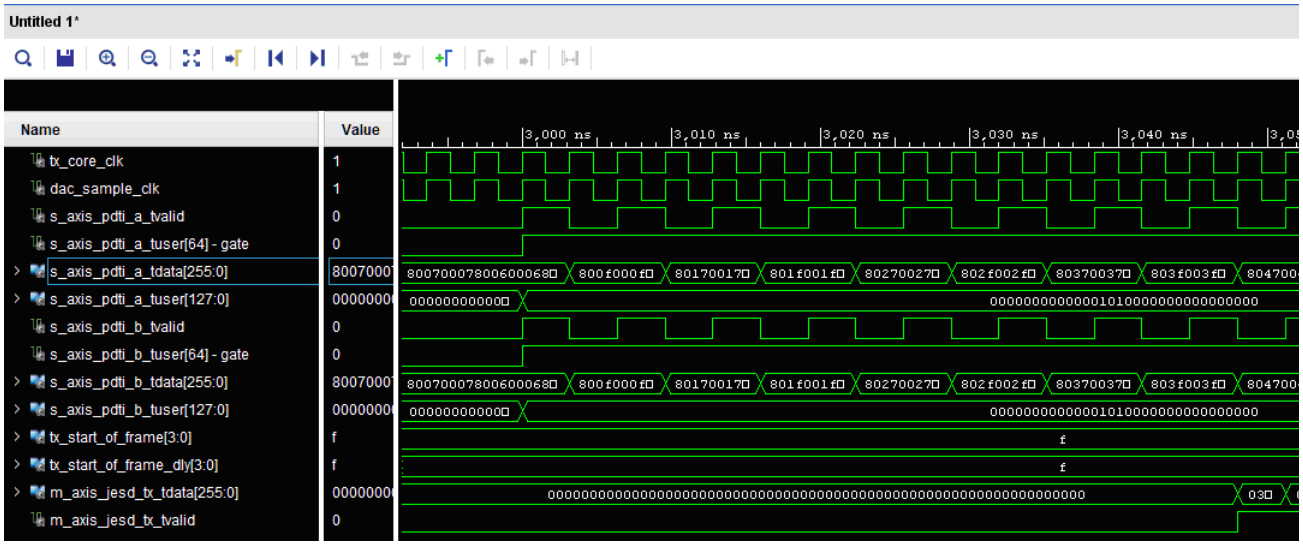
-- bit 1 – run channel B

```
CHAN                   : std_logic_vector(1 downto 0) := "11";
```

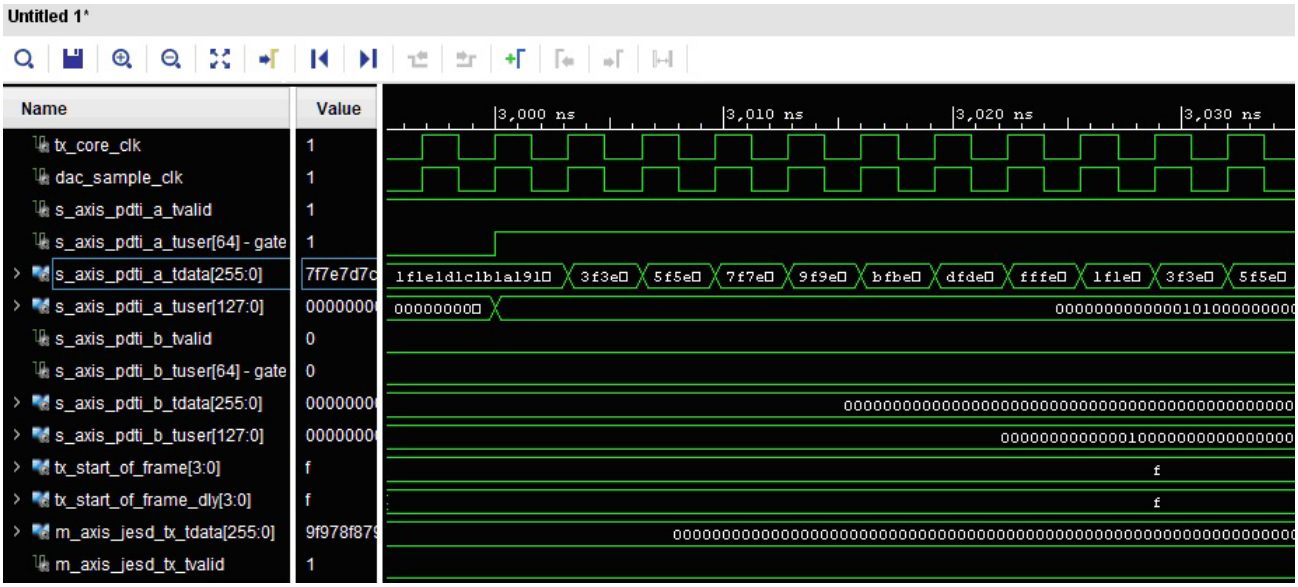
-- Test Wide DUC Mode – Applies only to JESD Format Mode "100" and "101"

```
WIDE_DUC_MODE          : std_logic := '1'
```

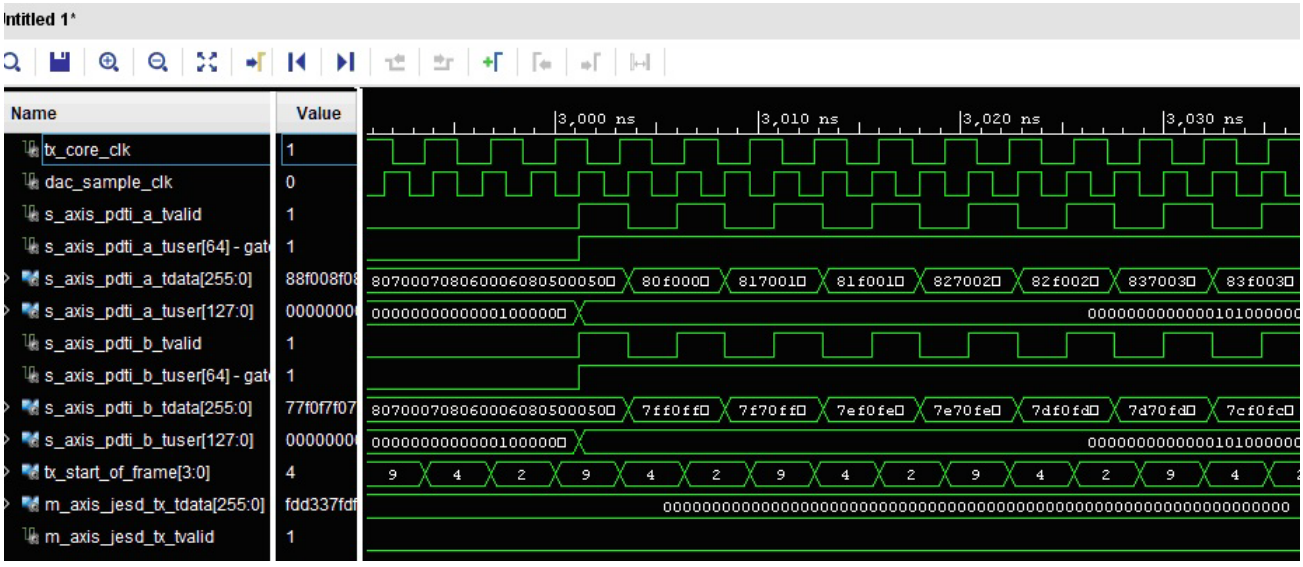
6.5.1 Jmode 001 – LMFSHd 84111 Dual Channel Mode 16–bit



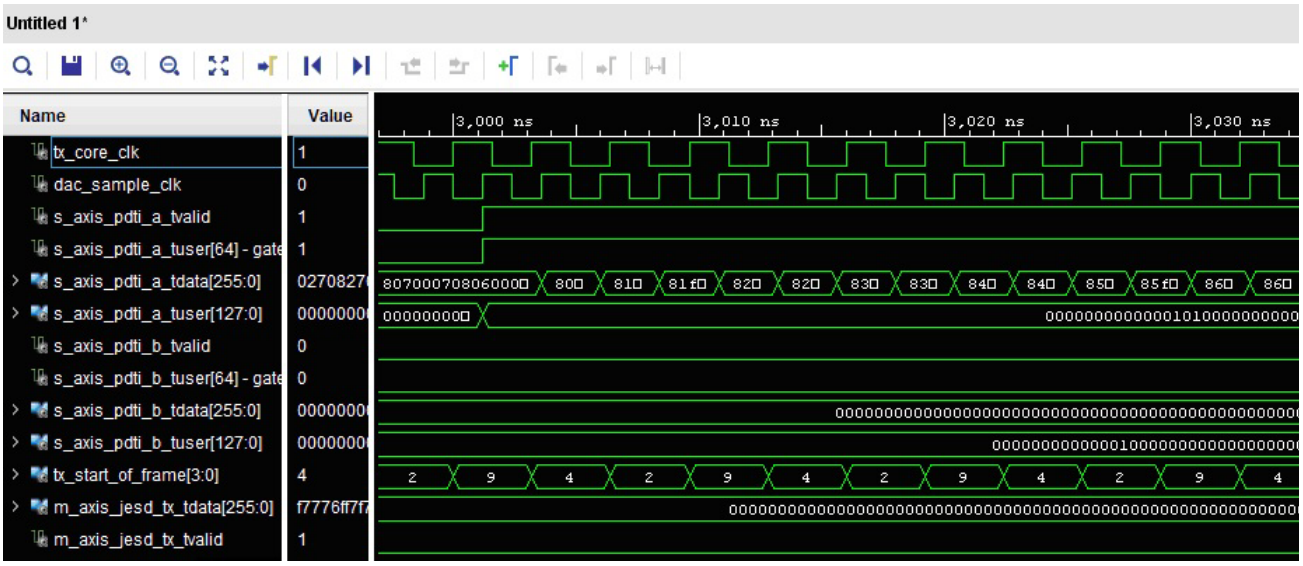
6.5.2 Jmode 011 – LMFSHd 84111 – Single Channel 8–bit



6.5.3 Jmode 100 – LMFSHd 82380 – Dual Channel 12–bit



6.5.4 Jmode 100 – LMFSHd 82380 – Single Channel 12–bit Wide DUC



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