IP CORE MANUAL

RF DAC Data Formatter IP

px_rf_dac_fmtr

PENTEK

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

Manual Part Number: 807.48434 Rev: 1.0 – November 13, 2018

Manual Revision History

Date	Version	Comments
11/13/18	1.0	Initial Release

Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Pentek products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Pentek hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON–INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Pentek shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in conjunction with, the Materials (including your use of Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage and loss was reasonably foreseeable or Pentek had been advised of the possibility of the same. Pentek assumes no obligation to correct any error contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the materials without prior written consent. Certain products are subject to the terms and conditions of Pentek's limited warranty, please refer to Pentek's Ordering and Warranty information which can be viewed at http://www.pentek.com/contact/customerinfo.cfm; IP cores may be subject to warranty and support terms contained in a license issued to you by Pentek. Pentek products are not designed or intended to be fail–safe or for use in any application requiring fail–safe performance; you assume sole risk and liability for the use of Pentek products in such critical applications.

Copyright

Trademarks

Pentek, Jade, and Navigator are trademarks or registered trademarks of Pentek, Inc.

ARM and AMBA are registered trademarks of ARM Limited. PCI, PCI Express, PCIe, and PCI–SIG are trademarks or registered trademarks of PCI–SIG. Xilinx, Kintex UltraScale, Vivado, and Platform Cable USB are registered trademarks of Xilinx Inc., of San Jose, CA.

Printed in the United States of America.

Table of Contents

Page

Table of Contents

IP Facts

Description

Pentek's Navigator™ RF DAC Data Formatter Core converts I/Q data or Real data AXI4–Streams to an AXI4–Stream which is compatible with the Xilinx RFSoC's Digital–to–Analog converters.

This core complies with the ARM® AMBA® AXI4 Specification. This user manual defines the hardware interface, software interface, and parameterization options for the RF DAC Data Formatter Core.

Features

- User selectable input data format (I/Q or Real) via AXI4–Lite CSR
- Real data (64–bits input & output) is 8–samples/cycle, and is input on the I–Data input bus
- I/Q data (64–bits on each input, 128–bit interleaved output) can be 4, 2 or 1 samples/cycle
- Output is always 8–samples/cycle

Table 1–1: IP Facts Table

Core Specifics	
Supported Design Family[a]	Zynq® Ultrascale+ RFSoC
Supported User Interfaces	AXI4–Lite and AXI4–Stream
Resources	See Table 2–1
Provided with the Core	
Design Files	VHDL
Example Design	Not Provided
Test Bench	VHDL
Constraints File	Not Provided[b]
Simulation Model	VHDL
Supported S/W Driver	HAL Software Support
Tested Design Flows	
Design Entry	Vivado® Design Suite 2018.2 or later
Simulation	Vivado VSim
Synthesis	Vivado Synthesis
Support	
Provided by Pentek fpgasupport@pentek.com	

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

This page is intentionally blank

Chapter 1: Overview

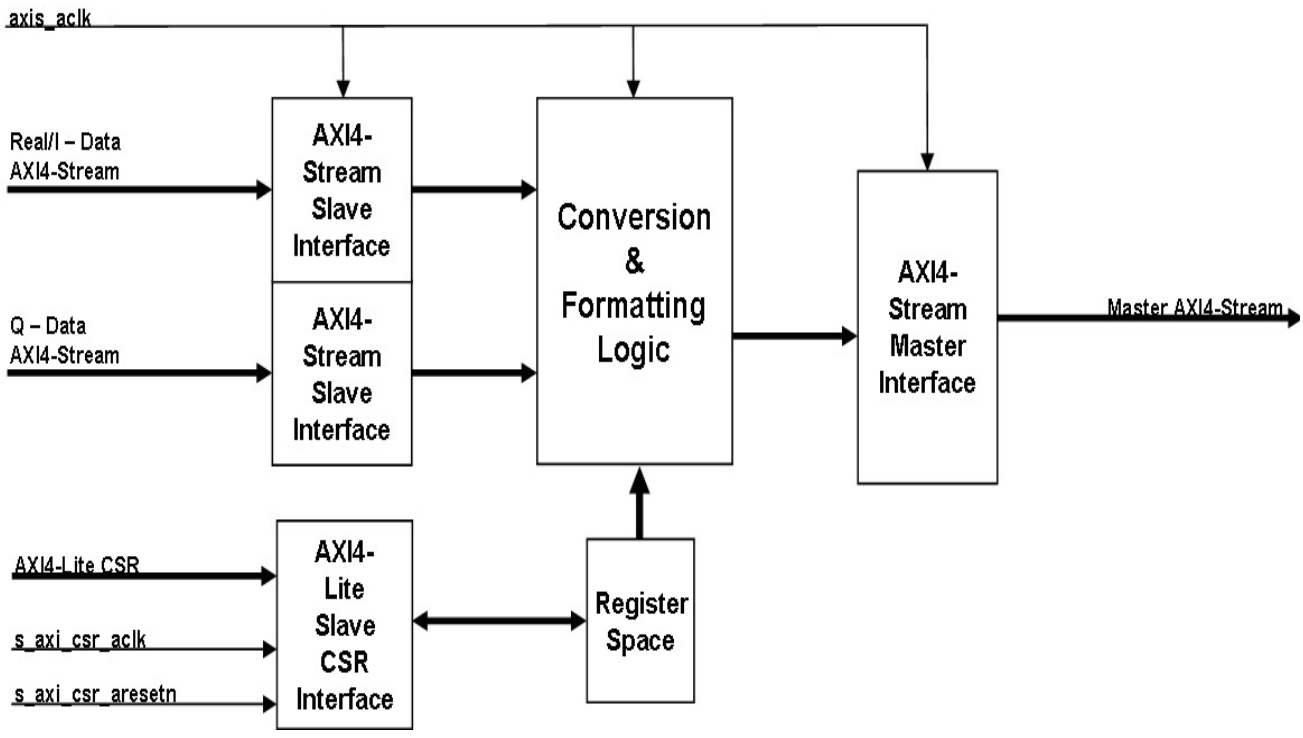
1.1 Functional Description

The RF DAC Data Formatter Core converts an input AXI4−Stream's data to an AXI4−Stream that is compatible with the Xilinx RFSoC's DAC. The input data can be either Real or I/Q, and is user−selectable via a control register.

Real data must be 64−bits wide and 8−samples/cycle, and must be applied to the I−Data AXI4−Stream Slave input bus. I/Q data must be 64−bits each, and can be 4−, 2− or 1−sample/cycle. The output data stream is fixed at 8−samples/cycle. For I/Q data at the lower sample rates, the valid data samples are right−justified.

Figure 1−1 is a top−level block diagram of the RF DAC Data Formatter Core. The modules within the block diagram are explained in other sections of this manual.

Figure 1−1: RF DAC Data Formatter Core Block Diagram



1.1 Functional Description (continued)

- **AXI4−Stream Slave Interfaces:** These modules implement the AXI4−Stream Slave interfaces for the I−Data/Real and the Q−Data AXI4−Stream inputs. For additional details about the AXI4−Stream interfaces, refer to Section 3.2.
- **AXI4−Lite Slave CSR 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 register for the core. The register is accessed by the user through the AXI4−Lite interface.
- **Conversion & Formatting Logic:** This module contains the necessary logic for converting the input data stream to the format required by the Xilinx RFSoC DACs.
- **AXI4−Stream Master Interface:** This module implements an AXI4−Master interface for the output data. For additional details about the AXI4−Stream interface, refer to Section 3.2.

1.2 Applications

This core is useful for converting Real or I/Q AXI4−Stream DAC data to a format which is compatible with the Xilinx RFSoC DACs.

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) *Zynq UltraScale+ RFSoC RF Data Converter Datasheet*

This page is intentionally blank

Chapter 2: General Product Specifications

2.1 Standards

The RF DAC Data Formatter Core has interfaces that comply with the *AMBA AXI4–Lite Protocol Specification* and the *AMBA AXI4–Stream Protocol Specification*.

2.2 Performance

The performance of the RF DAC Data Formatter Core is limited primarily 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 RF DAC Data Formatter Core has two incoming clock signals, the AXI4–Stream clock (**`axis_aclk`**) and AXI4–Lite Interface CSR clock (**`s_axi_csr_aclk`**). The AXI4–Lite Interface CSR clock has a maximum frequency of 250 MHz, and the AXI4–Stream clock has a maximum frequency of 500 MHz on a Zynq Ultrascale+ –2 speed grade FPGA. Note that 250 MHz is typically the PCI Express (PCIe) AXI bus clock frequency.

2.3 Resource Utilization

The resource utilization of the RF DAC Data Formatter Core is shown in Table 2–1. Resources have been estimated for the Zynq Ultrascale+ RFSOC XCZU27dr –2 speed grade device. These values were generated using the Vivado Design Suite.

Table 2–1: Resource Usage and Availability

Resource	# Used
LUTs	79
Flip–Flops	184

NOTE: Actual utilization may vary based on the user design in which the RF DAC Data Formatter Core is incorporated.

2.4 Limitations and Unsupported Features

This section is not applicable to this IP core.

2.5 Generic Parameters

This section is not applicable to this IP core.

Chapter 3: Port Descriptions

This chapter provides port descriptions for the following interface types:

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

3.1 AXI4–Lite Core Interfaces

The RF DAC Data Formatter Core uses the Control/Status Register (CSR) interface to access the control, status and interrupt registers from the user design.

3.1.1 Control/Status Register (CSR) Interface

The CSR interface is an AXI4–Lite Slave Interface that can be used to access the control register in the RF DAC Data Formatter Core. Table 3–1 defines the ports in the CSR Interface. See Chapter 4 for a Register memory map and bit definitions. See the *AMBA AXI4–Lite Specification* for more details on the AXI4–Lite interface.

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

Port	Direction	Width	Description
s_axi_csr_aclk	Input	1	**Clock**
s_axi_csr_aresetn	Input	1	**Reset:** Active low. This will reset the state machine within the core.
s_axi_csr_awaddr	Input	12	**Write Address:** Address used for write operations. It must be valid when **s_axi_csr_awvalid** is asserted and must be held until **s_axi_csr_awready** is asserted by the RF DAC Data Formatter Core.
s_axi_csr_awprot	Input	3	**Protection:** The RF DAC Data Formatter 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 RF DAC Data Formatter Core asserts **s_axi_csr_awready** when it is ready to accept the address. The **s_axi_csr_awvalid** must remain asserted until the rising clock edge after the assertion of **s_axi_csr_awready**.
s_axi_csr_awready	Output	1	**Write Address Ready:** This output is asserted by the RF DAC Data Formatter 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.

Table 3-1: Control/Status Register (CSR) Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
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 RF DAC Data Formatter 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 RF DAC Data Formatter 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 RF DAC Data Formatter Core when the write operation is complete and the Write Response is valid. It is held until **s_axi_csr_bready** is asserted by the user logic.
s_axi_csr_araddr	Input	12	**Read Address:** Address used for read operations. It must be valid when **s_axi_csr_arvalid** is asserted and must be held until **s_axi_csr_arready** is asserted by the RF DAC Data Formatter Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the RF DAC Data Formatter Core.

Table 3-1: Control/Status Register (CSR) Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axi_csr_arvalid	Input	1	**Read Address Valid:** This input must be asserted to indicate that a valid read address is available on **s_axi_csr_araddr**. The core asserts **s_axi_csr_arready** when it ready to accept the Read Address. This input must remain asserted until the rising clock edge after the assertion of **s_axi_csr_arready**.
s_axi_csr_arready	Output	1	**Read Address Ready:** This output is asserted by the RF DAC Data Formatter 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 RF DAC Data Formatter 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 RF DAC Data Formatter Core when the read is complete and the read data is available on **s_axi_csr_rdata**. It is held until **s_axi_csr_rready** is asserted by the user logic.
s_axi_csr_rready	Input	1	**Read Data Ready:** This signal is asserted by the user logic when it is ready to accept the Read Data.

3.2 AXI4–Stream Core Interfaces

The RF DAC Data Formatter Core has the following AXI4–Stream Interfaces, which are used to transfer the DAC data streams.

3.2.1 Stream Data Interface

These interfaces are used to bring data from the user design into the RF DAC Data Formatter Core. Table 3–2 defines the ports in the Stream Data Interface. See the *AMBA AXI4–Stream Specification* for more details on the operation of the AXI4–Stream Interface.

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

Port	Direction	Width	Description
axis_aclk	Input	1	**AXI4–Stream Clock**
axis_aresetn	Input	1	**AXI4–Stream Reset** **Note:** This reset is **NOT USED** in this core.
Real/I–Data AXI4–Stream Slave Interface			
s_axis_i_tvalid	Input	1	**Input I Data Valid:** This signal is asserted by the user design when data is valid on the **s_axis_i_tdata** bus. A data transfer takes place when both **s_axis_i_tvalid** and **s_axis_i_tready** are High in the same cycle. This input is used for 64–bit I data when I/Q data mode is selected, and for 64–bit Real data when Real data mode is selected (see **Section 4.1**).
s_axis_i_tready	Output	1	**Input I Data Ready:** This signal is asserted by the RF DAC Data Formatter Core when it is ready to accept data on the **s_axis_i_tdata** input.
s_axis_i_tdata	Input	64	**Real or I Input Data:** This is the input data bus for either the I–data component of the data when I/Q Data Mode is selected, or Real data when Real Data Mode is selected (see **Section 4.1**).
Q–Data AXI4–Stream Slave Interface			
s_axis_q_tvalid	Input	1	**Input Q Data Valid:** This signal is asserted by the user design when data is valid on its **s_axis_q_tdata** bus. A data transfer takes place when both **s_axis_q_tvalid** and **s_axis_q_tready** are High in the same cycle. This input is only used for 64–bit Q data when I/Q data mode is selected.

Table 3-2: AXI4-Stream Slave Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axis_q _tready	Output	1	**Input Q Data Ready:** This signal is asserted by the RF DAC Data Formatter Core when it is ready to accept data on the `s_axis_q_tdata` input.
s_axis_q _tdata	Input	64	**Q Input Data:** This is the input data bus for the Q–data component of the data when I/Q Data Mode is selected. It is unused when Real Data Mode is selected.

3.2.2 AXI4–Stream Master Interface

This interface is used to deliver the formatted data to the Xilinx RFSoC DAC. Table 3–3 defines the ports in the AXI4–Stream Master Interface. See the *AMBA AXI4–Stream Specification* for more details on the operation of the AXI4–Stream Interface.

Table 3-3: AXI4-Stream Master Interface Port Descriptions			
Port/ Signal Name	**Direction**	**Width**	**Description**
m_axis_tvalid	Output	1	**Output Data Valid:** This signal is asserted by the RF DAC Data Formatter Core when data is valid on the `m_axis_tdata` bus.
m_axis_tdata	Output	128	**Output Data:** This bus contains the data formatted for the Xilinx RFSoC DAC.

This page is intentionally blank

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the Register Space of the RF DAC Data Formatter Core. The memory map is provided in Table 4–1.

Table 4–1: Register Space Memory Map

Register Name	Address (Base Address +)	Access	Description
Data Mode	0x00	R/W	This register determines the input data format.

4.1 Data Mode Register

This register contains the control for the input data format. The Data Mode Register is illustrated in Figure 4–1 and described in Table 4–2.

Figure 4–1: Data Mode Register

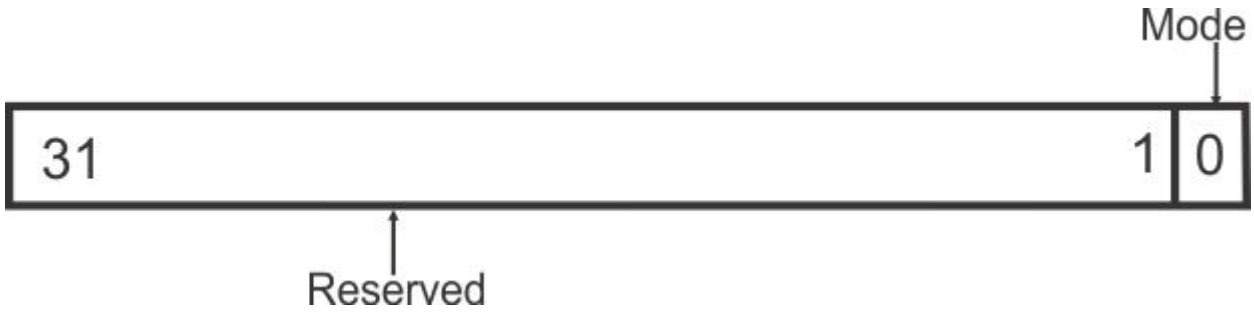


Table 4–2: Data Mode Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	mode	0	R/W	**Input Data Mode:** This bit sets the input format as follows: 0 = REAL Data 1 = I/Q Data

Chapter 5: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the RF DAC Data Formatter Core.

5.1 General Design Guidelines

The RF DAC Data Formatter Core provides the required logic to format incoming Real or I/Q Data as required for the Xilinx RFSoC DACs.

5.2 Clocking

AXI4−Lite Clock: **s_axi_csr_aclk**.

This clock is used to clock the AXI4−Lite Control/Status Register (**s_axi_csr**) interface of the core and its associated logic.

AXI4−Stream Clock: **axis_aclk**.

This is the clock used by all of the data path components as well as the slave and master AXI4−Stream ports.

5.3 Resets

CSR Reset: **s_axi_csr_aresetn**.

This is an active low reset synchronous with **s_axi_csr_clk**. When asserted, the control register is reset.

AXI4−Stream Reset: **axis_aresetn**.

NOTE: This reset is **not used** in the RF DAC Data Formatter Core.

5.4 Interrupts

This core has no interrupts.

5.5 Interface Operation

- **Control/Status Register Interface (s_axi_csr…):** This is the control register interface. It is associated with the **`s_axi_csr_aclk`**, and is a standard AXI4−Lite type interface. See Chapter 4 for the control register memory map and for more details on the register that can be accessed through this interface. For more details about this interface refer to Section 3.1.

- **AXI4−Stream Data Interfaces:** The **`s_axis_<i,q>`** and **`m_axis`** busses are the input and output data busses that carry the DAC data into and out of the core. These are standard AXI4−Stream type interfaces associated with the AXI4−Stream clock (**`axis_aclk`**). For more details about this interface refer to Section 3.2.

5.6 Programming Sequence

This section briefly describes the programming sequence for the RF DAC Data Formatter Core.

1) Write the desired value to the Control Register.

2) When DAC data is applied to the AXI4−Stream Slave input, observe a reformatted data stream at the output.

5.7 Timing Diagrams

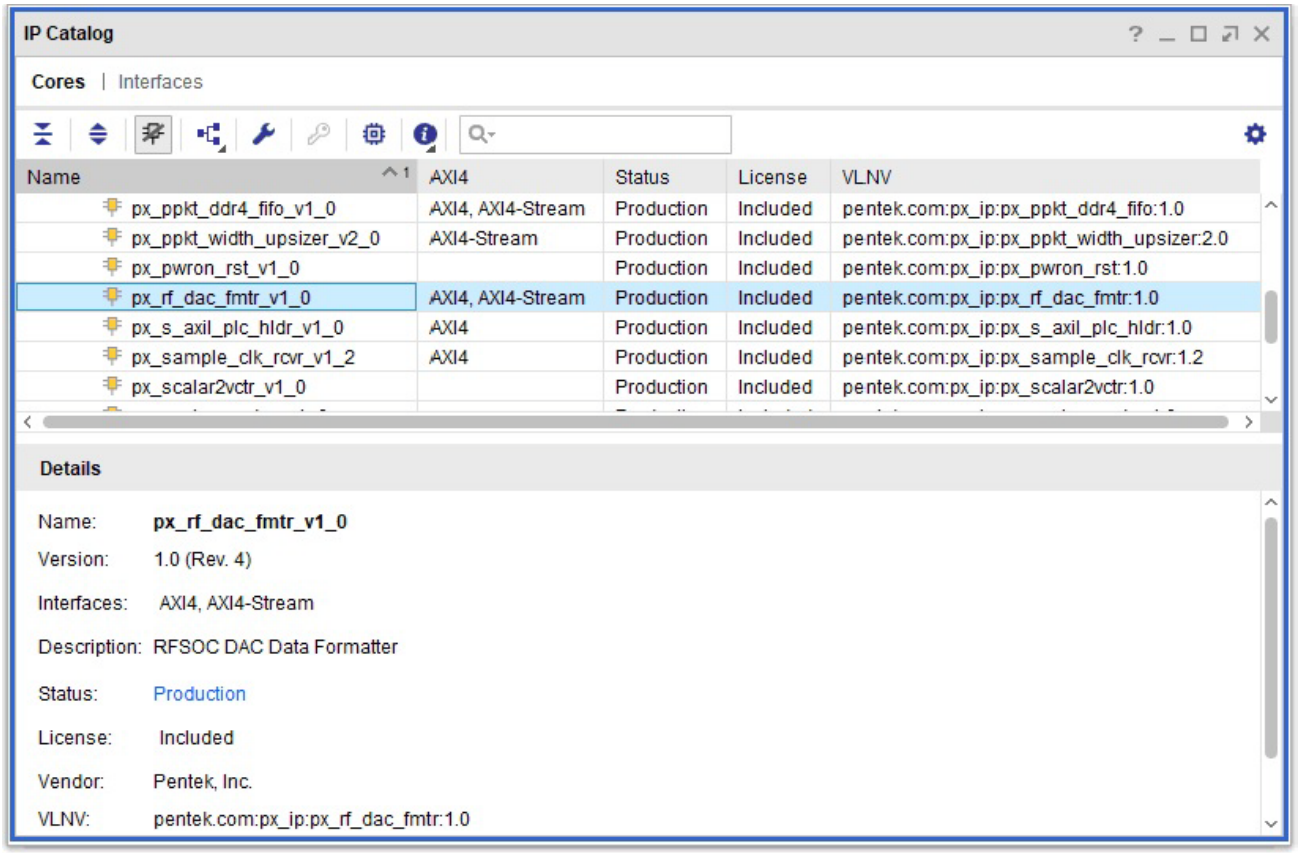
This section is not applicable to this IP core.

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek RF DAC Data Formatter 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_rf_dac_fmtr_v1_0** as shown in Figure 6–1.

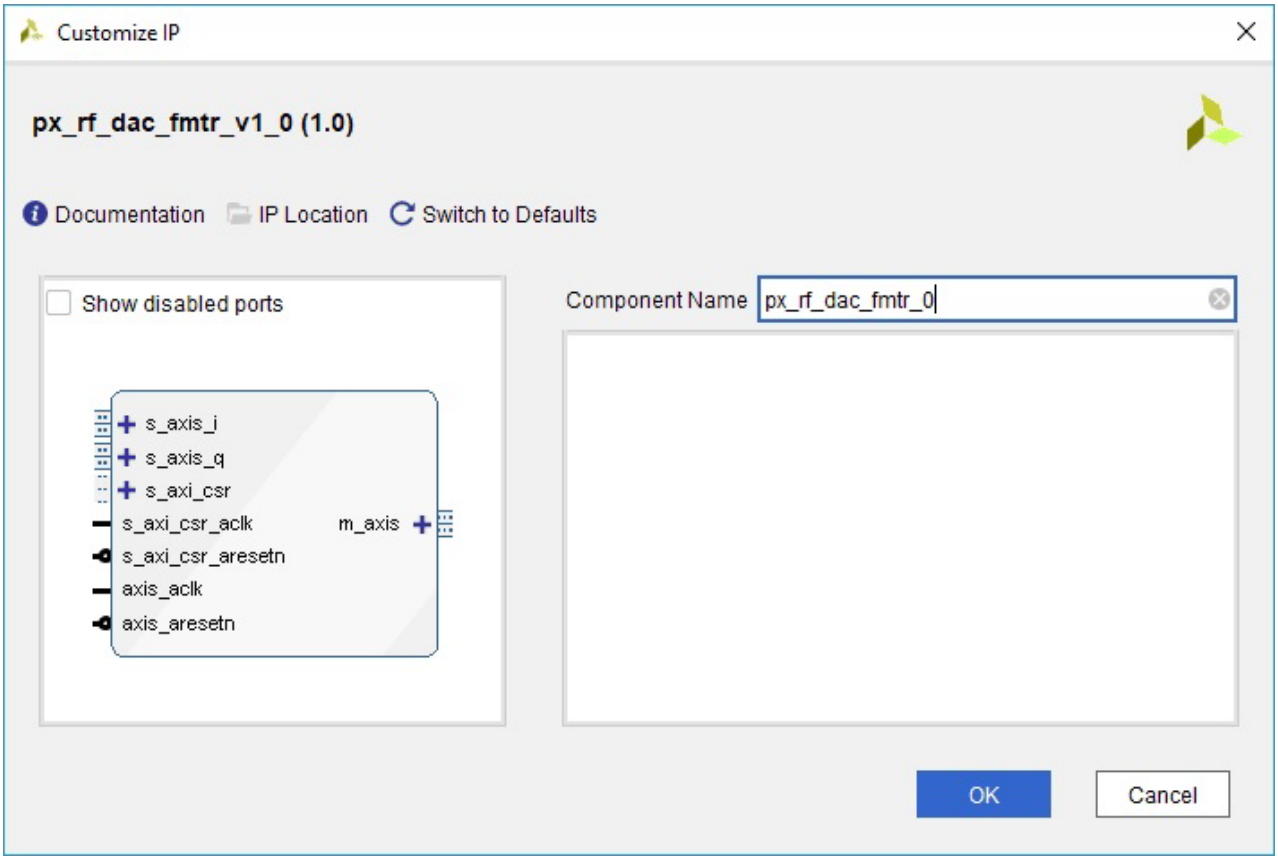
Figure 6–1: RF DAC Data Formatter Core in Pentek IP Catalog



6.1 Pentek IP Catalog (continued)

When you select the **px_rf_dac_fmtr_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: RF DAC Data Formatter IP Core 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 core in Vivado Design Suite environment.

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 Xilinx Zynq Ultrascale+ RFSoC FPGA family.

Clock Frequencies

The AXI4−Lite interface clock (**`s_axi_csr_aclk`**) frequency is 250 MHz.

The AXI4−Stream clock (**`axis_aclk`**) frequency is 500 MHz (Maximum).

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking and Placement

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

6.5 Simulation

This section is not applicable to this IP core.

6.6 Synthesis and Implementation

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