IP CORE MANUAL

RF ADC Control & Status IP

px_rf_adc_cntl_stat

PENTEK

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

Manual Part Number: 807.48428 Rev: 1.0 – October 30, 2018

Manual Revision History

Date	Version	Comments
10/30/18	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

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™ RF ADC Control & Status Core provides user access to the control and status signals of the Xilinx RFSoC ADC. It also provides automatic control for calibration freeze based on signal strength. Access to these control and status signals is facilitated by an AXI4–Lite CSR interface.

This core complies with the ARM® AMBA® *AXI4 Specification* and also provides a control/status register interface. This manual defines the hardware interface, software interface, and parameterization options for the RF ADC Control & Status Core.

Features

- Easy user access to the RFSoC ADC controls for Calibration Freeze, PL Events, etc.
- Optional automatic control for calibration freeze based on signal strength for each ADC
- Separate interrupts for over–range, over–voltage and over–threshold for each ADC
- Register access through AXI4–Lite CSR interface

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

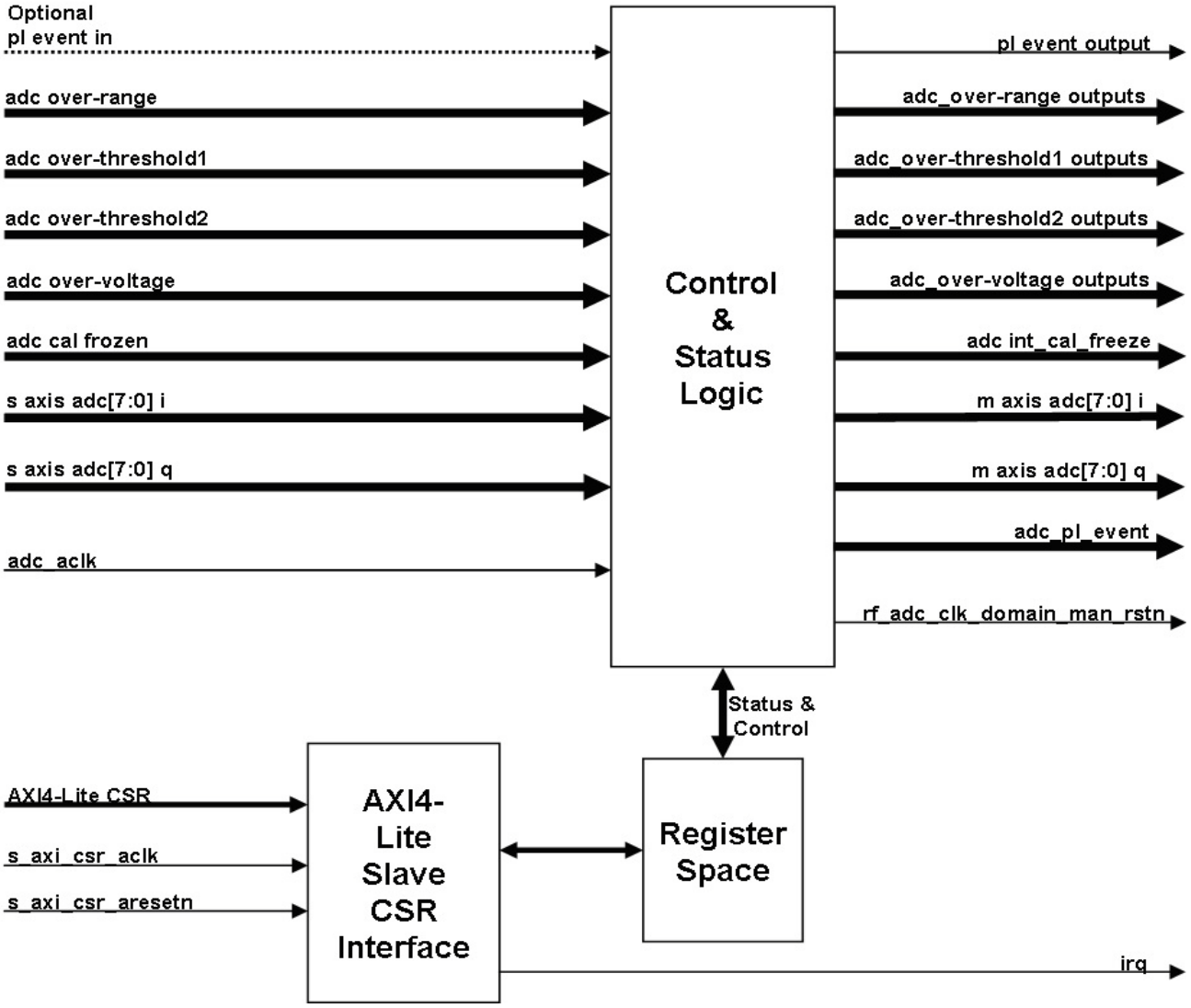
Chapter 1: Overview

1.1 Functional Description

The RF ADC Control & Status Core provides user access via the CSR AXI4−Lite bus to control calibration freeze, event enables and interrupt status for the Xilinx RFSoC ADCs. The core has an optional automatic calibration freeze feature which allows auto calibration only when the signal strength is sufficient to insure a useful calibration.

Figure 1−1 is a top−level block diagram of the Pentek RF ADC Control & Status Core. The modules within the block diagram are explained in the later sections of this man−ual.

Figure 1−1: RF ADC Control & Status Core Block Diagram



1.1 Functional Description (continued)

- **Control & Status Logic:** This module implements the control, status monitoring and interrupt functions of the core. It also implements the automatic calibration freeze control logic.

- **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.

- **Register Space:** This module contains the control, status and interrupt registers for the core. The registers are accessed by the user through the AXI4−Lite interface.

1.2 Applications

This core is useful for controlling and monitoring Xilinx RFSoC ADCs from a user−accessible AXI4−CSR interface.

1.3 System Requirements

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

1.4 Licensing and Ordering Information

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

1.5 Contacting Technical Support

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

1.6 Documentation

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

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

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

3) *Xilinx Zynq UltraScale+ RFSoC Data Sheet, DS926*

4) *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 RF ADC Control & Status Core has bus interfaces that comply with the *ARM AMBA AXI4–Lite Protocol Specification* and the *ARM AMBA AXI4–Stream Protocol Specification.*

2.2 Performance

The performance of the RF ADC Control & Status Core is limited by the FPGA logic speed. The values presented in this section should be used as an estimation guideline. Actual performance can vary.

2.2.1 Maximum Frequencies

The RF ADC Control & Status Core has two incoming clock signals, the ADC clock (**adc_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 ADC 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 ADC Control & Status 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	10,084
Flip–Flops	14,151

NOTE: **This table assumes that the core is generated targeting all 8 ADCs.**

NOTE: Actual utilization may vary based on the user design in which the RF ADC Control & Status Core is incorporated.

2.4 Limitations and Unsupported Features

This section is not applicable to this IP core.

2.5 Generic Parameters

The generic parameters of the RF ADC Control & Status Core are described in Table 2–2. These parameters can be set as required by the user application while customizing the core.

Table 2-2: Generic Parameters

Port/Signal Name	Type	Description
has_adc<7:0>	Boolean	**Has ADCx:** These parameters enable/disable the data path and control logic for each individual ADC. When set to FALSE, no logic is generated for the particular ADC. The default settings for these parameters are TRUE.
en_adc01_rts **en_adc23_rts** **en_adc45_rts** **en_adc67_rts**	Boolean	**Enable ADC Real–Time Signals:** These are the enables for the 4 dual–ADC tiles. The default settings for these parameters are TRUE.
en_adc01_frz **en_adc23_frz** **en_adc45_frz** **en_adc67_frz**	Boolean	**Enable ADC Freeze:** These are the enables for the calibration freeze function for each ADC tile. The default settings for these parameters are TRUE.
has_pl_event_in	Boolean	**Has pl_event_in input:** This parameter defines whether a **`pl_event_in`** input is implemented in the core. The default setting for this parameter is TRUE.
en_adc<7:0> **_thresh_outputs**	Boolean	**Enable ADC Threshold Outputs:** These are the enables for the threshold outputs for the core. The default settings for these parameters are TRUE.

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

3.1.1 Control/Status Register (CSR) Interface

The CSR interface is an AXI4–Lite Slave Interface that can be used to access the control and status registers in the RF ADC Control & Status Core. Table 3–1 defines the ports in the CSR Interface. See Chapter 4 for the 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/status registers to their initial states.
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 ADC Control & Status Core.
s_axi_csr_awprot	Input	3	**Protection:** The RF ADC Control & Status 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 ADC Control & Status 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 RF ADC Control & Status 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 RF ADC Control & Status 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 ADC Control & Status 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.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axi_csr_bvalid	Output	1	**Write Response Valid:** This signal is asserted by the RF ADC Control & Status 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 ADC Control & Status Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the RF ADC Control & Status Core.
s_axi_csr_arvalid	Input	1	**Read Address Valid:** This input must be asserted to indicate that a valid read address is available on **s_axi_csr_araddr**. The core asserts **s_axi_csr_arready** when it ready to accept the Read Address. This input must remain asserted until the rising clock edge after the assertion of **s_axi_csr_arready**.
s_axi_csr_arready	Output	1	**Read Address Ready:** This output is asserted by the RF ADC Control & Status 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 ADC Control & Status 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 ADC Control & Status 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 representing all of the enabled interrupt sources.

3.2 AXI4–Stream Core Interfaces

The RF ADC Control & Status Core has the following AXI4–Stream Interfaces, which are used to transfer the ADC data streams.

3.2.1 ADC Channel Data Interface

These interfaces are used to transfer data from each of the ADC channel data slave input ports, through the control and status logic to the ADC master channel data output ports of the ADC Control & Status Core. Table 3–2 defines the ports in theAXI4–Stream Data Interfaces. See Chapter 4 for the register memory map and bit definitions. See the *AMBA AXI4–Stream Specification* for more details on operation of the AXI4–Stream interfaces.

Table 3-2: Control/Status Register (CSR) Interface Port Descriptions

Port	Direction	Width	Description
AXI4–Stream Slave Interfaces			
s_axis_adc<7:0>_i_tvalid	Input	1	**ADC<7:0> Input i Data Valid:** This signal is asserted by the respective ADC when data is valid on its **s_axis_adc<7:0>_i_tdata** bus. A data transfer takes place when both **s_axis_adc<7:0>_i_tvalid** and **s_axis_adc<7:0>_i_tready** are High in the same cycle.
s_axis_adc<7:0>_i_tready	Output	1	**ADC<7:0> Input i Data Ready:** This signal is asserted by the ADC Control & Status Core when it is ready to accept data from the respective ADC.
s_axis_adc<7:0>_i_tdata	Input	128	**ADC<7:0> i Input Data**
s_axis_adc<7:0>_q_tvalid	Input	1	**ADC<7:0> Input q Data Valid:** This signal is asserted by the respective ADC when data is valid on its **s_axis_adc<7:0>_q_tdata** bus. A data transfer takes place when both **s_axis_adc<7:0>_q_tvalid** and **s_axis_adc<7:0>_q_tready** are High in the same cycle.
s_axis_adc<7:0>_q_tready	Output	1	**ADC<7:0> Input q Data Ready:** This signal is asserted by the ADC Control & Status Core when it is ready to accept data from the respective ADC.
s_axis_adc<7:0>_q_tdata	Input	128	**ADC<7:0> q Input Data**
AXI4– Stream Master Interfaces			
m_axis_adc<7:0>_i_tvalid	Output	1	**ADC<7:0> Output i Data Valid:** This signal is asserted by the ADC Control & Status Core when data is valid on the respective **m_axis_adc<7:0>_i_tdata** bus.

Table 3-2: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
m_axis_adc<7:0>_i_tdata	Output	128	**ADC<7:0> i Output Data**
m_axis_adc<7:0>_q_tvalid	Output	1	**ADC<7:0> Output q Data Valid:** This signal is asserted by the ADC Control & Status Core when data is valid on the respective **m_axis_adc<7:0>_q_tdata** bus.
m_axis_adc<7:0>_q_tdata	Output	128	**ADC<7:0> q Output Data**

3.3 I/O Signals

The I/O port/signal descriptions of the top level module of the RF ADC Control & Status Core are provided in Table 3–3.

Table 3–3: I/O Signal Descriptions			
Port/Signal Name	**Type**	**Direction**	**Description**
adc_clk	std_logic	Input	**ADC Clock:** This is the clock from the RF ADC. This is the clock with which most of the logic in the core (save the CSR logic) is associated.
s_axis_ptctl_tvalid	std_logic	Input	**Data Valid:** This signal is asserted when data is valid on the **s_axis_ptctl_tdata** bus.
s_axis_ptctl_tdata	std_logic_vector [31:0]	Input	**PTCTL Data:** This data bus is associated with **adc_aclk**, and is only used for synchronizing the **adc**_pl_event** outputs (see Section 4.3).
pl_event_in	std_logic	Input	**PL Event Trigger Input (Optional):** This input is a trigger option (see Section 4.3) for the pulse generator that drives the **adc**_pl_event** and the **pl_event_out** outputs. This input is only available when parameter **has_pl_event_in** is set to TRUE.
adc0_01_cal_frozen **adc0_23_cal_frozen** **adc1_01_cal_frozen** **adc1_23_cal_frozen** **adc2_01_cal_frozen** **adc2_23_cal_frozen** **adc3_01_cal_frozen** **adc3_23_cal_frozen**	std_logic	Input	**Calibration Frozen:** These active HIGH inputs indicate that calibrations are frozen for the respective ADC channel.

adc0_01_int_cal_freeze **adc0_23_int_cal_freeze** **adc1_01_int_cal_freeze** **adc1_23_int_cal_freeze** **adc2_01_int_cal_freeze** **adc2_23_int_cal_freeze** **adc3_01_int_cal_freeze** **adc3_23_int_cal_freeze**	std_logic	Output	**Initiate Calibration Freeze:** These active HIGH outputs force their respective ADC channel to freeze calibration.
adc0_01_over_range **adc0_23_over_range** **adc1_01_over_range** **adc1_23_over_range** **adc2_01_over_range** **adc2_23_over_range** **adc3_01_over_range** **adc3_23_over_range**	std_logic	Input	**ADC Over–Range:** These active HIGH inputs are associated with **adc_aclk**. When asserted, each input indicates that the respective ADC channel has exceeded the working range.
adc0_01_over_threshold1 **adc0_23_over_threshold1** **adc1_01_over_threshold1** **adc1_23_over_threshold1** **adc2_01_over_threshold1** **adc2_23_over_threshold1** **adc3_01_over_threshold1** **adc3_23_over_threshold1**	std_logic	Input	**ADC Over–Threshold 1:** These active HIGH inputs are associated with **adc_aclk**. When asserted, each input indicates that the respective ADC channel has exceeded the value set for Threshold 1.
adc0_01_over_threshold2 **adc0_23_over_threshold2** **adc1_01_over_threshold2** **adc1_23_over_threshold2** **adc2_01_over_threshold2** **adc2_23_over_threshold2** **adc3_01_over_threshold2** **adc3_23_over_threshold2**	std_logic	Input	**ADC Over–Threshold 2:** These active HIGH inputs are associated with **adc_aclk**. When asserted, each input indicates that the respective ADC cannel has exceeded the value set for Threshold 2.
adc0_01_over_voltage **adc0_23_over_voltage** **adc1_01_over_voltage** **adc1_23_over_voltage** **adc2_01_over_voltage** **adc2_23_over_voltage** **adc3_01_over_voltage** **adc3_23_over_voltage**	std_logic	Input	**ADC Over–Voltage:** These active HIGH inputs are associated with **adc_aclk**. When asserted, each input indicates that the respective ADC input has exceeded the allowable voltage range.

adc00_pl_event **adc01_pl_event** **adc02_pl_event** **adc03_pl_event** **adc10_pl_event** **adc11_pl_event** **adc12_pl_event** **adc13_pl_event** **adc20_pl_event** **adc21_pl_event** **adc22_pl_event** **adc23_pl_event** **adc30_pl_event** **adc31_pl_event** **adc32_pl_event** **adc33_pl_event**	std_logic	Output	**ADC PL Event:** These active HIGH, single–pulse outputs are associated with **`adc_aclk`**. They are controlled by the **PL Event Enables** register (see Section 4.3).
adc0_01_over_range_out **adc0_23_over_range_out** **adc1_01_over_range_out** **adc1_23_over_range_out** **adc2_01_over_range_out** **adc2_23_over_range_out** **adc3_01_over_range_out** **adc3_23_over_range_out**	std_logic	Output	**ADC Over–Range Output:** These are simply the **ADC Over–Range** inputs passed through the core.
adc0_01_over_threshold1_out **adc0_23_over_ threshold1_out** **adc1_01_over_ threshold1_out** **adc1_23_over_ threshold1_out** **adc2_01_over_ threshold1_out** **adc2_23_over_ threshold1_out** **adc3_01_over_ threshold1_out** **adc3_23_over_ threshold1_out**	std_logic	Output	**ADC Over–Threshold 1 Output:** These are simply the **ADC Over–Threshold 1** inputs passed through the core.
adc0_01_over_threshold2_out **adc0_23_over_ threshold2_out** **adc1_01_over_ threshold2_out** **adc1_23_over_ threshold2_out** **adc2_01_over_ threshold2_out** **adc2_23_over_ threshold2_out** **adc3_01_over_ threshold2_out** **adc3_23_over_ threshold2_out**	std_logic	Output	**ADC Over–Threshold 2 Output:** These are simply the **ADC Over–Threshold 2** inputs passed through the core.

adc0_01_over_voltage_out **adc0_23_over_ voltage _out** **adc1_01_over_ voltage _out** **adc1_23_over_ voltage _out** **adc2_01_over_ voltage _out** **adc2_23_over_ voltage _out** **adc3_01_over_ voltage _out** **adc3_23_over_ voltage _out**	std_logic	Output	**ADC Over–Voltage Output:** These are simply the **ADC Over–Voltage** inputs passed through the core.
pl_event_out	std_logic	Output	**PL Event Output:** This is a single–pulse output set–up by the by the **PL Event Enables** register (see Section 4.3).
rf_adc_clk_domain_man_rstn	std_logic	Output	**RF ADC Clock Domain Manual Reset:** This is a register–controlled reset in the RF ADC's clock domain.

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the register space of the RF ADC Control & Status Core. The memory map is provided in Table 4–1.

Table 4–1: Register Space Memory Map

Register Name	Address (Base Address +)	Access	Description
Calibration Freeze Control	0x00	R/W	Forces ADC to freeze calibration.
PL Event Reg Pulse	0x04	R/W	Initiates a pulse on the selected PL event outputs.
PL Event Enables	0x08	R/W	PL Event Enable for each ADC channel and trigger source select.
ADC Clock Domain Manual Reset	0x0C	R/W	Register controlled reset synchronized to the ADC clock domain.
Calibration Freeze Status	0x10	RO	Indicates whether calibration for the ADC channel has been frozen.
Tvalid Status	0x14	RO	Indicates the current status of the 8 ADC `tvalid` signals.
Reserved	0x18 – 0x1C	–	Reserved
Interrupt Enables	0x20	R/W	This register provides enables for the interrupts.
Interrupt Status	0x24	RO	This register provides current status of the interrupts.
Interrupt Flag	0x28	R/CLR	This register provides status of the interrupt flags.

4.1 Calibration Freeze Control Register

This register contains the controls for the automatic calibration freeze logic and for the manual calibration freeze. It is illustrated in Figure 4–1 and described in Table 4–2.

Figure 4–1: Calibration Freeze Control Register

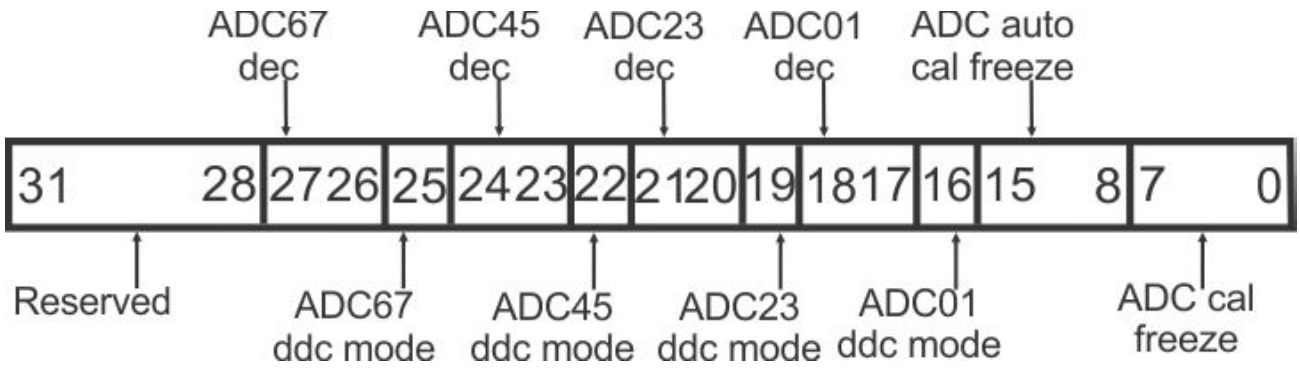


Table 4–2: Calibration Freeze Control Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:28	Reserved	N/A	N/A	**Reserved**
27:26	ADC67 dec	00	R/W	**ADC 6 & 7 Decimation Select:** These bits should reflect the decimation settings in the respective ADCs as follows: 00 = Decimate by 1 (NOT VALID when DDC Mode= 0) 01 = Decimate by 2 10 = Decimate by 4 11 = Decimate by 8
25	ADC67 ddc mode	0	R/W	**ADC 6 & 7 DDC Data Mode:** Selects the data mode of the AXI4–Stream interfaces for ADC 6 and ADC 7 as follows: 0 = REAL data is mapped to the respective "I" data bus 1 = I & Q data are mapped to their respective busses.
24:23	ADC45 dec	00	R/W	**ADC 4 & 5 Decimation Select:** These bits should reflect the decimation settings in the respective ADCs as follows: 00 = Decimate by 1 (NOT VALID when DDC Mode= 0) 01 = Decimate by 2 10 = Decimate by 4 11 = Decimate by 8

22	ADC45 ddc mode	0	R/W	**ADC 4 & 5 DDC Data Mode:** Selects the data mode of the AXI4−Stream interfaces for ADC 4 and ADC 5 as follows: 0 = REAL data is mapped to the respective "I" data bus 1 = I & Q data are mapped to their respective busses**.**
21:20	ADC23 dec	00	R/W	**ADC 2 & 3 Decimation Select:** These bits should reflect the decimation settings in the respective ADCs as follows: 00 = Decimate by 1(NOT VALID when DDC Mode= 0) 01 = Decimate by 2 10 = Decimate by 4 11 = Decimate by 8
19	ADC23 ddc mode	0	R/W	**ADC 2 & 3 DDC Data Mode:** Selects the data mode of the AXI4−Stream interfaces for ADC 2 and ADC 3 as follows: 0 = REAL data is mapped to the respective "I" data bus 1 = I & Q data are mapped to their respective busses.
18:17	ADC01 dec	00	R/W	**ADC 0 & 1 Decimation Select:** These bits should reflect the decimation settings in the respective ADCs as follows: 00 = Decimate by 1 (NOT VALID when DDC Mode= 0) 01 = Decimate by 2 10 = Decimate by 4 11 = Decimate by 8
16	ADC01 ddc mode	0	R/W	**ADC 0 & 1 DDC Data Mode:** Selects the data mode of the AXI4−Stream interfaces for ADC 0 and ADC 1 as follows: 0 = REAL data is mapped to the respective "I" data bus 1 = I & Q data are mapped to their respective busses.
15:8	ADC auto cal freeze	0x0	R/W	**ADC Automatic Calibration Freeze:** These bits, when set to '1', will enable the automatic calibration freeze logic for the respective ADC channel as follows: Bit 15: ADC 7 Bit 14: ADC 6 Bit 13: ADC 5 Bit 12: ADC 4 Bit 11: ADC 3 Bit 10: ADC 2 Bit 9: ADC 1 Bit 8: ADC 0
7:0	ADC cal freeze	0x0	R/W	**ADC Manual Calibration Freeze:** these bits, when set to '1', will force the respective ADC channel to freeze any further calibration cycles.

4.2 PL Event Pulse Register

This register initiates the pulse for the enabled PL Events (see Section 4.3). It is illustrated in Figure 4–2 and described in Table 4–3.

Figure 4–2: PL Event Pulse Register

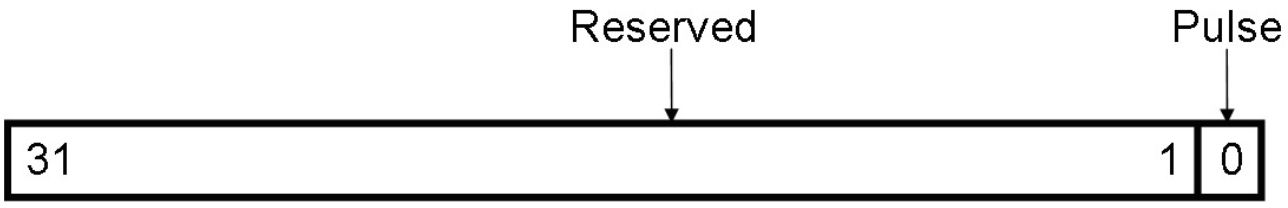


Table 4–3: PL Event Pulse Register (Base Address + 0x04)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	pulse	0	R/W	**PL Event Pulse:** When set to '1', will initiate a single, 1 x **adc_clk**–period wide pulse on each of the PL Event outputs that are enabled by the PL Event Enables, and have their respective Pulse Select set to "Register" ("01") (see Section 4.3).

4.3 PL Event Enable Register

This register enables the PL Events and sets the trigger source. It is illustrated in Figure 4–3 and described in Table 4–4.

Figure 4–3: PL Event Enable Register

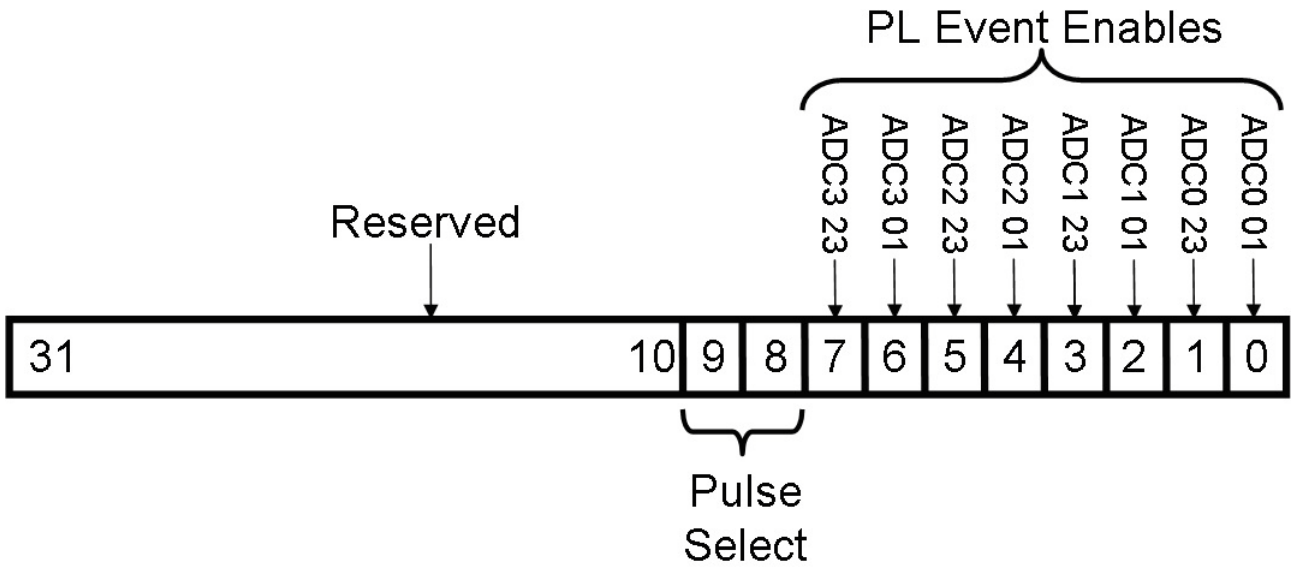


Table 4–4: PL Event Enable Register (Base Address + 0x08)

Bits	Field Name	Default Value	Access Type	Description
31:10	Reserved	N/A	N/A	**Reserved**
9:8	Pulse Select	00	R/W	**PL Event Pulse Source Select:** Selects the pulse source for the enabled PL Events as follows: "00" = OFF (no pulse) "01" = Register – pulse is initiated by control register (see Section 4.2) "10" = External pulse (**`pl_event_in`**) "11" = Pulse is initiated by a **`s_axis_ptctl`** SYNC
7:0	PL Event Enables	0x00	R/W	**PL Event Enables:** Enables the PL Events as follows: **`Bit 0: adc00_pl_event`** **`Bit 1: adc01_pl_event`** **`Bit 2: adc10_pl_event`** **`Bit 3: adc11_pl_event`** **`Bit 4: adc20_pl_event`** **`Bit 5: adc21_pl_event`** **`Bit 6: adc30_pl_event`** **`Bit 7: adc31_pl_event`**

4.4 ADC Clock Domain Manual Reset Register

This register provides a single−pulse reset signal that is synchronized to the ADC clock domain. It is illustrated in Figure 4−4 and described in Table 4−5.

Figure 4−4: ADC Clock Domain Manual Reset Register

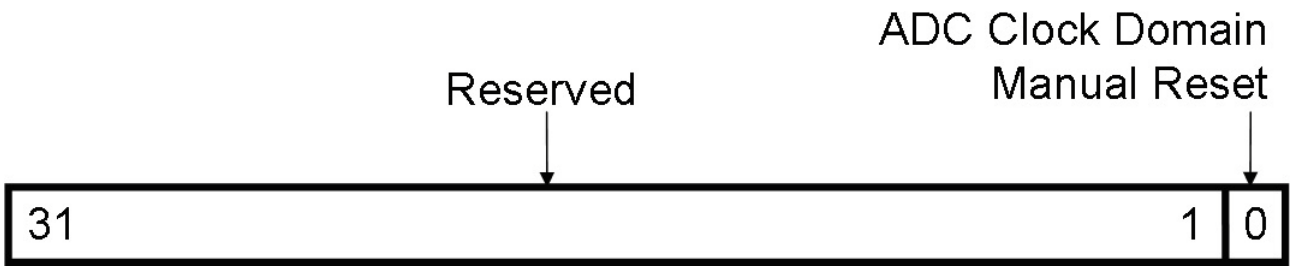


Table 4−5: ADC Clock Domain Manual Reset Register (Base Address + 0x0C)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	ADC Clock Domain Manual Reset	0	R/W	**ADC Clock Domain Manual Reset:** Setting this bit HIGH triggers an active−low pulse on the **`rf_adc_clk_domain_man_rstn`** output. The pulse is (at least) 3 x **`adc_aclk`** periods wide. The pulse asserts (LOW) asynchronously, but de−asserts (HIGH) synchronously to the **`adc_aclk`**.

4.5 Calibration Freeze Status Register

This register reports the status of the calibration freeze for each of the RF ADCs. It is illustrated in Figure 4–5 and described in Table 4–6.

Figure 4–5: Calibration Freeze Status Register

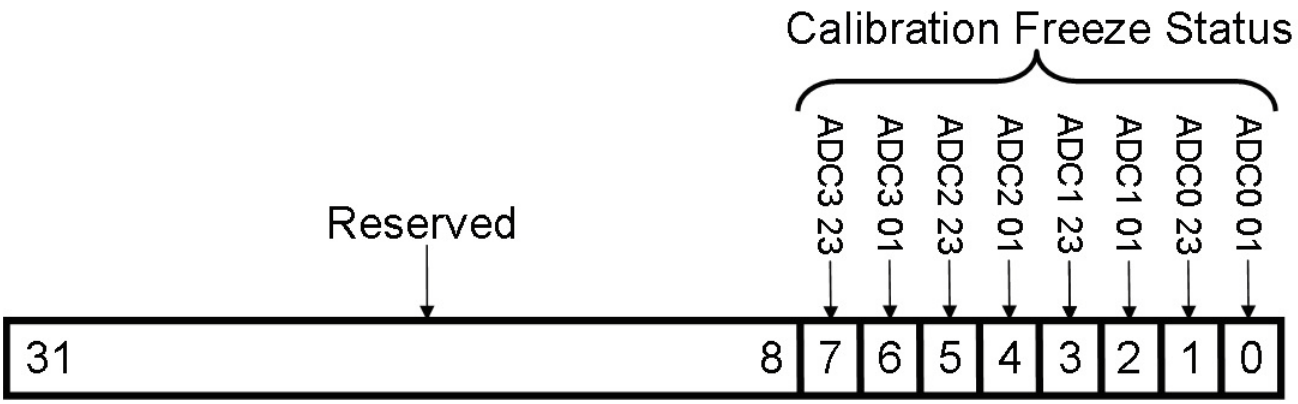


Table 4–6: Calibration Freeze Status Register (Base Address + 0x10)

Bits	Field Name	Default Value	Access Type	Description
31:8	Reserved	N/A	N/A	**Reserved**
7:0	Calibration Freeze Status	0x00	RO	**Calibration Freeze Status:** Reports the Calibration Freeze Status as follows: **Bit 0: adc0 01** **Bit 1: adc0 23** **Bit 2: adc1 01** **Bit 3: adc1 23** **Bit 4: adc2 01** **Bit 5: adc2 23** **Bit 6: adc3 01** **Bit 7: adc3 23**

4.6 Tvalid Status Register

This register reports the status of the "**tvalid**" signals for each of the RF ADCs. It is illustrated in Figure 4–6 and described in Table 4–7.

Figure 4–6: Tvalid Status Register

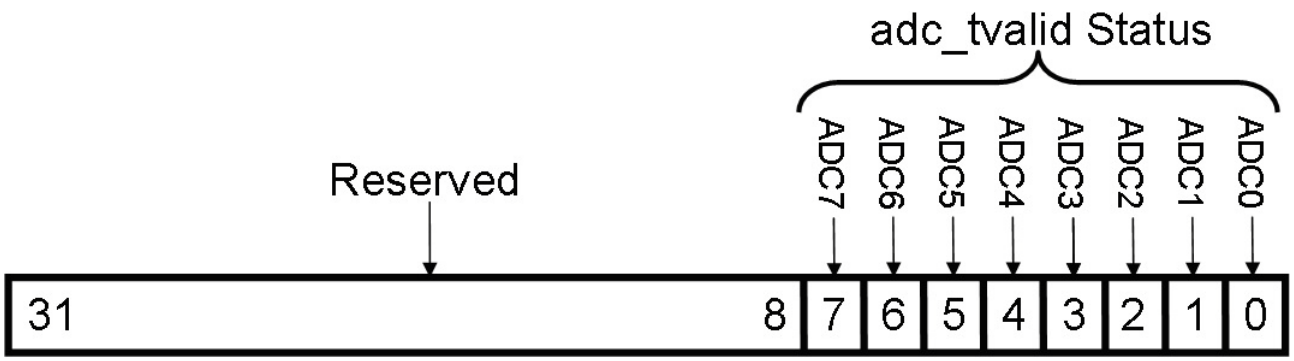


Table 4–7: Tvalid Status Register (Base Address + 0x14)

Bits	Field Name	Default Value	Access Type	Description
31:8	Reserved	N/A	N/A	**Reserved**
7:0	Tvalid Status	0x00	RO	**Tvalid Status:** Reports the status of the ADC **tvalid** inputs as follows: **Bit 0 (ADC0): s_axis_adc0_i_tvalid** **Bit 1 (ADC1): s_axis_adc1_i_tvalid** **Bit 2 (ADC2): s_axis_adc2_i_tvalid** **Bit 3 (ADC3): s_axis_adc3_i_tvalid** **Bit 4 (ADC4): s_axis_adc4_i_tvalid** **Bit 5 (ADC5): s_axis_adc5_i_tvalid** **Bit 6 (ADC6): s_axis_adc6_i_tvalid** **Bit 7 (ADC7): s_axis_adc7_i_tvalid**

4.7 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 4.8). It is illustrated in Figure 4–7 and described in Table 4–8.

Figure 4–7: Interrupt Enable Register

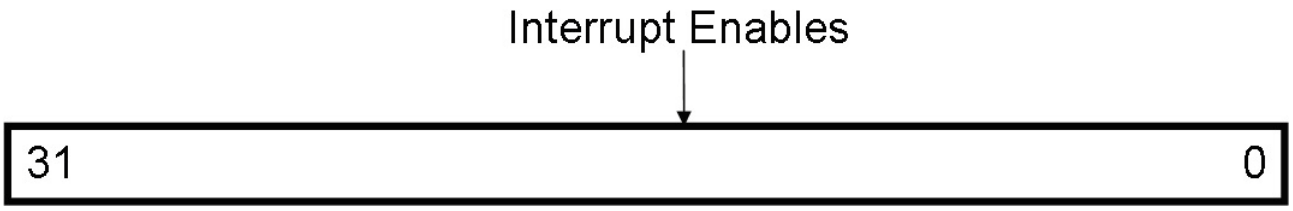


Table 4–8: Interrupt Enable Register (Base Address + 0x20)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:0	Interrupt Enables	0x0000	R/W	**Interrupt Enables:** Enable bits for the RF ADC out–of–range interrupts. 0 = Disable interrupt 1 = Enable interrupt The bits are mapped as follows: **Bit 0: adc0_01_over_range** **Bit 1: adc0_01_over_threshold1** **Bit 2: adc0_01_over_threshold2** **Bit 3: adc0_01_over_voltage** **Bit 4: adc0_23_over_range** **Bit 5: adc0_23_over_threshold1** **Bit 6: adc0_23_over_threshold2** **Bit 7: adc0_23_over_voltage** **Bit 8: adc1_01_over_range** **Bit 9: adc1_01_over_threshold1** **Bit 10: adc1_01_over_threshold2** **Bit 11: adc1_01_over_voltage** **Bit 12: adc1_23_over_range** **Bit 13: adc1_23_over_threshold1** **Bit 14: adc1_23_over_threshold2** **Bit 15: adc1_23_over_voltage** **Bit 16: adc2_01_over_range** **Bit 17: adc2_01_over_threshold1** **Bit 18: adc2_01_over_threshold2** **Bit 19: adc2_01_over_voltage** **Bit 20: adc2_23_over_range** **Bit 21: adc2_23_over_threshold1** **Bit 22: adc2_23_over_threshold2** **Bit 23: adc2_23_over_voltage** **Bit 24: adc3_01_over_range** **Bit 25: adc3_01_over_threshold1** **Bit 26: adc3_01_over_threshold2** **Bit 27: adc3_01_over_voltage** **Bit 28: adc3_23_over_range** **Bit 29: adc3_23_over_threshold1** **Bit 30: adc3_23_over_threshold2** **Bit 31: adc3_23_over_voltage**

4.8 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. It is illustrated in Figure 4−8 and described in Table 4−9.

Figure 4−8: Interrupt Status Register

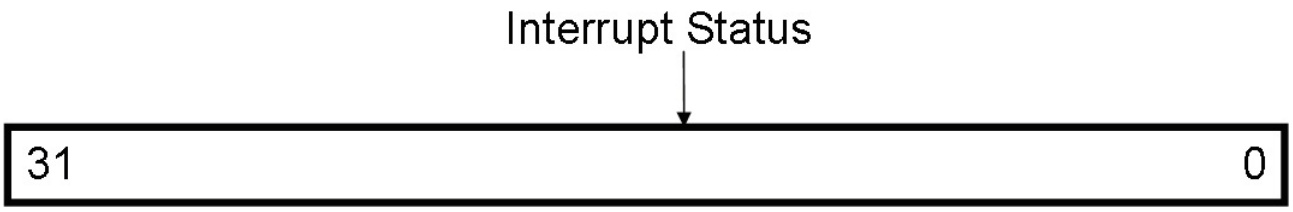


Table 4–9: Interrupt Status Register (Base Address + 0x24)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:0	Interrupt Status	0x0000	RO	**Interrupt Status:** Status bits for the RF ADC out–of–range interrupts. 0 = No interrupt 1 = Interrupt condition asserted The bits are mapped as follows: **Bit 0: adc0_01_over_range** **Bit 1: adc0_01_over_threshold1** **Bit 2: adc0_01_over_threshold2** **Bit 3: adc0_01_over_voltage** **Bit 4: adc0_23_over_range** **Bit 5: adc0_23_over_threshold1** **Bit 6: adc0_23_over_threshold2** **Bit 7: adc0_23_over_voltage** **Bit 8: adc1_01_over_range** **Bit 9: adc1_01_over_threshold1** **Bit 10: adc1_01_over_threshold2** **Bit 11: adc1_01_over_voltage** **Bit 12: adc1_23_over_range** **Bit 13: adc1_23_over_threshold1** **Bit 14: adc1_23_over_threshold2** **Bit 15: adc1_23_over_voltage** **Bit 16: adc2_01_over_range** **Bit 17: adc2_01_over_threshold1** **Bit 18: adc2_01_over_threshold2** **Bit 19: adc2_01_over_voltage** **Bit 20: adc2_23_over_range** **Bit 21: adc2_23_over_threshold1** **Bit 22: adc2_23_over_threshold2** **Bit 23: adc2_23_over_voltage** **Bit 24: adc3_01_over_range** **Bit 25: adc3_01_over_threshold1** **Bit 26: adc3_01_over_threshold2** **Bit 27: adc3_01_over_voltage** **Bit 28: adc3_23_over_range** **Bit 29: adc3_23_over_threshold1** **Bit 30: adc3_23_over_threshold2** **Bit 31: adc3_23_over_voltage**

4.9 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. It is illustrated in Figure 4–9 and described in Table 4–10.

Figure 4–9: Interrupt Flag Register

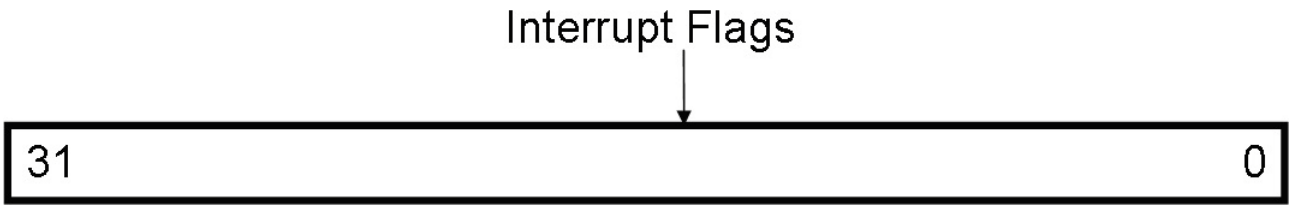


Table 4–10: Interrupt Flag Register (Base Address + 0x28)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:0	Interrupt Flags	0x0000	R/CLR	**Interrupt Flags:** Flag bits for the RF ADC out–of–range interrupts. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:**1 = Clear latch The bits are mapped as follows: **Bit 0: adc0_01_over_range** **Bit 1: adc0_01_over_threshold1** **Bit 2: adc0_01_over_threshold2** **Bit 3: adc0_01_over_voltage** **Bit 4: adc0_23_over_range** **Bit 5: adc0_23_over_threshold1** **Bit 6: adc0_23_over_threshold2** **Bit 7: adc0_23_over_voltage** **Bit 8: adc1_01_over_range** **Bit 9: adc1_01_over_threshold1** **Bit 10: adc1_01_over_threshold2** **Bit 11: adc1_01_over_voltage** **Bit 12: adc1_23_over_range** **Bit 13: adc1_23_over_threshold1** **Bit 14: adc1_23_over_threshold2** **Bit 15: adc1_23_over_voltage** **Bit 16: adc2_01_over_range** **Bit 17: adc2_01_over_threshold1** **Bit 18: adc2_01_over_threshold2** **Bit 19: adc2_01_over_voltage** **Bit 20: adc2_23_over_range** **Bit 21: adc2_23_over_threshold1** **Bit 22: adc2_23_over_threshold2** **Bit 23: adc2_23_over_voltage** **Bit 24: adc3_01_over_range** **Bit 25: adc3_01_over_threshold1** **Bit 26: adc3_01_over_threshold2** **Bit 27: adc3_01_over_voltage** **Bit 28: adc3_23_over_range** **Bit 29: adc3_23_over_threshold1** **Bit 30: adc3_23_over_threshold2** **Bit 31: adc3_23_over_voltage**

Chapter 5: Designing with the Core

This chapter provides guidelines and additional information to facilitate designing with the RF ADC Control & Status Core.

5.1 General Design Guidelines

The Pentek RF ADC Control & Status Core provides the required logic to control and monitor the Xilinx RFSoC ADCs. The user can customize the core by setting the generic parameters based on the application requirement as described in Section 2.5.

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.

ADC Clock: `adc_aclk`.

This clock is provided by the Xilinx RFSoC ADCs. It is used to clock all logic and I/O that are not directly associated with the AXI4−Lite interface.

5.3 Resets

CSR Reset: `s_axi_csr_aresetn`.

This is an active−low synchronous reset associated with the **`s_axi_csr_aclk`**. When asserted, all CSR state machines in the core are reset.

External Reset Output: `rf_adc_clk_domain_man_rstn`.

This output is a register−controlled reset, and is associated with the ADC Clock (**`adc_aclk`**). For further details on this reset see Section 4.4.

5.4 Interrupts

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

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

NOTE: Because this core uses edge–triggered interrupts, an interrupt condition which remains 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 (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 ADC Data Interfaces:** The **s_axis_adc<7:0>_<i,q>** and **m_axis_adc<7:0>_<i,q>** busses are the input and output data busses that carry the ADC data into and out of the core. These are standard AXI4–Stream type interfaces associated with the ADC clock (**adc_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 ADC Control & Status 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 status as needed.

5.7 Timing Diagrams

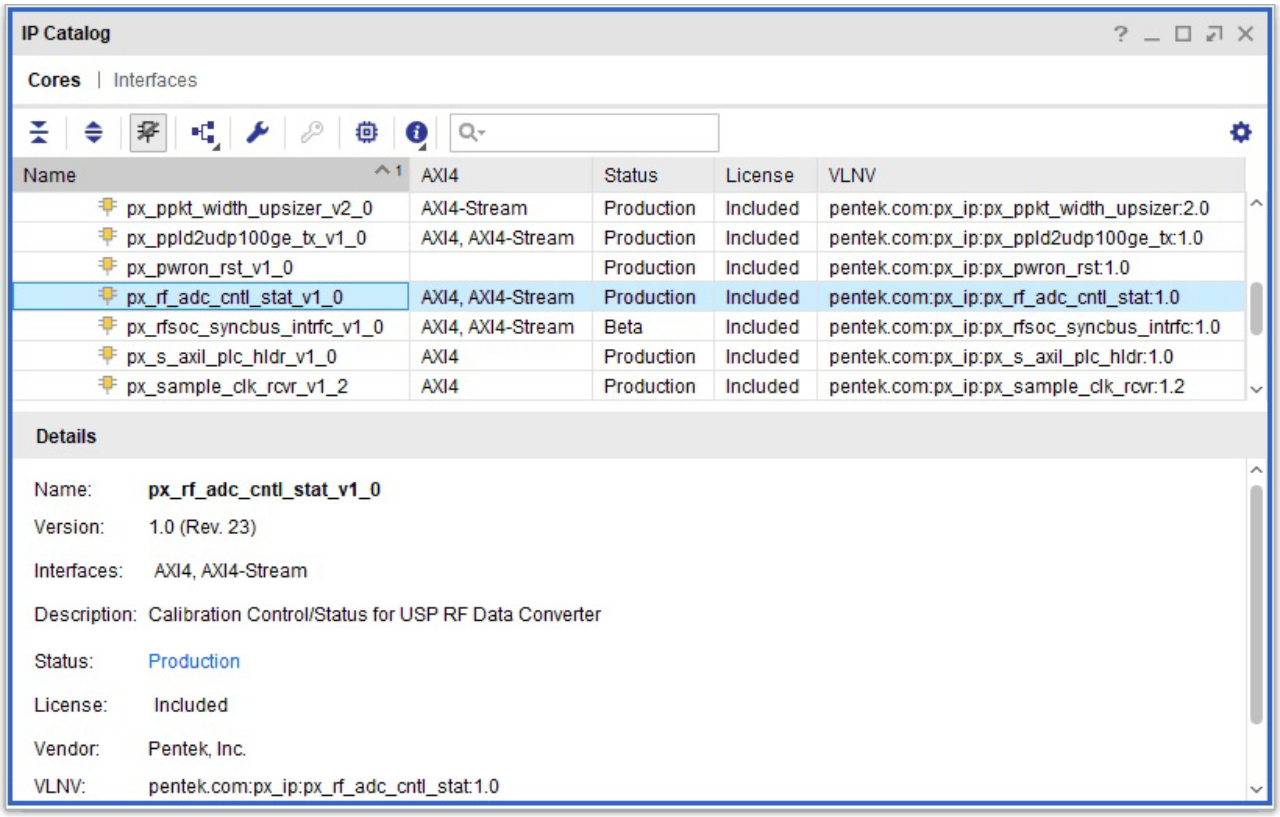
This section is not applicable to this IP core.

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek RF ADC Control & Status 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_adc_cntl_stat_v1_0`** as shown in Figure 6–1.

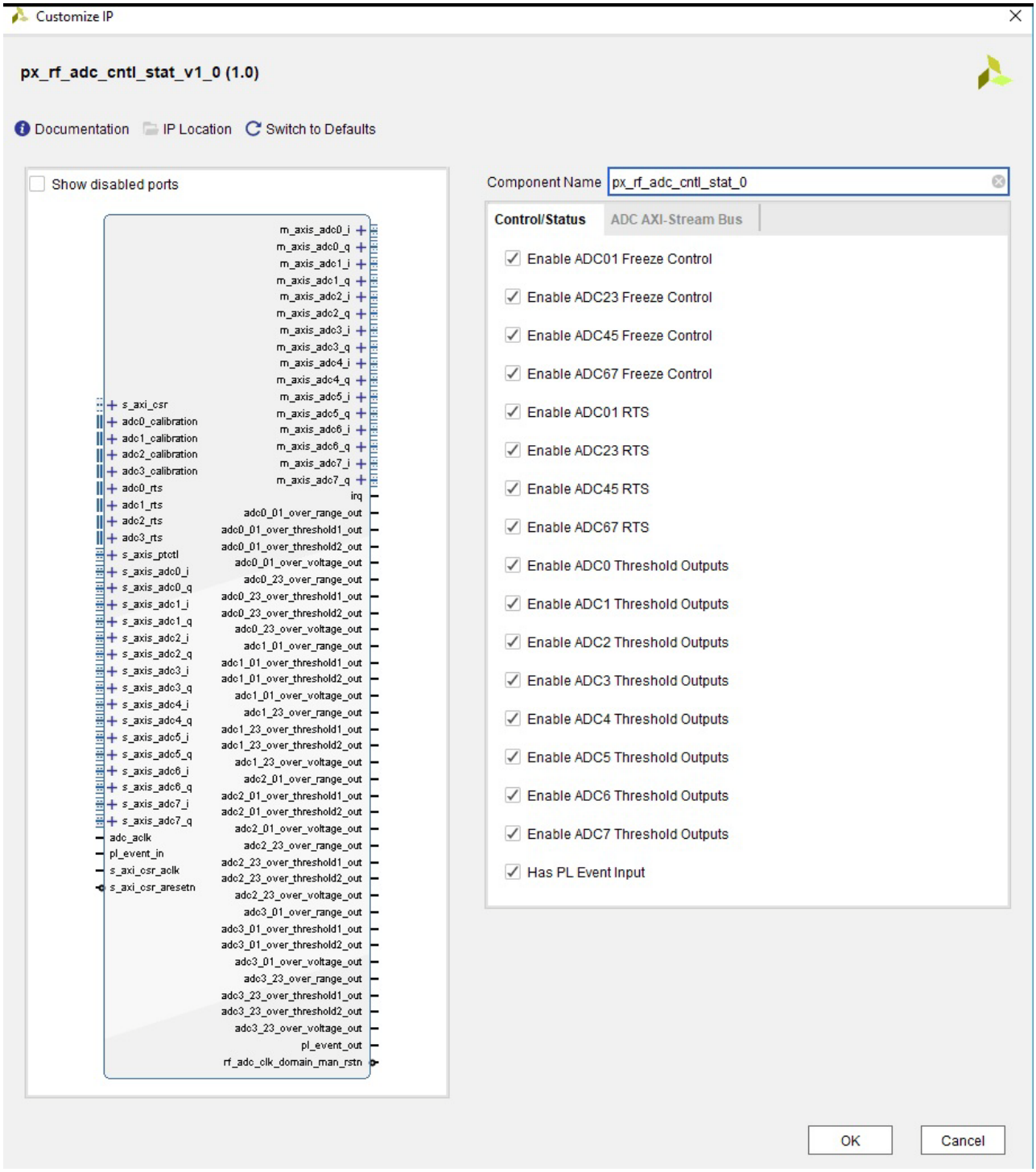
Figure 6–1: RF ADC Control & Status Core in Pentek IP Catalog



6.1 Pentek IP Catalog (continued)

When you select the **px_rf_adc_cntl_stat_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 ADC Control & Status Core IP Symbol



6.2 User Parameters

The user parameters of this RF ADC Control & Status Core are explained in Section 2.5 of this user manual.

6.3 Output Generation

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

6.4 Constraining the Core

This section contains information about constraining the RF ADC Control & Status Core in the Vivado Design Suite.

Required Constraints

The XDC constraints are not provided with the RF ADC Control & Status 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 FPGAs.

Clock Frequencies

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

The ADC clock (**`adc_aclk`**) is sourced from the Xilinx RFSoC ADC, and its frequency is defined by the setup of the ADC.

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*.