IP CORE MANUAL

Clock Frequency Counter IP

px_clock_frq_cntr

PENTEK

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

Copyright © 2017

Manual Part Number: 807.48404 Rev: 1.1 – September 26, 2017

Manual Revision History

Date	Version	Comments
9/7/17	1.0	Initial Release
9/26/17	1.1	Revised IP Facts, Sect 1.1, Sect 2.2.1, Sect 2.5, Table 3–2, Table 4–1, Table 4–3, Sect 5.2, Sect 6.1, and Sect 6.4.

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

Copyright © 2017, Pentek, Inc. All Rights Reserved. Contents of this publication may not be reproduced in any form without written permission.

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. Texas Instruments is a trademark of Texas Instruments, Incorporated. 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

IP Facts

Chapter 1: Overview

Chapter 2: General Product Specifications

Chapter 3: Port Descriptions

Table of Contents

Page

Chapter 4: Register Space

Chapter 5: Designing with the Core

Chapter 6: Design Flow Steps

IP Facts

Description

Pentek's Navigator™ Clock Frequency Counter Core is a clock frequency counter module that uses a user provided reference clock to calculate the frequency count of an input clock.

This core complies with the ARM® AMBA® AXI4 Specification. This user manual defines the hardware interface, software interface, and parameterization options for the Clock Frequency Counter Core.

Features

- Register access through AXI4–Lite interface that makes frequency value accessible to software
- Selectable reference clock frequency from 100 MHz to 250 MHz

Table 1–1: IP Facts Table

Core Specifics	
Supported Design Family[a]	Kintex® Ultrascale
Supported User Interfaces	AXI4–Lite
Resources	See Table 2–1
Provided with the Core	
Design Files	VHDL
Example Design	Not Provided
Test Bench	VHDL
Constraints File	Not Provided[b]
Simulation Model	VHDL
Supported S/W Driver	HAL Software Support
Tested Design Flows	
Design Entry	Vivado® Design Suite 2017.2 or later
Simulation	Vivado VSim
Synthesis	Vivado Synthesis
Support	
Provided by Pentek fpgasupport@pentek.com	

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

This page is intentionally blank

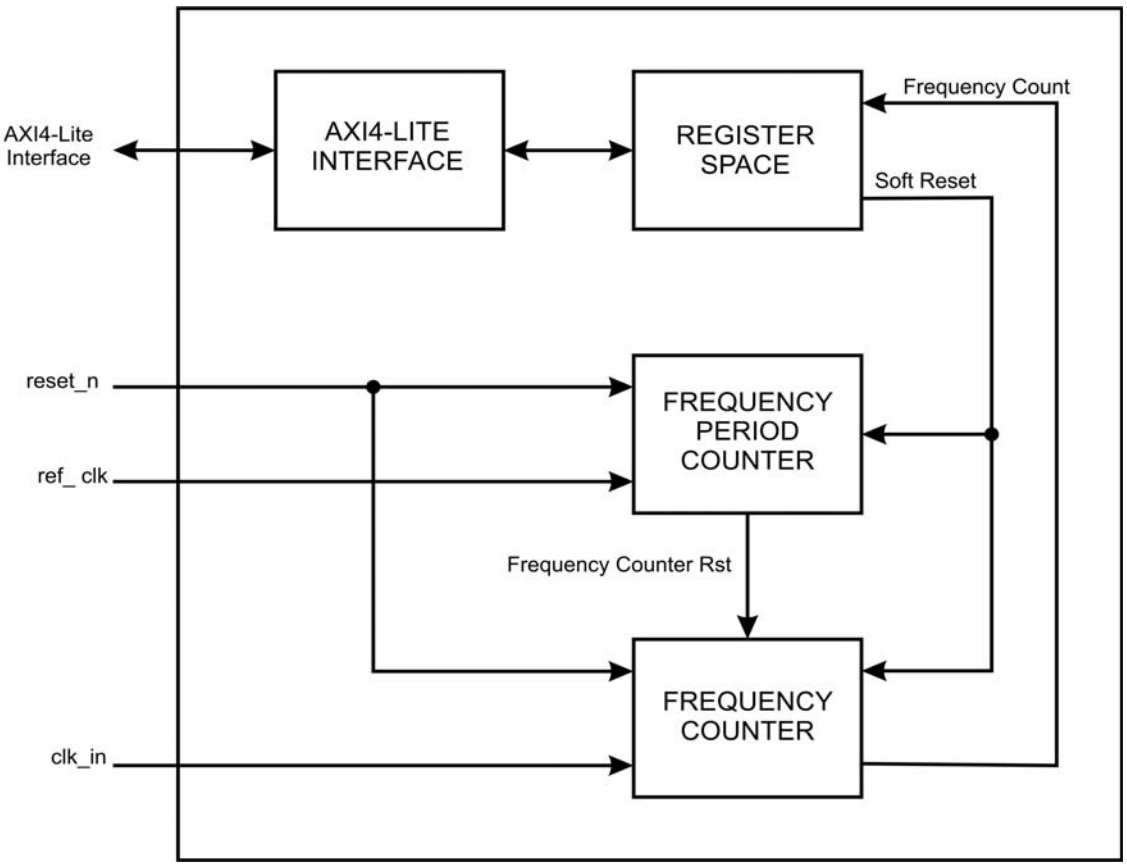
Chapter 1: Overview

1.1 Functional Description

Clock Frequency Counter Core is a clock frequency counter module that uses a refer−ence clock to calculate the frequency count of an input clock.

Figure 1−1 is a top−level block diagram of the Clock Frequency Counter Core. The modules in the block diagram are explained in other sections of this manual.

Figure 1−1: Clock Frequency Counter Core Block Diagram



- **AXI4−Lite Interface:** This module implements a 32−bit AXI4−Lite slave interface to access the register space. For more details about the AXI4−Lite Interface, refer to Section 3.1 AXI4−Lite Core Interfaces.

- **Register Space:** This module contains the control and status registers. Registers are accessed through the AXI4−Lite interface.

1.1 Functional Description (continued)

- **Freqency Period Counter:** This uses a user provided reference clock to generate a one second window. During this one second window, the Frequency Counter block counts the number of clock edges of the input clock. The counter is reset once per second.

- **Frequency Counter:** This counts the frequency of the input clock during a one second window and latches the count to a software register.

1.2 Applications

This core can be used to provide software with the frequency of a clock.

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) *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 Clock Frequency Counter Core has a bus interface that complies with the *ARM AMBA AXI4–Lite Protocol Specification*.

2.2 Performance

The performance of the Clock Frequency Counter Core is limited only by the FPGA logic speed. The values presented in this section should be used as an estimation guideline. Actual performance can vary.

2.2.1 Maximum Frequencies

This core requires a reference clock between 100MHz and 250MHz. The maximum frequency of the input clock is based on its clock constraint.

2.3 Resource Utilization

The resource utilization of the Clock Frequency Counter 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	73
Flip–Flops	211

NOTE: Actual utilization may vary based on the user design in which the Clock Frequency Counter Core is incorporated.

2.4 Limitations and Unsupported Features

This section is not applicable to this IP core.

2.5 Generic Parameters

Reference Clock Frequency: This parameter specifies what the frequency of the `ref_clk` is in Hz. It can be set to values between 100000000 and 250000000. See **Table 3–2**.

This page is intentionally blank

Chapter 3: Port Descriptions

This chapter provides port descriptions for the following interface type, and I/O signals:

- AXI4–Lite Core Interfaces
- I/O Signals

3.1 AXI4–Lite Core Interfaces

The Clock Frequency Counter 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 Clock Frequency Counter 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 details on the 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.
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 ADC Core.
s_axi_csr_awprot	Input	3	**Protection:** The ADC 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 ADC 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 ADC 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) Interface 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 ADC 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 ADC 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	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 ADC core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the ADC core.

Table 3-1: Control/Status Register (CSR) Interface 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 ADC 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 ADC 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 ADC 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 ADC 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 request output.

3.2 I/O Signals

The I/O port/signal descriptions of the top–level module of the Clock Frequency Counter Core are described in Table 3–2.

Table 3-2: I/O Signals

Port/Signal Name	Type	Direction	Description
Clock Signals			
clk_in	std_logic	Input	**Input Clock:** This is the clock that will be measured for frequency count.
ref_clk	std_logic	Input	**Reference Clock:** This clock is used to generate a one second window to measure **`clk_in`**. This parameter specifies what the frequency of the **`ref_clk`** is in Hz. It can be set to values between 100000000 and 250000000.
Other Control Signals			
reset_n	std_logic	Input	**Reset:** Hardware reset signal

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the register space of the Clock Frequency Counter Core. The memory map is provided in Table 4–1.

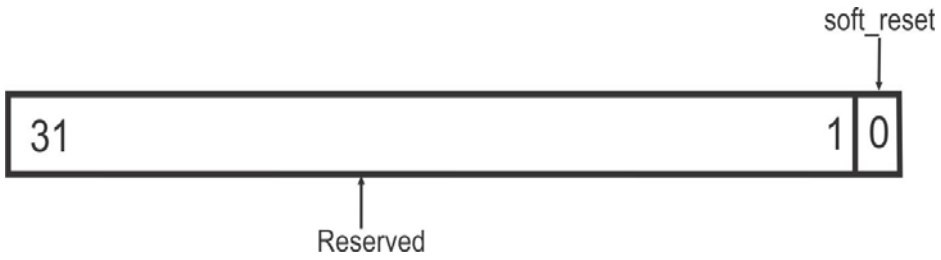
Table 4–1: Register Space Memory Map

Register Name	Address (Base Address + 0x0000)	Access	Description
Control Register	0x00	R/W	Control register
Frequency Counter Read Back Register	0x04	RO	Returns the frequency of the input clock in Hertz.

4.1 Control Register

This register is used to control the core. The Control Register is illustrated in Figure 4–1 and described in Table 4–2.

Figure 4–1: Control Register



.

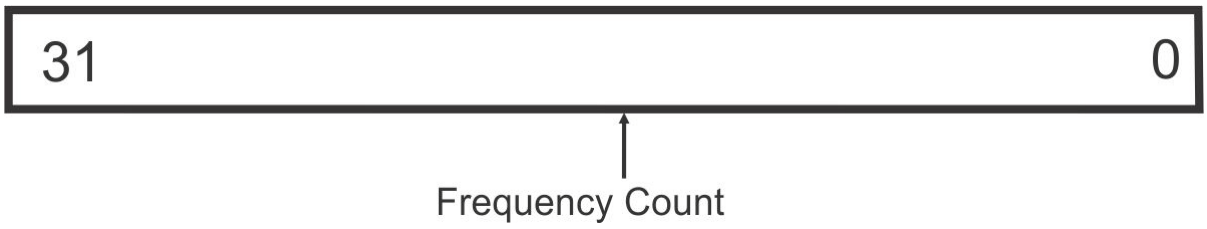
Table 4–2: Control Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	soft_reset	0	R/W	**Soft Reset:** This is toggled to reset the frequency counter. This is only required once.

4.2 Frequency Counter Read Back Register

This register returns the frequency count. The frequency count read back is updated once per second. It is necessary to wait one second for the value to be available after a reset. This register is illustrated in Figure 4–2 and described in Table 4–3.

Figure 4–2: Frequency Counter Read Back Register



.

Table 4–3: Frequency Counter Read Back Register (Base Address + 0x04)

Bits	Field Name	Default Value	Access Type	Description
31:0	frequency _count	0	R/W	**Frequency Count:** This provides read back of the frequency count. The frequency count is latched once per second. Upon reset, software must wait a minimum of one second before reading. The value is in Hertz.

This page is intentionally blank

Chapter 5: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the Clock Frequency Counter Core.

5.1 General Design Guidelines

The Clock Frequency Counter Pentek IP Core uses a 250 MHz clock to generate a one second window used to clock the number of clock edges of the input clock. This provides a frequency count accessible by a software register.

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.

Reference Clock: **ref_clk**

This clock is used to generate a one second window to calculate the frequency count of **Clk_in.**

Input Clock: **Clk_in**

This is the clock of interest. In this clock, the frequency is being counted.

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.

5.4 Interrupts

This section is not applicable for this IP core.

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 4 for the control register memory map and more details on the registers that can be accessed through this interface.

5.6 Programming Sequence

This section briefly describes the programming sequence for the Clock Frequency Counter Core.

1) Reset the frequency counter by toggling bit−0 at offset register 0x00.

2) The frequency count can be read back as often as possible after a minimum of one second after reset at offset register 0x04.

5.7 Timing Diagrams

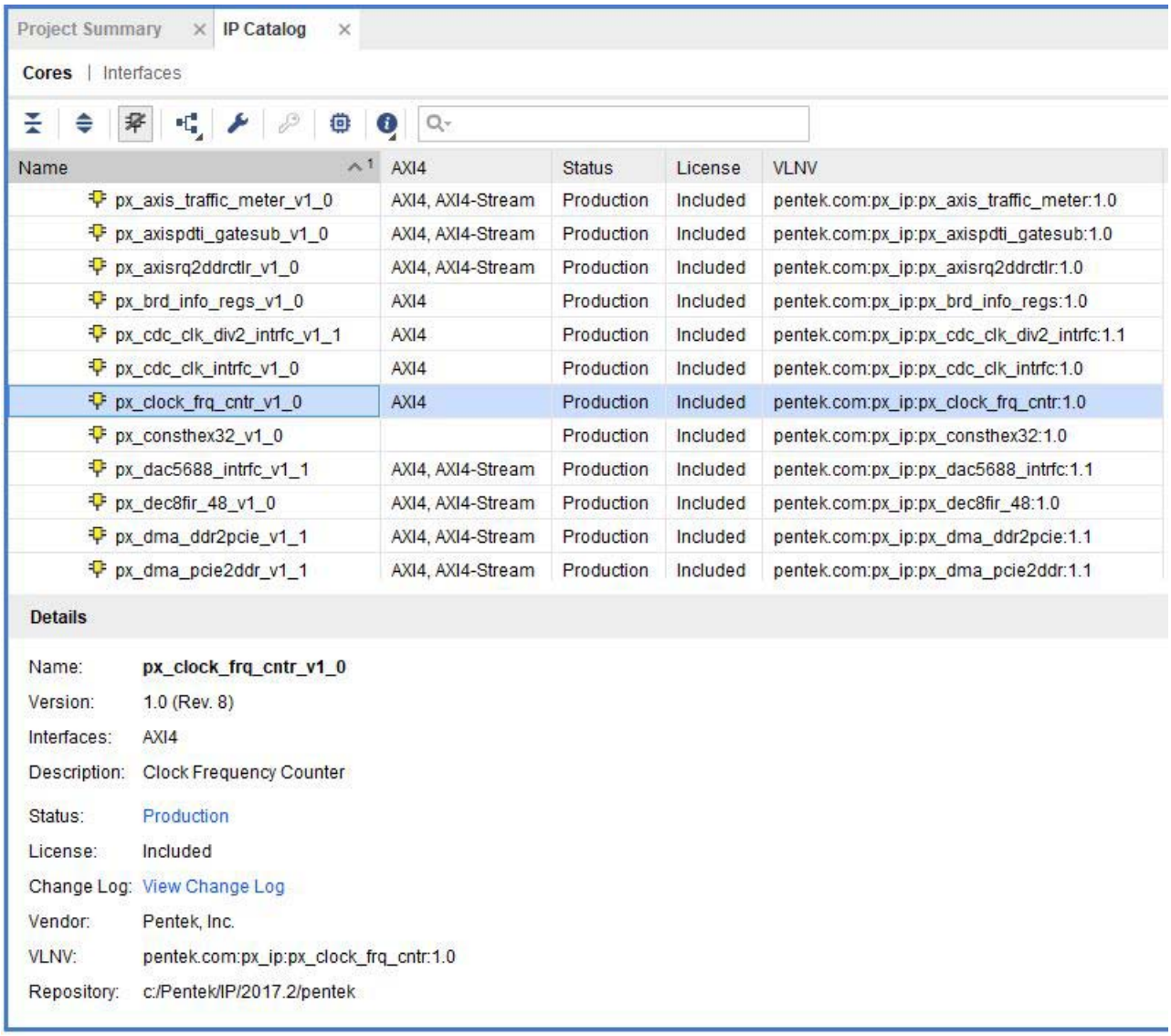
This test bench requires a lengthy simulation, thus timing diagrams are not provided. Test bench is provided and can be launched from the IP Core Vivado VSim project.

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek Clock Frequency Counter Core. It also includes 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_clock_frq_cntr_v1_0** as shown in Figure 6–1.

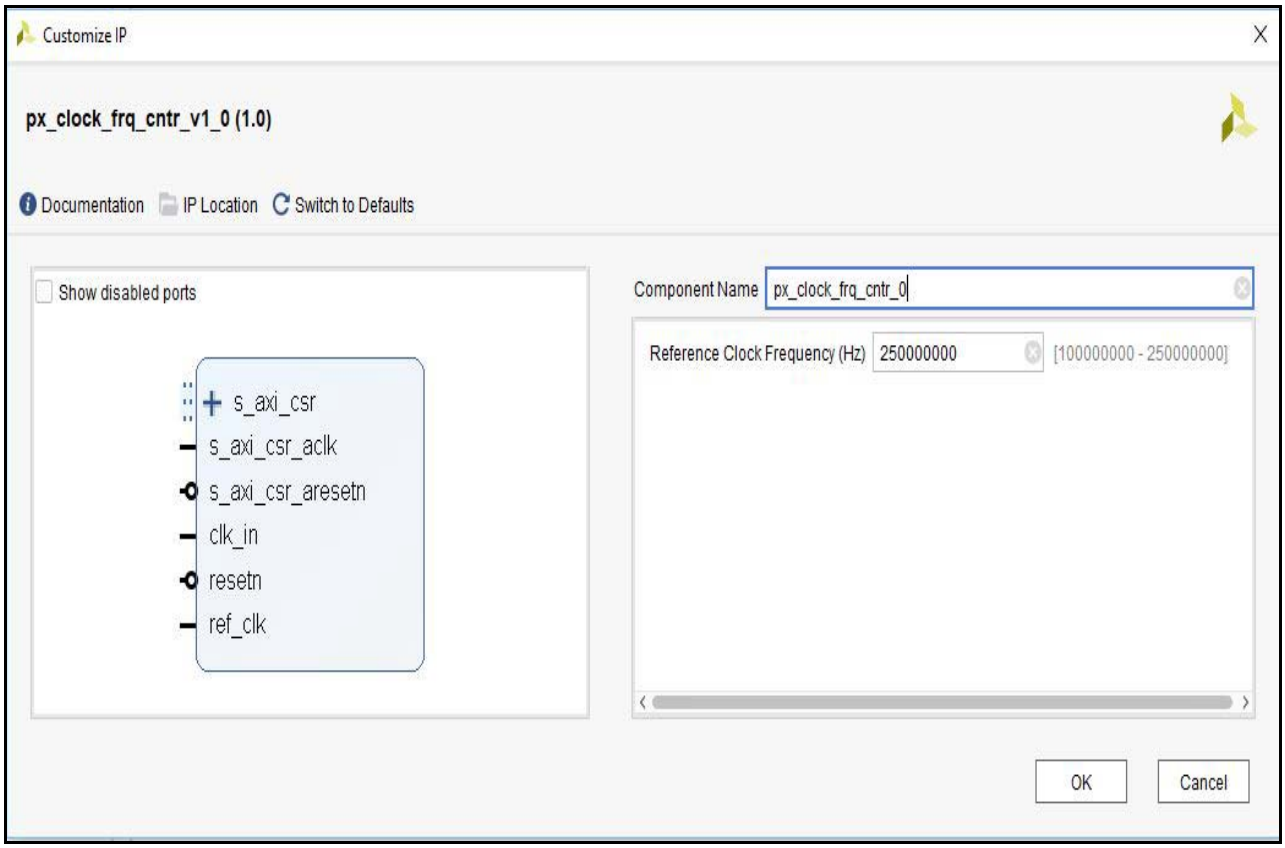
Figure 6–1: Clock Frequency Counter Core in Pentek IP Catalog



6.1 Pentek IP Catalog (continued)

When you select the **px_clock_frq_cntrc_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: Clock Frequency Counter Core IP Symbol



6.2 User Parameters

This section is not applicable for 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 Kintex Ultrascale FPGAs.

Clock Frequencies

The maximum clock frequency for this IP core is determined by the clock constraint of **Clk_In**. The AXI4−Lite interface clock (**s_axi_csr_aclk**) frequency is 250 MHz. **ref_clk** should be constrained to its frequency value and should be less than or equal to 250 Mhz.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking and Placement

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

6.5 Simulation

This test bench requires a lengthy simulation, thus timing diagrams are not provided. Test bench is provided and can be launched from the IP Core Vivado VSim project.

6.6 Synthesis and Implementation

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