IP CORE MANUAL

Sample Clock Receiver IP

px_sample_clk_rcvr

PENTEK

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

Manual Part Number: 807.48360 Rev: 1.1 – June 08, 2017

Manual Revision History

Date	Version	Comments
02/02/17	1.0	Initial Release
6/8/17	1.1	Updated to remove CDCM7005 Clock Generator

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. 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

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™ Sample Clock Receiver Core serves as an interface to the front panel differential clock input and generates a Sample Clock signal from the differential clock output of the front panel clock input.

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

Features

- Generates a Sample clock output from an input differential clock from the front panel clock input
- Register access through AXI4–Lite interface
- Replaces the Sample clock output with an incoming AXI system clock when the front panel clock input is unavailable
- Includes a FIFO to indicate the clock cycle count of the incoming differential clock signal from the front panel clock input

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	Not Provided
Constraints File	Not Provided[b]
Simulation Model	N/A
Supported S/W Driver	HAL Software Support
Tested Design Flows	
Design Entry	Vivado® Design Suite 2016.4 or later
Simulation	Vivado VSim
Synthesis	Vivado Synthesis
Support	
Provided by Pentek fpgasupport@pentek.com	

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

This page is intentionally blank

Chapter 1: Overview

1.1 Functional Description

The Sample Clock Receiver Core provides a transaction interface between the front panel differential clock input and other modules that run in the Sample clock domain. This core has an AXI4−Lite Slave Interface and is connected to the Register Space as shown in Figure 1−1.

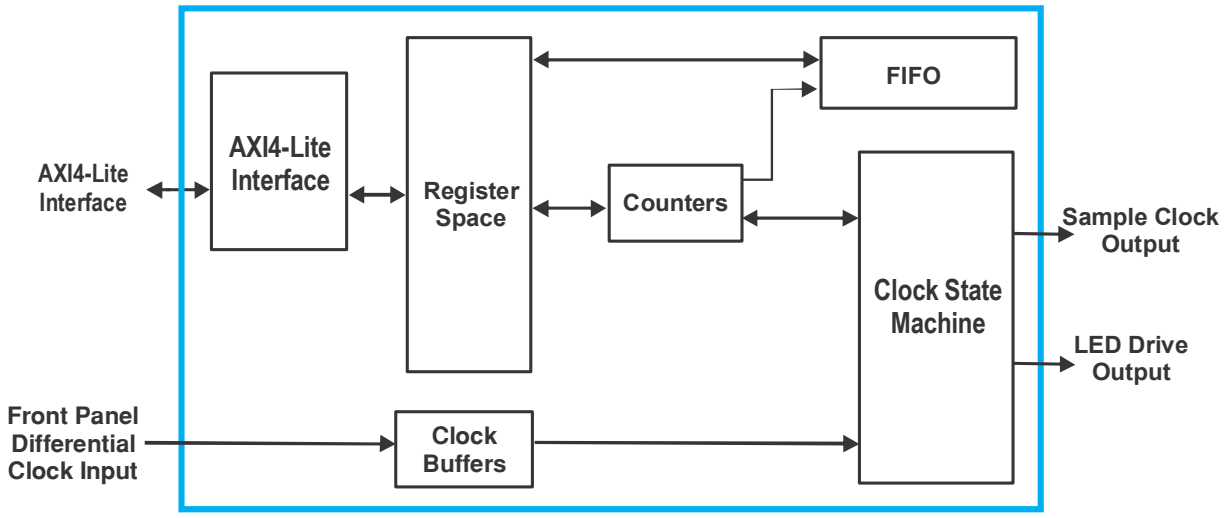
The core accepts an input **differential clock signal** from the front panel clock input, which is used to generate the **sample clock output**. The **Register Space** is connected to a **state machine** through **counters**. The state machine controls the sample clock output of the core. The **FIFO** is used to indicate the clock cycle count of the sample clock when there is a valid input differential clock from the front panel clock input, which can be used to determine its frequency.

On reset, the Sample Clock Reciever Core generates a sample clock output equivalent to half of the AXI bus clock frequency (**`s_axi_csr_aclk`**), which is then switched to the input differential clock signal frequency after it is detected by the core.

Figure 1−1 is a top−level block diagram of the Sample Clock Receiver Core. The mod−ules in the block diagram are explained in other sections of this manual.

- **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.

Figure 1−1: Sample Clock Receiver Core Block Diagram



1.1 Functional Description (continued)

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

- **Counters:** This module includes the timeout counters and frequency counters for the Sample Clock Receiver Core. This core has two 8−bit timeout counters which count the AXI clock cycles and front panel clock input cycles. This core also has a 31−bit AXI clock frequency counter, and a 32−bit sample clock frequency counter which starts after a valid clock input is detected from the front panel clock input.

- **Clock State Machine:** This state machine is used to control the Sample clock output of the core. The state machine determines if a valid clock is available at the differential clock input of the core coming from the front panel clock input. When there is no valid clock at the input, the sample clock output of the core is half the frequency of the AXI clock (**`s_axi_csr_aclk`**), referred to as the keepalive clock. When a valid input clock is available, it generates the sample clock from the input differential clock. This state machine has eight states which are explained as follows:

  - **Reset State:** When the incoming reset signal (**`s_axi_aresetn`**) is Low, the state machine is held in the Reset state, where all the signals are set to 0 and the sample clock output is equivalent to the keepalive clock frequency.

  - **Start Timeout Counter State:** When the reset signal (**`s_axi_csr_ asresetn`**) is set to High, the state machine moves from the Reset state to the Start Timeout counter state, where the AXI clock timeout counter starts incrementing based on the **`s_axi_csr_aclk`** and the clock timeout counter increments based on the incoming differential clock signal from the front panel clock input.

  - **Keepalive State:** The state machine moves to the Keepalive state after the Start Timeout Counter state. In this state, the sample clock output is the keepalive clock. A valid differential clock input is detected when the clock timeout counter reaches its maximum value. When a valid differential clock is detected, the state machine moves to the Good Clock state. When a differential clock is not available, the state machine moves to the next wait state after the AXI clock cycle timeout counter reaches its maximum value.

  - **Timeout State Wait1, Wait2, Wait 3:** These are the three wait states for the state machine after the Keepalive state when a valid clock is not present at the differential clock input. After these wait states the state machine moves back to the Start Timeout Counter state. At Timeout State Wait 3 the AXI clock timeout counter is reset to the default value of '0'.

  - **Good Clock State:** When a valid clock is detected at the differential clock input from the front panel clock input, the state machine moves from the Keepalive state to the Good Clock state and remains in this state as long as the input differential clock is present. In this state, the core generates a sample clock output from the incoming differential clock signal. When the input differential clock is not available, the state machine moves back to the Start Timeout Counter state.

1.1 Functional Description (continued)

- **FIFO:** This is a clock domain crossing FIFO, which is used to return the value of the sample clock frequency counter to the Status Register in the Register Space. This FIFO receives the sample clock frequency counter value in the sample clock domain and delivers the value to the Register Space in the AXI clock (**`s_axi_csr_aclk`**) domain.

1.2 Applications

This core can be used for interfacing any Kintex Ultrascale FPGA to the front panel clock input across an AXI4−Lite Interface. The Sample Clock Receiver Core is used to generate sample clocks on Pentek product families such as the Jade XMC modules.

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

This page is intentionally blank

Chapter 2: General Product Specifications

2.1 Standards

The Sample Clock Receiver Core has a bus interface that complies with the *ARM AMBA AXI4–Lite Protocol Specification*.

2.2 Performance

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

2.2.1 Maximum Frequencies

The Sample Clock Receiver Core is designed to meet a target frequency of 250 MHz on a Kintex Ultrascale –2 speed grade FPGA. 250 MHz is typically the PCI Express® (PCIe®) AXI Bus clock frequency.

2.3 Resource Utilization

The resource utilization of the Sample Clock Receiver 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	199
Flip–Flops	477
Memory LUT	24

NOTE: Actual utilization may vary based on the way in which the Sample Clock Receiver 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 Sample Clock Receiver Core are described in Table 2–2. These parameters can be set as required by the user while customizing the core.

Table 2-2: Generic Parameters

Port/Signal Name	Type	Description
differential_term	String	**Differential Termination:** Differential termination for the differential clock inputs. TERM_NONE = No differential termination for the inputs TERM_100 = 100 Ω differential termination
ibuf_low_pwr	Boolean	**Input Buffer Low Power:** Sets the input buffer performance. True = Input buffer implemented in low power mode; False = Input buffer implemented in high performance mode

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 Sample Clock Receiver 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 Sample Clock Receiver 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_aclk	Input	1	**Clock**
s_axi_aresetn	Input	1	**Reset:** Active low. This value will reset all control registers to their initial states.
s_axi_awaddr	Input	5	**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 Sample Clock Receiver Core.
s_axi_awprot	Input	3	**Protection:** The Sample Clock Receiver Core ignores these bits.
s_axi_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 Sample Clock Receiver 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_awready	Output	1	**Write Address Ready:** This output is asserted by the Sample Clock Receiver 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_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_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, indicates that the corresponding byte on **s_axi_csr_wdata** contains valid data. Bit 0 corresponds to the least significant byte, and bit 3 to the most significant byte.
s_axi_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_wready	Output	1	**Write Ready:** This signal is asserted by the Sample Clock Receiver 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_bresp	Output	2	**Write Response:** The Sample Clock Receiver Core indicates the 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_bvalid	Output	1	**Write Response Valid:** This signal is asserted by the 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_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_araddr	Input	5	**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 Sample Clock Receiver Core.
s_axi_arprot	Input	3	**Protection:** These bits are ignored by the Sample Clock Receiver Core
s_axi_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 Sample Clock Receiver Core asserts **s_axi_csr_arready** when it ready to accept the Read Address. The **s_axi_csr_arvalid** must be remain asserted until the rising clock edge after the assertion of **s_axi_csr_arready**.

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

Port	Direction	Width	Description
s_axi_arready	Output	1	**Read Address Ready:** This output is asserted by the Sample Clock Receiver 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_rdata	Output	32	**Read Data:** This value is the data read from the address specified by **s_axi_csr_araddr** when **s_axi_csr_arvalid** and **s_axi_csr_arready** are high on the same cycle.
s_axi_rresp	Output	2	**Read Response:** The Sample Clock Receiver Core indicates the 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_rready	Input	1	**Read Data Ready:** This signal is asserted by the user logic when it is ready to accept the Read Data.
s_axi_csr_rvalid	Output	1	**Read Data Valid:** This signal is asserted by the 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.
irq	Output	1	**Interrupt:** This is an active High, edge-type interrupt output.

3.2 I/O Signals

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

Table 3-2: I/O Signals

Port/Signal Name	Type	Direction	Description
Clock Input from Front Panel			
clk_in_p	std_logic	Input	**Differential Clock input from the Front Panel**
clk_in_n			
Clock Output			
sample_clk	std_logic	Output	**Sample Clock Output**
LED Drives			
led_n	std_logic	Output	**LED Output:** This LED drive signal indicates the status of the differential clock input from the front panel. Active low.
clk_good			**Clock Good:** When High, this output indicates that there is a valid differential clock input to the core from the front panel.
Reference Clock			
refclk	std_logic	Input	**Reference Clock:** 200 MHz

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the Register Space of the Sample Clock Receiver Core. The memory map is provided in Table 4–1.

Table 4–1: Register Space Memory Map

Register Name	Address (Base Address +)	Access	Description
Control Register	0x00	R/W	Control register to control the Clock disable signal of the core.
Clock Status Register	0x04	R	Status register to indicate the clock status in the Sample Clock Receiver Core.
Frequency Counter Register	0x08		Status Register to indicate the frequency of the output sample clock.
Interrupt Enable Register	0x0C	R/W	Interrupt enable bits
Interrupt Status Register	0x10	R	Interrupt source status bits
Interrupt Flag Register	0x14	R/Clr	Interrupt flag bits

4.1 Control Register

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

Figure 4–1: Control Register

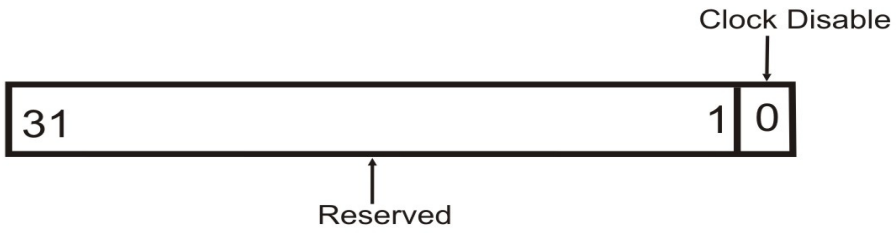


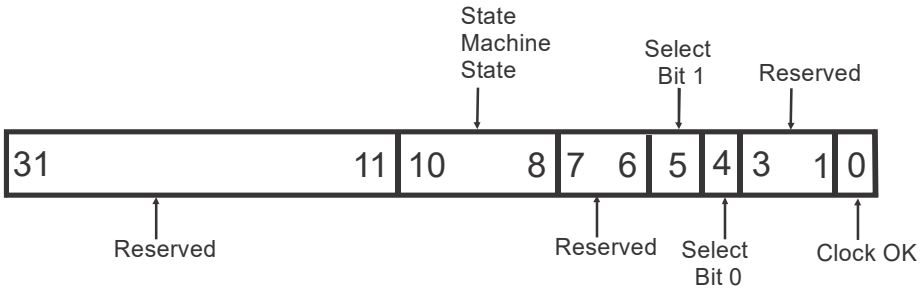
Table 4–2: Control Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	clock_disable	0	R/W	**Clock Disable:** This bit is used to disable the clock input from generating a sample clock output. 0 = Clock enabled 1 = Clock disabled

4.2 Clock Status Register

This register indicates the status of the state machine, select signals of the global clock buffer, and the status of the clock input to the core from the front panel. This register is illustrated in Figure 4–2 and described in Table 4–3.

Figure 4–2: Clock Status Register



.

Table 4–3: Clock Status Register (Base Address + 0x04)

Bits	Field Name	Default Value	Access Type	Description
31:11	Reserved	N/A	N/A	**Reserved**
10:8	sm_state	000	R	**State Machine State:** These bits indicate the current state of the clock state machine. For a detailed explanation of the state machine and its states refer to **Section 1.1** 000 – Reset State 001 – Start Timeout Counter State 010 – Keepalive State 011 – Timeout State 100 – Timeout State Wait1 101 – Timeout State Wait2 110 – Timeout State Wait3 111 – Good Clock State
7:6	Reserved	N/A	N/A	**Reserved**

Table 4–3: Clock Status Register (Base Address + 0x04) (Continued)

Bits	Field Name	Default Value	Access Type	Description
5	s1	0	R	**Select Bit 1 of the Global Clock Buffer:** This is the select bit of the global clock buffer (Xilinx Primitive) at the input I1. For this core the I1 input is the clock input from the front panel after passing through the differential clock buffer. 0 = Disabled 1 = Enabled
4	s0	0	R	**Select Bit 0 of the Global Clock Buffer:** This is the select bit of the global clock buffer at the input I0. For this core the I0 input is the keepalive clock generated from the input AXI clock (**s_axi_csr_aclk**). 0 = Disabled 1 = Enabled
3:1	Reserved	N/A	N/A	**Reserved**
0	clk_ok	0	R	**Clock OK:** This bit indicates status of the differential clock input from the front panel. 0 = Differential clock input not valid 1 = Differential clock input valid

4.3 Frequency Counter Register

This register indicates the clock cycle count of the sample clock signal, for a time period which is equivalent to the time taken by the 32−bit reference clock frequency counter to count from its initial value of 0x00000000 to the maximum value of 0x3FFFFFFF before being reset. The sample clock frequency counter is initiated after the differential clock input to the core is detected and the sample clock output is generated at the clock frequency of the output clock. This value is used to determine the sample clock frequency.

The sample clock frequency can be calculated by taking the ratio of the frequency counter value to the reference clock cycle count (0x3FFFFFFF) and multiplying the result with the reference clock frequency, which is 200 MHz. For example, if the sample clock is 200 MHz, then the frequency counter register will indicate a value of 0x3FFFFFFF. This register is illustrated in Figure 4−3 and described in Table 4−4.

Figure 4−3: Frequency Counter Register

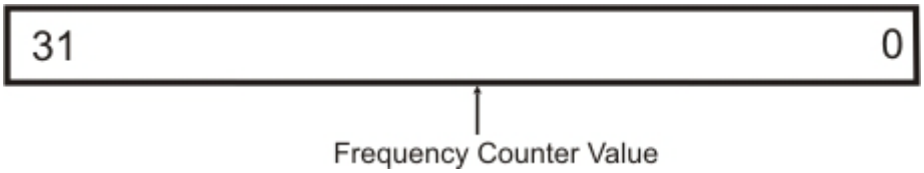


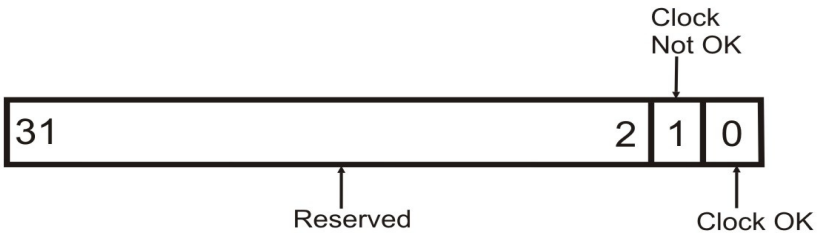
Table 4−4: Frequency Counter Register (Base Address + 0x08)

Bits	Field Name	Default Value	Access Type	Description
31:0	freq_cnt	0x00000000	R	**Frequency Counter Value:** This is the clock cycle count of the sample clock output of the core.

4.4 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 4.5). This register is illustrated in Figure 4–4 and described in Table 4–5.

Figure 4–4: Interrupt Enable Register



.

Table 4–5: Interrupt Enable Register (Base Address + 0x0C)

Bits	Field Name	Default Value	Access Type	Description
31:2	Reserved	N/A	N/A	**Reserved**
1	clk_not_ok	0	R/W	**Clock Not OK:** This bit enables/ disables the clock not OK interrupt source. The clock not OK interrupt source indicates that a valid clock is not present at the differential clock input from the front panel. 0 = Disable interrupt 1 = Enable interrupt
0	clk_ok	0	R/W	**Clock OK:** This bit enables/ disables the clock OK interrupt source. The clock OK interrupt source indicates that a valid clock is present at the differential clock input from the front panel. 0 = Disable interrupt 1 = Enable interrupt

4.5 Interrupt Status Register

The Interrupt Status Register has read–only access associated with each interrupt condition. A status bit changes to '1' when the source interrupt occurs. When a status bit in this register changes to '1,' the corresponding flag bit in the Interrupt Flag Register is set to '1'. A status bit in this register clears to '0' when that interrupt condition clears, whereas the associated flag bit in the Interrupt Flag Register remains at logic '1' until it is explicitly cleared by the user.

Some of the interrupt sources are transient and so may not appear in the Interrupt Status Register at the time it is read. In such cases use the Interrupt Flag Register to see the interrupt conditions that have occurred. The Interrupt Status Register's bits are shown in Figure 4–5 and described in Table 4–6.

Figure 4–5: Interrupt Status Register

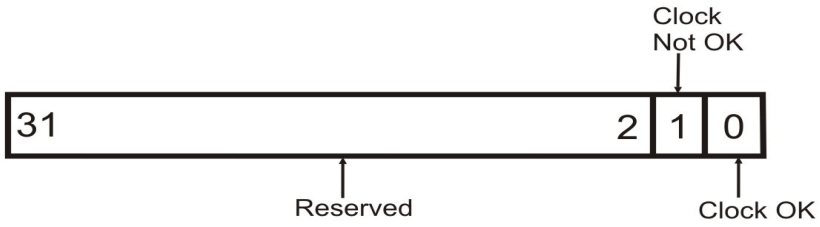


Table 4–6: Interrupt Status Register (Base Address + 0x10)

Bits	Field Name	Default Value	Access Type	Description
31:2	Reserved	N/A	N/A	**Reserved**
1	clk_not_ok	0	R	**Clock Not OK:** This bit indicates the status of the clock not OK interrupt source. The clock not OK interrupt source indicates that a valid clock is not present at the differential clock input from the front panel. 0 = No interrupt 1 = Interrupt condition asserted
0	clk_ok	0	R	**Clock OK:** This bit indicates the status of the clock OK interrupt source. The clock OK interrupt source indicates that a valid clock is present at the differential clock input from the front panel. 0 = No interrupt 1 = Interrupt condition asserted

4.6 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 enable register. The Interrupt Flag Register's bits are shown in Figure 4–6 and described in Table 4–7.

Figure 4–6: Interrupt Flag Register

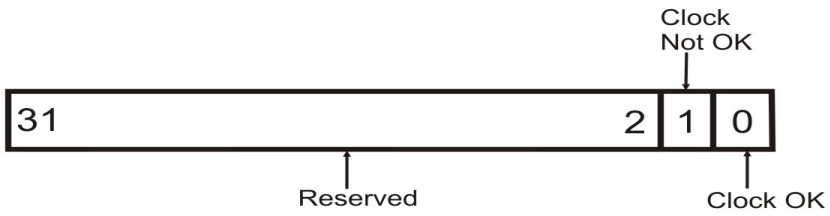


Table 4–7: Interrupt Flag Register (Base Address + 0x14)

Bits	Field Name	Default Value	Access Type	Description
31:4	Reserved	N/A	N/A	**Reserved**
1	clk_not_ok	0	R/Clr	**Clock Not OK:** This bit indicates the clock not OK interrupt flag. The clock not OK interrupt source indicates that a valid clock is not present at the differential clock input from the front panel. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
0	clk_ok	0	R/Clr	**Clock OK:** This bit indicates the clock OK interrupt flag. The clock OK interrupt source indicates that a valid clock is present at the differential clock input from the front panel. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch

Chapter 5: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the Sample Clock Receiver Core.

5.1 General Design Guidelines

The Sample Clock Receiver Core is used as an interface to the front panel clock input, to control the generation of the sample clock output. This core delivers a sample clock equivalent to the keepalive clock (Frequency = AXI clock Frequency/2) at reset and also when the differential clock input is not available to this IP core. When a differential clock input is available to the core, it generates a sample clock output equivalent to the differential clock frequency.

The Sample Clock Receiver Core also includes a FIFO which indicates the number of clock cycles of the sample clock when the input differential clock is available. The FIFO indicates the sample clock count between every reset of the reference clock frequency counter. The sample count value of the FIFO is stored in the Frequency Counter register.

5.2 Clocking

Main Clock: **s_axi_csr_aclk**

This clock is used to clock all the ports on the core, including the control/status register (CSR) interface.

5.3 Resets

Control/Status 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 core has an edge−type (rising edge−triggered) interrupt output. It 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 its **irq** output. Each interrupt event is stored in two registers, accessible on the **s_axi_csr** bus.

5.4 Interrupts (continued)

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

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

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

5.5 Interface Operation

CSR Interface: This is the Control/Status Register Interface. It is associated with the **`s_axi_csr_aclk`**. It is a standard AXI4−Lite type interface. See Chapter 4 for the 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 Sample Clock Receiver Core.

1) Ensure that the Interrupt Flag Register is cleared.

2) Enable your selected interrupt bits in the Interrupt Enable Register.

3) Observe the output, check the Interrupt Flag Register, and process and clear all interrupts.

5.7 Timing Diagrams

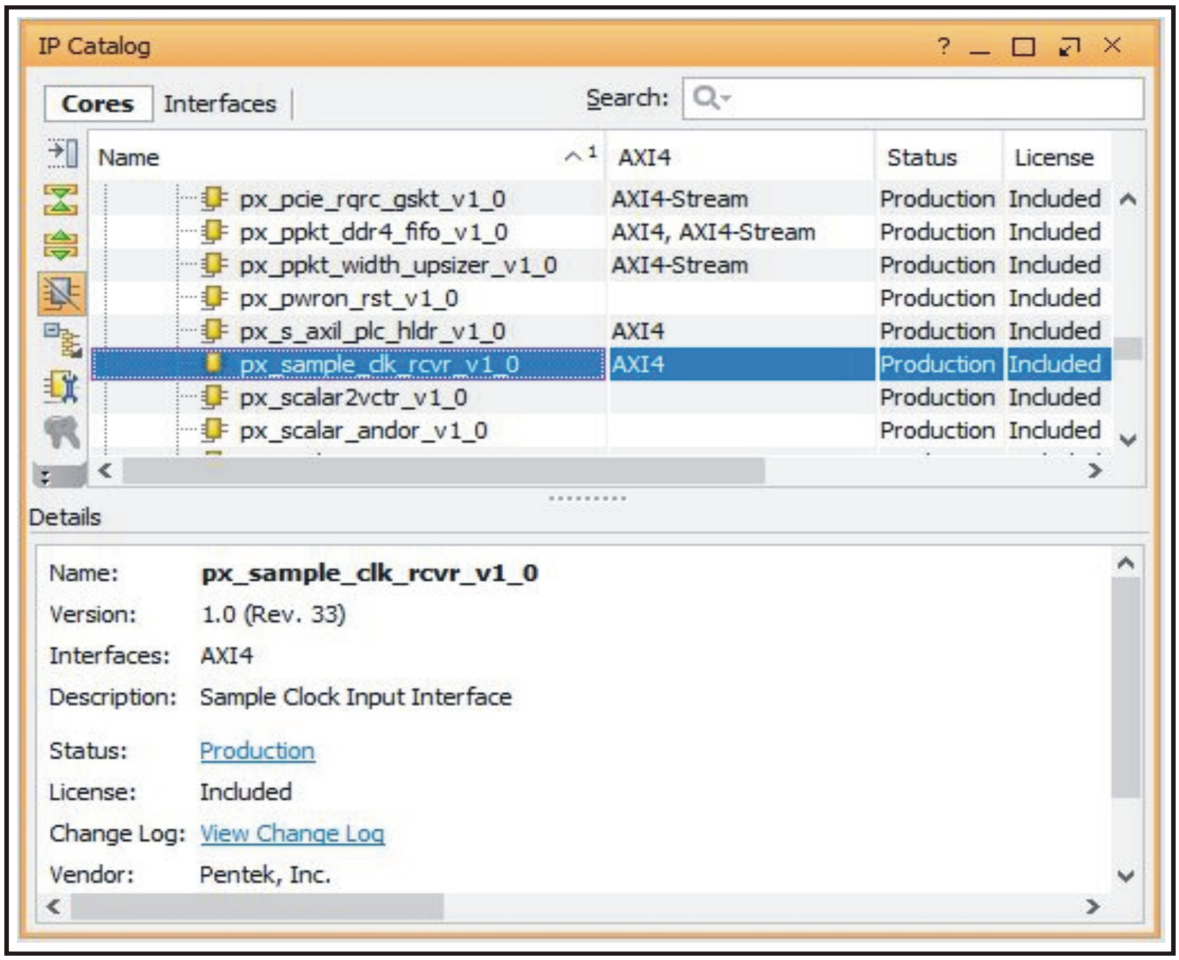
This section is not applicable to this IP core.

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek Sample Clock Receiver 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_sample_clk_rcvr_v1_0** as shown in Figure 6–1.

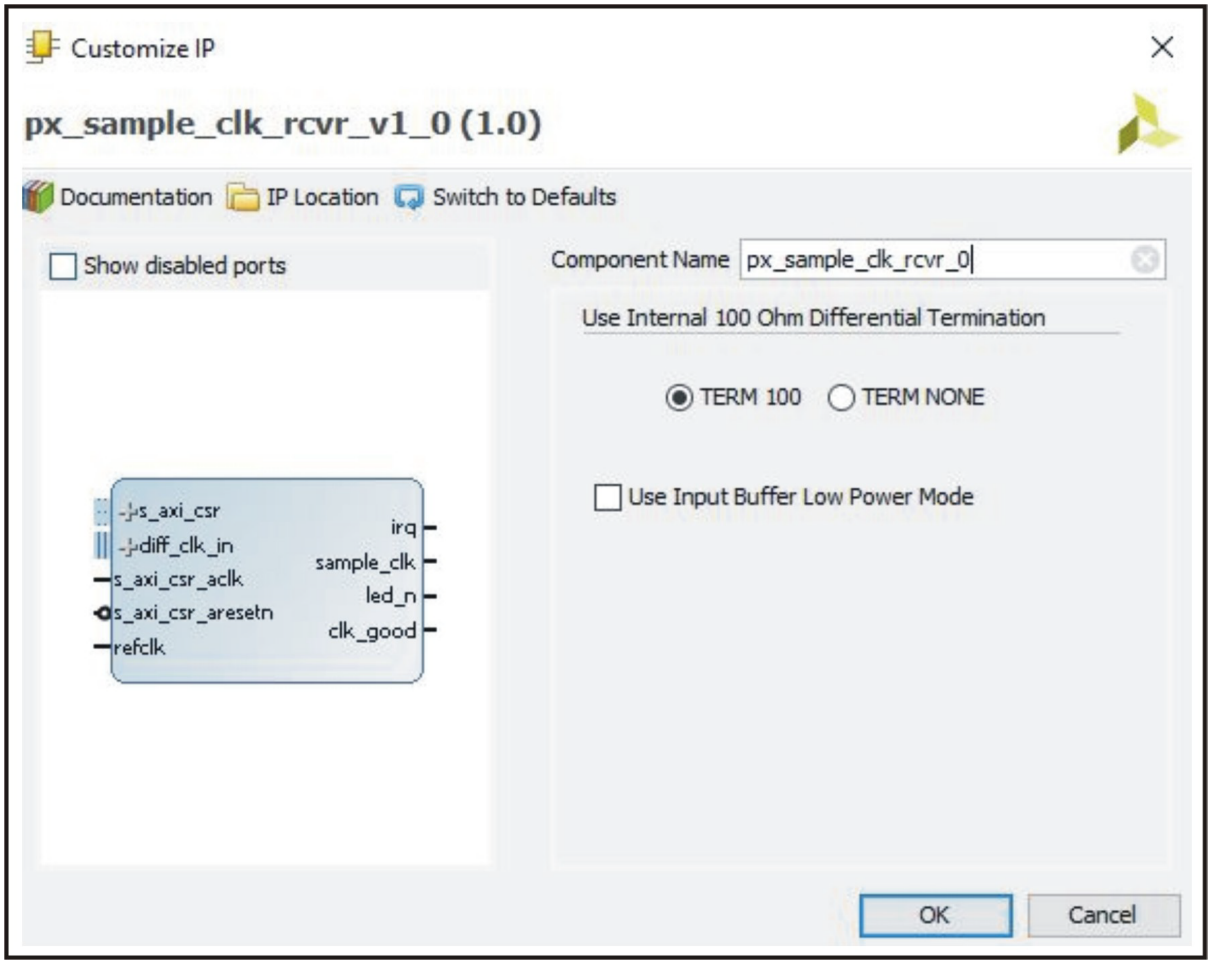
Figure 6–1: Sample Clock Receiver Core in Pentek IP Catalog



6.1 Pentek IP Catalog (continued)

When you select the **px_sample_clk_rcvr_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: Sample Clock Receiver Core IP Symbol



6.2 User Parameters

The user parameters of this IP core are described in Section 2.5 of this user manual.

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 for this core are not included in the Package IP. Clock constraints can be applied at the top level of the user design which includes this IP core.

Device, Package, and Speed Grade Selections

This IP works for the Kintex Ultrascale FPGAs.

Clock Frequencies

The clock frequency (**`s_axi_csr_aclk`**) for this IP core is 250 MHz.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking and Placement

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

6.5 Simulation

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*.

This page is intentionally blank