IP CORE MANUAL

AXI to CDCM7005 Controller IP

px_axil2cdc

PENTEK

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

Manual Part Number: 807.48305 Rev: 1.0 - December 09, 2016

Manual Revision History

Date	Version	Comments
12/09/16	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. 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

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™ AXI to CDCM7005 Controller Core provides initialization and control of the Texas Instruments™ CDCM7005 Clock Generator through its Serial Peripheral Interface (SPI).

This core complies with the ARM® AMBA® AXI4 Specification. This user manual defines the hardware interface, software interface, and parameterization options for the AXI to CDCM7005 Controller Core.

Features

- Provides serial input to the CDCM7005 clock generator SPI
- Register access through AXI4-Lite interface
- Software programmable clock divider for setting the desired clock frequency of the CDCM7005 clock generator
- Default words for the CDC SPI can be user defined and automatically loaded on reset

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[b]
Constraints File	Not Provided[c]
Simulation Model	VHDL
Supported S/W Driver	HAL Software Support
Tested Design Flows	
Design Entry	Vivado® Design Suite 2016.3 or later
Simulation	Vivado VSim
Synthesis	Vivado Synthesis
Support	
Provided by Pentek fpgasupport@pentek.com	

a.For a complete list of supported devices, see the *Vivado Design Suite Release Notes*.
b.Test bench will be available in the next revision of this core.
c.Clock constraints can be applied at the top-level module of the user design.

This page is intentionally blank

Chapter 1: Overview

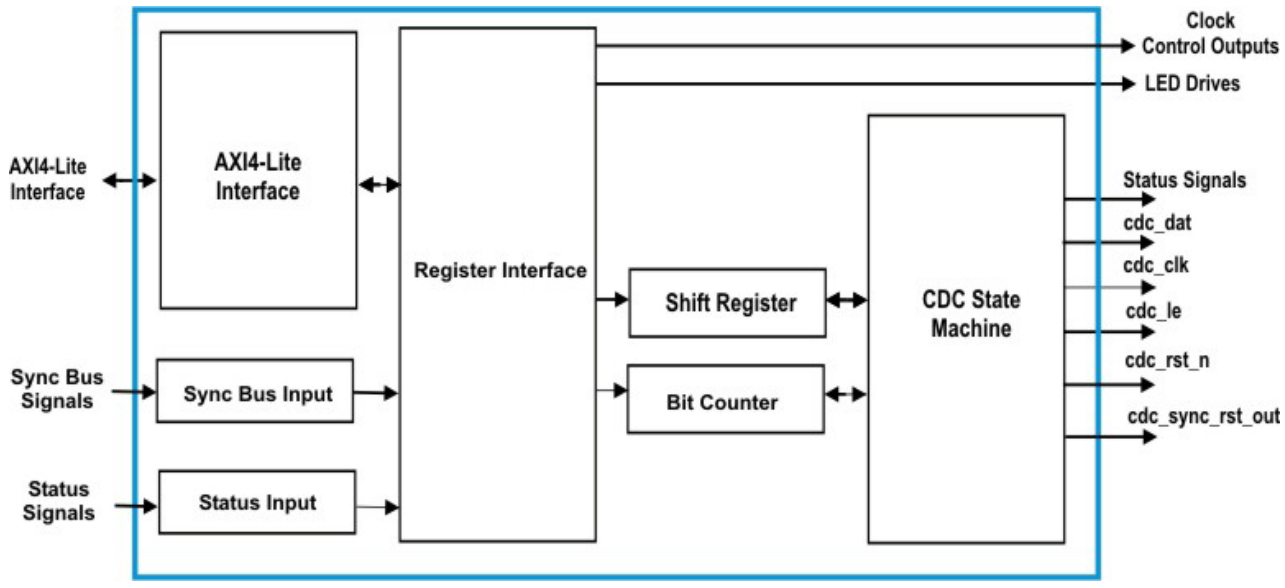
1.1 Functional Description

The AXI to CDCM7005 Controller Core provides a transaction interface to the AXI4-Lite Interface. The AXI4-Lite interface acts as a slave and is connected to the register space as shown in Figure 1-1. The core also has input status and sync bus signals.

The **sync bus signal input ports** can be enabled, when there are sync bus signals coming into the core, by setting the **has_sync_bus_sig** generic parameter to **True**. The Register Space is connected to the **CDC state machine** through a **shift register** and a **bit counter**. The state machine, along with the shift register and bit counter, provides the desired CDC interface SPI outputs which are used to initialize and control the **CDCM7005 Clock Generator**.

Figure 1-1 is a top-level block diagram of the AXI to CDCM7005 Controller Core. The modules within the block diagram are explained in the later sections of this manual.

Figure 1-1: AXI to CDCM7005 Controller 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.

1.1 Functional Description (continued)

- **Register Space:** This module contains control and status registers, including Interrupt Enable, Interrupt Status, and Interrupt Flag registers. Registers are accessed through the AXI4-Lite interface.
- **Shift Register and Bit Counter:** The shift register and bit counter work with the state machine to implement serial data transfer to the CDCM7005 Clock Generator across the CDC Interface of this core.
- **CDC State Machine:** This state machine is used to control the CDC Interface.

1.2 Applications

This core can be used for interfacing any Kintex Ultrascale or Virtex-7 FPGA to the CDCM7005 Clock Generator across an AXI4-Lite Interface. The CDCM7005 Clock Generator 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 provide supplemental material:

1) *Vivado Design Suite User Guide: Designing with IP*
2) *Vivado Design Suite User Guide: Programming and Debugging*
3) *ARM AMBA AXI4 Protocol Version 2.0 Specification*
   http://www.arm.com/products/system-ip/amba-specifications.php
4) *Texas Instruments CDCM7005 Clock Generator Datasheet*

Chapter 2: General Product Specifications

2.1 Standards

The AXI to CDCM7005 Controller Core has bus interfaces that comply with the A*RM AMBA AXI4-Lite Protocol Specification* and the *AMBA AXI4-Stream Protocol Specification*.

2.2 Performance

The performance of the AXI to CDCM7005 Controller 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 AXI to CDCM7005 Controller 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 AXI to CDCM7005 Controller 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	212
Flip-Flops	431

NOTE: Actual utilization may vary based on the user design in which the AXI to CDCM7005 Controller 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 AXI to CDCM7005 Controller 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
clock_division	Integer	**Clock Division:** This is the input clock divider value to generate desired clock frequency for the CDCM7005 clock generator SPI Interface. The output frequency will be **250 MHz/ clock_division** when AXI clock frequency is 250 MHz.
load_defaults	Boolean	**Load Defaults:** When set to True, the defaults words set by the user will be automatically loaded to the CDCM7005 clock generator at reset.
has_sync_bus_sig		**Has Sync Bus Signal:** Set to True when the user application has Sync Bus Signal inputs to the core.
default_word0	std_logic_vector	**Default word 0:** Default word 0 to be loaded to the CDCM7005 clock generator on reset. Defined by user[a].
default_word1		**Default word 1:** Default word 1 to be loaded to the CDCM7005 clock generator on reset. Defined by user[a].
default_word2		**Default word 2:** Default word 2 to be loaded to the CDCM7005 clock generator on reset. Defined by user[a].
default_word3		**Default word 3:** Default word 3 to be loaded to the CDCM7005 clock generator on reset. Defined by user[a].

a. The default values will depend on the Pentek Jade board this core is incorporated into and also the requirements of the user's application. Refer to the *Texas Instruments CDCM7005 Datasheet* for more details.

Chapter 3: Port Descriptions

This chapter provides details about the port descriptions for the following interface types:

- AXI4-Lite Core Interfaces
- I/O Signals

3.1 AXI4-Lite Core Interfaces

The AXI to CDCM7005 Controller 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 AXI to CDCM7005 Controller Core. Table 3-1 defines the ports in the CSR interface. See Chapter 4 for a Control/Status Register memory map and bit definitions. See the *AMBA AXI4-Lite Specification* for more details on operation of the AXI4-Lite interfaces.

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

Port	Direction	Width	Description
aclk	Input	1	**Clock**
aresetn	Input	1	**Reset:** Active low. This will reset the state machines within the core.
s_axi_csr_aresetn	Input	1	**Reset:** Active low. This value will reset all control registers to their initial states.
s_axi_csr_awaddr	Input	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 AXI to CDCM7005 Controller Core.
s_axi_csr_awprot	Input	3	**Protection:** The AXI to CDCM7005 Controller 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 AXI to CDCM7005 Controller 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 AXI to CDCM7005 Controller Core when it is ready to accept the write address. The address is latched when **s_axi_csr_awvalid** and **s_axi_csr_ awready** are high on the same cycle.
s_axi_csr_wdata	Input	32	**Write Data:** This data will be written to the address specified by **s_axi_csr_awaddr** when **s_axi_csr_wvalid** and **s_axi_csr_ wready** are both asserted. The value must be valid when **s_axi_csr_wvalid** is asserted and held until **s_axi_csr_wready** is also asserted.
s_axi_csr_wstrb	Input	4	**Write Strobes:** This signal when asserted indicates the number of bytes of valid data on **s_axi_csr_wdata** signal. Each of these bits, when asserted, indicate that the 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 AXI to CDCM7005 Controller 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 AXI to CDCM7005 Controller 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_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_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_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 AXI to CDCM7005 Controller Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the AXI to CDCM7005 Controller 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 the **s_axi_csr_araddr**. The AXI to CDCM7005 Controller Core asserts **s_axi_csr_arready** when it is ready to accept the Read Address. This input must remain asserted until the rising clock edge after the assertion **s_axi_ csr_arready**.
s_axi_csr_arready	Output	1	**Read Address Ready:** This output is asserted by the AXI to CDCM7005 Controller 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 AXI to CDCM7005 Controller 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 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.

3.2 I/O Signals

The I/O port/signal descriptions of the top-level module of the AXI to CDCM7005 Clock Generator Interface are described in Table 3-2.

Table 3-2: I/O Signals

Port/Signal Name	Type	Direction	Description
Status Input Signals			
cdc_stat_vcxo	std_logic	Input	**CDCM7005 VXCO Input Status:** Indicates the status of the VCXO Input to the CDCM7005 Clock Generator. 0 = VCXO Input is not valid 1 = VCXO Input is valid
cdc_stat_ref			**CDCM7005 Reference Clock Status:** Indicates the status of the Reference clock to the CDCM7005 Clock Generator. 0 = Reference Clock input is not valid 1 = Reference Clock input is valid
cdc_stat_lock			**CDCM7005 Phase Lock Status:** Indicates the status of the CDCM7005 Phase-Lock Loop. Meaningful only when **clk_mux_sel** modes are 00 and 01. 0 = CDCM7005 output not locked to the Reference clock 1 = CDCM7005 output locked to the Reference clock
CDCM7005 Interface			
cdc_dat	std_logic	Output	**CDCM7005 Data Input:** Data input for SPI of the CDCM7005 Clock Generator.
cdc_clk			**CDCM7005 Clock Input:** Clock input for SPI of the CDCM7005 Clock Generator.
cdc_le			**CDCM7005 Load Enable:** Load Enable Input for SPI of the CDCM7005 Clock Generator.
cdc_rst_n			**CDCM7005 Reset:** Active Low. Reset input for the CDCM7005 Clock Generator.
Clock Control Outputs			
vcxo_en	std_logic	Output	**VCXO Enable:** Active high. Enable signal for the VCXO output to the CDCM7005 clock generator.
vcxo_sel_n			**VCXO Select:** Active low. Select bit for the VCXO input of the CDCM7005. 0 - VCXO Clock input to CDCM7005 1 - External clock input to CDCM7005

Table 3-2: I/O Signals (Continued)

Port/Signal Name	Type	Direction	Description
clk_mux_sel	std_logic	Output	**CDCM7005 Clock mode Select**: These bits are copies of the mode selection control bits that go out to the CDCM7005 clock generator. They are used here to help define what a “valid” clock detect is in different clocking modes. **00 => VCXO with External Reference Clock** **01 => VCXO with Sync Bus Reference Clock** **10 => VCXO with no Reference Clock** **11 =>VCXO with no Reference Clock**
Status Outputs			
stat_vcxo	std_logic	Output	**VCXO Input Status:** Copy of **cdc_stat_vcxo** input signal, synchronized to **aclk**, for use by other modules in the user application.
stat_ref			**Reference Clock Status:** Copy of **cdc_stat_ref** input signal, synchronized to **aclk**, for use by other modules in the user application.
stat_lock			**Phase Lock Status:** Copy of **cdc_stat_lock** input signal, synchronized to **aclk**, for use by other modules in the user application.
Sync Bus Signals			
cdc_sync_rst_in_p	std_logic	Input	**CDCM7005 Sync Reset Differential Input:** This signal from the SYNC bus connector can be used to synchronize the CDCM7005 clock generators on multiple boards and align them in phase to a common reference clock.
cdc_sync_rst_in_n			
cdc_sync_rst_out		Output	**CDCM7005 Sync Reset Output:** CDC Reset output to the Sync Bus.
LED Drives			
cdc_stat_vcxo_led_n	std_logic	Output	**CDCM7005 VCXO Status LED:** Active low. Indicates the status of the VCXO input signal.
cdc_stat_ref_led_n			**CDCM7005 Reference Status LED:** Active low. Indicates the status of the Reference clock input signal.
cdc_stat_lock_led_n			**CDCM7005 Lock Status LED:** Active low. Indicates the Phase Lock Status of the CDCM7005 Clock generator.

This page is intentionally blank

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the register space of the AXI to CDCM7005 Controller Core. The memory map is provided in Table 4-1.

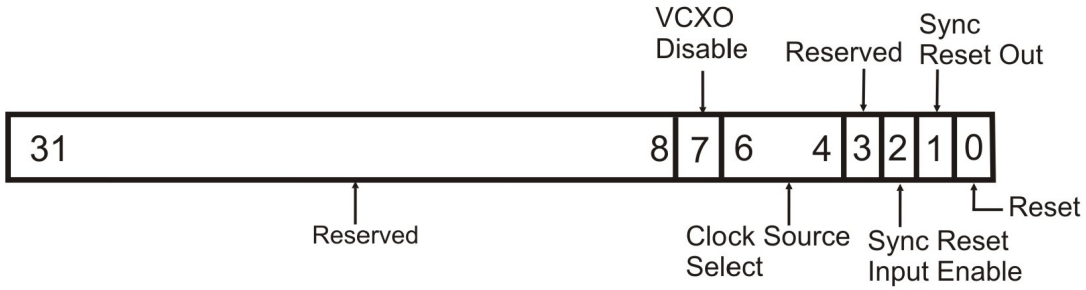
Table 4-1: Register Space Memory Map

Register Name	Address (Base Address +)	Access	Description
Control Register	0x00	R/W	Controls the CDC reset, CDC multiplexer, and the VCXO output disable.
Load Value Register	0x04		Stores the values to be transferred through the CDC interface.
Write Request Register	0x08		Write Request Control Register
Status Register	0x0C		Indicates the status of VCXO, Reference input, lock, default load, and state machine.
Reserved	0x10	N/A	Reserved
Interrupt Enable Register	0x14	R/W	Interrupt enable bits
Interrupt Status Register	0x18	R	Interrupt source status bits
Interrupt Flag Register	0x1C	R/Clr	Interrupt flag bits

4.1 Control Register

This register is used to control CDC Reset, CDC Reset output to the Sync Bus, CDC Reset input from the Sync Bus, Clock multiplexer control, and VCXO Output Enable. 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
31:8	Reserved	N/A	N/A	**Reserved**
7	vcxo_disable	0	R/W	**VCXO Disable:** Disable signal for the VCXO output to the CDCM7005 clock generator. 0 = Enable 1 = Disable
6:4	clk_src_sel	000	R/W	**Clock Source Select:** These bits define the different clock selection modes of the CDCM7005. They indicate the VCXO input and the reference clock selection input of the CDCM7005 clock generator. 000 - VCXO with External Reference clock 001 - VCXO with Sync Bus Reference clock 010 - VCXO with no reference 011 - VCXO with no reference 100 - External clock 101 - Sync Bus Clock 110 , 111 - undefined
3	Reserved	N/A	N/A	**Reserved**
2	cdc_sync_rst_in_en	0	R/W	**CDCM7005 Sync Reset Input Enable:** This bit when set to '1' will enable the **cdc_sync_rst_in** signal to be connected to the CDCM7005 clock generator. 0 = Disable 1 = Enable

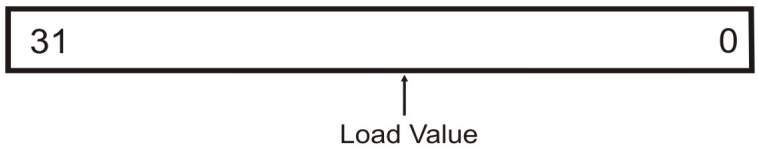
Table 4-2: Control Register (Base Address + 0x00) (Continued)

Bits	Field Name	Default Value	Access Type	Description
1	cdc_sync_rst_out	0	R/W	**CDCM7005 Sync Reset Out:** This is the sync reset output signal to the Sync bus. 0 = Run 1 = Reset
0	reg_rst	0	R/W	**CDCM7005 Reset:** 0 = Run 1 = Reset

4.2 Load Value Register

This register controls the data output on the CDC Interface. This register is illustrated in Figure 4-2 and described in Table 4-3.

Figure 4-2: Load Value Register



.

Table 4-3: Load Value Register (Base Address + 0x04)

Bits	Field Name	Default Value	Access Type	Description
31:0	cntl_word	0x00000000	R/W	**Load Value:** This register contains the data to be transferred over the CDC interface. Note that the register address where the data is written in the CDCM7005 is included in the data value (load value). For more details please refer to the *Texas Instruments CDCM7005 Clock Generator Datasheet*.

4.3 Write Request Register

The Write Request register controls write requests to write data over the CDC interface. It has a single defined bit in the least significant bit position, which is used to control the write request to enable or disable a write operation over the CDC Interface. This register is illustrated in Figure 4-2 and described in Table 4-3.

Figure 4-3: Write Request Register

.

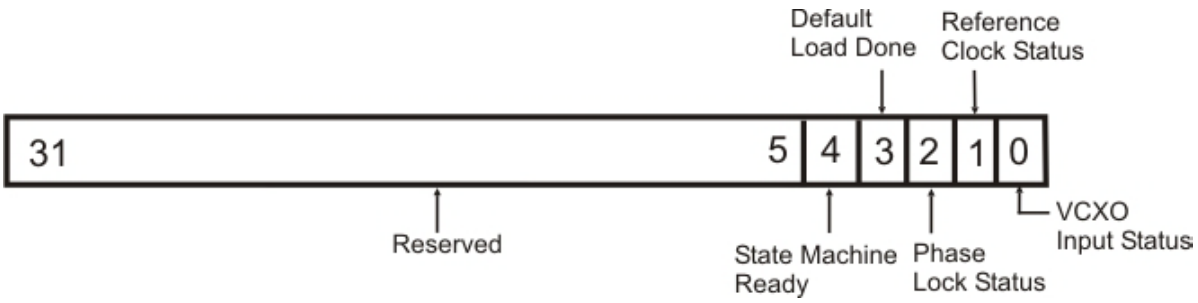
Table 4-4: Write Request Register (Base Address + 0x08)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	wr_req	0	R/W	**Write Request:** Toggle '1' then '0' to transfer the value in the load value register through the CDCM7005 SPI interface.

4.4 Status Register

This register contains the status of the AXI to CDCM7005 Controller Core. The status register is illustrated in Figure 4-2 and described in Table 4-3.

Figure 4-4: Status Register



.

Table 4-5: Status Register (Base Address + 0x0C)

Bits	Field Name	Default Value	Access Type	Description
31:5	Reserved	N/A	N/A	**Reserved**
4	sm_ready	0	R	**State Machine Ready:** Indicates if the state machine is ready for a new transfer over the CDC interface. 0 = State machine not ready 1 = State machine ready for new transfer **Note:** Do not initiate a transfer until the state machine signals ready through this bit.
3	default_load_ done	0	R	**Default Load Done:** This bit indicates the completion of write operation on reset. At reset, if the generic parameter **load_defaults** is set to True, then defaults words are written to the CDCM7005 clock generator. 0 = Defaults words being written 1 = Default words write complete **Note:** Do not initiate a transfer until this status bit is '1'.
2	cdc_stat_lock	-	R	**CDCM7005 Phase Lock Status:** Indicates the status of the phase-lock loop of the CDCM7005 clock generator. Meaningful only when **clk_src_sel** modes of the Control Register are 000 and 001. 0 = CDCM7005 output not locked to the Reference clock 1 = CDCM7005 output locked to the Reference clock

Table 4-5: Status Register (Base Address + 0x0C) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
1	cdc_stat_ref	-	R	**CDCM7005 Reference Clock Status:** Indicates the status of the Reference clock input to the CDCM7005 clock generator. 0 = Reference clock input not valid 1 = Reference clock input valid
0	cdc_stat_vcxo	-	R	**CDCM7005 VXCO Input Status:** Indicates the status of the VCXO input to the CDCM7005 clock generator. 0 = VCXO input not valid 1 = VCXO input valid

4.5 Interrupt Enable Register

The bits in the interrupt enable register are used to enable (or disable) the generation of interrupts based on the condition of certain circuit elements, known as interrupt sources. When a bit in this register associated with a given interrupt source is High, an interrupt will be generated by the rising edge of that source's Interrupt Status Register bit (See Section 4.6). This register is illustrated in Figure 4-5 and described in Table 4-6.

Figure 4-5: Interrupt Enable Register

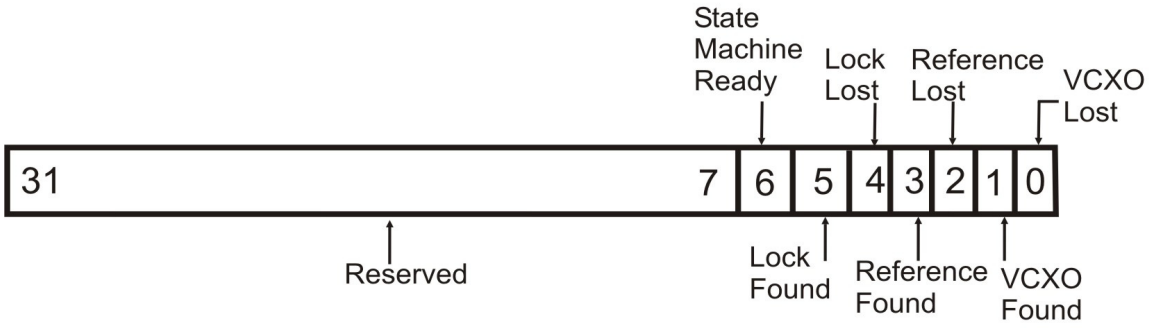


Table 4-6: Interrupt Enable Register (Base Address + 0x14)

Bits	Field Name	Default Value	Access Type	Description
31:7	Reserved	N/A	N/A	**Reserved**
6	sm_ready	0	R/W	**State Machine Ready:** This bit enables/ disables the state machine ready interrupt source. 0 = Disable interrupt 1 = Enable interrupt
5	lock_found	0	R/W	**CDCM7005 Lock Found:** This bit enables/ disables the CDCM7005 lock found interrupt source. The CDCM7005 lock found interrupt source indicates that the CDCM7005 output is locked to the reference clock. 0 = Disable interrupt 1 = Enable interrupt

Table 4-6: Interrupt Enable Register (Base Address + 0x14) (Continued)

Bits	Field Name	Default Value	Access Type	Description
4	lock_lost	0	R/W	**CDCM7005 Lock Lost:** This bit enables/ disables the CDCM7005 lock lost interrupt source. The CDCM7005 lock lost interrupt source indicates that the CDCM7005 output lock to the reference clock is lost. 0 = Disable interrupt 1 = Enable interrupt
3	ref_found			**CDCM7005 Reference Found:** This bit enables/ disables the CDCM7005 reference clock input found interrupt source. 0 = Disable interrupt 1 = Enable interrupt
2	ref_lost			**CDCM7005 Reference Lost:** This bit enables/ disables the CDCM7005 reference clock input lost interrupt source. 0 = Disable interrupt 1 = Enable Interrupt
1	vcxo_found			**CDCM7005 VCXO Found:** This bit enables/ disables the CDCM7005 VCXO input found interrupt source. 0 = Disable interrupt 1 = Enable interrupt
0	vcxo_lost			**CDCM7005 VCXO Lost:** This bit enables/ disables the CDCM7005 VCXO input lost interrupt source. 0 = Disable interrupt 1 = Enable interrupt

4.6 Interrupt Status Register

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

Some of the interrupt sources are transient and so may not appear in the Interrupt Status Register at the time it is read. In such cases use the Interrupt Flag Register to see the interrupt conditions that have occurred. The Interrupt Status Register is illustrated in Figure 4-6 and described in Table 4-7.

Figure 4-6: Interrupt Status Register

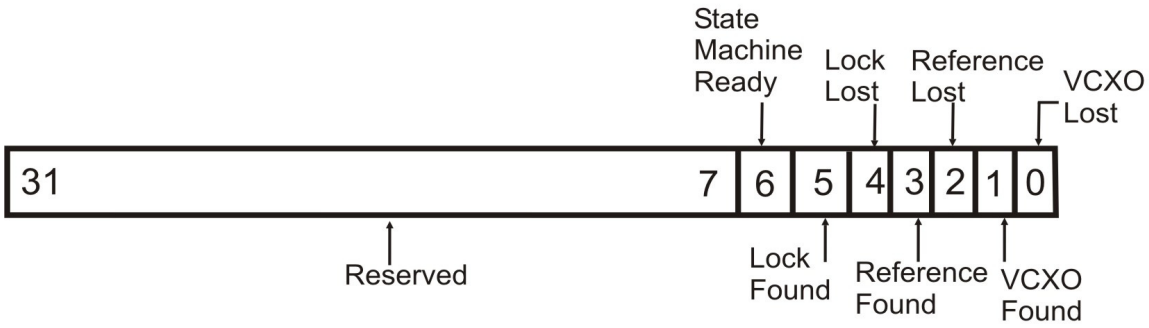


Table 4-7: Interrupt Status Register (Base Address + 0x18)

Bits	Field Name	Default Value	Access Type	Description
31:7	Reserved	N/A	N/A	**Reserved**
6	sm_ready	0	R	**State Machine Ready:** This bit indicates the status of the state machine ready interrupt source. 0 = No interrupt 1 = Interrupt condition asserted
5	lock_found	0	R	**CDCM7005 Lock Found:** This bit indicates the status of the CDCM7005 lock found interrupt source. The CDCM7005 lock found interrupt source indicates that the CDCM7005 output is locked to the reference clock. 0 = No interrupt 1 = Interrupt condition asserted

Table 4-7: Interrupt Status Register (Base Address + 0x18) (Continued)

Bits	Field Name	Default Value	Access Type	Description
4	lock_lost	0	R	**CDCM7005 Lock Lost:** This bit indicates the status of the CDCM7005 lock lost interrupt source. The CDCM7005 lock lost interrupt source indicates that the CDCM7005 output lock to the reference clock is lost. 0 = No interrupt 1 = Interrupt condition asserted
3	ref_found			**CDCM7005 Reference Found:** This bit indicates the status of the CDCM7005 reference clock input found interrupt source. 0 = No interrupt 1 = Interrupt condition asserted
2	ref_lost			**CDCM7005 Reference Lost:** This bit indicates the status of the CDCM7005 reference clock input lost interrupt source. 0 = No interrupt 1 = Interrupt condition asserted
1	vcxo_found			**CDCM7005 VCXO Found:** This bit indicates the status of the CDCM7005 VCXO input found interrupt source. 0 = No interrupt 1 = Interrupt condition asserted
0	vcxo_lost			**CDCM7005 VCXO Lost:** This bit indicates the status of the CDCM7005 VCXO input lost interrupt source. 0 = No interrupt 1 = Interrupt condition asserted

4.7 Interrupt Flag Register

The Interrupt Flag Register has read/clear access associated with each interrupt condition. When reset, this register has all bits set to '0' (cleared). Each flag bit in this register latches an interrupt occurrence. A '1' in any flag bit in this register indicates that an interrupt has occurred.

Note that when any status bit in the Interrupt Status Register, changes to '1' the corresponding flag bit in this register will also be set to '1'. However, when a status bit in the Interrupt Status Register clears from '1' to '0', the corresponding latched flag bit in this register does not clear, but remains at '1'. To clear the flag bits, write '1's to the desired bits. The flags are not affected by the enable register. The Interrupt Flag Register is illustrated in Figure 4-7 and described in Table 4-8.

Figure 4-7: Interrupt Flag Register

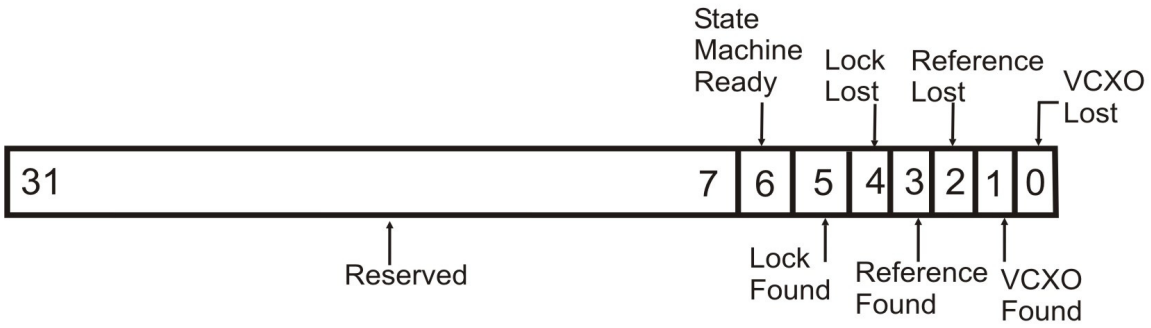


Table 4-8: Interrupt Flag Register (Base Address + 0x1C)

Bits	Field Name	Default Value	Access Type	Description
31:7	Reserved	N/A	N/A	**Reserved**
6	sm_ready	0	R/Clr	**State Machine Ready:** This bit indicates the state machine ready interrupt flag. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
5	lock_found	0	R/Clr	**CDCM7005 Lock Found:** This bit indicates the CDCM7005 lock found interrupt flag. The CDCM7005 lock found interrupt source indicates that the CDCM7005 output is locked to the reference clock. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch

Table 4-8: Interrupt Flag Register (Base Address + 0x1C) (Continued)

Bits	Field Name	Default Value	Access Type	Description
4	lock_lost	0	R/Clr	**CDCM7005 Lock Lost:** This bit indicates the CDCM7005 lock lost interrupt flag. The CDCM7005 lock lost interrupt source indicates that the CDCM7005 output lock to the reference clock is lost. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
3	ref_found			**CDCM7005 Reference Found:** This bit indicates the CDCM7005 reference clock input found interrupt flag. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
2	ref_lost			**CDCM7005 Reference Lost:** This bit indicates the CDCM7005 reference clock input lost interrupt flag. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
1	vcxo_found			**CDCM7005 VCXO Found:** This bit indicates the CDCM7005 VCXO input found interrupt flag. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
0	vcxo_lost			**CDCM7005 VCXO Lost:** This bit indicates the CDCM7005 VCXO input lost interrupt flag. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch

This page is intentionally blank

Chapter 5: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the AXI to CDCM7005 Controller Core.

5.1 General Design Guidelines

The AXI to CDCM7005 Controller Core is used as an interface to the Texas Instruments CDCM7005 clock generator and provides serial input to the SPI of the clock generator. It also supports loading default words automatically into the CDCM7005 clock generator on reset.

5.2 Clocking

Main Clock: **`aclk`**

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

5.3 Resets

Main reset: **`aresetn`**

This is an active low synchronous reset associated with the **`aclk`**. When asserted, this will reset the CDC state machine.

Control/Status Reset: **`s_axi_csr_aresetn`**

This is active low synchronous reset associated with the **`aclk`**. When asserted, this will reset all the control registers back to their initial default states. It does not reset any of the state machines within the core.

5.4 Interrupts

This core has an edge-type (rising edge-triggered) interrupt output. It is synchronous with the **aclk**. On the rising edge of any interrupt signal, a one clock cycle wide pulse is output from the core on its **irq** output. Each interrupt event is stored in two registers, accessible on the **s_axi_csr** bus. The Interrupt Status Register always reflects the current state of the interrupt condition, which may have changed since the generation of the interrupt.

5.4 Interrupts (continued)

The Interrupt Flag Register latches the occurrence of each interrupt, in a bit that retains its state until explicitly cleared.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 **aclk**. It is a standard AXI4-Lite Interface. See Chapter 4 for the control register memory map, which provides more details on the registers that can be accessed through this interface.

5.6 Programming Sequence

This section briefly describes the programming sequence of registers to initiate and complete a transaction on the AXI to CDCM7005 Controller Core.

1) Ensure that the Interrupt Flag Register is cleared.
2) Check for status of the ‘State Machine Ready’ bit in the status register. This indicates whether the state machine is busy in another transaction or free for a new transaction.
3) Set the control register.
4) Toggle the write request.
5) Wait until an **sm_ready** interrupt is received, check the interrupt flag register, and clear the interrupts.

5.7 Timing Diagrams

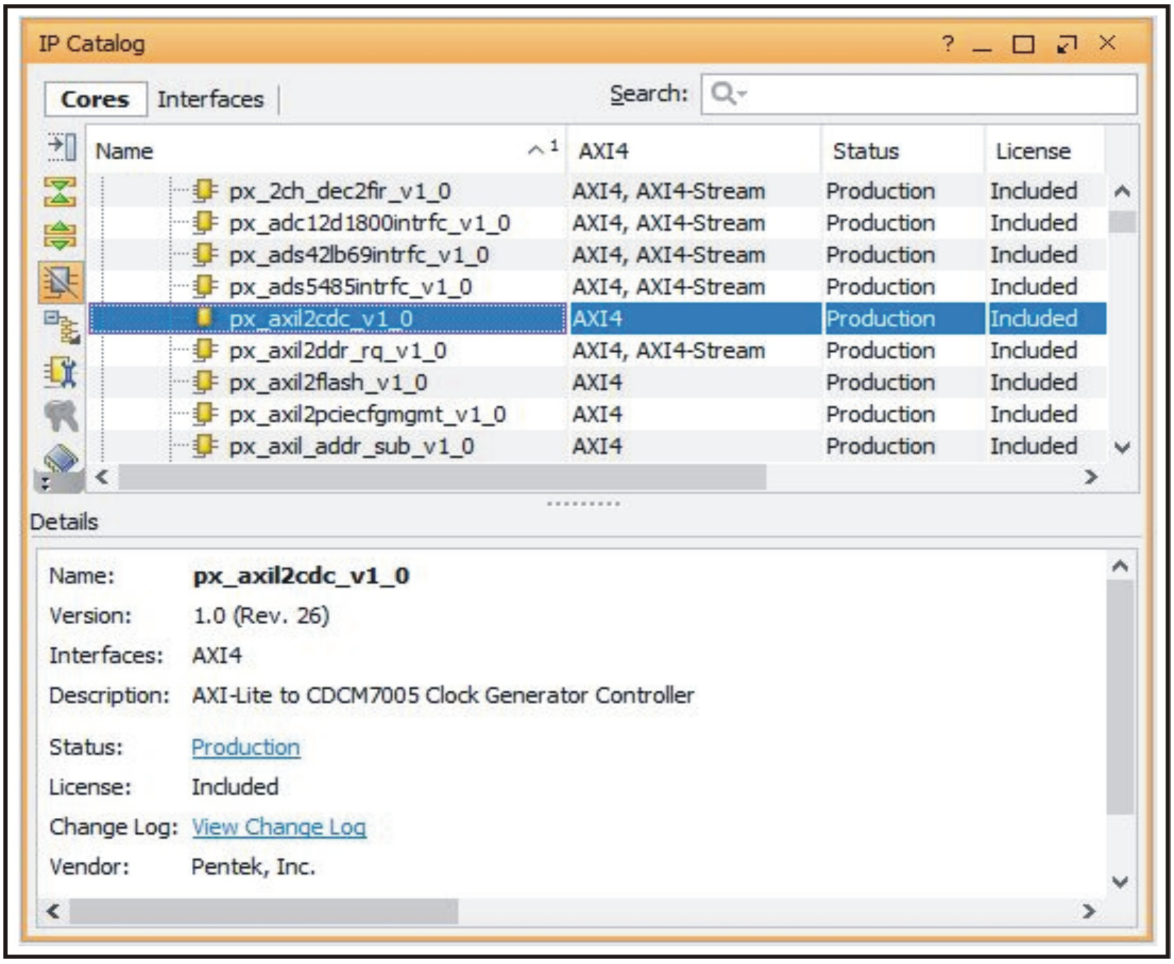
This section is not applicable to this IP core.

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek AXI to CDCM7005 Controller 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_axil2cdc_v1_0** as shown in Figure 6-1.

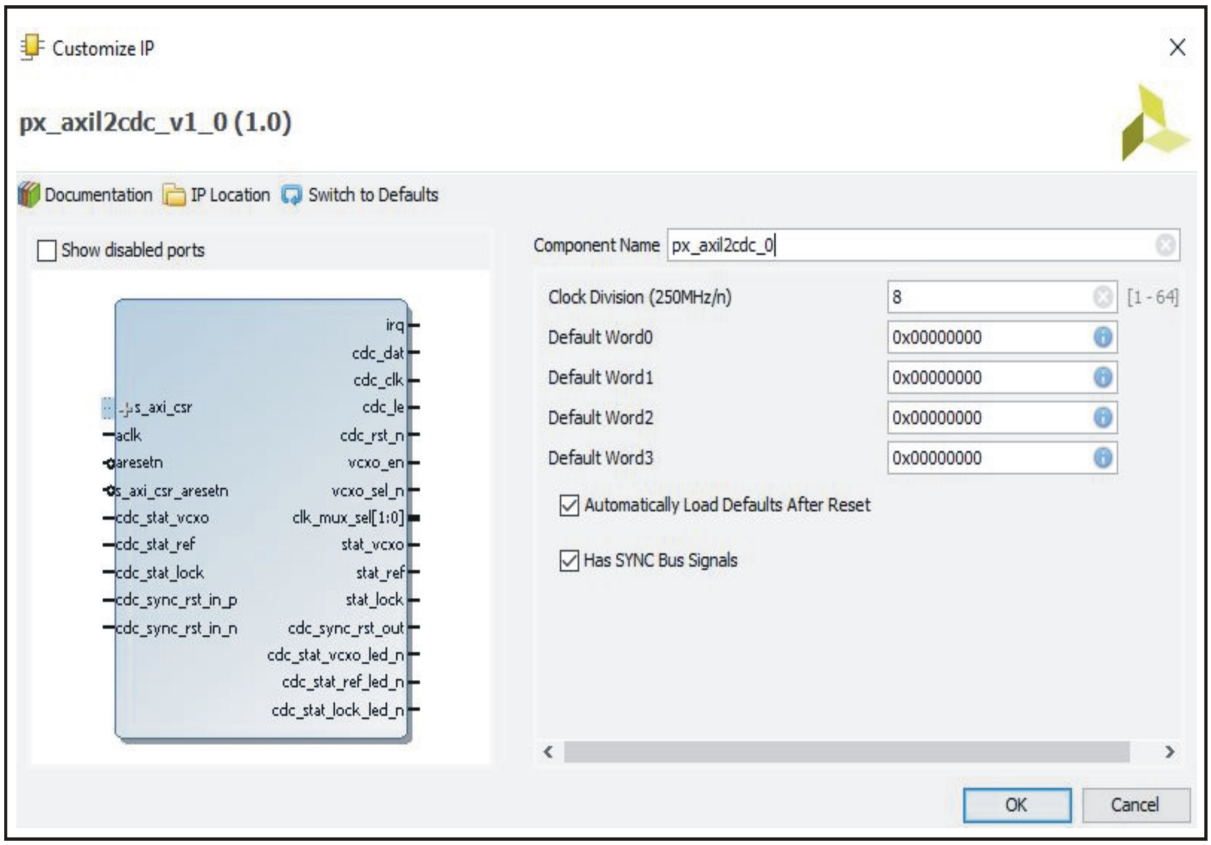
Figure 6-1: AXI to CDCM7005 Controller Core in Pentek IP Catalog



6.1 Pentek IP Catalog (continued)

When you select the **px_axil2cdc_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: AXI to CDCM7005 Controller Core IP Symbol



6.2 User Parameters

- **Clock Division:** The input clock frequency is divided by the clock division value, provided by the user, to obtain the desired clock frequency for the SPI of CDCM7005 clock generator. This value ranges from 1 - 64.
- **Default Word (4:0):** These are the default words transferred through the CDC Interface to the CDCM7005 clock generator when the load defaults parameter is set to True.
- **Load Defaults:** Setting this parameter to True will allow loading the default words into the CDCM7005 clock generator automatically at reset.
- **Has Sync Bus Signals:** This parameter can be set to True if the user design has Sync bus signals inputs to the 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 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 and Virtex-7 FPGAs.

Clock Frequencies

The clock frequency (**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

The test bench and the simulation results for this IP core will be available in the next revision of this user manual.

6.6 Synthesis and Implementation

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