IP CORE MANUAL

AXI-Lite to 16-Bit Parallel NOR Flash Interface IP

px_axil2flash

PENTEK

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

Manual Part Number: 807.48307 Rev: 1.0 - December 09, 2016

Manual Revision History

Date	Version	Comments
12/09/16	1.0	Initial Release

Legal Notices

Copyright

Trademarks

Printed in the United States of America.

Table of Contents

Page

Table of Contents

IP Facts

Description

Pentek's Navigator™ AXI4-Lite to 16-bit Parallel NOR Flash Interface Core provides a read/ write interface to the Micron® PC28F00AG18 Parallel NOR Flash memory.

This core complies with the ARM® AMBA® AXI4 Specification. This user manual defines the hardware interface, software interface, and parameterization options for the AXI4-Lite to 16-bit Parallel NOR Flash Interface Core.

Features

- Register access through AXI4-Lite interface
- Software programmable width of the Flash address

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 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.Clock constraints can be applied at the top-level module of the user design.

This page is intentionally blank

Chapter 1: Overview

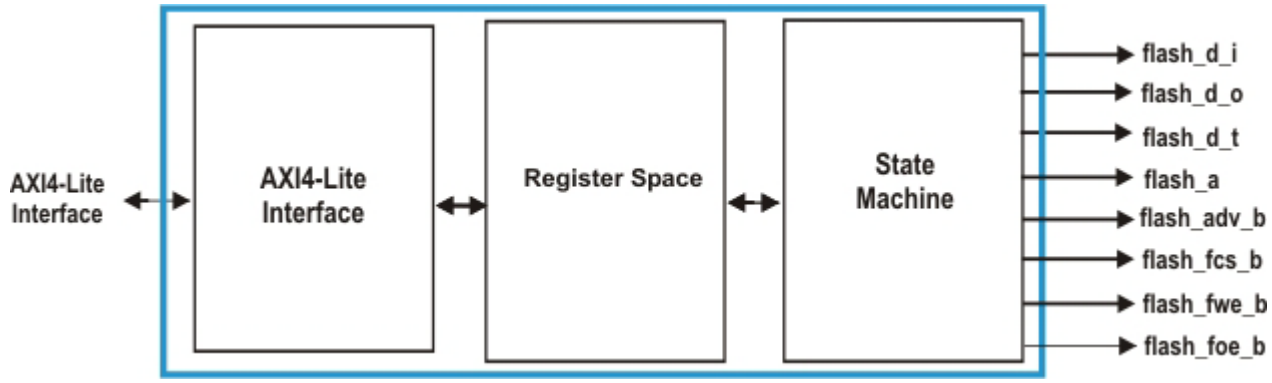
1.1 Functional Description

The AXI4-Lite to NOR Flash Interface Core provides a transaction interface to the Parallel NOR Flash through an AXI4-Lite Interface in the user design. This core accepts Flash read/ write requests across the AXI4-Lite interface and converts them into signals compatible with the Flash memory. The core performs asynchronous read/ write operations in the Flash Memory. The width of the Flash address bus can be defined by the user using the generic parameter **flash_addr_bits** (see Table 2-2).

The **AXI4-Lite Interface** acts as a slave and is connected to the **Register Space** as shown in Figure 1-1. The **Register Space** is connected to a **State Machine** that generates the desired outputs to the **Flash**. This core is expected to be externally connected to a tri-state buffer to implement open collector drivers for the flash data I/O signal.

Figure 1-1 is a top-level block diagram of the AXI4-Lite to NOR Flash Interface Core. The modules within the block diagram are explained in other sections of this manual.

Figure 1-1: AXI4-Lite to 16-bit Parallel NOR Flash Interface Core Block



- **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 control and status registers within the core, which are accessed through the AXI4-Lite interface.
- **State Machine:** This state machine is used to control and generate the necessary outputs to the Flash memory.

1.2 Applications

This core can be used for interfacing any Kintex Ultrascale FPGA to the 16-bit Parallel NOR Flash Memory across an AXI4-Lite Interface.

1.3 System Requirements

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

1.4 Licensing and Ordering Information

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

1.5 Contacting Technical Support

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

1.6 Documentation

This user manual is the main document for this IP core. The following documents 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) Micron 16-Bit Parallel NOR Flash Datasheet

Chapter 2: General Product Specifications

2.1 Standards

The AXI4-Lite to NOR Flash Interface Core has a bus interface that complies with the *ARM AMBA AXI4-Lite Protocol Specification*.

2.2 Performance

The performance of the AXI4-Lite to NOR Flash Interface 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 AXI4-Lite to NOR Flash Interface 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 AXI4-Lite to NOR Flash Interface 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	111
Flip-Flops	192

NOTE: Actual utilization may vary based on the user design in which the AXI4-Lite to NOR Flash Interface 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 AXI4-Lite to NOR Flash Interface 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
flash_addr_bits	Integer	**Flash Address Bits:** This generic parameter defines the width of the address bus of the Flash memory. It can range from 16 to 32 bits.

Chapter 3: Port Descriptions

This chapter provides port descriptions for the following interface types:

- AXI4-Lite Core Interfaces
- I/O Signals

3.1 AXI4-Lite Core Interfaces

The AXI4-Lite to NOR Flash Interface 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 registers in the AXI4-Lite to NOR Flash Interface 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
s_axi_aclk	Input	1	**Clock**
s_axi_aresetn	Input	1	**Reset:** Active low. This will reset the state machines within the core and reset the control registers to their initial states.
s_axi_awaddr	Input	4	**Write Address:** Address used for write operations. It must be valid when **s_axi_awvalid** is asserted and must be held until **s_axi_awready** is asserted by the Flash Interface core.
s_axi_awprot	Input	3	**Protection:** The Flash Interface 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_awaddr**. The Flash Interface core asserts **s_axi_awready** when it is ready to accept the address. The **s_axi_awvalid** must remain asserted until the rising clock edge after the assertion of **s_axi_awready**.
s_axi_awready	Output	1	**Write Address Ready:** This output is asserted by the Flash Interface core when it is ready to accept the write address.The address is latched when **s_axi_awvalid** and **s_axi_ 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_awaddr** when **s_axi_wvalid** and **s_axi_ wready** are both asserted. The value must be valid when **s_axi_wvalid** is asserted and held until **s_axi_wready** is also asserted.
s_axi_wstrb	Input	4	**Write Strobes:** This signal when asserted indicates the number of bytes of valid data on **s_axi_wdata** signal. Each of these bits, when asserted, indicate that the corresponding byte of **s_axi_wdata** contains valid data. Bit 0 corresponds to the least significant byte, and bit 3 to the most significant.
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_ wdata** is written into the register at address **s_axi_awaddr** when **s_axi_wready** and **s_axi_wvalid** are High on the same cycle.
s_axi_wready	Output	1	**Write Ready:** This signal is asserted by the Flash Interface core when it is ready to accept data. The value on **s_axi_wdata** is written into the register at address **s_axi_awaddr** when **s_axi_wready** and **s_axi_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 Flash Interface core indicates success or failure of a write transaction through this signal, which is valid when **s_axi_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_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	4	**Read Address:** Address used for read operations. It must be valid when **s_axi_arvalid** is asserted and must be held until **s_axi_arready** is asserted by the Flash Interface core.
s_axi_arprot	Input	3	**Protection:** These bits are ignored by the Flash Interface core
s_axi_arvalid	Input	1	**Read Address Valid:** This input must be asserted to indicate that a valid read address is available on the **s_axi_araddr**. The Flash Interface core asserts **s_axi_arready** when it 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_arready	Output	1	**Read Address Ready:** This output is asserted by the Flash Interface core when it is ready to accept the read address. The address is latched when **s_axi_arvalid** and **s_axi_ arready** are high on the same cycle.

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

Port	Direction	Width	Description
s_axi_rdata	Output	32	**Read Data:** This value is the data read from the address specified by the **s_axi_araddr** when **s_axi_arvalid** and **s_axi_arready** are high on the same cycle.
s_axi_rresp	Output	2	**Read Response:** The Flash Interface core indicates success or failure of a read transaction through this signal, which is valid when **s_axi_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_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 _rdata**. It is held until **s_axi_rready** is asserted by the user logic.
s_axi_rready	Input	1	**Read Data Ready:** This signal is asserted by the user logic when it is ready to accept the Read Data.

3.2 I/O Signals

The I/O port/signal descriptions of the top-level module of the AXI4-Lite to NOR Flash Interface Core are described in Table 3-2.

Table 3-2: I/O Signals

Port/Signal Name	Type	Direction	Description
flash_d_i [15:0]	std_logic_vector	Input	**Flash Input Data:** This is the incoming data from the Flash Memory through a tri-state buffer.
flash_d_o[15:0]		Output	**Flash Output Data:** This is the outgoing data from the Flash Interface Core to the tri-state buffer.
flash_d_t [15:0]			**Flash Output Data Enable:** This is the flash output data enable signal to the tri-state buffer.
flash_a [flash_addr_bits-1 : 0]			**Flash Address Output:** This is the flash address output where the read/ write operations are to be performed. The width of this bus is defined by the generic parameter **flash_addr_bits.**
flash_adv_b	std_logic		**Flash Address Valid:** Active Low. This bit is used to enable the address output to the Flash memory.
flash_fwe_b			**Flash Write Enable:** Active Low. When set to Low, this bit enables a write operation in the Flash memory.
flash_fcs_b			**Flash Chip Enable:** Active Low. When Low, this bit selects the die. When High, it deselects the die and places it in standby.
flash_foe_b			**Flash Output Enable:** Active Low. This bit is set to Low for Reads, and High for Writes.

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the register space of the AXI4-Lite to NOR Flash Interface Core. The memory map is provided in Table 4-1.

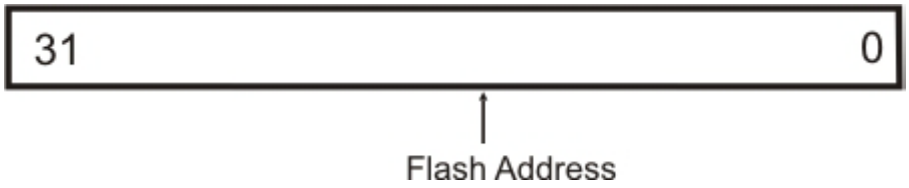
Table 4-1: Register Space Memory Map

Register Name	Address (Base Address +)	Access	Description
Flash Address	0x00	R/W	Controls the flash address
Flash Data	0x04		Control the Flash data

4.1 Flash Address Register

This register is used to control the Flash address where a read/ write operation is to be performed. This register is illustrated in Figure 4-1 and described in Table 4-2.

Figure 4-1: Flash Address Register



.

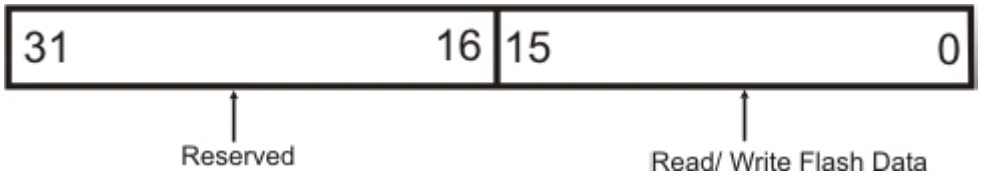
Table 4-2: Flash Address Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:0	flash_addr	0x00000000	R/W	**Flash Address**

4.2 Flash Data Register

When accessed during a write operation, this register holds the data to be written to the Flash memory, and during a read operation holds data read from the Flash memory. This register is illustrated in Figure 4-2 and described in Table 4-3.

Figure 4-2: Flash Data Register



.

Table 4-3: Flash Data Register (Base Address + 0x04)

Bits	Field Name	Default Value	Access Type	Description
31:0	Reseved	N/A	N/A	**Reserved**
15:0	data	0x00000000	R/W	**Read/ Write Flash Data**

Chapter 5: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the AXI4-Lite to NOR Flash Interface Core.

5.1 General Design Guidelines

The AXI4-Lite to NOR Flash Interface Core is used as an interface to the 16-bit Parallel NOR Flash Memory PC28F00AG18 from Micron. The data I/O of the core is to be connected externally to a tri-state buffer whose output connects to the Flash memory data I/O. It performs only asynchronous read/ write operations in the Flash memory.

5.2 Clocking

Main Clock: **s_axi_aclk**

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

5.3 Resets

Reset: **s_axi_aresetn**

This is active low synchronous reset associated with the **s_axi_aclk**. When asserted, this will reset all the control registers back to their initial default states.

NOTE: A new interrupt will only be generated by another rising edge on an interrupt source.

5.4 Interface Operation

CSR Interface: This is the control/status register interface. It is associated with the **s_axi_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.

5.5 Programming Sequence

This section briefly describes the programming sequence of registers to initiate and complete a transaction on the AXI4-Lite to NOR Flash Interface Core.

1) Set the desired Flash address[31:0].

2) Write/ read data to/ from the Flash memory.

5.6 Timing Diagrams

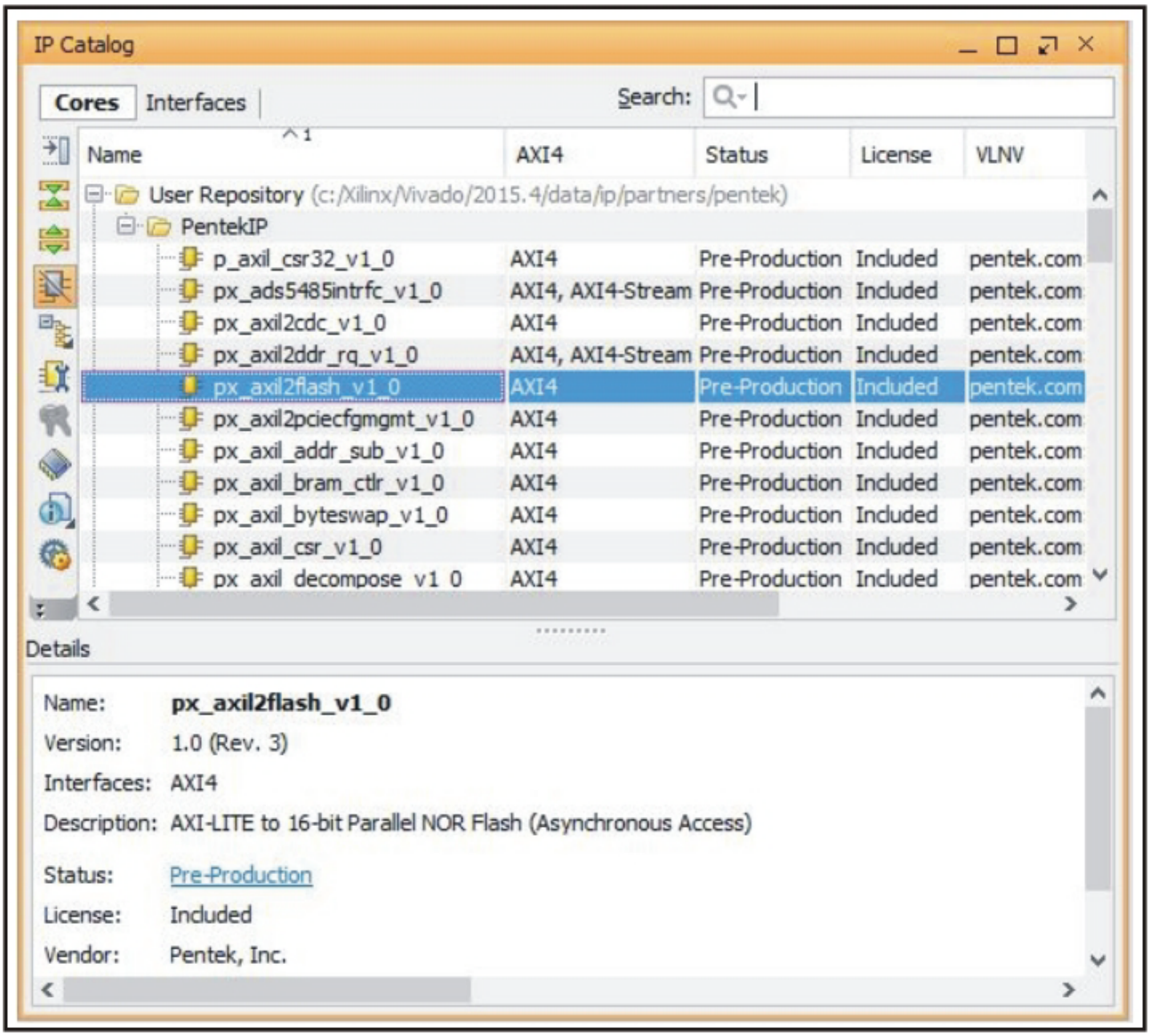
The timing diagram for the AXI4-Lite to NOR Flash Interface Core shown in Figure 6-3, is obtained by running the simulation of the test bench of the core in Vivado VSim environment. For more details about the test bench, refer to Chapter 6.5.

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek AXI4-Lite to NOR Flash Interface 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_axil2flash_v1_0** as shown in Figure 6-1.

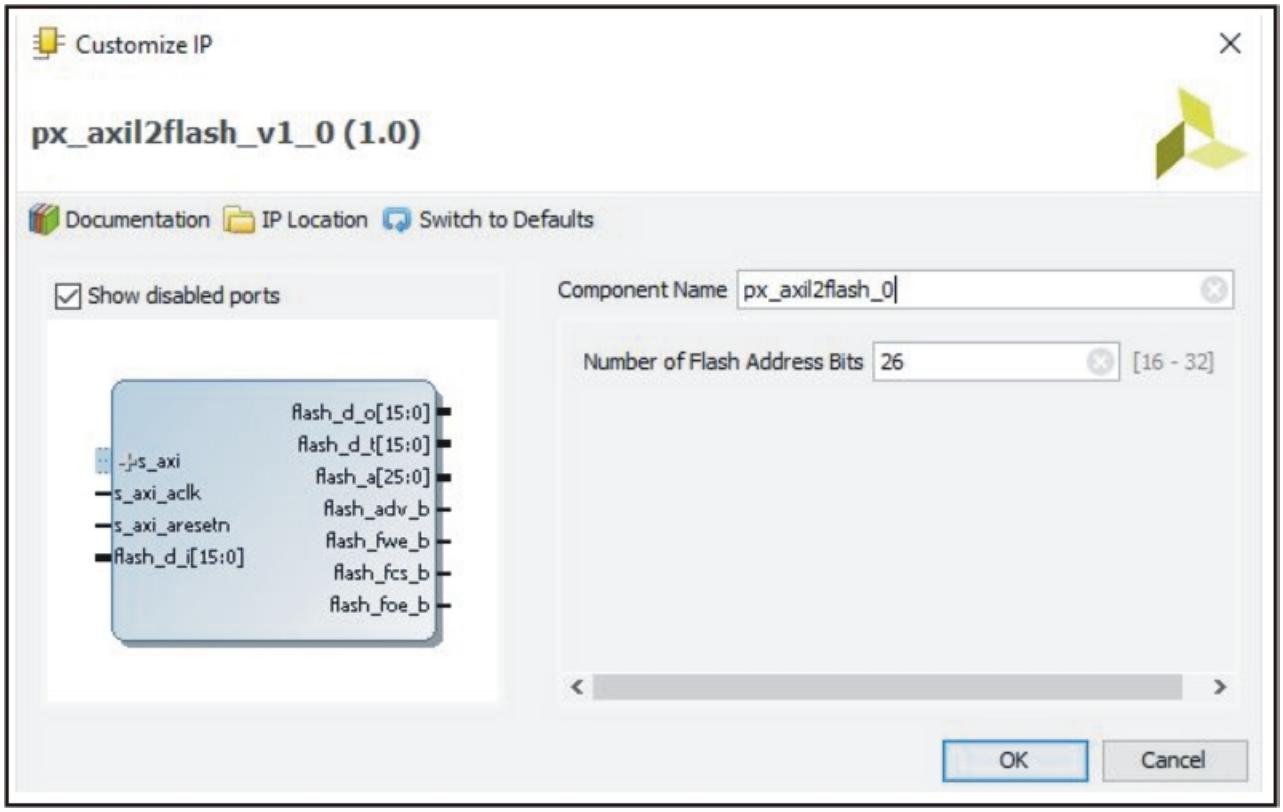
Figure 6-1: AXI4-Lite to 16-bit Parallel NOR Flash Interface Core in



6.1 Pentek IP Catalog (continued)

When you select the **px_axil2flash_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: AXI4-Lite to 16-bit Parallel NOR Flash Interface Core



6.2 User Parameters

For a detailed explanation of the user parameters, refer to Section 2.5.

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 (**s_axi_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.

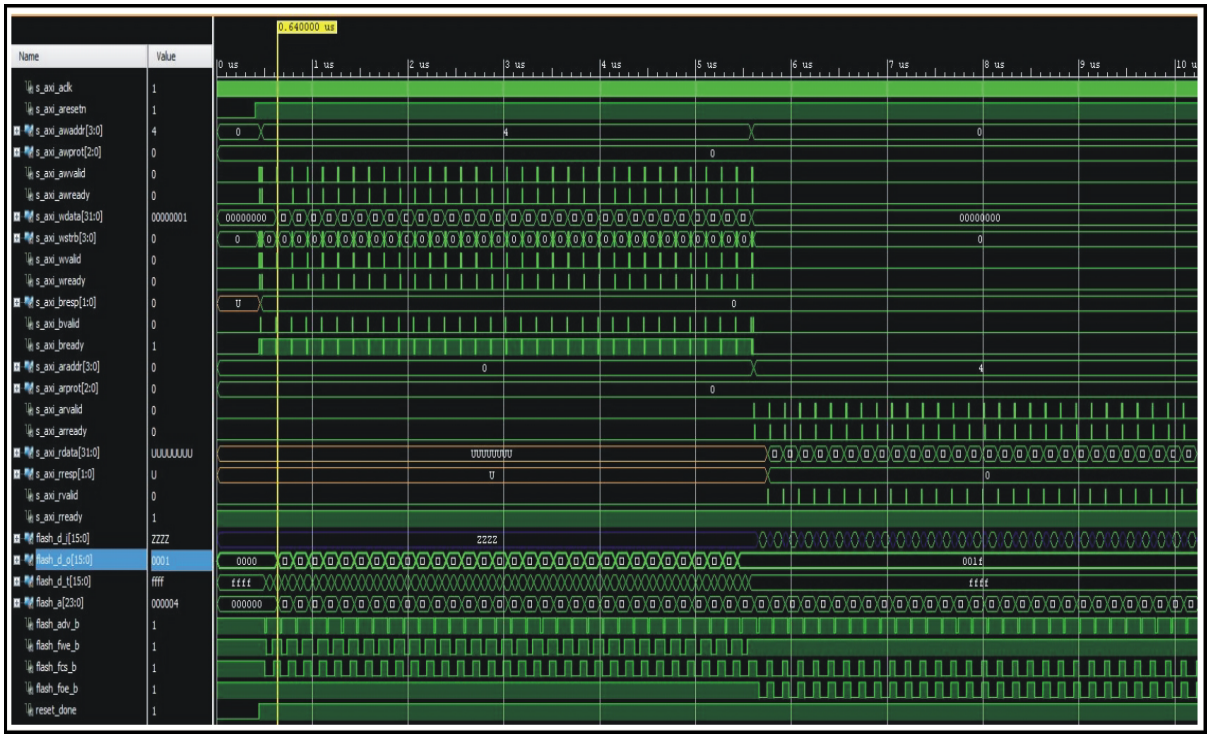
6.5 Simulation

The AXI4-Lite to NOR Flash Interface Core has a test bench which generates the output waveforms using the Vivado VSim environment. The test bench is designed to run at 250 MHz clock frequency.

The test bench sets the Flash Address bus width to 24 bits. It writes 32 DWords to the Flash memory and reads the same from it. Since the AXI4-Lite to NOR Flash Interface Core has auto increment mode, the Flash address is set once in the Flash Address Register and incrementing data (0 to 31) is written into 32 contiguous address locations. The data written is stored in a test memory array and fed as data for Flash read responses.

The programming procedure is the same as described in Section 5.5. When run, the simulation produces the results shown in Figure 6-3.

Figure 6-3: AXI4-Lite to 16-bit Parallel NOR Flash Interface Core Test Bench Simulation



6.6 Synthesis and Implementation

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