

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Acromag / Xembedded / Xycom APA7-500* at our website: *Click HERE*

Xilinx® 50T Artix® 7 FPGA Based AcroPack Modules

USER'S MANUAL

ACROMAG INCORPORATED
30765 South Wixom Road
Wixom, MI 48393-2417 U.S.A.
Tel: (248) 295-0310
Email: solutions@acromag.com

Copyright 2018, Acromag, Inc., Printed in the USA.
Data and specifications are subject to change without notice.
8501137C

Table of Contents

1. RELATED PUBLICATIONS 5
2. GENERAL INFORMATION 6
Intended Audience 6
Preface 6
Trademark, Trade Name and Copyright Information 6
Class A Product Warning 7
Environmental Protection Statement 7
AcroPack Information – All Models 7
Ordering Information 7
Key Features 8
Signal Interface Products 9
PCIe Interface Features 9
Software Support 9
Windows 9
VxWorks 10
Linux 10
ENGINEERING DESIGN KIT 10
3. PREPARATION FOR USE 10
UNPACKING AND INSPECTION 11
Card Cage Considerations 11
Board Installation 11
Non-Isolation Considerations 11
Field I/O Connector 12
4. PROGRAMMING INFORMATION 16
GETTING STARTED 17
FPGA Configuration 17
PCIe CONFIGURATION ADDRESS SPACE 17
CONFIGURATION REGISTERS 18
BAR0 MEMORY MAP 19
CDMA MEMORY MAP 19
INTERRUPT CONTROLLER 20
Interrupt Status Register 20
Interrupt Pending Register 21
Interrupt Enable Register 21
Interrupt Acknowledge Register 22
Set Interrupt Enable Register 23
Clear Interrupt Enable Register 23

Interrupt Vector Register 23
Master Enable Register 23
AXI-CDMA 24
CDMA Control Register 25
CDMA Status Register 27
CDMA Current Descriptor Pointer Register 30
CDMA Tail Descriptor Pointer Register 31
CDMA Source Address Register 31
CDMA Destination Address Register 32
CDMA Bytes to Transfer Register 33
Simple CDMA Programming Example 33
AXI-BAR0 Aperture Base Address 34
PCIe AXI-Bridge Control 35
PHY Status/Control Register 35
AXI Base Address Translation Configuration Register 36
Flash Memory 36
AXI XADC Analog to Digital Converter (System Monitor) 37
Field I/O 38
GPIO_1 "A" Data Register 38
GPIO_1 "A" 3-State Control Register 39
GPIO_2 "B" Data Register 39
GPIO_2 "B" 3-State Control Register 40
AXI Quad SPI 41
Quad SPI Software Reset Register 42
Quad SPI Control Register 42
Quad SPI Status Register 43
Quad SPI Data Transmit Register 46
Quad SPI Data Receive Register 46
Quad SPI Slave Select Register 47
Quad SPI IP Interrupt Status Register 47
Serial Flash Write/Read Example 48
Slot ID Register 49
5. THEORY OF OPERATION 49
PCI INTERFACE LOGIC 50
Clock Generation 50
32MB Serial Flash 50
JTAG Port 50

Encryption Key Storage 50
System Monitor (XADC) 51
6. FPGA FIRMWARE EXAMPLE DESIGN 51
APA7-50X Models Block Diagram Overview 51
7. DESIGN MODIFICATION WALKTHROUGH 54
Copy the Project 55
Edit the Block Diagram 56
Create a New Custom Peripheral 57
Add the Custom Peripheral to the Block Diagram 75
Implement the Modified Design 84
Generate the bitstream 85
Write the bitstream to flash. 86
8. SERVICE AND REPAIR 91
SERVICE AND REPAIR ASSISTANCE 91
PRELIMINARY SERVICE PROCEDURE 91
WHERE TO GET HELP 91
9. SPECIFICATIONS 92
PHYSICAL 92
POWER 92
PCIe BUS COMPLIANCE 92
PCIe Bus Data Rates 93
ENVIRONMENTAL 93
Reliability Prediction 94
All APA7-50X models can use the APA7-201E-LF MTBF Report. 94
FPGA 95
Appendix A 96
AP-CC-01 Heatsink Kit Installation 96
Certificate of Volatility 99
10. Revision History 100

All trademarks are the property of their respective owners.

IMPORTANT SAFETY CONSIDERATIONS

It is very important for the user to consider the possible adverse effects of power, wiring, component, sensor, or software failures in designing any type of control or monitoring system. This is especially important where economic property loss or human life is involved. It is important that the user employ satisfactory overall system design. It is agreed between the Buyer and Acromag, that this is the Buyer's responsibility.

The information contained in this manual is subject to change without notice. Acromag, Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Further, Acromag, Inc. assumes no responsibility for any errors that may appear in this manual and makes no commitment to update, or keep current, the information contained in this manual. No part of this manual may be copied or reproduced in any form, without the prior written consent of Acromag, Inc.

1. RELATED PUBLICATIONS

The following manuals and part specifications provide the necessary information for in depth understanding of the board and the Xilinx Vivado development environment.

Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics	ds181
7 Series FPGAs Memory Resources User Guide	ug473
7 Series FPGAs Configurable Logic Block User Guide	ug474
7 Series FPGAs SelectIO Resources User Guide	ug471
7 Series FPGAs Clocking Resources User Guide	ug472
7 Series DSP48E1 Slice User Guide	ug479
7 Series FPGAs GTX/GTH Transceivers User Guide	ug476
7 Series FPGAs Integrated Block for PCI Express v3.0 Product Guide	pg054
Vivado Design Suite Tutorial Design Flows Overview	ug888
Vivado Design Suite Tutorial Designing IP Subsystems Using IP Integrator	ug995
AXI Reference Guide	ug761
Methods for Integrating AXI4-based IP Using Vivado IP Integrator	xapp1204
LogiCORE IP AXI GPIO v2.0 Product Guide	pg144
LogiCORE IP AXI Central Direct Memory Access v4.1 Product Guide for Vivado Design Suite	pg034

LogiCORE IP AXI Interconnect v2.1 Product Guide	pg059
LogiCORE IP AXI Interrupt Controller (INTC) v4.1 Product Guide for Vivado Design Suite	pg099
LogiCORE IP AXI XADC Wizard v3.3	pg091
LogiCORE IP AXI Quad SPI v3.2 Product Guide	pg153
LogiCORE IP AXI Bridge for PCI Express v2.8	pg055
7 Series FPGAs and Zynq-7000 All Programmable SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter User Guide	ug480
Serial FLASH S25FL132K0XMFI040	S25FL1-K_00

The following resources regarding AcroPack modules are available for download on Acromag's website or by contacting your sales representative.

PCI Express MINI Card Electromechanical Specification, REV 1.2 http://www.pcisig.com

2. GENERAL INFORMATION

Intended Audience

This users' manual was written for technically qualified personnel who will be working with I/O devices using the AcroPack module.

Preface

The information contained in this manual is subject to change without notice, and Acromag, Inc. (Acromag) does not guarantee its accuracy. Acromag makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Further, Acromag assumes no responsibility for any errors that may appear in this manual and makes no commitment to update, or keep current, the information contained in this manual. No part of this manual may be copied or reproduced in any form, without the prior written consent of Acromag.

Trademark, Trade Name and Copyright Information

© 2018 by Acromag Incorporated. All rights reserved. Acromag and Xembedded are registered trademarks of Acromag Incorporated. All other trademarks, registered trademarks, trade names, and service marks are the property of their respective owners.

Class A Product Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may find it necessary to take adequate corrective measures.

Environmental Protection Statement

This product has been manufactured to satisfy environmental protection requirements where possible. Many components used (structural parts, circuit boards, connectors, etc.) are capable of being recycled. Final disposition of this product after its service life must be conducted in accordance with applicable country, state, or local laws or regulations.

AcroPack Information – All Models

The AcroPack IO modules are based on the PCI Express Mini Card Electromechanical specification and are 70mm in length with an added 100 pin field I/O connector.

The AcroPack is 19.05mm longer than the full length mini PCIe card at 50.95mm. It has the same mPCIe board width of 30mm and uses the same mPCIe standard board hold down standoff and screw keep out areas.

Ordering Information

The AcroPack ordering options are given in the following table.

Table 1 APA7-50x Models

MODEL	TTL Channels	EIA-485/422 Channels	LVDS Channels
APA7-501E-LF	48	0	0
APA7-502E-LF	0	24	0
APA7-503E-LF	24	12	0
APA7-504E-LF	0	0	24

Operating temperature range for all models: -40°C to 80°C. Applications requiring operating temperatures above 67°C will require purchase of AcroPack Heatsink Accessory AP-CC-01. See Appendix A for installation instructions.

Acromag provides an engineering design kit for the APA7 series products (sold separately), a "must buy" for first time APA7 series module purchasers. The design kit (model APA7-EDK) provides the user with the basic information required to develop a custom FPGA program for download to the Xilinx FPGA. The design kit includes a CD containing: schematics, parts list, part location drawing, example VHDL source, and other utility files. The APA7 series modules are intended for users fluent in the use of Xilinx FPGA design tools.

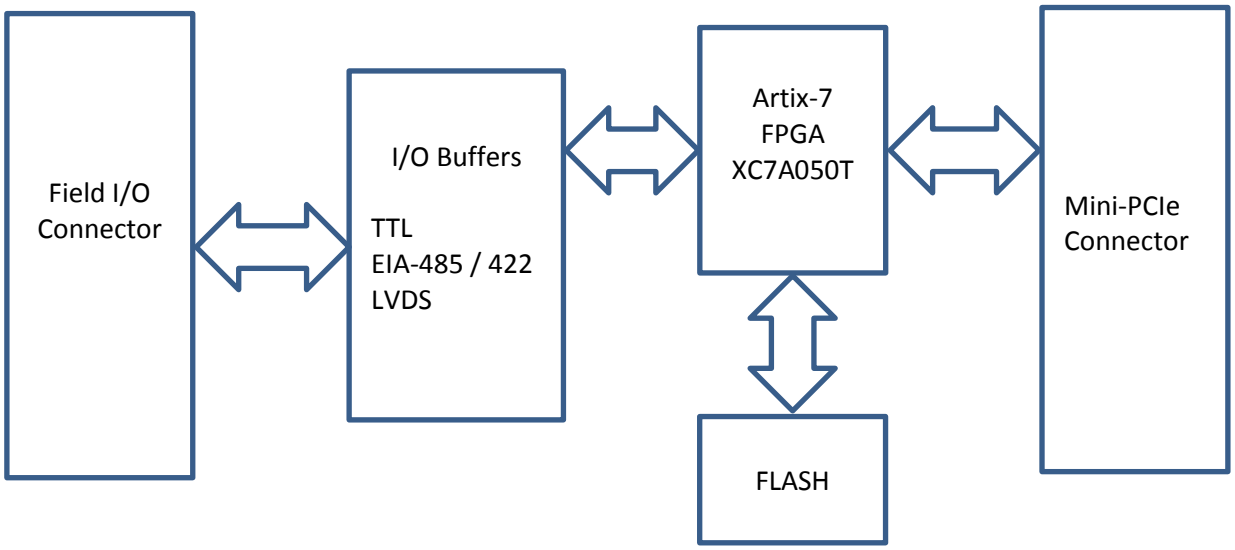


Figure 1 APA7-50X Block Diagram

Key Features

- The APA7-50X series modules provide an FPGA based user configurable bridge between a host processor and a custom digital interface via PCI express. Four models provide a variety of digital I/O standards. These modules feature a user programmable Xilinx Artix-7 series FPGA. Re-configuration of the FPGA is possible via a direct download into the Flash configuration memory over the PCIe bus. The on board Flash memory loaded with configuration data allows automatic configuration of the FPGA on power-up.

These modules include a one lane PCIe (PCI Express) interface with one DMA channel for data transfer between the PCIe bus and FPGA block memory.

The board features a 32 Mb Quad Serial Flash memory. The Flash provides storage for the FPGA configuration.

Reconfigurable Xilinx FPGA – In system configuration of the FPGA is implemented via a standard JTAG interface or the PCIe bus. This provides a means for implementation of custom user defined designs.

- High Channel Count Digital Interface – RS485, LVDS and TTL interface options are available. Models with up to 24 RS485, 24 LVDS, a mix of 12 RS485 and 24 TTL, or up to 48 TTL digital input/output channels are available.
- External Clock – An LVTTL external clock input is available on all models. The external clock is connected directly to a global clock pin on the Xilinx FPGA.
- 5V Tolerance – All TTL I/O are 5V tolerant.

- Channel Input / Output Control – The direction of the TTL digital channels is controlled in groups of eight channels. The direction of the differential digital signals is controlled in groups of four channels.
- Long Distance Data Transmission – Data transmission with RS485/RS422.
- Power Up & System Reset is Failsafe – For safety, all channels are configured as inputs upon power-up and after a system reset.

Signal Interface Products

This AcroPack Module will mate directly to all Acromag AP carriers. Once connected, the module is accessed via a 50 pin front panel connector.

The cables and termination panels are also available. For optimum performance with the APA7-50x modules, use of the shortest possible length of shielded I/O cable is recommended.

- **Reconfigurable Xilinx FPGA** –The FPGA loads its configuration data from flash memory each time power is applied to the module. The host processor can be used to change the flash configuration memory via the PCIe bus. This provides a means for creating custom user defined designs. The FPGA will configure from the updated flash memory on the next power cycle.
- **FPGA** - Xilinx Artix-7 XC7A50T

PCIe Interface Features

- **PCIe Bus** – The example design includes a PCI Express Generation 1 interface operating at a bus speed of 2.5 Gbps per lane per direction.
- **PCIe Bus Master** – The PCIe interface logic becomes the bus master to perform DMA transfers.
- **DMA Operation** – The example design includes a DMA controller to move data between the FPGA memory and the PCIe bus interface.
- **Compatibility** – PCI Express Base Specification v2.1 compliant PCI Express Endpoint. Provides one multifunction interrupt.

Software Support

The APA7-50X series products require support drivers specific to your operating system. Supported operating systems include: Linux®, Windows®, and VxWorks®.

Windows

Acromag provides software products (sold separately) to facilitate the development of Windows applications interfacing with AcroPack modules, VPX I/O board products, and PCIe I/O Cards. This software (model APSW-API-WIN) consists of low-level drivers and Dynamic Link Libraries (DLLs) that are compatible with a number of programming environments. The DLL functions provide a high-level interface to boards eliminating the need to perform low-level reads/writes of registers, and the writing of interrupt handlers.

VxWorks

Acromag provides a software product (sold separately) consisting of VxWorks software. This software (Model APSW-API-VXW) is composed of VxWorks (real time operating system) libraries for all AcroPack modules, VPX I/O board products, and PCIe I/O Cards. The software is implemented as a library of "C" functions which link with existing user code to make possible simple control of all Acromag AcroPack modules.

Linux

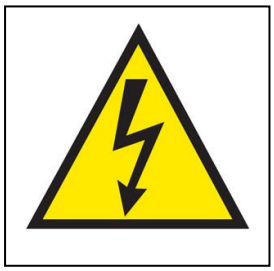
Acromag provides a software product consisting of Linux software. This software (Model APSW-API-LNX) is composed of Linux libraries for all AcroPack modules, VPX I/O board products, and PCIe I/O Cards. The software is implemented as a library of "C" functions which link with existing user code to make possible simple control of all Acromag AcroPack modules.

ENGINEERING DESIGN KIT

Acromag provides an engineering design kit for the AP products (sold separately), a "must buy" for first time APA7-50X series module purchasers. The design kit (model APA7-EDK) provides the user with the basic information required to develop a custom FPGA program for download to the Xilinx FPGA. The design kit includes a CD containing: schematic, parts lists, part location drawings, example Vivado™ projects, and other utility files. The Artix 7 series modules are intended for users fluent in the use of Xilinx FPGA design tools.

3. PREPARATION FOR USE

IMPORTANT PERSONAL AND PRODUCT SAFETY CONSIDERATIONS





It is very important for the user to consider the possible safety implications of power, wiring, component, sensor, or software failures in designing any type of control or monitoring system. This is especially important where personal injury or the loss of economic property or human life is possible. It is important that the user employ satisfactory overall system design. It is understood and agreed by the Buyer and Acromag that this is the Buyer's responsibility.

WARNING: This board utilizes static sensitive components and should only be handled at a static-safe workstation. This product is an electrostatic sensitive device and is packaged accordingly. Do not open or handle this product except at an electrostatic-free workstation. Additionally, do not ship or store this product near strong electrostatic, electromagnetic, magnetic, or radioactive fields unless the device is contained within its original manufacturer's packaging. Be aware that failure to comply with these guidelines will void the Acromag Limited Warranty.

UNPACKING AND INSPECTION

Upon receipt of this product, inspect the shipping carton for evidence of mishandling during transit. If the shipping carton is badly damaged or water stained, request that the carrier's agent be present when the carton is opened. If the carrier's agent is absent when the carton is opened and the contents of the carton are damaged, keep the carton and packing material for the agent's inspection.

For repairs to a product damaged in shipment, refer to the Acromag Service Policy to obtain return instructions. It is suggested that salvageable shipping cartons and packing material be saved for future use in the event the product must be shipped.

This board is physically protected with packing material and electrically protected with an anti-static bag during shipment. It is recommended that the board be visually inspected for evidence of mishandling prior to applying power.

The board utilizes static sensitive components and should only be handled at a static-safe workstation.

Card Cage Considerations

Refer to the specifications section for loading and power requirements. Be sure that the system power supplies are able to accommodate the power requirements of the system boards, plus the installed Acromag board, within the voltage tolerances specified.

In an air cooled assembly, adequate air circulation must be provided to prevent a temperature rise above the maximum operating temperature and to prolong the life of the electronics. If the installation is in an industrial environment and the board is exposed to environmental air, careful consideration should be given to air-filtering.

In a conduction cooled assembly, adequate thermal conduction must be provided to prevent a temperature rise above the maximum operating temperature.

Board Installation

Power should be removed from the board when installing AP modules, cables, termination panels, and field wiring. Model APA7-50x digital input boards have no hardware jumpers or switches to configure.

Non-Isolation Considerations

The board is non-isolated, since there is electrical continuity between the logic and field I/O grounds. As such, the field I/O connections are not isolated from the system. Care should be taken in designing installations without isolation to avoid noise pickup and ground loops caused by multiple ground connections.

Field I/O Connector

The field I/O interface connector P2 provides a mating interface between the APA7-50x modules and the carrier board.

Table 2 lists pin assignments for each of the APA7-50x field I/O signals. Every other pin of the 100 pin P2 connector is left unconnected.

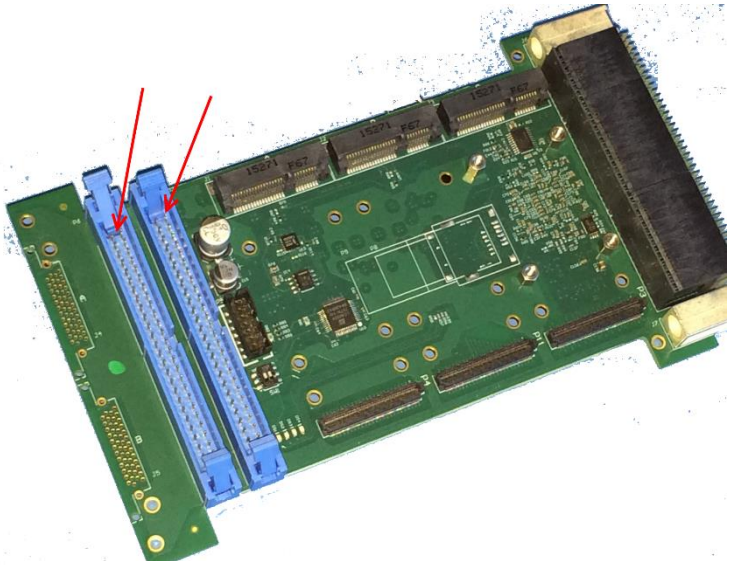
Table 2 Field I/O Connector Pin Assignments

Carrier Connector Ribbon[1]	Carrier Connector 50 pin Champ[2]	Carrier Connector 68 pin Champ	Module P2 Pin Number	Field I/O Signals		
				APA7-501	APA7-502 APA7-504	APA7-503
1	1	1	2	A1	B1_P	A1
2	26	35	1	A2	B1_N	A2
			4	Reserved/isolation		
			3			
3	2	2	6	A3	B2_P	A3
4	27	36	5	A4	B2_N	A4
			8	Reserved/isolation		
			7			
5	3	3	10	A5	B3_P	A5
6	28	37	9	A6	B3_N	A6
			12			
			11			
7	4	4	14	A7	B4_P	A7
8	29	38	13	A8	B4_N	A8
			16	Reserved/isolation		
			15			
9	5	5	18	A9	B5_P	A9
10	30	39	17	A10	B5_N	A10
			20	Reserved/isolation		
			19			
11	6	6	22	A11	B6_P	A11
12	31	40	21	A12	B6_N	A12
			24	Reserved/isolation		
			23			
13	7	7	26	A13	B7_P	A13
14	32	41	25	A14	B7_N	A14
			28	Reserved/isolation		
			27	Reserved/isolation		
15	8	8	30	A15	B8_P	A15
16	33	42	29	A16	B8_N	A16

Carrier Connector Ribbon[1]	Carrier Connector 50 pin Champ[2]	Carrier Connector 68 pin Champ	Module P2 Pin Number	Field I/O Signals		
				APA7-501	APA7-502 APA7-504	APA7-503
			32	Reserved/isolation		
			31			
17	9	9	34	A17	B9_P	A17
18	34	43	33	A18	B9_N	A18
			36	Reserved/isolation		
			35			
19	10	10	38	A19	B10_P	A19
20	35	44	37	A20	B10_N	A20
			40	Reserved/isolation		
			39			
21	11	11	42	A21	B11_P	A21
22	36	45	41	A22	B11_N	A22
			44	Reserved/isolation		
			43			
23	12	12	46	A23	B12_P	A23
24	37	46	45	A24	B12_N	A24
			48	Reserved/isolation		
			47			
25	13	13	50	B25	B13_P	B1_P
26	38	47	49	B26	B13_N	B1_N
			52	Reserved/isolation		
			51			
27	14	14	54	B27	B14_P	B2_P
28	39	48	53	B28	B14_N	B2_N
			56	Reserved/isolation		
			55			
29	15	15	58	B29	B15_P	B3_P
30	40	49	57	B30	B15_N	B3_N
			60	Reserved/isolation		
			59			
31	16	16	62	B31	B16_P	B4_P
32	41	50	61	B32	B16_N	B4_N
			64	Reserved/isolation		
			63	Reserved/isolation		
33	17	17	66	B33	B17_P	B5_P
34	42	51	65	B34	B17_N	B5_N

Carrier Connector Ribbon[1]	Carrier Connector 50 pin Champ[2]	Carrier Connector 68 pin Champ	Module P2 Pin Number	Field I/O Signals		
				APA7-501	APA7-502 APA7-504	APA7-503
			68	Reserved/isolation		
			67			
35	18	18	70	B35	B18_P	B6_P
36	43	52	69	B36	B18_N	B6_N
			72	Reserved/isolation		
			71			
37	19	19	74	B37	B19_P	B7_P
38	44	53	73	B38	B19_N	B7_N
			76	Reserved/isolation		
			75			
39	20	20	78	B39	B20_P	B8_P
40	45	54	77	B40	B20_N	B8_N
			80	Reserved/isolation		
			79			
41	21	21	82	B41	B21_P	B9_P
42	46	55	81	B42	B21_N	B9_N
			82	Reserved/isolation		
			83			
43	22	22	86	B43	B22_P	B10_P
44	47	56	85	B44	B22_N	B10_N
			88	Reserved/isolation		
			87			
45	23	23	90	B45	B23_P	B11_P
46	48	57	89	B46	B23_N	B11_N
			92	Reserved/isolation		
			91			
47	24	24	94	B47	B24_P	B12_P
48	49	58	93	B48	B24_N	B12_N
			96	Reserved/isolation		
			95			
49	25	25	98	EXT_CLK	EXT_CLK	EXT_CLK
50	50	59	97	GND	GND	GND
			100	Reserved/isolation		
			99	Reserved/isolation		

Note 1: VPX4500-CC-LF is an example of a carrier that uses the ribbon cable connector see image of carrier.



Note 2: APCe7020E-LF is an example of a carrier that uses the Champ connector see image of carrier.

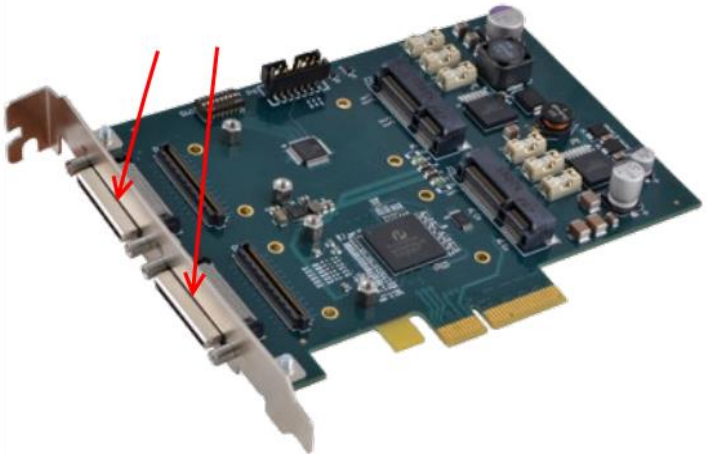


Table 3 Mini-PCIe Connector

Pin #	Name	Pin #	Name
51	**+5V**[2]	52	+3.3V[3]
49	**+12V**[2]	50	GND
47	**-12V**[2]	48	+1.5V
45	**Present**[2]	46	N.C. (LED_WPAN#)[1]
43	GND	44	N.C. (LED_WLAN#)[1]
41	+3.3V[3]	42	N.C. (LED_WWAN#)[1]
39	+3.3V[3]	40	GND
37	GND	38	N.C. (USB_D+)[1]
35	GND	36	N.C. (USB_D-)[1]
33	PETp0	34	GND
31	PETn0	32	SMB_DATA[4]
29	GND	30	SMB_CLK[4]
27	GND	28	+1.5V
25	PERp0	26	GND
23	PERn0	24	+3.3V[3]
21	GND	22	PERST#
19	**TDI** (UIM_C4)[1]	20	N.C. (W_DISABLE#)[1]
17	**TDO** (UIM_C8)[1]	18	GND
15	GND	16	UIM_VPP
13	RECLK+	14	UIM_RESET
11	REFCLK-	12	UIM_CLK
9	GND	10	UIM_DATA
7	CLKREQ#	8	UIM_PWR
5	**TCK** (COEX2)[1]	6	+1.5V
3	**TMS** (COEX1)[1]	4	GND
1	N.C. (WAKE#)[1]	2	+3.3V[3]

1. **Note 1:** The following mini-PCIe signals are not used by the APA7-50X: USB_D+, USB_D-, WAKE#, LED_WPAN#, LED_WLAN#, LED_WWAN#, W_DISABLE#, COEX1, COEX2, UIM_C4, UIM_C8.
2. **Note 2:** +5, +12, and -12 Volt power supplies have been assigned to pins that are reserved in the mini-PCIe specification. The Present signal is tied to circuit common on the AP module.
3. **Note 3:** All +3.3Vaux power pins are changed to system +3.3V power.
4. **Note 4:** The SM bus signals SMB_CLK and SMB_DATA will be used to clock a carrier location serial stream from the carrier. These signals are under the control of the AcroPack module.

4. PROGRAMMING INFORMATION

This Section provides the specific information necessary to program and operate the board.

GETTING STARTED

1. The APA7-50X series modules are shipped with the example design FPGA bitstream stored in the flash memory. This example design bitstream operates with the driver software included in the support package. Upon power-up the APA7-50X series module will automatically configure the FPGA with the example design bitstream stored in flash. As a first step become familiar with the appropriate example design for your model. The board will perform all the functions of the example design as described in this manual.

The Example Design Memory Map section gives a description of the I/O registers provided by the example design. It will allow testing of the PCIe interface, read/write of flash memory, all digital I/O ports, interrupts, and DMA operation. It is strongly recommended that you become familiar with the board features by using the example design as provided by Acromag.

CAUTION: Do not attempt to reconfigure the flash memory until after you have tested and become familiar with the APA7-50X series module as provided in the example design.

2. After you are familiar with the module's features and have tested it using the example design, proceed to the DESIGN MODIFICATION WALK THROUGH. Here you will modify the example design VHDL code slightly. The configuration flash must be overwritten to test your design. Once the flash is erased you will not be able to go back to the example design by simply powering down and restarting the board. If your code does not function as desired, you may need to reload the Acromag design example. You can reload the Acromag example design via the JTAG port using the Xilinx Vivado tool. Upon power-up, the example design provided by Acromag will again be loaded into the FPGA.

See the Flash Configuration section for a description of the steps required to write new data or to reprogram the example design code to the flash device.

FPGA Configuration

The APA7-50X series modules support configuration in the following modes:

- Master Quad SPI using serial Flash device.
- JTAG using Xilinx external program cable.

The FPGA configuration bitstream is loaded from flash memory on power-up.

PCIe CONFIGURATION ADDRESS SPACE

This APA7-50X series modules are PCI Express Base Specification Revision v2.1 compliant.

The PCIe bus is defined to address three distinct address spaces: I/O, memory, and configuration space. This board can be accessed via the PCIe bus memory, and configuration spaces.

The card's configuration registers are initialized by system software at power-up to configure the card. The board is a Plug-and-Play PCIe card. As a Plug-

and-Play card the board's base address and system interrupt request are not selected via jumpers but are assigned by system software upon power-up via the configuration registers. A PCIe bus configuration access is used to read/write the PCIe card's configuration registers.

When the computer is first powered-up, the computer's system configuration software scans the PCIe bus to determine what PCIe devices are present. The software also determines the configuration requirements of the PCIe card.

The system software accesses the configuration registers to determine how many blocks of memory space the module requires. It then programs the board's configuration registers with the unique memory base address.

Since this board is not fixed in address space, its device driver must use the mapping information stored in the board's Configuration Space registers to determine where the board is mapped in memory space.

The configuration registers are also used to indicate that the board requires an interrupt request. The system software then programs the configuration registers with the interrupt request assigned to the board.

CONFIGURATION REGISTERS

The PCIe specification requires software driven initialization and configuration via the Configuration Address space. This board provides 512 bytes of configuration registers for this purpose. It contains the configuration registers shown in Table 4 to facilitate Plug-and-Play compatibility.

The Configuration Registers are accessed via the Configuration Address and Data Ports. The most important Configuration Registers are the Base Address Registers and the Interrupt Register which must be read to determine the base address assigned to the board and the interrupt request that goes active on a board interrupt request.

Table 4 Configuration Registers

Reg. Num.	D31 D24	D23 D16	D15 D8	D7 D0
0	**Device ID** 0x7051 APA7-501E-LF 0X7052 APA7-502E-LF 0x7053 APA7-503E-LF 0x7054 APA7-504E-LF		**Vendor ID** 16D5	
1	Status		Command	
2	Class Code=118000			Rev ID=00
3	BIST	Header	Latency	Cache
4:5	64-bit Memory Base Address for Memory Accesses to PCIe interrupt, I/O registers, System Monitor registers, and Flash memory. 128K Space **(BAR0)**			
6:10	Not Used			

11	**Subsystem ID** 0x7051 APA7-501E-LF 0X7052 APA7-502E-LF 0x7053 APA7-503E-LF 0x7054 APA7-504E-LF		**Subsystem Vendor ID** 16D5	
12	Not Used			
13,14	Reserved			
15	Max_Lat	Min_Gnt	Inter. Pin	Inter. Line

This board is allocated a 4M byte block of memory (BAR0), to access the PCIe interrupt, I/O registers, XADC registers, and Flash memory. The PCIe bus decodes 4M bytes for BAR0 for this memory space.

BAR0 MEMORY MAP

The BAR0 memory address space is used to access the PCIe interrupt, Front I/O, Rear I/O, and P16 I/O registers, System Monitor registers, and Flash memory. Note that the base address for the board (BAR0) in memory space must be added to the addresses shown to properly access these registers.

Table 5 BAR0 Registers

BAR0 Base Address	Size	Description
0x0000_0000→0x0000_0FFF	4K	CDMA (see PG034)
0x0000_1000→0x0000_1FFF	4K	PCIe AXI Bridge Control (see PG054)
0x0000_2000→0x0000_2FFF	4K	Interrupt Controller (see PG099)
0x0000_3000→0x0000_3FFF	4K	XADC System Monitor (see DS790)
0x0000_4000→0x0000_4FFF	4K	Field I/O AXI_GPIO (see PG144)
0x0000_5000→0x0000_5FFF	4K	Flash AXI_QSPI (see pg153)
0x0000_6000→0x0000_6FFF	4K	Slot ID Register
0x0000_7000→0x0000_FFFF	36K	Reserved
0x0001_0000→0x0001_7FFF	32K	Block RAM

CDMA MEMORY MAP

The Central Direct Memory Access (CDMA) controller can access the following devices: Block RAM, AXI to PCI bridge (BAR0), and the GPIO registers.

BAR0 Base Address	Size	Description
0x0000_0000→0x0000_3FFF	16K	Reserved
0x0000_4000→0x0000_4FFF	4K	Field I/O AXI_GPIO (see PG144)

BAR0 Base Address	Size	Description
0x0000_5000→0x0000_FFFF	44K	Reserved
0x0001_0000→0x0001_7FFF	32K	Block RAM
0x0001_8000→0x00FF_FFFF	15M	Reserved
0x0100_0000→0x01FF_FFFF	16M	AXI to PCIe BAR0

INTERRUPT CONTROLLER

The AXI Interrupt Controller concentrates multiple interrupt inputs from peripheral devices to a single interrupt output to the system processor using the PCIe bus. The interrupt controller contains programmer accessible registers that allow interrupts to be enabled, queried and cleared under software control over the PCIe bus interface.

Table 6 Interrupt Controller Registers

BAR0 Base Addr+	Bit(s)	Description
0x0000_2000	31:0	Interrupt Status Register
0x0000_2004	31:0	Interrupt Pending Register
0x0000_2008	31:0	Interrupt Enable Register
0x0000_200C	31:0	Interrupt Acknowledge Register
0x0000_2010	31:0	Set Interrupt Enable Register
0x0000_2014	31:0	Clear Interrupt Enable Register
0x0000_2018	31:0	Interrupt Vector Register
0x0000_201C	31:0	Master Enable Register

Note that any registers/bits not mentioned will remain at the default value: logic low.

Interrupt Status Register

This Interrupt Status register (ISR) at BAR0 base address + offset 0x2000 is used to monitor board interrupts. When read, the contents of this register indicate the presence or absence of an active interrupt for each of the active interrupting sources. Each bit in this register that is set to a '1' indicates an active interrupt signal on the corresponding interrupt input. Bits that are '0'are not active. The bits in the ISR are independent of the interrupt enable

bits in the Interrupt Enable register. Interrupts, even if not enabled can still show up as active in the ISR.

Table 7 Interrupt Status Register (Read/Write) - (BAR0 + 0x2000)

Bit(s)	FUNCTION	
0	When set indicates an AXI CDMA interrupt. See the CDMA section for more information on the source of this interrupt.	
	0	Interrupt Inactive
	1	Interrupt Active
31-1	Reserved	
	0	NA
	1	NA

The ISR register is writable by software only until the Hardware Interrupt Enable bit in the MER has been set. Given these restrictions, when this register is written to, any data bits that are set to '1' will activate the corresponding interrupt just as if a hardware input became active. Data bits that are zero have no effect. This allows software to generate interrupts for test purposes.

Interrupt Pending Register

This Interrupt Pending register (IPR) at BAR0 base address + offset 0x2004 is used to monitor board interrupts. Reading the contents of this register indicates the presence or absence of an active interrupt signal that is also enabled. Each bit in this register is the logical AND of the bits in the Interrupt Status register and the Interrupt Enable register.

Table 8 Interrupt Pending Register (Read) - (BAR0 + 0x2004)

Bit(s)	FUNCTION	
0	When set indicates an AXI CDMA interrupt is pending. See the CDMA section for more information on the source of this	
	0	No Interrupt Pending
	1	Interrupt Pending
31-1	Reserved	
	0	NA
	1	NA

Interrupt Enable Register

This is a read/write register. Writing a '1' to a bit in this register enables the corresponding Interrupt Status bit to cause assertion of the interrupt output. This Interrupt Enable bit set to '0' does not inhibit an interrupt condition from being captured. It will still show up in the Interrupt Status register even when not enabled here. To show up in the Interrupt Pending register it needs to be enabled here. Writing a '0' to a bit disables, or masks, the generation of

interrupt output for the corresponding interrupt input signal. Note however, that disabling an interrupt input is not the same as clearing it. Disabling an active interrupt prevents that interrupt from reaching the IRQ output. When it is re-enabled, the interrupt immediately generates a request on the IRQ output. An interrupt must be cleared by writing to the Interrupt Acknowledge Register, as described below. Reading this Interrupt Enable register indicates which interrupt inputs are enabled; where a '1' indicates the input is enabled and a '0' indicates the input is disabled.

Table 9 Interrupt Enable Register (Read/Write) - (BAR0 + 0x2008)

Bit(s)	FUNCTION	
0	When set indicates an AXI CDMA interrupt is enabled. See the CDMA section for more information on the source of this interrupt.	
	0	Interrupt Disabled
	1	Interrupt Enabled
31-1	Reserved	
	0	NA
	1	NA

Interrupt Acknowledge Register

The Interrupt Acknowledge register is a write-only location that clears the interrupt request associated with selected interrupt inputs. Note that writing one to a bit in Interrupt Acknowledge register clears the corresponding bit in Interrupt Status register, and also clears the same bit itself in the Interrupt Acknowledge register.

Writing a '1' to a bit location in the Interrupt Acknowledge register will clear the interrupt request that was generated by the corresponding interrupt input. An interrupt input that is active and masked by writing a '0' to the corresponding bit in the Interrupt Enable register will remain active until cleared by acknowledging it. Unmasking an active interrupt causes an interrupt request output to be generated (if the Master Interrupt Enable bit-0 in the Master Enable register is set). Writing 0s has no effect as does writing a '1' to a bit that does not correspond to an active input or for which an interrupt input does not exist. The bit locations in the Interrupt Acknowledge register correspond with the bit locations given in the Interrupt Enable Register Table.

Table 10 Interrupt Acknowledge Register (Write) - (BAR0 + 0x200C)

Bit(s)	FUNCTION
0	Clear AXI CDMA interrupt request.
31-1	Reserved

Set Interrupt Enable Register

Set Interrupt Enable register is a location used to set Interrupt Enable register bits in a single atomic operation, rather than using a read / modify / write sequence. Writing a '1' to a bit location in the Set Interrupt Enable register will set the corresponding bit in the Interrupt Enable register. Writing 0s does nothing, as does writing a '1' to a bit location that corresponds to a non-existing interrupt input. The bit locations in the Set Interrupt Enable correspond with the bit locations given in the Interrupt Enable Register Table.

Table 11 Set Interrupt Enable Register (Write) - (BAR0 + 0x2010)

Bit(s)	FUNCTION
0	Set AXI CDMA interrupt request.
31-1	Reserved

Clear Interrupt Enable Register

Clear Interrupt Enable register is a location used to clear Interrupt Enable register bits in a single atomic operation, rather than using a read / modify / write sequence. Writing a '1' to a bit location in Clear Interrupt Enable register will clear the corresponding bit in the Interrupt Enable register. Writing 0s does nothing, as does writing a '1' to a bit location that corresponds to a non-existing interrupt input. The bit locations in the clear Interrupt Enable correspond with the bit locations given in the Interrupt Enable Register Table.

Table 12 Clear Interrupt Enable Register (Write) - (BAR0 + 0x2014)

Bit(s)	FUNCTION
0	Clear AXI CDMA interrupt request.
31-1	Reserved

Interrupt Vector Register

The Interrupt Vector register is a read-only register and contains the ordinal value of the highest priority, enabled, and active interrupt input. INT0 (always the LSB) is the highest priority interrupt input. Each successive input (to the left) has a corresponding lower interrupt priority. If no interrupt inputs are active, the Interrupt Vector register contains all 1s. This Interrupt Vector register acts as an index for giving the correct Interrupt Vector Address.

Table 13 Interrupt Vector Register (Read) - (BAR0 + 0x2018)

Bit(s)	FUNCTION
31-0	Ordinal value of the highest priority enabled active interrupt, 0xFFFFFFFF if no interrupt inputs are active

Master Enable Register

This is a 2-bit, read / write register. The two bits are mapped to the two least significant bits of the location. The least significant bit contains the Master Enable bit and the next bit contains the Hardware Interrupt Enable bit.

Writing a '1' to the Master Enable bit enables the IRQ output signal. Writing a '0' to the Master Enable bit disables the IRQ output, effectively masking all interrupt inputs. The Hardware Interrupt Enable bit is a write-once bit. At reset, this bit is reset to '0', allowing the software to write to the Interrupt Status register to generate interrupts for testing purposes, and disabling any hardware interrupt inputs. Writing a '1' to this bit enables the hardware interrupt inputs and disables software generated inputs. Writing a '1' also disables any further changes to this bit until the device has been reset. Writing 1s or 0s to any other bit location does nothing. When read, this register will reflect the state of the Master Enable and Hardware Interrupt Enable bits. All other bits will read as 0s.

Table 14 Master Enable Register (Read/Write) - (BAR0 + 0x201C)

Bit(s)	FUNCTION	
0	Master IRQ Enable	
	0	All Interrupts Disabled
	1	All Interrupts Enabled
1	Hardware Interrupt Enable	
	0	Software Interrupts Enabled
	1	Hardware Interrupts Only Enabled
31-2	Not Used (bits are read as logic "0")	

AXI-CDMA

The AXI Central Direct Memory Access (CDMA) core is a soft Xilinx Intellectual Property core. The CDMA provides direct memory access between system memory over the PCIe bus and the memory resident on the APA7-50X series module.

The basic mode of operation for the CDMA is Simple DMA. In this mode, the CDMA executes one programmed DMA command and then stops. This requires that the CDMA registers need to be set up by system software over the PCIe bus for each DMA operation required.

Scatter Gather is a mechanism that allows for automated DMA transfer scheduling via a pre-programmed instruction list of transfer descriptors (Scatter Gather Transfer Descriptor Definition). This instruction list is programmed by the user software application into a memory-resident data structure that must be accessible by the AXI CDMA Scatter Gather interface. This list of instructions is organized into what is referred to as a transfer descriptor chain. Each descriptor has an address pointer to the next descriptor to be processed. The last descriptor in the chain generally points back to the first descriptor in the chain but it is not required. The AXI CDMA Tail Descriptor Pointer register needs to be programmed with the address of the first word of the last descriptor of the chain. When the AXI CDMA executes the last descriptor and finds that the Tail Descriptor pointer register contents matches the address of the completed descriptor, the Scatter Gather Engine stops descriptor fetching and waits. See the Xilinx AXI Central Direct

Memory Access product guide PG034 for additional details for Scatter Gather operations.

Table 15 AXI CDMA Registers

BAR0 Base Addr+	Bit(s)	Description
0x0000_0000	31:0	CDMA Control Register
0x0000_0004	31:0	CDMA Status Register
0x0000_0008	31:0	Current Descriptor Pointer Register
0x0000_000C	31:0	Reserved
0x0000_0010	31:0	Tail Descriptor Pointer Register
0x0000_0014	31:0	Reserved
0x0000_0018	31:0	Source Address Register
0x0000_001C	31:0	Reserved
0x0000_0020	31:0	Destination Address Register
0x0000_0024	31:0	Reserved
0x0000_0028	31:0	Bytes to Transfer Register

Note that any registers/bits not mentioned will remain at the default value: logic low.

CDMA Control Register

This register provides software application control of the AXI CDMA.

Table 16 CDMA Control Register (Read/Write) - (BAR0 + 0x0000)

Bit(s)	FUNCTION	
0	This bit is reserved for future definition and will always return zero.	
1	Indicates tail pointer mode is enabled to the Scatter Gather Engine. This bit is fixed to 1 and always read as 1 when Scatter Gather is included. If the CDMA is built with Scatter Gather disabled (Simple Mode Only), the default value of the port is 0.	
	0	Tail Pointer Mode is Disabled
	1	Tail Pointer Mode is Enabled
2	Soft reset control for the AXI CDMA core. Setting this bit to a '1' causes the AXI CDMA to be reset. Reset is accomplished gracefully. Committed AXI4 transfers are then completed. Other queued transfers are flushed. After completion of a soft reset, all registers and bits are in the Reset State.	
	0	Reset Not in Progress

Bit(s)	FUNCTION	
	1	Reset in Progress
3	This bit controls the transfer mode of the CDMA. Setting this bit to a '1' causes the AXI CDMA to operate in a Scatter Gather mode. **Note:** This bit must only be changed when the CDMA engine is IDLE (CDMA Status bit-1 = '1'). Changing the state of this bit at any other time has undefined results. **Note:** This bit must be set to a 0 then back to 1 by the software application to force the CDMA Scatter Gather engine to use a new value written to the CDMA Current Descriptor Pointer register. **Note:** This bit must be set prior to setting Bit-13 of this CDMA Control register.	
	0	Simple DMA Mode
	1	Scatter Gather Mode
11-4	Reserved	
12	Interrupt on Complete Interrupt Enable. When set to '1', it allows an interrupt after completed DMA transfers.	
	0	Interrupt on Complete Disabled
	1	Interrupt on Complete Enabled
13	Interrupt on Delay Timer Interrupt Enable. When set to '1', it allows a delayed interrupt out. This is only used with Scatter Gather assisted transfers.	
	0	Delayed Interrupt Disabled
	1	Delayed Interrupt Enabled
14	Interrupt on Error Interrupt Enable. When set to '1', it allows an error to generate an interrupt out.	
	0	Error Interrupt Disabled
	1	Error Interrupt Enabled
15	Reserved	
23-16	Interrupt Threshold value. This field is used to set the Scatter Gather interrupt coalescing threshold. When Interrupt On Complete interrupt events occur, an internal counter counts down from the Interrupt Threshold setting. When the count reaches zero, an interrupt out is generated by the CDMA engine. **Note:** The minimum setting for the threshold is 0x01. A write of 0x00 to this register has no effect. If the CDMA is built with Scatter Gather disabled (Simple Mode Only), the default value of the port is zeros.	

Bit(s)	FUNCTION
31-24	Interrupt Delay Time Out. This value is used for setting the interrupt delay time out value. The interrupt time out is a mechanism for causing the CDMA engine to generate an interrupt after the delay time period has expired. This is used for cases when the interrupt threshold is not met after a period of time, and the CPU desires an interrupt to be generated. Timer begins counting when the CDMA is IDLE (CDMA Status bit-1 = '1'). This generally occurs when the CDMA has completed all scheduled work defined by the transfer descriptor chain (reached the tail pointer) and has not satisfied the Interrupt Threshold count. **Note:** Setting this value to zero disables the delay timer interrupt.

CDMA Status Register

This register provides status of the AXI CDMA.

Table 17 CDMA Status Register (Read/Write) - (BAR0 + 0x0004)

Bit(s)	FUNCTION	
0	This bit is reserved for future definition and will always return zero.	
1	CDMA Idle. Indicates the state of AXI CDMA operations. When set and in Simple DMA mode, the bit indicates the programmed transfer has completed and the CDMA is waiting for a new transfer to be programmed. Writing to the "Bytes to Transfer" register in Simple DMA mode causes the CDMA to start (not Idle). When set and in Scatter Gather mode, the bit indicates the Scatter Gather Engine has reached the tail pointer for the associated channel and all queued descriptors have been processed. Writing to the tail pointer register automatically restarts CDMA Scatter Gather operations.	
	0	Not Idle
	1	CDMA is Idle
2	Reserved	
3	Scatter Gather Included. This bit indicates if the AXI CDMA has been implemented with Scatter Gather support included (C_SG_ENABLE = 1). This is used by application software (drivers) to determine if Scatter Gather Mode can be utilized.	
	0	Scatter Gather not included
	1	Scatter Gather is included

Bit(s)	FUNCTION	
4	DMA Internal Error. This bit indicates that an internal error has been encountered by the DataMover on the data transport channel. This error can occur if a 0 value Byte to Transfer register is fed to the AXI DataMover or DataMover has an internal processing error. A Bytes to Transfer register value of 0 only happens if the register is written with zeros (in Simple DMA mode) or a Bytes to Transfer register value of zero is specified in the Control word of a fetched descriptor is set to 0 (Scatter Gather Mode). This error condition causes the AXI CDMA to gracefully halt. The CDMA Status register bit-1 is set to '1'when the CDMA has completed shut down. A reset (soft or hard) must be issued to clear the error condition.	
	0	No CDMA Internal Errors
	1	CDMA Internal Error detected. CDMA Engine halts.
5	DMA Slave Error. This bit indicates that an AXI slave error response has been received by the AXI DataMover during an AXI transfer (read or write). This error condition causes the AXI CDMA to gracefully halt. The CDMA Status register bit-1 is set to '1' when the CDMA has completed shut down. A reset (soft or hard) must be issued to clear the error condition.	
	0	No CDMA Slave Errors
	1	CDMA Slave Error detected. CDMA Engine halts.
6	DMA Decode Error. This bit indicates that an AXI decode error has been received by the AXI DataMover. This error occurs if the DataMover issues an address that does not have a mapping assignment to a slave device. This error condition causes the AXI CDMA to halt gracefully. The CDMA Status register bit-1 is set to '1' when the CDMA has completed shut down. A reset (soft or hard) must be issued to clear the error condition.	
	0	No CDMA Decode Errors
	1	CDMA Decode Error detected. CDMA Engine halts.
7	Reserved	
8	Scatter Gather Internal Error. This bit indicates that an internal error has been encountered by the Scatter Gather Engine. This error condition causes the AXI CDMA to gracefully halt. The CDMA Status register bit-1 is set to 1 when the CDMA has completed shut down. A reset (soft or hard) must be issued to clear the error condition.	
	0	No Scatter Gather Internal Errors
	1	Scatter Gather Internal Error. CDMA Engine halts.

Bit(s)	FUNCTION	
9	Scatter Gather Slave Error. This bit indicates that an AXI slave error response has been received by the Scatter Gather Engine during an AXI transfer (transfer descriptor read or write). This error condition causes the AXI CDMA to gracefully halt. The CDMA Status register bit-1 is set to 1 when the CDMA has completed shut down. A reset (soft or hard) must be issued to clear the error condition.	
	0	No Scatter Gather Slave Errors
	1	Scatter Gather Slave Error. CDMA Engine halts.
10	Scatter Gather Decode Error. This bit indicates that an AXI decode error has been received by the Scatter Gather Engine during an AXI transfer (transfer descriptor read or write). This error occurs if the Scatter Gather Engine issues an address that does not have a mapping assignment to a slave device. This error condition causes the AXI CDMA to gracefully halt. The CDMA Status register bit-1 is set to 1 when the CDMA has completed shut down. A reset (soft or hard) must be issued to clear the error condition.	
	0	No Scatter Gather Decode Errors
	1	Scatter Gather Decode Error. CDMA Engine halts.
11	Reserved	
12	Interrupt on Complete. When set to 1, this bit indicates an interrupt event has been generated on completion of a DMA transfer (either a Simple or Scatter Gather). If the Interrupt on Complete (bit-12) of the CDMA Control register = '1', an interrupt is generated from the AXI CDMA. A CPU write of 1 clears this bit to 0. **Note:** When operating in Scatter Gather mode, the criteria specified by the interrupt threshold must also be met.	
	0	No IOC Interrupt
	1	IOC Interrupt active
13	Interrupt on Delay. When set to 1, this bit indicates an interrupt event has been generated on a delay timer time out. If the Interrupt on Delay Timer bit-13 of the CDMA Control register = '1', an interrupt is generated from the AXI CDMA. A CPU write of 1 clears this bit to 0.	
	0	No Delay Interrupt
	1	Delay Interrupt Active
14	Interrupt on Error. When set to 1, this bit indicates an interrupt event has been generated due to an error condition. If the Interrupt on Error bit-14 of the CDMA Control register = '1', an interrupt is generated from the AXI CDMA. A CPU write of 1 clears this bit to 0.	
	0	No Error Interrupt
	1	Error Interrupt Active
15	Reserved	

Bit(s)	FUNCTION
23-16	Interrupt Threshold Status. This field reflects the current interrupt threshold value in the Scatter Gather Engine.
31-24	Interrupt Delay Time Status. This field reflects the current interrupt delay timer value in the Scatter Gather Engine.

CDMA Current Descriptor Pointer Register

Table 18 CDMA Current Descriptor Pointer Register (Read/Write) - (BAR0 + 0x00000008)

Bit(s)	FUNCTION
5-0	Writing to these bits has no effect and they are always read as zeros.
31-6	Current Descriptor Pointer. This register field is written by the software application (in Scatter Gather Mode) to set the starting address of the first transfer descriptor to execute for a Scatter Gather operation. The address written corresponds to a 32-bit system address with the least significant 6 bits truncated. This register field must contain a valid descriptor address prior to the software application writing the CDMA Tail Descriptor Pointer register value. Failure to do so results in an undefined operation by the CDMA. On error detection, the Current Descriptor Pointer register is updated to reflect the descriptor associated with the detected error. **Note:** The register should only be written by the Software application when the AXI CDMA is Idle.

CDMA Tail Descriptor Pointer Register

This register provides Tail Descriptor Pointer for the AXI CDMA Scatter Gather Descriptor Management.

Table 19 CDMA Tail Descriptor Pointer Register (Read/Write) - (BAR0 + 0x00000010)

Bit(s)	FUNCTION
5-0	Writing to these bits has no effect and they are always read as zeros.
31-6	Tail Descriptor Pointer. This register field is written by the software application (in Scatter Gather Mode) to set the current pause pointer for descriptor chain execution. The AXI CDMA Scatter Gather Engine pauses descriptor fetching after completing operations on the descriptor whose current descriptor pointer matches the tail descriptor pointer. When the AXI CDMA is in Scatter Gather Mode, a write by the software application to this register causes the AXI CDMA Scatter Gather Engine to start fetching descriptors starting from the Current Descriptor Pointer register value. If the Scatter Gather engine is paused at a tail pointer pause point, the Scatter Gather engine restarts descriptor execution at the next sequential transfer descriptor. If the AXI CDMA is not idle, writing to this register has no effect except to reposition the Scatter Gather pause point. **Note:** The software application must not move the tail pointer to a location that has not been updated with valid transfer descriptors. The software application must process and reallocate all completed descriptors, clear the completed bits and then move the tail pointer. The software application must move the pointer to the last descriptor address it has updated.

CDMA Source Address Register

This register provides the source address for simple DMA transfers by AXI CDMA.

If a location in system memory is the source address, it must be set with the AXI aperture base address 0x01000000 + the least significant 24-bits of the system memory address.

In addition, the physical address of the location in system memory must be set in the Address Translation Register which is described in the PCIe AXI-Bridge Control section.

Table 20 CDMA Source Address Register (Read/Write) - (BAR0 + 0x0018)

Bit(s)	FUNCTION
31-0	Source Address Register. This register is used by Simple DMA operations as the starting read address for DMA data transfers. The address value written can be at any byte offset. **Note:** The software application should only write to this register when the AXI CDMA is Idle.

CDMA Destination Address Register

This register provides the destination address for simple DMA transfers by AXI CDMA.

If a location in system memory is the destination address, it must be set with the AXI aperture base address 0x01000000 + the least significant 24-bits of the system memory address.

In addition, the physical address of the location in system memory must be set in the Address Translation Register which is described in the PCIe AXI-Bridge Control section.

Table 21 CDMA Destination Address Register (Read/Write) - (BAR0 + 0x0020)

Bit(s)	FUNCTION
31-0	Destination Address Register. This register is used by Simple DMA operations as the starting write address for DMA data transfers. **Note:** The software application should only write to this register when the AXI CDMA is Idle.

CDMA Bytes to Transfer Register

This register provides the value for the bytes to transfer for Simple DMA transfers by the AXI CDMA.

Table 22 CDMA Bytes to Transfer Register (Read/Write) - (BAR0 + 0x0028)

Bit(s)	FUNCTION
22-0	Bytes to Transfer. This register field is used for Simple DMA transfers and indicates the desired number of bytes to DMA from the Source Address to the Destination Address. A maximum of 8,388,606 bytes of data can be specified by this field for the associated transfer. **Writing to this register also initiates the Simple DMA transfer**. Note: A value of zero (0) is not allowed and causes a DMA internal error to be set by AXI CDMA. The software application should only write to this register when the AXI CDMA is Idle.
31-23	Writing to these bits has no effect, and they are always read as zeros.

Simple CDMA Programming Example

1. Verify the CDMA is idle. Read CDMA Status register bit-1 as logic '1'.

2. Program the CDMA Control register bit-12 to the desired state for interrupt generation on transfer completion.

3. Write the desired transfer source address to the Source Address register at 0x0018. The transfer data at the source address must be valid and ready for transfer. If we were to select the Block RAM memory as the source and wanted to start a move of data from the beginning of DDR, we would write 0x0001_0000 to the Source Address register at 0x0018.

4. Write the desired transfer destination address to the Destination Address register at 0x0020. If the destination is the system memory, then the following is required.

    a. Given physical address of buffer of 0x0000333012345678

    b. AXIBAR2PCIEBAR_0U <offset 000F0208> = 0x00003330

    c. AXIBAR2PCIEBAR_0L <offset 000F020C> = 0x12345678

    d. The least significant 24 bits of this address 0x12345678 must be removed and added to the AXI BAR0 Aperture Base address. The new AXI address is 0x01000000 + 0x00345678 = 0x01345678. Write 0x01345678 to 0x0020.

5. Write the number of bytes to transfer to the CDMA Bytes to Transfer register 0x0028. Writing this register also starts the transfer.

6. Either poll the CDMA Status register bit-1 for logic '1' or wait for the CDMA to generate an interrupt if enabled.

7. Clear the interrupt if generated by writing a '1' to bit-12 of the CDMA Status register.

8. Ready for another transfer. Go back to step 1.

AXI-BAR0 Aperture Base Address

The AXI BAR0 aperture base address of 0x01000000 is set as the base address on the AXI bus used to reach system host memory for CDMA transfers.

The address 0x01000000 is the AXI BAR0 Aperture Base address. In Vivado IP Integrator the address map will show that a 16Meg address space for the AXI BAR0 Aperture Base Address is reserved.

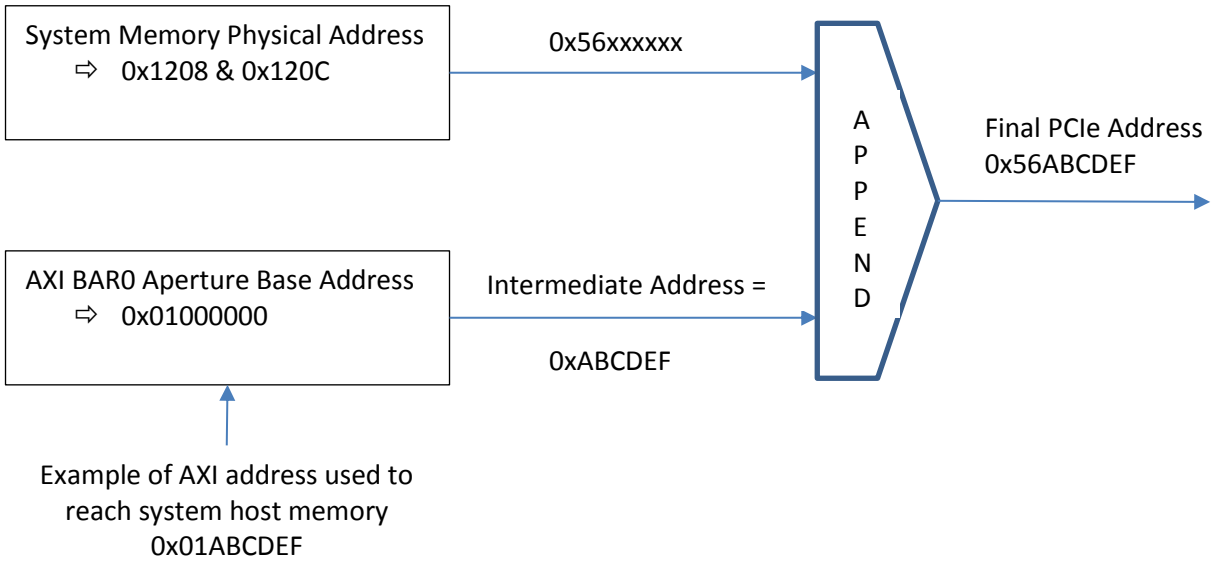
Table 23 AXI BAR0 Aperture Base Address

0x01000000→0x01FFFFFF	16M	Window into PCIe Interface AXI BAR0 Aperture Base Address

The following is an example of how the AXI BAR0 aperture base address is used.

For example, if the system buffer physical address 0x56ABCDEF were given, then the AXI Base Address Translation Configuration registers at BAR0 + 0x1208 and 0x120C must be set to 0x0 and 0x56 ABCDEF respectively.

The least significant 24 bits of this address 0x56ABCDEF must be removed and added to the AXI BAR0 Aperture Base address. The new AXI address is 0x01000000 + 0x00ABCDEF = 0x01ABCDEF. These values are then appended by the PCIe AXI bridge to give the final PCIe address of the system memory location.



PCIe AXI-Bridge Control

The PCIe AXI Bridge is an interface between the AXI bus and the PCIe system. This bridge provides the address translation between the AXI4 memory-mapped embedded system and the PCIe system. The AXI Bridge for PCIe translates the AXI memory read or writes to PCIe Transaction Layer Packets (TLP) packets and translates PCIe memory read and write request TLP packets to AXI interface commands.

Table 24 PCIe AXI Bridge Control Registers

BAR0 Base Addr+	Bit(s)	Description
0x0000_1000→ 0x0000_1140	31:0	See Xilinx pg055 Memory Map
0x0000_1144	31:0	PHY Status/Control Register
0x0000_1148→ 0x0000_1204	31:0	See Xilinx pg055 Memory Map
0x0000_1208	31:0	Address Translation Register Upper AXIBAR2PCIEBAR_0U
0x0000_120C	31:0	Address Translation Register Lower AXIBAR2PCIEBAR_0L
0x0000_1210→ 0x0000_1FFF	31:0	See Xilinx pg055 Memory Map

PHY Status/Control Register

This register provides the status of the current PHY state, as well as control of speed and rate switching for Gen2-capable cores.

Table 25 PHY Status/Control Register (Read/Write) - (BAR0 + 0x1144)

Bit(s)	FUNCTION	
0	Reports the current link rate.	
	0	2.5 GT/s
	1	5.0 GT/s
2-1	Reports the current link width.	
	00	x1
	01	x2
	10	x4
	11	x8
8-3	Reports the current Link Training and Status State Machine state. Encoding is specific to the underlying Integrated Block.	
	x	
	x	
10-9	Reports the current lane reversal mode.	
	00	No reversal
	01	Lanes 1:0 reversed

Bit(s)	FUNCTION	
	10	Lanes 3:0 reversed
	11	Lanes 7:0 reversed
11	Reports the current PHY Link-up state.	
	0	Link down
	1	Link up
15-12	Reserved	
31-16	See Xilinx pg055 PHY Status/Control Register	

AXI Base Address Translation Configuration Register

The address space for PCIe is different than the AXI address space. To access one address space from another address space requires an address translation process.

These registers are needed for DMA transfers that move data to the system memory buffer. The location of the system memory buffer is loaded into these registers.

AXI Base Address Translation Configuration register at BAR0 + 0x1208 must be written with the most significant 32 bits of the address in system memory to which the DMA transfer is to read or write. An example of the c code used to set this register with the physical address is shown below.

AXI Base Address Translation Configuration register at BAR0 + 0x120C must be written with the least significant 32 bits of the address in system memory to which the DMA transfer is to read or write. An example of the c code used to set this register with the physical address is shown below.

Example C code:

```
#define AXI2PCIeBAR_0U        (*(DWORD *)(u64BaseAddress + 0x1208))

#define AXI2PCIeBAR_0L        (*(DWORD *)(u64BaseAddress + 0x120C))

iStatus = PCIe7A_DmaGetBuffPhysAddress(iHandle, &u64PhyAddr);

    AXI2PCIeBAR_0U = (DWORD)(u64PhyAddr >> 32);

    AXI2PCIeBAR_0L = (DWORD)(u64PhyAddr & 0xffffffff);
```

This sets the system memory physical address which will be appended with the values written into either the DMA source or destination registers at 0x1018 or 0x1020 respectively. See the example in the CDMA section for additional details.

Flash Memory

The Serial flash memory provides 4M bytes of non-volatile memory for storing the FPGA configuration bitstream. The lower portion of the flash address space is used to store the FPGA configuration bitstream. The upper portion is available for general non-volatile / program storage. The serial flash is

accessible through the AXI Quad SPI interface. The flash device is Spansion part number S25FL132K0XMFI040. See data sheet S25FL1-K_00.

Table 26 Flash Memory Map (Read/Write)

Sector Size (kbyte)	Sector Count	Sector Range	Address Range (byte addresses)	Usage
4	1024	SA1023	3FF000 – 3FFFFF	Available for user data / program storage
		⋮	⋮	
		SA536	218000 - 218FFF	
		SA535	217000 - 217FFF	FPGA Configuration Bitstream
		⋮	⋮	
		SA0	000000 - 000FFF	

AXI XADC Analog to Digital Converter (System Monitor)

The XADC Analog to Digital Converter is used to monitor the die temperature and supply voltages of the FGPA. The XADC channel sequencer is configured to continuously sample the temperature, Vccint and Vccaux channels. The results from the A/D conversions can be read at the addresses given in column one of Table 27.

Data bits 15 to 4 of these registers hold the "ADCcode" representing the temperature, Vccint, or Vccaux value. Data bits 3 to 0 are not used.

The 12-bits output from the ADC can be converted to temperature using the following equation.

$$Temperature(°C) = \frac{ADCcode \times 503.975}{4096} - 273.15$$

The 12-bits output from the ADC can be converted to voltage using the following equation.

$$SupplyVoltage(volts) = \frac{ADCcode}{4096} \times 3V$$

Additional information regarding the XADC can be found in the Xilinx XADC product guide PG099 and the user guide UG480

Table 27 System Monitor Register Map – (BAR0 + 0x32xx)

Address	Status Register
0x0000_3200	Temperature
0x0000_3204	V_{CCINT}
0x0000_3208	V_{CCAUX}
0x0000_3280	Maximum Temperature
0x0000_3284	Maximum V_{CCINT}
0x0000_3288	Maximum V_{CCAUX}
0x0000_3290	Minimum Temperature
0x0000_3294	Minimum V_{CCINT}
0x0000_3298	Minimum V_{CCAUX}

The expected values for V_{CCINT}, $V_{CCINT,}$ and Temperature are shown in Table 28.

Table 28 Expected Operating Parameters

Parameter	Min	Max
V_{CCINT} Volts	0.95	1.05
V_{CCAUX}	1.71	1.89
Temperature (recommended)	-40°C	100°C

Field I/O

The field I/O in the example design provided has been configured as two bidirectional ports A and B and two direction control registers A and B. It is an instance of Xilinx's LogiCORE IP AXI GPIO. The Xilinx IP core has been configured as follows:

GPIO1 – width 24, bidirectional
GPIO2 – width 24, bidirectional
interrupts disabled

Table 29 lists the AXI GPIO registers.

Table 29 GPIO Register Map - (BAR0 + 0x4000)

Address	Status Register
0x0000_4000	GPIO_1 "A" Data Register
0x0000_4004	GPIO_1 "A" 3-State Control Register
0x0000_4008	GPIO_2 "B" Data Register
0x0000_400C	GPIO_2 "B" 3-State Control Register

GPIO_1 "A" Data Register

The GPIO channel 1 "A" data register is connected to the single ended field I/O signals A1 thru A24 in AcroPack models APA7-501 and APA7-503. Table 2 shows the pin mapping for the signals.

Input signal levels are determined by reading this register. Output signals are set by writing to this register when the appropriate direction control bit in the GPIO_1 "A" 3-State Control Register is set to output.

Table 30 GPIO 1 "A" Data Register - (BAR0 + 0x4000)

Bit(s)	GPIO_1 "A" Data Register APA7-501, APA7-503
0	Field I/O A1
⋮	
23	Field I/O A24
24 – 31	Reserved

GPIO_1 "A" 3-State Control Register

The GPIO channel 1 "A" 3-State Control Register controls the I/O direction of both the on-chip I/O buffers and the external transceivers. The external transceivers are arranged in banks of eight field I/O signals. One bit in the 3-State Control Register is assigned to each bank. This register is not used in models APA7-502 and APA7-504

Table 31 GPIO_1 "A" 3-State Control Register – (BAR0 + 0x4004)

Bit(s)	GPIO_1 "A" 3-State Control Register APA7-501 -503	
0	Field I/O A1 thru A8 direction control	
	0	output
	1	input
1	Field I/O A9 thru A16 direction control	
	0	output
	1	input
2	Field I/O A17 thru A24 direction control	
	0	output
	1	input
3 – 31	Reserved	

GPIO_2 "B" Data Register

The GPIO channel 2 "B" data register is connected to the single ended field I/O signals B25 thru B48 in AcroPack model APA7-501. For models APA7-502 and APA7-504, the GPIO channel 2 "B" data register is connected to the differential field I/O signals B1 thru B24. For model APA7-503, the GPIO channel 2 "B" data register is connected to the differential field I/O signals B1 thru B12. Table 2 shows the pin mapping for these signals.

Input signal levels are determined by reading this register. Output signals are set by writing to this register when the appropriate direction control bit is set to output.

Table 32 GPIO_2 "B" Data Register APA7-501 – (BAR0 + 0x4008)

Bit(s)	GPIO_2 "B" Data Register APA7-501
0	Field I/O B25
⋮	
23	Field I/O B48
24 – 31	Reserved

Table 33 GPIO_2 "B" Data Register APA7-502, APA7-504 - (BAR0 + 0x4008)

Bit(s)	GPIO_2 "B" Data Register APA7-502, APA7-504
0	Field I/O B1
⋮	
23	Field I/O B24
24 – 21	Reserved

Table 34 GPIO_2 "B" Data Register APA7-503 - (BAR0 + 0x4008)

Bit(s)	GPIO_2 "B" Data Register APA7-503
0	Field I/O B1
⋮	⋮
11	Field I/O B12
12 – 31	Reserved

GPIO_2 "B" 3-State Control Register

The GPIO_2 "B" 3-State Control Register controls the I/O direction of both the on-chip I/O buffers and the external transceivers. A separate direction control signal is provided for each bank of eight I/O in model APA7-501. For Models APA7-501, APA7-503, and APA7-504, each bit in the 3-State Control Register sets the direction of a single I/O line.

Table 35 GPIO_2 "B" 3-State Control Register – (BAR0 + 0x400C)

Bit(s)	GPIO_2 "B" 3-State Control Register – APA7-501	
0	Field I/O B25 thru B32 direction control	
	0	output
	1	input
1	Field I/O B33 thru B40 direction control	

Bit(s)	GPIO_2 "B" 3-State Control Register – APA7-501	
	0	output
	1	input
2	Field I/O B41 thru B48 direction control	
	0	output
	1	input
3 – 31	Reserved	

Table 36 GPIO_2 "B" 3-State Control Register - (BAR0 + 0x400C)

Bit(s)	GPIO_2 "B" 3-State Control Register – APA7-501/4	
0	Field I/O B1 direction control	
	0	output
	1	input
⋮		
23	Field I/O B24 direction control	
	0	output
	1	input
24 – 31	Reserved	

Table 37 GPIO_2 "B" 3-State Control Register - (BAR0 + 0x400C)

Bit(s)	GPIO_2 "B" 3-State Control Register – APA7-503	
0	Field I/O B1 direction control	
	0	output
	1	input
⋮		
11	Field I/O B12 direction control	
	0	output
	1	input
12 – 31	Reserved	

AXI Quad SPI

The serial flash memory is accessed through the AXI Quad SPI interface. For a thorough understanding of the interface refer to the Xilinx product guide pg153 and the commands section of the flash memory data sheet S25FL1-K_00. Following is a description of the AXI Quad SPI registers. The AXI Quad SPI interface has been configured as standard mode (Quad message not supported), master mode enabled, no FIFOs, and the interrupt output is not connected. The clock frequency ratio is set at 2 resulting in a 31.25 MHz SPI clock. With the interrupt output unconnected, the Quad SPI IP Interrupt Status Register is still useful for reporting error conditions.

Address	Register
0x0000_5040	Quad SPI Software Reset Register
0x0000_5060	Quad SPI Control Register
0x0000_5064	Quad SPI Status Register
0x0000_5068	Quad SPI Data Transmit Register
0x0000_506C	Quad SPI Data Receive Register
0x0000_5070	Quad SPI Slave Select Register
0x0000 5020	Quad SPI IP Interrupt Status Register

Quad SPI Software Reset Register

The Software Reset Register (SRR) permits resetting the core independently of other cores in the system. Writing 0x0000_000a to the SRR resets the core register for four AXI clock cycles. Any other write access generates undefined results and results in an error. The bit assignment in the software reset register is shown in and described in Table 38. Any attempt to read this register returns undefined data.

Table 38 Quad SPI Software Reset Register (BAR0 + 0x5040)

Bit(s)	Software Reset Register
0 - 31	The only allowed operation on this register is a write of 0x0000000a, which resets the AXI Quad SPI core

Quad SPI Control Register

The SPI Control Register (SPICR) allows programmer control over various aspects of the AXI Quad SPI core. The bit assignment in the SPICR is described in Table 39.

Table 39 Quad SPI Control Register (BAR0 + 0x5060)

Bit(s)	Quad SPI Control Register	
0	Local loopback mode:	
	0	Normal Operation
	1	Loopback mode. The transmitter output is internally connected to the receiver input. The receiver and transmitter operate normally, except that received data (from remote slave) is ignored.
1	SPI system enable: Setting this bit to 1 enables the SPI devices as noted here. When set to:	
	0	SPI system disabled. Both master and slave outputs are in 3-state and slave inputs are ignored.
	1	SPI system enabled. Master outputs active (for example, IO0 (MOSI) and SCK in idle state) and slave outputs become active if SS becomes asserted. The master starts transferring when transmit data is available.
2	Master (SPI master mode): Setting this bit configures the SPI device as a master or a slave. When set to:	
	0	Slave configuration
	1	Master configuration

Bit(s)	Quad SPI Control Register	
3	Clock Polarity: Setting this bit defines the clock polarity. When set to:	
	0	Active-High clock; SCK idles low.
	1	Active-Low clock; SCK idles high.
4	Clock phase: Setting this bit selects one of two fundamentally different transfer formats.	
	0	Data valid on first SCK edge after SS asserted
	1	Data valid on second SCK edge after SS asserted
5	TX Register reset: When written with '1', this bit forces a reset of the transmit FIFO to the empty condition. This bit will read as '0'.	
	0	Transmit FIFO normal operation
	1	Reset transmit FIFO pointer
6	RX FIFO reset: When written with '1', this bit forces a reset of the receive FIFO to the empty condition. This bit will read as '0'.	
	0	Receive FIFO normal operation
	1	Reset receive FIFO pointer
7	Manual slave select assertion enable: This bit forces the data in the slave select register to be asserted on the slave select output anytime the device is configured as a master and the device is enabled (SPE asserted). When set to:	
	0	Slave select output asserted by master core logic.
	1	Slave select output follows data in slave select register.
8	Master transaction inhibit: This bit inhibits master transactions. This bit has no effect on slave operation. When set to:	
	0	Master transactions enabled.
	1	1 = Master transactions disabled. Note: This bit immediately inhibits the transaction. Setting this bit while transfer is in progress would result in unpredictable outcome.
9	LSB first: This bit selects LSB first data transfer format. The default transfer format is MSB first. When set to:	
	0	MSB first transfer format.
	1	LSB first transfer format. Note: In Dual/Quad SPI mode, only the MSB first mode of the core is allowed.
10 - 31	Reserved	

Quad SPI Status Register

The Quad SPI Status Register (SPISR) is a read-only register that provides the status of some aspects of the AXI Quad SPI core to the programmer. The bit

assignment in the SPISR is described in Table 40. Writing to the SPISR does not modify the register contents.

Table 40 Quad SPI Status Register (BAR0 + 0x5064)

Bit(s)	Quad SPI Status Register	
0	Receive Register Empty: This bit is set High when the receive register is empty.	
	0	Rx register Is not empty
	1	Rx register is empty

Bit(s)	Quad SPI Status Register	
1	Receive Register Full: This bit is set High when the Receive Register is full.	
	0	Rx register not full
	1	Rx register full
2	Transmit Register empty: This bit is set to High when the transmit register is empty. This bit goes High as soon as the TX register becomes empty.	
	0	Tx FIFO not empty
	1	Tx FIFO empty
3	Transmit Register Full: This bit is set High when the transmit register is full.	
	0	Tx register not full
	1	Tx register full
4	Mode-fault error flag: This flag is set if the SS signal goes active while the SPI device is configured as a master. MODF is automatically cleared by reading the SPISR. A Low-to-High MODF transition generates a single-cycle strobe interrupt.	
	0	No error
	1	Error condition detected
5	Slave_Mode_Select flag: This flag is asserted when the core is configured in slave mode. Slave_Mode_Select is activated as soon as the master SPI core asserts the chip select pin for the core.	
	0	Asserted when core configured in slave mode and selected by external SPI master
	1	Default in standard mode
6	CPOL_CPHA_Error flag: When set indicates an invalid combination of CPOL and CPHA has been selected.	
	0	Default.
	1	The CPOL and CPHA are set to 01 or 10.
7	Slave mode error flag.	
	0	Master mode is set in the control register (SPICR).
	1	This bit is set when the core is configured with dual or quad SPI mode and the master is set to 0 in the control register (SPICR).
8	MSB error flag	
	0	Default
	1	LSB first bit set in SPICR (not valid for quad mode)
9	Loopback error flag	
	0	Default. The loopback bit in the control register is at default state.
	1	When the SPI command, address, and data bits are set to be transferred in other than standard SPI protocol mode and this bit is set in control register (SPICR).
10	Command error flag	
	0	Default

Bit(s)	Quad SPI Status Register	
	1	The first entry in SPI DTR FIFO is incompatible with the command list for the attached flash device
11 - 31	Reserved	

Quad SPI Data Transmit Register

The Quad SPI Data Transmit Register (SPI DTR) is written with the data to be transmitted on the SPI bus. After the SPE bit is set to 1 in master mode or spisel is active in the slave mode, the data is transferred from the SPI DTR to the shift register.

If a transfer is in progress, the data in the SPI DTR is loaded into the shift register as soon as the data in the shift register is transferred to the SPI DRR and a new transfer starts. The data is held in the SPI DTR until replaced by a subsequent write. The SPI DTR is described in Table 41.

This register cannot be read and can only be written when it is known that space for the data is available. If an attempt to write is made on a full register, the AXI write transaction completes with an error condition. Reading the SPI DTR is not allowed and the read transaction results in undefined data.

Table 41 Quad SPI Data Transmit Register (BAR0 + 0x5068)

Bit(s)	Quad SPI Data Transfer Register
0 - 7	SPI transmit data
8 - 31	Reserved

Quad SPI Data Receive Register

The Quad SPI Data Receive Register (SPI DRR) is used to read data that is received from the SPI bus. This is a double-buffered register. The received data is placed in this register after each complete transfer.

If the SPI DRR was not read and is full, the most recently transferred data is lost and a receive overrun interrupt occurs.

If an attempt is made to read an empty receive register, it gives out an error in the Status register. Writes to the SPI DRR do not modify the register contents and return with a successful OK response.

The power-on reset values for the SPI DRR are unknown. When known data has been written into the SPI DRR during core transactions, the data in this register can be considered for reading. The SPI DRR is described Table 42.

IMPORTANT: Based on the command that is issued to the SPI device, a certain unwanted number of bytes are written to the Receive register. These bytes have to be discarded.

Table 42 Quad SPI Data Receive Register (BAR0 + 0x506C)

Bit(s)	Quad SPI Data Receive Register
0 - 7	SPI receive data

Bit(s)	Quad SPI Data Receive Register
8 - 31	Reserved

Quad SPI Slave Select Register

The SPI Slave Select Register (SPISSR) contains an active-low, one-hot encoded slave select vector SS of length one. The bit assignment in the SPISSR is described in Table 43.

Table 43 Quad SPI Slave Select Register (BAR0 + 0x5070)

Bit(s)	Quad SPI Slave Select Register
0	Set to zero to select flash device
1 - 31	Reserved

Quad SPI IP Interrupt Status Register

The IP Interrupt Status Register (IPISR) collects all of the events. Bit assignments are described in Table 44. The interrupt register is a read/toggle-on-write register. Writing a 1 to a bit position within the register causes the corresponding bit to toggle. All register bits are cleared on reset.

Table 44 IP Interrupt Status Register (BAR0 + 0x5020)

Bit(s)	IP Interrupt Status Register
0	Mode-fault error. This interrupt is generated if the SS signal goes active while the SPI device is configured as a master. This bit is set immediately on SS going active.
1	Slave mode-fault error. This interrupt is generated if the SS signal goes active while the SPI device is configured as a slave, but is not enabled. This bit is set immediately on SS going active and continually set if SS is active and the device is not enabled.
2	Data transmit register empty. It is set when the last byte of data has been transferred out to the external flash memory. In master mode if this bit is set to 1, no more SPI transfers are permitted.
3	Data transmit register underrun. This bit is set at the end of a SPI element transfer by a one-clock period strobe to the interrupt register when data is requested from an empty transmit register by the SPI core logic to perform a SPI transfer. This can occur only when the SPI device is configured as a slave in standard SPI configuration and is enabled by the SPE bit as set. All zeros are loaded in the shift register and transmitted by the slave in an under-run condition.
4	Data receive register full. This bit is set at the end of the SPI element transfer, when the receive register has been filled by a one-clock period strobe to the interrupt register.

Bit(s)	IP Interrupt Status Register
5	Data receive register overrun. This bit is set by a one-clock period strobe to the interrupt register when an attempt to write data to a full receive register is made by the SPI core logic to complete a SPI transfer. This can occur when the SPI device is in either master or slave mode (in standard SPI mode) or if the IP is configured in SPI master mode (dual or quad SPI mode).
6	Reserved
7	Slave select mode. The assertion of this bit is applicable only when the core is configured in slave mode in standard SPI configuration. This bit is set when the other SPI master core selects the core by asserting the slave select line. This bit is set by a one-clock period strobe to the interrupt register. Note: This bit is applicable only in standard SPI slave mode.
8 – 31	Reserved

Serial Flash Write/Read Example

This section describes a suggested sequence for accessing the serial Flash. This example reads the Manufacturer's ID and device ID from the flash.

One time, at startup, write the Quad SPI Software Reset Register and the Quad SPI Control registers.

Write 0x0000000A to address 0x00005040 to reset the interface

Write 0x000000E6 to address 0x00005060 to configure the Quad SPI interface as a master, set the Clock Phase and Polarity, and select manual slave select mode.

At the start of each Flash command
Write 0x00000000 to address 0x00005070 to select the slave device.

Write 0x00000090 to address 0x00005068 to write the Flash command to initiate the read ID register sequence to the transmit data register.

Read from 0x00005064 until bit 1, receive register full is set.

Read from 0x0000506C to empty the receive data register.

Write 0x00000000 to address 0x00005068 to write command byte 2 to the transmit data register.

Read from 0x00005064 until bit 1, receive register full is set.

Read from 0x0000506C to empty the receive data register.

Write 0x00000000 to address 0x00005068 to write command byte 3 to the transmit data register.

Read from 0x00005064 until bit 1, receive register full is set.

Read from 0x0000506C to empty the receive data register.

Write 0x00000000 to address 0x00005068 to write command byte 4 to the transmit data register.

Read from 0x00005064 until bit 1, receive register full is set.

Read from 0x0000506C to empty the receive data register. The value should be 0xFF.

Write 0x00000000 to address 0x00005068 to write command byte 5 to the transmit data register.

Read from 0x00005064 until bit 1, receive register full is set.

Read from 0x0000506C to empty the receive data register. The value should be the manufacturer's ID 0x01.

Write 0x00000000 to address 0x00005068 to write command byte 6 to the transmit data register.

Read from 0x00005064 until bit 1, receive register full is set.

Read from 0x0000506C to empty the receive data register. The value should be the device ID 0x15.

At the end of each Flash command
Write 0x00000001 to address 0x00005070 to de-select the slave device.

Slot ID Register

The Slot ID Register provides the interface to the carrier's Slot ID CPLD. A read of this register triggers a serial transfer from the carrier's Slot ID CPLD to the Slot ID register. The unique ID enables system software to distinguish this particular AcroPack module from other similar modules.

Table 45 Slot ID Register (BAR0 +0x6000)

Bit(s)	Slot ID Register	
0 - 2	Carrier site	
	"000"	site "A"
	"001"	site "B"
	⋮	⋮
	"111"	site "H"
3 - 7	Carrier identification. These bits are set by the backplane global address (VPX carrier).	
8 – 31	Reserved	

5. THEORY OF OPERATION

This section contains information regarding the design of the board. A description of the basic functionality of the circuitry used on the board is also

provided. Refer to Figure 1 APA7-50X Block Diagram as you review this material.

PCI INTERFACE LOGIC

The PCIe bus interface logic implemented in the Acromag example design provides a one lane PCIe Gen 1 interface to the carrier/CPU. This interface provides access to the example design board functions.

The PCIe bus endpoint interface logic is contained within the FPGA. This logic includes support for PCIe commands, including: configuration read/write, and memory read/write. In addition, the PCIe interface supports requester and/or completion accesses. Maximum payload size of up to 1024 bytes is supported.

The logic also implements interrupt requests via message signaled interrupts. Messages are used to assert and de-assert virtual interrupt lines on the link to emulate the Legacy PCI interrupt INTA# signal.

Clock Generation

All clocks are derived from the 100 MHz PCIe REF_CLK signal.

32MB Serial Flash

The serial FLASH memory provides 32 megabytes of non-volatile storage that is used for FPGA configuration and data storage. The FLASH device is Spansion part number 25FL132F. The lower 16 megabytes of memory space are dedicated to storage of the FPGA bitstream. The remaining 16 megabytes are available for MicroBlaze program and data storage.

An AXI Quad SPI v3.2 block provides the interface between the internal AXI bus and the Spansion FLASH device.

JTAG Port

The JTAG port can be used to program the FPGA and access the device for hardware and software debug. The JTAG signals are routed to the mini-PCIe connector. The carrier routes the JTAG signals from the mini-PCIe connector to a JTAG programming header on the carrier.

The JTAG port allows a host computer to download a bitstream to the FPGA or FLASH using the Xilinx Vivado software tool. In addition, the JTAG port allows debug tools such as the ChipScope™ Pro Analyzer tool or a software debugger to access the FPGA.

Encryption Key Storage

The FPGA bitstream which contains sensitive customer IP, can be protected with 256-bit AES encryption and HMAC/SHA-256 authentication to prevent unauthorized copying of the design. The FPGA performs decryption on the fly during configuration using an internally stored 256-bit key. This key can reside in volatile RAM or in nonvolatile eFUSE bits.

System Monitor (XADC)

The System Monitor provides status information for the 7 series device. The system monitor is located in the center of the FPGA die. The System Monitor function is built around dual 12-bit, 1-megasamples per second Analog-to-Digital Converter. The system monitor is used to measure FPGA physical operating parameters such as on-chip power supply voltages and die temperature.

6. FPGA FIRMWARE EXAMPLE DESIGN

Acromag provides an FPGA firmware example design that provides host access to each of the hardware peripherals on the APA7-50X modules. The example design is intended to be a starting point from which customers will develop their customized applications. The example design is implemented using the Xilinx Vivado development environment "Project Mode" design flow.

APA7-50X Models Block Diagram Overview

The hierarchy of the example design is shown in Figure 2 below. Figure 2 is a screen clipping from the "Hierarchy" tab of the "Sources" pane in the Vivado development environment. The example design consists of a combination of VHDL source files and a Xilinx Vivado IP Integrator block diagram. At the highest level of the hierarchy is the AP750X_top VHDL source file, shown in the figure with the least levels of indentation. This top-level module instantiates the following two lower level modules: AP750X.bd and DirectionChangeA and B. "AP750X.bd" is the Vivado IP Integrator block diagram source where the majority of the example design logic is located. "DirectionChangeA and B" controls the timing of the direction control signals of the bi-directional transceivers.

The IP Integrator block diagram labeled "AP750X.bd" is compiled to produce a lower level hierarchy of VHDL source files that are shown beneath "system". The "+" symbol to the left of each of the labeled subsystems indicates that additional source files exist at lower levels in the hierarchy that are not shown.

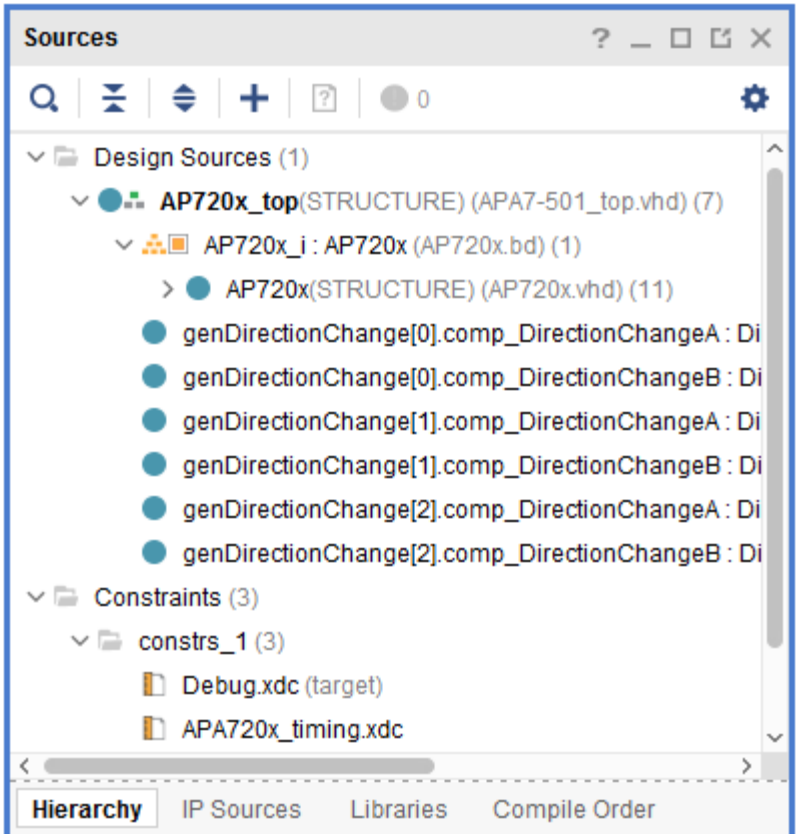


Figure 2 Design Sources Hierarchy

The top level block diagram of the example design is shown in

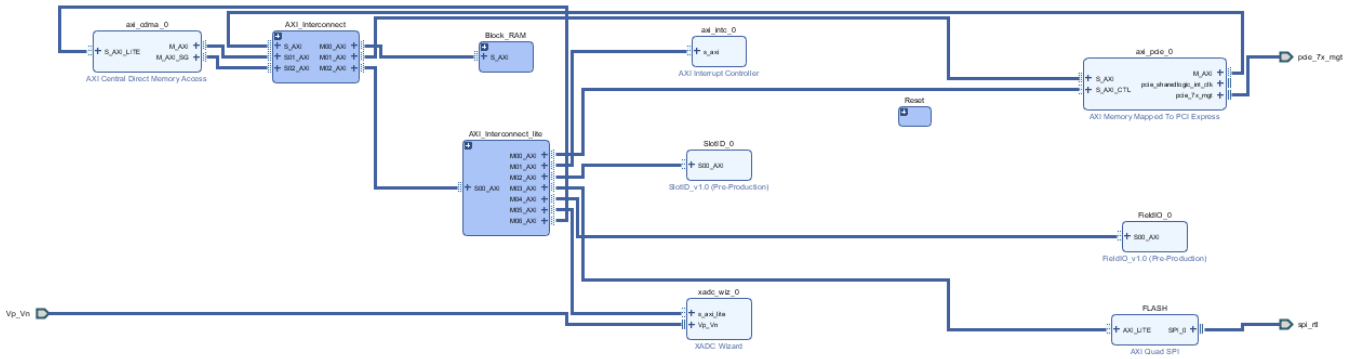


Figure 3 below (select IP Integrator -> Open Block Design). This view has reduced detail, showing only the AXI interfaces between blocks and interfaces to external I/Os. (For more detail on AXI bus see UG761). Blocks with a dark background color include lower level blocks which are expanded and shown in the following paragraphs. The PCIe interface provides the path through which the host processor communicates with the AcroPack module peripherals. This diagram shows the peripherals that are accessible from the host processor:

- Field I/O through the FieldIO_v1.0
- Flash memory through the Quad SPI interface
- Interrupt controller
- Slot ID
- CDMA
- Block RAM
- System Monitor through the XADC peripheral

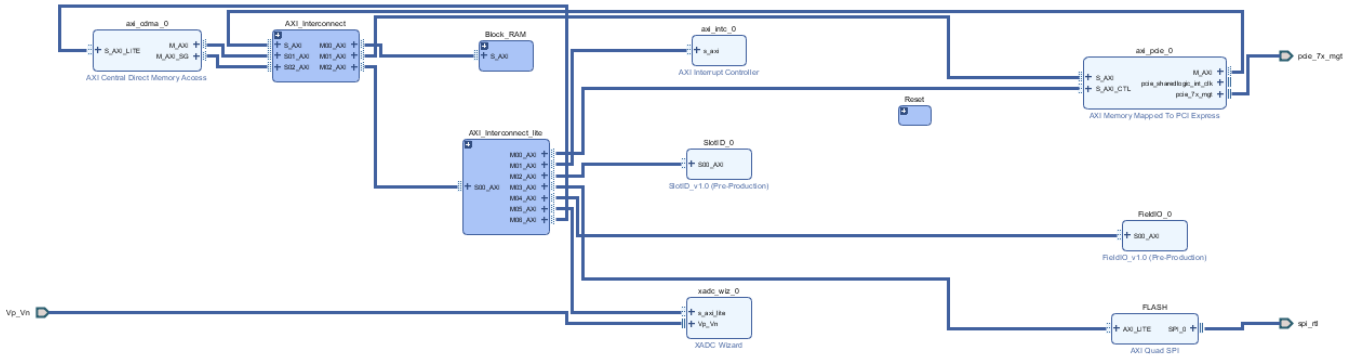


Figure 3 Block Diagram Top Level

Table 46 shows a screen clipping from the Address Editor tab of the Block Design view. The offset address and block size for each peripheral can be set using this editor. This table shows the master interfaces of each peripheral and lists the devices that each can access along with their offset addresses.

Diagram | Address Editor

Cell	Slave Interface	Base Name	Offset Address	Range	High Address
axi_pcie_0					
M_AXI (32 address bits : 4G)					
xadc_wiz_0	s_axi_lite	Reg	0x0000_3000	4K	0x0000_3FFF
SlotID_0	S00_AXI	S00_AXI_reg	0x0000_6000	4K	0x0000_6FFF
FLASH	AXI_LITE	Reg	0x0000_5000	4K	0x0000_5FFF
FieldIO_0	S00_AXI	S00_AXI_reg	0x0000_4000	4K	0x0000_4FFF
Block_RAM/axi_bram_ctrl_0	S_AXI	Mem0	0x0001_0000	32K	0x0001_7FFF
axi_pcie_0	S_AXI	BAR0	0x0100_0000	16M	0x01FF_FFFF
axi_pcie_0	S_AXI_CTL	CTL0	0x0000_1000	4K	0x0000_1FFF
axi_intc_0	s_axi	Reg	0x0000_2000	4K	0x0000_2FFF
axi_cdma_0	S_AXI_LITE	Reg	0x0000_0000	4K	0x0000_0FFF
axi_cdma_0					
Data_SG (32 address bits : 4G)					
Excluded Address Segments (7)					
xadc_wiz_0	s_axi_lite	Reg	0x0000_3000	4K	0x0000_3FFF
SlotID_0	S00_AXI	S00_AXI_reg	0x0000_6000	4K	0x0000_6FFF
FLASH	AXI_LITE	Reg	0x0000_5000	4K	0x0000_5FFF
axi_pcie_0	S_AXI	BAR0	0x0100_0000	16M	0x01FF_FFFF
axi_pcie_0	S_AXI_CTL	CTL0	0x0000_1000	4K	0x0000_1FFF
axi_intc_0	s_axi	Reg	0x0000_2000	4K	0x0000_2FFF
axi_cdma_0	S_AXI_LITE	Reg	0x0000_0000	4K	0x0000_0FFF
Block_RAM/axi_bram_ctrl_0	S_AXI	Mem0	0x0001_0000	32K	0x0001_7FFF

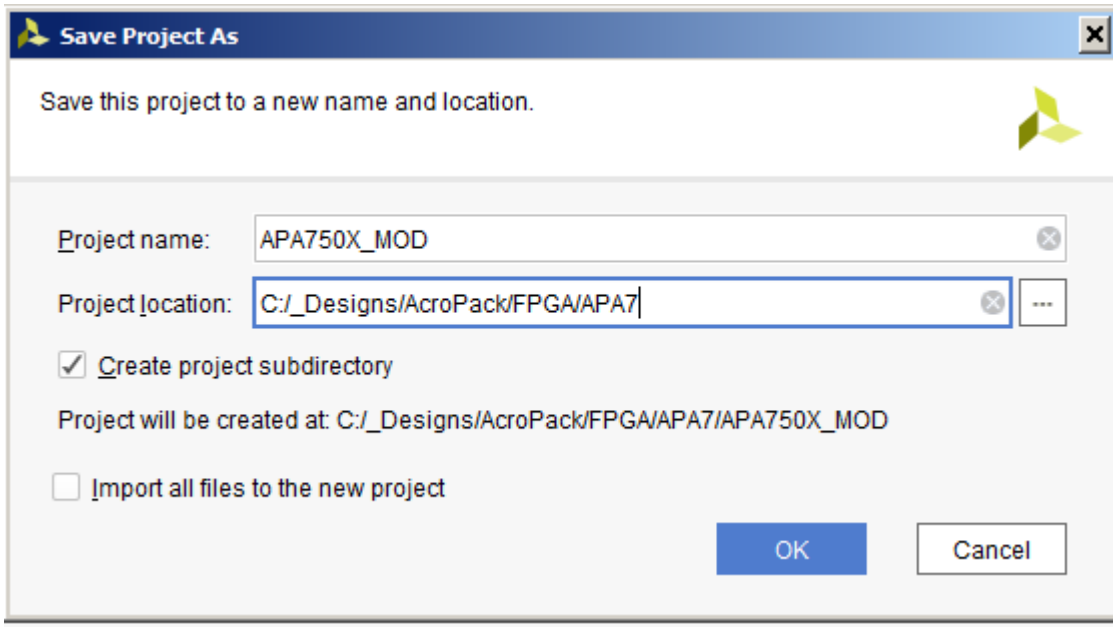
Table 46 APA7-50X Address Map

7. DESIGN MODIFICATION WALKTHROUGH

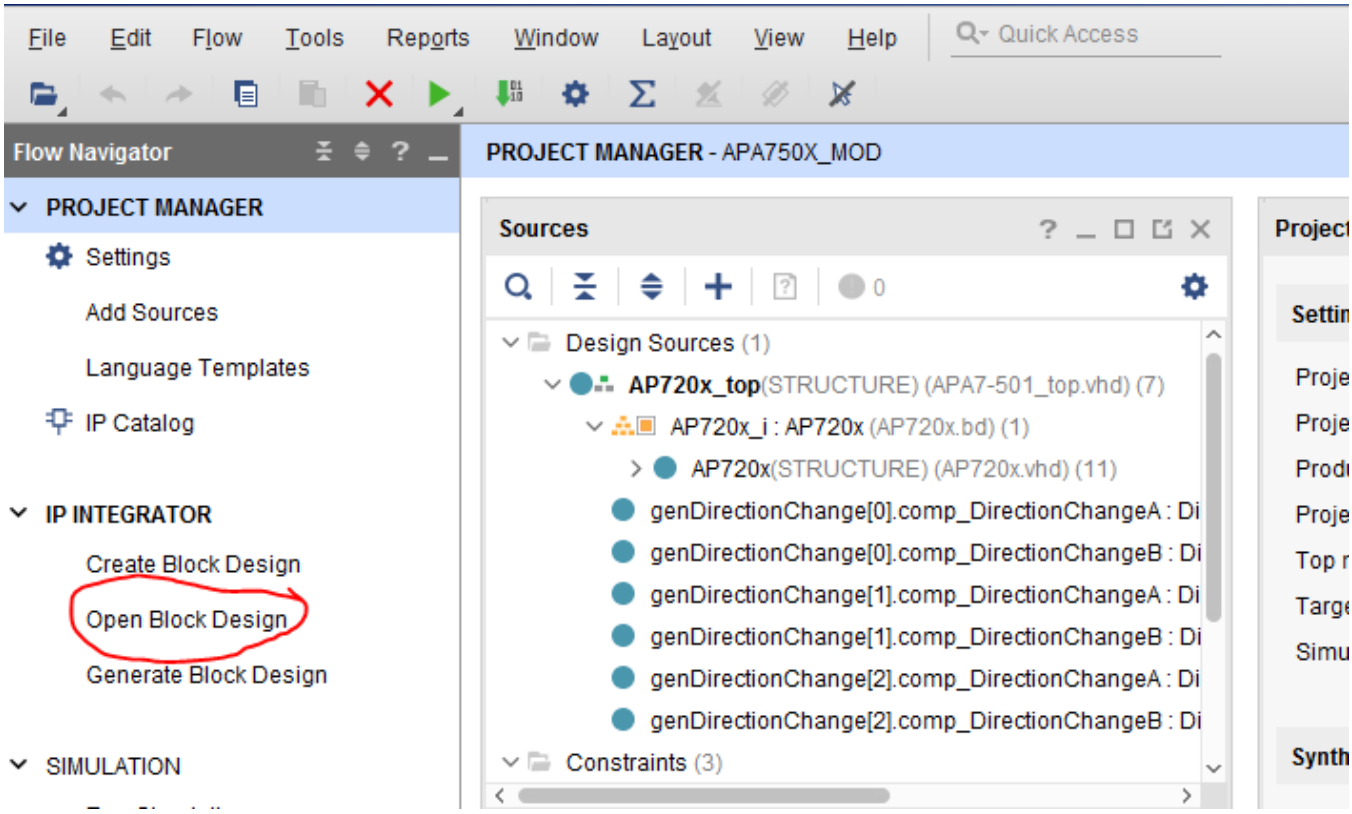
This section details the steps required to make a modification to the example design and compile the project files to produce a new FPGA configuration file. The process to update the configuration flash with the new file will also be covered. In order follow along with this walkthrough, you must have first purchased the APA7-EDK product and copied the Vivado Project Files from the CD to the "C" drive of your PC. You will also need to have Xilinx Vivado 2018.2 installed. In this walk through we will show how the Field I/O IP is generated. This walk through will show how to update the block diagram and the top level VHDL source.

Copy the Project

To begin, copy the example design project to a new work area. Open the example project C:\APA7-501\APA7-501.xpr. Under the File => Project select "Save As" command. Enter "APA750X_MOD" in the Project name field. Verify the "Create project subdirectory" check box is checked. Hit OK.



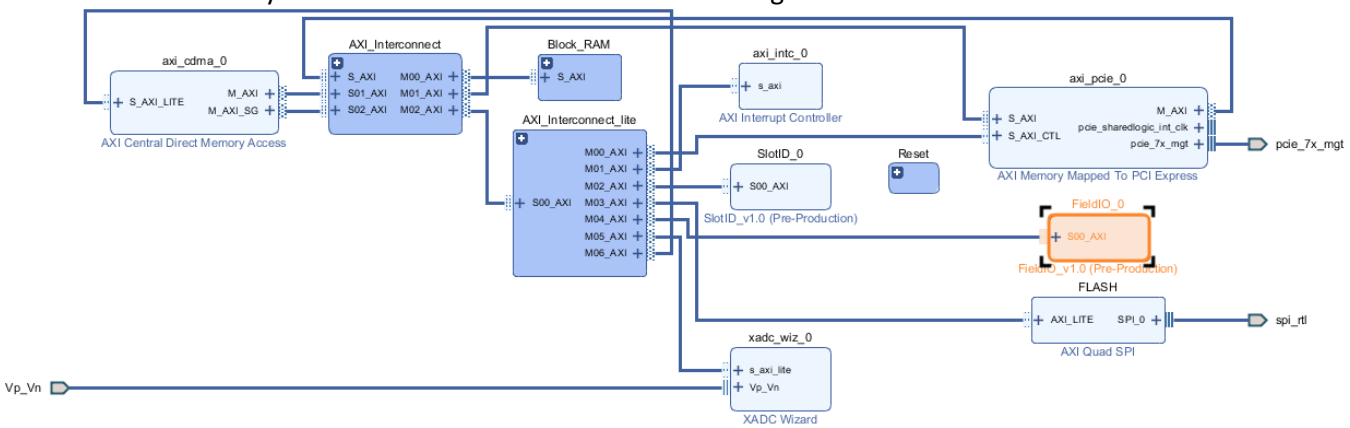
Vivado closes the APA7-501 project. The APA750X_MOD project is now open. Click on "Open Block Design" under IP Integrator in the Flow Navigator pane.



Edit the Block Diagram

Click on the "show interconnections only"  button to reduce the detail in the diagram. Click on the "regenerate layout"  button. The reduced detail view matches the screen clips in this user's manual.

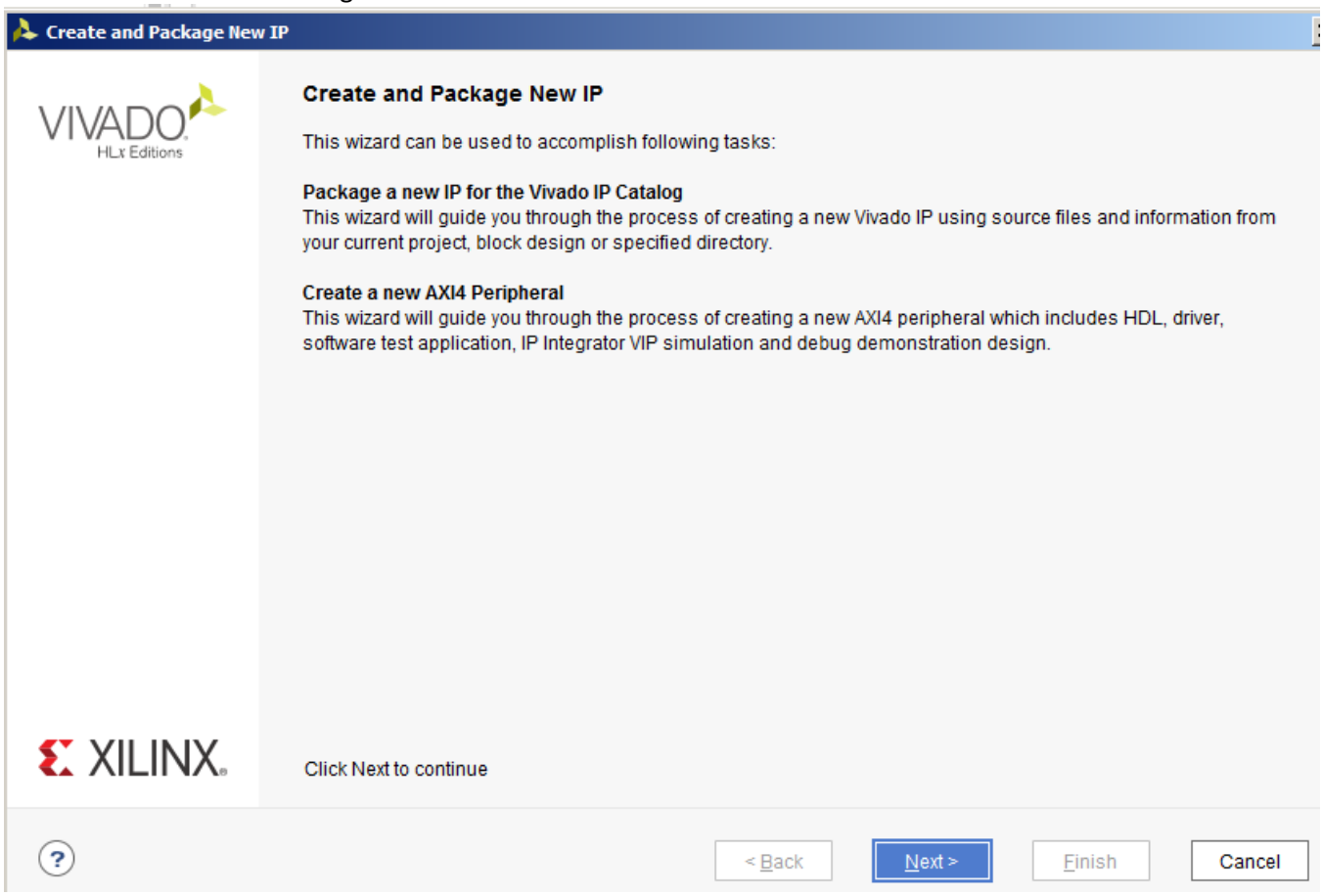
Hold down the Ctrl key and select the "FIELD IO" block in the diagram.



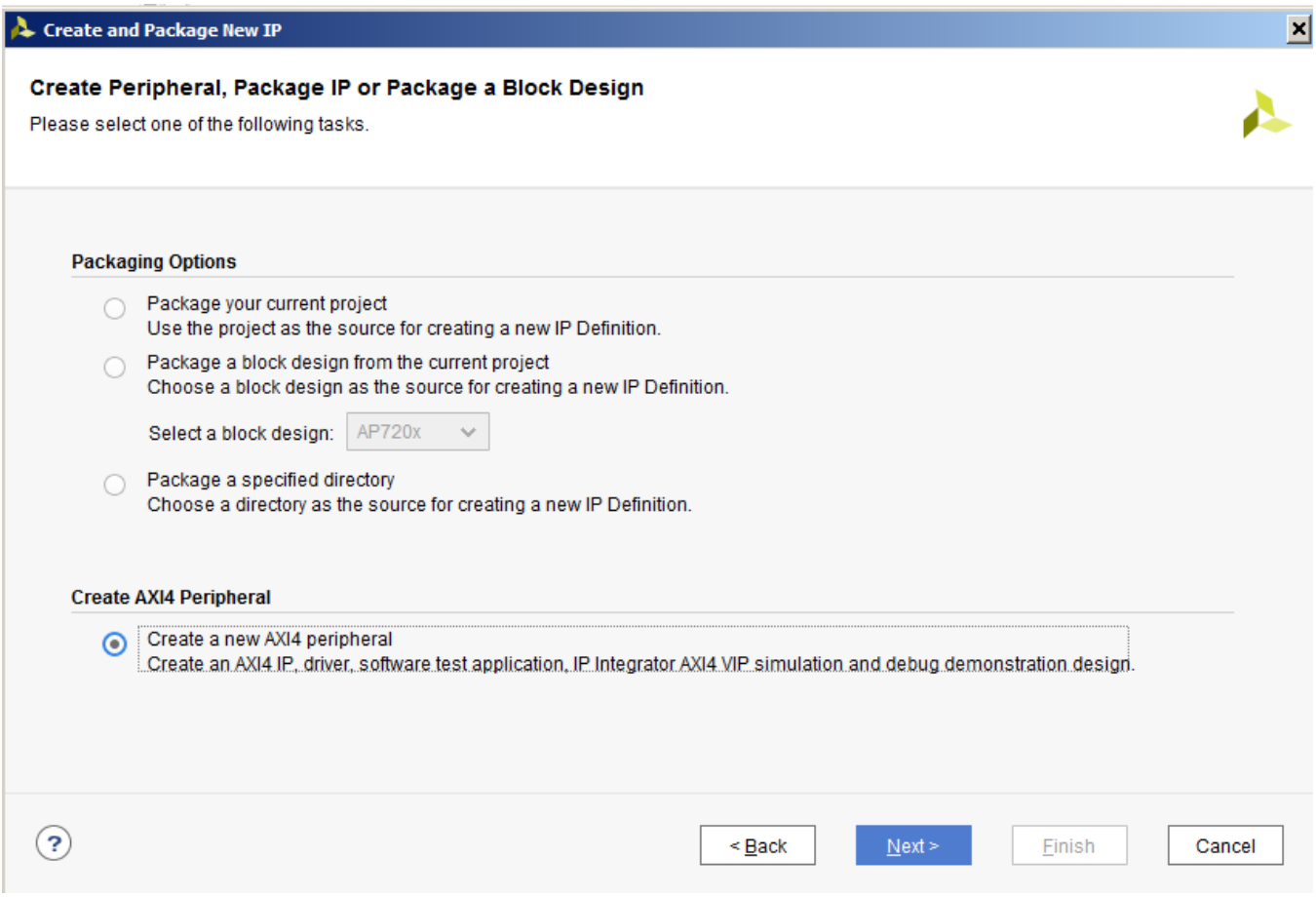
Hit the delete key to delete the "FIELD_IO" block.

Create a New Custom Peripheral

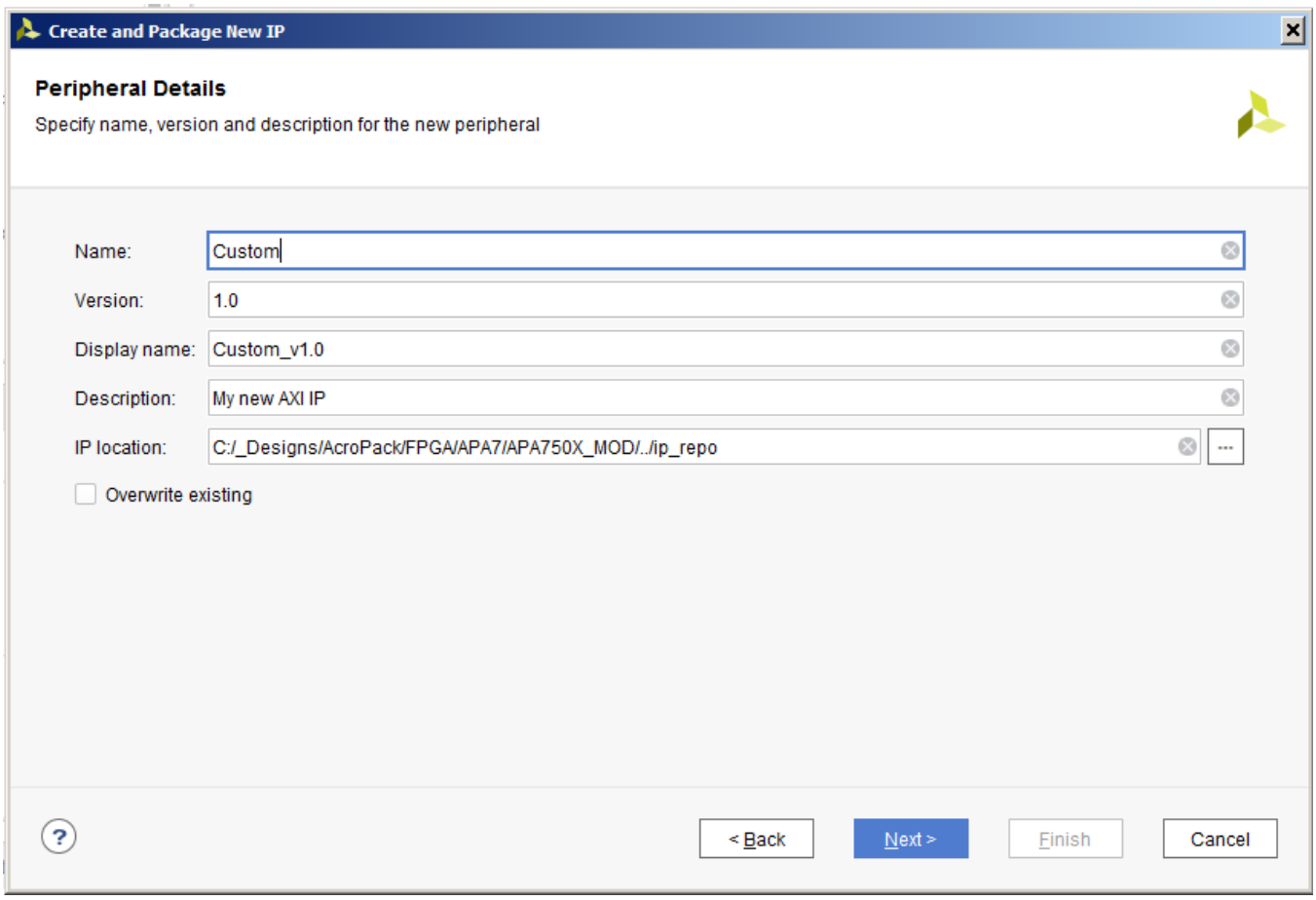
Select the "Create and Package New IP" command from the Tools menu.



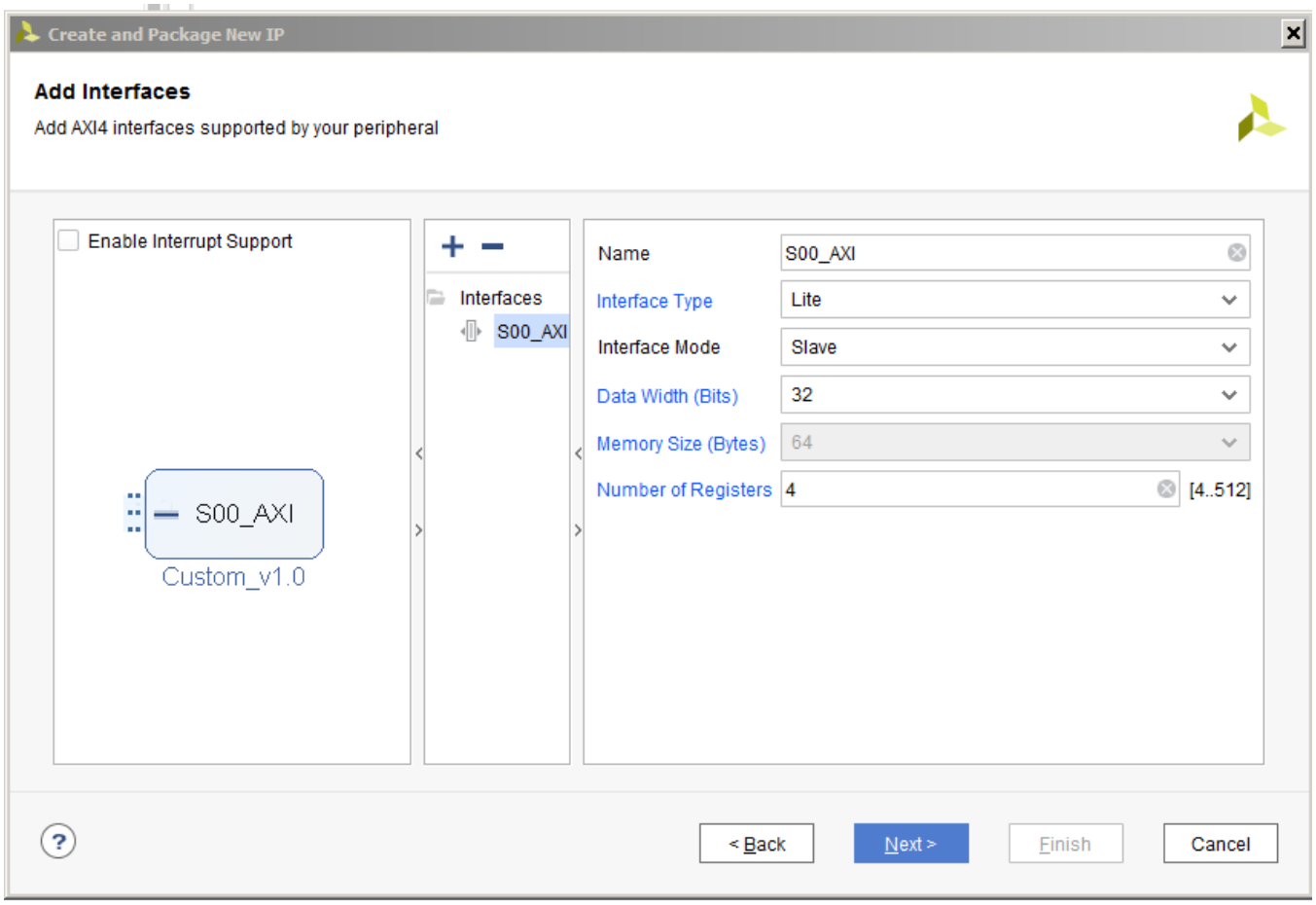
Click "Next". The following dialog box appears.



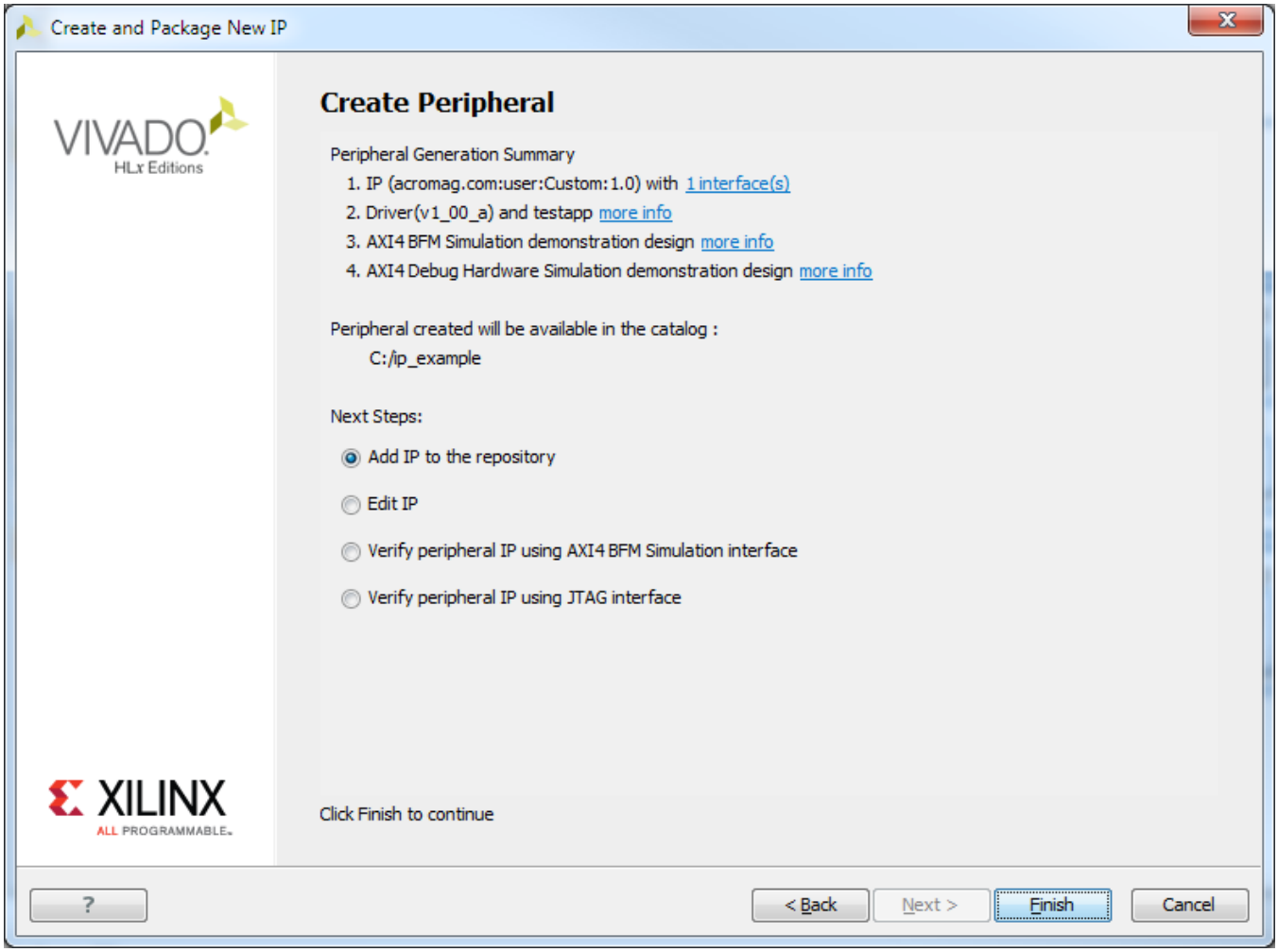
Select "Create a new AXI4 Peripheral". Click "Next". The following dialog box appears.



Enter "Custom" for the name of the peripheral.
Click "Next" to accept the remaining default entries.



The software will offer an AXI lite slave peripheral with four registers. Click "Next" to accept the suggested architecture. Click "Finish" to complete the creation of the custom peripheral.



The "Custom" peripheral is created and added to the IP library. Next we will modify the custom peripheral. Open the VHDL source file for the custom peripheral at location C:_Designs\AcroPack\FPGA\APA7\ip_repo\Custom_1.0\hdl\Custom_v1_0_S00_AXI.vhd in a text editor. The first few lines are shown below.

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3   use ieee.numeric_std.all;
4
5   entity Custom_v1_0_S00_AXI is
6       generic (
7           -- Users to add parameters here
8
9           -- User parameters ends
10          -- Do not modify the parameters beyond this line
11
12          -- Width of S_AXI data bus
13          C_S_AXI_DATA_WIDTH  : integer   := 32;
14          -- Width of S_AXI address bus
15          C_S_AXI_ADDR_WIDTH  : integer   := 4
16      );
17      port (
18          -- Users to add ports here
19
20          -- User ports ends
21          -- Do not modify the ports beyond this line
22
23          -- Global Clock Signal
24          S_AXI_ACLK  : in std_logic;
25          -- Global Reset Signal. This Signal is Active LOW
26          S_AXI_ARESETN   : in std_logic;
27          -- Write address (issued by master, acceped by Slave)
28          S_AXI_AWADDR    : in std_logic_vector(C_S_AXI_ADDR_WIDTH-1 downto 0);
```

Under the port statement, see the comment –Users to add ports here. We will add the ports to control the bidirectional I/O buffers below this line.

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3   use ieee.numeric_std.all;
4
5   entity Custom_v1_0_S00_AXI is
6     generic (
7       -- Users to add parameters here
8
9       -- User parameters ends
10      -- Do not modify the parameters beyond this line
11
12      -- Width of S_AXI data bus
13      C_S_AXI_DATA_WIDTH  : integer := 32;
14      -- Width of S_AXI address bus
15      C_S_AXI_ADDR_WIDTH  : integer := 4
16    );
17    port (
18      -- Users to add ports here
19      GPIO_A_tri_i : in STD_LOGIC_VECTOR ( 23 downto 0 );
20      GPIO_A_tri_o : out STD_LOGIC_VECTOR ( 23 downto 0 );
21      GPIO_A_tri_t : out STD_LOGIC_VECTOR ( 23 downto 0 );
22      GPIO_B_tri_i : in STD_LOGIC_VECTOR ( 23 downto 0 );
23      GPIO_B_tri_o : out STD_LOGIC_VECTOR ( 23 downto 0 );
24      GPIO_B_tri_t : out STD_LOGIC_VECTOR ( 23 downto 0 );
25
26      -- User ports ends
27      -- Do not modify the ports beyond this line
28
```

The example peripheral contains four 32bit registers that are accessible from the AXI bus. We will connect slv_reg0 to GPIO_A_tri_o, slv_reg1 to GPIO_A_tri_t, slv_reg2 to GPIO_B_tri_o, and slv_reg3 to GPIO_B_tri_t.

```
344   -- Implement memory mapped register select and read logic generation
345   -- Slave register read enable is asserted when valid address is available
346   -- and the slave is ready to accept the read address.
347   slv_reg_rden <= axi_arready and S_AXI_ARVALID and (not axi_rvalid) ;
348
349   process (slv_reg0, slv_reg1, slv_reg2, slv_reg3, axi_araddr, S_AXI_ARESETN, slv_reg_rden)
350   variable loc_addr :std_logic_vector(OPT_MEM_ADDR_BITS downto 0);
351   begin
352       -- Address decoding for reading registers
353       loc_addr := axi_araddr(ADDR_LSB + OPT_MEM_ADDR_BITS downto ADDR_LSB);
354       case loc_addr is
355         when b"00" =>
356           reg_data_out <= slv_reg0;
357         when b"01" =>
358           reg_data_out <= slv_reg1;
359         when b"10" =>
360           reg_data_out <= slv_reg2;
361         when b"11" =>
362           reg_data_out <= slv_reg3;
363         when others =>
364           reg_data_out  <= (others => '0');
365       end case;
366   end process;
```

Make the following modifications to the VHDL source:
Add lines 357, 362, and 390 - 393.

```
344   -- Implement memory mapped register select and read logic generation
345   -- Slave register read enable is asserted when valid address is available
346   -- and the slave is ready to accept the read address.
347   slv_reg_rden <= axi_arready and S_AXI_ARVALID and (not axi_rvalid) ;
348
349   process (slv_reg0, slv_reg1, slv_reg2, slv_reg3, axi_araddr, S_AXI_ARESETN, slv_reg_rden)
350   variable loc_addr :std_logic_vector(OPT_MEM_ADDR_BITS downto 0);
351   begin
352       -- Address decoding for reading registers
353       loc_addr := axi_araddr(ADDR_LSB + OPT_MEM_ADDR_BITS downto ADDR_LSB);
354       case loc_addr is
355         when b"00" =>
356           -- reg_data_out <= slv_reg0;
357           reg_data_out <= x"00" & GPIO_A_tri_i;
358         when b"01" =>
359           reg_data_out <= slv_reg1;
360         when b"10" =>
361           -- reg_data_out <= slv_reg2;
362           reg_data_out <= x"00" & GPIO_B_tri_i;
363         when b"11" =>
364           reg_data_out <= slv_reg3;
365         when others =>
366           reg_data_out  <= (others => '0');
367       end case;
368   end process;
369
```

```
387
388
389     -- Add user logic here
390       GPIO_A_tri_o <= slv_reg0(23 downto 0);
391       GPIO_A_tri_t <= slv_reg1(23 downto 0);
392       GPIO_B_tri_o <= slv_reg2(23 downto 0);
393       GPIO_B_tri_t <= slv_reg3(23 downto 0);
394
395     -- User logic ends
396
```

Save the changes.

Open the VHDL source file for the custom peripheral at location C:\ip_example\Custom_1.0\hdl\Custom_v1_0.vhd in a text editor. The first few lines are shown below.

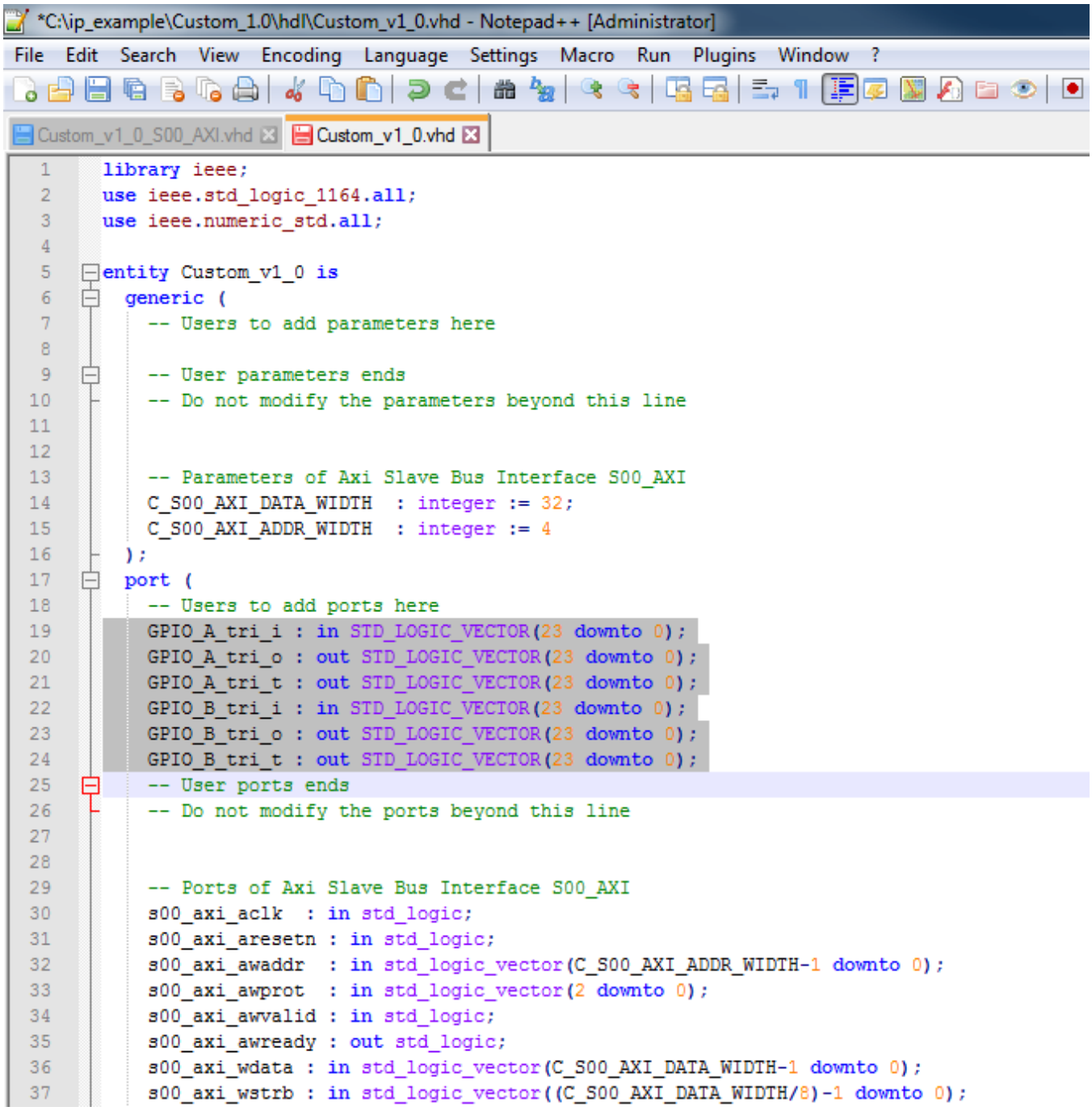
C:\ip_example\Custom_1.0\hdl\Custom_v1_0.vhd - Notepad++ [Administrator]

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

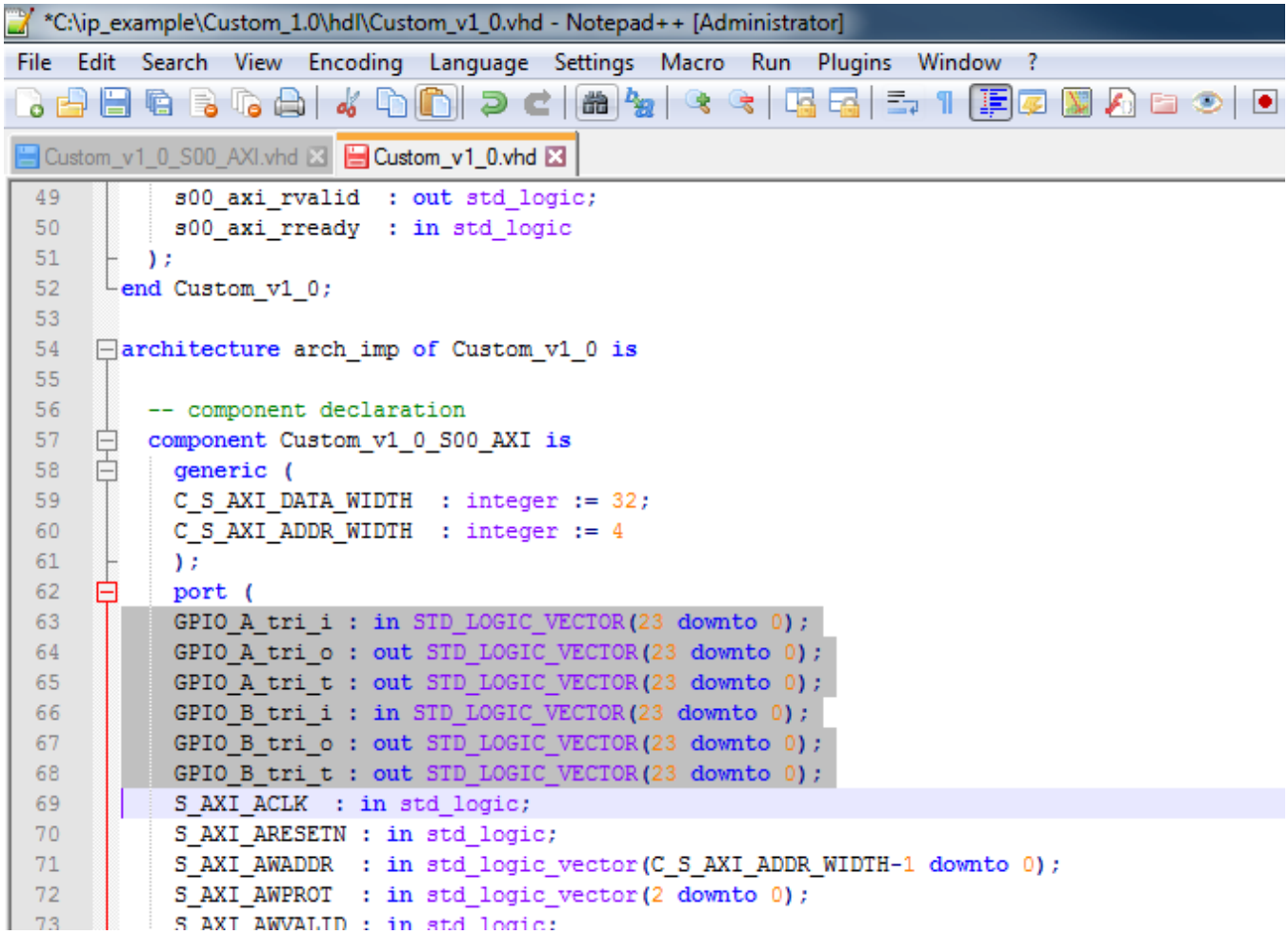
Custom_v1_0_S00_AXI.vhd | Custom_v1_0.vhd

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity Custom_v1_0 is
6    generic (
7      -- Users to add parameters here
8
9      -- User parameters ends
10     -- Do not modify the parameters beyond this line
11
12
13     -- Parameters of Axi Slave Bus Interface S00_AXI
14     C_S00_AXI_DATA_WIDTH  : integer := 32;
15     C_S00_AXI_ADDR_WIDTH  : integer := 4
16   );
17   port (
18     -- Users to add ports here
19
20     -- User ports ends
21     -- Do not modify the ports beyond this line
22
23
24     -- Ports of Axi Slave Bus Interface S00_AXI
25     s00_axi_aclk  : in std_logic;
26     s00_axi_aresetn : in std_logic;
27     s00_axi_awaddr  : in std_logic_vector(C_S00_AXI_ADDR_WIDTH-1 downto 0);
28     s00_axi_awprot  : in std_logic_vector(2 downto 0);
29     s00_axi_awvalid : in std_logic;
30     s00_axi_awready : out std_logic;
31     s00_axi_wdata : in std_logic_vector(C_S00_AXI_DATA_WIDTH-1 downto 0);
32     s00_axi_wstrb : in std_logic_vector((C_S00_AXI_DATA_WIDTH/8)-1 downto 0);
33     s00_axi_wvalid  : in std_logic;
```

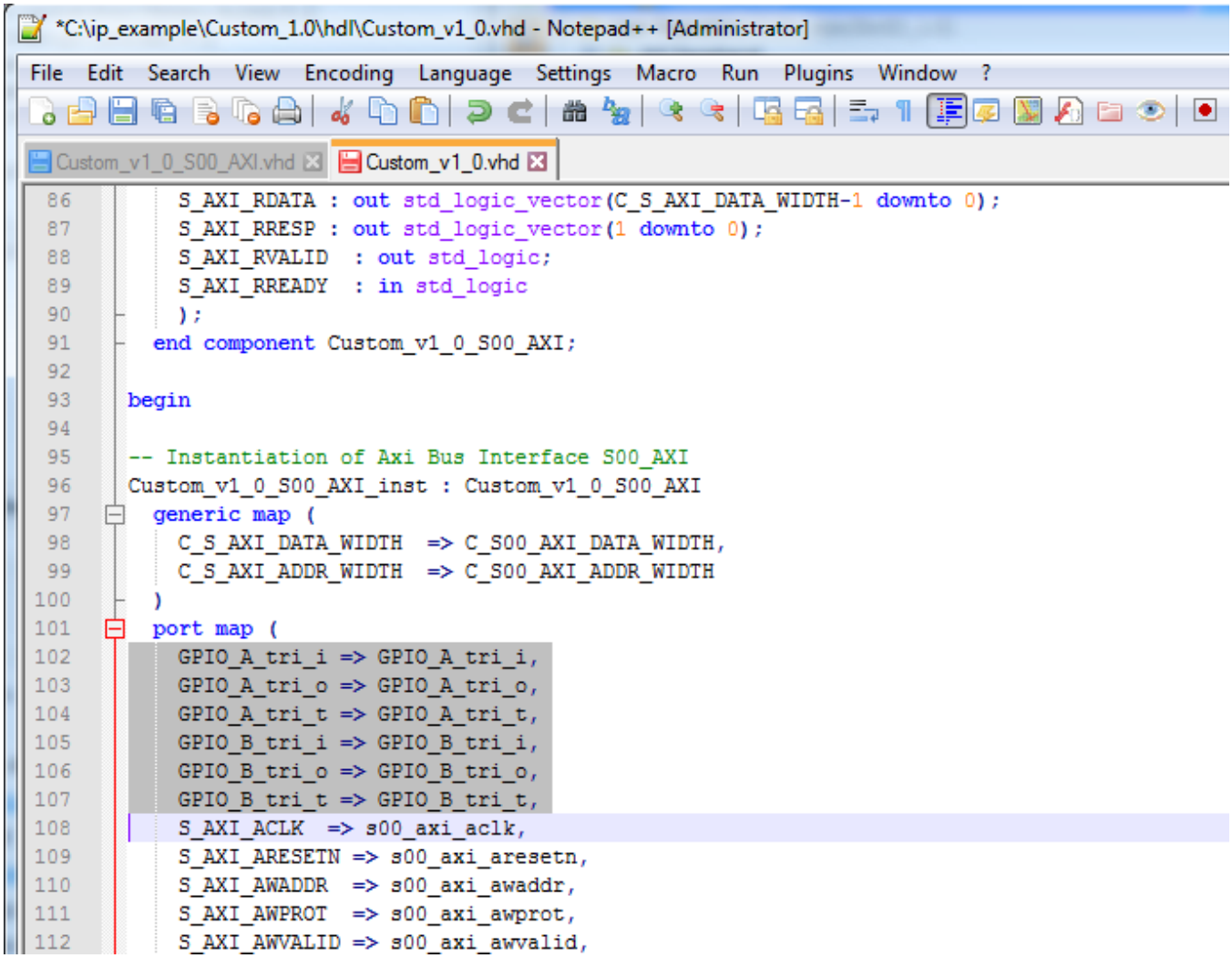
Under the port statement, see the comment –Users to add ports here. We will add the ports to control the bidirectional I/O buffers below this line.



Add lines 19-24.

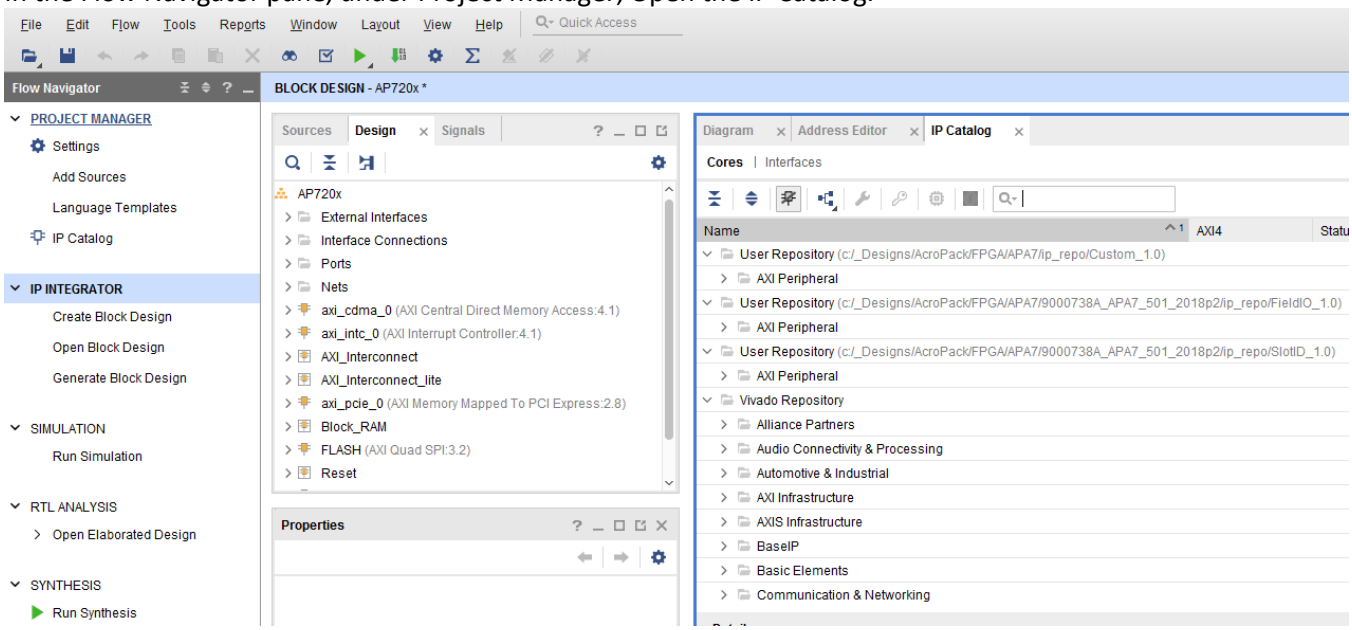


Add lines 63-68.

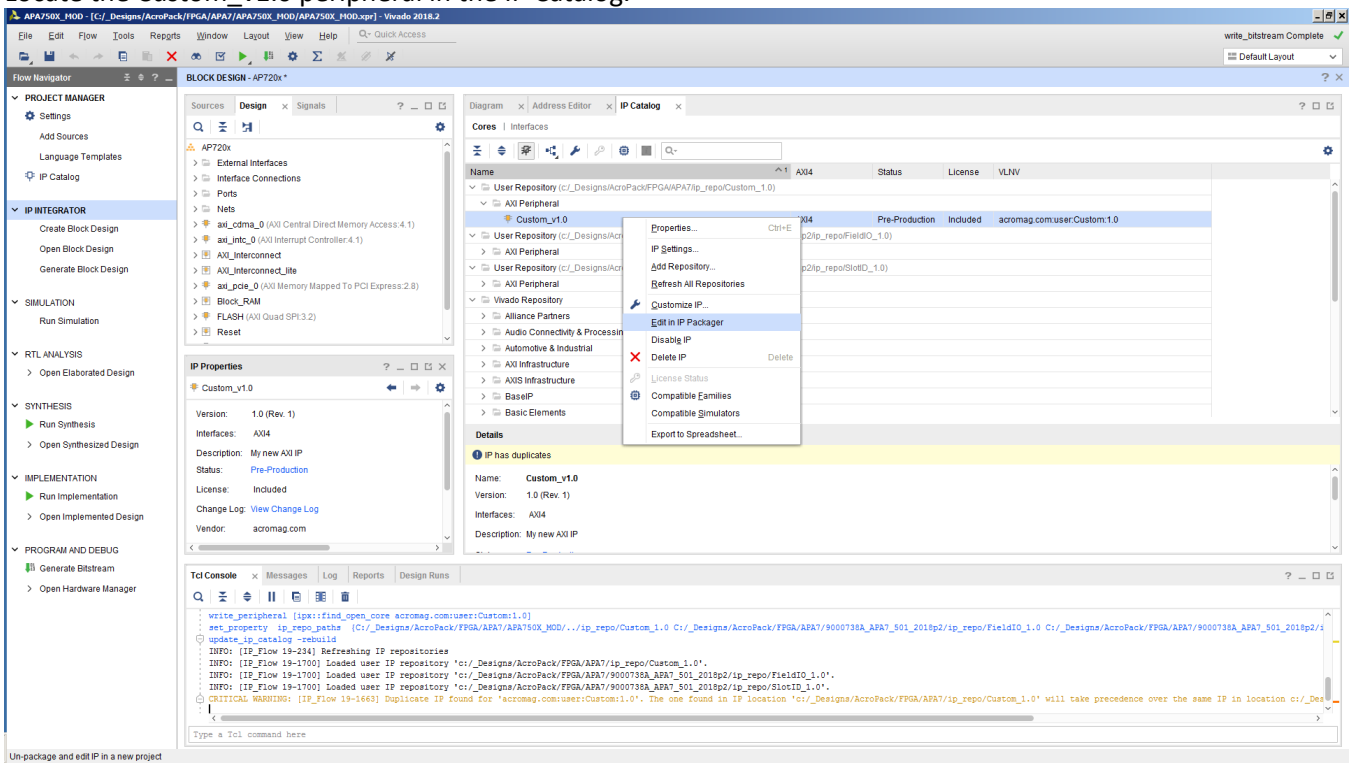


Add lines 102-107. Save the file.

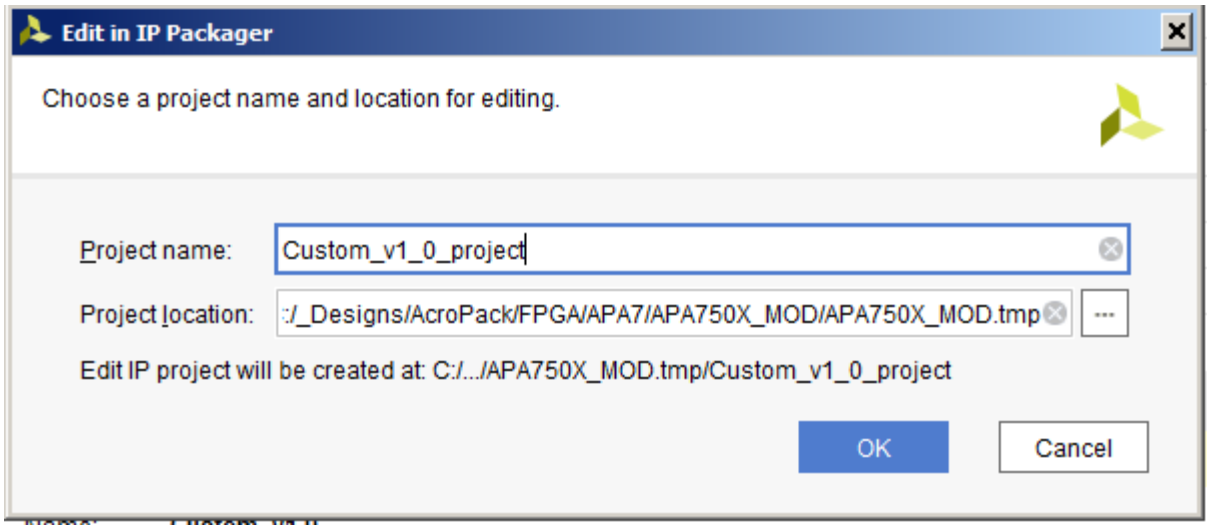
In the Flow Navigator pane, under Project Manager, Open the IP Catalog.



Locate the Custom_v1.0 peripheral in the IP Catalog.

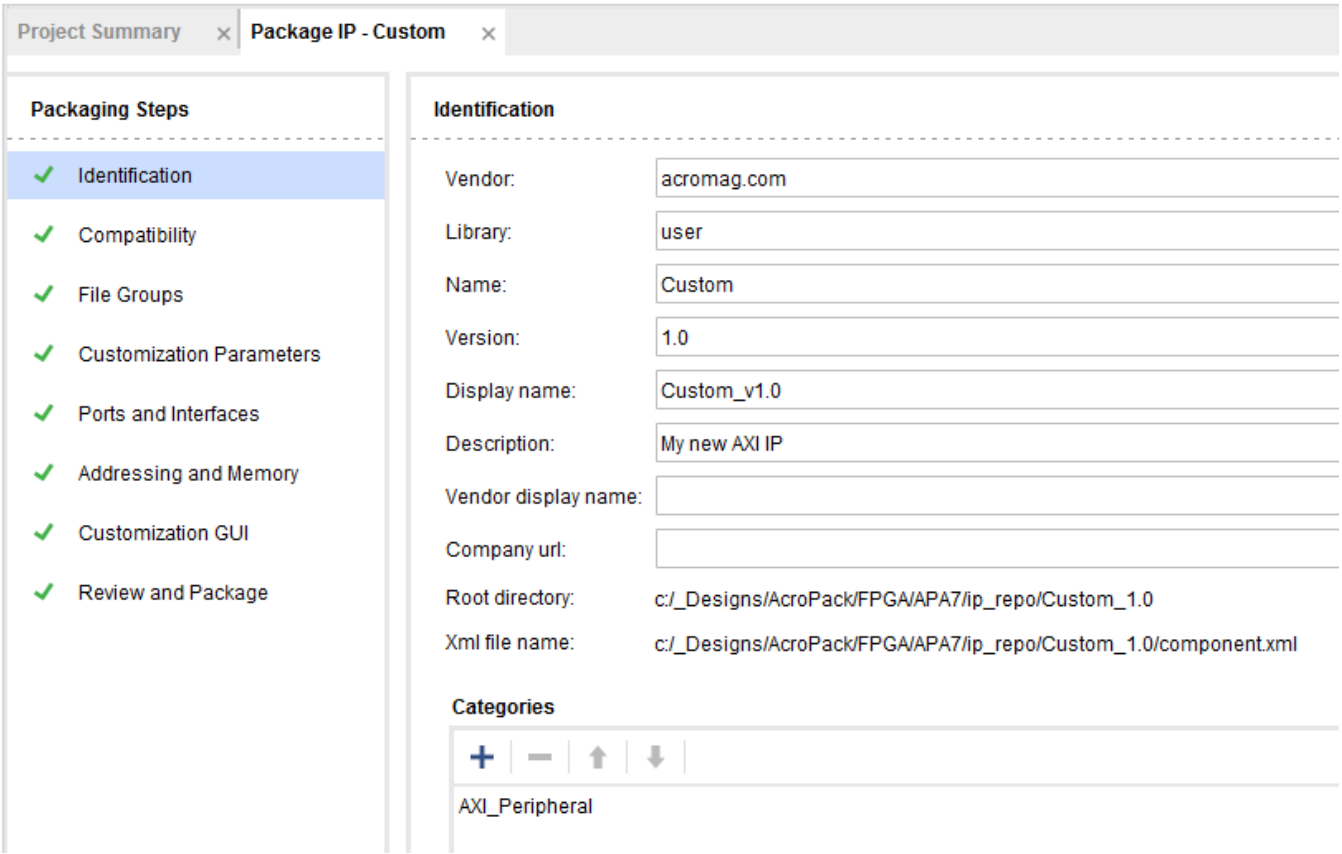


Right click on the Custom_V1.0 peripheral and select the "Edit in IP Packager" from the dropdown list. The following dialog box appears:

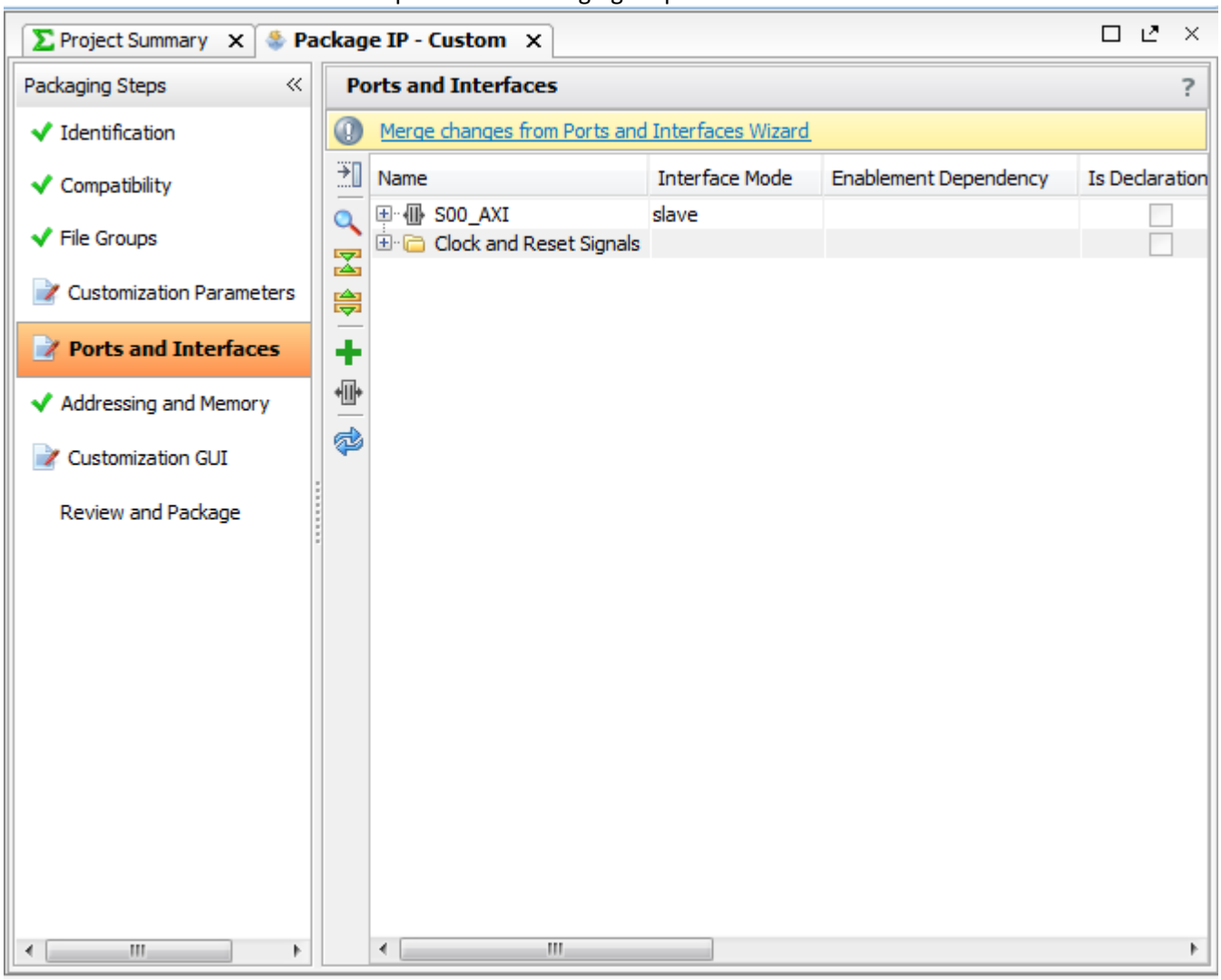


Click OK.

A second instance of Vivado is launched with the Custom_v1_0 project open in the Package IP window.

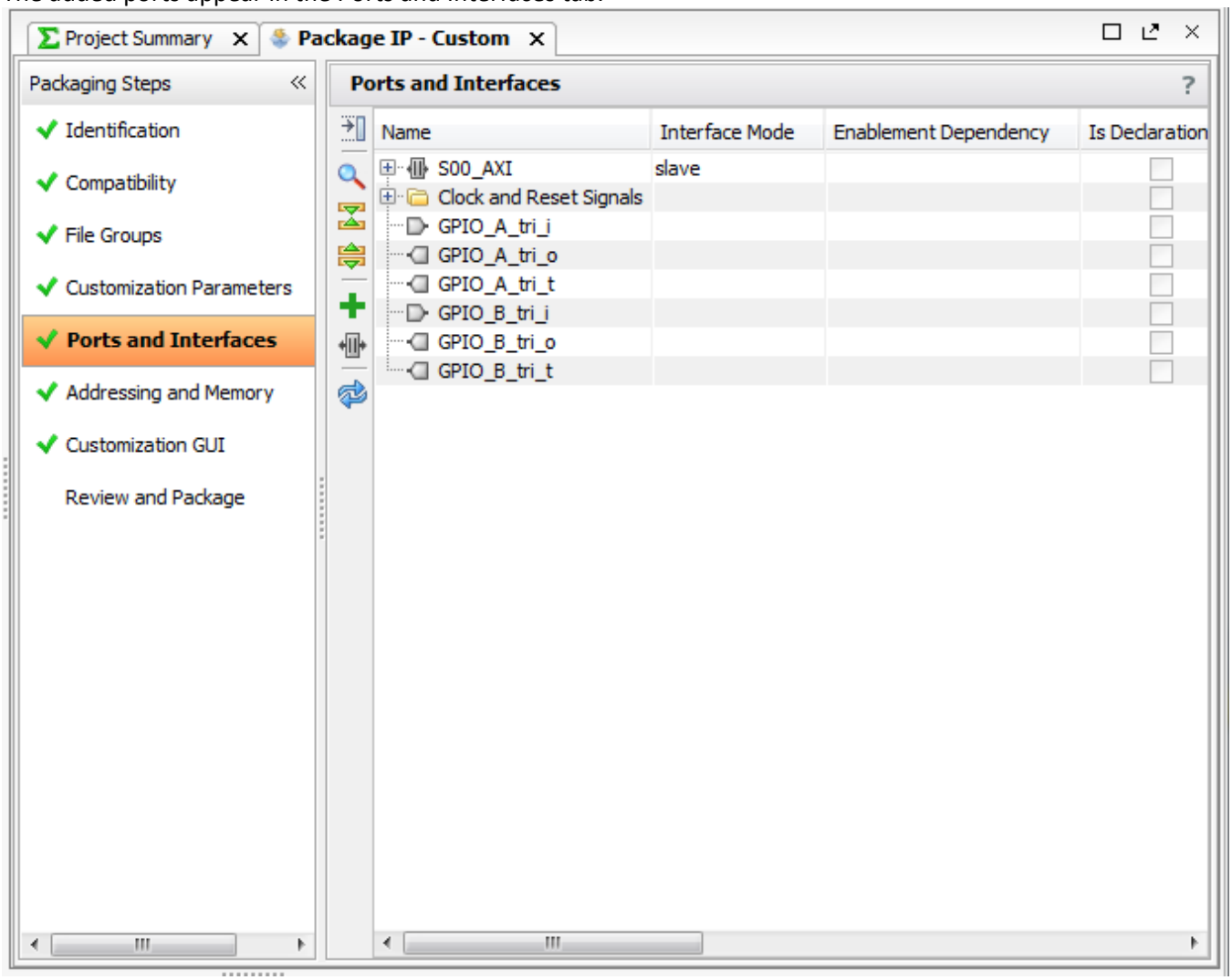


Select the "Ports and Interfaces" step from the Packaging Steps list.

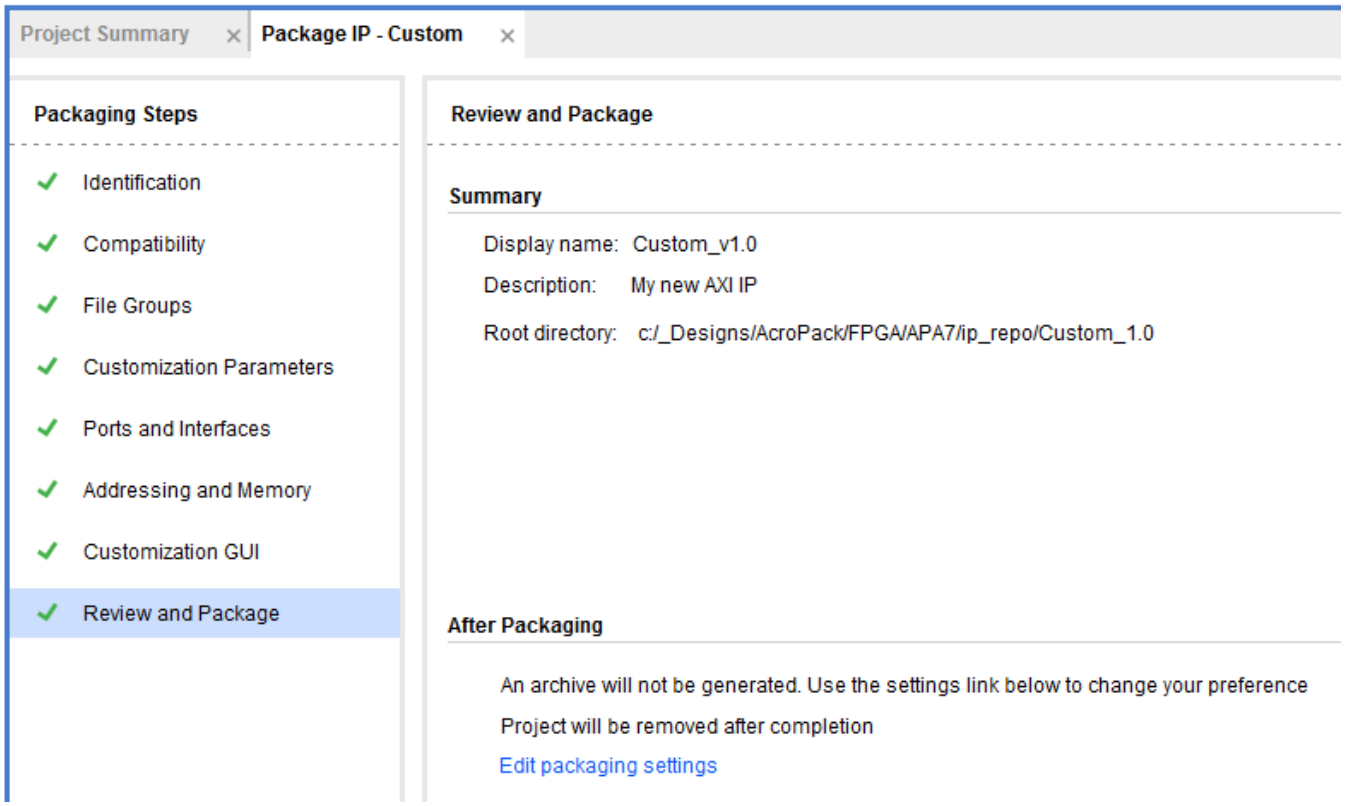


Click on "Merge changes from Ports and Interfaces Wizard".

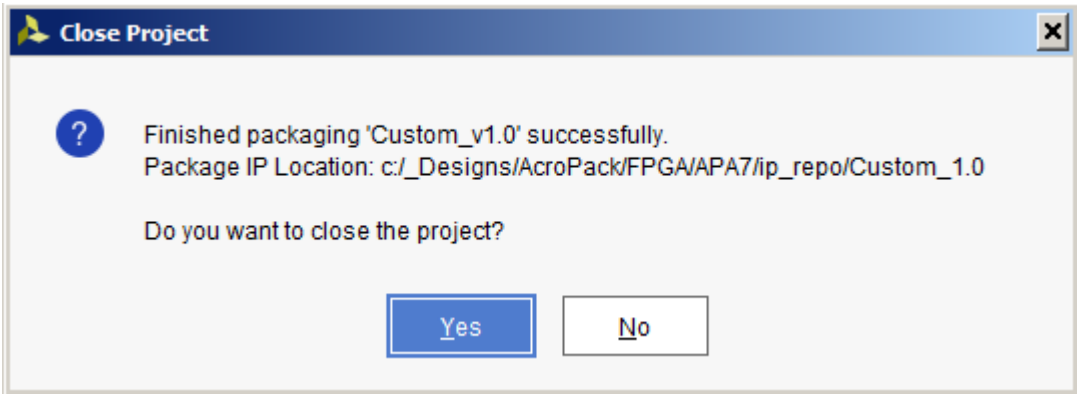
The added ports appear in the Ports and Interfaces tab.



Click on the "Review and Package" step.



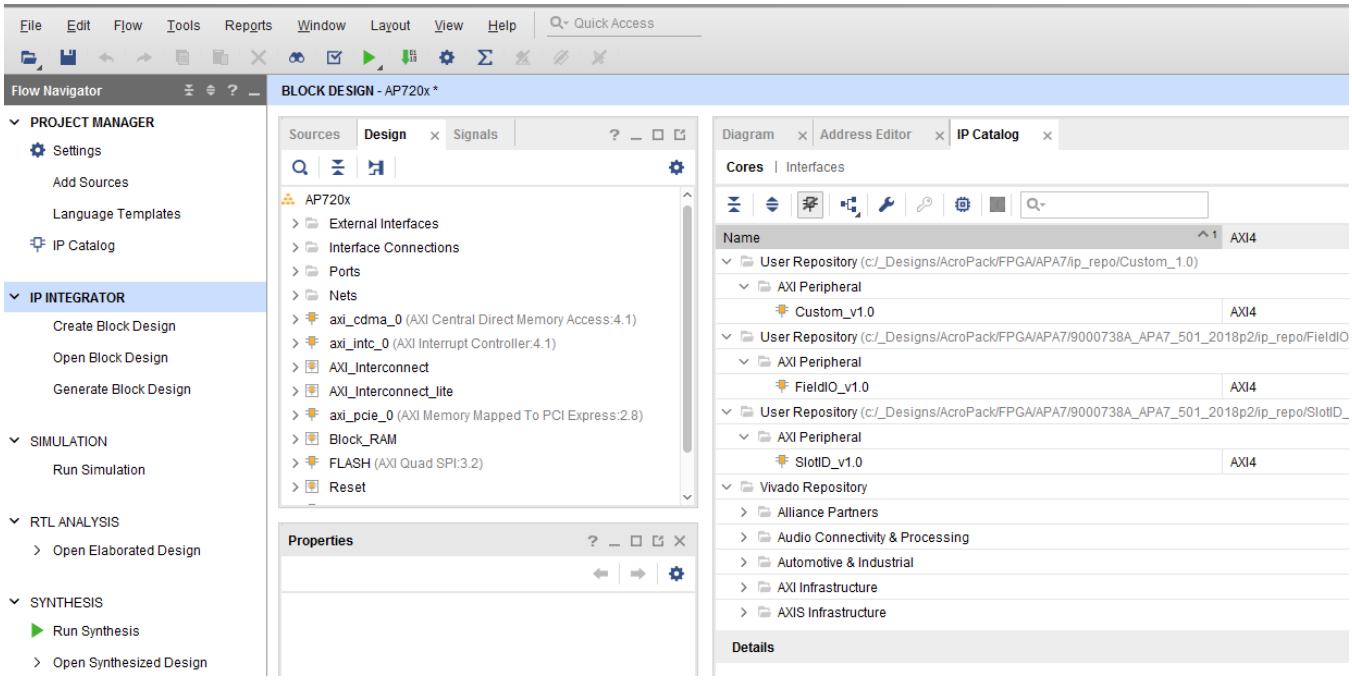
Click on the "Re-Package IP" button. The following dialog box appears.



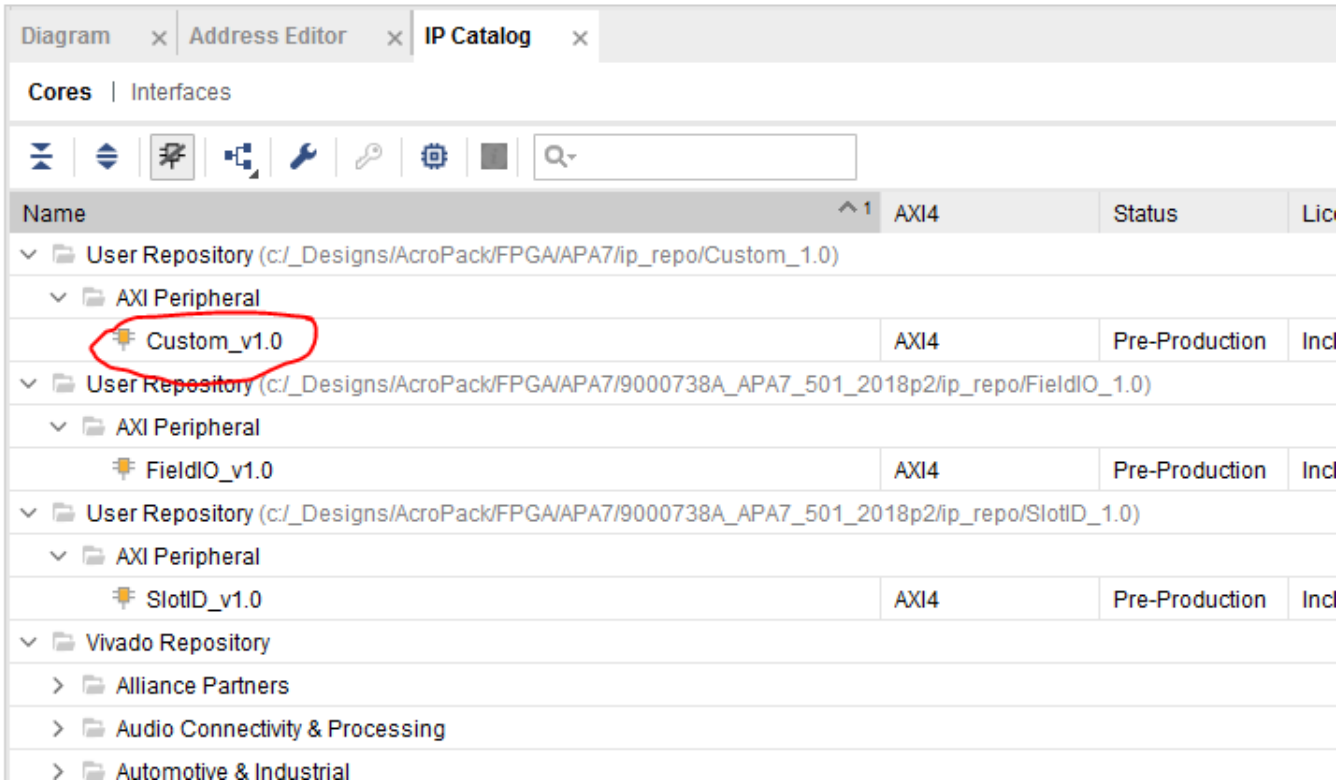
Click on the "Yes" button. The second instance of Vivado closes.

Add the Custom Peripheral to the Block Diagram

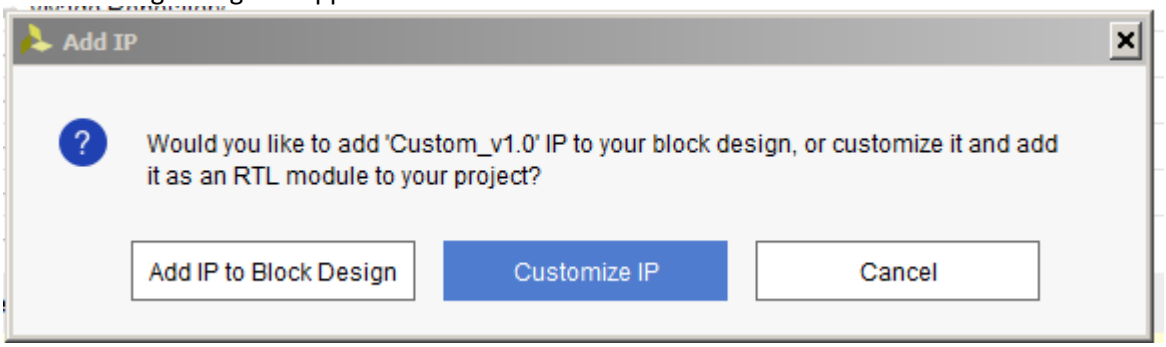
Return to the first Vivado instance, the "APA750X_MOD" project and ensure the "IP Catalog" tab is visible in the "Block Design" pane.



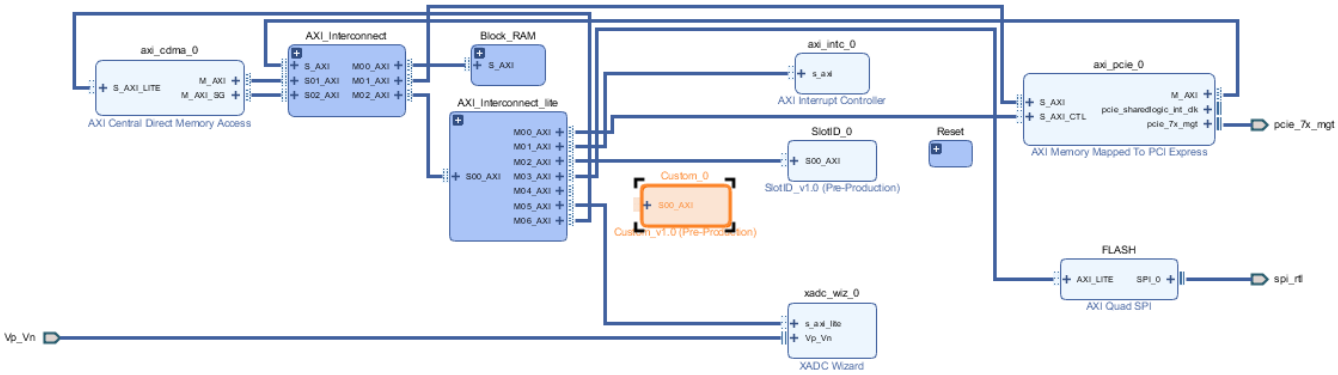
Double click the line "Custom_V1.0"



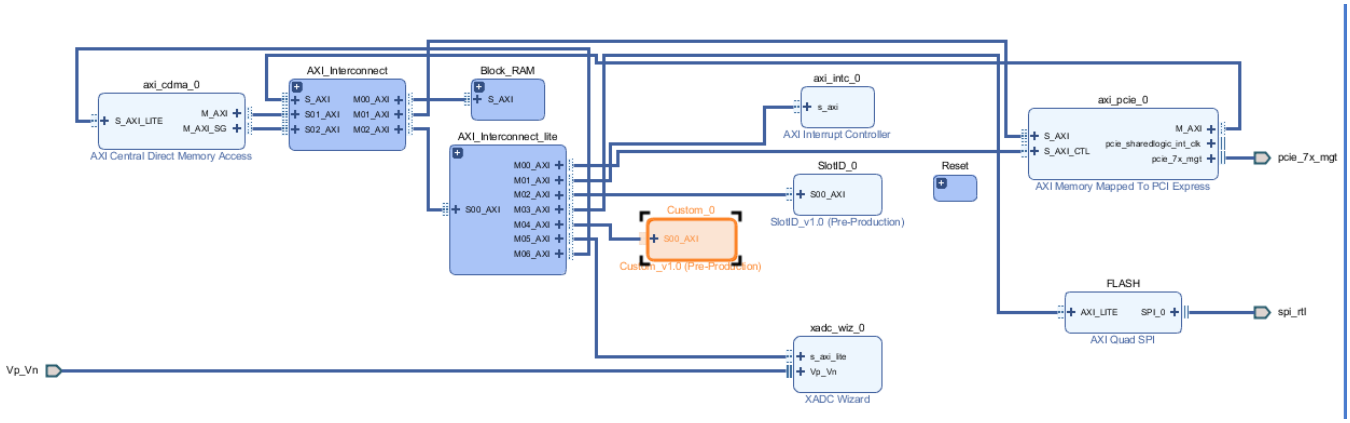
The following dialog box appears:



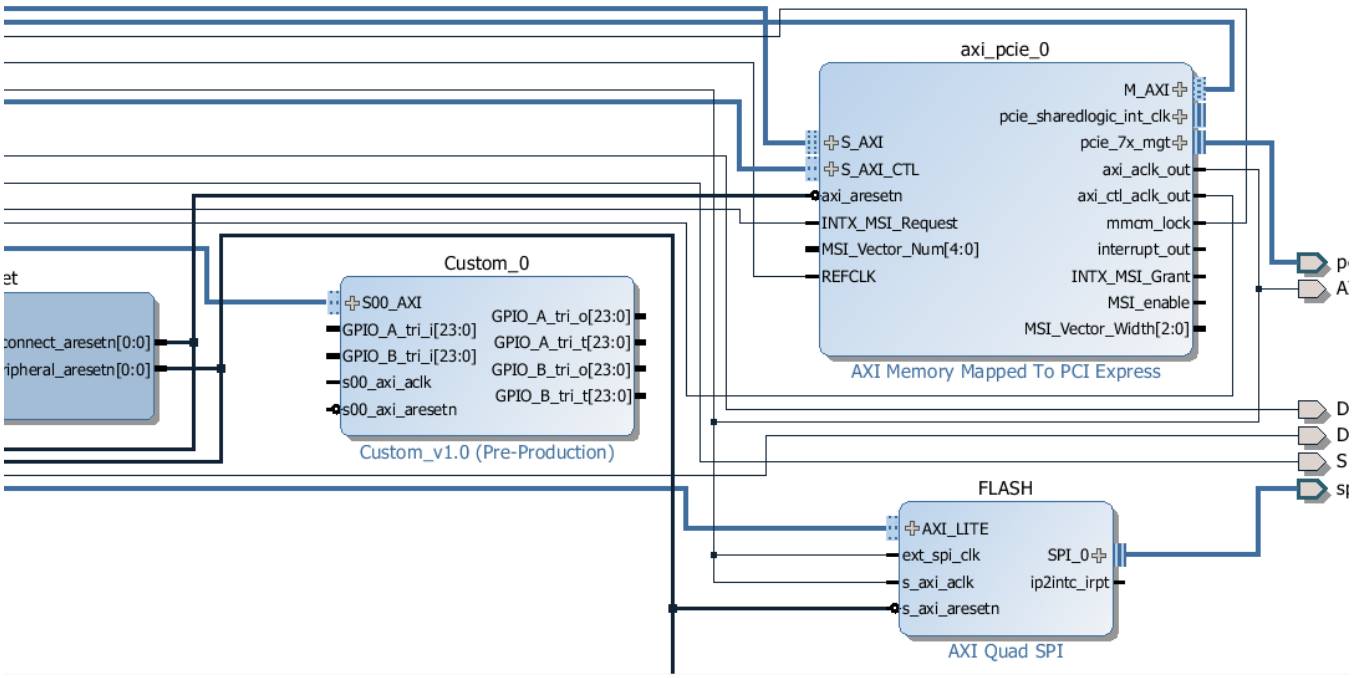
Click the "Add IP to Block Design" button. The "Custom" block appears on the block diagram.



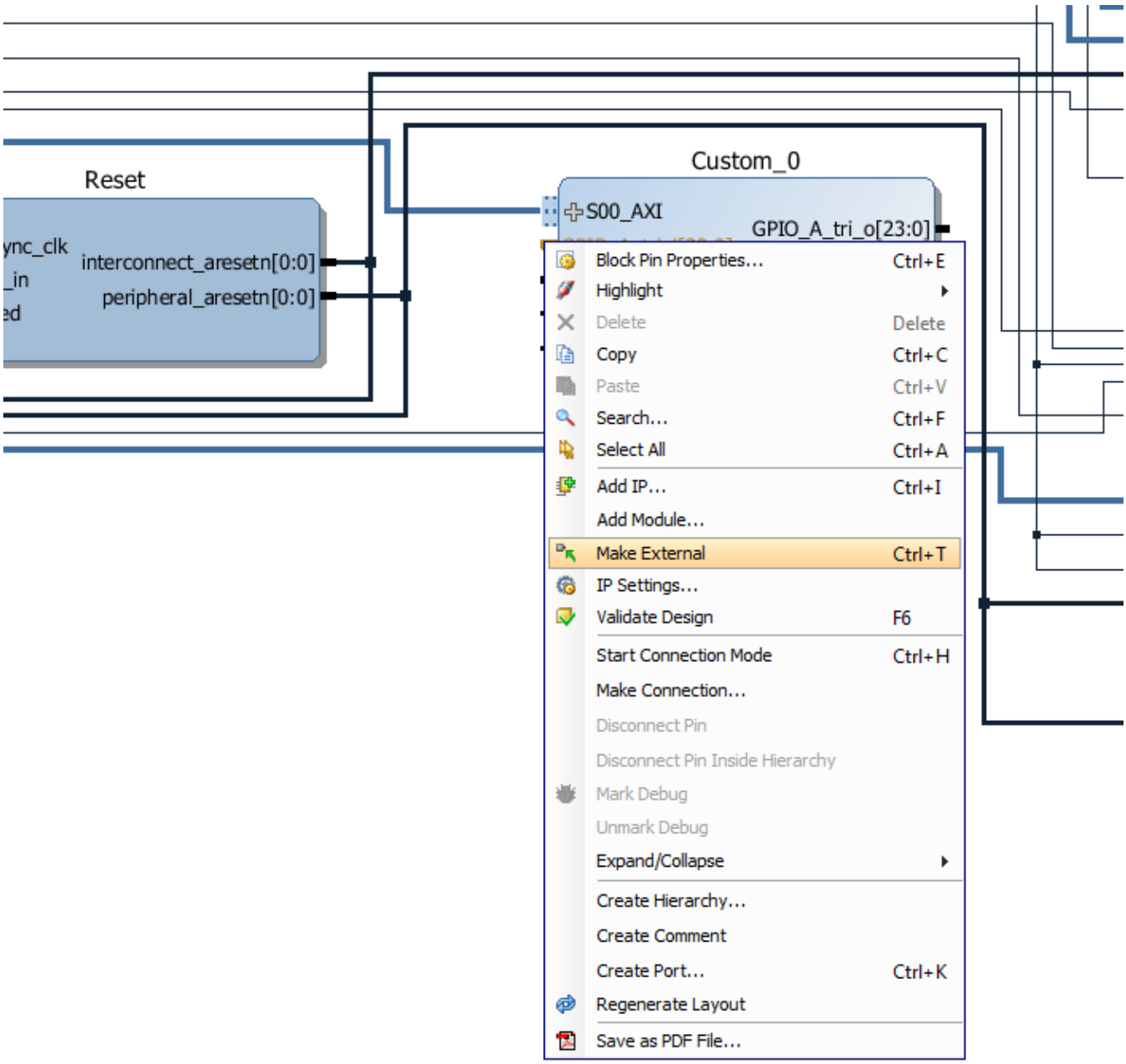
Connect the "M04_AXI" pin of the "AXI_Interrconnet_lite" block to the "S00_AXI" pin of the "Custom_0" block as shown in the diagram below.



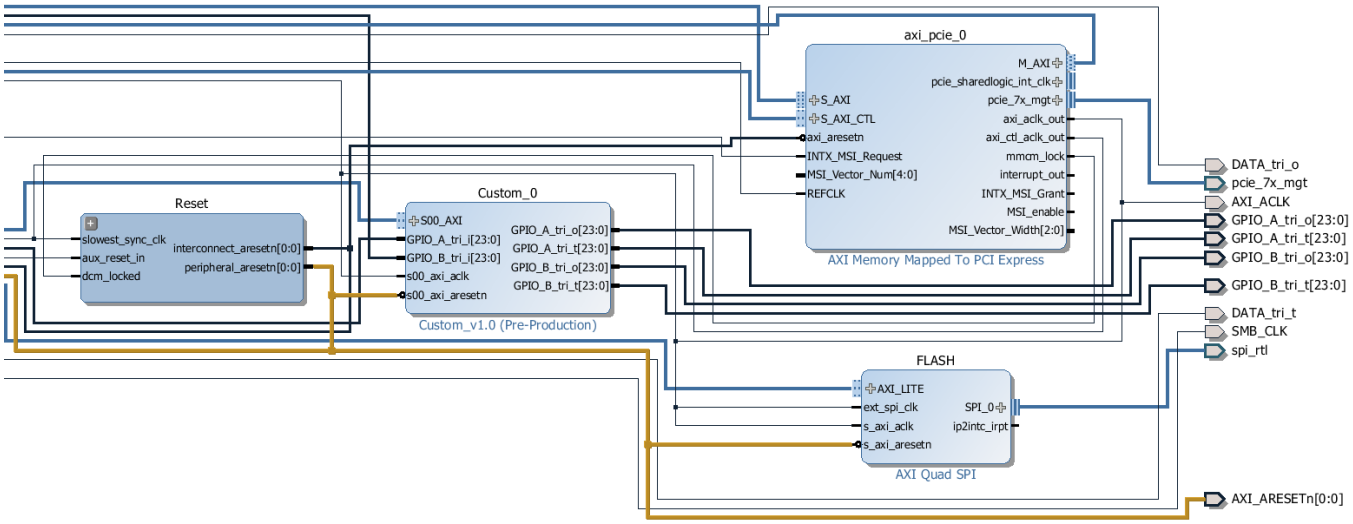
Click on the "show interface connections only"  button to restore the view of the block pins.


axi_pcie_0
M_AXI
pcie_sharedlogic_int_clk
S_AXI
pcie_7x_mgt
S_AXI_CTL
axi_aclk_out
axi_aresetn
axi_ctl_aclk_out
INTX_MSI_Request
mmcm_lock
MSI_Vector_Num[4:0]
interrupt_out
REFCLK
INTX_MSI_Grant
MSI_enable
MSI_Vector_Width[2:0]
AXI Memory Mapped To PCI Express
Custom_0
S00_AXI
GPIO_A_tri_o[23:0]
GPIO_A_tri_i[23:0]
GPIO_A_tri_t[23:0]
GPIO_B_tri_i[23:0]
GPIO_B_tri_o[23:0]
s00_axi_aclk
GPIO_B_tri_t[23:0]
s00_axi_aresetn
Custom_v1.0 (Pre-Production)
FLASH
AXI_LITE
ext_spi_clk
SPI_0
s_axi_aclk
ip2intc_irpt
s_axi_aresetn
AXI Quad SPI

Right-click on the "GPIO_A_tri_i[23:0]" pin of the Custom_0 block.

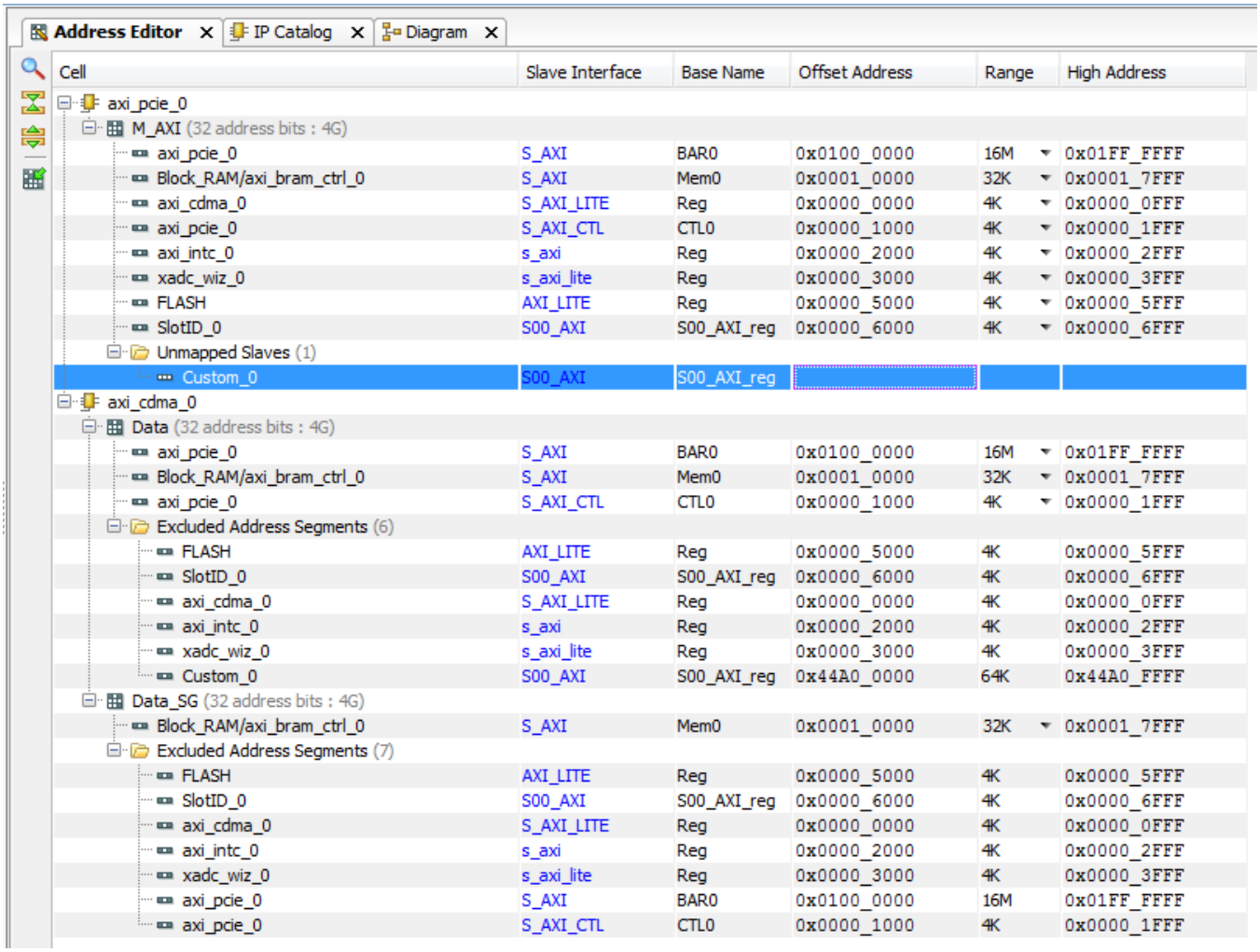


Select the "Make External" command from the list. Repeat these steps for the "GPIO_B_tri_i[23:0]", "GPIO_A_tri_o[23:0]", "GPIO_A_tri_t[23:0]", "GPIO_B_tri_o[23:0]", and "GPIO_B_tri_t[23:0]" pins. Connect the "s00_axi_aclk" pin of the "Custom_0" block to the "axi_aclk_out" pin of the "axi_pcie_0" block. Connect the ""s00_axi_aresetn" pin of the "Custom_0" block to the "peripheral_aresetn[0:0]" pin of the "Reset" block.



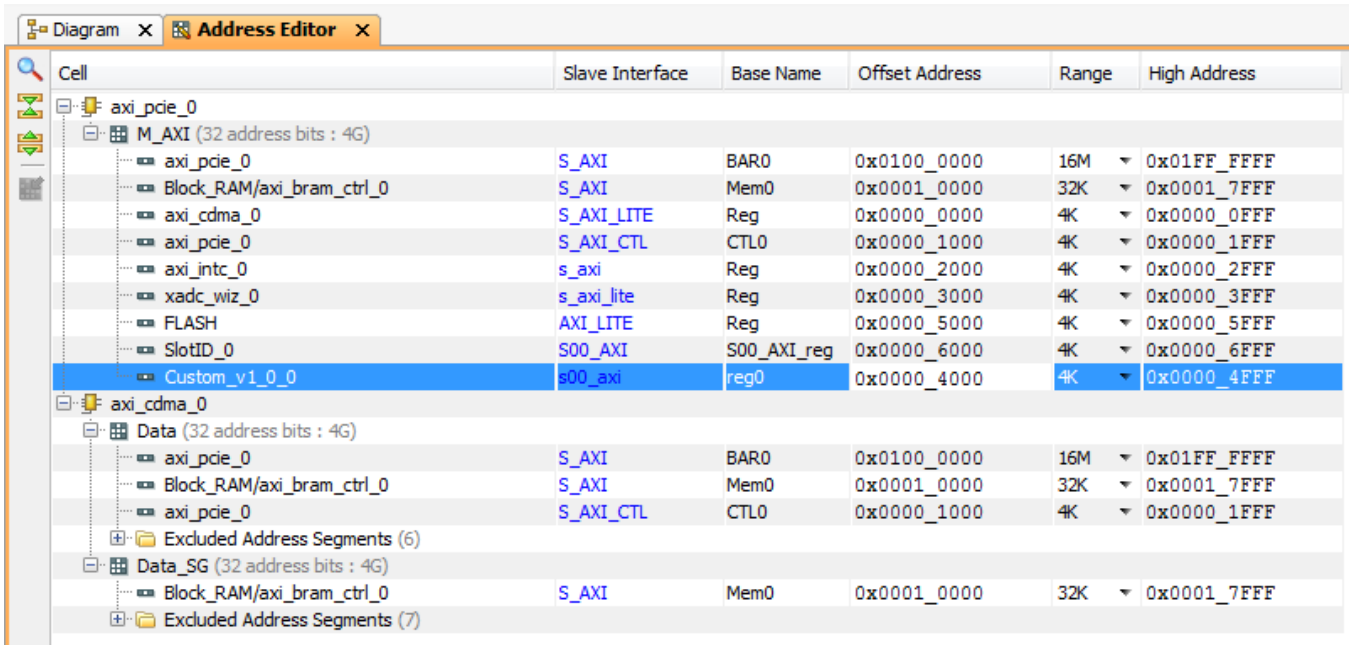
We want the new Custom peripheral to reside at the same address as the GPIO peripheral it is replacing. We will need to change the "Offset Address" of the Custom peripheral. Click on the "Address Editor" tab to open the Address editor.

The Address Editor shows the Custom_0 peripheral is accessible from the axi_pcie_0 master and it is an "Unmapped Slave".



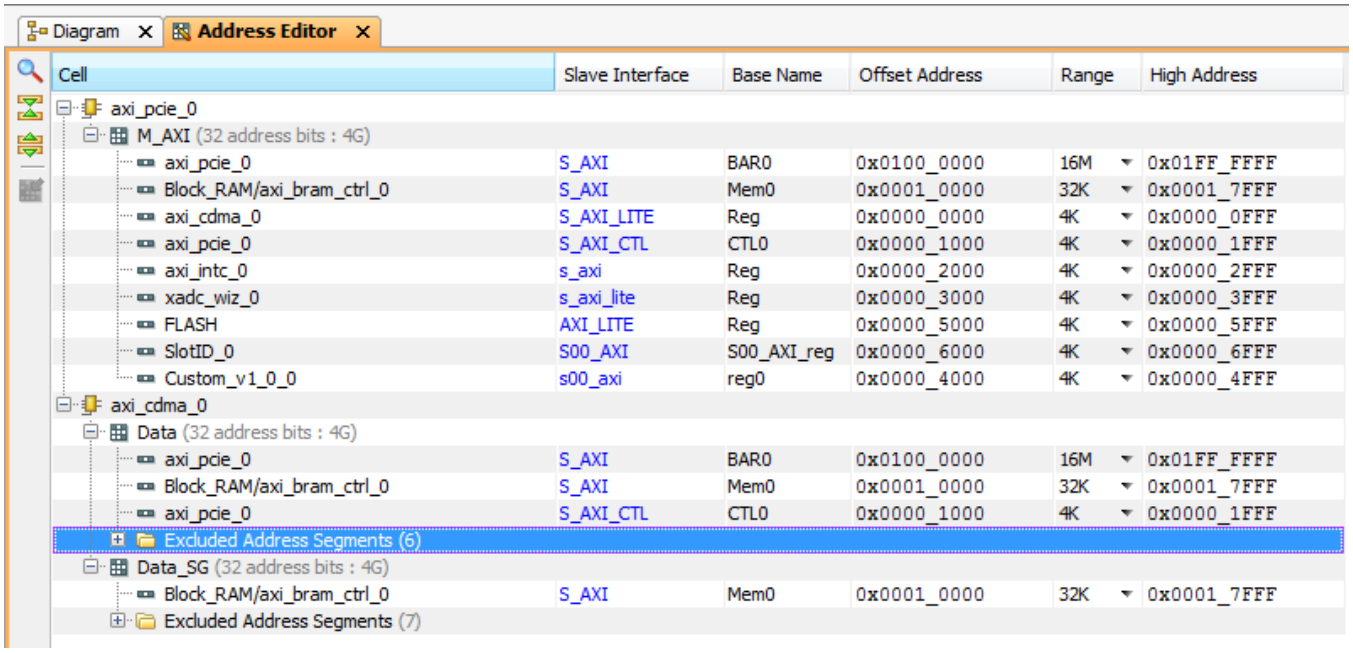
Cell	Slave Interface	Base Name	Offset Address	Range	High Address
axi_pcie_0					
M_AXI (32 address bits : 4G)					
axi_pcie_0	S_AXI	BAR0	0x0100_0000	16M	0x01FF_FFFF
Block_RAM/axi_bram_ctrl_0	S_AXI	Mem0	0x0001_0000	32K	0x0001_7FFF
axi_cdma_0	S_AXI_LITE	Reg	0x0000_0000	4K	0x0000_0FFF
axi_pcie_0	S_AXI_CTL	CTL0	0x0000_1000	4K	0x0000_1FFF
axi_intc_0	s_axi	Reg	0x0000_2000	4K	0x0000_2FFF
xadc_wiz_0	s_axi_lite	Reg	0x0000_3000	4K	0x0000_3FFF
FLASH	AXI_LITE	Reg	0x0000_5000	4K	0x0000_5FFF
SlotID_0	S00_AXI	S00_AXI_reg	0x0000_6000	4K	0x0000_6FFF
Unmapped Slaves (1)					
Custom_0	S00_AXI	S00_AXI_reg			
axi_cdma_0					
Data (32 address bits : 4G)					
axi_pcie_0	S_AXI	BAR0	0x0100_0000	16M	0x01FF_FFFF
Block_RAM/axi_bram_ctrl_0	S_AXI	Mem0	0x0001_0000	32K	0x0001_7FFF
axi_pcie_0	S_AXI_CTL	CTL0	0x0000_1000	4K	0x0000_1FFF
Excluded Address Segments (6)					
FLASH	AXI_LITE	Reg	0x0000_5000	4K	0x0000_5FFF
SlotID_0	S00_AXI	S00_AXI_reg	0x0000_6000	4K	0x0000_6FFF
axi_cdma_0	S_AXI_LITE	Reg	0x0000_0000	4K	0x0000_0FFF
axi_intc_0	s_axi	Reg	0x0000_2000	4K	0x0000_2FFF
xadc_wiz_0	s_axi_lite	Reg	0x0000_3000	4K	0x0000_3FFF
Custom_0	S00_AXI	S00_AXI_reg	0x44A0_0000	64K	0x44A0_FFFF
Data_SG (32 address bits : 4G)					
Block_RAM/axi_bram_ctrl_0	S_AXI	Mem0	0x0001_0000	32K	0x0001_7FFF
Excluded Address Segments (7)					
FLASH	AXI_LITE	Reg	0x0000_5000	4K	0x0000_5FFF
SlotID_0	S00_AXI	S00_AXI_reg	0x0000_6000	4K	0x0000_6FFF
axi_cdma_0	S_AXI_LITE	Reg	0x0000_0000	4K	0x0000_0FFF
axi_intc_0	s_axi	Reg	0x0000_2000	4K	0x0000_2FFF
xadc_wiz_0	s_axi_lite	Reg	0x0000_3000	4K	0x0000_3FFF
axi_pcie_0	S_AXI	BAR0	0x0100_0000	16M	0x01FF_FFFF
axi_pcie_0	S_AXI_CTL	CTL0	0x0000_1000	4K	0x0000_1FFF

Right-click on the "Custom_0" peripheral and select the "Assign Address" command. The system assigns an address to the "Cusom_0" peripheral.
The offset address and range can be changed by entering the new desired offset address and picking the desired range from the picklist. First, select the 4K range from the pick list, and then enter the new offset address 0x0000_4000.



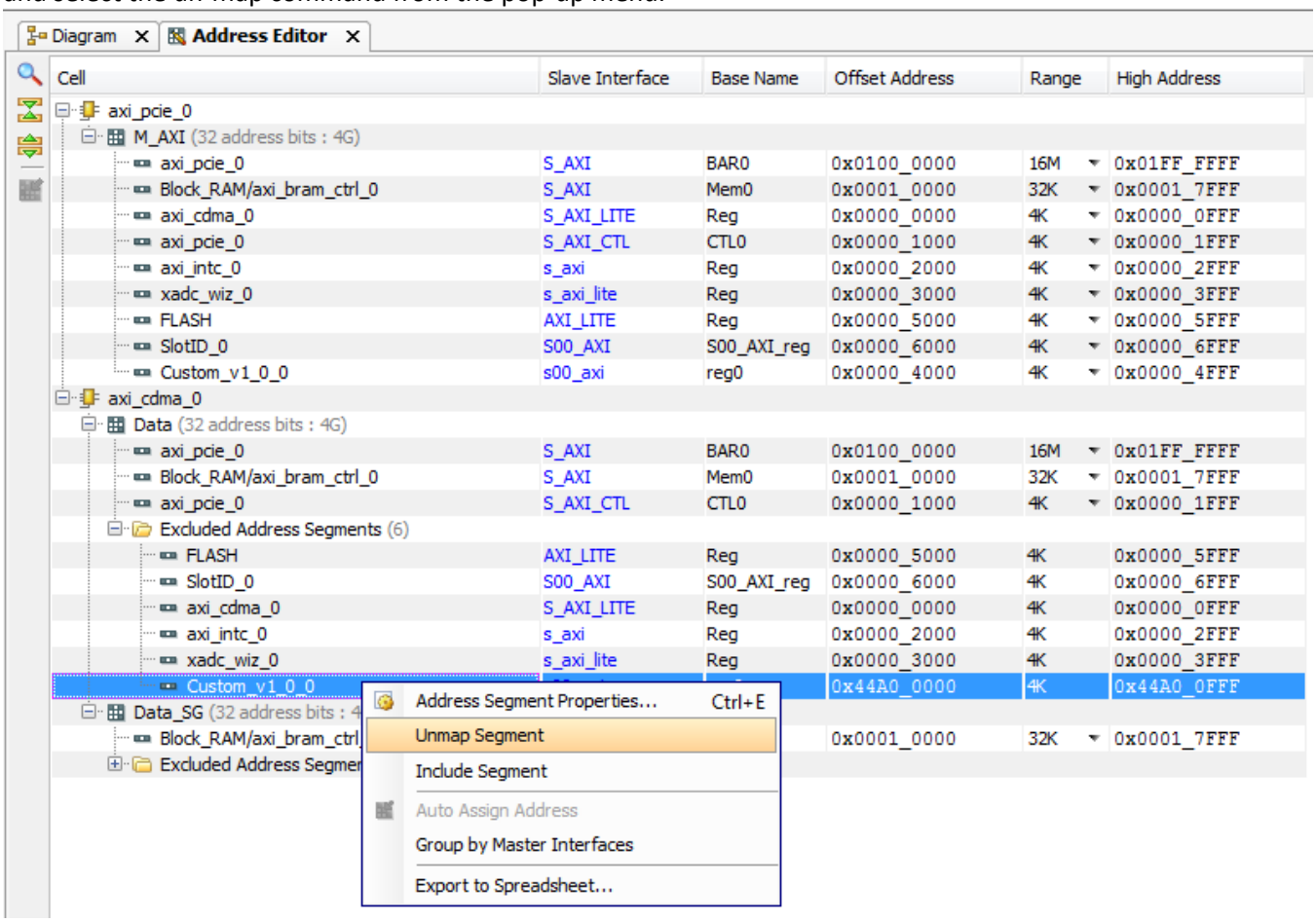
Cell	Slave Interface	Base Name	Offset Address	Range	High Address
axi_pcie_0					
M_AXI (32 address bits : 4G)					
axi_pcie_0	S_AXI	BAR0	0x0100_0000	16M	0x01FF_FFFF
Block_RAM/axi_bram_ctrl_0	S_AXI	Mem0	0x0001_0000	32K	0x0001_7FFF
axi_cdma_0	S_AXI_LITE	Reg	0x0000_0000	4K	0x0000_0FFF
axi_pcie_0	S_AXI_CTL	CTL0	0x0000_1000	4K	0x0000_1FFF
axi_intc_0	s_axi	Reg	0x0000_2000	4K	0x0000_2FFF
xadc_wiz_0	s_axi_lite	Reg	0x0000_3000	4K	0x0000_3FFF
FLASH	AXI_LITE	Reg	0x0000_5000	4K	0x0000_5FFF
SlotID_0	S00_AXI	S00_AXI_reg	0x0000_6000	4K	0x0000_6FFF
Custom_v1_0_0	s00_axi	reg0	0x0000_4000	4K	0x0000_4FFF
axi_cdma_0					
Data (32 address bits : 4G)					
axi_pcie_0	S_AXI	BAR0	0x0100_0000	16M	0x01FF_FFFF
Block_RAM/axi_bram_ctrl_0	S_AXI	Mem0	0x0001_0000	32K	0x0001_7FFF
axi_pcie_0	S_AXI_CTL	CTL0	0x0000_1000	4K	0x0000_1FFF
Excluded Address Segments (6)					
Data_SG (32 address bits : 4G)					
Block_RAM/axi_bram_ctrl_0	S_AXI	Mem0	0x0001_0000	32K	0x0001_7FFF
Excluded Address Segments (7)					

The address change must also be done for the axi_cdma_0 peripheral. Expand the Excluded Address Segments.

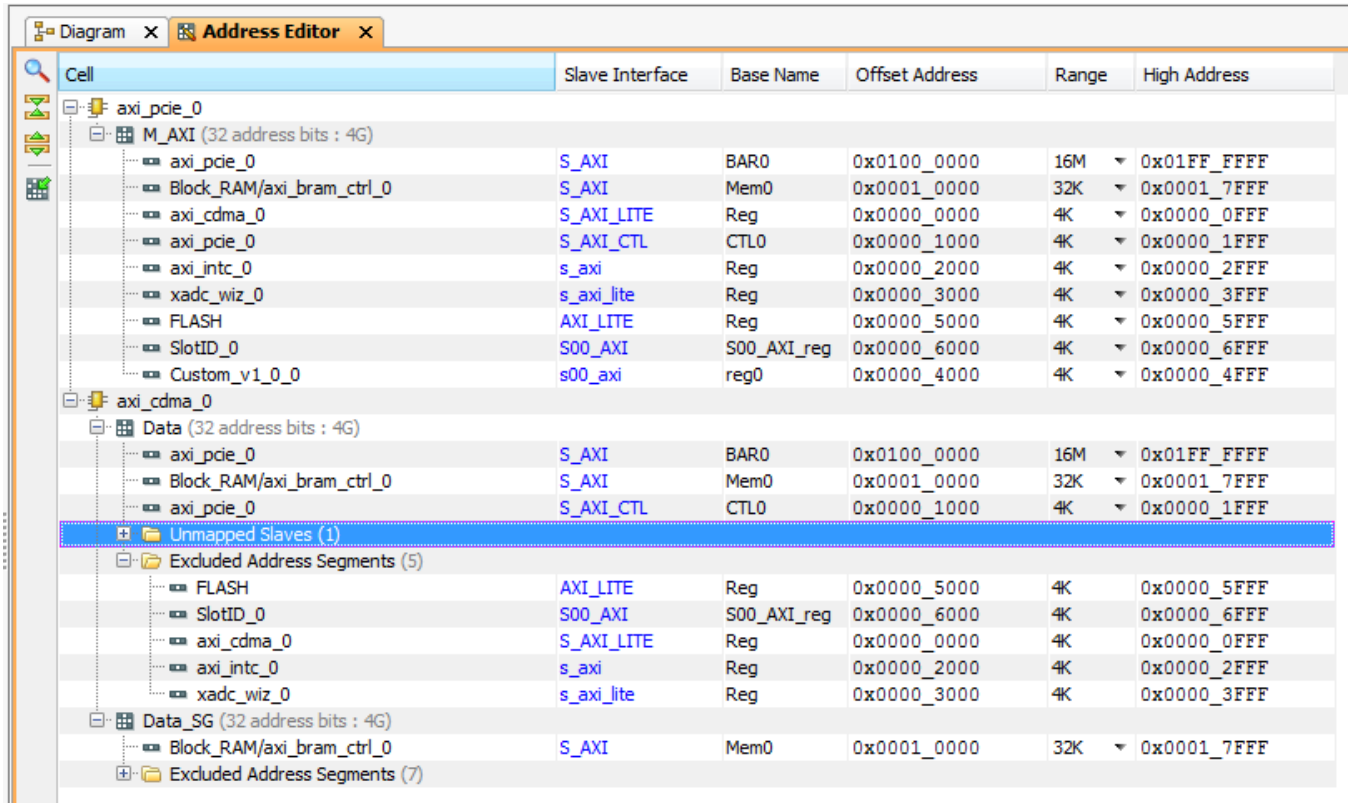


Cell	Slave Interface	Base Name	Offset Address	Range	High Address
axi_pcie_0					
M_AXI (32 address bits : 4G)					
axi_pcie_0	S_AXI	BAR0	0x0100_0000	16M	0x01FF_FFFF
Block_RAM/axi_bram_ctrl_0	S_AXI	Mem0	0x0001_0000	32K	0x0001_7FFF
axi_cdma_0	S_AXI_LITE	Reg	0x0000_0000	4K	0x0000_0FFF
axi_pcie_0	S_AXI_CTL	CTL0	0x0000_1000	4K	0x0000_1FFF
axi_intc_0	s_axi	Reg	0x0000_2000	4K	0x0000_2FFF
xadc_wiz_0	s_axi_lite	Reg	0x0000_3000	4K	0x0000_3FFF
FLASH	AXI_LITE	Reg	0x0000_5000	4K	0x0000_5FFF
SlotID_0	S00_AXI	S00_AXI_reg	0x0000_6000	4K	0x0000_6FFF
Custom_v1_0_0	s00_axi	reg0	0x0000_4000	4K	0x0000_4FFF
axi_cdma_0					
Data (32 address bits : 4G)					
axi_pcie_0	S_AXI	BAR0	0x0100_0000	16M	0x01FF_FFFF
Block_RAM/axi_bram_ctrl_0	S_AXI	Mem0	0x0001_0000	32K	0x0001_7FFF
axi_pcie_0	S_AXI_CTL	CTL0	0x0000_1000	4K	0x0000_1FFF
Excluded Address Segments (6)					
Data_SG (32 address bits : 4G)					
Block_RAM/axi_bram_ctrl_0	S_AXI	Mem0	0x0001_0000	32K	0x0001_7FFF
Excluded Address Segments (7)					

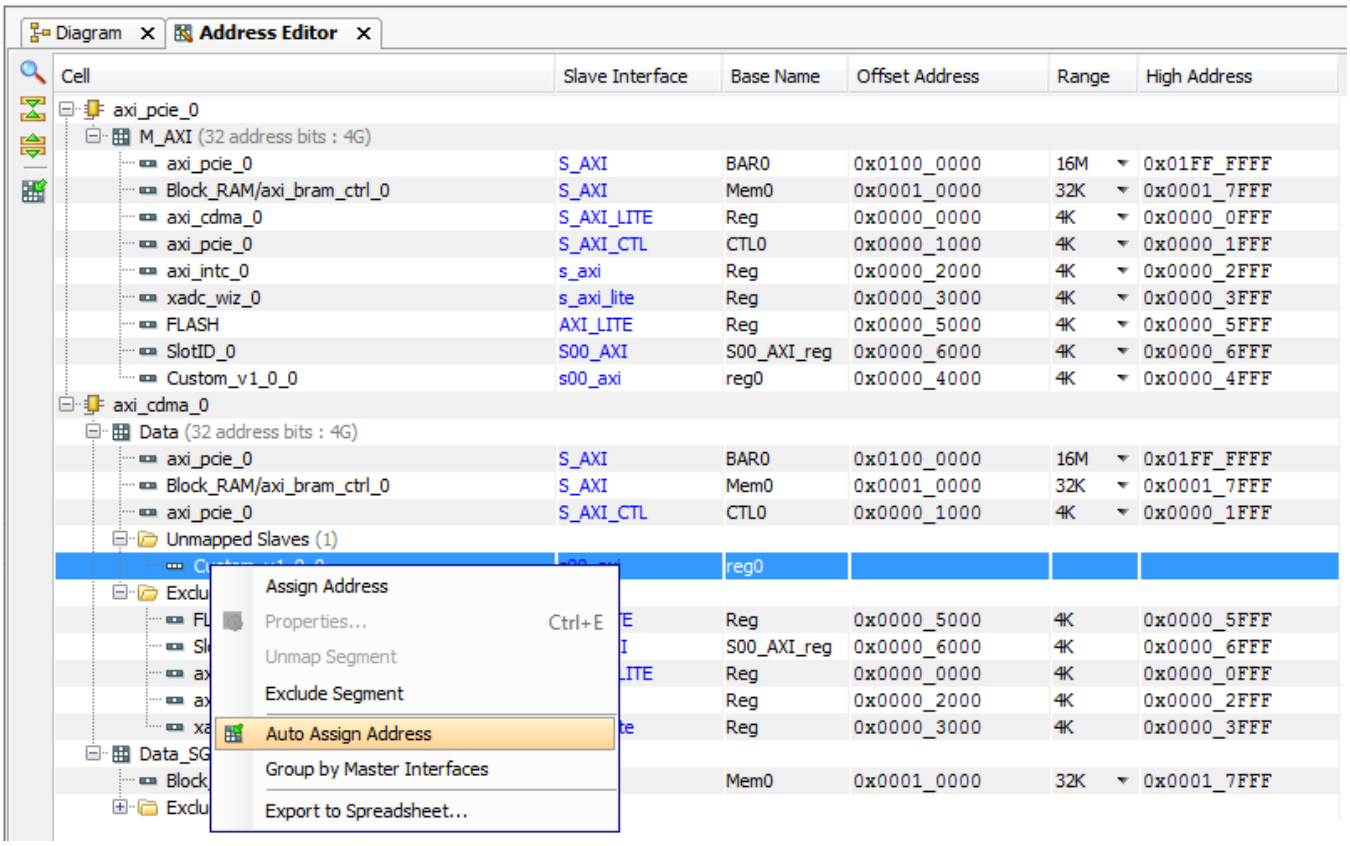
The Vivado software will not allow a change to an Excluded Address Segment, so first we will have to un-map the Custom_v1_0_0 peripheral. Right-click on the Custom_v1_0_0 line under the Excluded Address Segment and select the un-map command from the pop-up menu.



The Custom_v1_0_0 peripheral has now moved to the Unmapped Slaves segment. Expand the Unmapped Slaves Segment.



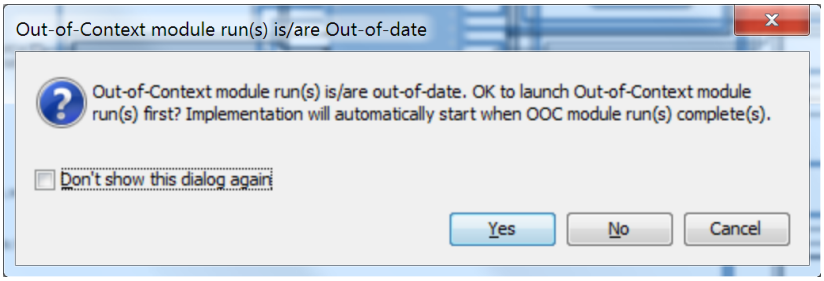
Right-click on the Custom_v1_0_0 peripheral and select the Auto Assign Address command from the pop-up menu. The address assignment for the Custom_v1_0_0 peripheral in the axi_cdma_0 address space will now match that of the axi_pcie_0 address space. The Custom_v1_0_0 peripheral will also return to the Excluded Address Segment. This will prevent an error in the validation process that follows.



Return to the Diagram tab and click on the Validate Design button .

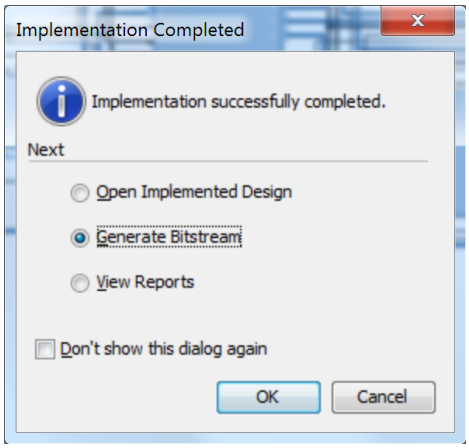
Implement the Modified Design

Click "OK" to dismiss the warning. Click the Run Implementation  icon on the tool bar. Click Yes to launch the Out-of_Context module runs.

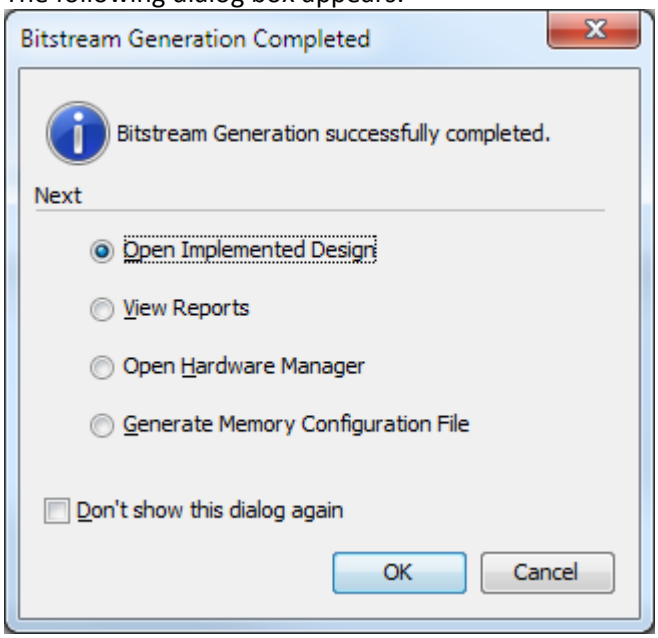


Save the project.

Generate the bitstream.

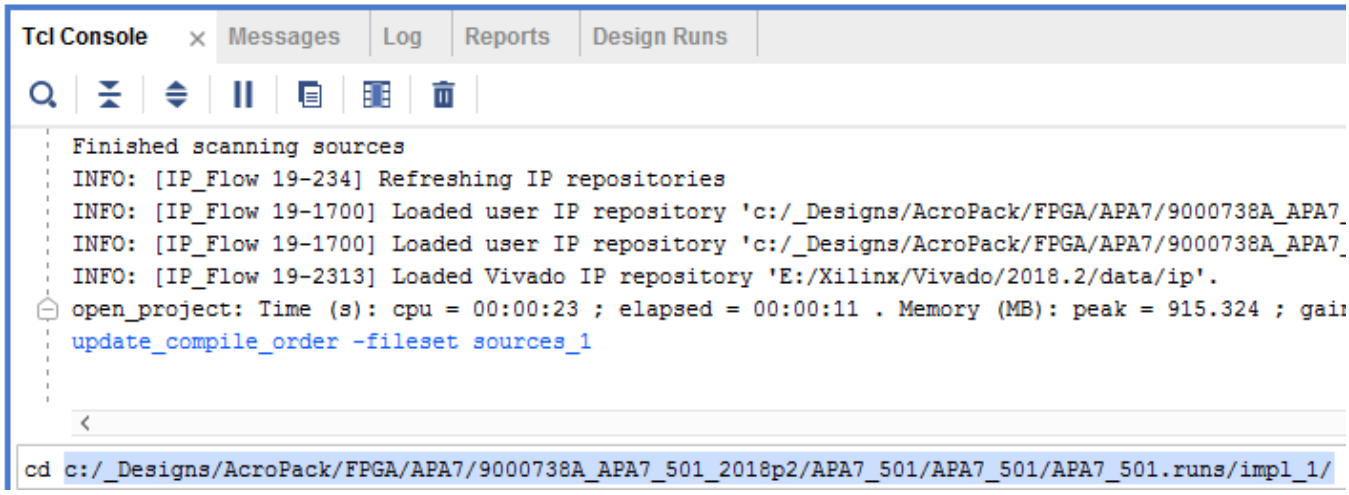


Click "OK" to generate the bitstream.
The following dialog box appears:



Click "Cancel".

Next, run a TCL command to create an ".MCS" file from the ".bit" file. Open the TCL console by clicking on the "TCL Console" tab. Enter the following command in the TCL command bar: "cd c:/_Designs/AcroPack/FPGA/APA7/9000738A_APA7_501_2018p2/APA7_501/APA7_501/APA7_501.runs/impl_1 "



Type the command: write_cfgmem -force -format MCS -size 32 -interface SPIx4 -loadbit "up 0x0 AP750X_top.bit" "APA7-501".

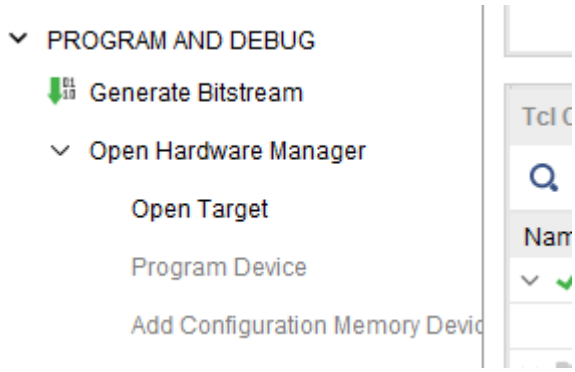
The file "APA7-501.mcs" is created in the folder : c:/_Designs/AcroPack/FPGA/APA7/9000738A_APA7_501_2018p2/APA7_501/APA7_501/APA7_501.runs/impl_1

Write the bitstream to flash.

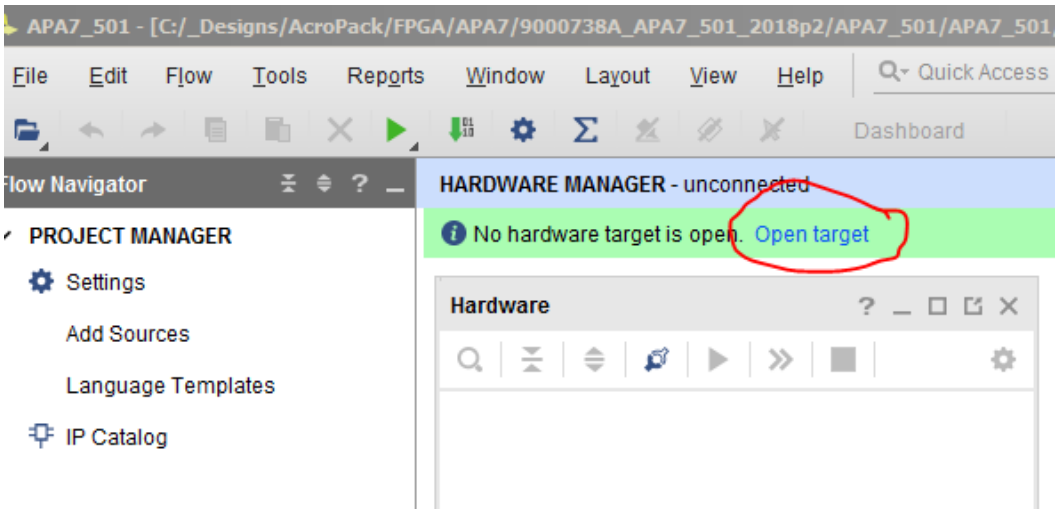
The bitstream is loaded into the FPGA from flash memory on power-up. In order to test the bitstream, it must first be written to flash memory. There are two methods for writing the bitstream to flash memory. You can use Acromag's APA7 Demo program to write the flash, or you can write the flash using the Vivado hardware manager via the JTAG programmer. This example will use the JTAG programmer. Prior to executing the following steps, you must connect a programmer to the carrier's JTAG connector.

Make sure you have a JTAG programmer connected to an Acromag AcroPack carrier with and APA7-50X module installed on it.

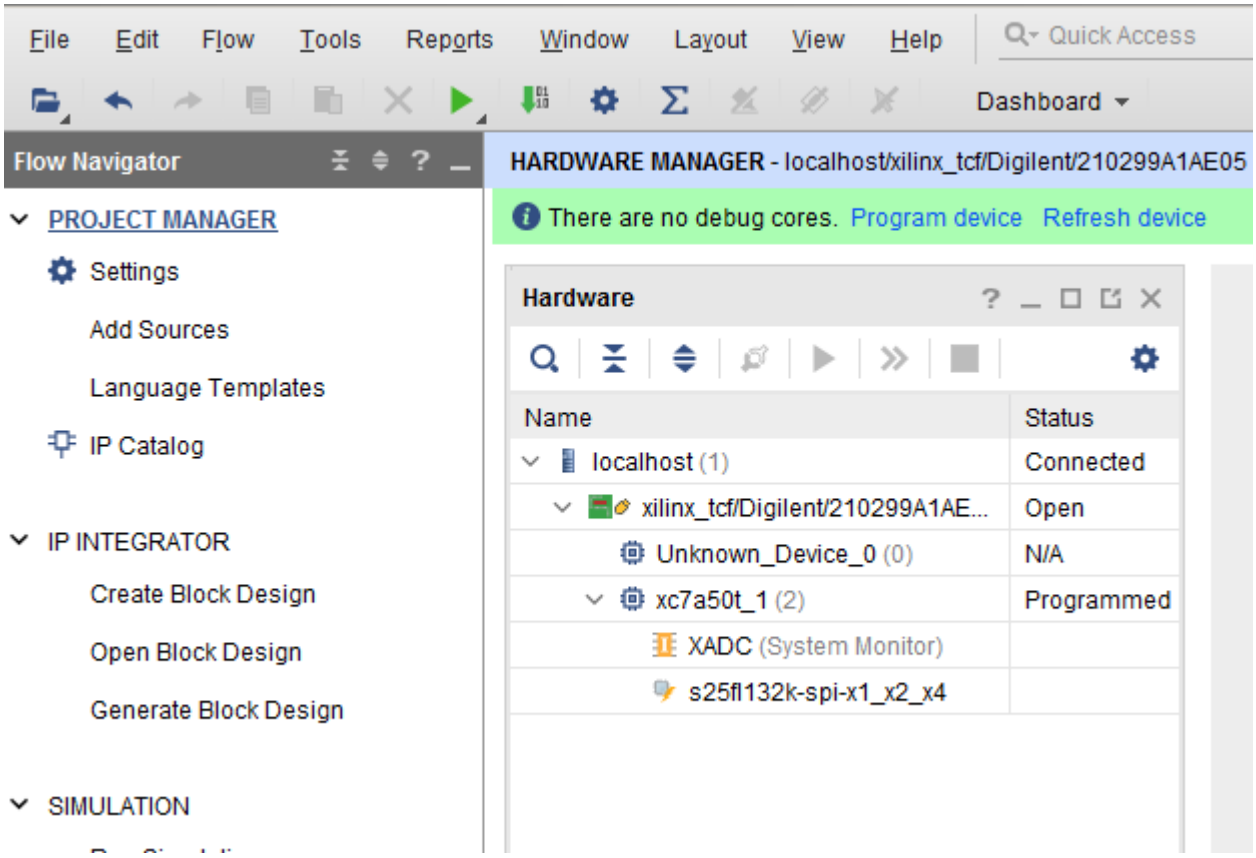
Open the hardware manager by clicking on the "Open Hardware Manager" line under "Program and Dialog" in the "Project Manager" pane.



Then click on "Open Target" => Auto Connect

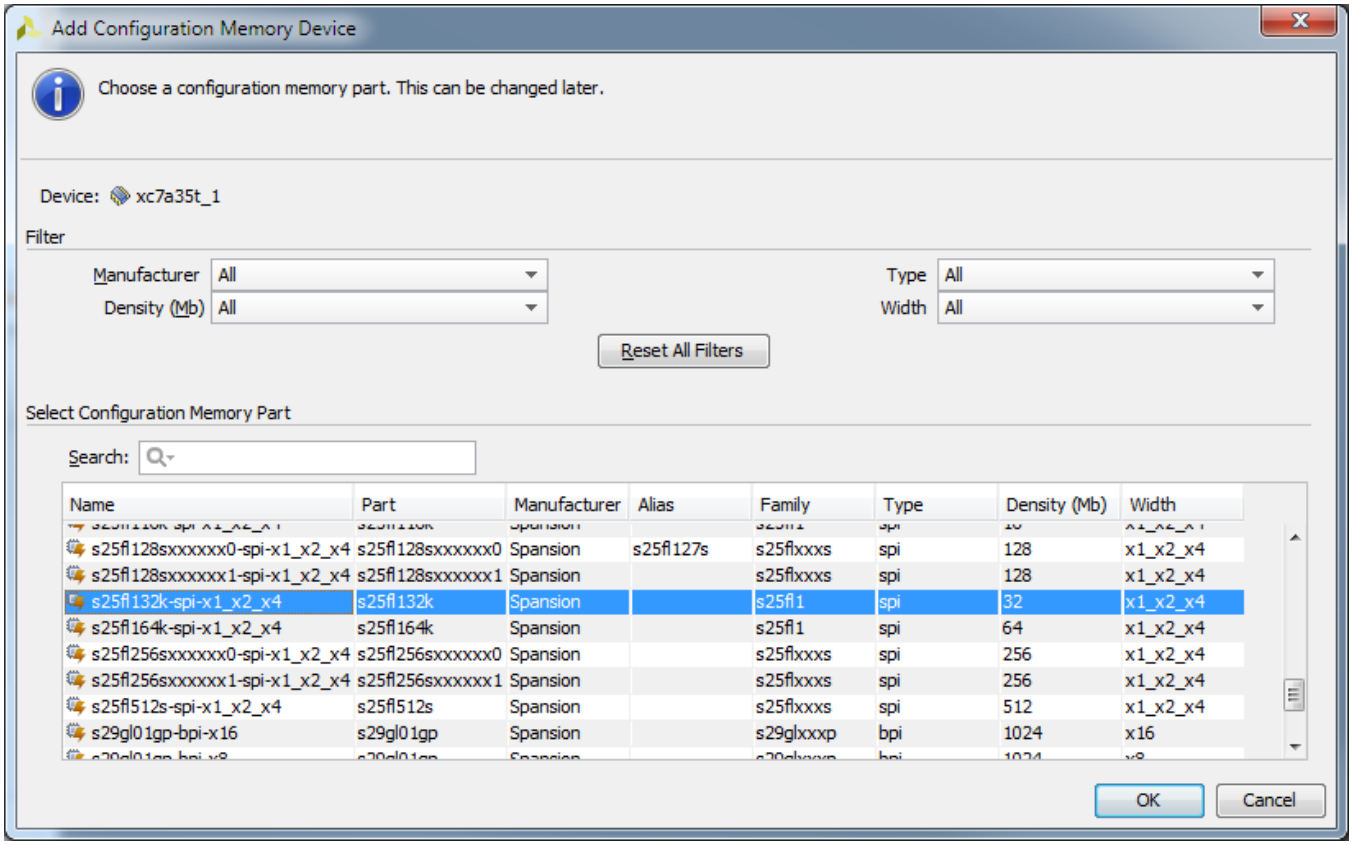


The Vivado hardware manager should detect the presence of the JTAG programmer and the APA7-50X module. The following hierarchy should be displayed in the hardware pane.

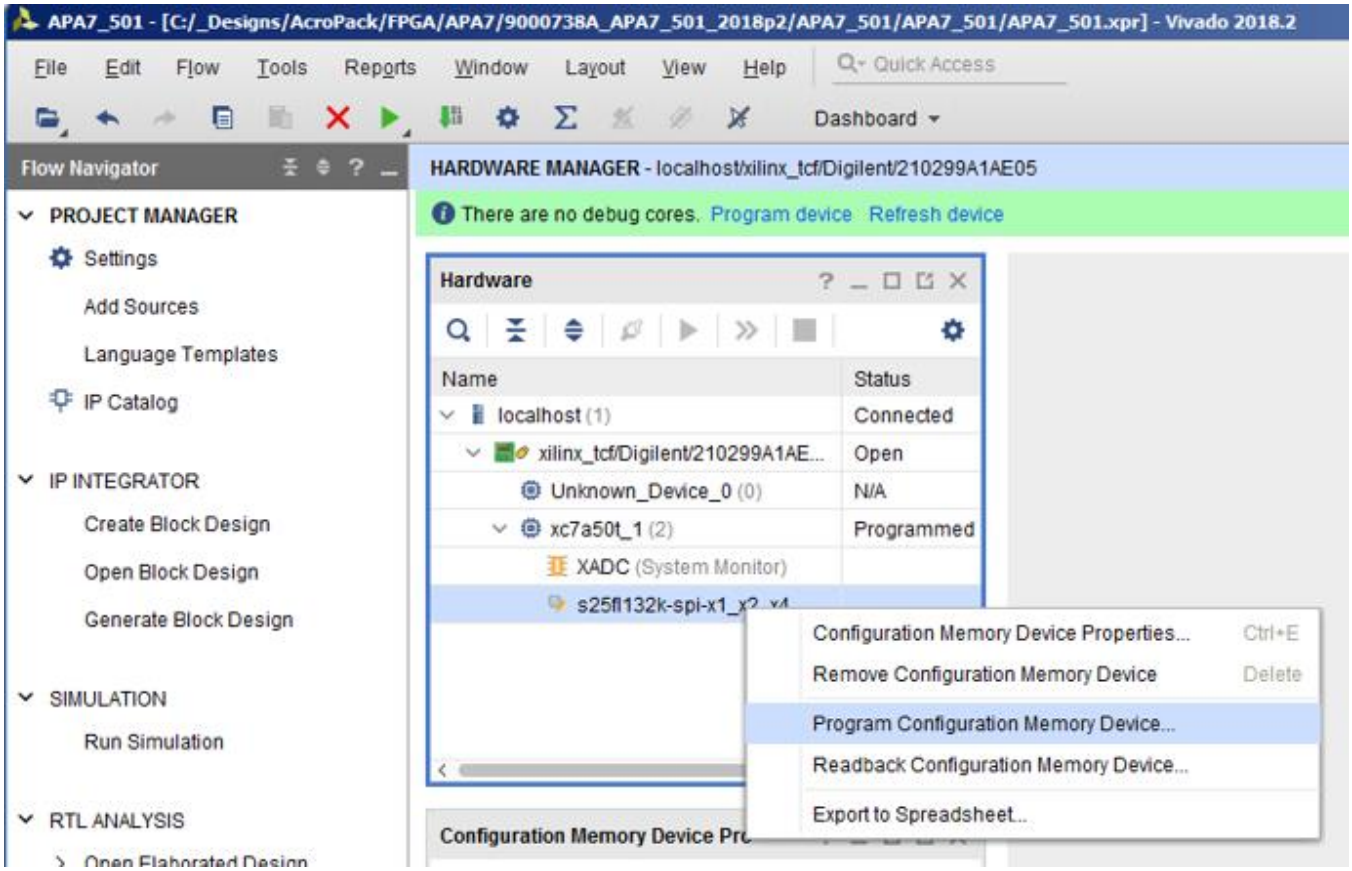


If the s25FL132K-spi-x1 device does not show up below the xc7a50t_1, right-click on the xc7a50t_1 device in the hierarchy and then select the "Add Configuration Memory Device…" command.

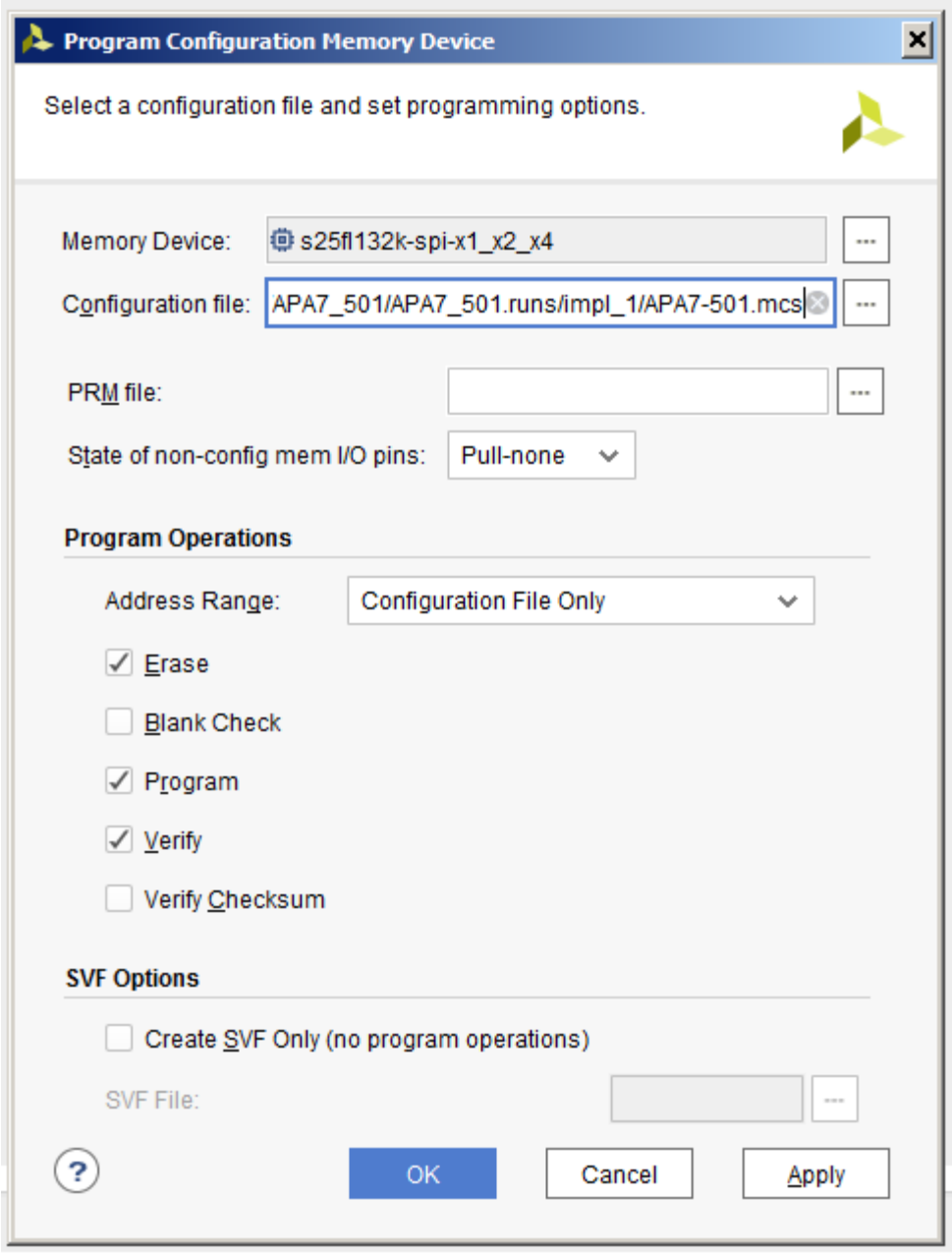
The "Add Configuration Memory Device" dialog box appears. Scroll and select the Spansion 25fl132k part from the list. Click OK.



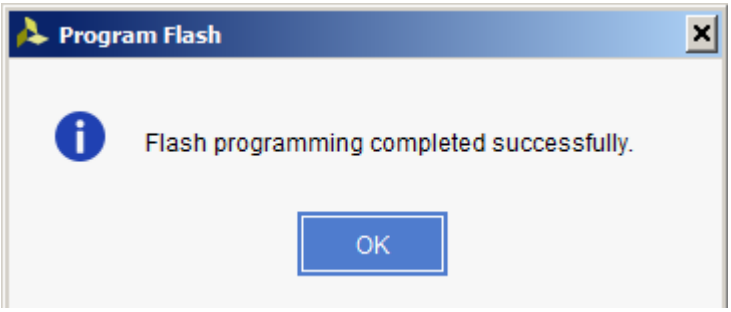
Right mouse select s25fl132k-spi-x1 => Program Configuration Memory Device



Click the ... button on the "Configuration file" line and browse to the location of the configuration file. Selected OK.



The flash erase/program operation will begin.



Select OK.

The flash is now programmed with the revised configuration file. Cycle the power to the FPGA to initiate the FPGA's bitstream load. You may verify the operation of the new firmware using Acromag's APA7Xdemo program.

8. SERVICE AND REPAIR

SERVICE AND REPAIR ASSISTANCE

Surface-Mounted Technology (SMT) boards are generally difficult to repair. It is highly recommended that a non-functioning board be returned to Acromag for repair. The board can be damaged unless special SMT repair and service tools are used. Further, Acromag has automated test equipment that thoroughly checks the performance of each board.

Please refer to Acromag's Service Policy Bulletin or contact Acromag for complete details on how to obtain parts and repair.

PRELIMINARY SERVICE PROCEDURE

Before beginning repair, be sure that all of the procedures in Section 3, Preparation for Use, have been followed. Also, refer to the documentation of your carrier board to verify that it is correctly configured. Replacement of the carrier and/or AP module with one that is known to work correctly is a good technique to isolate a faulty board.

CAUTION: POWER MUST BE TURNED OFF BEFORE REMOVING OR INSERTING BOARDS

WHERE TO GET HELP

If you continue to have problems, your next step should be to visit the Acromag worldwide web site at acromag.com. Our web site contains the most up-to-date product and software information.

For assistance from Acromag's website go to the "Support" tab and submit/search the Acromag forum or fill out the contact us form. You can get additional documentation on the product webpage by using the search feature. If you like to speak with an application engineer:

Phone: 248-295-0310

Email: solutions@acromag.com

9. SPECIFICATIONS

PHYSICAL

Length ..... 70.0 mm (3.150 in)
Width ..... 30.0 mm (1.181 in)
Weight APA7-501..... 7 g
Weight APA7-502..... 7 g
Weight APA7-503..... 7 g
Weight APA7-504..... 7 g

POWER

Power will vary dependent on the application and is limited by the cooling capacity provided by the chassis the module and carrier are installed in. Acromag carriers are designed to provide a maximum of 1.1 Amps on the 3.3 Volt supply. Power values are given for the Acromag example design.

APA7-501E-LF
+3.3 Volts ..... 0.5 A typical, 1 A max.

APA7-502 E-LF
+3.3 Volts ..... 0.5 A typical, 1 A max.

APA7-503 E-LF
+3.3 Volts ..... 0.5 A typical, 1 A max.

APA7-504 E-LF
+3.3 Volts ..... 0.5 A typical, 1 A max.

PCIe BUS COMPLIANCE

Specification..... This device meets or exceeds all written PCI Express specifications per revision 2.1 dated March 4, 2009.

PCIe Bus Data Rates

PCIe Gen 1 (1 lane)	Giga bit / second	Bytes / second
Signaling Rate	2.5 Gb/s	312 Mbyte/s
Ideal Rate[1]	2 Gb/s	250 Mbyte/s
Header Burden plus 4byte Sample Rate[2]	0.332 Gb/s	41.6 Mbyte/s
Actual Measured 4 Byte Read Rate[3]	0.019 Gb/s	2.35 Mbytes/s
Actual Measured DMA 4 Byte Read Rate[4]	0.327 Gb/s	40.9 Mbyte/s
Actual Measured 4 Byte Write Rate[5]	0.320 Gb/s	40 Mbyte/s

Note 1: PCIe x1 Gen 1 = 2.5GT/s (with 10-bit encoding we have a 20% loss in possible throughput due to encoding) giving 2.0 G bits/sec or 250M Bytes/sec.

Note 2: With PCIe we have a header for address and read/write command that is sent with every packet. This header is 20 Bytes with data payload of 4 Bytes (for our typical AcroPack). For each 4 Byte data sample, 24 Bytes are sent.

$\frac{250MByte/s}{24\,Bytes}$ = 10.4 M samples/sec or 41.6 M Bytes/sec or 0.332 G bit/sec

Note 3: For our typical AcroPack have measured back to back 4 Byte read operations completing every 1.7usec. A read operation starts with a host read request. The AcroPack must process the read and fetch the data and then generated the completion back to the host. The host then sends a message back that says I got it. This back and forth hand shaking slows down individual reads.

Note 4: DMA Read of 1024 sample takes 100us. Each sample is 4 Bytes. 100us/1024=0.0977us per sample or 4/0.0977us = 40.94Mbyte/s. We use DMA transfers to improve data transfers on the AP341/2 and AP225/235.

Note 5: Simple write operations are just as fast as DMA read operations. Write data is presented to the AcroPack in one transaction. Measured 4-byte back to back write accesses taking place every 100ns.

ENVIRONMENTAL

Operating Temperature ..... APA7-50X -40°C to +80°C
Applications requiring operating temperatures above *70*°C will require purchase of *AcroPack Heatsink Accessory* ***AP-CC-01.***
Relative Humidity..... 5-95% non-condensing
Storage Temperature..... -55 to +125°C

Non-Isolated..... The PCIe bus and the AcroPack module commons have a direct electrical connection.
EMC Directives ..... AcroPack complies with EMC Directive 2014/30/EC.
Immunity per EN 61000-6-2..... This is a Class A product
Electrostatic Discharge Immunity (ESD) per IEC 61000-4-2.
Radiated Field Immunity (RFI)..... per IEC 61000-4-3.
Electrical Fast Transient Immunity (EFT) per IEC 61000-4-4
Surge Immunity..... per IEC 61000-4-5.
Conducted RF Immunity (CRFI)..... per IEC 61000-4-6.

Emissions per EN 61000-6-4: ..... This is a Class A product
Enclosure Port..... per CISPR 16
Low Voltage AC Mains Port..... per CISPR 16.

RoHS Directive 2011/65/EU – All Models
In compliance per EN 50581.

Vibration and Shock..... This AcroPack is designed to comply with the following Vibration and Shock standards:
Vibration, Sinusoidal Operating: . Designed to comply with IEC 60068-2-6: 10-500Hz, 5G, 2 Hours/axis
Vibration, Random Operating: .... Designed to comply with IEC 60068-2-64: 10-500Hz, 5G-rms, 2 Hours/axis.
Shock, Operating: ..... Designed to comply with IEC 60068-2-27: 30G, 11ms half sine, 50G, 3mS half sine, 18 shocks at 6 orientations for both test levels.

Reliability Prediction

All APA7-50X models can use the APA7-201E-LF MTBF Report.

MTBF (Mean Time Between Failure): MTBF in hours using MIL-HDBK-217F, FN2. *Per MIL-HDBK-217, Ground Benign, Controlled,* $G_B G_C$

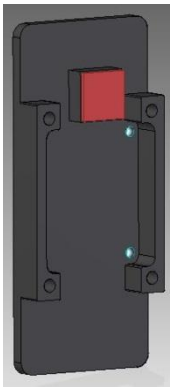
Temperature	MTBF (Hours)	MTBF (Years)	Failure Rate (FIT[1])
25°C	1,902,262	217.2	525.7
40°C	1,083,259	123.7	923.1

[1] FIT is Failures in 10^9 hours.

FPGA	**Xilinx XC7A50T-1CPG236I** (I for Extended Temperature) • 52,160 Logic Cells • 600,000 Distributed RAM Bits • 2,700 Kbit Block RAMs • 120 DSP 48E Slices • 5 MMCM and 5 PPL
TTL Input and Output Electrical Characteristics	• VIH: 2.0V minimum • VIL: 0.8V maximum • IOH: –32.0mA • IOL: 64mA • VOH: 2.0V minimum • VOL: 0.55V maximum at 64mA
Digital Propagation Delay	• Driver/Receiver Input to Output Delay = 3.5ns Maximum
TTL Channel Pull-up Resistors	• 4.7KΩ pull-up resistor is installed for each TTL Output/Input.
RS485/RS422 Electrical Characteristics	• 3 V Max.: Driver Common Mode Output Voltage • –7V to 12V Common Mode Input/Output Voltage Range • Driver Differential Vout = 1.5V Minimum, RL = 54Ω, • 100mV Typical: Input Hysteresis • 96KΩ Minimum Input Resistance • Data Rates up to 20Mbps
Differential Propagation Delay	• Driver Input to Output Delay = 27ns Typical, 40ns Maximum • Receiver Input to Output Delay = 33ns Typical, 60ns Maximum
RS485/RS422	120Ω termination resistor are not installed but can be added with installation in the requested open resistor locations.
LVDS I/O Electrical Characteristics	• 247mV Min., 454mV Max.: LVDS Driver Output Voltage with 50Ω load. • 1.37 V Max.: Common Mode Output Voltage. • -50 mV Min to +50mV Max: LVDS Input Threshold Voltage • Compatible with either LVDS TIA/EIA-644 or M-LVDS TIA/EIA-899 for Multipoint Data Interchange
LVDS Propagation Delay	• Driver Propagation Delay Time = 2.7ns Maximum • Driver Output Signal Transition Time = 1.0ns Maximum • Receiver Propagation Delay Time = 4.5ns Maximum • Receiver Output Signal Transition Time = 1.5ns Maximum
Maximum Data Rate	• Maximum Data Rate: 100MHz with 1 meter enter connect cable
Termination Resistors	• **Termination Resistors**: Non-removable 100Ω termination resistors are in place for each of the LVDS channels.

Appendix A

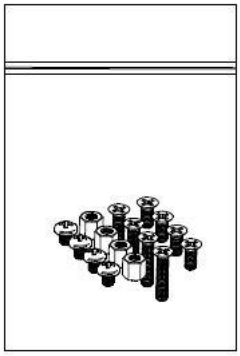
AP-CC-01 Heatsink Kit Installation



Bottom view



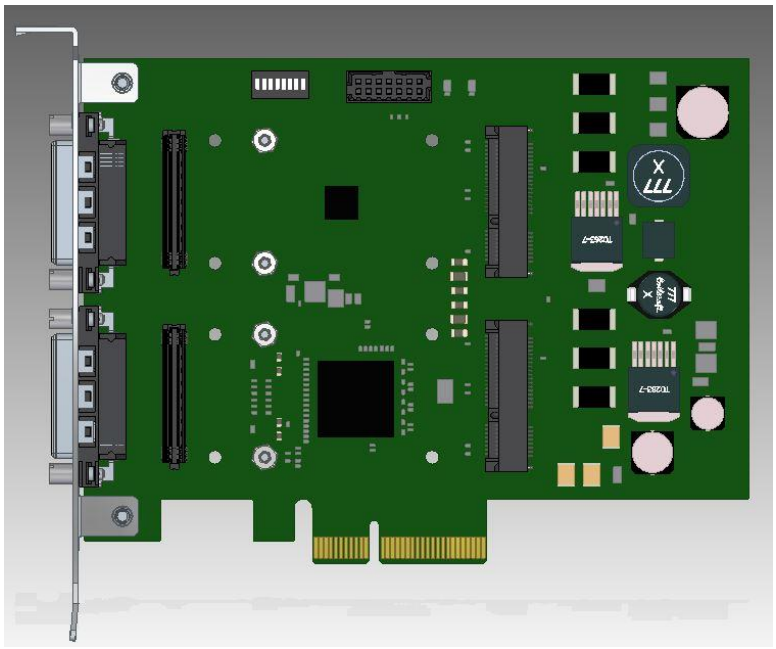
Top view



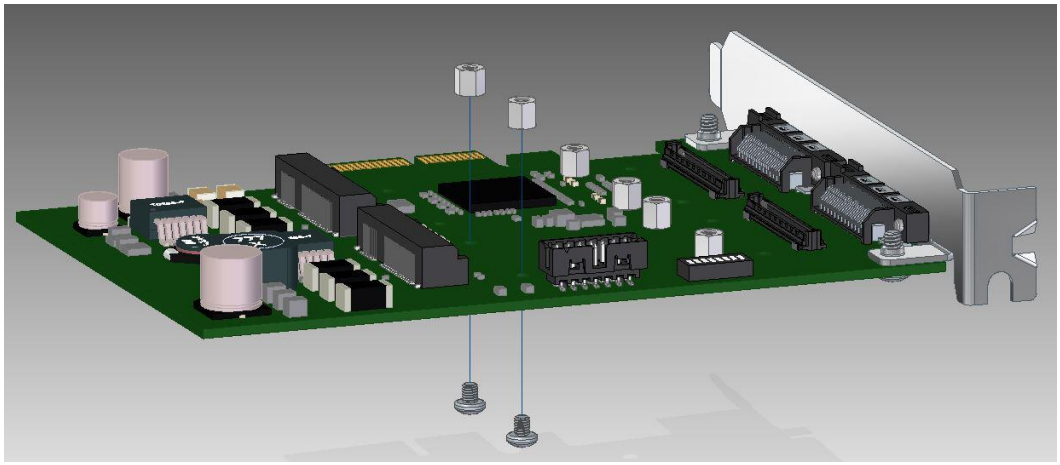
Hardware

AP-CC-01 Heat Sink Kit

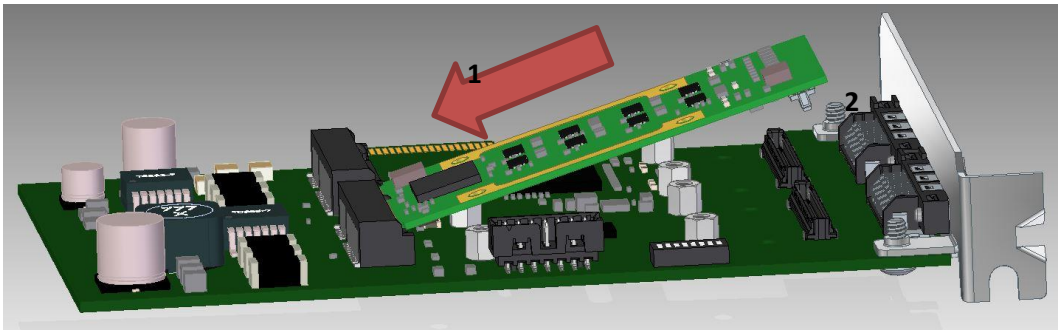
This example will show how to install the AP-CC-01 Heatsink kit with an APCe7020 carrier.



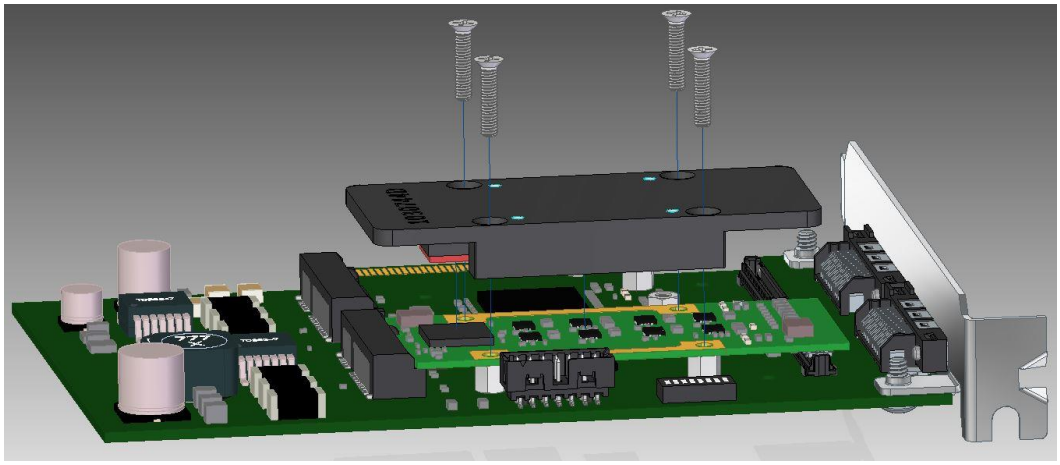
1. Install two standoffs and secure with two screws.



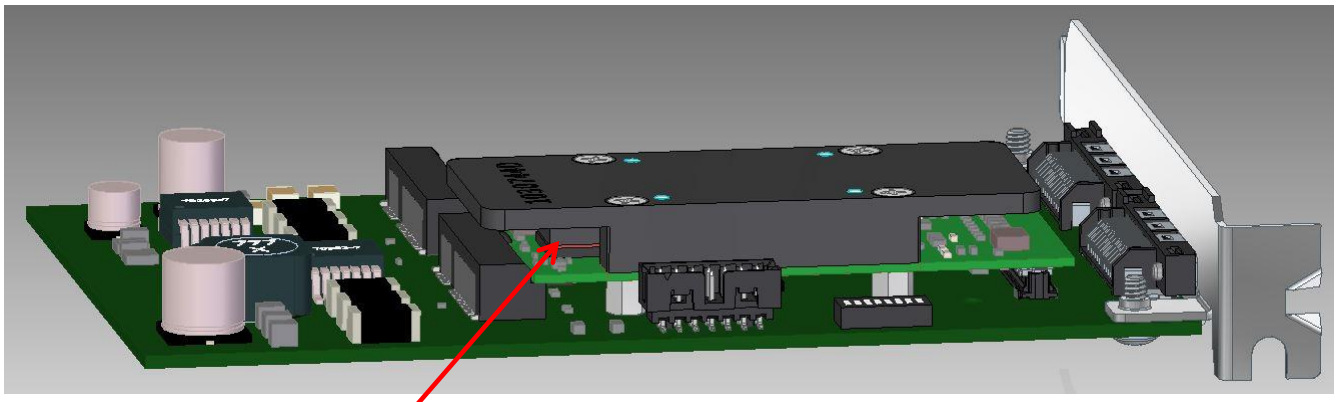
2. Install the AcroPack module.



3. Install the Heatsink and secure with 4 screws.



4. AP-CC-01 Installation is complete.



Note: Make sure the thermal pad is making contact with the FPGA IC.

Certificate of Volatility

<table>
<tr><th colspan="5">Certificate of Volatility</th></tr>
<tr><td colspan="2">Acromag Models:
APA7-501E-LF
APA7-502E-LF
APA7-503E-LF
APA7-504E-LF</td><td colspan="3">Manufacturer:
Acromag, Inc.
30765 Wixom Rd
Wixom, MI 48393</td></tr>
<tr><th colspan="5">Volatile Memory</th></tr>
<tr><td colspan="5">Does this product contain Volatile memory (i.e. Memory of whose contents are lost when power is removed)
■ Yes □ No</td></tr>
<tr><td>Type (SRAM, SDRAM, etc.)
FPGA based RAM</td><td>Size:
50 x 36-Kbit</td><td>User Modifiable
■ Yes
□ No</td><td>Function:
Data storage for FPGA</td><td>Process to Sanitize:
Power Down</td></tr>
<tr><th colspan="5">Non-Volatile Memory</th></tr>
<tr><td colspan="5">Does this product contain Non-Volatile memory (i.e. Memory of whose contents is retained when power is removed)
■ Yes □ No</td></tr>
<tr><td>Type (EEPROM, Flash, etc.)
Flash</td><td>Size:
32Mbyte</td><td>User Modifiable
■ Yes
□ No</td><td>Function:
Storage of Code for FPGA</td><td>Process to Sanitize:
Clear Flash memory by erasing all sectors of the Flash</td></tr>
<tr><td>Type (EEPROM, Flash, etc.)
One Time Programmable area in flash device</td><td>Size:
272 bytes</td><td>User Modifiable
□ Yes
■ No</td><td>Function:
The OTP area has been disabled by writing the lock registers with zeroes.</td><td>Process to Sanitize:
Not applicable</td></tr>
<tr><th colspan="5">Acromag Representative</th></tr>
<tr><td>Name:
Russ Nieves</td><td>Title:
Dir. of Sales and Marketing</td><td>Email:
rnieves@acromag.com</td><td>Office Phone:
248-295-0823</td><td>Office Fax:
248-624-9234</td></tr>
</table>

10. Revision History

Release Date	Version	EGR/DOC	Description of Revision
01 NOV 2018	A	LMP/ARP	Initial release.
04 JAN 2019	B	LMP/ARP	Removed reference to original APA7-20x models, and updated Serial Flash Write/Read Example on page 49.
13 SEP 2019	C	LMP/ARP	Added Appendix A on page 96.

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP