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 XMC-6VLXF* at our website: ***Click HERE***

XMC-6VLX Conduction Cooled Virtex-6 Based FPGA XMC Module

USER'S MANUAL

ACROMAG INCORPORATED
30765 South Wixom Road
P.O. BOX 437
Wixom, MI 48393-7037 U.S.A.
Tel: (248) 295-0310

Copyright 2012, Acromag, Inc., Printed in the USA.
Data and specifications are subject to change without notice.
8500-931C

Table of Contents

1.0 GENERAL INFORMATION 5

Ordering Information 6

Key Features 6

PCIe Interface Features 7

Software 8

ENGINEERING DESIGN KIT 8
BOARD DLL CONTROL SOFTWARE 8
BOARD VxWORKS SOFTWARE 8
BOARD Linux SOFTWARE 8

2.0 PREPARATION FOR USE 9

Unpacking and Inspecting 9

Card Cage Considerations 9

Heat Sink Considerations 9

Board Installation 9
Default Hardware Configuration 10
P16 Secondary XMC Connector 10
Rear P4 Field I/O Connector 11
Mezzanine Connector 12

Non-Isolation Considerations 13

3.0 PROGRAMMING INFORMATION 14

GETTING STARTED 14

Virtex 6 Configuration 15
Platform Flash Xilinx Configuration 15
BPI Flash Xilinx Configuration 15

PCIe CONFIGURATION ADDRESS SPACE 16

CONFIGURATION REGISTERS 16

BAR0 MEMORY MAP 18

INTERRUPT CONTROLLER 19

Interrupt Status Register (Read/Write) - (BAR0 + 0x00100000) 19
Interrupt Pending Register (Read) - (BAR0 + 0x00100004) 20
Interrupt Enable Register (Read/Write) - (BAR0 + 0x00100008) 20
Interrupt Acknowledge Register (Write) - (BAR0 + 0x0010000C) 21
Set Interrupt Enable Register (Write) - (BAR0 + 0x00100010) 21
Clear Interrupt Enable Register (Write) - (BAR0 + 0x00100014) 22
Interrupt Vector Register (Read) - (BAR0 + 0x00100018) 22
Master Enable Register (Read/Write) - (BAR0 + 0x0010001C) 22

AXI-CDMA 23
CDMA Control Register (Read/Write) - (BAR0 + 0x000A0000) 24
CDMA Status Register (Read/Write) - (BAR0 + 0x000A0004) 26
CDMA Current Descriptor Pointer Register (Read/Write) - (BAR0 + 0x000A0008) 29
CDMA Tail Descriptor Pointer Register (Read/Write) - (BAR0 + 0x000A0010) 30
CDMA Source Address Register (Read/Write) - (BAR0 + 0x000A0018) 31
CDMA Destination Address Register (Read/Write) - (BAR0 + 0x000A0020) 31
CDMA Bytes to Transfer Register (Read/Write) - (BAR0 + 0x000A0028) 32
Simple CDMA Programming Example 32

AXI-BAR0 Aperture Base Address 33

PCIe AXI-Bridge Control 34
Physical Side Interface Status/Control Register (Read/Write) - (BAR0 + 0x000F0144) 34
AXI Base Address Translation Configuration Register (Read Only) - (BAR0 + 0xF0208/0xF020C) 35

FPGA Fabric MEMORY MAP 36
Front, Rear, and P16 I/O Registers (Read/Write) – (BAR0 + 0x301000 to 0x 301FFF) 37
Front I/O Interrupt Status/Clear Register (Read/Write) - (BAR0 + 0x300000) 37
DDR Memory Test Status Register (Read/Write) - (BAR0 + 0x300008) 38
XMC Board Identification Code Register (Read Only) - (BAR0 + 0x30000C) 38
Configuration Control (Read/Write) – (BAR0 + 0x300100) 38
Aurora Monitor (Read/Write) – (BAR0 + 0x300104) 39
Flash Introduction 39
Flash Status (Read Only) – (BAR0 + 0x300200) 41
Flash Control (Write Only) – (BAR0 + 0x300204) 42
Flash Read (Read Only) – (BAR0 + 0x300208) 42
Flash Start Write (Write Only) – (BAR0 + 0x30020C) 43
Flash Erase Block (Write Only) – (BAR0 + 0x300210) 43
Flash Data Register (Read/Write) – (BAR0 + 0x300214) 43
Flash Address (Read/Write) – (BAR0 + 0x300218) 43
Simple BPI Flash Programming Example 44
Simple Platform Flash Programming Example 44
System Monitor Status/Control Register (Read/Write) – (BAR0 + 0x300300) 45
System Monitor Address Register (Write Only) – (BAR0 + 0x300304) 45
Front Input Data Register (Read Only) - (BAR0 + 0x301000) 46

Front Output Data Register (Read/Write) - (BAR0 + 0x301004) 46
Front I/O Interrupt Enable Register (Read/Write) - (BAR0 + 0x301008) 47
Interrupt Type (COS or H/L) Configuration Register (Read/Write) - (BAR0 + 0x30100C) 47
Interrupt Polarity Register (Read/Write) - (BAR0 + 0x301010) 48
Rear Input Data Register (Read Only) - (BAR0 + 0x301100) 49
Rear Output Data Register (Read/Write) - (BAR0 + 0x301104) 50
P16 Input Data Register (Read Only) - (BAR0 + 0x301200) 51
P16 Output Data Register (Write Only) - (BAR0 + 0x301204) 52

BAR2 MEMORY MAP 53
QDR Memory (Read/Write) – (BAR2 + 0x0000000 to 0x00FFFFFF) 53

4.0 THEORY OF OPERATION 54
PCI INTERFACE LOGIC 55
DDR3 Memory 55
QDR II+ SRAM Memory 55
Clock Generation 56
Multi-Gigabit Transceivers (GTX MGTs) 56
SFP Module Connectors 56
USB-to-UART Bridge 56
16MB Platform Flash 57
32MB Linear BPI Flash 57
Configuration Flash Design Considerations 58
JTAG Port 58
DIP Switch 59
Power System Devices 59
System Monitor 61

5.0 XPS EMBEDDED SYSTEM 62
Xilinx ISE Example Projects 62
Microblaze Peripherals 64
Host Peripherals 65
Common Peripherals 65
SDK 65
Acromag Peripheral Repository 67
Acromag modified library files 68
Running a Program from BPI FLASH Memory 69
Running LWIP Echo Server 72
EDK File Organization 74
Example EDK Design Modification Walkthrough 74

6.0 SERVICE AND REPAIR 101

Service and Repair Assistance 101

Preliminary Service Procedure 101

Where to Get Help 101

7.0 SPECIFICATIONS 102

PHYSICAL 102
POWER REQUIREMENTS 102
ENVIRONMENTAL 102
User Programmable (U2) FPGA 103
EDK Example Design 103
P15 Connector 103
P16 Connector 104
P4 REAR I/O 104
MEZZANINE FRONT I/O 104
Board Oscillators 105
DDR3 Memory 105
QDR II+ SRAM Memory 105
16MB Platform Flash 106
32MB Linear BPI Flash 106
PCIe Bus Interface 106

XMC-6VLX Block Diagram 108

Certificate of Volatility 109

REVISION HISTORY 110

All trademarks are the property of their respective owners.

IMPORTANT SAFETY CONSIDERATIONS

You must consider the possible negative effects of power, wiring, component, sensor, or software failure in the design of any type of control or monitoring system. This is very important where property loss or human life is involved. It is important that you perform satisfactory overall system design and it is agreed between you and Acromag, that this is your responsibility.

The information of this manual may change without notice. 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, Inc.

RELATED PUBLICATIONS

The following manuals and part specifications provide the necessary information for in depth understanding of the XMC-6VLX board.

Virtex-6 Data Book Spec	**http://www.xilinx.com**
CY7C1565KV18-400BZC Spec.	**http://www.cypress.com**
MT41J128M16HA-15EIT Spec.	**http://www.micron.com**

1.0 GENERAL INFORMATION

This XMC-6VLX is an XMC module with the heart of the design being the Virtex 6 reprogrammable FPGA. The re-configurable XMC-6VLX modules use the Xilinx Virtex 6 XC6VLX FPGA. Re-configuration of the FPGA is possible via a direct download into the Xilinx Platform Flash over the PCIe bus. The on board Platform Flash memory loaded with FPGA configuration data allows automatic Xilinx configuration on power-up.

The XMC-6VLX is an XMC module with the following interfaces. Eight high speed serial lanes are allocated to the XMC P15 connector. These lanes can be used for an 8 lane PCIe (PCI Express) implementation, Serial RapidIO, or 10 Gigabit Ethernet. The example design will support a 4 lane Gen 1 PCIe implementation with one DMA channel for data transfer between PCIe and on board QDRII memory.

Eight high speed serial lanes are also allocated to the XMC P16 connector. These eight serial lanes can be used for Serial RapidIO, PCIe, 10 Gigabit Ethernet, or Xilinx Aurora. The example design will support an 8 lane Aurora implementation for use of these lanes. Two Virtex 6 global clocks and 34 select I/O signals will also be provided on the P16 connector. Select I/O signals are 2.5V Virtex 6 I/O pins that can be selected from single-ended I/O standards (LVCMOS, HSTL, and SSTL) and differential I/O standards (LVDS, HT, LVPECL, BLVDS, Differential HSTL and SSTL).

One P4 rear I/O connector will provide two global clock differential pairs, and 30 LVDS signal pairs.

One mezzanine connector will provide JTAG, USB signals, two global differential clock pairs, 11 LVDS signal pairs, and two ground signals. Two high speed serial interfaces are also routed from the FPGA to this mezzanine module connector.

The board will provide 2 Meg x 72-bit QDRII SRAM, 128 Meg x 64-bit DDR3 SDRAM, 4 Meg x 16-bit parallel Flash, and Xilinx 128 Megabit Platform Flash. The parallel Flash will interface to the FPGA for MicroBlaze CPU program

code storage. The Xilinx 128 Megabit Platform Flash will contain the power up configuration bit file for the Virtex 6 FPGA.

Ordering Information

The following table lists the orderable models and their corresponding operating temperature range.

Models XMC-6VLX240 and XMC-6VLX365 are conduction-cooled without front I/O.

Table 1.1: *The XMC-6VLX boards are available in the standard temperature range.*

MODEL	OPERATING TEMPERATURE RANGE
XMC-6VLX240	0°C to +70°C
XMC-6VLX365	0°C to +70°C

Key Features

An XMC-6VLX block diagram, found at the end of this manual, illustrates the key features listed below.

- **Reconfigurable Xilinx FPGA –** In system configuration of the FPGA is performed through flash configuration. The PCIe bus can be used to change the flash configuration memory. This provides a means for creating custom user defined designs. The Virtex 6 will configure from the updated Platform flash on the next power cycle.
- **QDRII SRAM –** Provides 2 Meg x 72-bit QDRII SRAM. SRAM is linked to the Virtex 6 device for PCIe bus, and DMA engine access. It can also be linked to front, rear, and high speed data path access.
- **DDR3 SDRAM –** Provides 128 Meg x 64-bit DDR3 SDRAM. SDRAM is linked to the Virtex 6 device for MicroBlaze and Ethernet access.
- **P15 High Speed Interface –** The Eight high speed serial lanes are allocated to the XMC P15 connector. These lanes can be used for an 8 lane PCIe (PCI Express) implementation, Serial RapidIO, or 10 Gigabit Ethernet. The example design will support a 4 lane Gen 1 PCIe implementation.
- **P16 High Speed Interface –** The Eight high speed serial lanes are allocated to the XMC P16 connector. These lanes can be used for Serial RapidIO, PCIe, 10 Gigabit Ethernet, or Xilinx Aurora. The example design will support an 8 lane Aurora loopback implementation.
- **Interface to Rear P4 Connector –** The Virtex 6 FPGA is directly connected to 64 pins of the rear P4 connector. All 2.5volt IO standards supported by the Virtex 6 device are available. The example design

provides LVCMOS single ended signaling.

- **Interface to Mezzanine Connector –** The Virtex 6 FPGA is directly connected to a front mezzanine connector. All 2.5volt IO standards supported by the Virtex 6 device are available on 13 signal pairs. JTAG for Virtex 6 configuration and use with Xilinx ChipScope FPGA signal analysis tool. USB is provides a MicroBlaze debug terminal port. Two high speed serial interfaces are also routed from the FPGA to this mezzanine module connector.
- **Example Design Provided –** The example VHDL design includes implementation of the PCIe bus 4 lane Gen 1, control of digital front and rear I/O, and QDRII read/write interface logic, SFP module interface for 1 Gig Ethernet with DDR3 Memory.

PCIe Interface Features

- **PCIe Bus –** An example design is provided with four lane PCI Express Generation 1 operating at a bus speed of 2.5 Gbps per lane per direction. This gives up to 2GBytes/sec data rate on the bus. FPGA supports Gen1 8-lane (x8) or Gen2 4-lane (x4).
- **PCIe Bus Master –** The PCIe interface logic becomes the bus master to perform DMA transfers.
- **DMA Operation –** The PCIe bus interface supports one DMA channel capable of transferring data to and from the on board QDRII SRAM.
- **Compatibility** – PCI Express Base Specification v2.0 compliant PCI Express Endpoint. Provides one multifunction interrupt. The XMC-6VLX is compatible with XMC VITA 42.3 specification for P15.

Software

The XMC-6VLX products will require support drivers specific to your operating system.

ENGINEERING DESIGN KIT

Acromag provides an engineering design kit for the 6VLX boards (sold separately), a "must buy" for first time 6VLX module purchasers. The design kit (model XMC-6VLX-EDK) provides the user with the basic information required to develop a custom FPGA program for download to the Xilinx user-programmable FPGA. The design kit includes a CD containing: schematics, parts list, part location drawing, example VHDL source, and other utility files. The 6VLX modules are intended for users fluent in the use of Xilinx FPGA design tools.

BOARD DLL CONTROL SOFTWARE

Acromag provides software products (sold separately) to facilitate the development of Windows® applications interfacing with Acromag PMC, XMC, and VPX I/O board products, PCI and PCIe I/O Cards, and CompactPCI I/O Cards. This software (models PCISW-API-WIN32 and PCISW-API-WIN64) 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.

BOARD VxWORKS SOFTWARE

Acromag provides a software product (sold separately) consisting of board VxWorks® software. This software (Model PMCSW-API-VXW) is composed of VxWorks® (real time operating system) libraries for all Acromag PMC, XMC, and VPX I/O board products, PCI and PCIe I/O Cards, and CompactPCI 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 PCI and PCIe boards.

BOARD Linux SOFTWARE

Acromag provides a software product consisting of board Linux® software. This software (Model PMCSW-API-LNX) is composed of Linux® libraries for all Acromag PMC, XMC, and VPX I/O board products, PCI and PCIe I/O cards, and CompactPCI I/O cards. The software supports X86 PCI bus only and is implemented as library of "C" functions which link with existing user code to make possible simple control of all Acromag PCI and PCIe boards.

2.0 PREPARATION FOR USE

Unpacking and Inspecting



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

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. However, it is recommended that the board be visually inspected for evidence of mishandling prior to applying power.

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 thermo conduction must be provided to prevent a temperature rise above the maximum operating temperature.

Heat Sink Considerations

When the XMC-6VLX board is mounted on a conduction cooled carrier it may be necessary to remove the Acromag provided heat sink. The heat sink can be removed by removal of the 7 screws retaining the heat sink. These 7 screws are visible on the back of the board. After the 7 screws are removed, lift the heat sink off and wipe clean the silver colored thermal interface material from the Virtex 6 device.

Board Installation

Remove power from the system before installing board, cables, termination panels, and field wiring.

Default Hardware Configuration

The board may be configured differently, depending on the application. When the board is shipped from the factory, it is configured with the Acromag example design as follows:

- Slave SelectMap using Platform Flash (XCF128X) with onboard 48 MHz oscillator.

The control registers must be programmed to the desired configuration before starting data input or output operation (see section 3).

P16 Secondary XMC Connector

The P16 secondary XMC connector connects directly to the user-programmable FPGA for both high speed Giga bit data signals and standard I/O user signals. The user I/O pins are connected to FPGA banks with VCCO pins powered by 2.5 volts. Thus these user I/O pins will support the 2.5 volt IOStandards. The IOSTANDARD attribute can be set in the user constraints file (UCF). For example, P16 user I/O can be defined for LVDS_25 (Low-Voltage Differential Signaling). The example design defines the P16 I/O to LVCMOS25 (low voltage CMOS) in the user constraints file. The tables included in the P16 Input Data Register and P16 Output Data Register sections can be used to map the LVCMOS signal to the signal names given in this table. The 2.5 volt IOStandards available are listed in the Virtex-6 User Guide available from Xilinx.

Table 2.1: *Board P16 Secondary XMC Connections*

The example design implements 2.5volt LVCMOS I/O to the P16 connector.

Alternatively, 2.5volt LVDS I/O can be used on the rear connector.

Pin	A	B	C	D	E	F
1	DP00+	DP00-	S18G_N	DP01+	DP01-	S18G_P
2	GND	GND	S16_N	GND	GND	S17_N
3	DP02+	DP02-	S16_P	DP03+	DP03-	S17_P
4	GND	GND	S14_N	GND	GND	S15_N
5	DP04+	DP04-	S14_P	DP05+	DP05-	S15_P
6	GND	GND	S12_N	GND	GND	S13_N
7	DP06+	DP06-	S12_P	DP07+	DP07-	S13_P
8	GND	GND	S10_N	GND	GND	S11_N
9	DP08+	DP08-	S10_P	DP09+	DP09-	S11_P
10	GND	GND	S8_N	GND	GND	S9_N
11	DP10+	DP10-	S8_P	DP11+	DP11-	S9_P
12	GND	GND	S6_N	GND	GND	S7_N
13	DP12+	DP12-	S6_P	DP13+	DP13-	S7_P
14	GND	GND	S4_N	GND	GND	S5_N
15	DP14+	DP14-	S4_P	DP15+	DP15-	S5_P
16	GND	GND	S2_N	GND	GND	S3_N
17	DP16+	DP16-	S2_P	DP17+	DP17-	S3_P
18	GND	GND	S0G_N	GND	GND	S1_N
19	DP18+	DP18-	S0G_P	DP19+	DP19-	S1_P

As LVDS signal pairs, the signals can be grouped to match the ANSI/VITA 46.0 X38s pattern map. A total of 19 differential signal pairs are provided. These differential signal pairs connect to column C and F of the P16 XMC

connector as shown in the following table. For example S3_P and S3_N form a signal pair. There are two global clock differential pairs available (S0G_P, S0G_N) and (S18G_P, S18G_N). The P identifies the Positive input while the N identifies the Negative input.

This XMC P16 Secondary connector is a 114-pin Samtec ASP-103614-05 connector. The connector complies with the ANSI/VITA 42.3-2006.

Rear P4 Field I/O Connector

The rear I/O P4 connector connects directly to the user-programmable FPGA. The VCCO pins are powered by 2.5 volts and thus will support the 2.5 volt IOStandards. The IOSTANDARD attribute can be set in the user constraints file (UCF). The example design defines the Rear P4 I/O to LVCMOS25 (low voltage CMOS) in the user constraints file. The tables included in the Rear Input Data Register and Rear Output Data Register sections can be used to map the LVCMOS signal to the signal names given in the table below.

The rear I/O can alternatively be defined for LVDS_25 (Low-Voltage Differential Signaling) in the user constraints file. The 2.5 volt IOStandards available are listed in the Virtex-6 User Guide available from Xilinx.

As LVDS signal pairs, the signals can be grouped as 32 LVDS I/O pairs. The LVDS pairs are arranged in the same row in table 2.2. For example, RIO1_P and RIO1_N form a signal pair. The P identifies the Positive input while the N identifies the Negative input.

Table 2.2: *Board Rear Field I/O Pin Connections.*

The example design implements 2.5volt LVCMOS I/O to the rear connector.

Alternatively, 2.5volt LVDS I/O can be used on the rear connector.

Ch.	Positive Pin Description	Pin	Negative Pin Description	Pin
0	RIO0_GCLK_P	1	RIO0_GCLK_N	3
1	RIO1_P	2	RIO1_N	4
2	RIO2_P	5	RIO2_N	7
3	RIO3_P	6	RIO3_N	8
4	RIO4_P	9	RIO4_N	11
5	RIO5_P	10	RIO5_N	12
6	RIO6_P	13	RIO6_N	15
7	RIO7_P	14	RIO7_N	16
8	RIO8_P	17	RIO8_N	19
9	RIO9_P	18	RIO9_N	20
10	RIO10_P	21	RIO10_N	23
11	RIO11_P	22	RIO11_N	24
12	RIO12_P	25	RIO12_N	27
13	RIO13_P	26	RIO13_N	28
14	RIO14_P	29	RIO14_N	31
15	RIO15_P	30	RIO15_N	32
16	RIO16_P	33	RIO16_N	35
17	RIO17_P	34	RIO17_N	36
18	RIO18_P	37	RIO18_N	39
19	RIO19_P	38	RIO19_N	40

20	RIO20_P	41	RIO20_N	43
21	RIO21_P	42	RIO21_N	44
22	RIO22_P	45	RIO22_N	47
23	RIO23_P	46	RIO23_N	48
24	RIO24_P	49	RIO24_N	51
25	RIO25_P	50	RIO25_N	52
26	RIO26_P	53	RIO26_N	55
27	RIO27_P	54	RIO27_N	56
28	RIO28_P	57	RIO28_N	59
29	RIO29_P	58	RIO29_N	60
30	RIO30_P	61	RIO30_N	63
31	RIO31_GCLK_P	62	RIO31_GCLK_N	64

This connector is a 64-pin female receptacle header (AMP 120527-1 or equivalent) which mates to the male connector on the carrier board (AMP 120521-1 or equivalent).

Mezzanine Connector

The mezzanine connector provides interface to JTAG, USB and 13 differential I/O signal pairs. Two of these 13 signal pairs are routed to global clock pins on the Virtex 6 device. Two high speed serial interfaces are also routed from the FPGA to this mezzanine module connector.

The 26 front I/O signals connect directly to the user-programmable FPGA. The VCCO pins are powered by 2.5 volts and thus will support the 2.5 volt IOStandards. The IOSTANDARD attribute can be set in the user constraints file (UCF). The example design defines the Front I/O to LVCMOS25 (low voltage CMOS) in the user constraints file. The tables included in the Front Input Data Register and Front Output Data Register sections can be used to map the LVCMOS signal to the signal names given in the table below.

The Front I/O can alternatively be defined for LVDS_25 (Low-Voltage Differential Signaling) in the user constraints file. The 2.5 volt IOStandards available are listed in the Virtex-6 User Guide available from Xilinx.

Table 2.3: *Board Mezzanine Pin Connections.*

The example design implements 2.5volt LVCMOS I/O to the front connector.

Alternatively, 2.5volt LVDS I/O can be used on the front connector.

Ch.	Positive Pin	Pin	Negative Pin	Pin
	TCK	A57	TMS	B50
	TDO	A58	TDI	A64
0	FIO0_P	B55	FIO0_N	B54
1	FIO1_P	B51	FIO1_N	B52
2	FIO2_P	B47	FIO2_N	B48
3	FIO3_P	B44	FIO3_N	B45
4	FIO4_P	B41	FIO4_N	B42
5	FIO5_P	B38	FIO5_N	B39
6	FIO6_P	B35	FIO6_N	B36
7	FIO7_P	B32	FIO7_N	B33
8	FIO8_P	B29	FIO8_N	B30
9	FIO9_P	B26	FIO9_N	B27
10	FIO10_P	B23	FIO10_N	B24

11	FIO11_GCLK_P	B20	FIO11_GCLK_N	B21
12	FIO12_GCLK_P	B17	FIO12_GCLK_N	B18
	USB_D+	A7	USB_D-	A6
	USB_VBUS(from host)	A9		
	SFP1_TX_Fault	B14	SFP2_TX_Fault	B11
	SFP1_LOS	B15	SFP2_LOS	B12
	SFP1_Present_n	A16	SFP2_ Present_n	A15
	SFP1_clock	A18	SFP2_clock	A19
	SFP1_data	A22	SFP2_data	A21
	SFP1_TX_Disable	A25	SFP2_TX_Disable	A24
	SFP1_TX_P	A28	SFP1_TX_N	A29
	SFP1_RX_P	A31	SFP1_RX_N	A32
	SFP2_TX_P	A34	SFP2_TX_N	A35
	SFP2_RX_P	A37	SFP2_RX_N	A38
	+3.3V	B10, B13, B16, B53		
	+2.5V	B1, B2, B3		
	+12V	A2		
	GND	A5, A8, A11A14, A17, A20, A23, A26, A17,		
	GND	A20, A23, A26, A27, A30, A33, A36, A39,		
	GND	A42, A45, A48, A51, A54, A67, A70, A73		
	GND	B19, B22, B25, B28, B31, B34, B37, B40,		
	GND	B43, B46, B49, B73		

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.

3.0 PROGRAMMING INFORMATION

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

GETTING STARTED

1. The XMC-6VLX board is shipped with the user-programmable Xilinx FPGA code stored in the Platform Configuration flash memory (U1). Upon power-up the XMC-6VLX will automatically configure the FPGA with the example design code stored in flash. As a first step become familiar with the XMC-6VLX, with the example code supplied by Acromag. 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 operations performed by the example design. It will allow testing of PCIe interface, read/write of QDRII SRAM, all digital I/O ports, interrupts, testing of both SFP Ports, P16 Aurora loopback, 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 XMC-6VLX as provided in the example design.

2. After you are familiar with the XMC-6VLX and have tested it using the example design, you can move on to step 2. Here you will modify the example design VHDL code slightly. The Xilinx Platform Configuration flash must be overwritten to test your code. 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 return to use of the Acromag design example. You can reload the Acromag example design via the EDK board and the JTAG port using the Xilinx iMPACT 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 Platform flash device. Registers are provided in the FPGA Programming Memory Map to implement Platform or BPI flash erase and reprogram operations.

Virtex 6 Configuration

The XMC-6VLX board supports configuration in the following modes:

- Slave SelectMAP using Xilinx Platform Flash XL with onboard 48 MHz oscillator. The Xilinx Platform flash configuration device contains the Acromag example design.
- Master BPI-Up using Linear BPI Flash device. A BPI flash device is recommended for MicroBlaze program storage.
- JTAG using Xilinx external program cable.

Upon a power-up cycle, the contents of the Xilinx Platform flash device are downloaded to the FPGA. See chapter 4 for DIP switch setting options.

Platform Flash Xilinx Configuration

The Xilinx Platform flash configuration data can be reprogrammed using the PCIe bus interface or the JTAG interface. The following is the general procedure for reprogramming the Platform flash memory and reconfiguration of the Xilinx FPGA:

1. Set DIP switch 5 to the OFF position. This will enable selection of the Platform flash device and disable selection of the BPI flash device. Set DIP switch 2 and 3 to the ON position. This will select Slave SelectMAP mode.
2. Power Cycle the System with the XMC-6VLX module. At power-up the configuration file will automatically be loaded into the FPGA provided the DIP switch is set as described in the preceding step for configuration from Platform flash.

BPI Flash Xilinx Configuration

The Byte-wide Peripheral Interface (BPI) flash is recommended for MicroBlaze CPU program code storage. The following is the general procedure for reprogramming the BPI flash memory:

1. Set DIP switch 5 to the ON position. This will enable selection of the BPI flash device and disable selection of the Platform flash device.
2. See the Flash Configuration section for a description of the steps required to write new data or program code to the BPI Flash device. Registers are provided in the FPGA Programming Memory Map to implement BPI flash erase and reprogram operations.

PCIe CONFIGURATION ADDRESS SPACE

This board is a PCI Express Base Specification Revision v2.0 compliant PCIe bus board.

The PCI 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 relocatable and 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 3.1, 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 3.1 *Configuration Registers*

Reg. Num.	D31 D24	D23 D16	D15 D8	D7 D0
0	**Device ID** 0x6301 XMC-6VLX240F 0x6302 XMC-6VLX365F 0x6303 XMC-6VLX240 0x6304 XMC-6VLX365		**Vendor ID** 16D5	
1	Status		Command	
2	Class Code=118000			Rev ID=00
3	BIST	Header	Latency	Cache
4	64-bit Memory Base Address for Memory Accesses to PCIe interrupt, I/O registers, System Monitor registers, and Flash memory. 4M Space **(BAR0)**			
5	64-bit Memory Base Address for Memory Accesses to QDRII memory. 16M Space **(BAR2)**			
6:10	Not Used			
11	**Subsystem ID** 0x6301 XMC-6VLX240F 0x6302 XMC-6VLX365F 0x6303 XMC-6VLX240 0x6304 XMC-6VLX365		**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, System Monitor registers, and Flash memory. The PCIe bus decodes 4M bytes for BAR0 for this memory space.

This board is also allocated a 16M byte block of memory (BAR2), to access QDRII memory. The PCIe bus decodes 16M bytes for BAR2 for this memory space.

BAR0 MEMORY MAP

The BAR0 memory address space is used to access the PCIe interrupt, Front, Rear, 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 3.2: *BAR0 Registers*

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

BAR0 Base Address	Size	Description
0x00000000→0x0000FFFF	64K	Reserved
0x00010000→0x0001FFFF	64K	Reserved
0x00020000→0x0009FFFF	512K	Reserved
0x000A0000→0x000AFFFF	64K	AXI CDMA (DS792)
0x000B0000→0x000EFFFF	256K	Reserved
0x000F0000→0x000FFFFF	64K	PCIe AXI Bridge Control (DS820)
0x00100000→0x010FFFFF	64K	PCIe Interrupt Controller
0x00110000→0x002FFFFF	2M	Reserved
0x00300000→0x0030FFFF	64K	Flash, I/O Reg, System Monitor, and Aurora
0x00310000→0x003FFFFF	1M	Reserved

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 3.3: *Interrupt Controller Registers*

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

BAR0 Base Addr+	Bit(s)	Description
0x00100000	31:0	Interrupt Status Register
0x00100004	31:0	Interrupt Pending Register
0x00100008	31:0	Interrupt Enable Register
0x0010000C	31:0	Interrupt Acknowledge Register
0x00100010	31:0	Set Interrupt Enable Register
0x00100014	31:0	Clear Interrupt Enable Register
0x00100018	31:0	Interrupt Vector Register
0x0010001C	31:0	Master Enable Register

Interrupt Status Register (Read/Write) - (BAR0 + 0x00100000)

This Interrupt Status register (ISR) at BAR0 base address + offset 0x100000 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 3.4: *Interrupt Status Register*

Bit(s)	FUNCTION	
0	This bit when set indicates a Xilinx Fabric interrupt from the Front I/O interface. See the Front I/O interrupt section for source of this interrupt.	
	0	Disabled
	1	Enabled
1	This bit when set indicates an AXI CDMA interrupt. See the CDMA section for source of this interrupt.	
	0	Disabled
	1	Enabled
31-2	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 (Read) - (BAR0 + 0x00100004)

This Interrupt Pending register (IPR) at BAR0 base address + offset 0x100004 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 3.5: *Interrupt Pending Register*

Bit(s)	FUNCTION	
0	This bit when set indicates a Xilinx Fabric interrupt from the Front I/O interface. See the Front I/O interrupt section for source of this interrupt.	
	0	Disabled
	1	Enabled
1	This bit when set indicates an AXI CDMA interrupt. See the CDMA section for source of this interrupt.	
	0	Disabled
	1	Enabled
31-2	Reserved	
	0	NA
	1	NA

Interrupt Enable Register (Read/Write) - (BAR0 + 0x00100008)

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 3.6: *Interrupt Enable Register*

Bit(s)	FUNCTION	
0	This bit when set indicates a Xilinx Fabric interrupt from the Front I/O interface is enabled. See the Front I/O interrupt section for source of this interrupt.	
	0	Disabled
	1	Enabled
1	This bit when set indicates an AXI CDMA interrupt enable. See the CDMA section for source of this interrupt.	
	0	Disabled
	1	Enabled
31-2	Reserved	
	0	NA
	1	NA

Interrupt Acknowledge Register (Write) - (BAR0 + 0x0010000C)

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.

Set Interrupt Enable Register (Write) - (BAR0 + 0x00100010)

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 wiring 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.

Clear Interrupt Enable Register (Write) - (BAR0 + 0x00100014)

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 wiring 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.

Interrupt Vector Register (Read) - (BAR0 + 0x00100018)

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.

Master Enable Register (Read/Write) - (BAR0 + 0x0010001C)

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 3.7: *Master Enable Register*

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 on the PCIe bus and the memory resident on the XMC-6VLX.

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 sequential 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 matches the address of the completed descriptor, the Scatter Gather Engine stops descriptor fetching and waits. See the Xilinx DS792 data sheet for the AXI Central Direct Memory Access for additional details for Scatter Gather operations.

Table 3.8:*AXI CDMA Registers*

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

BAR0 Base Addr+	Bit(s)	Description
0x000A0000	31:0	CDMA Control Register
0x000A0004	31:0	CDMA Status Register
0x000A0008	31:0	Current Descriptor Pointer Register
0x000A000C	31:0	Reserved
0x000A0010	31:0	Tail Descriptor Pointer Register
0x000A0014	31:0	Reserved
0x000A0018	31:0	Source Address Register
0x000A001C	31:0	Reserved
0x000A0020	31:0	Destination Address Register
0x000A0024	31:0	Reserved
0x000A0028	31:0	Bytes to Transfer Register

CDMA Control Register (Read/Write) - (BAR0 + 0x000A0000)

This register provides software application control of the AXI CDMA.

Table 3.9: *CDMA Control Register*

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
	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.	
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 (Read/Write) - (BAR0 + 0x000A0004)

This register provides status of the AXI CDMA.

Table 3.10: *CDMA Status Register*

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
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.
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	Reservered	
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 (Read/Write) - (BAR0 + 0x000A0008)

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

Table 3.11: *CDMA Current Descriptor Pointer Register*

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 (Read/Write) - (BAR0 + 0x000A0010)

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

Table 3.12: *CDMA Tail Descriptor Pointer Register*

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 (Read/Write) - (BAR0 + 0x000A0018)

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

Note if the QDR memory is the source the base address from which QDR memory starts is 0x30000000.

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 3.13: *CDMA Source Address Register*

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 (Read/Write) - (BAR0 + 0x000A0020)

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

Note if the QDR memory is the destination the base address from which QDR memory starts is 0x30000000.

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 3.14: *CDMA Destination Address Register*

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 (Read/Write) - (BAR0 + 0x000A0028)

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

Table 3.15: *CDMA Bytes to Transfer Register*

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 and 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 0xA0018. The transfer data at the source address must be valid and ready for transfer. If we were to select the QDR memory as the source and wanted to start a move of data from the beginning of QDR, we would write 0x30000000 to the Source Address register at 0xA0018.
4. Write the desired transfer destination address to the Destination Address register at 0xA0020. 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 0xA0020.
5. Write the number of bytes to transfer to the CDMA Bytes to Transfer register 0xA0028. 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 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 the Xilinx Platform Studio the address map will show that a 16Meg address space for the AXI BAR0 Aperture Base Address is reserved.

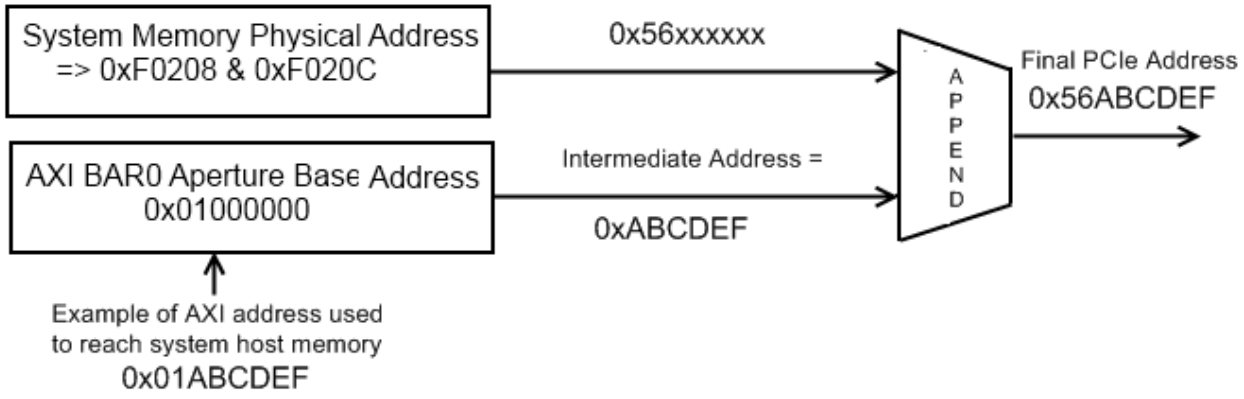
Table 3.16: *AXI BAR0 Aperture Base Address*

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 + 0xF0208 and 0xF020C 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 Virtex 6 hardware to give the final PCIe address of the system memory location.



PCIe AXI-Bridge Control

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

Table 3.17:*PCIe AXI Bridge Control Registers*

BAR0 Base Addr+	Bit(s)	Description
0x000F0000→ 0x000F0140	31:0	See Xilinx DS820 Memory Map
0x000F0144	31:0	Physical Side Interface Status
0x000F0148→ 0x000F0204	31:0	See Xilinx DS820 Memory Map
0x000F0208	31:0	Address Translation Register Upper AXIBAR2PCIEBAR_0U
0x000F020C	31:0	Address Translation Register Lower AXIBAR2PCIEBAR_0L
0x000F0210→ 0x000F0FFF	31:0	See Xilinx DS820 Memory Map

Physical Side Interface Status/Control Register (Read/Write) - (BAR0 + 0x000F0144)

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

Table 3.18: *CDMA Control Register*

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
	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 DS820 PHY Status/Control Register	

AXI Base Address Translation Configuration Register (Read Only) - (BAR0 + 0xF0208/0xF020C)

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

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

AXI Base Address Translation Configuration register at BAR0 + 0xF0208 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 + 0xF020C 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 + 0xF0208)

#define AXI2PCIeBAR_0L          (*(DWORD *)(u64BaseAddress +0xF020C)

iStatus = PCIe6VLX_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 0xA0018 or 0xA0020 respectively. See the example in the CDMA section for additional details.

FPGA Fabric MEMORY MAP

The BAR0 FPGA Fabric memory address space is used to access the Flash Configuration, Front, Rear, and P16 I/O registers and System Monitor registers. This memory space contains FPGA functions implemented in the FPGA fabric. All other logic is implemented using Xilinx Platform Studio. 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 3.19: *BAR0 Registers*

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

BAR0 Base Addr+	Bit(s)	Description
0x300000	31:0	Interrupt Status/Clear
0x300004	31:0	Reserved
0x300008	31:0	DDR Memory Test Status Register
0x30000C	31:0	Board Identification Register
0x300010→ 0x3000FF	31:0	Reserved
0x300100	31:0	Configuration Control
0x300104	31:0	Aurora Monitor
0x300108→ 0x3001FF	31:0	Reserved
0x300200	0	Flash Status
0x300204	0	Flash Control
0x300208	0	Flash Read
0x30020C	0	Flash Start Write
0x300210	0	Flash Erase Sector
0x300214	15:0	Flash Data Register
0x300218	24:0	Flash Address Register
0x30021C→ 0x3002FF	31:0	Reserved
0x300300	31:0	System Monitor Status/Control Register
0x300304	31:0	System Monitor Address Register
0x300308→ 0x300FFF	31:0	Reserved

Front, Rear, and P16 I/O Registers (Read/Write) – (BAR0 + 0x301000 to 0x 301FFF)

The BAR0 memory space from 0x301000 to 0x301FFF is used to access the Front, Rear, and P16 I/O registers.

Table 3.20: *BAR0 Registers*

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

BAR0 Base Addr+	Bit(s)	Description
0x301000	31:0	Front Input Data Register
0x301004	31:0	Front Output Data Register
0x301008	3:0	Interrupt Enable
0x30100C	3:0	Interrupt Type
0x301010	3:0	Interrupt Polarity
0x301014→ 0x3010FF	31:0	Reserved
0x301100	31:0	Rear Input Data Register
0x301104	31:0	Rear Output Data Register
0x301108→ 0x3011FF	31:0	Reserved
0x301200	31:0	P16 Input Data Register
0x301204	31:0	P16 Output Data Register
0x301208→ 0x30FFFF	31:0	Reserved

Front I/O Interrupt Status/Clear Register (Read/Write) - (BAR0 + 0x300000)

This read/write register is used to determine the pending status of FPGA fabric interrupts and release pending interrupts. This interrupt status/clear registers reflect the status of each of the front write channel interrupts. Read of this bit reflects the interrupt pending status. Read of a "1" indicates that an interrupt is pending for the corresponding channel. Write of a logic "1" to this bit to release the corresponding channel's pending interrupt. Writing "0" to a bit location has no effect, a pending interrupt will remain pending. Front write channel 0 interrupt status is identified via data bit-0 while front write channel 3 status is identified via data bit-3 of this register at BAR0 plus 0x300000.

DDR Memory Test Status Register (Read/Write) - (BAR0 + 0x300008)

This read/write register is used to determine the Flash BPI memory read, and DDR memory read/write test status. Read of bit-0 reflects the Flash BPI memory read status. Read of a "1" indicates a BPI Flash read error. Read of bit-1 reflects the DDR memory read/write status. Read of a "1" indicates a DDR memory read or write error. Read of bit-2 reflects that all tests passed. Read of a "1" indicates that both the Flash and DDR memory tests passed.

XMC Board Identification Code Register (Read Only) - (BAR0 + 0x30000C)

The XMC Board Identification Code register at BAR0 plus 0x30000C stores an ID code that can used to uniquely identify the XMC Virtex 6 card. This register will read A3 hex as provided by the Acromag example design. The user can change the hardware setting of this register in the programmable FPGA code. This ID code can be used to properly assign software drivers to multiple XMC boards that may have the same device and vender ID in a given system.

Configuration Control (Read/Write) – (BAR0 + 0x300100)

This read/write register configuration control register has multiple functions. This Configuration Control register is accessed at base address plus 0x300100. The Configuration Control register bit-0 is used to select one of the two flash memory devices for erase or program read/write operations. The Configuration Control register bit-1 must be set to logic 1 to select Platform flash address flow though mode. This is only required when bit-0 of this register is set to logic 0 and Platform flash erase and read/write operation using the PCIe bus are required. Bit-1 of the Configuration Control register must be set to logic 0 when Configuration of the Platform flash device is implemented using JTAG.

Table 3.21: *Configuration Control Register*

Bit(s)	Configuration Control Register	
0	Flash Memory Select:	
	0	Platform Flash Configuration is selected (16M byte)
	1	BPI Flash Configuration is selected (32M byte)
1	Platform Flash Address Flow Through	
	0	Write logic low has no effect.
	1	Write logic high to select address flow through
2	CCLK Clock Control	
	0	Write logic low enables the CCLK.
	1	Write logic high disables the CCLK
3-31	Reserved	
	0	Write logic low has no effect
	1	Write logic high has no effect

Aurora Monitor (Read/Write) – (BAR0 + 0x300104)

This read/write register Aurora Monitor register is used to monitor eight Aurora loopback lanes that are on the P16 connector. This Aurora Monitor register is accessed at base address plus 0x300104. The Aurora Monitor register bit-0 is used take the Aurora link into and out of reset. Set to logic '1' the link is held in reset and set to logic '0' the link is removed from reset.

Table 3.22: *Aurora Monitor Register*

Bit(s)	Aurora Monitor Register	
0	Aurora Reset Control:	
	0	Removed from Reset
	1	Held in Reset
1		Reserved
2	Channel UP	
	0	Loopback Channel is down
	1	Loopback Channel is up
3-15		Reserved
16-23	Link	
	0	Link is down
	1	Link is up
24-31	Reserved	
	0	Write logic low has no effect
	1	Write logic high has no effect

Flash Introduction

The BPI flash memory has 32M bytes of program code or data storage available. The Platform flash memory has 16M bytes of program code storage available.

The system CPU provides control of all in-system read, write, and erase operations for both the BPI flash and the Xilinx Platform flash devices via the PCIe bus. The on-chip FPGA logic automatically executes the algorithms and timings necessary for block erase and program. A Status Register indicates erase or program completion and any errors that may have occurred.

The BPI flash device has 256 individual erasable memory blocks each 64K words deep. The Platform flash device has 128 individual erasable memory blocks each 64K words deep. See the memory maps for both flash memory devices below. The least significant 16 bits A15 to A0 are used to select the 64K words of each block.

Figure 3.23: *BPI Flash Memory Map*

The most significant flash address lines A23 to A16 are used to select one of 256 flash 64 Kword blocks as shown in this figure.

A15 to A14 are used to select one of the four 16 Kword top blocks.

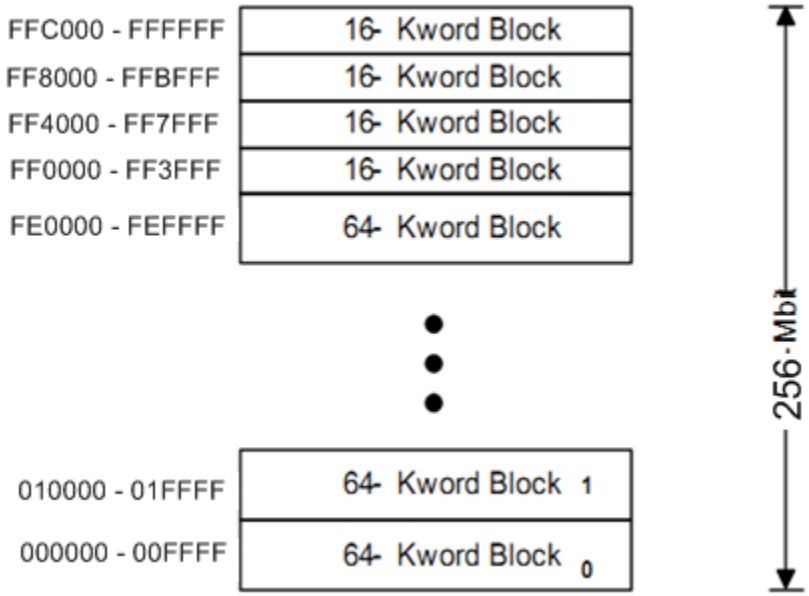
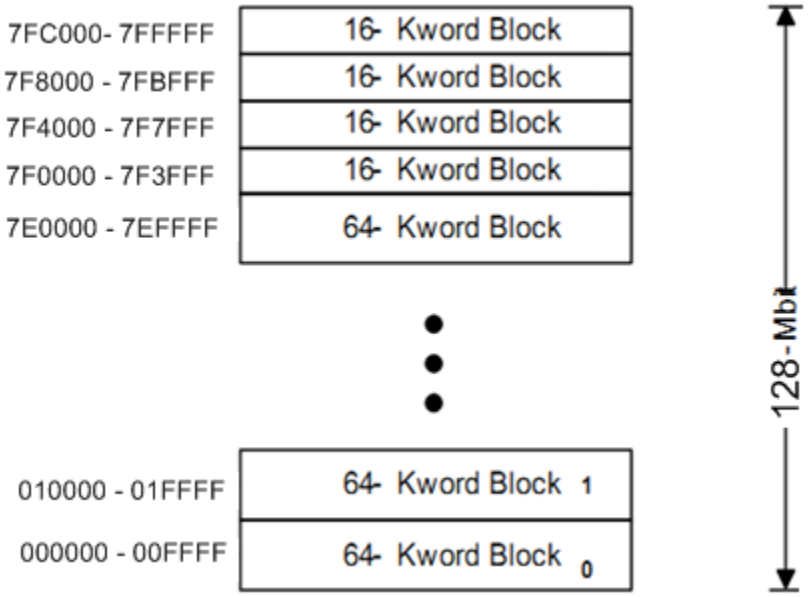


Figure 3.24: *Platform Flash Memory Map*

The most significant flash address lines A22 to A16 are used to select one of 128 flash 64 Kword blocks as shown in this figure.

A15 to A14 are used to select one of the four 16 Kword top blocks.



Flash Status (Read Only) – (BAR0 + 0x300200)

This read only register is used to read the status of the BPI or Platform flash chip. The Flash Status register is at base address plus 0x300200. Status Register data is output on DQ[7:0], while 0x00 is output on DQ[15:8].

Table 3.25: *Flash Status Register*

Bit(s)	Status Register Description	
0	Buffered Enhanced Factory Programming (BEFP)	
	x	BEFP is not available
	x	BEFP is not available
1	Block Lock Status	
	0	Block not locked
	1	Block locked
2	Program Suspend Status	
	0	Program Suspend not active
	1	Program Suspend active
3	Vpp Status	
	0	Vpp Acceptable
	1	Vpp Low
4	Program Status	
	0	Program Successful
	1	Program Error
5	Erase Status	
	0	Erase Successful
	1	Erase Error
6	Erase Suspend Status	
	0	Not Suspended
	1	Suspended
7	Device Ready Status	
	0	Device is busy
	1	Device is ready SR[6:1] are valid

The Clear Status Register command is used to clear the Status Register error bits.

Flash Control (Write Only) – (BAR0 + 0x300204)

A Flash Control command is executed by writing this register at base address plus 0x300204. Write to Flash Control register at 0x300204 with bit-0 set to logic '1' will initiate a Block Unlock sequence to the Flash device. ***The address of the block that is unlocked must first be written to the Flash Address register at 0x300218.***

Bit-1 of this register is used to initiate a reset of the BPI flash chip. A Flash Reset command is executed by writing logic 1 to bit-1 of this register at base address plus 0x300204. Writing the flash reset command resets the chip to reading data mode.

Bit-2 of this register is used to clear the Flash Status of the BPI or Platform flash chip. A Clear Flash Status command is executed by writing logic 1 to bit-2 of this register at base address plus 0x300204.

Bit-3 of this register is used to set Platform Flash Asynchronous Mode. Write to Flash Control register at 0x300204 with bit-3 set to logic '1' will initiate a set configuration sequence to the Flash device. ***The flash address must first be written to the Flash Address register at 0x300218 with data value 0x8000 to select asynchronous mode.***

Table 3.26: *Flash Control Register*

Note Block Unlock , PBI Flash Reset, and Clear Flash Status can not be simultaneously set in this register. Only one operation can be selected at a time.

Bit(s)	Flash Control Register Description	
0	Block Unlock	
	0	Write logic low has no effect.
	1	Write logic high to initiate Block Unlock
1	BPI Flash Reset:	
	0	Write logic low has no effect.
	1	Write logic high to initiate BPI flash
2	Clear Flash Status	
	0	Write logic low has no effect.
	1	Write logic high to initiate Clear Flash Status
3	Platform Flash Asynchronous Mode	
	0	Write logic low has no effect.
	1	Write logic high to select Asynchronous Mode
4-31	Reserved	
	0	Write logic low has no effect
	1	Write logic high has no effect

Flash Read (Read Only) – (BAR0 + 0x300208)

A Flash Read command is executed by reading this register at base address plus 0x300208. Prior to issue of a Flash Read the Flash Address registers must be set with the desired address to be read. See the Flash Address registers at base address plus 0x300218.

Flash Start Write (Write Only) – (BAR0 + 0x30020C)

This write only register is used to initiate the write of a 16-bit data value to the flash chip. A Flash Start Write command is executed by writing logic 1 to bit-0 of this register at base address plus 0x30020C. Prior to issuing of a Flash Start Write the Flash Data and Address registers must be set with the desired data and address to be written. See the Flash Data and Address registers at base address plus 0x300214 and 0x300218.

Issuing a Flash Start Write will automatically increment this address after the previously issued Flash Write has completed. Thus, the address will not need to be set prior to issuing the next Flash Start Write if consecutive addresses are to be written.

Flash Erase Block (Write Only) – (BAR0 + 0x300210)

This write only register is used to erase the contents of the addressed flash block. A flash bit cannot be programmed from logic 0 to logic 1. Only an erase block operation can convert logic 0 back to logic 1. ***Prior to reprogramming of the flash chip a flash erase block command must be performed.***

A flash erase block command is executed by writing logic 1 to bit-0 of this register at base address plus 0x300210. Verify that the flash chip is not busy from a previous operation before beginning a new operation. This is accomplished by reading the flash status register.

Any other flash commands written to the flash chip during execution of the flash erase block operation will be ignored. Note that a hardware reset during the sector erase operation will immediately terminate the operation.

Flash Data Register (Read/Write) – (BAR0 + 0x300214)

This read/write register holds the 16-bit data which is sent to the flash chip upon issuing of a flash start write command.

Flash Address (Read/Write) – (BAR0 + 0x300218)

This read/write register holds the address to which the flash chip is written upon issue of a flash start write command.

Simple BPI Flash Programming Example

1. Write 0x1 to address 0x300100, BPI flash device is selected.
2. Read address 0x300200, Status register value 0x80.
3. Write 0x0 to address 0x300218, flash address set to block 0.
4. Write 0x1 to address 0x300204, flash block 0 unlock.
5. Write 0x1 to address 0x300210, flash block 0 erased.
6. Write 0x3A3A to address 0x300214, flash data register set with 0x3A3A.
7. Write 0x1 to address 0x30020C, flash data written to flash.
8. Write 0x0 to address 0x300218, set flash address back to 0.
9. Read address 0x300208, flash data at address 0x0 is 0x3A3A.

Simple Platform Flash Programming Example

1. Write 0x2 to address 0x300100, Platform flash address flow though is selected.
2. Write 0x8000 to address 0x300218, Flash Address. Bit -15 is set for asynchronous mode.
3. Write 0x8 to address 0x300204, Platform flash set to asynchronous mode.
4. Read address 0x300200, Status register value 0x80.
5. Write 0x0 to address 0x300218, flash address set to block 0.
6. Write 0x1 to address 0x300204, flash block 0 unlock.
7. Write 0x1 to address 0x300210, flash block 0 erased.
8. Write 0x3AA3 to address 0x300214, flash data register set with 0x3AA3.
9. Write 0x1 to address 0x30020C, flash data written to flash.
10. Write 0x0 to address 0x300218, set flash address back to 0.
11. Read address 0x300208, flash data at address 0x0 is 0x3AA3.

System Monitor Status/Control Register (Read/Write) – (BAR0 + 0x300300)

This read/write register will access the system monitor register at the address set in the System Monitor Address Register.

For example, the address of the System Monitor Status register that is to be accessed is first set via the System Monitor Address register at BAR0 plus 0x300304. Next, this register at BAR0 plus 0x300300 is read. Bits 22 to 16 of this register hold the address of the system monitor register that is accessed. Data bits 15 to 6 of this register hold the "ADCcode" temperature, Vccint, or Vccaux value. Data bits 5 to 0 are not used. Valid addresses are given in column one of the table below.

Reading or writing this register is possible via 32-bit data transfers.

The 10-bits digitized and output from the ADC can be converted to temperature by using the following equation.

$$Temperature(°C) = \frac{ADCcode \times 503.975}{1024} - 273.15$$

The 10-bits digitized and output from the ADC can be converted to voltage by using the following equation.

$$SupplyVoltage(volts) = \frac{ADCcode}{1024} \times 3V$$

System Monitor Address Register (Write Only) – (BAR0 + 0x300304)

This write only register is used to set the system monitor address register with a valid address for the System Monitor internal status or control registers. Valid addresses are given in the following table. Additional addresses can be found in the Xilinx System Monitor document UG370 (available from Xilinx). Writing this register is possible via 32-bit data transfers.

The address value written to this register can be read on bits 22 to 16 of the System Monitor Status/Control register at BAR0 plus 0x300300.

Table 3.27: *System Monitor Register Map*

Address	Status Register
0x00	Temperature
0x01	Vccint
0x02	Vccaux
0x20	Maximum Temperature
0x21	Maximum Vccint
0x22	Maximum Vccaux
0x24	Minimum Temperature
0x25	Minimum Vccint
0x26	Minimum Vccaux

Front Input Data Register (Read Only) - (BAR0 + 0x301000)

The front I/O can also be configured as differential channels with 2 global clock signal pairs.

The front input data register is used to access the individual input channels. The front input includes 13 LVCMOS single ended channels. Each channel is controlled by a corresponding data bit as shown in the Front Input Data Register Table.

Channel input signal levels are determined by reading this register. Channel output signals are set by writing to the front output data register at base address plus 0x301004.

This front input data register is a read only register. Channel read operations use 32-bit, 16-bit or 8-bit data transfers. All channels of this register are fixed as input channels.

Table 3.28: *BAR0 Front Input Data Register*

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

Register Bit	Channel	VHDL Name	Schematic Name
0	0	FI(0)	FIO0_N
1	1	FI(1)	FIO1_N
2	2	FI(2)	FIO2_N
3	3	FI(3)	FIO3_N
4	4	FI(4)	FIO4_N
5	5	FI(5)	FIO5_N
6	6	FI(6)	FIO6_N
7	7	FI(7)	FIO7_N
8	8	FI(8)	FIO8_N
9	9	FI(9)	FIO9_N
10	10	FI(10)	FIO10_N
11	11	FI(11)	FIO11_GCLK_N
12	12	FI(12)	FIO12_GCLK_N

Front Output Data Register (Read/Write) - (BAR0 + 0x301004)

The front output data register is used to access the individual LVCMOS output channels. This includes 13 channels. Each channel is controlled by a corresponding data bit as shown in the Front Output Data Register Table.

Channel output signals are controlled by writing this register. Channel input signals are accessed by reading the front input data register at base address plus 0x301000.

This front output data register is a read/writable register. Channel operations use 32-bit, 16-bit or 8-bit data transfers. All channels of this register are fixed as output channels.

Table 3.29: *BAR0 Front Output Data Register*

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

Register Bit	Channel	VHDL Name	Schematic Name
0	0	FO(0)	FIO0_P
1	1	FO(1)	FIO1_P
2	2	FO(2)	FIO2_P
3	3	FO(3)	FIO3_P
4	4	FO(4)	FIO4_P
5	5	FO(5)	FIO5_P
6	6	FO(6)	FIO6_P
7	7	FO(7)	FIO7_P
8	8	FO(8)	FIO8_P
9	9	FO(9)	FIO9_P
10	10	FO(10)	FIO10_P
11	11	FO(11)	FIO11_GCLK_P
12	12	FO(12)	FIO12_GCLK_P

Front I/O Interrupt Enable Register (Read/Write) - (BAR0 + 0x301008)

The Front I/O Interrupt Enable Register provides a map bit for each front output write register from 0 to 3. A "0" bit will prevent the corresponding output channel from generating an interrupt. A "1" bit will allow the corresponding channel to generate an interrupt.

The Front I/O Interrupt Enable register at the base address + offset 0x301008 is used to control front output 0 through 3 interrupts via data bits 0 to 3. Bits 4 to 31 are not used and will always read as "0".

All channel interrupts are disabled (set to "0") following a power-on or software reset. Reading or writing to this register is possible via 32-bit, 16-bit or 8-bit data transfers. Additional steps required to enable interrupts are described in the Interrupt Controller sections.

Interrupt Type (COS or H/L) Configuration Register (Read/Write) - (BAR0 + 0x30100C)

The Interrupt Type Configuration Register determines the type of output channel transition that will generate an interrupt for each of the four possible interrupting channels. A "0" bit selects interrupt on level. An interrupt will be generated when the output channel level specified by the Interrupt Polarity Register occurs (i.e. Low or High level transition interrupt). A "1" bit means the interrupt will occur when a Change-Of-State (COS) occurs at the corresponding output channel (i.e. any state transition, low to high or high to low).

The Interrupt Type Configuration register at base address +0x30100C is used to control channels 0 through 3 as mapped in the Interrupt Enable Register. For example, channel 0 is controlled via data bit-0. Bits 4 to 31 are not used and will always read as "0".

All bits are set to "0" following a reset which means that, if enabled, the

outputs will cause interrupts for the levels specified by the Interrupt Polarity Register.

Channel read or write operations use 8-bit, 16-bit, or 32-bit data transfers. Note that no interrupts will occur unless they are enabled by the Interrupt Enable Register.

Interrupt Polarity Register (Read/Write) - (BAR0 + 0x301010)

The Interrupt Polarity Register determines the level that will cause a channel interrupt to occur for each of the channels enabled for level interrupts. A "0" bit specifies that an interrupt will occur when the corresponding output channel is low (i.e. a "0" in the output channel data register). A "1" bit means that an interrupt will occur when the output channel is high (i.e. a "1" in the output channel data register). Note that no interrupts will occur unless they are enabled by the Interrupt Enable Register. Further, the Interrupt Polarity Register will have no effect if the Change-of-State (COS) interrupt type is configured by the Interrupt Type Configuration Register.

The Interrupt Polarity register at the base address + offset 0x301010 is used to control differential channels 0 through 3 as mapped in the Interrupt Enable Register. For example, channel 0 is controlled via data bit-0. Bits 4 to 31 are not used and will always read as "0".

All bits are set to "0" following a reset, which means that the output will cause interrupts when they are logic low (provided they are enabled for interrupt on level).

Rear Input Data Register (Read Only) - (BAR0 + 0x301100)

The rear I/O can also be configured as differential channels with 2 global clock signal pairs.

The rear input data register is used to access the individual input channels. The rear input includes 32 LVCMOS single ended channels. Each channel is controlled by a corresponding data bit as shown in the Rear Input Data Register Table.

Channel input signal levels are determined by reading this register. Channel output signals are set by writing to the rear output data register at base address plus 0x301104.

This rear input data register is a read only register. Channel read operations use 32-bit, 16-bit or 8-bit data transfers. All channels of this register are fixed as input channels.

Table 3.30: *BAR0 Rear Input Data Register*

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

Register Bit	Channel	VHDL Name	Schematic Name
0	0	RI(0)	RIO0_GCLK_P
1	1	RI(1)	RIO1_P
2	2	RI(2)	RIO2_P
3	3	RI(3)	RIO3_P
4	4	RI(4)	RIO4_P
5	5	RI(5)	RIO5_P
6	6	RI(6)	RIO6_P
7	7	RI(7)	RIO7_P
8	8	RI(8)	RIO8_P
9	9	RI(9)	RIO9_P
10	10	RI(10)	RIO10_P
11	11	RI(11)	RIO11_P
12	12	RI(12)	RIO12_P
13	13	RI(13)	RIO13_P
14	14	RI(14)	RIO14_P
15	15	RI(15)	RIO15_P
16	16	RI(16)	RIO16_P
17	17	RI(17)	RIO17_P
18	18	RI(18)	RIO18_P
19	19	RI(19)	RIO19_P
20	20	RI(20)	RIO20_P
21	21	RI(21)	RIO21_P
22	22	RI(22)	RIO22_P
23	23	RI(23)	RIO23_P
24	24	RI(24)	RIO24_P
25	25	RI(25)	RIO25_P
26	26	RI(26)	RIO26_P
27	27	RI(27)	RIO27_P
28	28	RI(28)	RIO28_P
29	29	RI(29)	RIO29_P
30	30	RI(30)	RIO30_P
31	31	RI(31)	RIO31_GCLK_P

Rear Output Data Register (Read/Write) - (BAR0 + 0x301104)

The rear output data register is used to access the individual LVCMOS output channels. This includes 32 single ended channels. Each channel is controlled by a corresponding data bit as shown in the Rear Output Data Register Table.

Channel output signals are controlled by writing this register. Channel input signals are accessed by reading the rear input data register at base address plus 0x301100.

This rear output data register is a read/writable register. Channel operations use 32-bit, 16-bit or 8-bit data transfers. All channels of this register are fixed as output channels.

Table 3.31: *BAR0 Rear Output Data Register*

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

Register Bit	Channel	VHDL Name	Schematic Name
0	0	RO(0)	RIO0_GCLK_N
1	1	RO(1)	RIO1_N
2	2	RO(2)	RIO2_N
3	3	RO(3)	RIO3_N
4	4	RO(4)	RIO4_N
5	5	RO(5)	RIO5_N
6	6	RO(6)	RIO6_N
7	7	RO(7)	RIO7_N
8	8	RO(8)	RIO8_N
9	9	RO(9)	RIO9_N
10	10	RO(10)	RIO10_N
11	11	RO(11)	RIO11_N
12	12	RO(12)	RIO12_N
13	13	RO(13)	RIO13_N
14	14	RO(14)	RIO14_N
15	15	RO(15)	RIO15_N
16	16	RO(16)	RIO16_N
17	17	RO(17)	RIO17_N
18	18	RO(18)	RIO18_N
19	19	RO(19)	RIO19_N
20	20	RO(20)	RIO20_N
21	21	RO(21)	RIO21_N
22	22	RO(22)	RIO22_N
23	23	RO(23)	RIO23_N
24	24	RO(24)	RIO24_N
25	25	RO(25)	RIO25_N
26	26	RO(26)	RIO26_N
27	27	RO(27)	RIO27_N
28	28	RO(28)	RIO28_N
29	29	RO(29)	RIO29_N
30	30	RO(30)	RIO30_N
31	31	RO(31)	RIO31_GCLK_N

P16 Input Data Register (Read Only) - (BAR0 + 0x301200)

The P16 input data register is used to access the individual LVDS input channels. This includes 10 differential channels which include 2 global clock signal pairs. Each channel is controlled by a corresponding data bit as shown in the P16 Input Data Register Table.

Channel input signal levels are determined by reading this register. Channel output signals are set by writing to the P16 output data register at base address plus 0x301204.

This P16 input data register is a read only register. Channel read operations use 32-bit, 16-bit or 8-bit data transfers. All channels of this register are fixed as input channels.

Table 3.32: *BAR0 P16 Input Data Register*

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

Register Bit	Channel	VHDL Name	Schematic Name
0	0	P16_SI(0)	P16_SIO16_N
1	1	P16_SI(1)	P16_SIO14_N
2	2	P16_SI(2)	P16_SIO12_N
3	3	P16_SI(3)	P16_SIO10_N
4	4	P16_SI(4)	P16_SIO8_N
5	5	P16_SI(5)	P16_SIO6_N
6	6	P16_SI(6)	P16_SIO4_N
7	7	P16_SI(7)	P16_SIO2_N
8	8	P16_SI(8)	P16_SIO0_GCLK_N
9	9	P16_SI(9)	P16_SIO18_GCLK_P
10	10	P16_SI(10)	P16_SIO17_P
11	11	P16_SI(11)	P16_SIO15_P
12	12	P16_SI(12)	P16_SIO13_P
13	13	P16_SI(13)	P16_SIO11_P
14	14	P16_SI(14)	P16_SIO9_P
15	15	P16_SI(15)	P16_SIO7_P
16	16	P16_SI(16)	P16_SIO5_P
17	17	P16_SI(17)	P16_SIO3_P
18	18	P16_SI(18)	P16_SIO1_N

P16 Output Data Register (Write Only) - (BAR0 + 0x301204)

The P16 output data register is used to access the individual LVDS output channels. This includes 9 differential output channels. Each channel is controlled by a corresponding data bit as shown in the P16 Output Data Register Table.

Channel output signal levels are controlled by writing this register. Channel input signals are accessed by reading the P16 input data register at base address plus 0x301200.

This P16 output data register is a write only register. Channel write operations use 32-bit, 16-bit or 8-bit data transfers. All channels of this register are fixed as output channels.

Table 3.33: *BAR0 P16 Output Data Register*

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

Register Bit	Channel	VHDL Name	Schematic Name
0	0	P16_SO(0)	P16_SIO18_GCLK_N
1	1	P16_SO(1)	P16_SIO16_P
2	2	P16_SO(2)	P16_SIO14_P
3	3	P16_SO(3)	P16_SIO12_P
4	4	P16_SO(4)	P16_SIO10_P
5	5	P16_SO(5)	P16_SIO8_P
6	6	P16_SO(6)	P16_SIO6_P
7	7	P16_SO(7)	P16_SIO4_P
8	8	P16_SO(8)	P16_SIO2_P
9	9	P16_SO(9)	P16_SIO0_GCLK_P
10	10	P16_SO(10)	P16_SIO17_N
11	11	P16_SO(11)	P16_SIO15_N
12	12	P16_SO(12)	P16_SIO13_N
13	13	P16_SO(13)	P16_SIO11_N
14	14	P16_SO(14)	P16_SIO9_N
15	15	P16_SO(15)	P16_SIO7_N
16	16	P16_SO(16)	P16_SIO5_N
17	17	P16_SO(17)	P16_SIO3_N
18	18	P16_SO(18)	P16_SIO1_P

BAR2 MEMORY MAP

QDR Memory

16MB of QDR memory is provided on the XMC-6VLX board. The 16MB QDR memory is provided as 2 Meg x 72-bits. The QDR memory connects directly to the Virtex 6 FPGA using a 72-bits. This allow for fast data transfer to and from this memory and the user application and the PCIe bus. This design allows for the user to maximize data throughput between the Field I/O's and the controlling processor.

There is automatic DMA initiator available that will trigger upon a user set condition. See AXI CDMA Registers for more information on DMA operation to and from QDR memory.

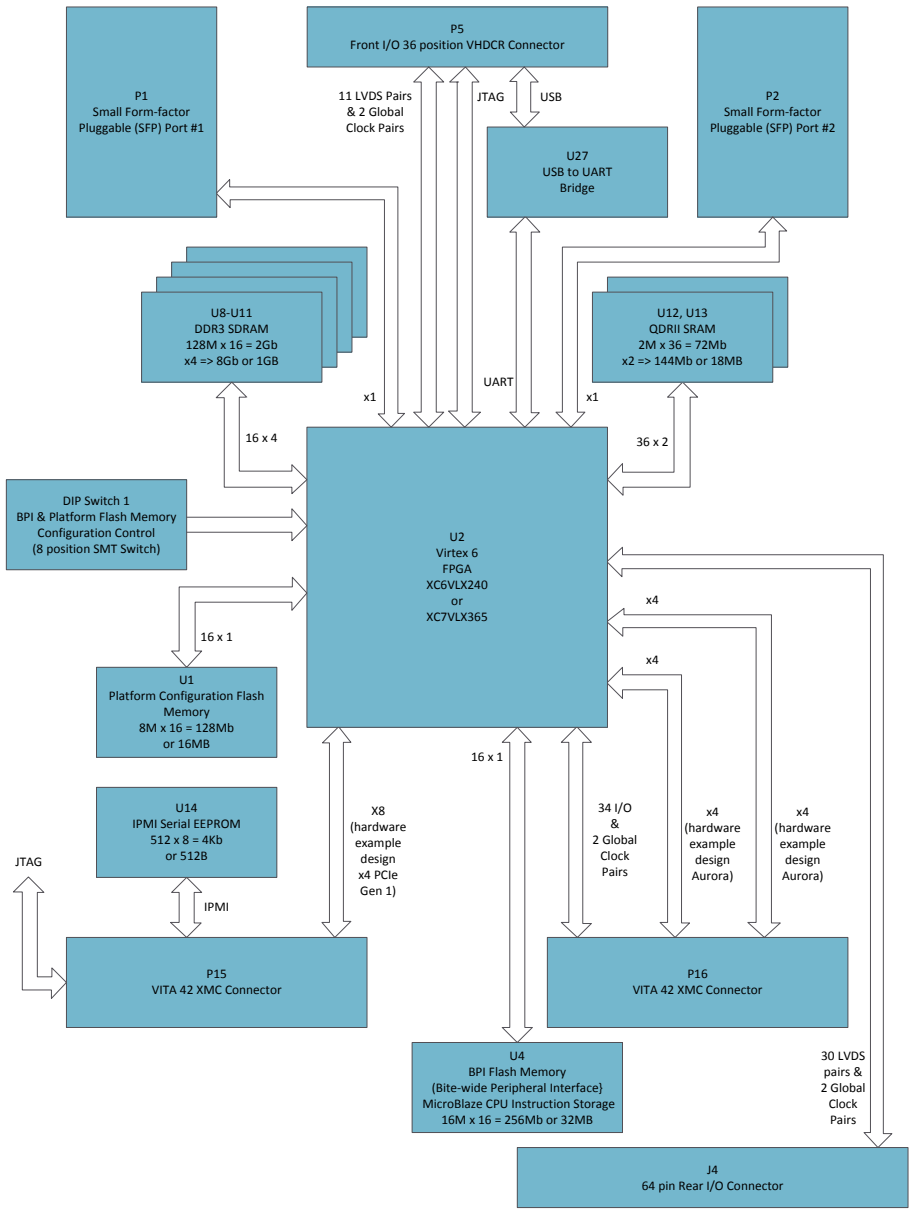
QDR Memory (Read/Write) – (BAR2 + 0x0000000 to 0x00FFFFFF)

The QDR memory 16 Mega byte space is used to provide read or write access to on board QDR memory. This memory space allows access to the QDR directly from the Virtex 6 FPGA. The memory device has a 2 Meg x 72-bits memory configuration. Reading or writing to this memory space using DMA access is also possible as 64-bit transfers.

4.0 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 the XMC-6VLX Block Diagram, shown below as you review this material.

Figure 4.1: *XMC-6VLX Block Diagram*



PCI INTERFACE LOGIC

The Acromag example design PCIe bus interface logic on this board provides a 2.5Gbps interface to the carrier/CPU board per PCI Express Specification v2.0. The interface to the carrier/CPU board allows control of example design board functions.

The PCIe bus endpoint interface logic is contained within the Virtex 6 FPGA. This logic includes support for PCIe commands, including: configuration read/write, and memory read/write. In addition, the PCIe interface requester and or completion accesses. Payload of up to 256 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.

DDR3 Memory

A 128 Meg x 64-bit of DDR3 memory is provided for user applications. Four DDR3 memory devices are used to form a 64-bit data bus. Each of the devices (U8, U9, U10 and U11) are 128 Meg x 16 bit (2Gb) in size. All four device add to 8Gb or 1GByte total memory. The DDR3 interface is implemented in FPGA banks 15, 16, 26, and 36. DCI VRP/N resistor connections are implemented on banks 15 and 36. DCI functionality in bank 15 is achieved in the UCF by cascading DCI between adjacent banks as follows:

```
CONFIG DCI_CASCADE = "15 16";
```

The memory interface logic requires a set of FPGA "No Connect" pins. These are found in the UCF as CONFIG PROHIBIT pins as follows:

```
CONFIG PROHIBIT = A16,D34,F33,K16,K26,L15,N28,N33;
CONFIG PROHIBIT = F31,K14;
```

On board termination devices are provided at the DDR3 device for termination of the address and data as received from the FPGA.

QDR II+ SRAM Memory

A 2 Meg x 72-bit or 16MB of QDRII+ SRAM memory is provided for user applications. Two QDRII+ memory devices are used to form a 72-bit data bus. Each of the devices (U12 and U13) are 2 Meg x 36 bit (72Mb) is size. Both QDR device together total 144Mb or 18MBytes. The QDRII+ interface is implemented in FPGA banks 13, 22, 23, 32 and 33. DCI VRP/N resistor connections are implemented on bank 22. DCI functionality in bank 23 is achieved in the UCF by cascading DCI between adjacent banks as follows:

```
CONFIG DCI_CASCADE = "22 23";
```

On board termination devices are provided at the QDRII+ device for termination of the address and data as received from the FPGA. Termination devices are also provided near the FPGA for QDRII+ data driven signals.

Clock Generation

There are three FPGA fabric clock sources available on the board.

One 2.5V LVDS differential 200 MHz oscillator (U23) is wired to the FPGA global clock input pins D11 and E11. The 200 MHz signal names are clk200_ref_p and clk200_ref_n.

Another 2.5V LVDS differential 125 MHz oscillator (U21) is wired to the FPGA global clock input pins K13 and K12. The 125 MHz signal names are sys_clk_f0_p and sys_clk_f0_n.

Another 2.5V LVDS differential 125 MHz oscillator (U31) is wired to the FPGA MGT clock input pins (H5, H6) and (AD6, AD5). The 125 MHz signal names are SFP_CLK_P and SFP_CLK_N.

Multi-Gigabit Transceivers (GTX MGTs)

The XMC-6VLX provides access to 18 MGTs.

- Eight (8) of the MGTs are wired to the PCIe x8 Endpoint (P15) XMC connector.
- Eight (8) of the MGTs are wired to the (P16) XMC connector.
- Two (2) MGTs are wired to SFP connectors (U16, and U17).

One 2.5V LVDS differential 125 MHz oscillator (U31) is wired to the FPGA MGT clock input pins (H5, H6) and (AD6, AD5). The 125 MHz signal names are SFP_CLK_P and SFP_CLK_N.

SFP Module Connectors

The board contains two small form-factor pluggable (SFP) connectors U16 and U17 and cage assemblies that accept SFP modules. The SFP interfaces are connected to MGT Bank 116 on the FPGA. The SFP module serial ID interface is connected to the FPGA. The control and status signals for the SFP modules are connected to DIP switch and the FPGA. The DIP switch position 7 and 8 to control SFP rate select. Internal FPGA logic controls the SFP signals including Tx_Fault, Rx_LOS, and Mod_ABS. SFP signal Tx_Disable is hard wired to ground.

USB-to-UART Bridge

The XMC-6VLX board contains a Silicon Labs CP2103GM USB-to-UART bridge device (U27) which allows connection from the Virtex 6 device to a host

computer with a USB cable.

Xilinx UART IP is implemented in the FPGA fabric using the Xilinx platform studio UART Lite IP. The FPGA supports the USB-to-UART bridge using four signal pins: Transmit (TX), Receive (RX), Request to Send (RTS), and Clear to Send (CTS). These signals are driven from pins (W32=TX, W25=RX, Y28=CTS, Y27=RTS) of bank 14 of the Virtex 6.

Silicon Labs provides royalty-free virtual COM port drivers which permit the CP2103GM USB-to-UART bridge to appear as a COM port to host computer communications application software (for example, HyperTerm or TeraTerm). The COM port device driver must be installed on the host PC prior to establishing communications with the XMC-6VLX.

With power to the XMC-6VLX board, install the CP2103GM COM port Drivers from www.silabs.com. On the host system set the device manager properties. My Computer -> Properties -> Device Manager. Right-select on USB to UART Bridge -> select Properties. Under the Port Setting tab -> Select Advanced -> Set the COM port to an open Com Port setting. Using HyperTerm or TeraTerm select the same COM port and set the Baud rate to 9600.

16MB Platform Flash

A 16 MByte (128Mb) Xilinx XCF128X-FTG64C Platform Flash device is used with an onboard 48 MHz oscillator to configure the Virtex 6 FPGA in less than 100ms from power valid. This is required by the PCI Express Card Electromechanical Specification. This allows the PCIe interface to be recognized and enumerated when plugged into a host PC.

To achieve the fastest configuration speed, the FPGA mode pins are set to Slave SelectMap (M0=Off, M1=On, M2=On) and the onboard 48 MHz clock source external to the FPGA is used for configuration. Configuration DIP switch 1, switch 4, controls the 48 MHz oscillator enable (switch 4 = Off enables the oscillator). Also see the Configuration Control register at BAR0 + 0x300100 bit-2 for control of this signal.

32MB Linear BPI Flash

A Linear BPI Flash memory on the board provides 32 MByte of non-volatile storage that can be used for MicroBlaze program code or data storage. The Linear BPI Flash shares the dual use flash data, address and control pins in parallel with the XCF128 Platform Flash.

The BPI_Flash net is used to select the BPI Flash or the XCF128 Platform Flash. Power-on configuration is selected by the BPI_Flash net which is tied to DIP switch position 5 and is also wired to an FPGA pin. DIP switch position 5 set On (closed) select the BPI Flash while Off (open) select the Platform Flash. The DIP switch selection can be overridden by the FPGA after configuration by controlling the logic level of the BPI_Flash net. Logic

high on this net/pin selects BPI Flash device. See the Configuration Control register at BAR0 + 0x300100 bit-0 for control of this signal.

Configuration Flash Design Considerations

After FPGA configuration, the FPGA design can disable the configuration flash or access the configuration flash to read/write code or data.

When the FPGA design does not use the configuration flash, the FPGA design should drive the FPGA BPI_Flash pin high in order to disable the configuration Platform flash and put this flash into a quiescent, low-power state. Otherwise, the Platform Flash, can continue to drive its array data onto the data bus causing unnecessary switching noise and power consumption.

To drive the FPGA BPI_Flash pin high set the Configuration Control register at BAR0 + 0x300100 bit-0 to logic high.

JTAG Port

The JTAG port can be used to program the Virtex 6 FPGA and access the device for hardware and software debug. The default

The JTAG port also allows a host computer to download bitstreams to the FPGA using the Xilinx iMPACT 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.

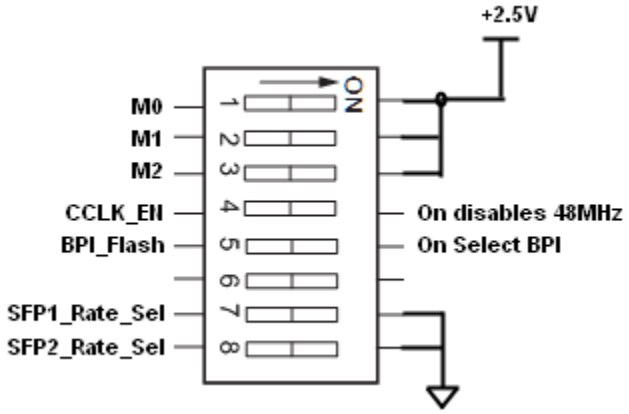
Through the connection made by the temporary design in the FPGA, iMPACT can indirectly program the BPI flash or the Platform Flash from the JTAG port.

DIP Switch

DIP switch DIP1A is a multi-purpose switch.

- FPGA Mode: switches 1, 2, and 3 control the FPGA mode. The supported configuration methods, Slave SelectMAP, Master BPI-Up, and JTAG are selected by setting M[2:0] options.
- CCLK_En: switch 4, controls the enable pin of the 48 MHz oscillator (U3). When switch 4 is open/off U3 drives a 48 MHz clock onto the FPGA_CCLK signal.
- BPI_Flash: switch 5, is use to select between the Xilinx Platform Flash or the Linear BPI Flash for the FPGA boot memory device.
- SFP_Rate_Sel: switch 7 and 8 control the rate select to SFP modules 1 and 2 respectively.

Figure 4.2: *Multi-Purpose Select DIP Switch*



The supported configuration methods, Slave SelectMAP, Master BPI-Up, and JTAG are selected by setting M[2:0] options (of the 8-position DIP switch) as shown in the following table.

Table 4.1: *Configuration Details*

In JTAG mode switch5 On selects FPGA access to BPI Flash. Alternatively, set switch5 Off for FPGA access to the Platform Flash.

Switch	Name	Configuration Mode		
Switch	**Name**	**Slave SelectMAP**	**BPI Flash**	**JTAG**
1	M0	Off	Off	On
2	M1	On	On	Off
3	M2	On	Off	On
4	CCLK_EN	Off(48 MHz)	On (FPGA)	On (TCK)
5	BPI_Flash	Off	On	On

Power System Devices

The power to the XMC-6VLX is taken from the XMC P15 connector VPWR_5/12 pins. The VPWR_5/12 power is the V in voltage to the four LTM4602 devices U18, U19, U20, and U25. These LTM4602 devices output +1.0V, +1.5V, +2.5V and +1.8V voltages. The +1.5 V supply is input to both

source/sink linear regulators TPS51120 devices U15 and U30. The +1.8 V supply is input to both MIC61300 devices U22 and U24.

Figure 4.3: *Power Distribution*

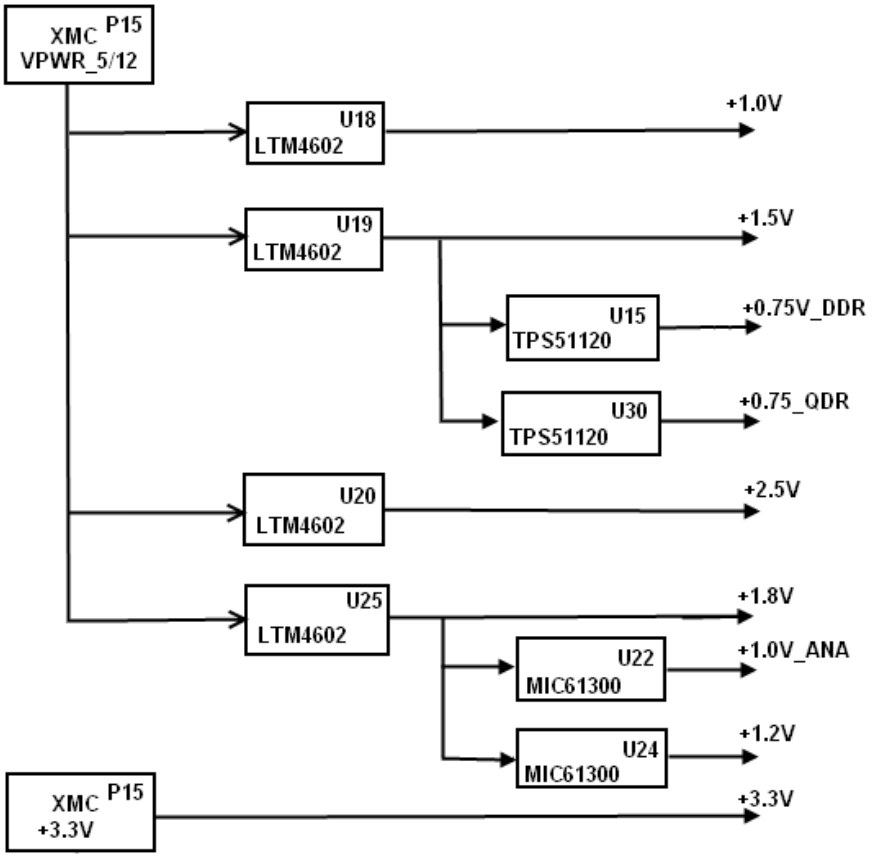


Table 4.2: *Power System Devices*

Device	Reference Designator	Description	Power Rail Name	Power Rail Current
LTM4602	U18	FPGA VCCINT	+1.0V	6.0 A
LTM4602	U19	VCCO, DDR3, QDRII+	+1.5V	6.0 A
TPS51120	U15	DDR3 Termination	+0.75V_DDR	+/-2 A
TPS51120	U30	QDRII+ Termination	+0.75V_QDR	+/-2 A
LTM4602	U20	VCCO, Flash	+2.5V	6.0 A
LTM4602	U25	Flash, MGT	+1.8V	6.0 A
MIC61300	U22	MGT	+1.0V_ANA	3.0 A
MIC61300	U24	MGT	+1.2V	3.0 A

System Monitor

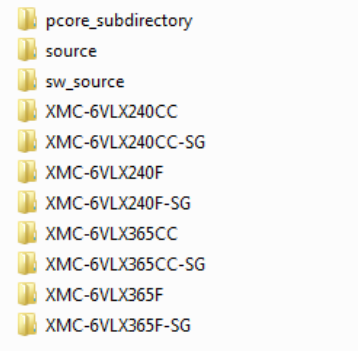
The System Monitor provides information regarding the Virtex 6 device temperature and power supply conditions via JTAG and the PCIe bus interface. The system monitor is located in the center of the Virtex 6 die. The System Monitor function is built around a 10-bit, 200-kilosamples per second Analog-to-Digital Converter. The system monitor is used to measure FPGA physical operating parameters like on-chip power supply voltages and die temperature.

5.0 XPS Embedded System

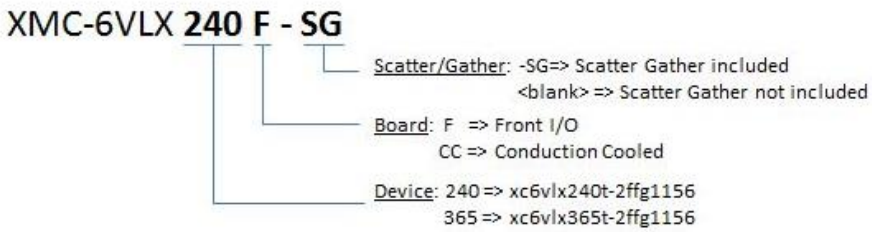
The example design consists of a Xilinx ISE project with an embedded XPS project. It includes interfaces to all of the peripheral components connected to the FPGA. Driver software is also supplied that exercises the peripheral components using the host processor through the PCI Express bus or the embedded Microblaze processer. It is expected that an XMC-V6 user's project will require a subset of the example project's interfaces in addition to user specific functions. Acromag recommends beginning a new user project by copying the Acromag example project, deleting any unnecessary peripherals, and then adding the user defined functionality. The XMC-V6 block diagram shows the structure of the example system with peripherals that are accessible from the microblaze processor, peripherals that are accessible from the host processor, and peripherals that are common to both the microblaze processor and the host processor. Master interfaces are shown with a blue square symbol. Slave interfaces are shown with a red circle symbol.

Xilinx ISE Example Projects

The Acromag EDK delivers prepackaged example designs to program the Xilinx Virtex 6 FPGA. The XMC-V6 project folder will contain the following subdirectories:



The naming convention used for the projects are as follows:



Special note regarding the *-SG* projects. SG stands for Scatter Gather and is a

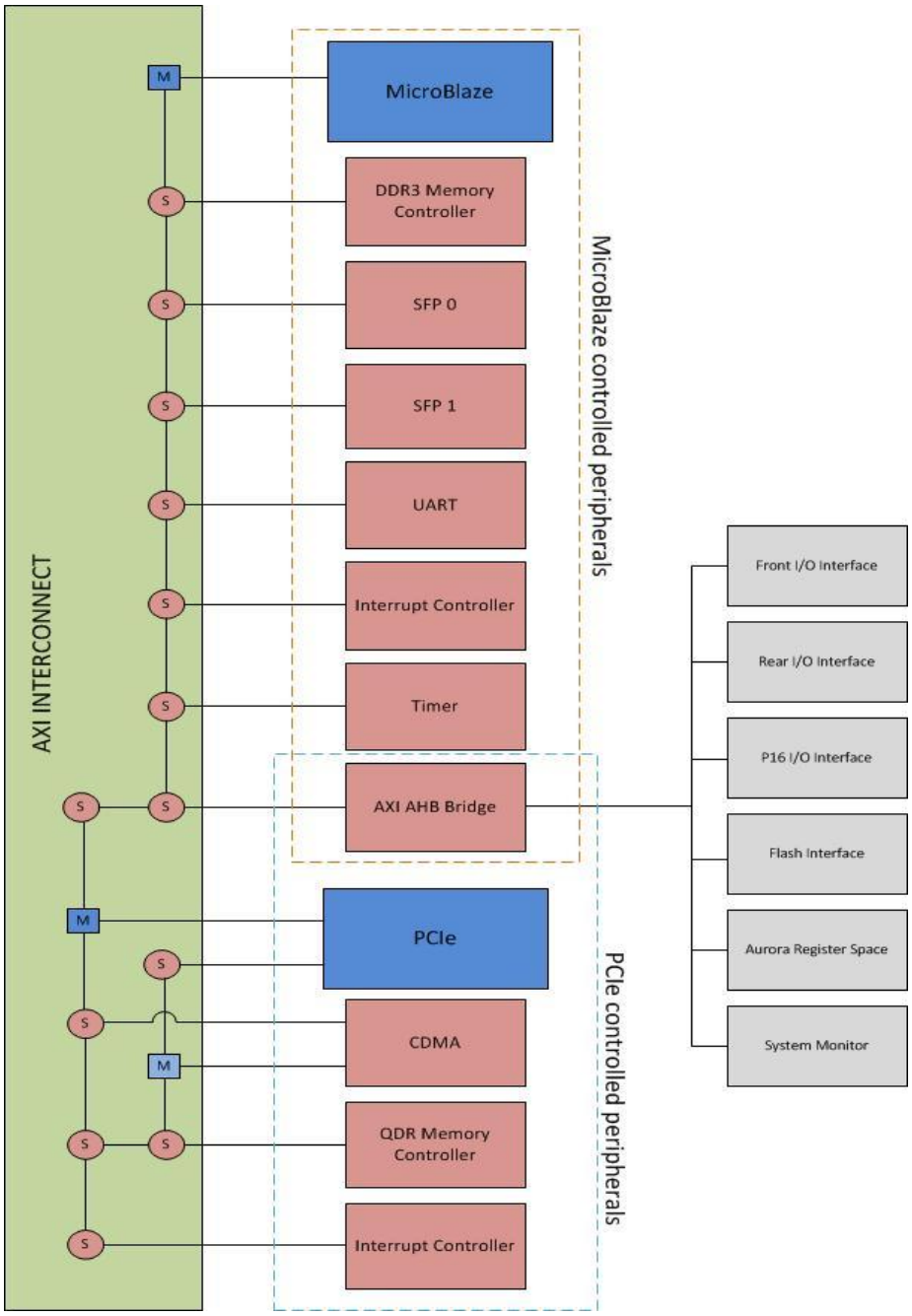
feature of the Xilinx AXI-CDMA core. Because of a limitation in the Xilinx AXI Interconnect, it is not possible to connect buses larger than 32 bits to the PCIe control bus (where PCIe registers reside). AXI PCIe control bus is an AXI4 Lite protocol and can only support single 32 bit transactions. The AXI CDMA bus is an AXI4 Full protocol capable of bursts of various lengths. (see "AMBA AXI Protocol Specification" for further details).

For some users, it may be desirable to use Scatter-Gather mode, in which case a descriptor list can be set up in QDR memory to move data to or from host memory. In order for the AXI CDMA core to move data from the QDR memory to or from host memory, it must first write the base address translation registers in the AXI PCIe core with an address translation. In this mode, the AXI CDMA would need to be connected to the AXI PCIe Control Bus but the data width of the CDMA bus must be set to 32 bits.

However, if Scatter-Gather is not essential, it can be disabled and the data width of the AXI CDMA can be set to 64 bits (or larger) to improve through-put.

Acromag includes both Scatter-Gather and no Scatter-Gather versions of the projects to illustrate both examples.

Figure 4.1: *Internal FPGA Functions*



Microblaze Peripherals

The embedded system consists of the following IP blocks: Microblaze processor, DDR3 SDRAM, two Ethernet SFP ports with supporting DMA controllers, UART, interrupt controller, timer, and the common peripherals over the AXI to AHB bridge.

The embedded MicroBlaze processor is used to echoes packets sent to its Ethernet interface. There are three major components within the

embedded system that make up the Ethernet interface :

Ethernet core VHDL source: axi_ethernet_v3_01_aC:\XMC-V6\pcore_subdirectory\pcores\axi_ethernet_v3_01_a\hdl\vhdl

DMA controller VHDL source: axi_dma_v6_01_aC:\XMC-V6\pcore_subdirectory\pcores\axi_dma_v6_01_a\hdl\vhdl

And software running on the microblaze : lwip_echoserverC:\XMC-6VLX365F\SDK\lwip_echo_server_dual\src

Xilinx provides documentation for each of these XPS core components. The Xilinx Documentation Navigator tool can be used to find Xilinx provided documentation.

The VHDL source for these components has had a few minor changes made to what Xilinx provided. These changes were necessary in order to get two Ethernet interfaces to work correctly in a single system. Some software modifications were also necessary for the same reason.

Examine the configuration of the hardware components axi_ethernet and axi_dma in Xilinx Platform Studio. Knowing how the hardware is configured will allow you to focus on the particular sections of the Xilinx documentation that are pertinent.

Host Peripherals

The host processor can access the following peripheral devices and IP blocks through the PCI Express interface: common peripherals, Central DMA and QDRII+ SRAM, and an interrupt controller.

Common Peripherals

The following common peripherals can be accessed from either the host processor through the PCI Express interface, or the embedded Microblaze processor: Front I/O, Rear I/O, P16 I/O, FLASH, Aurora registers, and the system monitor.

SDK

The following Microblaze programs are included in the example project: hello_world, lwip_echo_server_dual, and srec_bootloader.

hello_world

Hello_world - transmits the text "hello world" over the UART interface. Connect a USB cable to a host PC that has the USB/UART device driver installed (see section USB-to-UART Bridge in the Theory of Operation Chapter). Start a terminal emulator program such as hyperlynx and see the text displayed on the terminal. The serial interface parameters are: 9600 baud, 8 data bits, one stop bit, no parity.

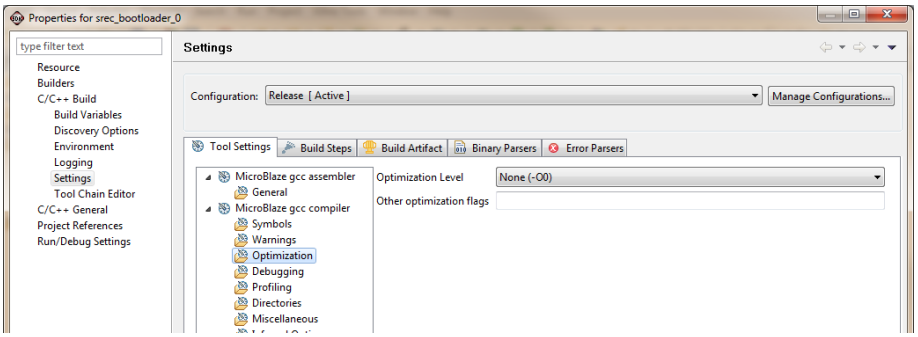
lwip_echo_server_dual

lwip_echo_server_dual – echoes TCP/IP packets sent to it over the Ethernet ports. This program is used to test the Ethernet ports. Connect the appropriate copper or fiber Ethernet cable to a host PC and "ping" the XMC-V6. The IP address is 192.168.1.10, subnet mask 255.255.255.128 for port 1, and 192.168.1.138, subnet mask 255.255.255.128 for port 2. Make sure that the host PC subnet address is also 255.255.255.128 and the host PC IP address is in the same subnet as the XMC-V6 port that it is attached to.

srec_bootloader

srec_bootloader – This program is embedded in the example FPGA configuration bitstream. It is loaded into block RAM memory at configuration time and begins executing when reset is released. It copies a microblaze program stored in BPI FLASH (in Motorola "S" record format) to DDR3 memory, and then executes the program. See section "Error! eference source not found." for instructions on copying a program to FLASH memory.

IMPORTANT! When building the Acromag modified version of the SREC_BOOTLOADER program in the Xilinx SDK tool, select the Optimization Level to 'None (-O0)' in the C/C++ Build Settings of the project as shown below.



peripheral_tests

peripheral_tests – Runs a basic test on each other peripherals instantiated in the XPS system. The results of each test are displayed on the terminal.

Acromag Peripheral Repository

Acromag has modified a few of the Xilinx supplied peripheral cores or created custom cores to support XMC-V6 specific requirements. The following cores are included in the pcore_subdirectory: axi_ethernet, axi_pcie, axi_enhanced_pcie, axi_to_qdr_mc, util_bufr_core, axi_cdma, util_ds_buf and util_ds_buf_mgtclk. These are used in the example design and will take priority over the Xilinx cores because of the "Project Peripheral Repository Search Path" option in the "Project Options" menu of the ISE tool points to this repository.

Modifications to the AXI Ethernet Core

The AXI Ethernet pcore supplied by Xilinx includes a regional clock buffer instantiated in a lower level of the hierarchy. This is okay when a single instance of the pcore is included in the design, but the XMC-V6 has two Ethernet ports that must be located in the same clock region. There is only one regional clock buffer available in a region, so instantiating two of the Xilinx supplied axi_ethernet pcores resulted in a map error. The regional clock buffer had to be moved from the lower level to the top level so that it could be shared between two instances of axi_ethernet. The pcore is configured for a 1000Base-X physical interface at 1 Gbs. This configuration is compatible with both the 1000Base-X and the 1000Base-T SFP modules available from Acromag.

util_bufr_core Core

A pcore was created to instantiate a regional clock buffer in XPS. This clock buffer was required by the modified axi_ethernet pcore.

util_buf_ds_mgtclk Core

A pcore was created to instantiate an IBUFDS_GTXE1 clock buffer in XPS. This clock buffer was required by the modified axi_ethernet pcore.

AXI QDRII+ Memory Controller

Xilinx did not provide a complete AXI interface (in ISE 14.1) to the memory controller when it is configured for a QDR II+ application. Acromag has provided the missing functionality. This AXI QDR II+ pcore has separate 256 bit read and write interfaces. It supports continuous simultaneous 125 MHz read and write bursts.

Modifications to the CDMA Core

Fixed a bug in the TCL script that failed to close an output file upon termination (in ISE 14.1). The supported devices list was modified to include only the Virtex 6.

Modifications to the AXI PCIe Core

Modifications to the AXI PCIe core were necessary to separate the reset from the rest of the AXI system. When resetting the AXI system, form example through MicroBlaze, the PCIe configuration space would also be reset preventing the host from communicating with the FPGA through the AXI PCIe core until a power-cycle of the board was done. The axi_enhanced_pcie core is a Xilinx "helper core" for the AXI PCIe core and was modified to bring the 250MHz clock to the top level which was generated by a MMCM internal to this core.

Modifications to the util_ds_buf

Modifications to the MPD (Microprocessor Definition) file were necessary to this core to allow the frequency of the connected clock to be passed to the Xilinx EDK/XPS tool in order to derive timing constraints clocks and signals downstream. Changes were done in accordance to Xilinx Answer Record #42642. The Acromag design uses this core to buffer the 100MHz differential PCIe Reference Clock to the AXI PCIe core.

Acromag modified library files

Acromag has modified two of the Xilinx supplied source files used to build a board support package that includes the Lightweight Internet Protocol (LWIP) library. The files are located in the folder C:\XMC-V6\sw_source\modified xilinx library. The file names are *xaxiemacif_dma.c* and *xpqueue.c*. These files have been modified to support multiple instances of the Ethernet interface. As delivered by Xilinx, the library would only support a single instance of an Ethernet interface. These files must be copied to the following directory to overwrite the existing files delivered through the Xilinx installer:

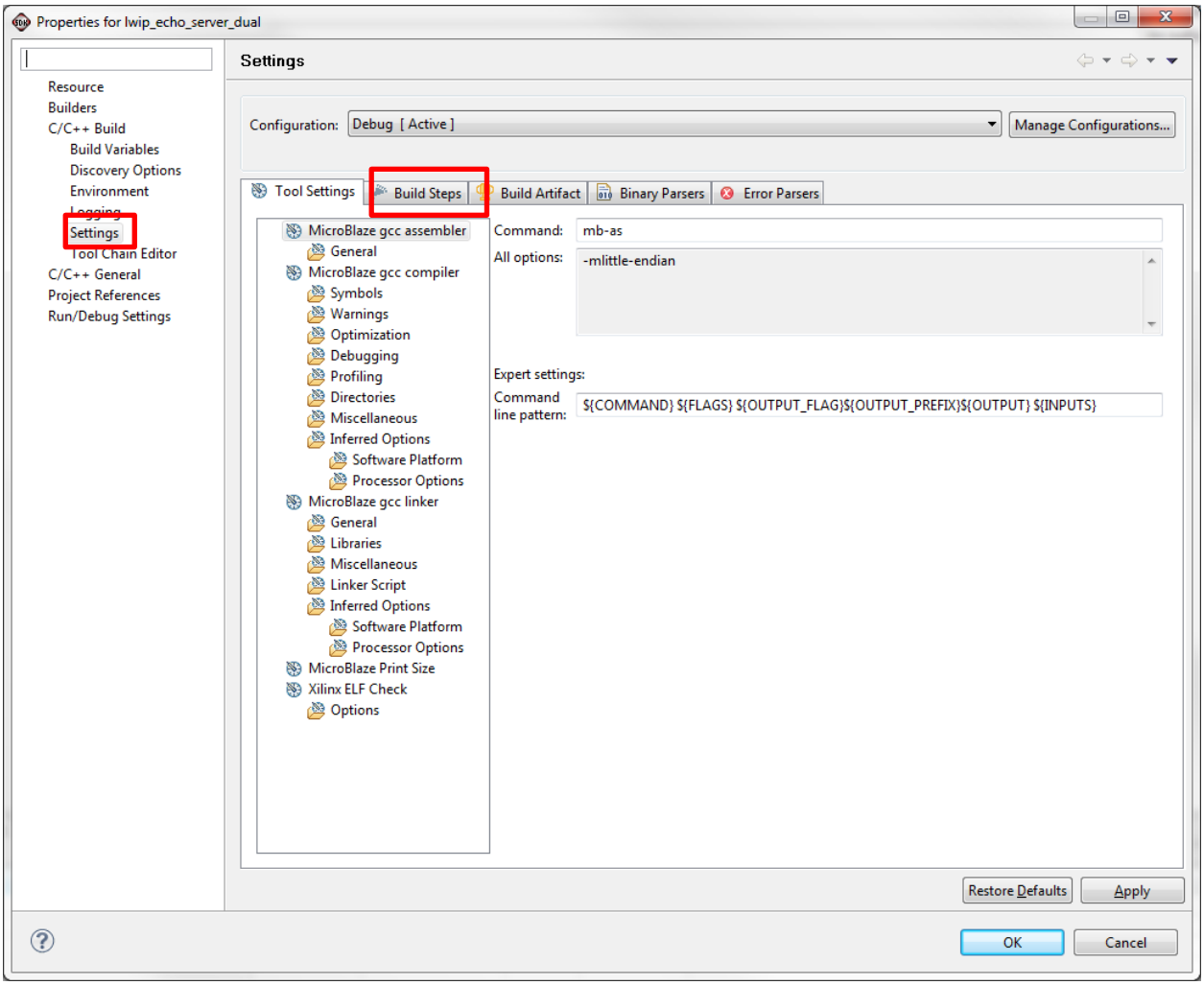
C:\Xilinx\14.2\ISE_DS\EDK\sw\ThirdParty\sw_services\lwip140_v1_02_a

src\contrib\ports\xilinx\netif

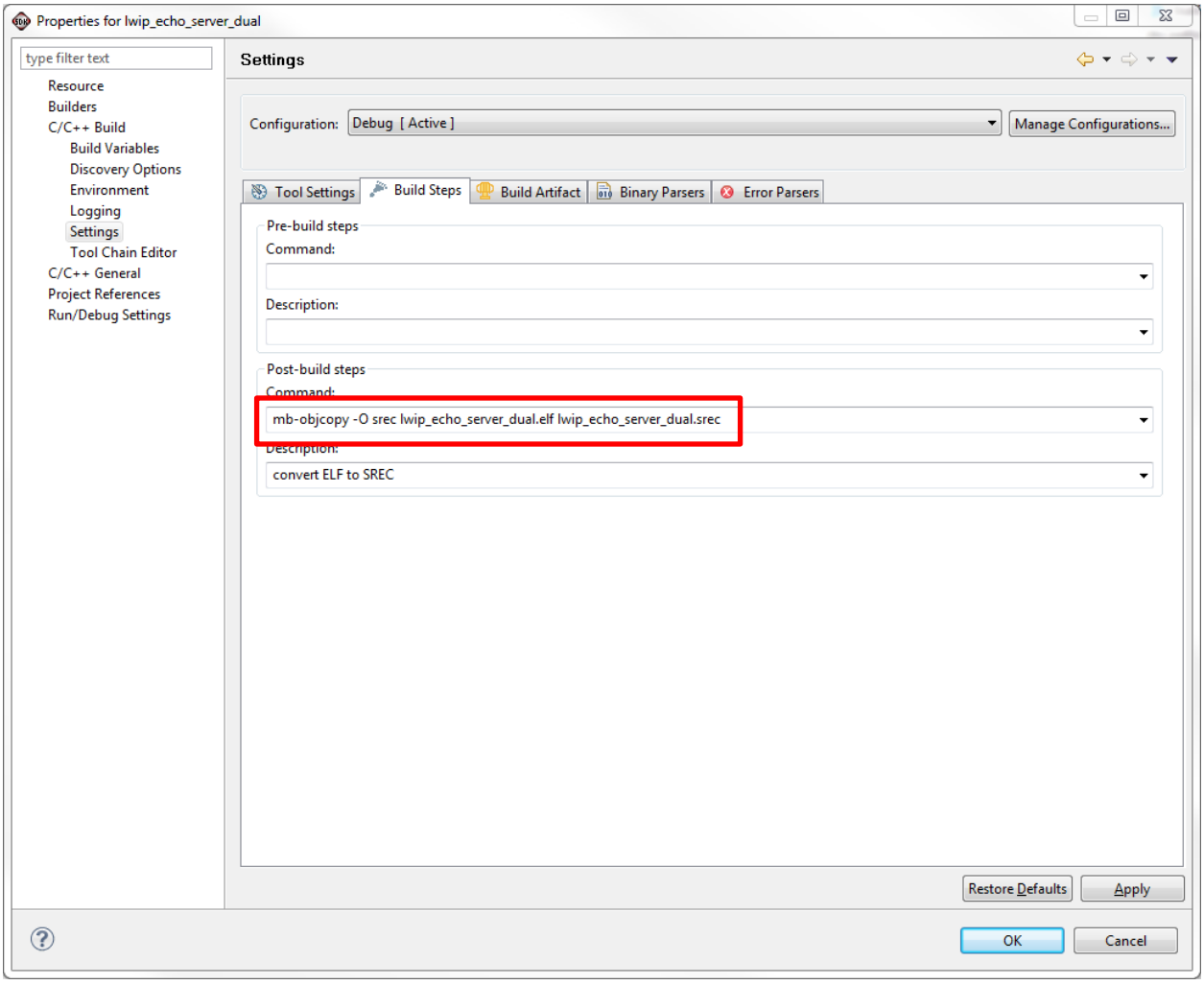
Running a Program from BPI FLASH Memory

This section describes the steps required to create an "S" record file that can be loaded into FLASH memory and executed upon initial application of power.

From within SDK, open the project properties to display the following window:



Enter the highlighted text in the post-build command line, substituting your project name for lwip_echo_server_dual:


Properties for lwip_echo_server_dual
type filter text
Resource
Builders
C/C++ Build
Build Variables
Discovery Options
Environment
Logging
Settings
Tool Chain Editor
C/C++ General
Project References
Run/Debug Settings
Settings
Configuration: Debug [Active]
Manage Configurations...
Tool Settings
Build Steps
Build Artifact
Binary Parsers
Error Parsers
Pre-build steps
Command:
Description:
Post-build steps
Command:
mb-objcopy -O srec lwip_echo_server_dual.elf lwip_echo_server_dual.srec
Description:
convert ELF to SREC
Restore Defaults
Apply
OK
Cancel

Build the project to produce the "S" record file.

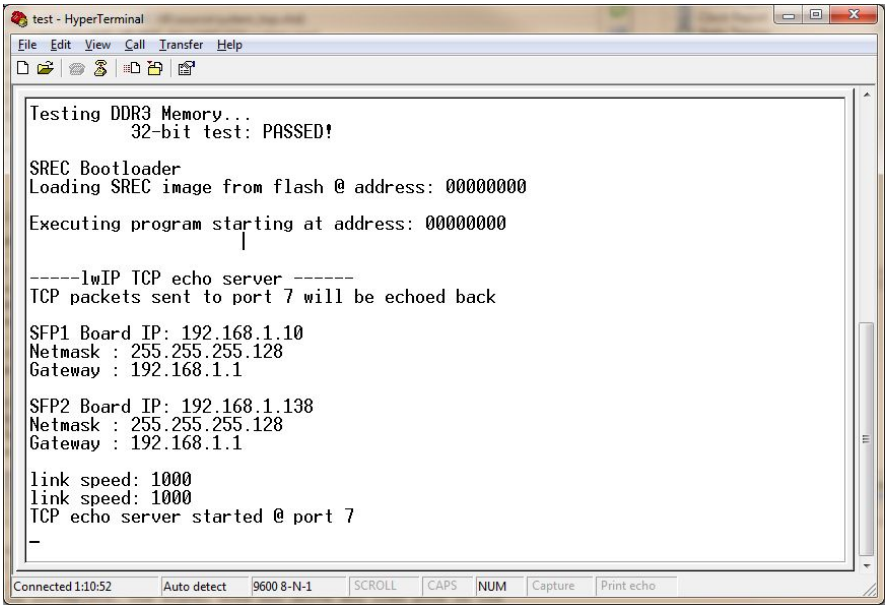
Use the Acromag PCIe 6VLX demo program pertaining to your operation system (window, link, or VxWorks).

Select the Flash programming menu.

Within the Flash programming menu Choose the BPI FLASH.

Enter the path to the program lwip_echo_server_dual.srec file you would like to write to the FLASH. On the next power cycle the boot-loader will copy the program from FLASH to DDR3 memory and execute the program from DDR3 memory.

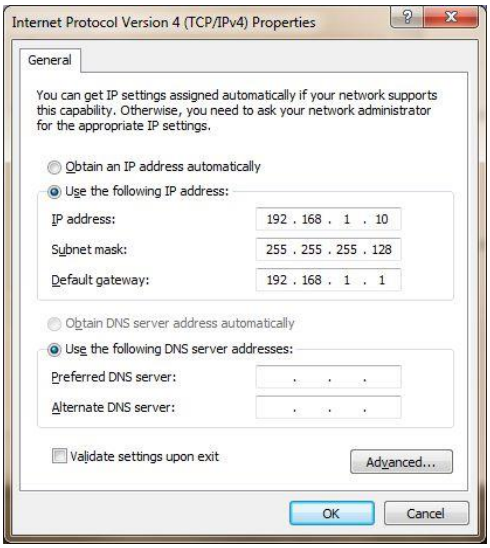
Please note, a modified version of the Xilinx bootloader code is embedded into the MCS file loaded into the Virtex 6 FPGA. The bootloader will move program data from BPI Flash into DDR3 memory to be executed. It will check the DDR3 memory for any problems with the S-Record and MicroBlaze will begin executing that program. If for example, the LWIP program is loaded into BPI Flash and a terminal emulation program such as HyperTerminal is running, power-cycling the board should result in the following being displayed:



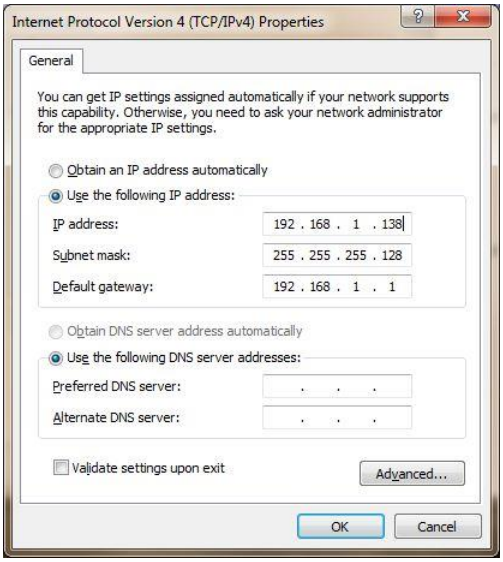
Running LWIP Echo Server

This section describes some helpful tips on how to run the Echo Server program. As mentioned above, in the lwip_echo_server_dual description of the Error! Reference source not found. project, the network adaptor ettings on the host PC(s) need to be modified. The hardware will support both SFP modules (on XMC-6VLX240/365F) to work at the same time. In a Windows OS, simply access the Local Area Connection Properties and modify the Internet Protocol Version 4 (TCP/IPv4) settings to the following:

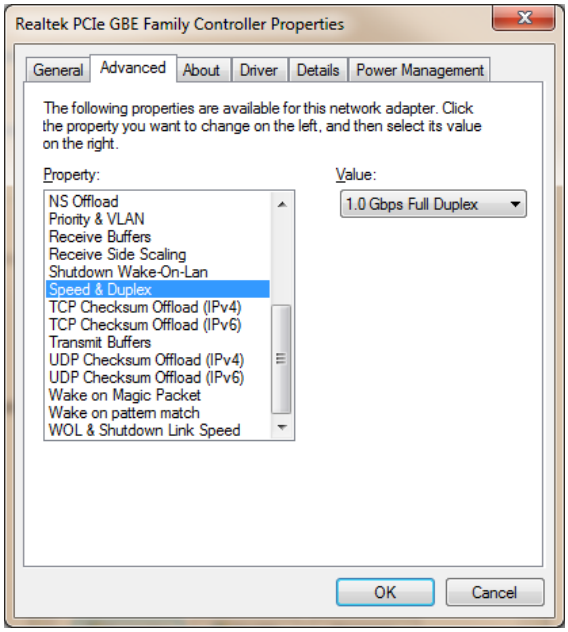
SFP1:



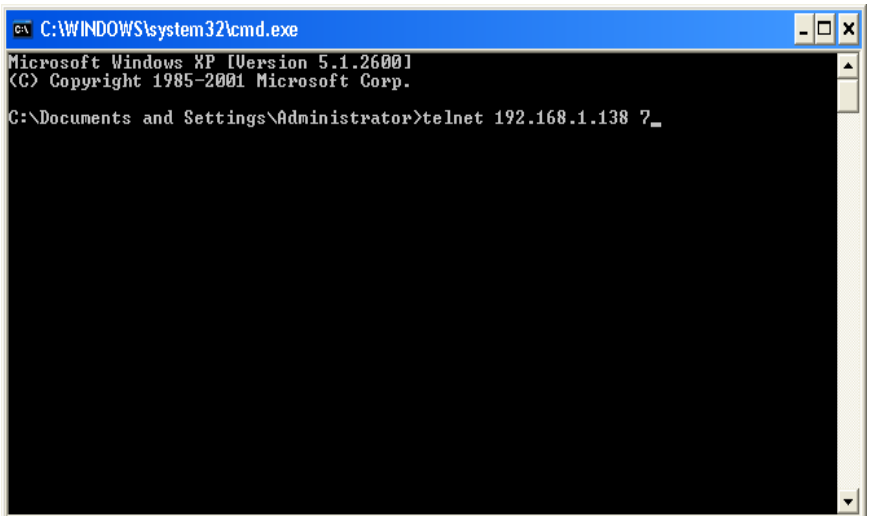
SFP2:

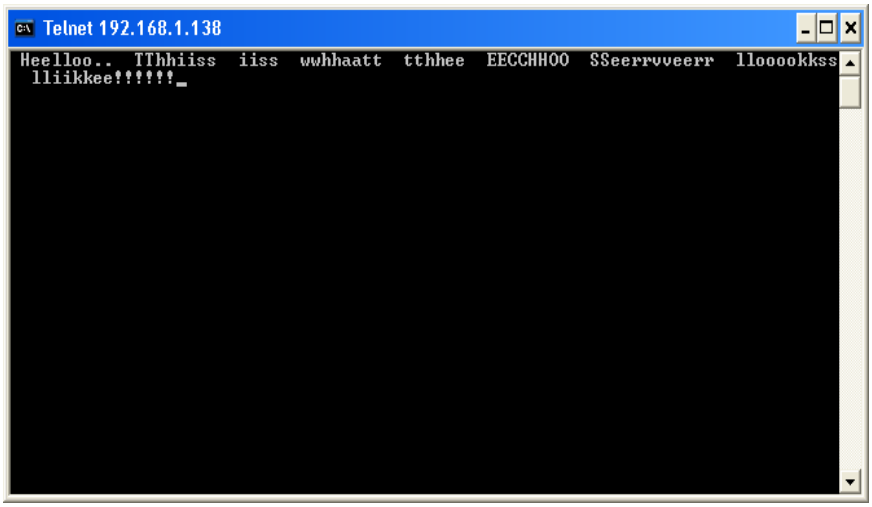


Also, the Network Adaptor speed must be set to 1.0Gbps Full Duplex as shown:



Next, run the program either directly from SDK (see "EDK Concepts, Tools and Techniques" document from www.xilinx.com) or as described above in the "Error! Reference source not found.." When the program is running on he Virtex 6 FPGA, Ping or Telnet can be used to communicate with the Ethernet cores inside the FPGA. Ping requests can be made by typing *ping 192.168.1.XXX* into a Window CMD terminal. This will send out ICMP (Internet Control Message Protocol) packets and the Ethernet core inside the Xilinx Virtex 6 FPGA will send back acknowledgments. Telnet can be used to start the Echo Server program. Telnet is a TCP based protocol that will and this test will echo back any information sent to the Ethernet cores. See below for an example of how to run the Echo Server and for more information on Ethernet applications see the Xilinx document xapp1026.pdf available at http://www.xilinx.com.





EDK File Organization

The XMC-V6 EDK design files are organized by the following directory structure:

pcore_subdirectory This directory contains the design files for the Acromag modified pcores described in the above section.

source The source directory contains the VHDL source files for the common peripherals.

sw_soure The sw_source directory contains the Xilinx SDK library files that were modified by Acromag to suit the XMC-V6 product.

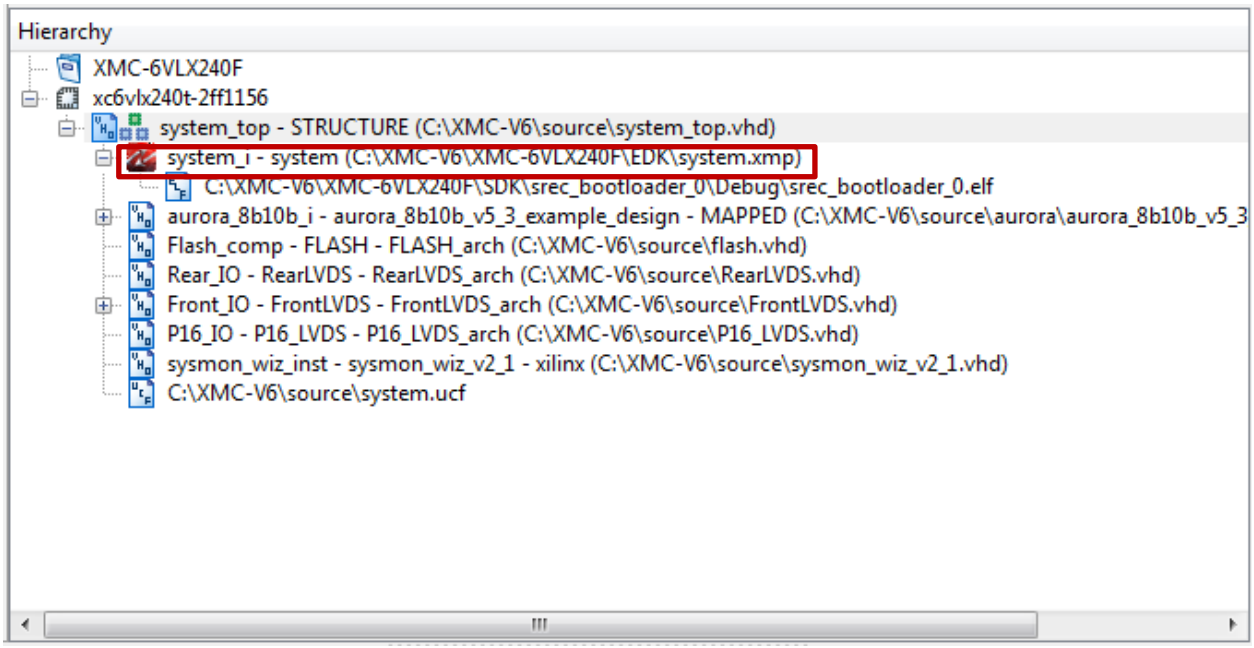
XMC-6VLX240F This directory is the parent for design files that are specific to the particular variant of XMC-V6. Subdirectories further organize the design files into Xilinx tool specific folders (EDK, iMPACT, ISE and SDK) which include the tool specific project files (.xmp, .xise, or .ipf).

Example EDK Design Modification Walkthrough

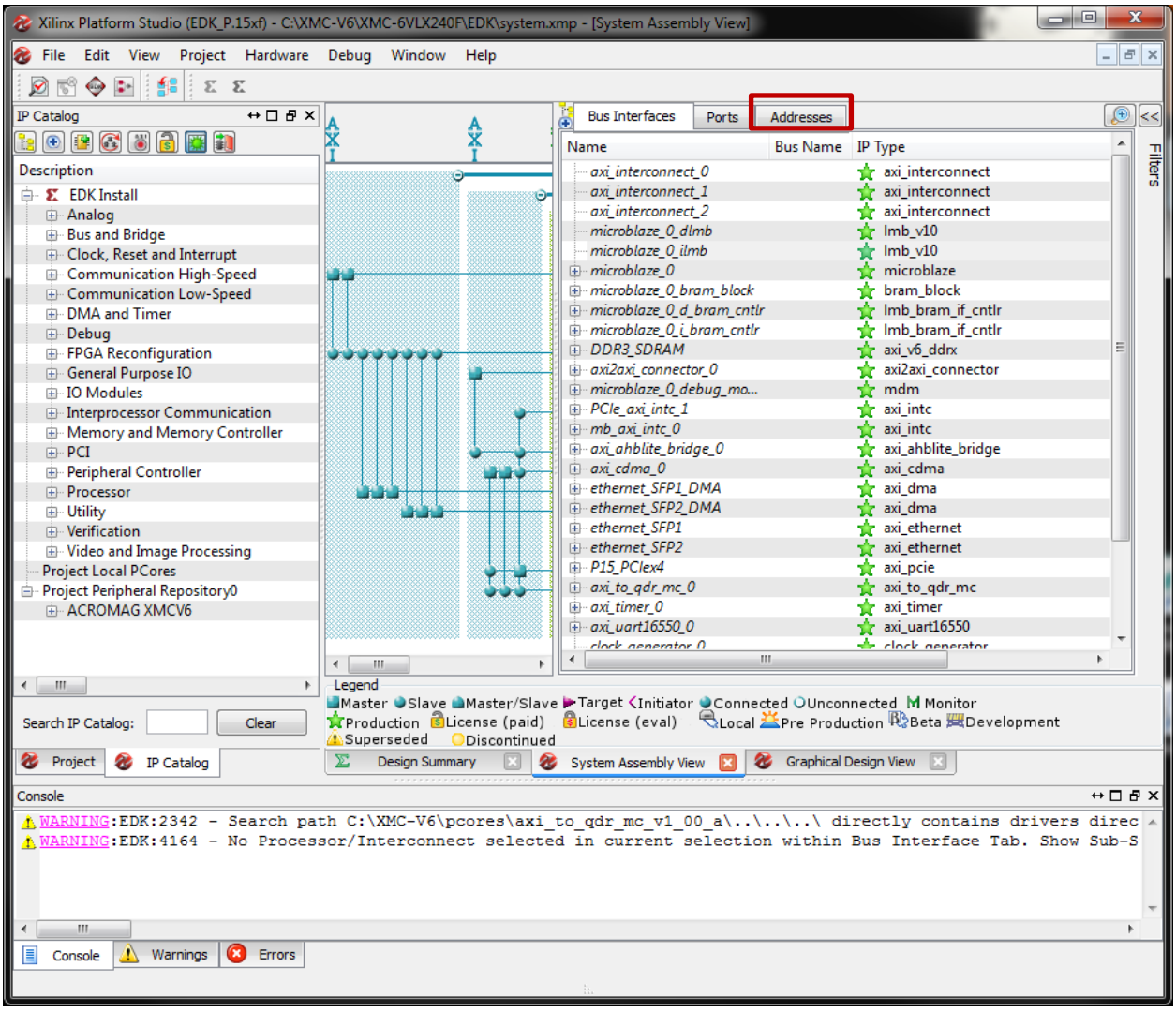
This section describes in detail the steps needed to re-locate the DDR3 SDRAM in the address space.

First, let's take a look at the original address map. Open ISE then navigate to the directory C:\XMC-V6\XMC-6VLX240F\ISE\XMC-6VLX240F. Open the project file XMC-6VLX240F.xise

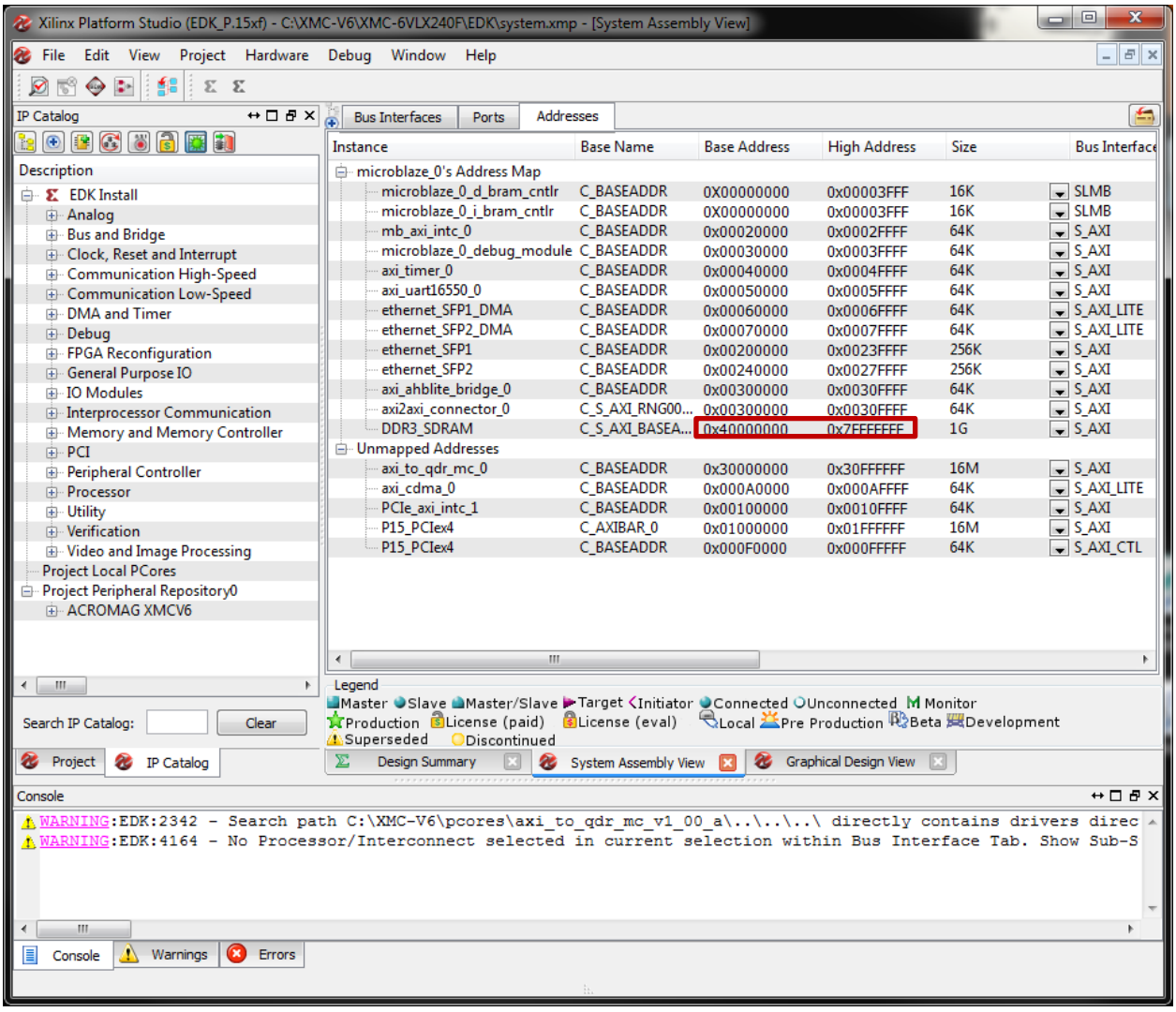
In the hierarchy pane right click on system_i and select "open" from the pop-up menu to open the embedded system in the EDK tool.


Hierarchy
XMC-6VLX240F
xc6vlx240t-2ff1156
system_top - STRUCTURE (C:\XMC-V6\source\system_top.vhd)
system_i - system (C:\XMC-V6\XMC-6VLX240F\EDK\system.xmp)
C:\XMC-V6\XMC-6VLX240F\SDK\srec_bootloader_0\Debug\srec_bootloader_0.elf
aurora_8b10b_i - aurora_8b10b_v5_3_example_design - MAPPED (C:\XMC-V6\source\aurora\aurora_8b10b_v5_3
Flash_comp - FLASH - FLASH_arch (C:\XMC-V6\source\flash.vhd)
Rear_IO - RearLVDS - RearLVDS_arch (C:\XMC-V6\source\RearLVDS.vhd)
Front_IO - FrontLVDS - FrontLVDS_arch (C:\XMC-V6\source\FrontLVDS.vhd)
P16_IO - P16_LVDS - P16_LVDS_arch (C:\XMC-V6\source\P16_LVDS.vhd)
sysmon_wiz_inst - sysmon_wiz_v2_1 - xilinx (C:\XMC-V6\source\sysmon_wiz_v2_1.vhd)
C:\XMC-V6\source\system.ucf

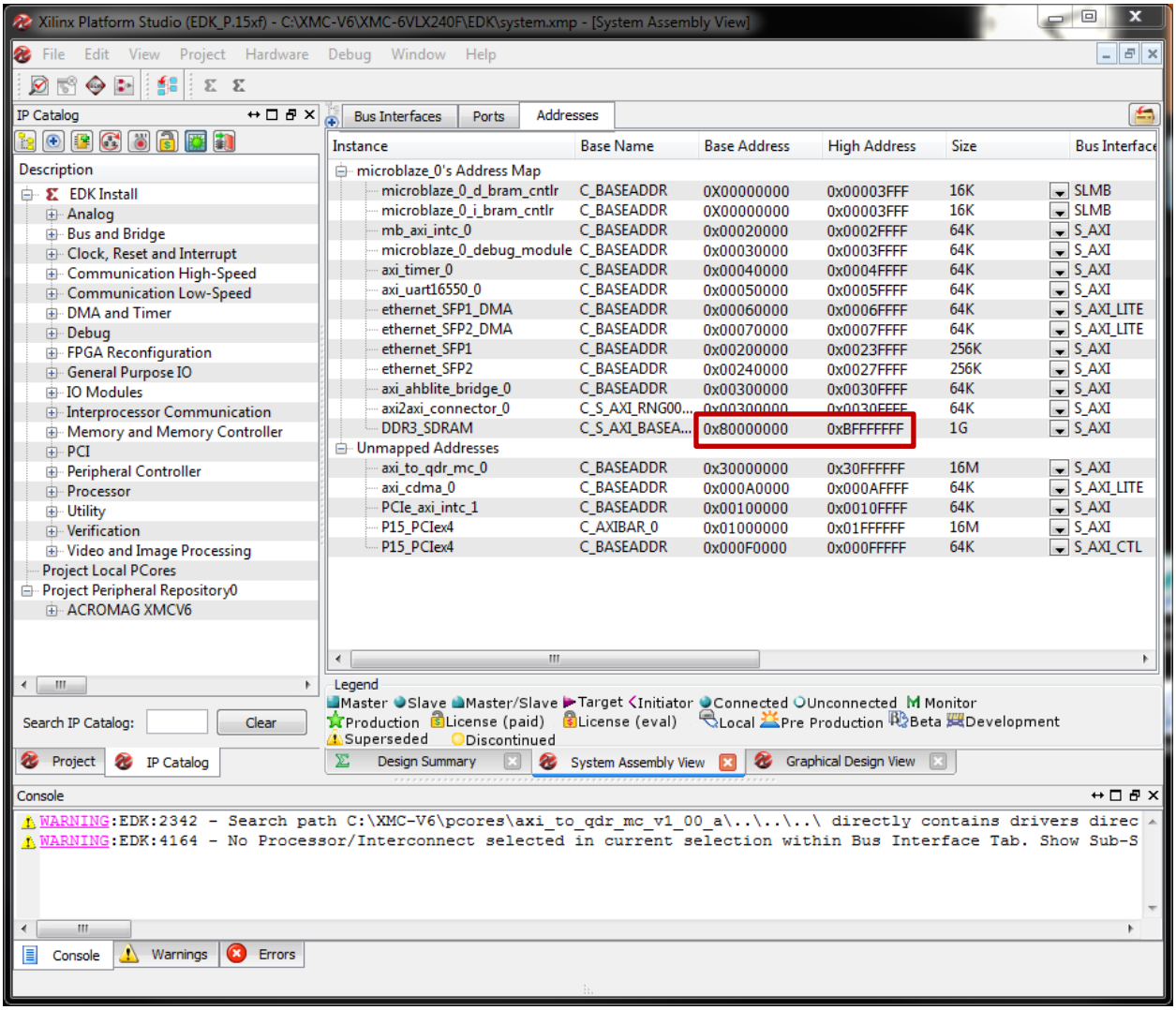
Click on the "address" tab to display the address map.



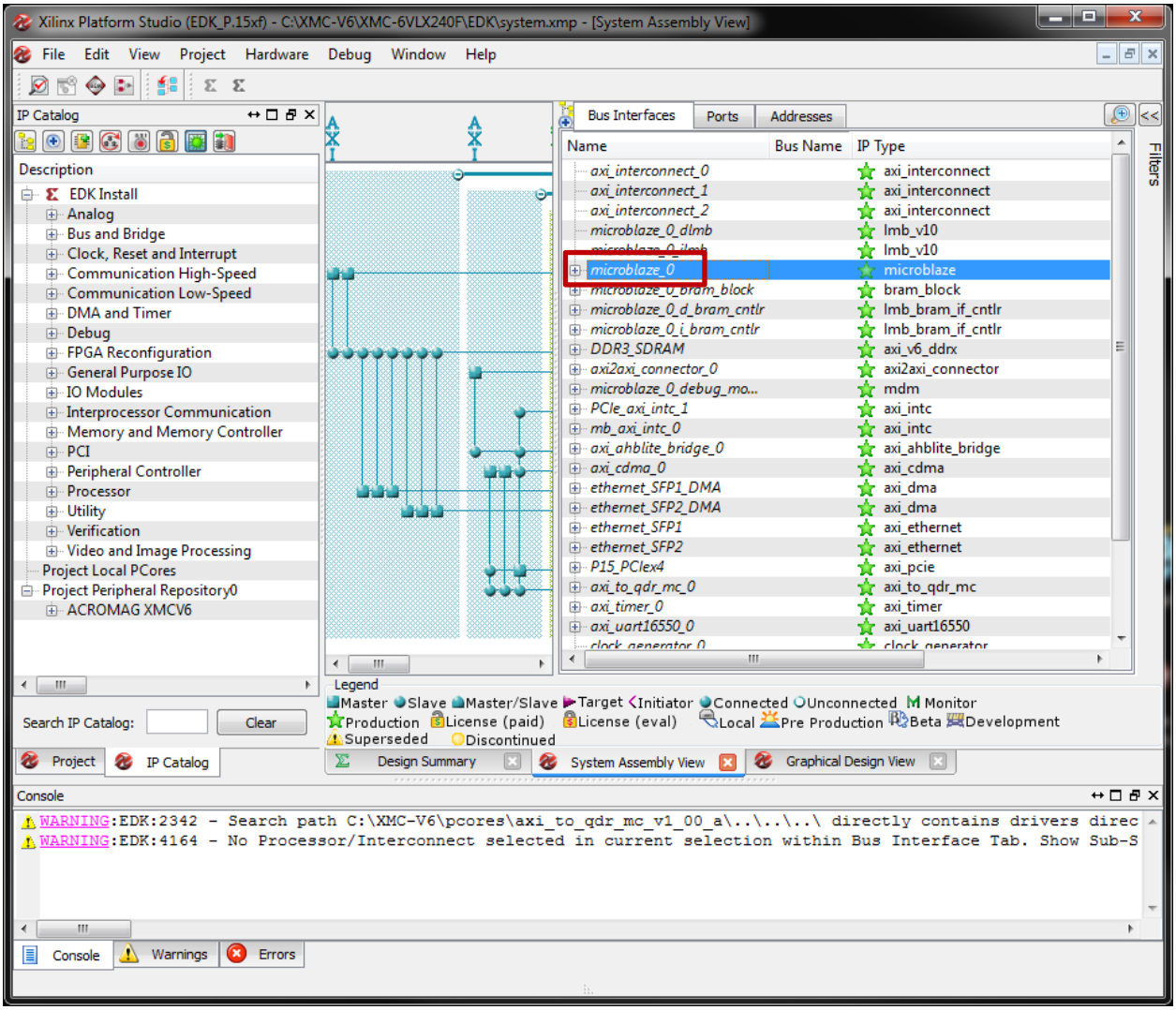
The DDR3 SDRAM is currently located at address 0x40000000.



We will move the base address of the DDR3 SDRAM to address 0x80000000. Click in the Base Address column of the DDR3_SDRAM row and change the "4" to "8" and then click somewhere outside of that cell. EDK will calculate the new High Address.



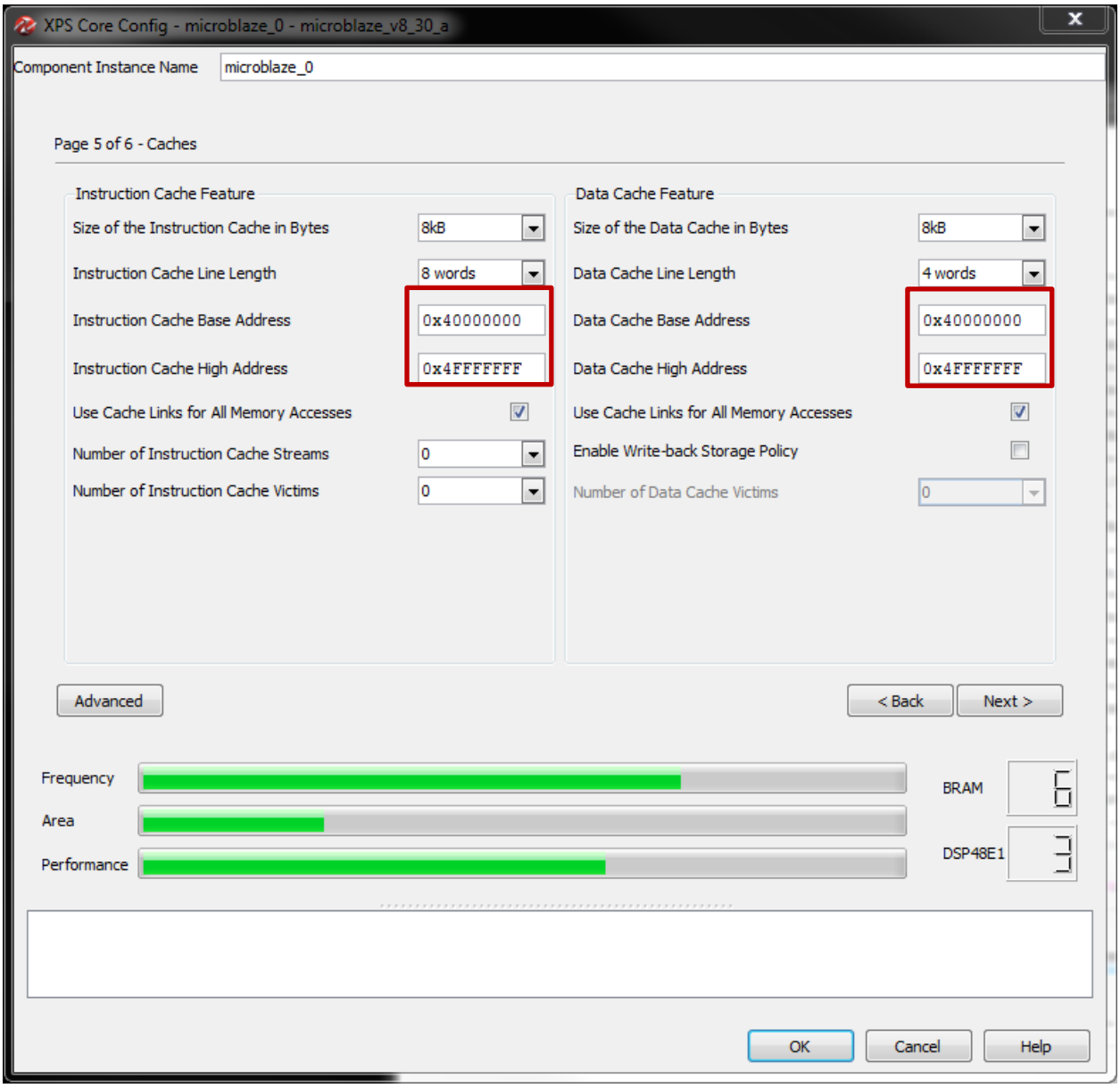
The cacheable range address parameter of the microblaze configuration must also be updated. Select the "Bus Interfaces" tab and then right-click on microblaze_0 and select "Configure IP" from the pop-up menu.



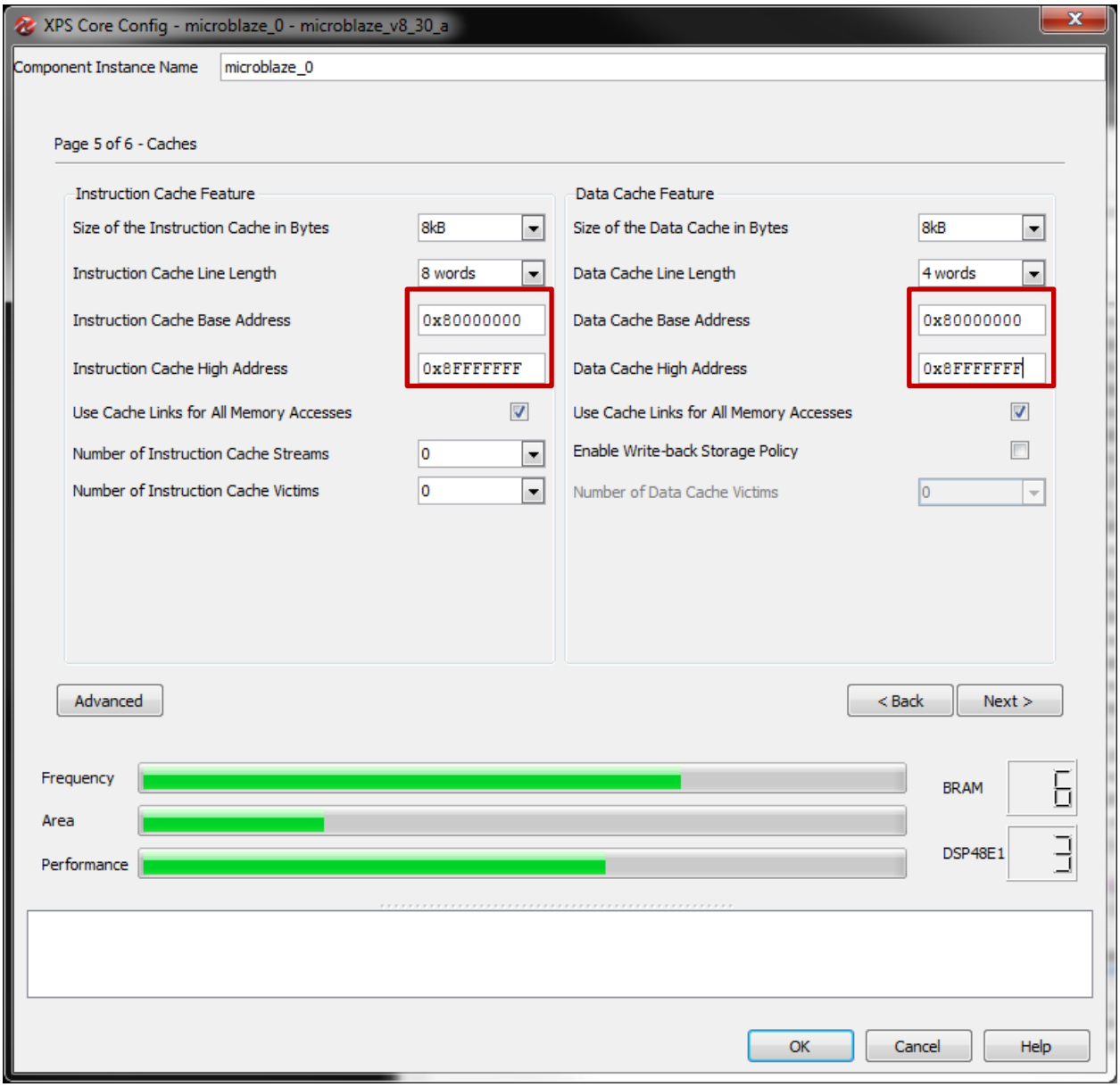
Click "Next" 4 times to arrive at the "Caches" page. Update the instruction and data cache base and high addresses to align with the new address for DDR3 SDRAM



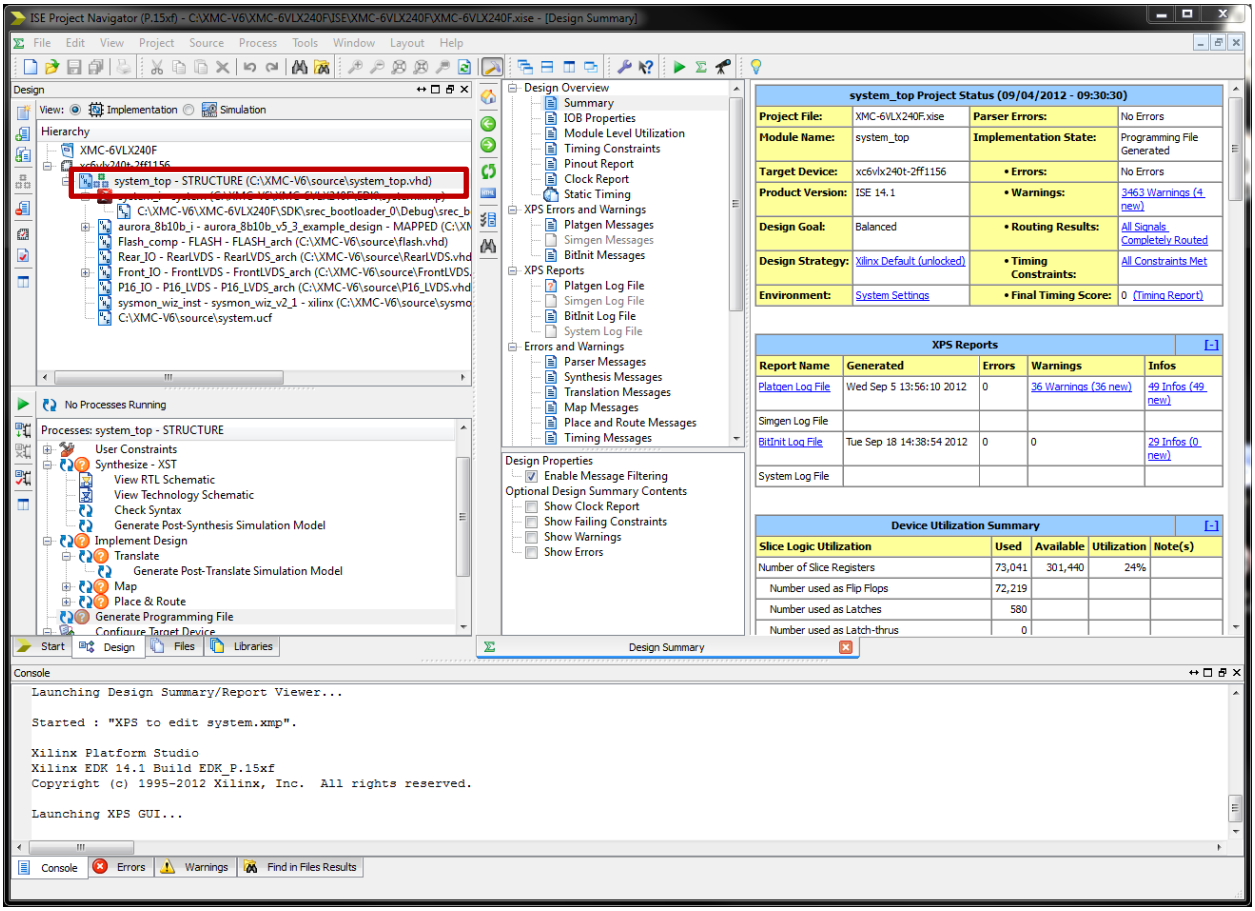
The cacheable address range is currently set to 0x40000000 to 0x4FFFFFFF.



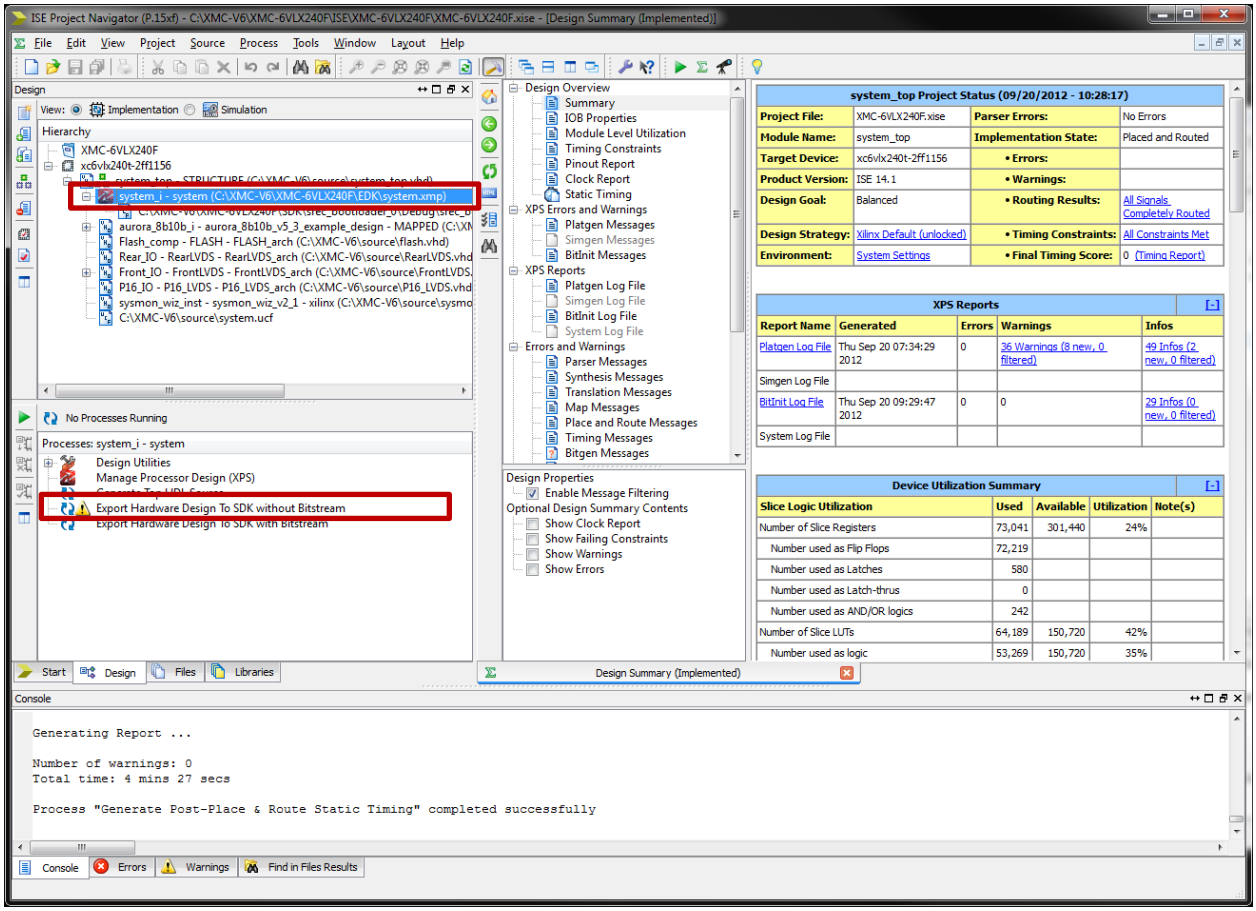
Set the instruction and data cache base addresses and high addresses to 0x80000000 to 0x8FFFFFFF.



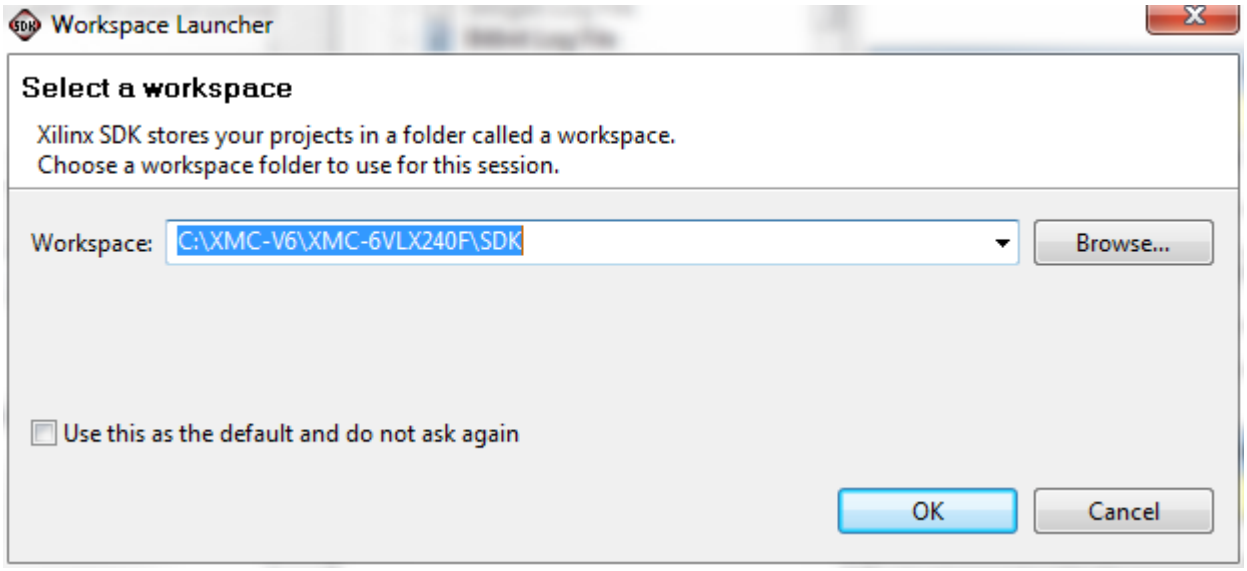
Click "OK to accept the changes. Next exit XPS and return to ISE to compile the updated system. Right-click on system_top in the Heirarchy pane and then select "implement top module" from the pop-up menu.



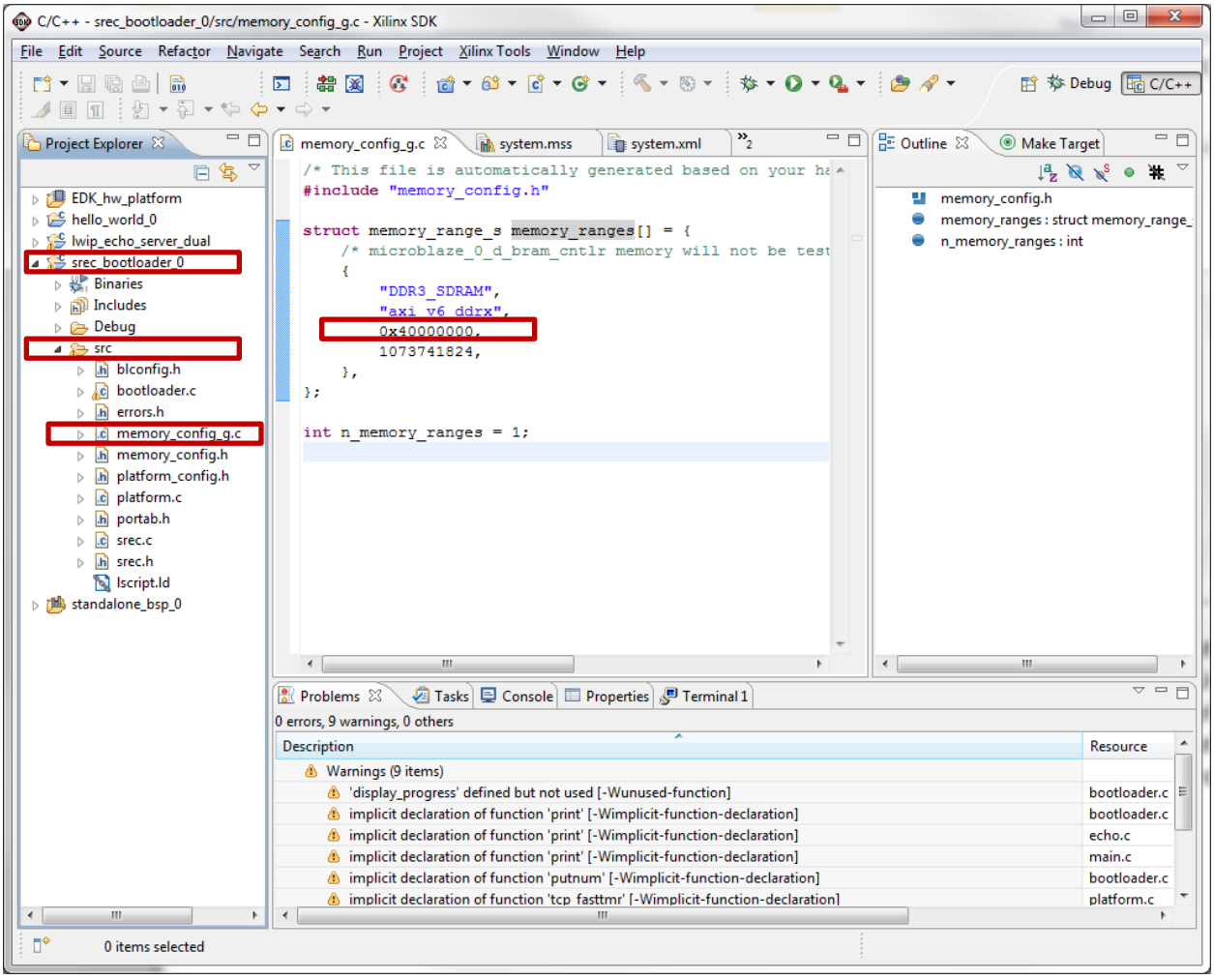
After the place and route process is completed, the hardware definition files used by SDK will need to be updated. Click on "system_i" in the Heirarchy pane and notice the processes available in the process pane. Double-click on "Export Hardware to SDK without Bitstream".



ISE will ask for confirmation of the SDK workspace path. Confirm that the path is C:\XMC-V6\XMC-6VLX240F\SDK. Click on OK, SDK will then launch.

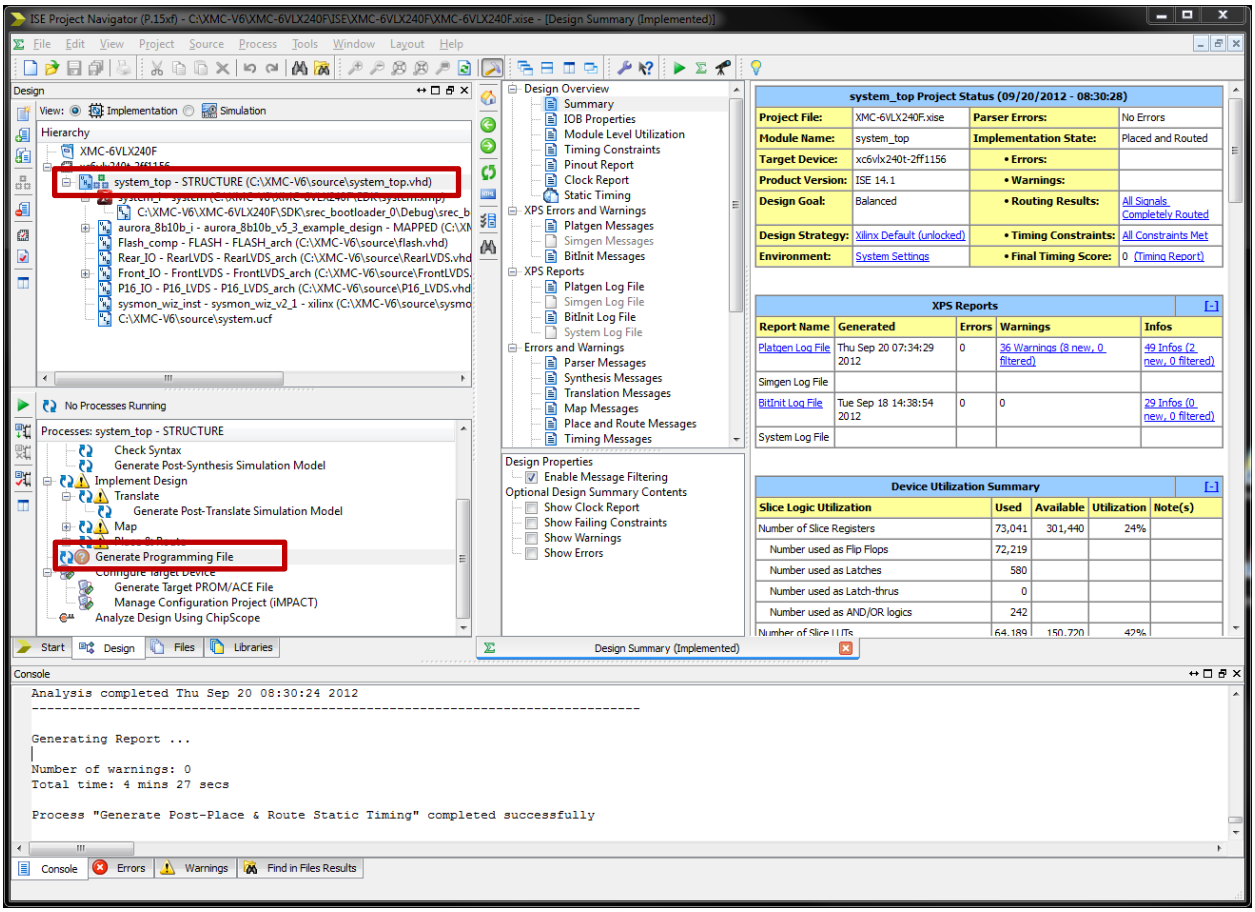


SDK will detect the change in the hardware description files and automatically rebuild all of the software projects. Unfortunately the change to the DDR3 SDRAM address is not reflected in all of the places it needs to be. The memory_config_g_c.c file used by the srec_bootloader program must be updated with the DDR3 SDRAM address. In the "Project Explorer" pane expand the srec_bootloader_0 proejct and then expand the "src" folder. Double-click on memory_config_g to open it in the text editor. Change the base address constant "0x40000000" to "0x80000000" and save the changes. The project will automatically rebuild.

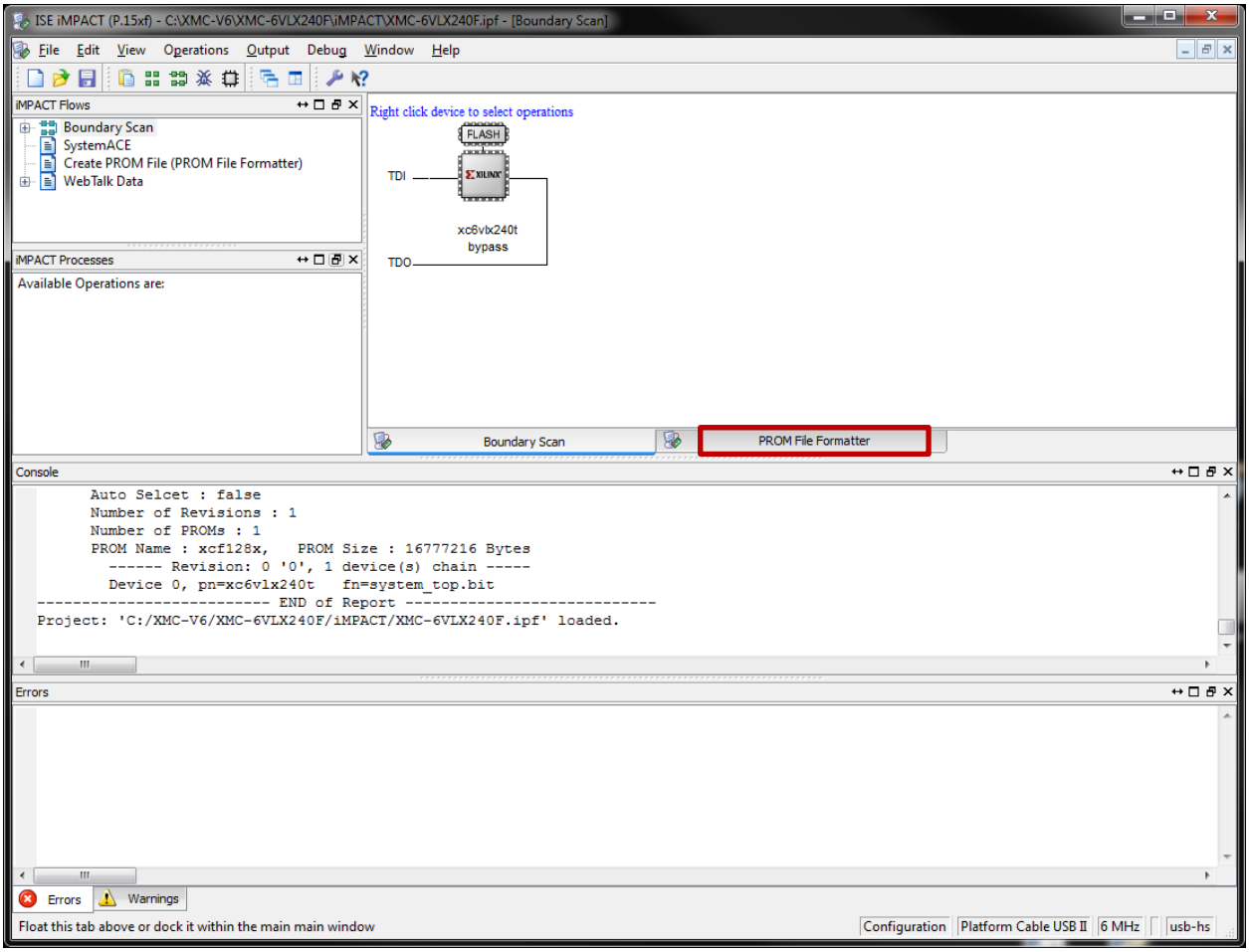


The srec_bootloader_0.elf program is included in the FPGA bitstream that is loaded into the FPGA on power-up. A new bitstream must be generated that includes the updates to the FPGA firmware as well as the updated software.

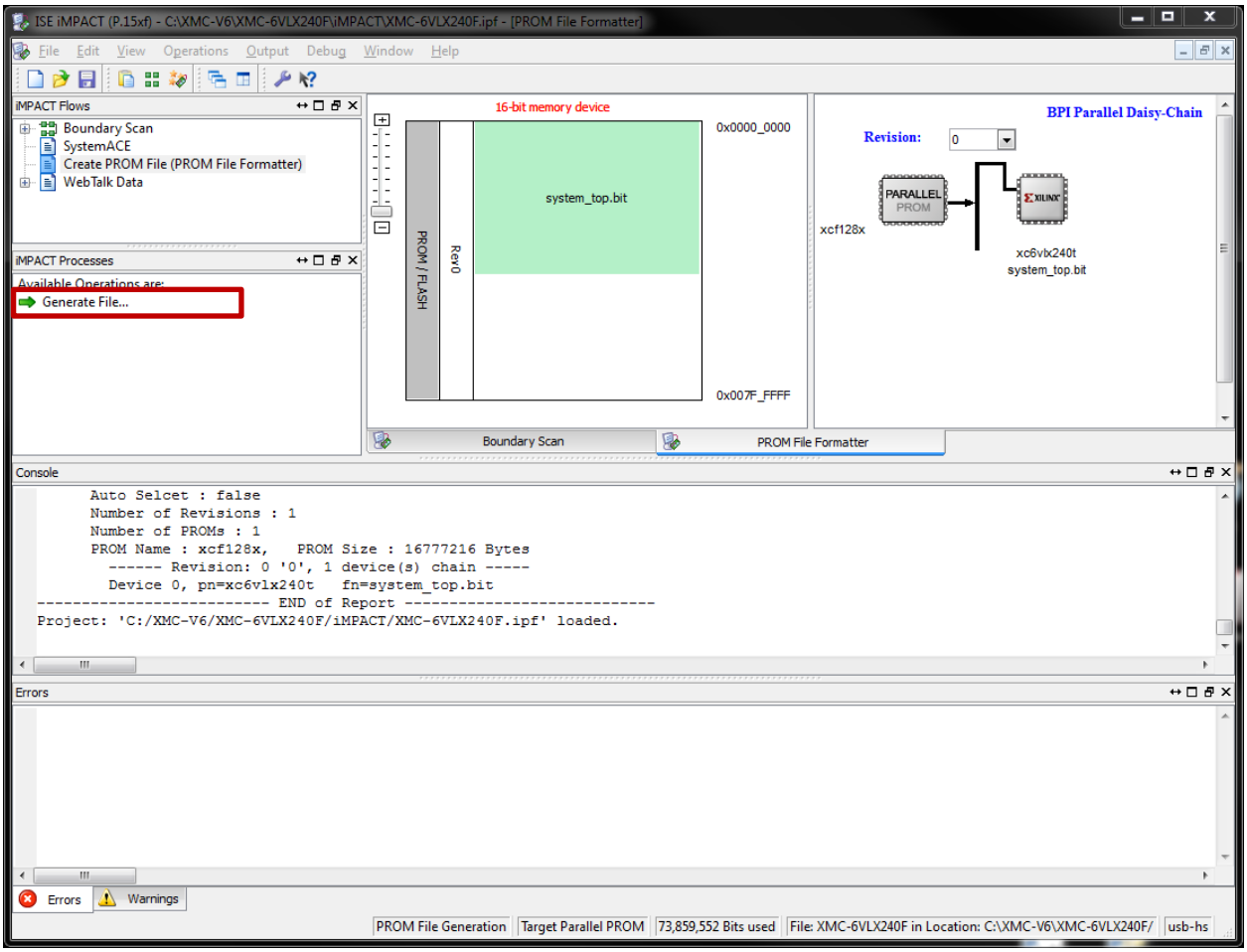
Click on "system_top" in the hierarchy pane and then double-click on "Generate Programming File" in the Processes pane.



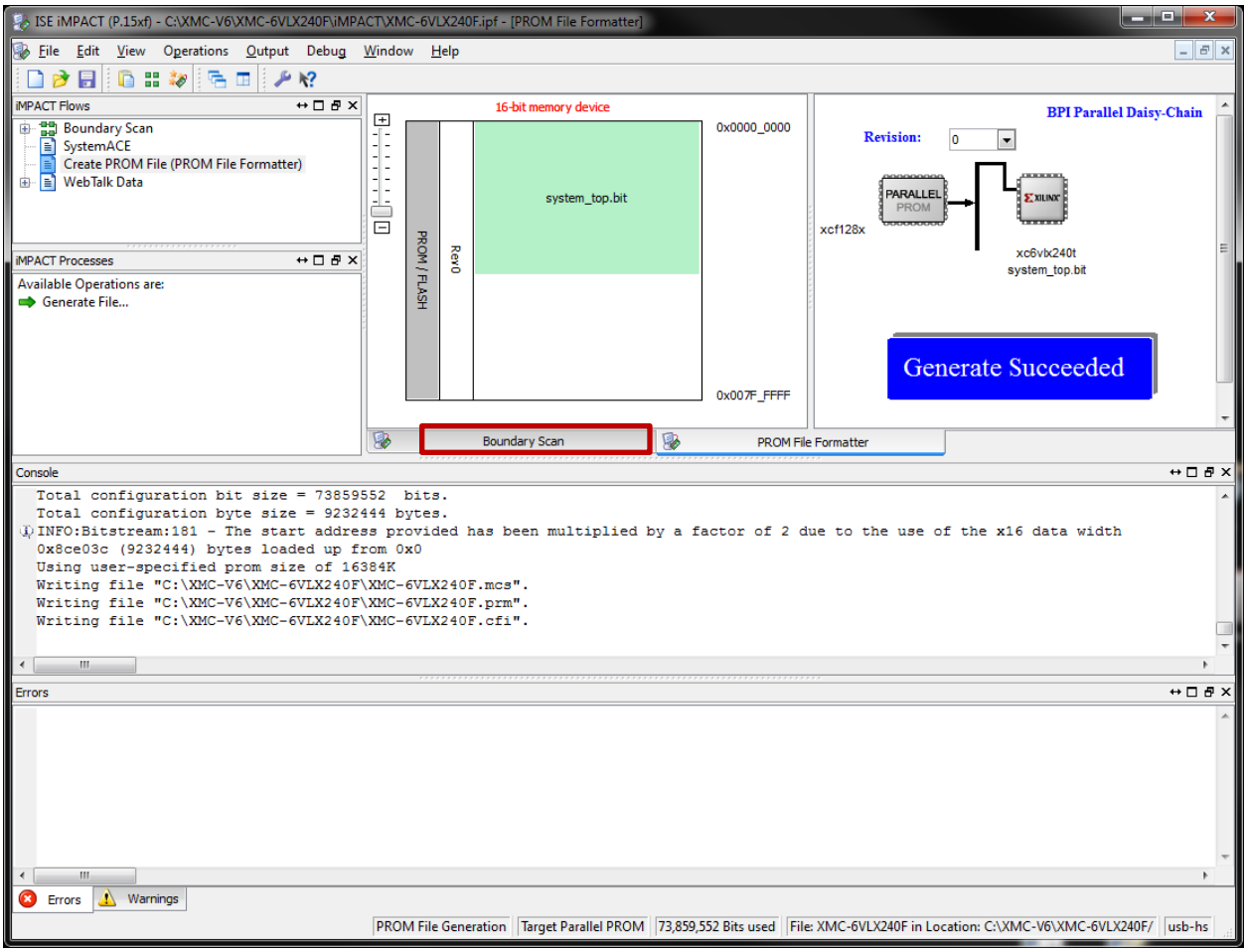
After the "Generate Programming File" process has completed, launch iMPACT to covert the .bit file to a .mcs file. If iMPACT doesn't automatically open the project file XMC-6VLX640F.ipf, then click on FILE->Open and navigate to C:\XMC-V6\XMC-6VLX240F\iMPACT\ XMC-6VLX640F.ipf and select it. Next, select the "PROM File Formatter" tab.



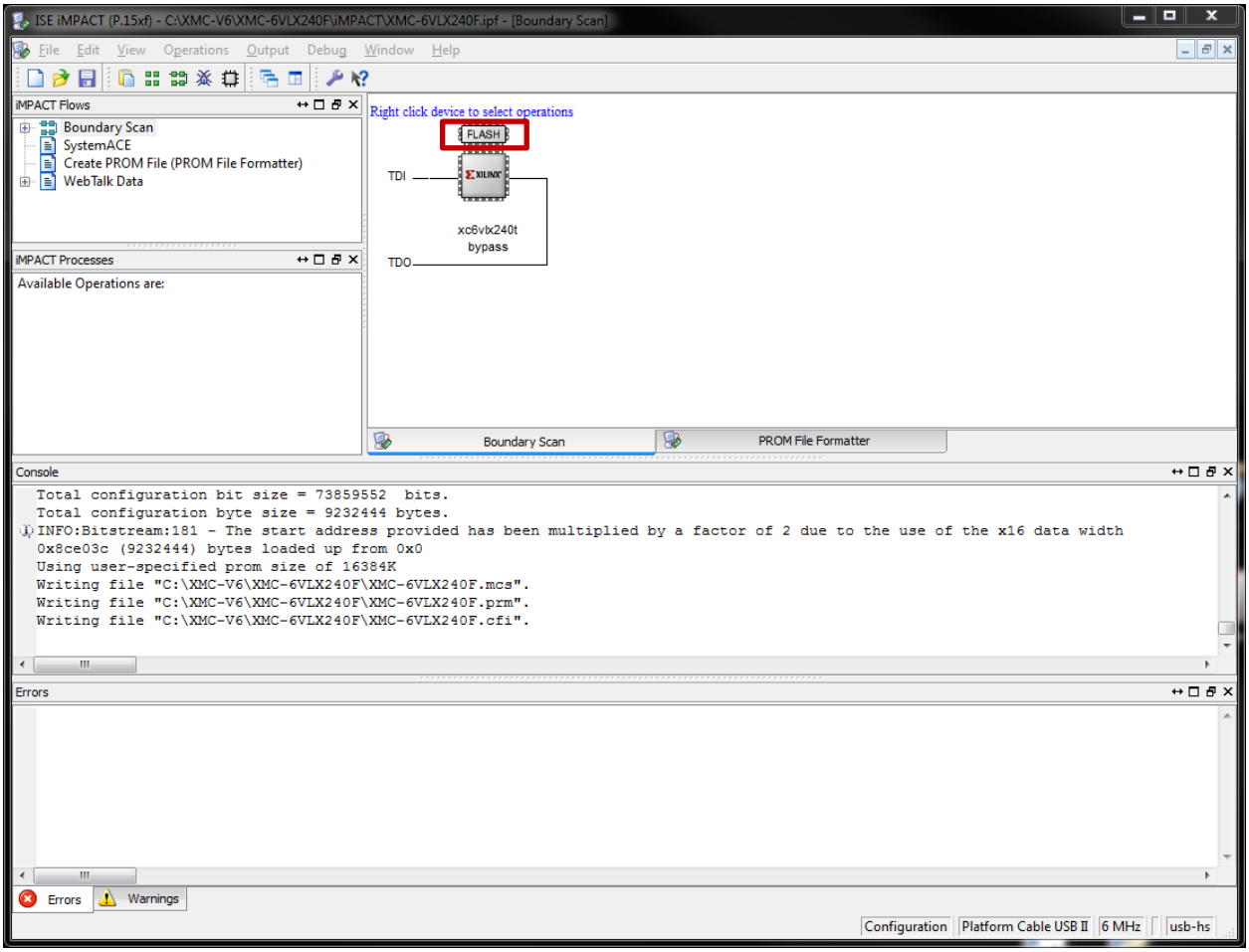
Double-click on "Generate File…" in the iMPACT Processes pane. This process will create the file XMC-6VLX240F.mcs



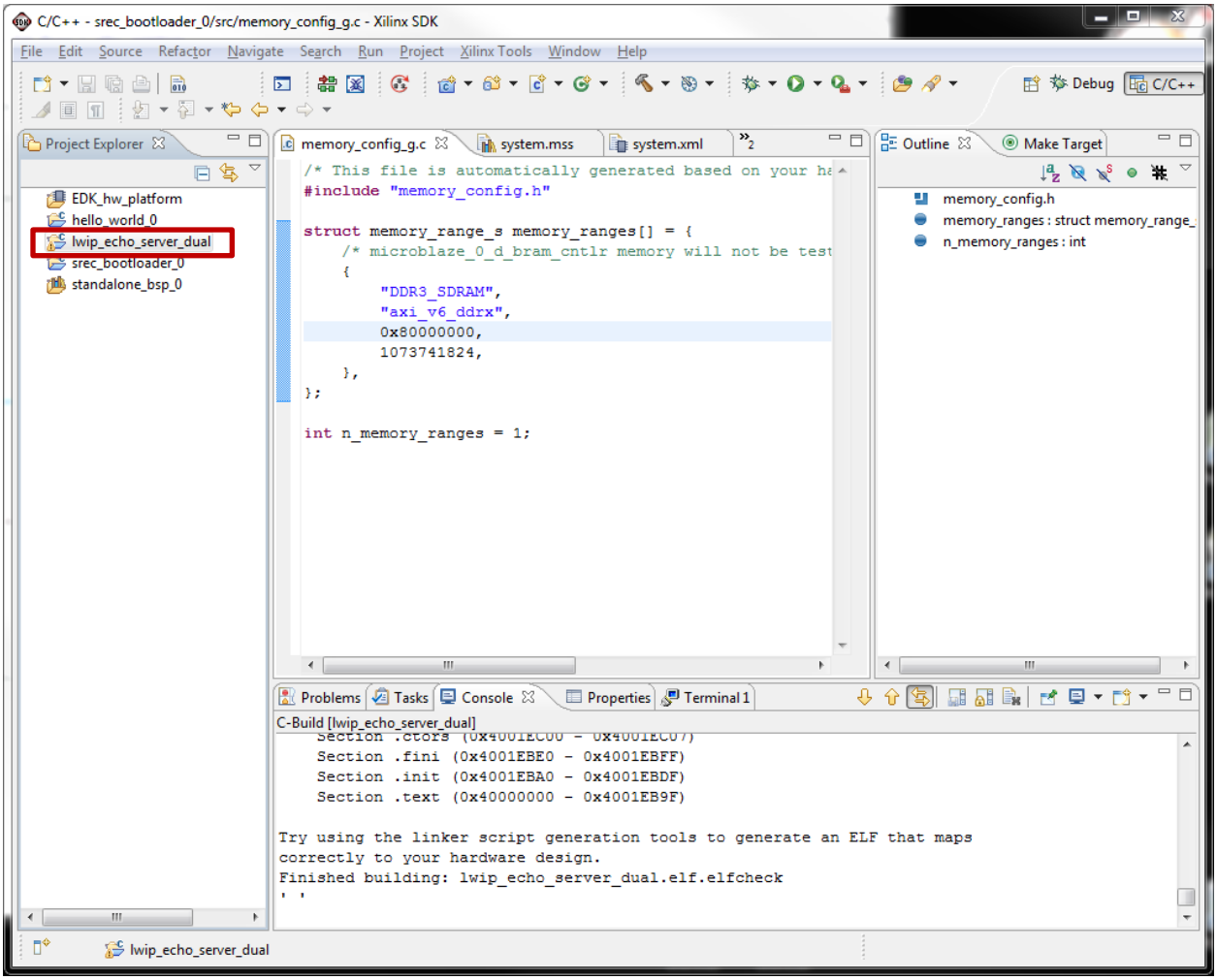
Click on the "Boundary Scan" tab



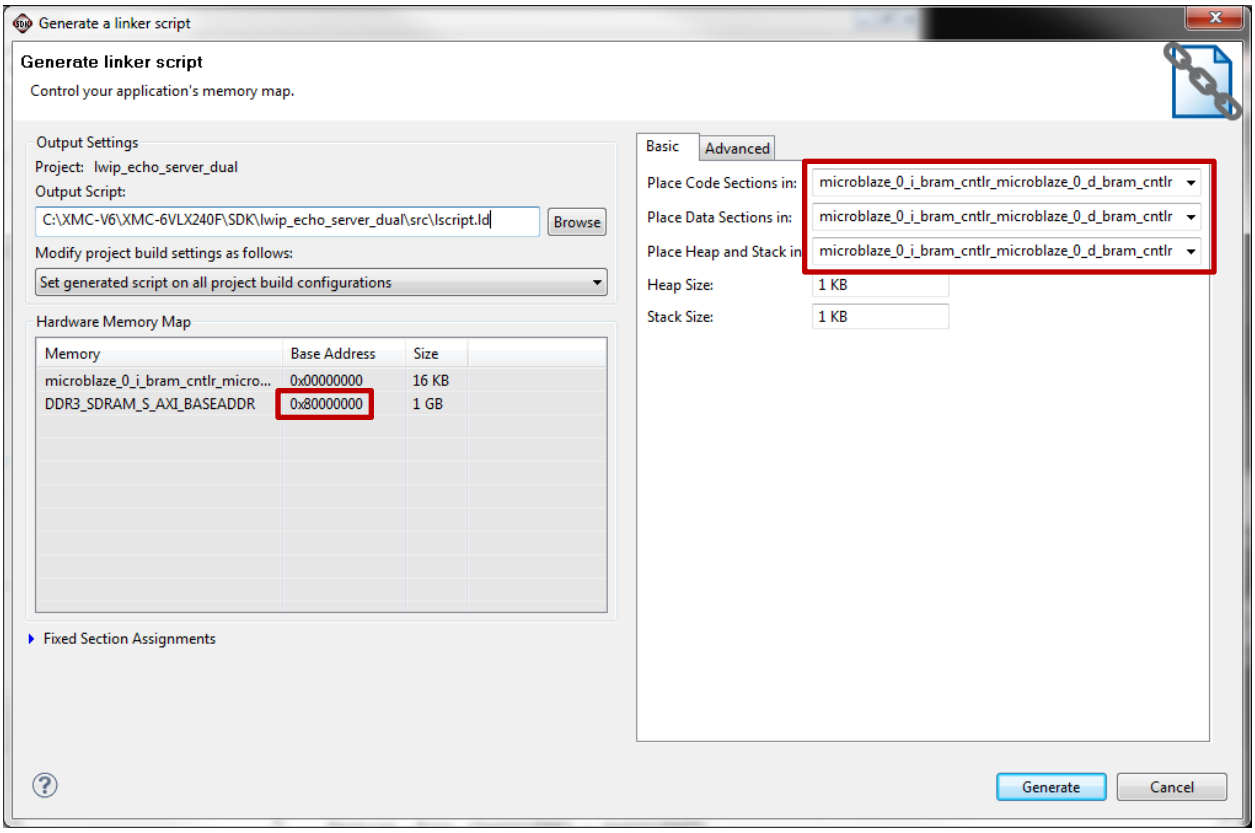
In this step the updated bitstream will be written to the platform FLASH on the XMC-V6. A Platform USB II cable or equivalent must be connected to the JTAG port. Right-Click on the "FLASH" device attached to the FPGA in the diagram. Select "Program" from the pop-up menu. iMPACT will likely report a failure message the first time. Select "Program" a second time.



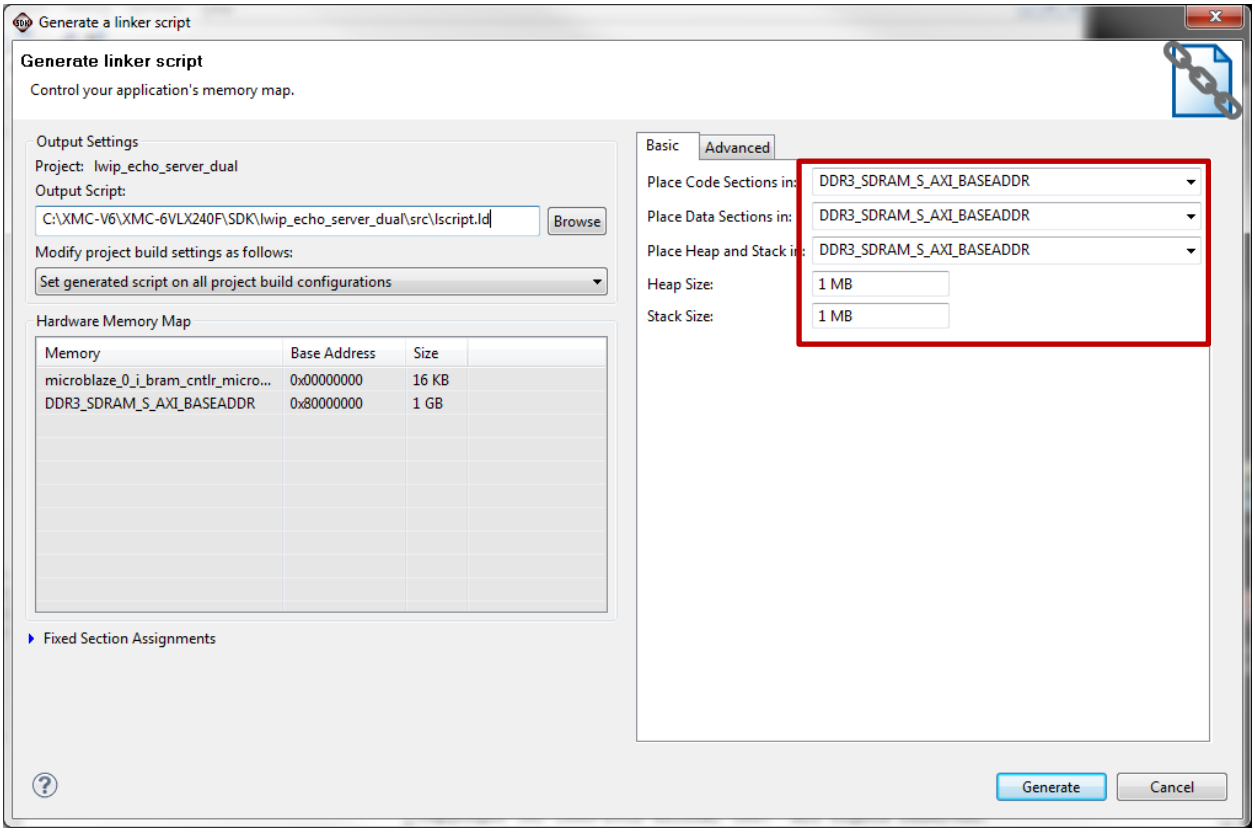
The XMC-6V is delivered with the lwip_echo_server_dual program loaded in the BPI FLASH memory. On power-up the srec_boot_loader program copies the lwip_echo_server_dual program from FLASH memory to DDR3 SDRAM and then executes the lwip_echo_server_dual program in DDR3 SDRAM. The linker script for the program lwip_echo_server_dual must be updated to use the new base address assigned to DDR3 SDRAM. Right-click on the lwip_echo_server_dual project in the "Project Explorer" pane and select "Generate Linker Script" from the pop-up menu.



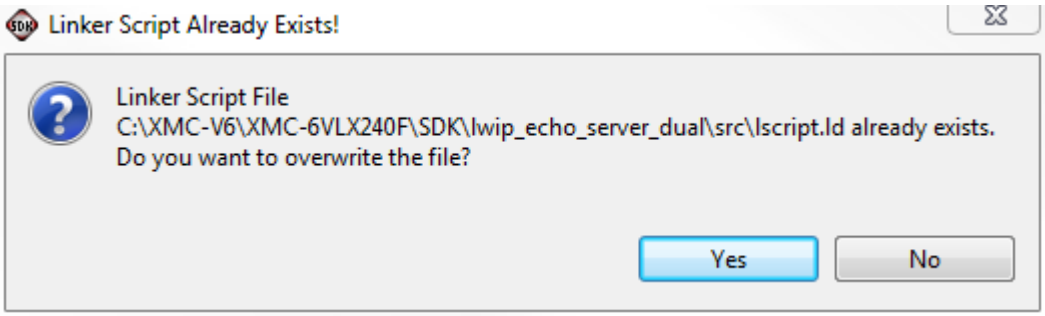
The base address of the DDR3 SDRAM has been updated automatically, but the code, data, and heap sections are currently located in block RAM.



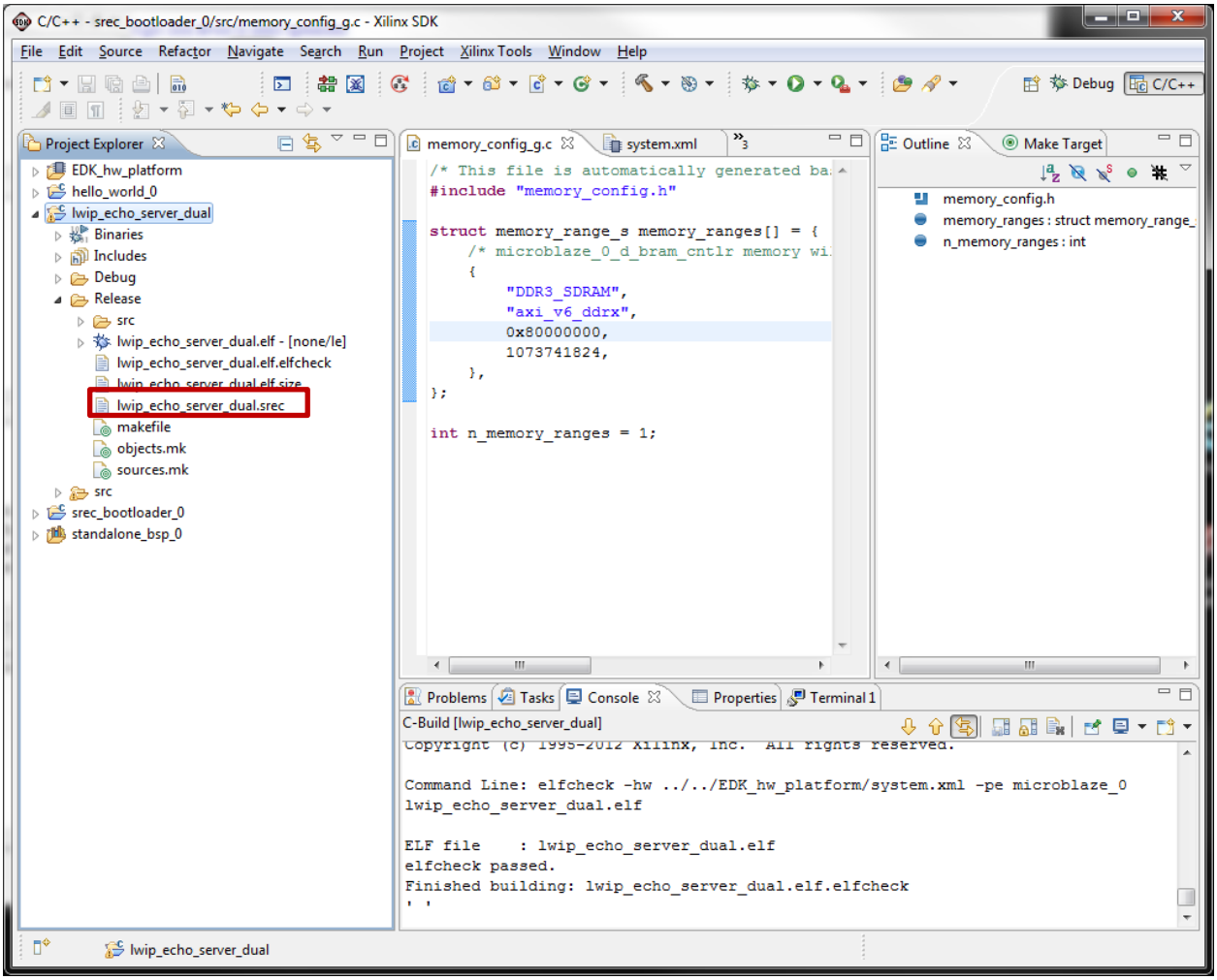
Select DDR3_SDRAM_S_AXI_BASEADDR from the drop-down list for each of code section, data section, and heap and stack. Enter 1048576 for the heap size and the stack size. 1 MB will be displayed in the box. Click on "Generate".



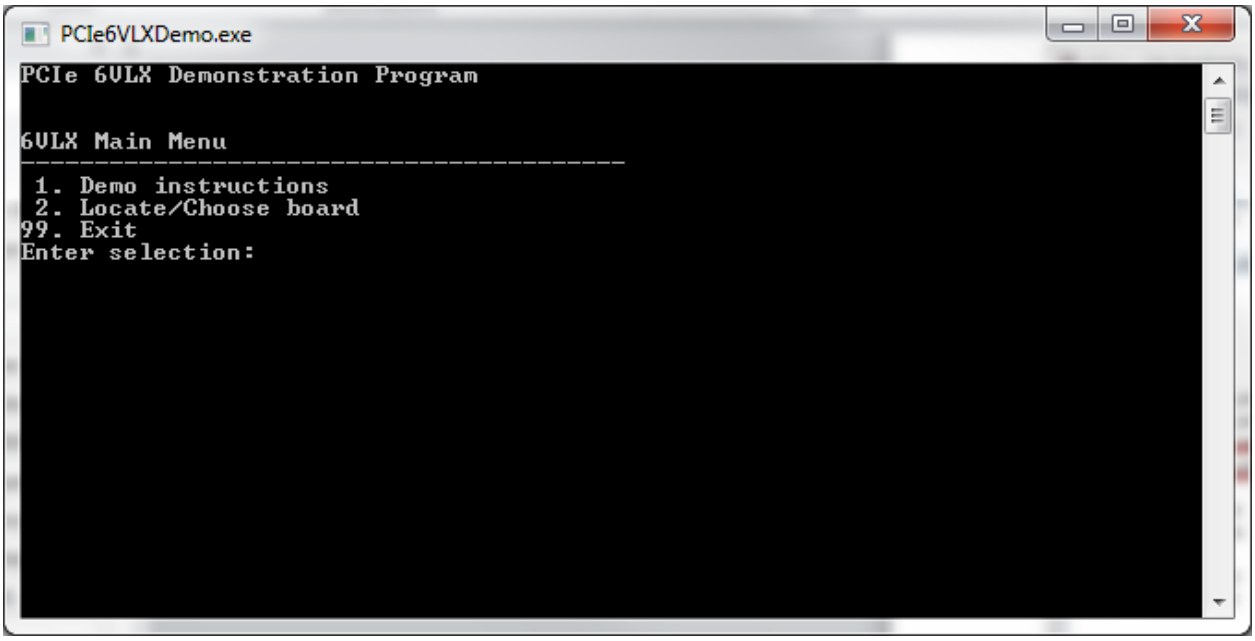
A message box will appear. Click Yes to overwrite the existing file.



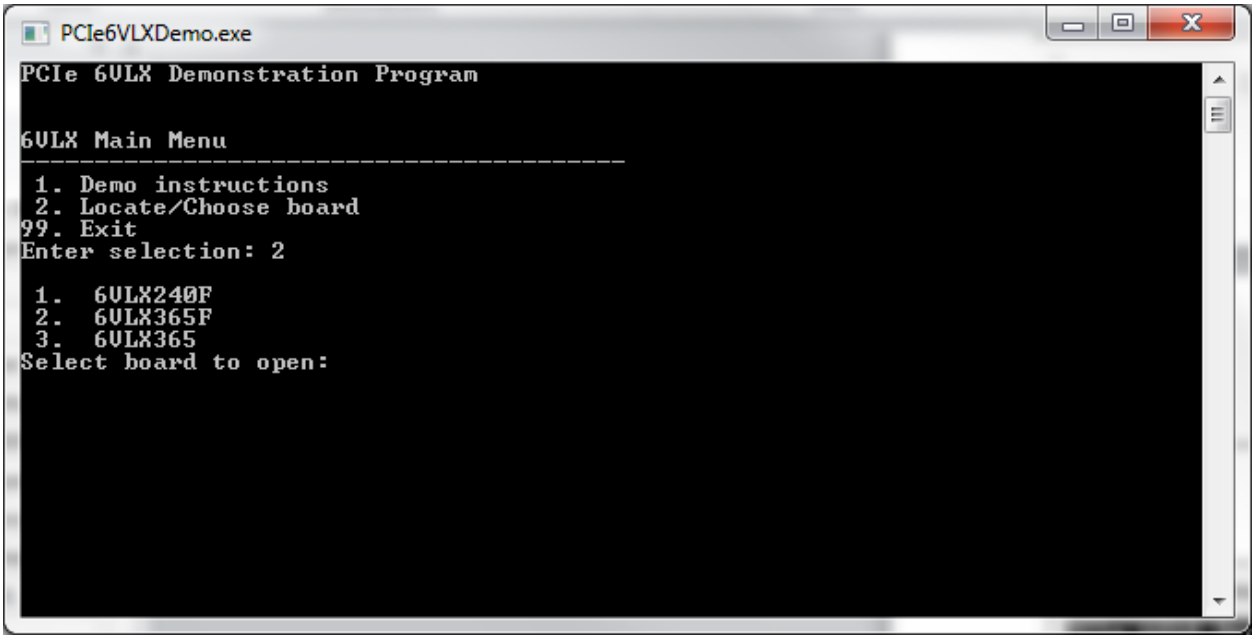
The project will automatically re-build and create a new lwip_echo_server_dual.srec file. This is the file that will be written to the BPI FLASH.



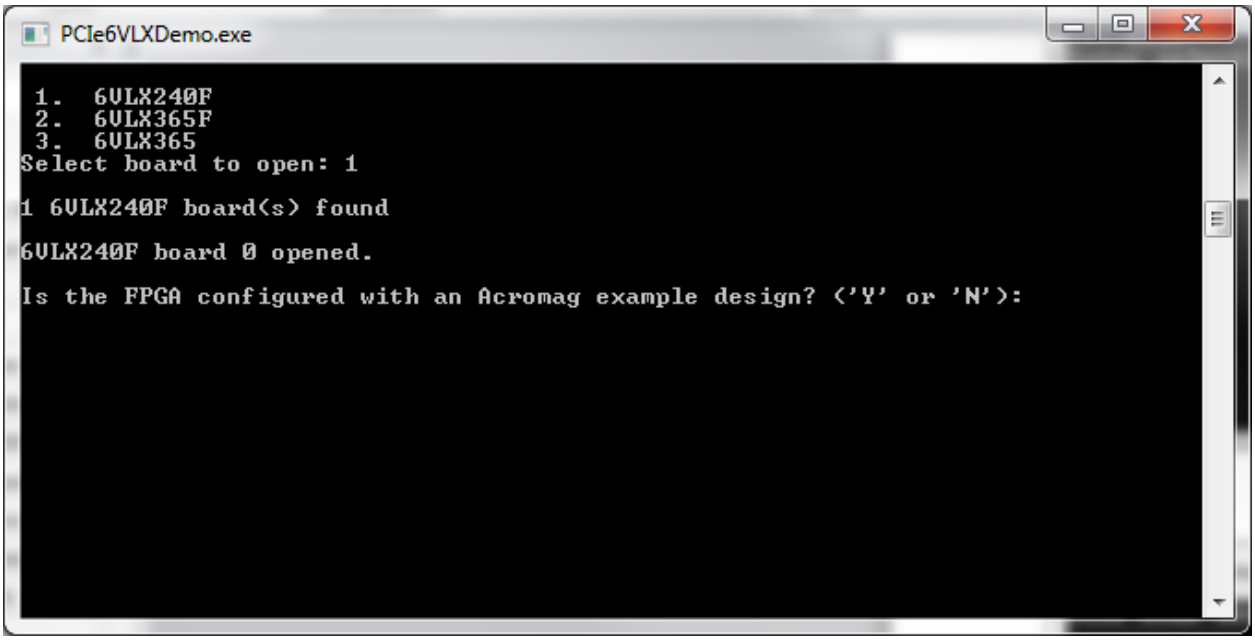
Run the Acromag PCIe6VLX demo program. Enter "2" to Locate/Choose board.



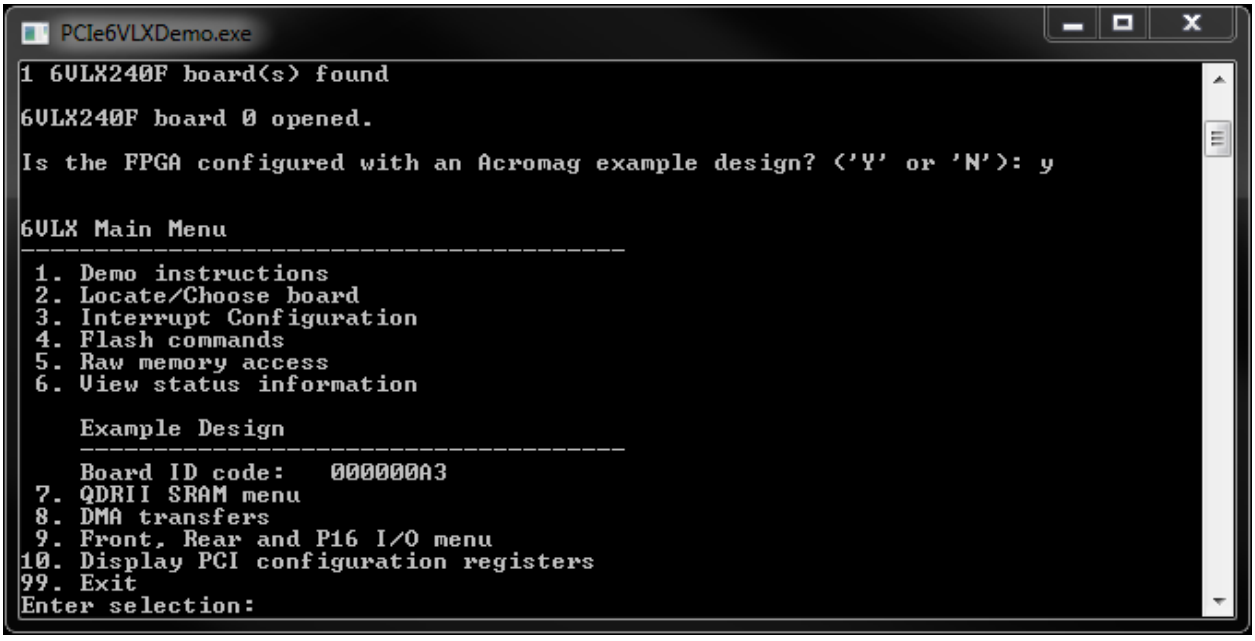
Enter the appropriate number to select the XMC-V6 variant that you have installed.



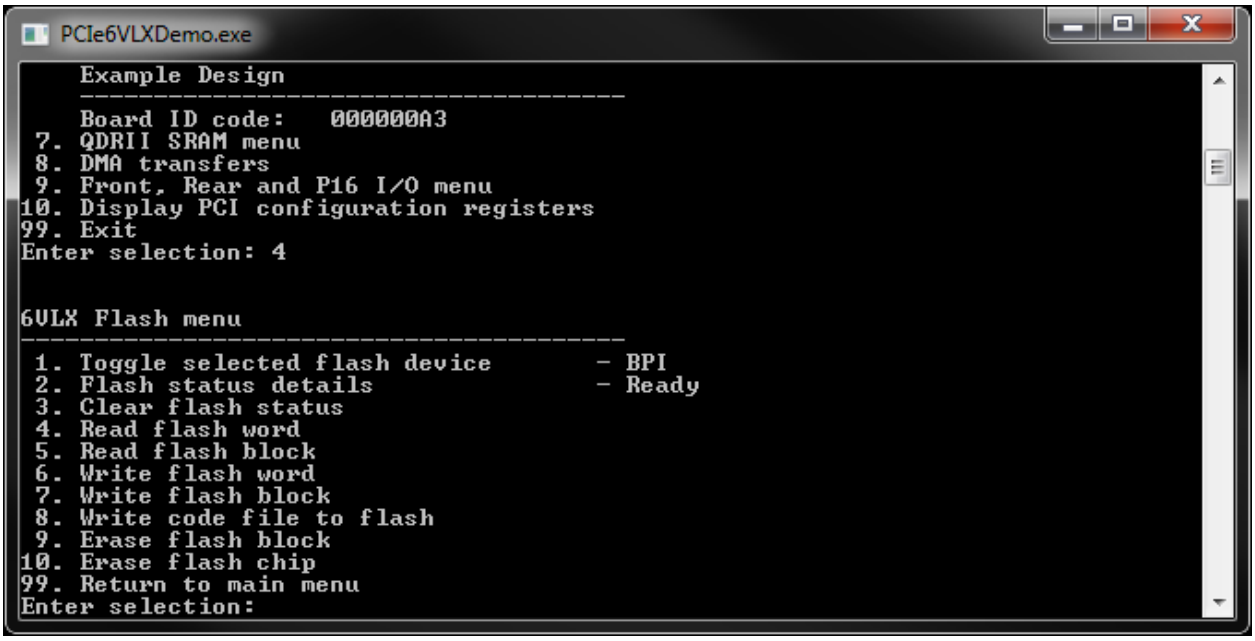
Enter "Y" to indicate the FPGA is configured with the Acromag example design.



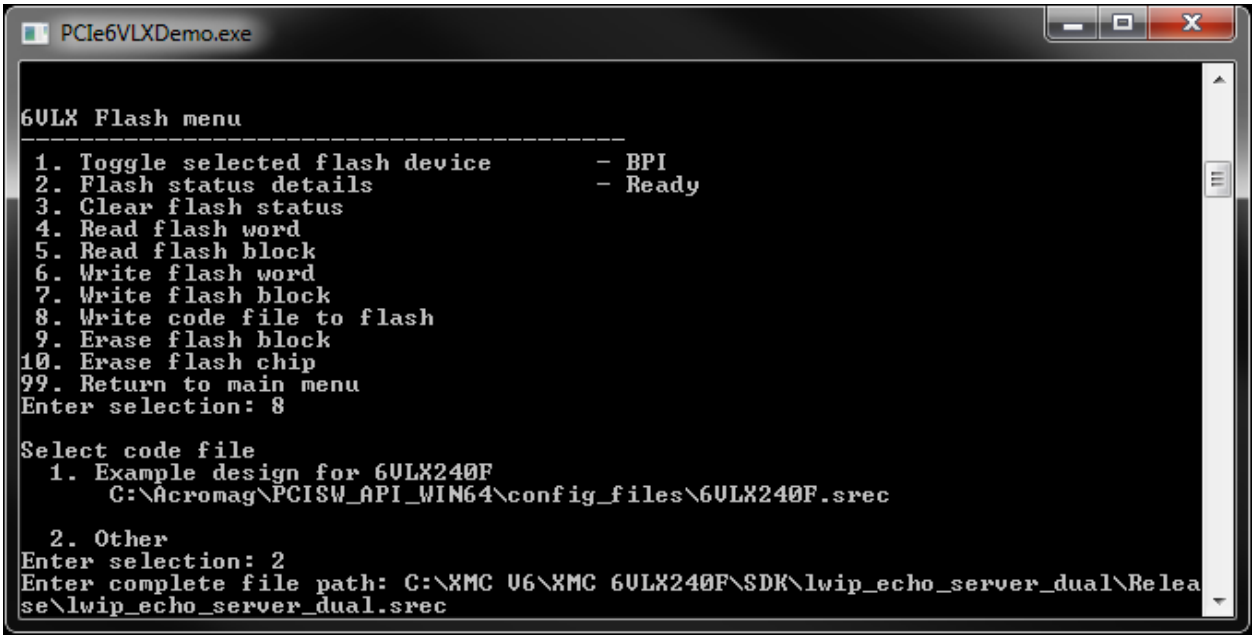
Select function "4" Flash commands.



If BPI is not the currently selected FLASH, then select function "1" Toggle selected flash device. Select "8" Write code file to flash.



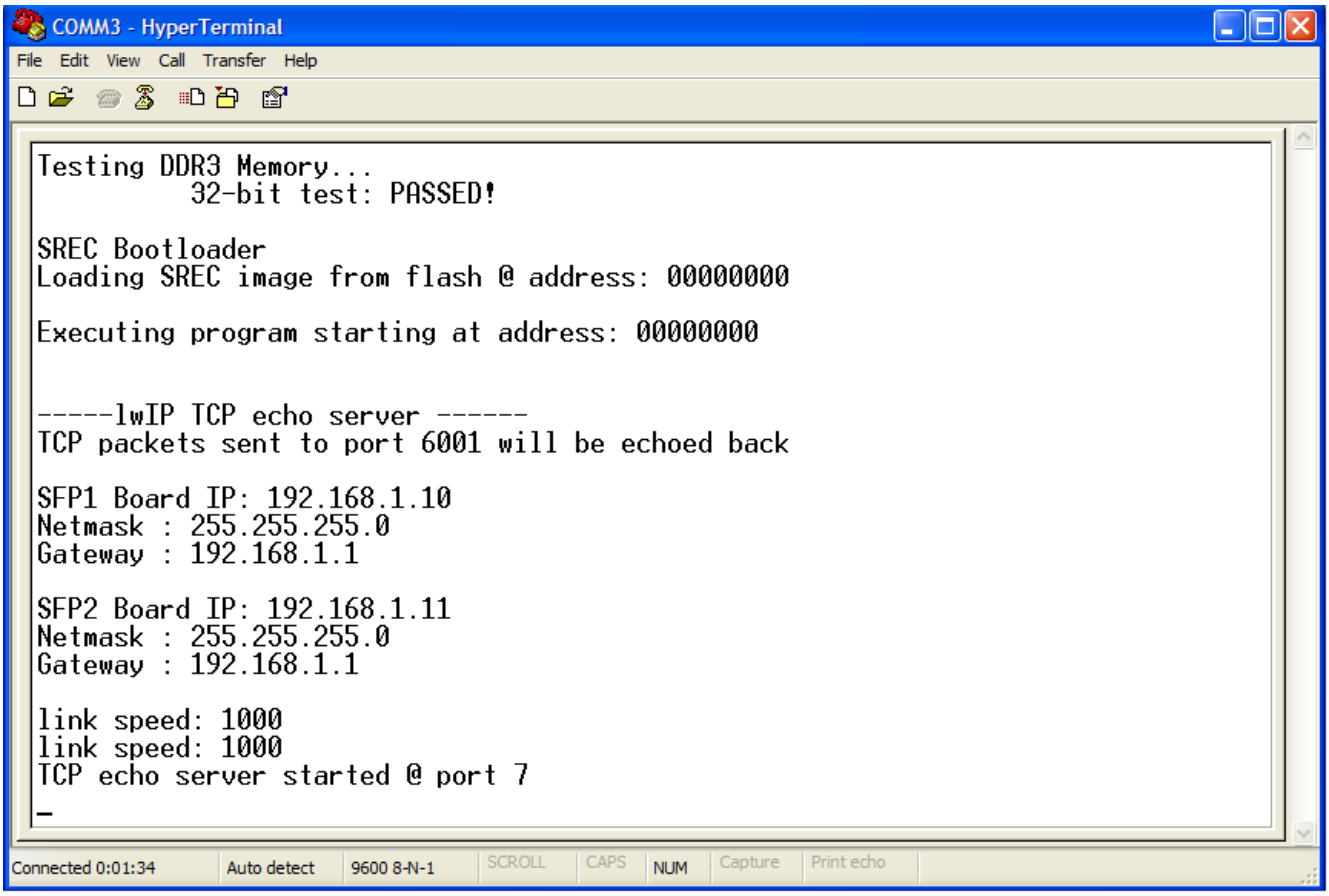
Select "2" Other, and enter the path to the lwip_echo_server_dual.srec file C:\XMC-V6\XMC-6VLX240F\SDK\lwip_echo_server_dual\Release\lwip_echo_server_dual.srec



After the programming operation is complete enter "99" twice and answer "Y" to exit the program.



To summarize, the bitstream in platform FLASH has been updated with the new hardware containing the relocated DDR3 SDRAM and the updated srec_bootloader program. The BPI FLASH contains the updated lwip_echo_server_dual program. The srec_booltloader will load the lwip_echo_server_dual into DDR3 SDRAM and execute it on power-up. Both programs will report their progress by writing messages to the serial port. To view the progress messages displayed on power-up, a separate PC must be running a terminal emulator program such as hyper-terminal connected to COMM3. The serial port parameters in the terminal emulator should be configured for 9600 baud, 8 data bits, 1 stop bit, and no parity. A USB cable must be connected from the XMC-V6 USB port to the USB port on the separate PC. The following progress message will be displayed in the terminal emulator when power is applied to the XMC-V6.



Since the lwip_echo_server_dual program is successfully executing out of DDR3 SDRAM we have validated our change to the DDR3 SDRAM base address.

6.0 SERVICE AND REPAIR

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 easily damaged unless special SMT repair and service tools are used. Further, Acromag has automated test equipment that thoroughly checks the performance of each board. When a board is first produced and when any repair is made, it is tested before shipment.

Service and Repair Assistance

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

Preliminary Service Procedure

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

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

Heat Sink Removal

When the XMC-6VLX board is mounted on a conduction cooled carrier it may be necessary to remove the Acromag provided heat sink. The heat sink can be removed by removal of the 7 screws retaining the heat sink. These 7 screws are visible on the back of the board. After the 7 screws are removed, lift the heat sink off and wipe clean the silver colored thermal interface material from the Virtex 6 device.

Where to Get Help

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

Acromag's application engineers can also be contacted directly for technical assistance via email, telephone, or FAX through the contact information listed at the bottom of this page. When needed, complete repair services are also available.

7.0 SPECIFICATIONS

PHYSICAL

Height	13.5 mm (0.531 in)
Stacking Height	10.0 mm (0.394 in)
Depth	149.0 mm (5.866 in)
Width	74.0 mm (2.913 in)
Board Thickness	2.21 mm (0.08 in)
Unit Weight:	4.288255oz (0.12157Kg)

POWER REQUIREMENTS

Power will vary dependent on the application. Power values are given of Acromag Example Design.

3.3 VDC (±5%)	Typical 50 mA	Max. 100 mA
+12/5 VDC (as +12V) (±5%)	Typical 1.8A	Max. 2.0A
-12 VDC (±5%)	0mA	

On Board 1.0V Power to Virtex 6 FPGA	**Current Rating** (Maximum available for the user-programmable FPGA)
1.0V (±5%)	6A Maximum

ENVIRONMENTAL

Operating Temperature

Model	*Operating Temperature*
XMC-6VLX240 **XMC-6VLX365**	***0°C to 70°C***

Relative Humidity: 5-95% Non-Condensing.

Storage Temperature: -55°C to 100°C.

Non-Isolated: PCIe bus and field commons have a direct electrical connection.

Designed to comply with EMC Directive 2004/108/EC Class B

Radiated Field Immunity (RFI): Complies with IEC 61000-4-3 with no register upsets.

Conducted R F Immunity (CRFI): Complies with IEC 61000-4-6 with no register upsets.

Surge Immunity: Not required for signal I/O per IEC 61000-4-5.

Electric Fast Transient (EFT) Immunity: Complies with IEC 61000-4-4 Level 2 (0.5KV at field I/O terminals).

Electrostatic Discharge (ESD) Immunity: Complies with EN61000-4-2 Level 3 (8KV enclosure port air discharge) Level 2 (4KV enclosure port contact discharge).

Radiated Emissions: Meets or exceeds European Norm 61000-6-3:2007 for class B equipment. Shielded cable with I/O connections in shielded enclosure is required to meet compliance.

User Programmable (U2) FPGA

XC6VLX240T-1FF1156

- 241,152 Logic Cells
- 3,650 Kbit Distributed RAM
- 416 36 Kbit Block RAMs
- 768 DSP48E1 Slices
- 12 Mixed Mode Clock Managers
- 2 Interface Blocks for PCI Express
- 4 Ethernet MACs

XC6VLX365T-1FF1156

- 364,032 Logic Cells
- 4,130 Kbit Distributed RAM
- 416 36 Kbit Block RAMs
- 576 DSP48E1 Slices
- 12 Mixed Mode Clock Managers
- 2 Interface Blocks for PCI Express

4 Ethernet MACs

EDK Example Design

Xilinx XC6VLX240T-1FF1156 Resource Usage

- Slice Registers 64,465 Used 301,440 Available 21% Utilization
- Slice LUTs 68,276 Used 150,720 Available 45% Utilization
- MMCMs 4 Used 12 Available 33% Utilization

Xilinx XC6VLX365T-1FF1156 Resource Usage

- Slice Registers 64,465 Used 455,040 Available 14% Utilization
- Slice LUTs 68,257 Used 227,520 Available 30% Utilization
- MMCMs 4 Used 12 Available 33% Utilization

P15 Connector

114 pin Samtec ASP-103614-05 connector complies with ANSI/VITA 42.3-2006

P15 is the primary XMC connector
8 Gigabit differential pairs are provided
(Operating data rate between 480 Mb/s and 6.6 Gbit/s)
System Management (XMC provides hardware definition information read by the external controller using IPMI commands and I2C serial bus transactions.)
3.3V power (4 pins at 1A/pin)
3.3V auxiliary power (1 pin at 1A/pin)
Variable power (5V or 12V) (8 pins at 1A/pin)

P16 Connector

114 pin Samtec ASP-103614-05 connector complies with ANSI/VITA 42.3-2006

P16 secondary XMC connector
8 Gigabit differential pairs are provided
(Operating data rate between 480 Mb/s and 6.6 Gbit/s)
38 standard user I/O are also available (Example design uses as 19 differential LVDS pairs)

P4 REAR I/O

P4 Rear I/O interface: 64-pin female receptacle header (AMP 120527-1 or equivalent). This connector provides 64 rear I/O connections.
The rear I/O P4 PMC connector connects directly to banks 25 and 35 of the FPGA. Bank 25 and 35 Vcco pins are powered by 2.5 volts and thus will support the 2.5 volt IOStandards. Refer to the Virtex-6 SelectIO User Guide (available from Xilinx) for more information on the IOStandards available. The example design defines the rear I/O with 2.5 volt LVDS.

- Maximum Recommended Clock Rate…….150MHz (6.7ns clock period)
- Vcco Supply Voltage ……..2.5 volt
- VOH Output High Voltage……..1.602 volt
- VOL Output Low Voltage……0.898 volt
- VODIFF Differential Output Voltage ……350m volt typical
- VOCM Output Common Mode Voltage…..1.25 volt typical
- VIDIFF Differential Input Voltage……..100m volt minimum
- VICM Input Common Mode Voltage……0.3 volt min, 1.2 volt typical, 2.35 volt max

MEZZANINE FRONT I/O

This XMC module uses the 150 pin Samtec connector part number QSS-075-01-L-D-A which mates with the mezzanine module connector part number QTS-075-02-L-D-A-K and the Acromag Virtex 6 EDK module for external USB, JTAG, and 13 differential I/O or 26 single ended user signals.

Board Oscillators

Board Crystal Oscillators: 125MHz (U21, U31)
Frequency Stability: ± 0.00315% or 31.5ppm

Board Crystal Oscillator: 200MHz (U3)
Frequency Stability: ± 0.00315% or 31.5ppm

Board Crystal Oscillator: 48MHz (U3)
Frequency Stability: ± 0.0050% or 50ppm

DDR3 Memory

128 Meg x 16-bit Micron Device MT41J128M16HA-15EIT uses a double data rate architecture.

Four MT41J128M16HA-15EIT memory devices (U8, U9, U10 and U11) are used to form a 64-bit data bus.

128 Meg x 16-bit =2Gb each device

8Gb = 1GB total all four devices together

DDR3 memory devices are wired to FPGA banks 15, 16, 26, and 36.

DCI VRP/N resistor connections are implemented on banks 15 and 36.

DCI functionality in bank 15 is achieved in the UCF by cascading DCI between adjacent banks as follows:

```
CONFIG DCI_CASCADE = "15 16";
```

The memory interface logic require a set of FPGA "No Connect" pins. These are found in the UCF as CONFIG PROHIBIT pins as follows:

```
CONFIG PROHIBIT = A16,D34,F33,K16,K26,L15,N28,N33;
CONFIG PROHIBIT = F31,K14;
```

QDR II+ SRAM Memory

Two CY7C1565KV18-400BZI memory devices (U12 and U13) are used to form a 72-bit data bus.

2 Meg x 36-bit Cypress CY7C1565KV18-400BZI memory QDR II+ are synchronous pipelined Burst SRAMs equipped with separate read and write ports.

2 Meg x 36-bit =72Mb each device

144Mb = 18MB total both devices together

QDRII+ memory devices are wired to FPGA banks 13, 22, 23, 32 and 33.

DCI VRP/N resistor connections are implemented on bank 22. DCI functionality in bank 23 is achieved in the UCF by cascading DCI between adjacent banks as follows:

CONFIG DCI_CASCADE = "22 23";

16MB Platform Flash

16 MByte (128Mb) Xilinx XCF128X-FTG64C Platform Flash XL device is used to configure the Virtex 6 FPGA.

32MB Linear BPI Flash

32 MByte Micron/Numonyx PC28F256P30TF Non-volatile storage that can be used for MicroBlaze software storage. The Linear BPI Flash shares the dual use configuration pins in parallel with the XCF128 Platform Flash XL. There are a total 256 addressable blocks each 64-Kwords.

PCIe Bus Interface

XMC Compatibility: Conforms to PCI Express Base Specification v2.0, and XMC Specification, P1386.1

ANSI/VITA 42.0: Complies with XMC module mechanicals and connectors

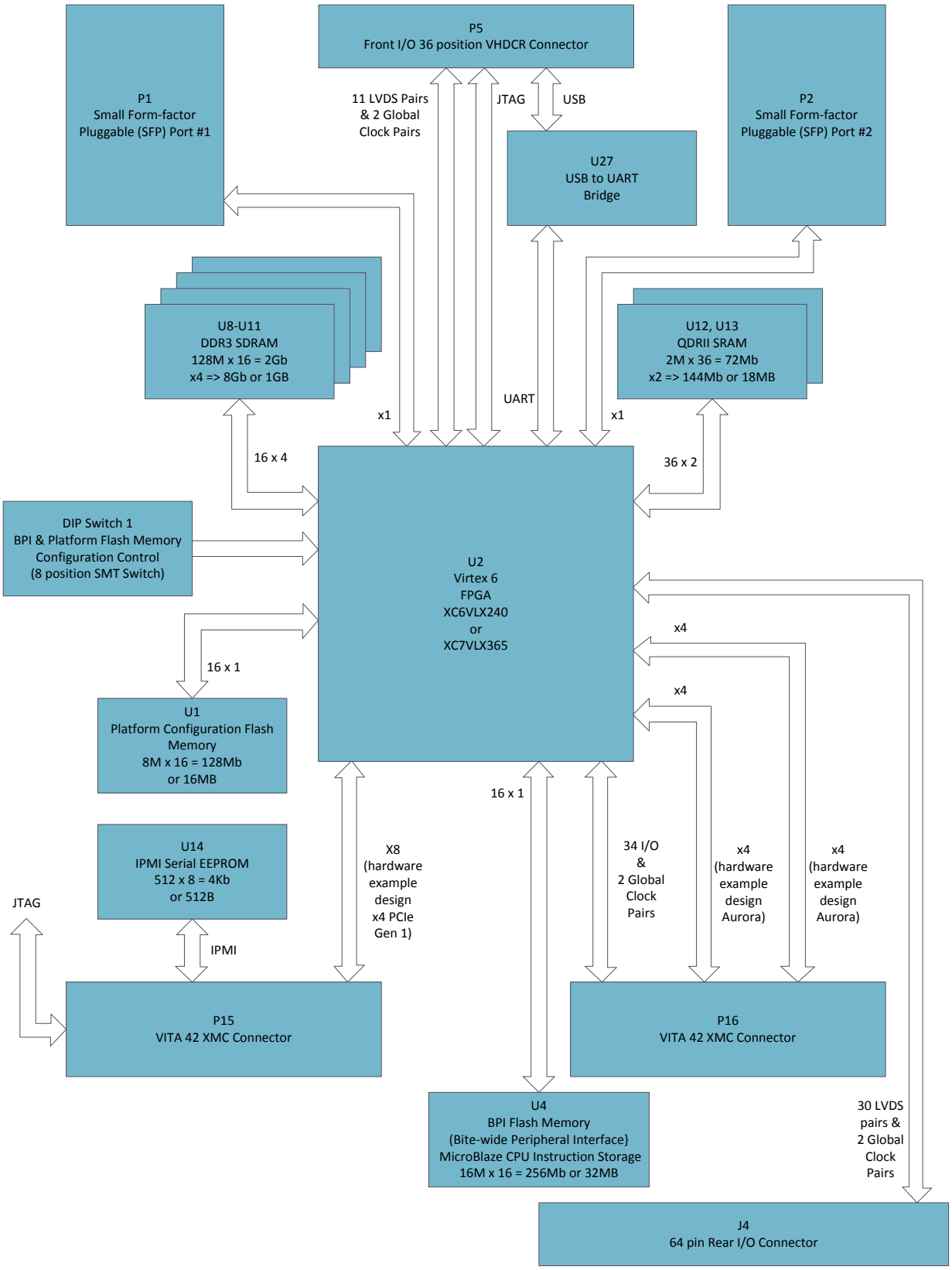
ANSI/VITA 42.3: XMC module with PCI Express Interface

4M Byte Memory Space Required (BAR0): 64 bit Base Address Register for access to Flash Configuration Registers, and System Monitor Registers, Front, Rear, and P16 I/O Registers.

16M Byte Memory Space Required (BAR2): 64 bit Base Address Register for access to QDR memory.

Interrupts: Source of interrupt can be from the programmable FPGA. Messages are used to assert and de-assert virtual interrupt lines on the link to emulate the Legacy PCI interrupt INTA# signal. Design also supports interrupt requests via message signaled interrupts.

XMC-6VLX Block Diagram



Certificate of Volatility

Certificate of Volatility				
Acromag Model XMC-6VLX240F(E) XMC-6VLX365F(E)	Manufacturer: Acromag, Inc. 30765 Wixom Rd Wixom, MI 48393			
Volatile Memory				
Does this product contain Volatile memory (i.e. Memory of whose contents are lost when power is removed) ■ Yes □ No				
Type (SRAM, SDRAM, etc.) SRAM	Size: 2 Meg x 72-bit	User Modifiable ■ Yes □ No	Function: Data storage for FPGA	Process to Sanitize: Power Down
Type (SRAM, SDRAM, etc.) FPGA based RAM	Size: 128 Meg x 64-bit	User Modifiable ■ Yes □ No	Function: Data storage for FPGA	Process to Sanitize: Power Down
Non-Volatile Memory				
Does this product contain Non-Volatile memory (i.e. Memory of whose contents is retained when power is removed) ■ Yes □ No				
Type(EEPROM, Flash, etc.) Flash	Size: 16Mbyte	User Modifiable ■ Yes □ No	Function: Storage of Code for FPGA	Process to Sanitize: Clear Flash memory by erasing all sectors of the Flash
Type(EEPROM, Flash, etc.) Flash	Size: 32Mbyte	User Modifiable ■ Yes □ No	Function: Storage of Code for MicroBlaze	Process to Sanitize: Clear Flash memory by erasing all sectors of the Flash
Type(EEPROM, Flash, etc.) Flash	Size: 512x8-bit	User Modifiable □ Yes ■ No	Function: Storage of Code for IPMI Interface Device	Process to Sanitize: Not Applicable
Acromag Representative				
Name: Joseph Primeau	Title: Dir. of Sales and Marketing	Email: jprimeau@acromag.com	Office Phone: 248-295-0823	Office Fax: 248-624-9234

Revision History

The following table shows the revision history for this document:

Release Date	Version	EGR/DOC	Description of Revision
10-OCT-12	A	LMP	Initial Acromag release.
17-DEC-12	B	LMP/LMP	Removed reference to extended temperature grade products. Added power supply requirements.
14-AUG-13	C	LMP/LMP	Additional text added to pages 64-65 describing Ethernet and Microblaze. Added text describing removal of heat sink to allow XMC-6VLX mount into a conduction cooled carrier.

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