Series PMC-VFX70
Virtex-5 Based FPGA
PMC Module

USER'S MANUAL

ACROMAG INCORPORATED
30765 South Wixom Road
P.O. BOX 437
Wixom, MI 48393-7037 U.S.A.
solutions@acromag.com

Tel: (248) 295-0310
Fax: (248) 624-9234

8500-813-D09G006

TABLE OF CONTENTS

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.

1.0 General Information

2.0 PREPARATION FOR USE

3.0 PROGRAMMING INFORMATION

4.0 THEORY OF OPERATION

5.0 SERVICE AND REPAIR

6.0 SPECIFICATIONS

DRAWINGS

Trademarks are the property of their respective owners.

RELATED PUBLICATIONS

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

Virtex-5 Data Book	**http://www.xilinx.com**
IDT70T3519S Spec.	**http://www.idt.com**
MT47H64M16HR Spec	**http://www.micron.com**
CY23EP05 Specification	**http://www.cypress.com**

1.0 GENERAL INFORMATION

The re-configurable PMC-VFX modules use the Xilinx Virtex-5 FX FPGA. Re-configuration of the FPGA is possible via a direct download into the Xilinx FPGA over the PCI bus. In addition, on board flash memory can be loaded with FPGA configuration data for automatic Xilinx configuration on power-up. Flash programming is also implemented over the PCI bus.

The example design includes an interface to the user rear I/O and front I/O connectors and an example memory interface controller to the 64M x 32-bit DDR-SDRAM. The example design also includes an interface to the 256K x 64-bit SRAM with DMA hardware support.

Table 1.1: *The PMC-VFX boards are available in both standard and extended temperature ranges*

MODEL	FPGA	OPERATING TEMPERATURE RANGE
PMC-VFX70	XC5VFX70T	0°C to +70°C
PMC-VFX70E	XC5VFX70T	-40°C to +85°C

KEY FEATURES

- **Reconfigurable Xilinx FPGA** – In system configuration of the FPGA is performed through a flash configuration device or via the PCI bus. This provides a means for creating custom user defined designs.

- **64M x 32 DDR-SDRAM –** A 64M x 32-bit double data rate (DDR2) dynamic random-access memory (DRAM) is directly accessed from the PowerPC CPU core. The DDR memory is also accessible from the Xilinx user-programmable FPGA logic and PCI bus.

- **256K x 64 Dual-Port SRAM –** A 256K x 64-bit dual-port static random access memory (SRAM) is included. One port of the SRAM provides a direct link from the PCI bus to the SRAM memory. The second port of the SRAM provides a direct link to the Xilinx user programmable FPGA.

- **Interface to Front Multifunction Modules –** Various mezzanine modules ("AXM" model prefex), ordered separately, allow the user to select the Front I/O required for their application.

- **Interface to Rear P4 Connector –** The Virtex 5 FPGA is directly connected to 64 pins of the rear P4 connector. All 2.5volt IO standards supported by the Virtex 5 device are available. The example design provides low voltage differential signaling as 32 LVDS input/output signals.

- **Write Disable Jumper –** User configurable flash memory can be hardware write disabled by removal of an on board zero ohm surface mount resistor.

- **FPGA Interfaces –** The example VHDL design includes implementation of the Local bus interface, control of digital front and rear I/O, SRAM read/write interface logic, and DDR2-SDRAM memory interface from the Local bus.

- **Embedded PowerPC CPU Core –** The example provides PowerPC access to the DDR SDRAM and internal FPGA Block RAM from which the boot loop program runs. Also, a PowerPC to Flash data FIFO interface is used for loading of the Flash Program code into DDR memory. Two UART interfaces for PowerPC application development are also provided in the example design.

PCI INTERFACE FEATURES

- **PCI Bus Modes -** The board supports PCI-X at 100MHz, 66MHz and 33MHz.

- **PCI Bus Master –** The PCI interface logic becomes the bus master to perform DMA transfers.

- **DMA Operation –** The PCI bus interface supports two independent DMA channels capable of transferring data to and from the on board SRAM. The example design implements DMA block and demand modes of operation.

- **64, 32, 16, 8-bit I/O** - Register Read/Write is performed through data transfer cycles in the PCI memory space. All registers can be accessed via 32, 16, or 8-bit data transfers. Access to Dual Port Memory can be accessed via 64, or 32-bit transfers.

- **Compatibility** – Complies with PCI Local Bus Specification Revision 3.0. Provides one multifunction interrupt. Board is 3.3V signaling compliant.

- **Supply Voltage Requirement –** The board requires that 3.3 volts external power be provided on the 3.3 volt signal lines of the PCI bus connector.

ENGINEERING DESIGN KIT

Acromag provides an engineering design kit for the VFX boards (sold separately), a “must buy” for first time VFX module purchasers. The design kit (model PMC-VFX-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, Xilinx XBD file, and other utility files. The VFX modules are intended for users fluent in the use of Xilinx FPGA design tools. An Acromag Getting Started Guide and the Acromag custom XBD help simplify the development of custom applications.

BOARD DLL CONTROL SOFTWARE

Acromag provides a software product (sold separately) to facilitate the development of 32-bit Windows (2000/XP/Vista®) applications for Acromag PMC I/O board products, PCI I/O Cards, and CompactPCI I/O Cards. This software (Model PCISW-API-WIN) consists of low-level drivers and Windows 32 Dynamic Link Libraries (DLLs) that are compatible with a number of programming environments including Visual C++™, Visual Basic .NET®, Borland C++ Builder® and others. 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 I/O board products, PCI 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 boards.

BOARD QNX SOFTWARE

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

2.0 PREPARATION FOR USE

UNPACKING AND INSPECTION

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

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

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

CAUTION
SENSITIVE ELECTRONIC DEVICES
DO NOT SHIP OR STORE NEAR STRONG ELECTROSTATIC, ELECTROMAGNETIC, MAGNETIC OR RADIOACTIVE FIELDS

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

CARD CAGE CONSIDERATIONS

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

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

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

IMPORTANT: *Adequate air circulation or conduction cooling must be provided to prevent a temperature rise above the maximum operating temperature.*

BOARD CONFIGURATION

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 as follows:

- The on board flash memory device (U6) is read/write enabled.
- The default configuration of the programmable software control register bits at power-up are described in section 3.
- The control registers must be programmed to the desired configuration before starting data input or output operation.

Front Panel Field I/O Connector

The front panel connector provides the field I/O interface connections via optional mezzanine I/O modules, purchased separately.

Rear P4 Field I/O Connector

The rear I/O P4 PMC 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 IO Standards. The IOSTANDARD attribute can be set in the user constraints file (UCF). For example, rear I/O can be defined for LVCMOS25 (low voltage CMOS). The example design defines the rear I/O to LVDS_25 (Low-Voltage Differential Signaling) in the user constraints file. The 2.5 volt IO Standards available are listed in table 6-39 of the Virtex 5 User Guide available from Xilinx.

The example design defines the rear I/O connector with 32 LVDS I/O pairs. The LVDS pairs are arranged in the same row in table 2.1. For example, RIO0_P and RIO0_N form a signal pair. The P identifies the Positive input while the N identifies the Negative input.

***Table 2.1:** Board Rear Field I/O Pin Connections*

The example design implements 2.5volt LVDS I/O to the rear connector. Signal pairs are routed to pins (1,2), (3,4) etc.

Ch.	Positive Pin Description	Pin	Negative Pin Description	Pin
0	RIO0_P	1	RIO0_N	2
1	RIO1_P	3	RIO1_N	4
2	RIO2_P	5	RIO2_N	6
3	RIO3_P	7	RIO3_N	8
4	RIO4_P	9	RIO4_N	10
5	RIO5_P	11	RIO5_N	12
6	RIO6_P	13	RIO6_N	14
7	RIO7_P	15	RIO7_N	16
8	RIO8_P	17	RIO8_N	18
9	RIO9_P	19	RIO9_N	20
10	RIO10_P	21	RIO10_N	22
11	RIO11_P	23	RIO11_N	24
12	RIO12_P	25	RIO12_N	26
13	RIO13_P	27	RIO13_N	28
14	RIO14_P	29	RIO14_N	30
15	RIO15_P	31	RIO15_N	32
16	RIO16_P	33	RIO16_N	34
17	RIO17_P	35	RIO17_N	36
18	RIO18_P	37	RIO18_N	38
19	RIO19_P	39	RIO19_N	40
20	RIO20_P	41	RIO20_N	42
21	RIO21_P	43	RIO21_N	44
22	RIO22_P	45	RIO22_N	46
23	RIO23_P	47	RIO23_N	48
24	RIO24_P	49	RIO24_N	50
25	RIO25_P	51	RIO25_N	52
26	RIO26_P	53	RIO26_N	54
27	RIO27_P	55	RIO27_N	56
28	RIO28_P	57	RIO28_N	58
29	RIO29_P	59	RIO29_N	60
30	RIO30_P	61	RIO30_N	62
31	RIO31_P	63	RIO31_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/CPU board (AMP 120521-1 or equivalent).

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.

Standalone Operation

Powering the PMC-VFX as an independent board is possible using the J7 board through holes. As an independent standalone board the PMC-

VFX board would not be plugged into a PMC slot. It must, in this case, receive power though the J7 contact holes and provide passive resistive pullups on all PCI bus signals required to be pulled up by the system card. Note, the board could be damaged if the required pull-up resistors are not used. To independently power the board, the required +5 volt and +3.3 volt power supplies must be provided via the J7 contact holes. The holes have 29 mil openings with 60 mil pads. The location of the J7 contact holes on the board are shown in diagram 4502-083 at the end of this manual.

Flash Write Disable Resistor

By default the flash memory (U6) is read/write enabled. Removal of resistor R172 disables writing the flash configuration device. Refer to Resistor Location Drawing 4502-088, at the end of this manual, to identify the board location of R172.

3.0 PROGRAMMING INFORMATION

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

GETTING STARTED

1. The PMC VFX board is shipped with the user-programmable Xilinx FPGA code stored in flash memory. Upon power-up the PMC VFX will automatically configure the FPGA with the example design code stored in flash. As a first step become familiar with the PMC VFX, as supplied by Acromag. The board will perform all the functions of the example design.

   The Example Design Memory Map section gives a description of the I/O operations performed by the example design. It will allow testing of digital I/O, interrupts, read/write of dual port SRAM, read/write of double data rate SDRAM, testing of both DMA channels, and PowerPC program execution. It is strongly recommended that you become familiar with the board features by using the example design as provided by Acromag.

   Do not attempt to reconfigure the flash memory until after you have tested and become familiar with the PMC VFX as provided in the example design.

2. After you are familiar with the PMC VFX 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. It is recommended that you test this modified example design using the reconfiguration direct method. It is not recommended that the flash be overwritten until you have tested your code. The reconfigure direct method will allow programming of the FPGA directly from the PCI bus. If for some reason the PMC VFX does not perform as expected, you can power the PMC VFX down. Upon power-up, the example design provided by Acromag will again be loaded into the FPGA.

   The general procedure is to first develop your PowerPC code while leaving the reprogrammable FPGA example code in flash memory. The reprogrammable FPGA example design after power-up or direct reconfiguration will run a bootloop program from Block RAM. See the PMC-VFX Getting Started Guide for an overview of the bootloop program.

See the Direct PCI bus to Xilinx Configuration section for a description of the steps required to perform reconfiguration directly from the PCI bus. The registers provided in the FPGA Programming Memory Map are used to implement a direct reconfiguration.

3. After you have thoroughly tested your customized FPGA design, you can erase the flash and write your code to flash. 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.

   See the Flash Configuration section for a description of the steps required to write new code or to reprogram the example design code into the flash memory. The registers provided in the FPGA Programming Memory Map are used to implement flash erase and reprogram operations.

PCI CONFIGURATION ADDRESS SPACE

This board is a PCI Specification version 3.0 compliant PCI bus master/target 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 PCI 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 PCI card. As a Plug-and-Play card the board's base address and system interrupt request line are not selected via jumpers but are assigned by system software upon power-up via the configuration registers. A PCI bus configuration access is used to read/write the PCI card's configuration registers.

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

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

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 and which interrupt line will be used.

The memory maps in this chapter reflect byte accesses using the "Little Endian" byte ordering format. Little Endian uses even-byte addresses to store the low-order byte. Little-Endian means that the least significant byte is stored at the lowest memory address and the most significant byte is stored at the highest memory address. The Intel x86 family of microprocessors uses "Little Endian" byte ordering.

Big Endian is the convention used in the Motorola 68000 microprocessor family and is the VMEbus convention. In Big Endian, the lower-order byte is stored at odd-byte addresses. Big-endian means that the most significant byte is stored at the lowest memory address and the least significant byte is stored at the highest memory address.

	Low address	**Layout of a 64-bit long int**						**High address**
Little Endian	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Big Endian	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0

CONFIGURATION REGISTERS

The PCI 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 Line Register which must be read to determine the base address assigned to the board and the interrupt request line 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** 0x5605 (VFX70)		**Vendor ID** 16D5	
1	Status		Command	
2	Class Code=118000			Rev ID=00
3	BIST	Header	Latency	Cache
4	32-bit Memory Base Address for Memory Accesses to PCI interrupt, and DMA Registers, 4K Space**(PCIBAR0)**			
5	32-bit Memory Base Address/64-bit Data to Dual Port Memory, 4M Space**(PCIBAR1)**			
6	32-bit Memory Base Address/32-bit Data to Virtex 5 User Registers, 4M Space**(PCIBAR2)**			
7 : 10	Not Used			
11	**Subsystem ID** 0x5605 (VFX70)		**Subsystem Vendor ID** 16D5	
12	Not Used			
13,14	Reserved			
15	Max_Lat	Min_Gnt	Inter. Pin	Inter. Line

This board is allocated memory space address (PCIBAR0) to access the PCI interrupt, and DMA registers. The PCI bus decodes 4K bytes for these memory space registers. PCIBAR0 space is accessed using 32-bit data transfers. This board is allocated a 4M byte block of memory (PCIBAR1) that is addressable in the PCI bus memory space to access the board’s Dual Port Memory using 32-bit or 64-bit transfers.

In addition, this board is allocated a 4M byte block of memory (PCIBAR2) that is addressable in the PCI bus memory space. PCIBAR2 space is used to access the board's flash configuration functions and the reprogrammable Virtex 5 FPGA functions via 32-bit data transfers.

PCIBAR0 MEMORY MAP

The PCIBAR0 registers are implemented in the PCI bus interface chip and not the user programmable FPGA. As such, the user cannot change the logic functions implemented in PCIBAR0. These registers are read/write registers that are software controlled. These registers provide interrupt control/status and DMA control/status. The Interrupt Control/Status is at PCIBAR0 base address plus 00H offset. The DMA registers are at PCIBAR0 base address plus offset 100H to 124H. These registers control the transfer direction, size, system address, and PMC addresses for DMA channels 0 and 1.

Table 3.2: *PCIBAR0 Registers*

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

PCIBAR0 Base Addr+	Bit(s)	Description
00H	31:0	Interrupt Control/Status
04H	31:0	DMA Status/Abort Register
08H	31:0	Global Interrupt Enable (Bit-31)
0CH -> FFH	31:0	Reserved
100H	31:0	DMA Channel 0 System Starting Address
104H	31:0	DMA Channel 0 PMC Board Starting Address
108H	31:0	DMA Channel 0 Transfer Size in bytes
10CH	7:0	DMA Channel 0 Command
110H	0	DMA Channel 0 Start DMA Transfer Bit
114H	31:0	DMA Channel 1 System Starting Address
118H	31:0	DMA Channel 1 PMC Board Starting Address
11CH	31:0	DMA Channel 1 Transfer Size in bytes
120H	7:0	DMA Channel 1 Command
124H	0	DMA Channel 1 Start DMA Transfer Bit
128H->FFFH	31:0	Reserved

The Dual Port SRAM control registers at PCIBAR2 must also be used to set up a DMA Demand Mode transfer. The Demand mode transfer is initiated by driving signals DREQ0# or DREQ1# active. The SRAM control register method allows a DMA transfer to be initiated when an FPGA generated address counter is equal to the DMA Channel Threshold

Register. That is, when the predetermined amount of data is available in the SRAM the hardware will automatically start a DMA transfer.

Interrupt Control/Status Register (Read/Write) - (PCIBAR0 + 00H)

INTERRUPT REGISTER

This Interrupt Control/Status register at PCIBAR0 base address + offset 00H is used to monitor and clear pending board interrupts. An interrupt can originate from the two DMA channels or U7, the user-programmable FPGA. All board interrupts are enabled or disabled via bit-31 of the Global Interrupt Enable register at PCIBAR0 + 08H.

Bit(s)	FUNCTION	
0	This bit when set indicates a pending board interrupt. It reflects a pending interrupt from DMA channel 0 or DMA channel 1 or the U7 FPGA. It will reflect this status even if the Board Interrupt enable bit-31 is disabled.	
	0	No Interrupt Pending
	1	Interrupt Pending
1	DMA Channel 0 Interrupt Pending Status. Bit-16 must be set to logic high for this bit to go active. Write logic high to clear bit.	
	0	No Interrupt Pending
	1	Interrupt Pending
2	DMA Channel 1 Interrupt Pending Status. Bit-17 must be set to logic high for this bit to go active. Write logic high to clear bit.	
	0	No Interrupt Pending
	1	Interrupt Pending
3	U7 Programmable FGPA Interrupt Pending Status. Bit-18 must be set to logic high for this bit to go active.	
	0	No Interrupt Pending
	1	Interrupt Pending
4-15	Not Used (bits are read as logic "0")	
16	DMA Channel 0 Interrupt Enable	
	0	DMA Channel 0 Interrupt Disabled
	1	DMA Channel 0 Interrupt Enabled
17	DMA Channel 1 Interrupt Enable	
	0	DMA Channel 1 Interrupt Disabled
	1	DMA Channel 1 Interrupt Enabled
18	U7 Programmable FPGA Interrupt Enable	
	0	Interrupt Disabled
	1	Interrupt Enabled
19-31	Not Used (bits are read as logic "0")	

Table 3.3: *Interrupt Control/Status Register When designing software drivers it is best to treat this register as two 16-bit registers. The upper 16-bits are Interrupt Control bits and the lower 16-bits are Interrupt Status.*

A board pending interrupt is identified via bit-0 of this register. Logic high on bit-0 indicates a board pending interrupt. Bit-0 indicates a pending interrupt as long as DMA Channel 0 or DMA Channel 1 or U7 Programmable FPGA interrupt pending status bits 1, 2, or 3 respectively, remain active.

A DMA channel 0 interrupt pending status can be cleared/released by writing logic high to bit-1, the interrupt pending status bit. Likewise, writing logic high to bit-2 of this register clears DMA channel 1 interrupt pending status. U7 Programmable FPGA interrupt Pending status will pass the interrupt status of U7 only when bit-18 is set to logic high.

The Software Reset and Status Register (see Table 3.7) at PCIBAR2 + 8000H can be read to identify the exact source of the Programmable Virtex-5 FPGA interrupt.

Bits 16 to 18 of this register are used to enable or disable interrupts from specific functions. This Interrupt register must have bits 16 and 17 set to a logic high in order for DMA interrupts to occur on DMA channels 0 and 1, respectively. Bit-18 must be set to logic high to enable interrupts from U7, the programmable FPGA.

If present, the mezzanine board interrupt enable bits must also be set if interrupts are to originate from the mezzanine board which are passed through the programmable FPGA to this register's pending status bits.

DMA BAR0 REGISTERS

DMA Status/Abort Register (Read/Write) - (PCIBAR0 + 04H)

Table 3.4 *DMA Status Register*

Bit(s)	FUNCTION	
0	DMA Channel 0 Transfer Complete. This bit is cleared by write of logic high to this bit or start of a new DMA transfer.	
	0	Transfer not Complete
	1	Transfer Completed
1	DMA Channel 1 Transfer Complete. This bit is cleared by write of logic high to this bit or start of a new DMA transfer.	
	0	Transfer not Complete
	1	Transfer Completed
2-7	Not Used (bits are read as logic "0")	
8	DMA Channel 0 transfer abort on write of logic high to this bit.	
	0	No Action
	1	Abort Channel 0 DMA transfer
9	DMA Channel 1 transfer abort on write of logic high to this bit.	
	0	No Action
	1	Abort Channel 1 DMA transfer
10-15	Not Used (bits are read as logic "0")	
16-19	DMA Channel 0 State Encoding	
	0000	Transfer Completed Successfully
	0001 to 0111	Transfer Aborted
	1000 to 1100	Transfer not yet completed
20-23	DMA Channel 1 State Encoding	
	0000	Transfer Completed Successfully
	0001 to 0111	Transfer Aborted
	1000 to 1100	Transfer not yet completed
24-31	Not Used (bits are read as logic "0")	

DMA BAR0 REGISTERS

This DMA Status register at PCIBAR0 base address plus 04H is used to identify a DMA transfer complete status and issue a DMA channel abort.

The DMA complete status bit 0 or 1 will remain logic high until cleared by writing logic high back to the same bit. The start of a new DMA transfer (software or hardware initiated) will also clear a set Transfer Complete bit.

DMA transfers must start aligned to a Lword boundary when implemented as a 32-bit transfer and a Double Lword boundary when performing 64-bit DMA transfers.

This register can be read or written via 32-bit, 16-bit or 8-bit data transfers.

Global Interrupt Enable Bit-31 (Read/Write) - (PCIBAR0 + 08H)

GLOBAL INTERRUPT ENABLE

This Global Interrupt Enable bit at PCIBAR0 base address + offset 08H is used to enable all board interrupts. An interrupt can originate from the two DMA channels or U7, the programmable FPGA. All board interrupts are enabled when bit-31 is set to logic high. Likewise, board interrupts are disabled with bit-31 set to logic low. Bit-31 of this register can be read or written.

Table 3.5 *Global Interrupt Enable Bit*

Bit(s)	FUNCTION	
0-30	Not Used (bits are read as logic "0")	
31	PMC Board Interrupt Enable. This bit must be set to enable the PCI bus interrupt signal to be driven active by the board.	
	0	Board Interrupts Disabled
	1	Board Interrupts Enabled

DMA System Starting Address Registers (Read/Write) - (PCIBAR0 + 100H and 114H)

The DMA System Starting Address register meaning depends on the selected DMA mode (see bit-3 of DMA command register). For Direct DMA Mode this address register specifies the physical address of a contiguous memory buffer where data will be read/written. For scatter-gather DMA mode this address register points to the first element of the chained-listed of page descriptors.

The DMA System Starting Address Register at PCIBAR0 base address plus 100H (114H) is used to set the DMA channel 0 (1) data starting address. Writing to these registers is possible via 32-bit, 16-bit or 8-bit data transfers.

DMA PMC Board Starting Address Registers (Read/Write) - (PCIBAR0 + 104H and 118H)

The DMA PMC Board Starting Address register specifies the physical address of the board's Dual Port SRAM memory where data will be read/written.

The DMA PMC Board Starting Address Register at PCIBAR0 base address plus 104H (118H) is used to set the DMA channel 0 (1) data starting address. Writing to these registers is possible via 32-bit, 16-bit or 8-bit data transfers.

DMA Transfer Size Registers (Read/Write) - (PCIBAR0 + 108H and 11CH)

The DMA Transfer Size Register is used to set the size of the DMA transfer that moves data between system memory and the board's Dual Port SRAM. The transfer size indicates the total amount of data to transfer, in units of bytes.

The onboard static RAM has 2-MegaByte maximum capacity. As such, the maximum value that can be written to this register is 1FFFFF hex which corresponds to 2^{21}- 1.

The DMA Transfer Size Register at PCIBAR0 base address +108H (11CH) is used to set the DMA channel 0 (1) data transfer size. Writing to these registers is possible via 32-bit, 16-bit or 8-bit data transfers.

DMA Command Registers (Read/Write) - (PCIBAR0 + 10CH and 120H)

The DMA Command Register is used to set the priority, 64-bit versus 32-bit mode, Scatter Gather enable, and to indicate the command to be used for the DMA transfer.

The DMA Command Register at PCIBAR0 + 10CH (1200H) is used for DMA channel 0 (1) commands. Writing to these registers is possible via 32-bit, 16-bit or 8-bit data transfers.

***Table 3.6** DMA Command Register*

DMA transfers must start aligned to a Lword boundary when implemented as a 32-bit transfer and a Double Lword boundary when performing 64-bit DMA transfers.

Bit(s)	FUNCTION	
0	Not Used (bit is read as logic '0')	
1	Priority: Setting this bit marks the DMA channel as high-priority. A higher-priority channel takes precedence over the second channel and thus has access to the PCI bus more quickly.	
	0	Low Priority
	1	High Priority
2	64-bit or 32bit Mode	
	0	32-bit Mode DMA Transfers
	1	64-Bit Mode DMA Transfers
3	Scatter-gather: Setting this bit enables scatter-gather mode	
	0	Direct DMA Mode
	1	Scatter-Gather Enabled
7 to 4	DMA Command	
	'1110'	Memory Read Burst (Memory Read Line)
	'1111'	Memory Write Burst (Memory Write and Invalidate)
8 to 11	DMA Byte Enables	
	'0000'	All bytes are transferred. These bits should always be logic low

DMA Start Transfer Register (Write) - (PCIBAR0 + 110H and 124H)

The DMA Start Transfer Register is used software start a DMA transfer. A channel 0 (1) start transfer is initiated when PCIBAR0 + 110H (124H) is written. Writing to these registers is possible via 32-bit, 16-bit or 8-bit data transfers.

PCIBAR2 MEMORY MAP

The memory space address map used to program the FPGA and flash device is shown in Table 3.7. Note that the base address for the board (PCIBAR2) in memory space must be added to the addresses shown to properly access these registers. Register accesses as 32, 16, and 8-bit transfers in memory space are permitted. All addresses in PCIBAR2 from 0 to 7FFF hex are fixed and cannot be changed by the user via the programmable Virtex5 FPGA.

Table 3.7
PCIBAR2 Memory Map

BAR2 Addr+	D31 D08	D07 D00	BAR2 Addr+
0003	Not Used[1]	Configuration Status Register	0000
0007	Flash Logic Software Reset Bit-15	Configuration Control Register	0004
000B	Not Used[1]	Configuration Data	0008
000F	Not Used[1]	Flash Status 1 Register	000C
0013	Not Used[1]	Flash Status 2 Register	0010
0017	Not Used[1]	Flash Read	0014
001B	Not Used[1]	Flash Reset	0018
001F	Not Used[1]	Flash Start Write	001C
0023	Not Used[1]	Flash Erase Sector	0020
0027	Not Used[1]	Flash Erase Chip	0024
002B	Not Used[1]	Flash Data Register	0028
002F	Not Used[1]	Flash Address 7->0	002C
0033	Not Used[1]	Flash Address 15->8	0030
0037	Not Used[1]	Flash Address 23->16	0034
003B	Not Used[1]	Flash Address 24	0038
003F	PCI bus FPGA System Monitor Status/Control Register		003C
0043	PCI bus FPGA System Monitor Address Register		0040
↓	Not Used[1]		↓
7FFF	Not Used[1]		7FFC

[1] The board will return 0 for all addresses that are "Not Used".

EXAMPLE DESIGN MEMORY MAP

Table 3.7: Example Design BAR2 Memory Map

1. The board will return 0 for all addresses that are "Not Used".

BAR2 Addr+	D31 D16	D15 D00	BAR2 Addr+
8003	Software Reset and Status Register		8000
8007 ↓ 802B	Mezzanine Module Memory Space		8004 ↓ 8028
802F	Rear I/O Connector Read Register		802C
8033	Rear I/O Connector Write Register		8030
8037	DMA Control Register		8034
803B	FPGA Port SRAM Register Data Lines 31 to 0		8038
803F	FPGA Port SRAM Register Data Lines 63 to 32		803C
8043	FPGA Port-SRAM Control Register		8040
8047	FPGA Port-SRAM Address Register		8044
804B	DMA Channel 0 Threshold Register (DP-SRAM)		8048
804F	DMA Channel 1 Threshold Register (DP-SRAM)		804C
8053	Address Reset Register 0 (DP-SRAM)		8050
8057	Address Reset Register 1 (DP-SRAM)		8054
805B	PMC Board Identification Code: “A3” for Acromag Example Design		8058
805F	DDR-SDRAM Control/Status Register		805C
8063	DDR-SDRAM Address Register		8060
8067	DDR-SDRAM Read Register		8064
806B	DDR-SDRAM Write Register		8068
806F	DDR-SDRAM Mask Register		806C
8073	PowerPC Read Register		8070
8077 ↓ 8087	Not Used[1]		8074 ↓ 8084
808B	Reprogrammable FPGA System Monitor Status/Control Register		8088
808F	Reprogrammable FPGA System Monitor Address Register		808C
8093 ↓ 3FFFFF	Additional Mezzanine Module Space 8100-> 8137 Otherwise Not Used[1]		8090 ↓ 3FFFFC

This memory map reflects byte accesses using the "Little Endian" byte ordering format. Little Endian uses even-byte addresses to store the low-order byte. The Intel x86 family of microprocessors uses "Little Endian" byte ordering. In Big Endian, the lower-order byte is stored at odd-byte addresses.

Flash Configuration

The VFX board uses a flash configuration device to store programming information for the Xilinx reprogrammable FPGA. In addition, the VFX board uses the second half of the flash configuration device to store PowerPC program code and data. The flash configuration device and FPGA are hardwired together so that during power-up the contents of the configuration device are downloaded to the FPGA.

Flash Reprogrammable FPGA Configuration

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

1) Disable auto-configuration by setting bit-0 (Stop Configuration) of the Configuration Control register to logic high.
2) Clear the Xilinx FPGA of its previous configuration by setting the Configuration Control register bit-2 to logic high. Software must also keep bit-0 set to logic high.
3) Read INIT as logic high (Bit-1 of Configuration Status register) before programming is initiated.
4) Verify that the Flash Chip is not busy by reading bit-7, of the Flash Status 2 register at base address plus 10H, as logic 0 before starting a new Flash operation.
5) Erase the current flash contents by using the Flash Erase Sector method. The first 128 sectors (0 to 127) are reserved for the reprogrammable FPGA configuration data. The next 128 sectors (128 to 255) are reserved for PowerPC program code and data. Flash erase sectors are implemented by setting bit-0 of the Flash Erase Sector register to logic high. There are 256 flash sectors, which are addressed via the most significant eight flash address lines. The most significant flash address line is set via the Flash Address 24 register at base address plus 38H. The next seven address lines A23->A17 are at Flash Address 23-17 register at base address plus 34H. Issuing a Flash Erase Sector command will erase the contents of the flash chip only in the sector specified.
6) Verify that the Flash Chip is not busy by reading bit-7, of the Flash Status 2 register at base address plus 10H, as logic 0 before going to the next step.
7) After the first 128 sectors have been erased, download the Configuration file to the flash configuration chip via the PCI bus.
   i) Write the byte to be sent to the Flash Data register at base address plus 28H.
   ii) Write the address of the Flash Chip to receive the new data byte to the Flash Address registers at base address plus 2CH, 30H, 34H, and 38H. Issuing a Flash Start Write will automatically increment this address after the prior Flash Write has been completed. Thus, the address will not need to be set prior to issuing the next Flash Start Write. The first byte of the reprogrammable FPGA configuration file should be written to address 0 of the Flash Chip. The Flash Start Write operation will take 9µ seconds to complete.

iii) Issue a Flash Start Write command to the Flash Chip by writing logic 1 to bit-0 of base address plus 1CH.
iv) Verify that the Flash Chip is not busy by reading bit-7 as logic 0 of the Flash Status 2 register at base address plus 10H before going back to step i to write the next byte.

8) Enable auto-configuration by setting bit-0 (Stop Configuration) of the Configuration Control register to logic low.
9) Verify that the configuration is complete by reading DONE (bit-0 of Configuration Status Register) as logic high.
10) Thereafter, at power-up the configuration file will automatically be loaded into the FPGA.

Flash PowerPC Program Code and Data

The PowerPC program code and data is stored to flash using the PCI bus interface. The PowerPC code and data are stored to flash sectors 128 to 255. The second half of the 256 sector flash device is reserved for PowerPC program code and data. The following is the general procedure for reprogramming the flash memory with the PowerPC program code and data:

1) Verify that the Flash Chip is not busy by reading bit-7, of the Flash Status 2 register at base address plus 10H, as logic 0 before starting a new Flash operation.
2) Erase the current PowerPC program code flash contents by using the Flash Erase Sector method. Do not erase the first 128 sectors (0 to 127) they are reserved for the reprogrammable FPGA configuration data. The next 128 sectors (128 to 255) are reserved for PowerPC program code and data. Flash sectors 128 to 255 must be erased before new PowerPC code and data can be written to flash. Flash erase sectors are implemented by setting bit-0 of the Flash Erase Sector register to logic high. There are 256 flash sectors, which are addressed via the most significant eight flash address lines. The most significant seven flash address lines are set via the Flash Address 24 register at base address plus 38H Flash Address 23-17 register at base address plus 34H. Issuing a Flash Erase Sector command will erase the contents of the flash chip only in the sector specified.
3) Verify that the Flash Chip is not busy by reading bit-7, of the Flash Status 2 register at base address plus 10H, as logic 0 before going to the next step.
4) After sectors 128-255 have been erased, download the PowerPC program code and data file to the flash chip via the PCI bus. The size of the PowerPC file in bytes must be written to the first four locations in sector 128. The first PowerPC program code value is written to sector 128 plus 5. This PowerPC program code/data must be written as a multiple of 4 bytes. See table 3.8 for the byte order of the data written to flash.

Table 3.8: *PowerPC Flash Memory Organization*

Flash Sector Address	Data Bits	Description
128+0	D31 to D24	PowerPC Code Size (Number of Bytes) Most Significant Byte
128+1	D23 to D16	PowerPC Code Size (Number of Bytes)
128+2	D15 to D08	PowerPC Code Size (Number of Bytes)
128+3	D07 to D00	PowerPC Code Size (Number of Bytes) Least Significant Byte
128+4	D31 to D24	First PowerPC Program Code
128+5	D23 to D16	First PowerPC Program Code
128+6	D15 to D08	First PowerPC Program Code
128+7	D07 to D00	First PowerPC Program Code
128+8	D31 to D24	Second PowerPC Program Code
128+9	D23 to D16	Second PowerPC Program Code
128+10	D15 to D08	Second PowerPC Program Code
128+11	D07 to D00	Second PowerPC Program Code
↓	↓	↓

i) Write the byte to be sent to the Flash Data register at base address plus 28H.

ii) Write the address of the Flash Chip to receive the new data byte to the Flash Address registers at base address plus 2CH, 30H, 34H, and 38H. Issuing a Flash Start Write will automatically increment this address after the prior Flash Write has been completed. Thus, the address will not need to be set prior to issuing the next Flash Start Write. The first byte of the reprogrammable FPGA configuration file should be written to address 0 of the Flash Chip. The Flash Start Write operation will take 9μ seconds to complete.

iii) Issue a Flash Start Write command to the Flash Chip by writing logic 1 to bit-0 of base address plus 1CH.

iv) Verify that the Flash Chip is not busy by reading bit-7 as logic 0 of the Flash Status 2 register at base address plus 10H before going back to step i to write the next byte.

5) Thereafter, at power-up the updated PowerPC program code/data will automatically be loaded into DDR2 memory.

Direct PCI bus to Xilinx Configuration

Configuration of the Xilinx FPGA can be implemented directly from the PCI bus. The following is the general procedure for re-configuration of the Xilinx FPGA via the PCI bus:

1) Disable auto-configuration by setting bit-0 (Stop Configuration) of the Configuration Control register to logic high.
2) Clear the Xilinx FPGA of its previous configuration by setting the Configuration Control register bit-2 to logic high.
3) Read INIT as logic high (Bit-1 of Configuration Status register) before programming is initiated.
4) Download the Configuration file directly to the Xilinx FPGA by writing to the Configuration Data register. The entire configuration file must be written to the Xilinx FPGA one byte at a time to the Configuration Data register at base address plus 08H.
5) Verify that the configuration is complete by reading DONE (bit-0 of Configuration Status Register) as logic high. DONE is expected to be logic high immediately after the last byte of the configuration file is written to the Xilinx FPGA.
6) Unless the flash memory is used, at each power-up the configuration file will need to be reloaded into the FPGA.

Direct PCI bus to DDR2 Memory of PowerPC Program Code and Data

The PowerPC program code and data can be written directly into DDR2 memory from the PCI bus. It is recommended that PowerPC program code and data be written directly to DDR2 memory while program development is in progress. This will allow one to always go back to the example design as provided in flash.

The general procedure is to first develop your PowerPC code while leaving the reprogrammable FPGA as provided in the example design. The reprogrammable FPGA example design, after power-up or direct reconfiguration, will run a bootloop program which moves the flash program code to DDR2 memory. After the data is moved into DDR memory, the bootloop program will write the Block RAM address 0xFFFFF038 with value 0x33333333. This 0x33333333 value indicates that the DDR program code has been moved to DDR memory. Next, the bootloop program will set Block RAM address 0xFFFFF030 to 0x33AA33AA to indicate that it has jumped to address 0x0 in DDR memory to start execution of the DDR resident program. After the PowerPC has executed the DDR 2 resident program, it jumps back to the bootloop program and stays in a while loop checking Block RAM address 0xFFFFF034 for the value 0x5A5A5A5A. When the PowerPC is in the while loop, a new program can be directly loaded into DDR2 memory using the PCI bus. After the new program is downloaded, the Block RAM address 0xFFFFF034 is written with value 0x0 and the bootloop program will again jump to address 0x0 in DDR memory to start execution of the DDR resident program.

The following is the general procedure for programming the DDR2 memory with the PowerPC program code and data:

1) Write the 32-bit data value that is to be written to the DDR-SDRAM to the DDR SDRAM Write registers at base address + 8068H.
2) Set the DDR-SDRAM Mask bits as desired at base address + 806CH. A value of 0H would enable all bytes to be written.
3) Set the DDR-SDRAM Address register at PCIBAR2 + 8060H with the DDR-SDRAM write address location. The DDR SDRAM

memory starts at PowerPC base address 0x0. The DDR SDRAM high address is 0x0FFFFFFF. The Block RAM memory starts at PowerPC base address 0xFFFF0000 and has a high address of 0xFFFFFFFF.

4) Issue the Write Command. Set bit-0 of the SDRAM Control/Status Register at PCIBAR2 + 805CH to logic high.
5) After the DDR2 program has been written to DDR2 memory, Block RAM address 0xFFFFF034 is written with value 0x0 using steps 1 to 4 above.

Read of the DDR memory is also supported. Read of DDR memory is described in the DDR SDRAM Read Register section.

CONFIGURATION CONTROL REGISTERS

Configuration Status Register (Read Only) – (PCIBAR2 + 00H)

This read only register reflects the status of configuration complete and Xilinx configuration clear bits. This Configuration Status register is read at base address plus 0H.

Table 3.9: *Configuration Status Register*

Bit(s)	FUNCTION	
0	DONE:	
	0	Xilinx FPGA is not configured
	1	Xilinx FPGA configuration is complete
1	INIT:	
	0	INIT is held low until the Xilinx FPGA is clear of its current configuration
	1	INIT transitions high when the clearing of the current Xilinx FPGA configuration is complete
2 to 7	Not Used (bits are read as logic "0")	

CONFIGURATION CONTROL REGISTERS

Configuration Control (Read/Write) – (PCIBAR2 + 04H)

This read/write register is used to stop Xilinx configuration, and clear Xilinx configuration memory. This Configuration Control register is accessed at base address plus 04H.

Table 3.10: *Configuration Control Register*

Bit(s)	FUNCTION	
0	Stop Xilinx Configuration:	
	0	Enable Xilinx FPGA configuration
	1	Stop Xilinx FPGA configuration (This bit should be set to logic high until after the Flash device is written with valid program data).
1	Not Used (bit is read as logic "0")	
2	Clear Current Xilinx Configuration:	
	0	Logic low has no effect.
	1	Logic high resets the Xilinx configuration logic. Re-configuration can begin after INIT transitions high.
3 to 14	Not Used (bits are read as logic "0")	
15	Logic "1" = Flash Logic Software Reset	

Configuration Data (Write Only) – (PCIBAR2 + 08H)

This write only register is used to write Xilinx configuration data directly to the Xilinx FPGA from the PCI bus. The Configuration Data register is accessed at base address plus 08H. The entire configuration file must be written to the Xilinx FPGA one byte at a time. Configuration complete is verified by reading DONE (bit-0 of the Configuration Status Register) as logic high.

A write to the Configuration Data register while auto-configuration from Flash is active will cause the Xilinx configuration to fail. Auto-configuration is stopped by writing logic 1 to bit-0 of the Configuration Control register at base address plus 04H.

The Xilinx FPGA should also be cleared of its current configuration prior to loading a new configuration file. The FPGA is cleared of its current configuration by writing logic 1 to bit-2 at address plus 04H.

FLASH CONTROL REGISTERS

Flash Status 1 (Read Only) – (PCIBAR2 + 0CH)

This read only register is used to read the DQ5 status of the flash chip. The Flash Status 1 register is at base address plus 0CH.

Table 3.11: *Flash Status 1 Register*

Bit(s)	FUNCTION	
0 to 4	Not Used (bits are read as logic “1 or 0”)	
5	DQ5:	
	0	Chip enabled for reading array data.
	1	The system must issue the Flash Reset command to re-enable the device for reading array data if DQ5 goes high. DQ5 will go high during a Flash Start Write, Flash Erase Chip, or Flash Erase Sector operation.
6 and 7	Not Used (bits are read as logic “1 or 0”)	

FLASH CONTROL REGISTERS

Flash Status 2 (Read Only) – (PCIBAR2 + 10H)

This read only register is used to read the ready or busy status of the flash chip. The Flash Status 2 register is at base address plus 10H. The system must first verify that that Flash Chip is not busy before executing a new Flash command. The Flash Chip is busy if bit-7 of this register is set to logic 1. The Flash will always be Busy while bit-0 of the Configuration Control register is set to logic “0”.

Table 3.12: *Flash Status 2 Register*

Bit(s)	FUNCTION	
0 to 6	Not Used (bits are read as logic “0”)	
7	Busy / Ready~ Set bit-0 of the Configuration Control register to logic “1” before monitoring this busy bit.	
	0	Flash Chip is Ready
	1	Flash Chip is Busy

Flash Read (Read Only) – (PCIBAR2 + 14H)

A Flash Read command is executed by reading this register at base address plus 14H. 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 2CH, 30H, 34H, and 38H.

The system must issue the Flash Reset command to re-enable the device for reading array data if DQ5 goes high. DQ5 can go high during a Flash Start Write, Flash Erase Chip, or Flash Erase Sector operation. DQ5 can be monitored via the Flash Status 1 register at base address plus 0CH.

Flash Reset (Write Only) – (PCIBAR2 + 18H)

This write only register is used to initiate a reset of the flash chip. A Flash Reset command is executed by writing logic 1 to bit-0 of this register at base address plus 18H. Writing the flash reset command resets the chip to reading data mode. Flash reset can be useful when busy is held active.

Flash Start Write (Write Only) – (PCIBAR2 + 1CH)

This write only register is used to initiate the write of a byte 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 1CH. 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 28H, 2CH, 30H, 34H. and 38H.

Issuing of 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 of the next Flash Start Write if consecutive addresses are to be written.

Flash Erase Sector (Write Only) – (PCIBAR2 + 20H)

FLASH CONTROL REGISTERS

A Flash Erase Sector command is executed by writing logic 1 to bit-0 of this register at base address plus 20H. Verify that the Flash Chip is not busy from a previous operation before beginning a new operation. This is accomplished by reading bit-7, of the Flash Status 2 register, as logic 0.

There are 256 flash sectors, which are addressed via the most significant eight flash address lines. The most significant eight flash address lines are set via the Flash Address 24-17 register at base address plus 34H and 38H. Issuing of a Flash Erase Sector command will erase the contents of the flash chip only in the sector specified.

A flash bit cannot be programmed from logic 0 to logic 1. Only an erase chip operation can convert logic 0 back to logic 1. **Prior to reprogramming of the flash chip a Flash Erase Chip or Flash Erase Sector command must be performed.**

The system can determine the status of the erase operation by reading the Flash Ready/Busy status. Bit-7 of the Flash Status 2 register, at base address plus 10H, will read as logic 0 when chip erase is completed.

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

Flash Erase Chip (Write Only) – (PCIBAR2 + 24H)

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

A Flash Erase Chip command is executed by writing logic 1 to bit-0 of this register at base address plus 24H. Verify that the Flash Chip is not busy from a previous operation before beginning a new operation. This is accomplished by reading bit-7, of the Flash Status 2 register, as logic 0. Also, it may be necessary to issue a flash logic state machine reset prior to issue of the Flash Erase Chip command. This is accomplished by write of 0x8000 hex to address PCIBAR2 + 4.

The system can determine the status of the erase operation by reading the Flash Ready/Busy status. Bit-7 of the Flash Status 2 register, at base address plus 10H, will read as logic 0 when chip erase is completed.

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

Flash Data Register (Read/Write) – (PCIBAR2 + 28H)

This read/write register holds the data byte which is sent to the flash chip upon issuing of a Flash Start Write command.

Although only the least significant 8 bits of this register are used, reading or writing this register is possible via 32-bit, 16-bit or 8-bit data transfers.

FLASH REGISTERS

Flash Address 7->0 (Read/Write) – (PCIBAR2 + 2CH)

This read/write register holds the least significant byte of the address to which the flash chip is written upon issue of a Flash Start Write command.

Although only the least significant 8 bits of this register are used, reading or writing this register is possible via 32-bit, 16-bit or 8-bit data transfers.

Flash Address 15->8 (Read/Write) – (PCIBAR2 + 30H)

This read/write register sets bits 15 to 8 of the address to which the flash chip is written upon issue of a Flash Start Write command.

Although only the least significant 8 bits of this register are used, reading or writing this register is possible via 32-bit, 16-bit or 8-bit data transfers.

Flash Address 23->16 (Read/Write) – (PCIBAR2 + 34H)

This read/write register sets bits 23 to 16 of the address to which the flash chip is written upon issue of a Flash Start Write command.

Reading or writing this register is possible via 32-bit, 16-bit or 8-bit data transfers.

Flash Address 24 (Read/Write) – (PCIBAR2 + 38H)

This read/write register sets bit 24 of the address to which the flash chip is written upon issue of a Flash Start Write command.

Reading or writing this register is possible via 32-bit, 16-bit or 8-bit data transfers.

System Monitor Status/Control Register (Read/Write) – (PCIBAR2 + 3CH)

SYSTEM MONITOR REGISTERS U5 PCI bus

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 PCIBAR2 plus 40H. Next, this register at PCIBAR2 plus 3CH is read. Bits 22 to 16 of this register hold the address of system monitor register that is accessed. Data bits 15 to 6 of this register hold the temperature, Vccint, or Vccaux value. Data bits 5 to 0 are not used and bits 15 to 6 need to be right justified to properly represent the 10-bit digital output value. Valid addresses are given in column one of Table 3.13.

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) – (PCIBAR2 + 40H)

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 UG192 (available from Xilinx). Writing this register is possible via 32-bit transfers.

The address value written to this register can be read on bits 22 to 16 of the System Monitor Status/Control register at PCIBAR2 plus 3CH.

Table 3.13: *System Monitor Register Map*

Address	Status Register
00h	Temperature
01h	Vccint
02h	Vccaux
20h	Maximum Temperature
21h	Maximum Vccint
22h	Maximum Vccaux
24h	Minimum Temperature
25h	Minimum Vccint
26h	Minimum Vccaux

PCIBAR2 U7 FPGA REGISTERS

Software Reset and Status Register (Read/Write) – (PCIBAR2 + 8000H)

This read/write register is used to Software reset the board, monitor and clear pending interrupts, select the on board active clock, and identify an attached mezzanine module.

Bits 0 to 7 of this register are used to monitor the interrupt pending status of interrupts originating from the front mezzanine module. Writing a logic "1" to a bit will clear the corresponding channel's pending interrupt status.

USERo CLOCK CONTROL

Note USERo selects the Local bus clock.

Bit 8 of this register controls the USERo signal. The USERo control signal is used to select between the 133MHz clock and the user defined clock (PLL_CLK). The user defined clock is defined in the example code of the FPGA and output on signal PLL_CLK. The Digital Clock Manager of the FPGA offers a wide range of clock management features including clock multiplication and division for generation of a user defined clock (PLL_CLK). A 133MHz crystal generated clock signal (FPGA_CLK_PLL) is input to the FPGA for use in generation of the user-defined clock signal PLL_CLK. The PLL_CLK can be a minimum of 10MHz and a maximum of 133MHz. Since the PLL_CLK signal is generated and driven by the FPGA, it will only be available after the FPGA is configured. See the example VHDL file included in the engineering design kit and the Xilinx documentation on the Digital Clock Manager for more information.

The USERo signal is controlled via a bit-8 of the Software Reset and Status Register at PCIBAR2 plus 8000H. The USERo control bit-8 by default is set to a logic low to select the PLL_CLK clock as the board clock frequency. Bit-8 set to logic high will select the 133MHz clock as the board clock frequency.

Bits 15 to 13 of this register will read "001" for all Acromag digital I/O mezzanine modules. These bits will read "010" when the AXM-A30 high speed analog input mezzanine module is present.

Bits 27 and 28 are DMA acknowledgement bits and will read a logic high while the corresponding DMA channel transfer is active.

Bit 31 of this register when set to a logic "1" will issue a reset signal to the FPGA hardware.

Reading or writing to this register is possible via 32-bit, 16-bit or 8-bit data transfers.

Table 3.14: *Software Reset and Status Register*

BIT	FUNCTION	
7-0	Mezzanine interrupt status is identified via data bits 0 to 7. Read of a “1” indicates that an interrupt is pending for the corresponding data bit. A pending interrupt will remain active until disabled via the mezzanine interrupt control registers and cleared by writing a logic “1” to the corresponding bit of this register.	
	Logic “0”	Interrupt Not Pending
	Logic “1”	Interrupt Pending/Clear
8	USERo Control	
	Logic “0”	Board clock = PLL_CLK (Default)
	Logic “1”	Board clock 133MHz
12-9[1]	Not Used[1]	
15-13	Mezzanine Identification Code: “001” for all Acromag digital I/O mezzanine boards “010” for the AXM-A30 mezzanine board	
26-16	Not Used[1]	
27	DACK0 Status Logic high is a valid acknowledgement for DMA channel 0	
28	DACK1 Status Logic high is a valid acknowledgement for DMA channel 1	
29	Not Used[1]	
30	Not Used[1]	
Bit-31	The most significant bit of this register when set to a logic “1” will issue a software reset.	
	Logic “0”	No Operation
	Logic “1”	Software reset issued to Xilinx user-programmable FPGA

1. All bits labeled “Not Used” will return logic “0” when read.

REAR Read REGISTER

Rear I/O Connector Read Register (Read Only) - (PCIBAR2 + 802CH)

The Rear I/O Connector Read Register is used to read the LVDS input status of 16 channels. This example design has 16 rear I/O channels, identified in Table 3.15, programmed as LVDS input only channels. Table 2.1 shows each channel and it’s corresponding P4 connector pin assignment.

This Rear I/O Connector Read register is a read only register and writing to this register has no effect on the LVDS input channels. Reading from this register is possible via 32-bit, 16-bit or 8-bit data transfers.

REAR Write REGISTER

Rear I/O Connector Write Register (Read/Write) - (PCIBAR2 + 8030H)

The Rear I/O Connector Write Register is used to set 16 LVDS output channels. This example design has 16 channels, identified in Table 3.15, fixed as LVDS output only channels. Table 2.1 shows the P4 connector pins and their corresponding channel identifiers.

This Rear I/O Connector Write register is written to set the LVDS output channels and can also be read to verify the output channel settings. Reading from this register is possible via 32-bit, 16-bit or 8-bit data transfers.

Table 3.15: *Rear I/O Registers*

Column 1 identifies the write data bit that drives the output channel listed in column 2. Column 1 also identifies the read data bit that returns the input channel listed in column 3. For example data bit 0 drives output channel 1 when written and returns channel 0 register setting when read.

1. *All bits labeled “Not Used” will return logic “0” when read.*

Write/Read Data Register Bit	Rear Connector Write Output Channels	Rear Connector Read Input Channels
0	1	0
1	2	3
2	4	5
3	6	7
4	8	9
5	10	11
6	12	13
7	14	15
8	16	17
9	18	19
10	20	21
11	22	23
12	24	25
13	26	27
14	28	29
15	30	31
16-31	Not Used[1]	Not Used[1]

DMA REGISTERS AT PCIBAR2

DMA Control Register (Read/Write) - (PCIBAR2 + 8034H)

The DMA Control Register is used to request a DMA Demand mode transfer. The hardware signals **DREQ0 and DREQ1** are driven active by software setting of bits 0 or 1 of this register to request the DMA transfer. The transfer must include the Static RAM Memory as either the source or the destination.

For software triggered DMA, bit-0 is used to request a DMA channel 0 transfer while bit-1 is used to request a channel 1 transfer. The bit must be set to logic high to request a transfer. Once set, the bit will remain asserted until the DMA transfer has completed. If both bits are set simultaneously, the channel 0 DMA transfer will be implemented first followed by channel 1.

In user applications a data ready condition, such as a memory buffer full condition, can be physically tied (via logic in the FPGA) to the DREQ0 or DREQ1 FPGA signals to cause the DMA transfer to start.

DUAL PORT SRAM REGISTERS

FPGA- SRAM Data Register (Read/Write) – (PCIBAR2 + 8038H and 803CH)

The FPGA-SRAM Data Read Register is provided to access the SRAM port that links directly to the user-programmable Virtex-5 FPGA. Reading or writing PCIBAR2 + 8038H accesses the SRAM least significant data lines 31 to 0. Reading or writing PCIBAR2+ 803CH accesses the most significant SRAM data lines 63 to 32. Reading or writing these registers is only possible in 32-bit transfers. The address for the SRAM read or write is initialized by the Dual Port SRAM Internal Address register at PCIBAR2 + 8044H. With each additional read or write to PCIBAR2+ 803CH the address is automatically incremented.

Writing the SRAM would proceed by first setting the Address register at PCIBAR2 + 8044H. Next the least significant 32-bit data word is written to PCIBAR2 + 8038H. Finally, after the most significant 32-bit data word is written at PCIBAR2+ 803CH the address is automatically incremented.

FPGA-Port SRAM Control Register (Read/Write) – (PCIBAR2 + 8040H)

This read/write register is used to control the Dual-Port SRAM including enabling write, automatic DMA transfer and automatic address reset on DMA thresholds.

The default power-up state of this register is logic low. A reset will set all bits in this register to "0". Reading or writing to this register is possible via 32-bit, 16-bit or 8-bit data transfers.

***Table 3.16:** FPGA-Port SRAM Control Register*

1. *Bits are not used and will return logic "0" when read.*
2. *All DMA transfer settings in the DMA Registers at PCIBAR0 should be set prior to enabling automatic DMA transfers.*
3. *WARNING: Before enabling Address Reset on DMA Thresholds (bits 3 & 4), verify that the "DMA Ch. 0 Threshold Register" is not equal to the "Address Reset Register 0" and the "DMA Ch. 1 Threshold Register" is not equal to the "Address Reset Register 1." If these registers are equal and automatic reset is enabled an infinite loop will be created within the internal logic of the FPGA.*

BIT	FUNCTION	
0	This bit controls the vhdl signal SRAM_ENABLE. This signal must be set to logic high to enable writes to SRAM from the FPGA. The SRAM Internal Address register must also be set with the start address at which the data begins filling the SRAM.	
	Logic "0"	Disable Write and Enable Read
	Logic "1"	Enable Write and Disable Read
1	If enabled via this bit a DMA channel 0 request will be issued when the internal address counter is equal to the DMA Channel 0 Threshold Register. This will have the same effect as writing a 1 to bit 0 of the DMA Control Register at PCIBAR2 plus 8034H. See Synchronous DP-SRAM in Section 4.0 for further details on using this feature.[2]	
	Logic "0"	Disable Auto DMA Request Channel 0
	Logic "1"	Enable Auto DMA Request Channel 0
2	If enabled via this bit a DMA Channel 1 request will be issued when the internal address counter is equal to the DMA Channel 1 Threshold Register. This will have the same effect as writing a 1 to bit 1 of the DMA Control Register at PCIBAR2 plus 8034H.[2]	
	Logic "0"	Disable Auto DMA Request Channel 1
	Logic "1"	Enable Auto DMA Request Channel 1
3	If enabled via this bit the Internal Address Counter will be loaded with the value in Address Reset Register 0 when the counter is equal to the DMA Channel 0 Threshold Register. See the Address Reset Register description for further details. DMA does not have to be enabled to use this feature.[3]	
	Logic "0"	Disable Add. Reset on DMA Ch. 0 Threshold
	Logic "1"	Enable Add. Reset on DMA Ch. 0 Threshold
4	If enabled via this bit the Internal Address Counter will be loaded with the value in Address Reset Register 1 when the counter is equal to the DMA Channel 1 Threshold Register. See the Address Reset Register description for further details. DMA does not have to be enabled to use this feature.[3]	
	Logic "0"	Disable Add. Reset on DMA Ch. 1 Threshold
	Logic "1"	Enable Add. Reset on DMA Ch. 1 Threshold
5-15	Not Used[1]	

DP-SRAM REGISTERS

FPGA-Port SRAM Internal Address Register (Read/Write) – (PCIBAR2 + 8044H)

The FPGA-Port SRAM Internal Address Register is used to view and set the internal SRAM address. The FPGA will only write using 64-bit data transfers allowing for 3FFFFH (2^{18}-1) unique memory accesses. **Reading** bits 0 to 17 of this register will return the internal SRAM address. *Due to delays during data processing and the PCI transfer the actual internal address may be slightly greater than the address read.* **Writing** to bits 0 through 17 of this register will set the Internal SRAM Address to the provided value. Writing logic '1' to bit 31 of this register or a system reset will cause the Internal SRAM Address to reset to "00000H" (the start of the SRAM memory). **Reading or writing to this register is possible via 32-bit data transfers, only.**

Warning: To guarantee functionality disable DP-SRAM write cycles (via bit 0 of the DP-SRAM Control Registers) before writing to the DP-SRAM Internal Address Register.

The SRAM Internal Address will automatically be incremented upon a write or read of the most significant SRAM Data Port at PCIBAR2+ 803CH.

Table 3.17: *FPGA-Port SRAM Internal Address Register*

FPGA-Port SRAM Internal Address Register		
D31	**D30-D18**	**D17-D0**
SRAM Internal Address Reset	Not Used (Read as logic '0')	SRAM Internal Address

FPGA-Port SRAM DMA Channel 0/1 Threshold Registers (Read/Write) – (PCIBAR2 + 8048H/ 804CH)

The FPGA-Port SRAM DMA Channel 0/1 Threshold Registers are used to initiate an automatic DMA transfer. When the internal address counter is equal to the value in the DMA Channel 0 Threshold Register, a Channel 0 DMA request will be initiated. Similarly, when the internal address counter is equal the value in the DMA Channel 1 Threshold Register and there is valid data at that address, a Channel 1 DMA request will be initiated. This feature must be enabled via bits 1 and 2 (for Channels 0 & 1, respectively) of the FPGA-SRAM Control Register. Note that DMA settings must be set prior to the initiated transfer on both the PCIBAR0 and PCIBAR2 registers. A DMA transfer in progress is indicated via bits 0 and 1, for DMA Channels 0 and 1, respectively, in the DMA Control Register. See the DMA Registers section of this manual for further details. **Reading** bits 0 to 17 of either register will return the corresponding DMA Threshold. **Writing** to bits 0 through 17 of this register will set the corresponding DMA Threshold to the provided value. **Reading or writing to this register is possible via 32-bit data transfers only.**

Note: An SRAM DMA Request will occur only after a data write cycle to the address defined by the DMA Threshold Registers.

Table 3.18: *Dual-Port DMA Threshold Registers*

FPGA-Port SRAM DMA Channel 0/1 Threshold Registers			
Register	**D31-D18**	**D17-D0**	**Reset Value**
DMA Channel 0 Threshold Reg.	Not Used (Read as logic '0')	Channel 0 Threshold Address	1FFFFH
DMA Channel 1 Threshold Reg.	Not Used (Read as logic '0')	Channel 1 Threshold Address	3FFFFH

DP-SRAM REGISTERS

WARNING: The "DMA Ch. 0 Threshold Register" must not equal the "Address Reset Register 0" and the "DMA Ch. 1 Threshold Register" must not equal the "Address Reset Register 1." If these registers are equal and the address reset is enabled via the FPGA Port-SRAM Control Register an infinite loop will be created within the internal logic of the FPGA.

FPGA-Port SRAM Address Reset Registers 0/1 (Read/Write) – (PCIBAR2 + 8050H/ 8054H)

The FPGA-Port SRAM Address Reset Registers are used to reset the internal address counter to a user-defined value immediately upon reaching the DMA Threshold value. For example, after an SRAM write cycle where the internal address counter is equal to the value defined in the DMA Channel 0 Threshold Register, the internal address counter will then be loaded with the value defined in the Address Reset Register 0. Similarly, after a SRAM write cycle where the internal address counter is equal to the value defined in the DMA Channel 1 Threshold Register, the internal address counter will then be loaded with the value defined in the Address Reset Register 1. This allows for the internal address counter to be changed without any interruption in the transfer of data from the front connector input to the DP-SRAM. This feature must be enabled via bits 3 and 4 (for Channel 0 & 1 thresholds, respectively) of the FPGA Port-SRAM Control Register. Note that the DMA transfers *do not* have to be enabled for this feature to function. **Reading** bits 0 to 17 of either register will return the corresponding internal address reset value. **Writing** to bits 0 through 17 of this register will set the corresponding internal address reset register to the provided value. Bits 18 to 31 are not used and will return logic '0' when read. A system reset will cause these registers to reset to "00000H".

ID Code REGISTER

PMC Board Identification Code Register (Read Only) - (PCIBAR2 + 8058H)

The PMC Board Identification Code register at PCIBAR2 plus 8058H stores an ID code that can used to uniquely identify the PMC Virtex 5 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 PMC boards that may have the same device and vender ID in a given system.

Reading from this register is possible via 32-bit, 16-bit or 8-bit data transfers.

DDR-SDRAM REGISTERS

DDR SDRAM Control/Status Register (Read/Write) –(PCIBAR2 + 805CH)

The DDR SDRAM and Block RAM are directly interfaced to the PowerPC via the processor local bus. An interface from the PCI bus to the DDR SDRAM and Block RAM is implemented with a master to Slave PLB interface. This interface from the PCI bus allows access the DDR SDRAM and Block RAM by way of the PowerPC crossbar switch. The DDR SDRAM memory space range is from 0X0 to 0X0FFFFFFF. The Block RAM memory space range is from 0XFFFF0000 to 0XFFFFFFFF. Access to the DDR SDRAM and Block RAM address space from the PCI bus is performed with use of the DDR SDRAM Control/Status, DDR SDRAM Address, DDR SDRAM Read, DDR SDRAM Write, and DDR SDRAM Mask Registers.

Table 3.19: *DDR-SDRAM Control/Statusl Register*

1. *All bits labeled "Not Used" will return logic "0" when read.*

BIT	FUNCTION	
0	Start DDR-SDRAM write operation. The DDR-SDRAM Write and Mask registers must first be written with the desired data that are to be burst out to the DDR-SDRAM. In addition, the DDR-SDRAM Address register must be written with the write address prior to setting this bit.	
	Logic "0"	No operation performed
	Logic "1"	Write Transfer Performed
1	Start DDR-SDRAM read operation. The Address Register must be written with the start address location prior to setting this bit. The DDR-SDRAM Read register contains valid read data.	
	Logic "0"	No operation performed
	Logic "1"	Read Transfer Performed
2	Logic "1" = Read Complete Valid read data is available in the DDR-SDRAM Read register. This bit is cleared at the start of the next DDR-SDRAM read.	
3	Logic "1" = Write Complete Write of DDR-SDRAM has be initiated. This bit is clear at the start of the next DDR-SDRAM write.	
4	Logic "1" = Read Error This bit is clear at the start of the next DDR-SDRAM read.	
5	Logic "1" = Write Error This bit is clear at the start of the next DDR-SDRAM write.	
6	Logic "1" = Address Phase Time Out This bit is cleared at the start of the next DDR-SDRAM read or write.	
7-31	Not Used[1]	

This read/write register is used to control read or write to DDR SDRAM. A DDR-SDRAM write operation will require the DDR-SDRAM Write register to be preloaded with a 32-bit data values prior to issuing the write operation by setting bit-0 of this register to logic high. The DDR-SDRAM Read register will contain a 32-bit data value following the issue of a read operation by setting bit-1 of this register to logic high. Prior to issue of a read or write with this register, the DDR-SDRAM Address register must first be written with the desired address location for the access.

The valid read data, read error, write error, and address phase time out bits when set will stay set until the start of the next DDR-SDRAM read or write operation. A read or write operation is started by setting bits 0 or 1 of this register.

DDR SDRAM Address Register (Read/Write) –(PCIBAR2 + 8060H)

This read/write register is used to set the address at which the DDR-SDRAM is read or written. This address must be a valid DDR-SDRAM address in the PowerPC address space. The DDR SDRAM memory space range is from 0X0 to 0X0FFFFFFF. The Block RAM memory space range is from 0XFFFF0000 to 0XFFFFFFFF.

This address should be incremented by 4 for each consecutive 32-bit access.

Table 3.20: *DDR-SDRAM Addres Register*

1. *All bits labeled "Not Used" will return logic "0" when read.*

BIT	FUNCTION
A31-A0	DDR-SDRAM address is written to these bits.

DDR SDRAM Read Register (Read Only) – (PCIBAR2 + 8064H)

The DDR-SDRAM Read register is read only and holds the last data value read from the DDR-SDRAM.

A DDR-SDRAM read is implemented by executing the following steps.

a) Set the DDR-SDRAM Address register at PCIBAR2 + 8060H with the DDR-SDRAM address location.
b) Set the start read bit of the DDR-SDRAM Control register at PCIBAR2 + 805C. Set bit-1 of the SDRAM Control Register at 805CH to logic high.

The data is read from the SDRAM and moved to the SDRAM Read Register at 8064H. Data is valid in the read register only after bit-2 of the DDR-SDRAM Control/Status register at PCIBAR2 + 805C is set to logic high.

Table 3.21: *DDR-SDRAM Read Registers*

DDR SDRAM Read Registers		
Base Addr+	**D31-D0**	**Base Addr+**
8067	DDR-SDRAM Read Register D0	**8064**

DDR SDRAM Write Register (Write Only) – (PCIBAR2 + 8068H)

The DDR-SDRAM Write register holds the data value that is to be written to the DDR-SDRAM.

A DDR-SDRAM write is implemented by executing the following steps.

1) Write the 32-bit data value that is to be written to the DDR-SDRAM to the registers at base address + 8068H.
2) Set the DDR-SDRAM Mask bits as desired at base address + 806CH. A value of 0H would enable all bytes to be written.
3) Set the DDR-SDRAM Address register at PCIBAR2 + 8060H with the DDR-SDRAM write address location.
4) Issue the Write Command. Set bit-0 of the SDRAM Control/Status Register at PCIBAR2 + 805CH to logic high.

DDR SDRAM Mask Register (Read/Write) – (PCIBAR2 + 806CH)

The DDR-SDRAM mask register holds the write mask data bits that accompany the write data as it is written to the DDR-SDRAM. If a given data mask (DM) bit is set low, the corresponding data will be written to memory; if the DM bit is set high, the corresponding data will be ignored, and a write will not be executed to that byte location.

Table 3.22: *DDR-SDRAM Mask Register*

DDR SDRAM Mask Register Bit	Write Register	Byte Masked
0	D0	Byte 0 (D0 to D7)
1		Byte 1 (D8 to D15)
2		Byte 2 (D16 to D23)
3		Byte 3 (D24 to D31)
4-31	Not Used[1]	

2. All bits labeled "Not Used" will return logic "0" when read.

Read or writing this register is possible via 32, 16 or 8-bit transfers.

PowerPC Read Register (Read Only) – (PCIBAR2 + 8070H)

The PowerPC Read register stores/holds the value written by the PowerPC to this register. The PowerPC writes this register with a write access to PowerPC address 0xFFFE8000. PCI bus read of this register is possible via 32, 16, or 8-bit reads.

This register serves as an example of how the PowerPC could communicate with a controller connected to the PCI bus.

SYSTEM MONITOR REGISTERS U7 FPGA

System Monitor Status/Control Register (Read/Write) – (PCIBAR2 + 8088H)

This read/write register will access the system monitor register at the address set in the System Monitor Address Register at PCIBAR2 plus 808CH.

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 PCIBAR2 plus 808CH. Next, this register at PCIBAR2 plus 8088H 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 temperature, Vccint, or Vccaux value. Data bits 5 to 0 are not used. Valid addresses are given in column one of Table 3.23.

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) – (PCIBAR2 + 808CH)

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 UG192 (available from Xilinx). Reading or 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 PCIBAR2 plus 8088hex.

Table 3.23: *System Monitor Register Map*

Address	Status Register
00h	Temperature
01h	Vccint
02h	Vccaux
20h	Maximum Temperature
21h	Maximum Vccint
22h	Maximum Vccaux
24h	Minimum Temperature
25h	Minimum Vccint
26h	Minimum Vccaux

PCIBAR1 MEMORY MAP

DUAL PORT MEMORY

A 256K x 64-bit Dual Port synchronous SRAM (DP-SRAM) is provided on the VFX board. One port of the SRAM connects directly to the local bus to allow for PCI bus access. The remaining port connects directly with the user-programmable FPGA. This design allows for the user to maximize data throughput between the Field I/O's and the controlling processor.

There are two automatic DMA initiators available that will trigger upon a user set threshold. Furthermore, upon a DMA transfer, the internal counter can be reset to a user specified value. See DMA Registers for more information on these operations. These features can be individually controlled through the SRAM Control Registers.

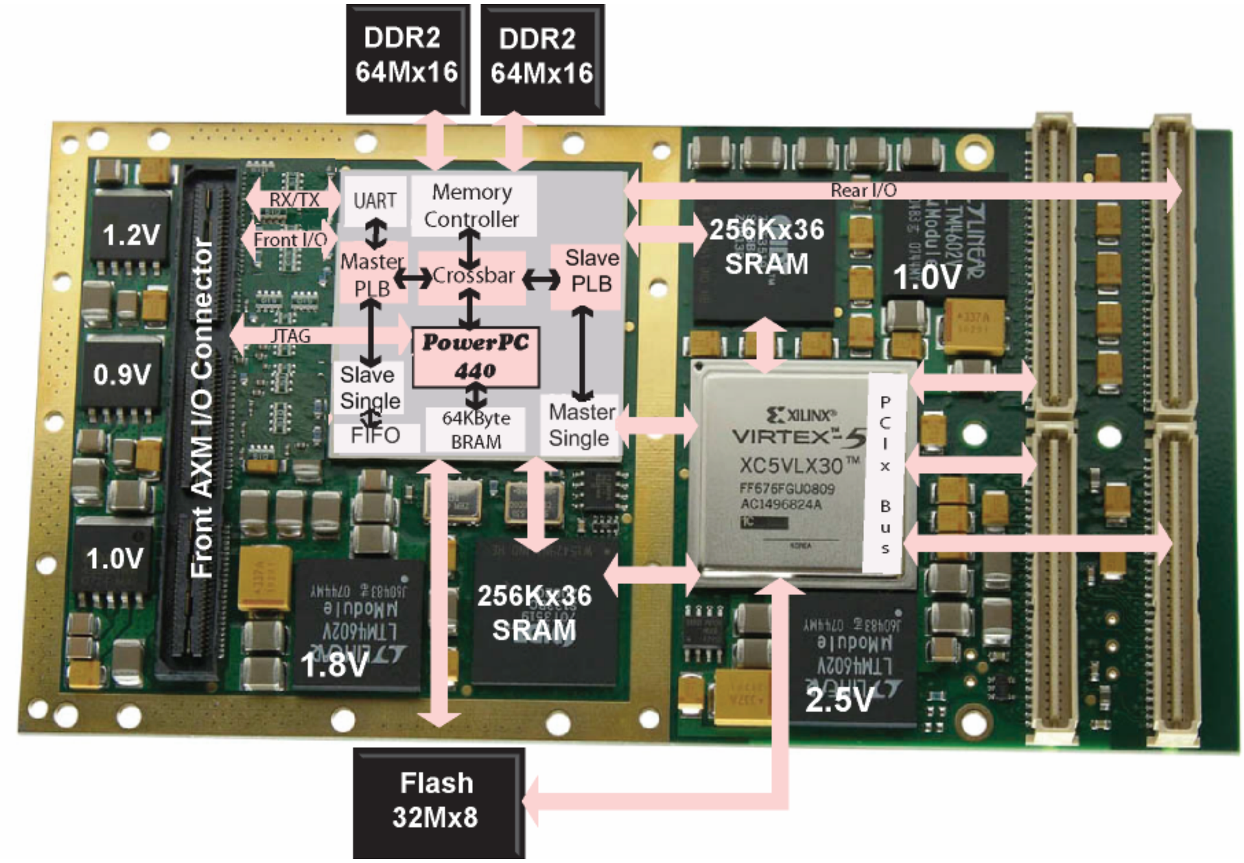
Static RAM Memory (Read/Write) – (PCIBAR1 + 0H to 1FFFFFH)

The Static RAM memory space is used to provide read or write access to on board SRAM memory. This memory space allows access to the SRAM from the port on the PCI bus side of the SRAM. The Static RAM device has a 256K x 64-bit memory configuration (two 256K x 32-bit devices). Reading or writing to this memory space using DMA access is also only possible in 32-bit or 64-bit transfers.

The FPGA-Port SRAM Register at PCIBAR2 + 8038H and 803CH are provided for testing the SRAM port that links directly to the user-programmable Virtex-5 FPGA.

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.



PCI INTERFACE LOGIC

The PCIx bus interface logic on this board provides a 100Mhz, 66MHz, or 33MHz 64/32-bit interface to the carrier/CPU board per PCI Local Bus Specification 3.0. The interface to the carrier/CPU board allows complete control of all board functions.

Note that the VFX board requires that system 3.3 volts be present on the PCI bus 3.3V pins. There are some older systems that do not provide 3.3 Volts on the PCI bus 3.3 volt pins. The VFX boards will not work in those systems.

THEORY OF OPERATION CONTINUED

This is a master/target board, with the PCI bus interface logic contained within the board. This logic includes support for PCI commands, including: configuration read/write, and memory read/write. In addition, the PCI interface performs target abort, retry, and disconnect. The logic also implements interrupt requests via interrupt line INTA#.

The board becomes the PCI bus master to perform DMA transfers on channels 0 and 1. The DMA transfers can be started via software or hardware. Hardware signal DREQ0# driven active by the programmable FPGA will start a DMA channel 0 transfer. Hardware signal DREQ1# driven active will start a DMA channel 1 transfer. To identify the pins corresponding to these signals, see the user constraints (.UCF) file provided in the engineering design kit. The DACK0# and DACK1# signals will go active upon the start of a DMA transfer and remain active until its completion. The example device driver software (purchased separately) can be used to exercise DMA block (software) and demand (hardware) modes of operation.

SYNCHRONOUS Dual-Port SRAM

A 256K x 64-bit synchronous Dual-Port SRAM is provided on the board. One port of the SRAM interfaces to the PCI bus interface chip, the Xilinx Virtex-5 LX30 device (U5). The other port connects directly to the programmable FPGA (U7). This configuration allows for a continuous data flow from the field inputs through the FPGA to the SRAM and then to the PCI bus. Both ports of the SRAM operate in Pipeline mode. This allows for faster operational speed but does cause a one-cycle delay during read operations. The pins corresponding to the control signals, address, and data buses are in the user constraint (.UCF) file provided in the engineering design kit.

The SRAM port connected directly to the user-programmable FPGA (U7) supports continuous writes or single cycle reads. The SRAM port connected to the PCI bus, through U5, supports reading and writing using a continuous, single cycle, or DMA transfers. For single cycle accesses address and control signals are applied to the SRAM during one clock cycle, and either a write will occur on the next cycle or a read in two clock cycles. DMA accesses operate using the continuous burst method for maximum data throughput. The control signal, starting address, and data (if writing) are applied to the SRAM during one clock cycle. Then, during a write DMA transfer, new data is applied to the bus every subsequent clock cycle until the transfer is complete.

During DMA transfers the address is held constant and incremented internally in the Dual-Port SRAM. Please refer to the IDT70T3519S133BC Data Sheet (See Related Publications) for more detailed information.

DDR2 SDRAM

The board contains two 64M x 16-bit DDR2 SDRAM devices. The DDR2 SDRAM uses a double data rate architecture to achieve high-speed operation. The double data rate architecture has an interface designed to transfer two 32-bit data words per clock cycle.

DQS is edge-aligned with data for reading data and center-aligned with data for writing data.

PLBV46 Master Single Core

The DDR2 SDRAM can be accessed directly from the PowerPC or indirectly from the PCI bus. The DDR2 Memory is resident in the PowerPC memory space at address 0x0 to 0x0FFFFFFF. This DDR2 memory can be accessed in this address space directly from the PowerPC or from the PCI bus using the **DDR SDRAM Write Register or DDR SDRAM Read Register** PCI at base PCI bus Bar2 base address plus 0x8068 and 0x8064, respectively. Access to the DDR SDRAM and Block RAM address space from the PCI bus is performed by way of the PLBV46 Master Single core.

The following figure shows how a master device like the PCI bus can access the DRR2-SDRAM via the crossbar's slave processor local bus (SPLB) interface. This interface allows 32-bit read or write accesses to the DDR2-SDRAM from the PCI bus. The PCI bus accesses are performed through the XC5VLX30 (U5) FPGA.

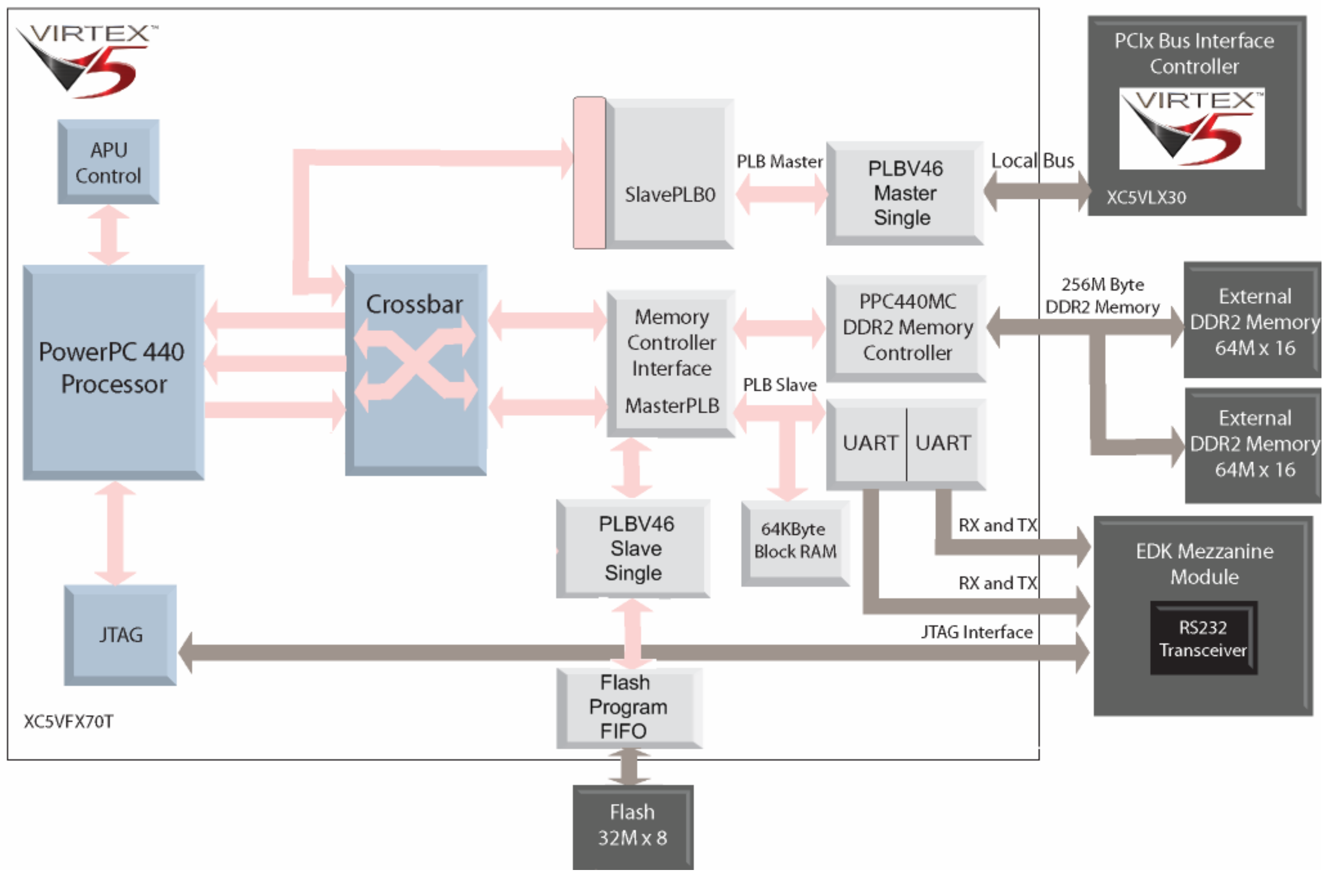
PLBV46 Slave Single Core

A user interface to the PLBV46 Slave Single core provides PowerPC read access of the Flash Program Code FIFO. The XC5VFX70T VHDL code executes the move of the PowerPC program code/data from flash memory starting at address sector 128 to the Flash Program Code FIFO.

The PLBV46 Slave Single core also handles PowerPC write access to the PowerPC_Read_Reg. The PowerPC_Read_Reg can be read by the PCI bus at PCI bus BAR2 address plus 0x8070.

64K Byte Block RAM

The PowerPC uses a 64K byte Block RAM memory space for storage and execution of a bootloop program that executes upon power-up or reconfiguration. The Block RAM is resident in the PowerPC memory space at address 0xFFFF0000 to 0xFFFFFFFF.



JTAG Interface

The JTAG interface to the PowerPC is used to download programs and run them in debug mode one instruction at a time. The JTAG cable, available from Xilinx, plugs into your USB or parallel port of your PC at one end. At the other end the JTAG flat ribbon cable plugs into the Acromag AXM-VFX-EDK board. This allows communication with the PowerPC from the Xilinx Platform Studio software using a PC.

UART Interface

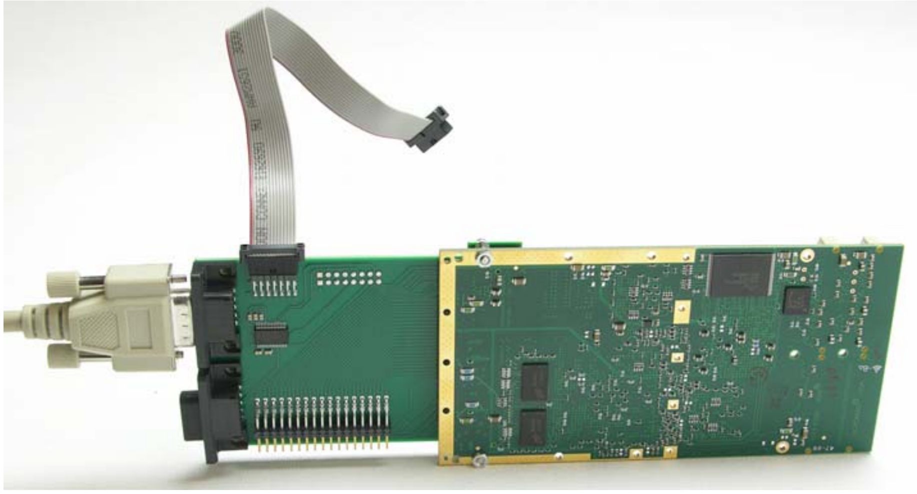
The PowerPC uses a MasterPLB interface to a UARTLITE soft-cores out to the AXM-VFX-EDK board for display of program data output. A second UART is also provided for additional program debug and development.

RS232 Null Modem Cabl

Connect your RS232 null modem cable to the top left connector of the EDK board and to your hyper-terminal com port. Set the Bit Per Seconds to 9600, Data Bits to 8, Parity to None, and one Stop Bit.

Xilinx JTAG Program Cable

Install the JTAG program cable as recommended by Xilinx. The flat 14 conductor ribbon cable must be connected to the JTAG port of the VFX-EDK board as shown in the following figure.



Local Bus Signals

The local bus interface between the PCI bus interface chip (U5) and the user-programmable FPGA (U7) consists of the following signals.

The Local Address bus (LA) bits 21 to 2 are used to decode the 4M byte address space allocated by the PCI bus to BAR2. Also, LA(26) bit-26 of the local address bus is logic high when the PCI bus is performing an access to BAR2 address space.

LBE0_n, LBE1_n, LBE2_n and LBE3_n are the local bus byte enables. LBE0_n when logic low indicates that the least significant byte on data lines D7 to D0 is selected for the read or write transfer. Likewise LBE3_n when logic low indicates that the most significant byte on data lines D31 to D24 is selected for the read or write transfer.

The Local Data (LD) bus bits 31 to 0 are bi-directional signals used for both read and write data transfers.

ADS_n, the address data strobe signal (driven by the PCI bus interface chip U5), will pulse low for one local bus clock cycle at the start of a new read or write access.

Readyn must be driven low by the programmable FPGA (U7) and held low until RdyAck_n is driven low by the PCI bus interface chip (U5). This is shown on the write and read diagrams that follow.

The LW_R_n signal, when logic high, indicates a write transfer in which data is moving from the PCI bus to the reprogrammable FPGA (U7). This signal, when logic low, indicates a read transfer in which data is moving from the reprogrammable FPGA (U7) to the PCI bus.

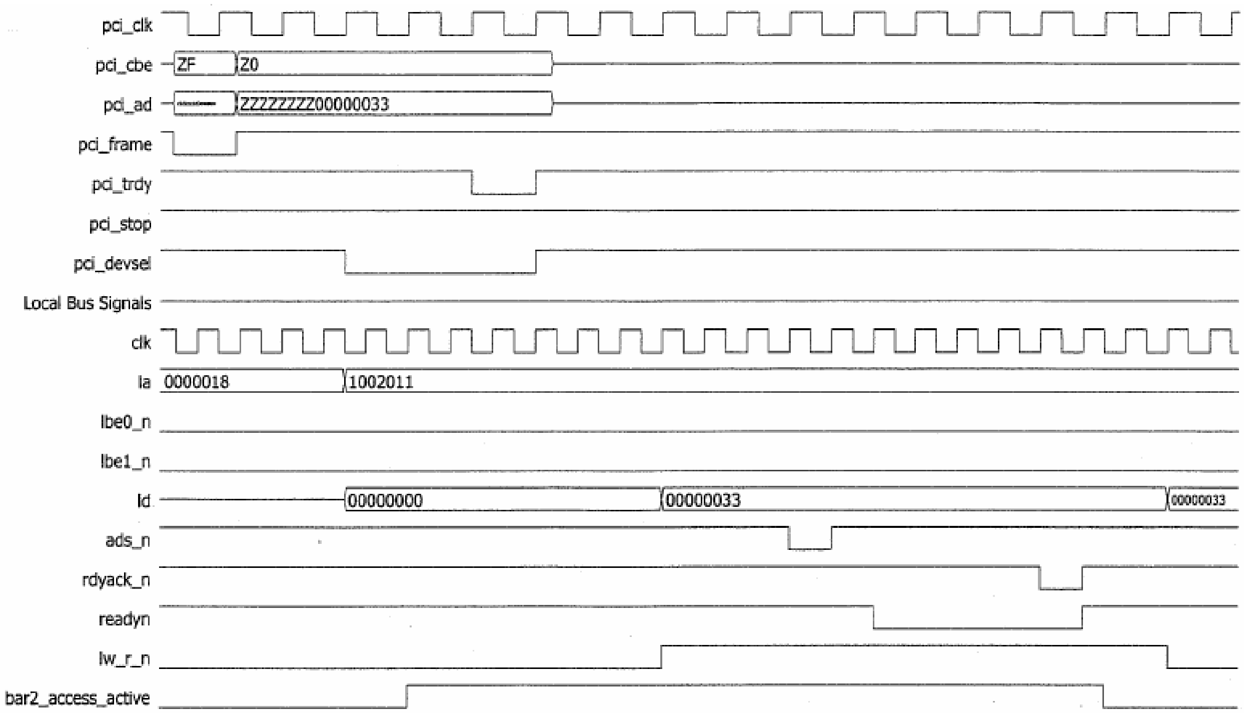
The BAR2_Access_Active signal shown in the write and read diagrams is local to the PCI bus FPGA (U5). This signal is used to properly respond to the PCI bus access.

Local Bus CLOCK CONTROL

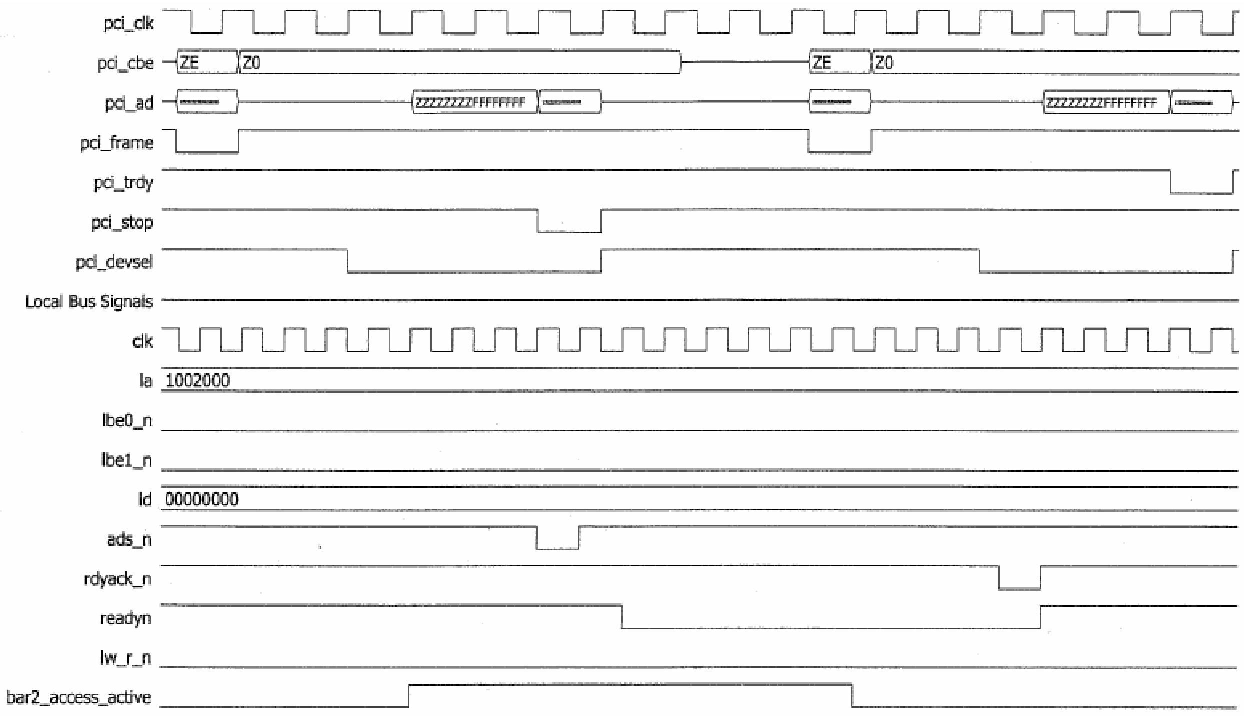
Clk, the local bus clock, as seen in the Local Bus Write and Read Cycle Diagram can be one of two sources. By default clk is a Digital Clock Manager (DCM) generated 53.2MHz frequency. Clk can also be selected to be the on board 133MHz frequency.

The Local bus clk signal is controlled by USERo. The board clock is routed to the Dual Port SRAM and user-programmable FPGA (U7) using a low skew clock driver (Cypress CY23EP05). The on board 133MHz crystal oscillator is input to the user-programmable FPGA via signal FPGA_CLK_PLL. After the user-programmable FPGA (U7) is configured, an FPGA DCM generated clock signal (PLL_CLK) is selected as the board clock (the default condition). By setting bit-8 of the Software Reset and Status register, at PCIBAR2 plus 8000H, to a logic high the 133MHz clock may be selected as the board clock. By setting bit-8 to a logic low the PLL_CLK becomes the board clock frequency. The default state of bit-8 is logic low.

Local Bus Write Cycle Diagram



Local Bus Read Cycle Diagram



5.0 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 fully tested.

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

Before beginning repair, be sure that all of the procedures in Section 2, Preparation For Use, have been followed. Also, refer to the documentation of your carrier/CPU 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.

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

WHERE TO GET HELP

www.acromag.com

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.

Choose the "Support" hyperlink in our website's top navigation row then select "Embedded Board Products Support" or go to http://www.acromag.com/subb_support.cfm to access:

- Application Notes
- Frequently Asked Questions (FAQ's)
- Knowledge Base
- Tutorials
- Software Updates/Drivers

An email question can be submitted from within the Knowledge Base or through the "Contact Us" hyperlink at the top of any web page.

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

6.0 SPECIFICATIONS

PHYSICAL

Single PMC Board

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

PMC-VFX: 3.59oz (0.1016Kg), typical

Connectors

- **PMC** PCI Local Bus Interface: Four 64-pin female receptacle header (AMP 120527-1 or equivalent). Three of these connectors interface to the PCI bus. The fourth connector provides 64 rear I/O connections.
- Front Field I/O Connector on PMC board (Samtec QSS-075-01-L-D-A); Mating Mezzanine Connector (Samtec QTS-075-01-L-D-A) with 5mm stack height or Samtec QTS-075-02-L-D-A with 8mm stack height.

Table 6.1: Power Requirements for Example Design. Power will vary dependent on the application.

5V Maximum rise time of 100m seconds

Power Requirements		**PMC Modules**
5V (±5%)	Typical	1600mA
	Max.	1920mA
3.3V (±5%)	Typical	700mA
	Max.	840mA
+/-12V (±5%)	Typical	0mA
	Max.	0mA

On Board 1.0V Power to Virtex-5 FPGA	**Current Rating** (Maximum available for the user-programmable FPGA)
1.0V (±5%)	4A Maximum

ENVIRONMENTAL

Operating Temperature: 0 to +70°C. –40°C to +85°C (E Version)

Conduction Cooled PCI mezzanine card: Complies with ANSI/VITA 20-2001 (R2005). The PMC VFX, without a faceplate, is fully compatible with a conduction cooled host card.

Relative Humidity: 5-95% Non-Condensing.

Storage Temperature: -55°C to 125°C.

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

SPECIFICATIONS

Radiated Field Immunity (RFI): Designed to comply with EN61000-4-3 (3V/m, 80 to 1000MHz AM & 900MHz. keyed) and European Norm EN50082-1 with no register upsets.

Conducted R F Immunity (CRFI): Designed to comply with EN61000-4-6 (3V/rms, 150KHz to 80MHz) and European Norm EN50082-1 with no register upsets.

Surge Immunity: Not required for signal I/O per European Norm EN50082-1.

Radiated Emissions: Meets or exceeds European Norm EN50081-1 for class B equipment. Shielded cable with I/O connections in shielded enclosure are required to meet compliance.

Reliability Prediction

Mean Time Between Failure: 636,634 hours (VFX70) @ 25°C, Using MIL-HDBK-217F, Notice 2.

User Programmable FPGA U7 (PMC-VFX70)

Xilinx XC5VFX70T-1FF1136

- 44,800 CLB Flip Flops
- 820,000 Distributed RAM Bits
- 296 18Kbit Block RAMs
- 128 DSP Slices
- 6 Clock Management Tiles

PowerPC 440 Core Example Design:

- sys_clk_pin = dcm_clk_s = 200MHz
- CPMC440CLK = proc_clk_s = 125MHz
- mi_mcclk_200 = dcm_clk_s = 200MHz
- CPMMCCLK = sys_clk_s = 125MHz

System Monitor Functionality:

- 0°C to +85°C minimum to maximum operating temperature limits (Commercial Version)
- -40°C to +100°C minimum to maximum operating temperature limits (E version)
- +125°C Absolute Maximum Junction Temperature
- 0.95V to 1.05 volt minimum to maximum Vccint voltage limits
- 2.375V to 2.625 volt minimum to maximum Vccaux voltage limits

REAR I/O

The rear I/O P4 PMC connector connects directly to banks 1, 4 and 21 of the U7 FPGA. The bank 1, 4 and 21 Vcco pins are powered by 2.5 volts and thus will support the 2.5 volt IOStandards. Table 6-39 of the Virtex 5 User Guide (available from Xilinx) lists all the supported IOStandards available. The example design defines the rear I/O with 2.5 volt LVDS.

- Maximum Recommeded Clock Rate…….150MHz (6.7ns clock period)
- Vcco Supply Voltage ……………..…...….2.5 volt
- V_{OH} Output High Voltage……………..…..…..1.602 volt
- V_{OL} Output Low Voltage…………………..….0.898 volt
- V_{ODIFF} Differential Output Voltage ……..…350m volt typical
- V_{OCM} Output Common Mode Voltage…...1.25 volt typical
- V_{IDIFF} Differential Input Voltage…………..100m volt minimum
- V_{ICM} Input Common Mode Voltage………0.3 volt min, 1.2 volt typical,
  2.2 volt max

FRONT I/O

See the mezzanine module users manual for front I/O specifications. This PMC 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

Write Disable Jumper

Write Disable Jumper: Removal of surface mount resistor R172 disables write to the to the Xilinx FPGA configuration flash device. The location of R172 is shown in diagram 4502-088.

Board Crystal Oscillator: 133MHz
Frequency Stability: ± 0.0020% or 20ppm
Cypress Clock Buffer Operating Range: 10Mhz to 133MHz

Double Data Rate 2 SDRAM

64M x 32-bit Density
Micron MT47H64M16HR-3IT:E (Two 64M x 16 parts)

SDRAM Crystal Oscillator: 200MHz
Frequency Stability: ± 0.01% or 100ppm

Dual Port SRAM

256K x 64-bit Integrated Devices Technology IDT70T3519S133BC, Maximum 133 Megahertz Speed (Two 256K x 32 parts)

Flash Memory

32M x 8-bit 256 addressable sectors of which 41 are used for FPGA Configuration

PCI Local Bus Interface

PMC Compatibility: Conforms to PCI Bus Specification, Revision 3.0 and PMC Specification, P1386.1

PCI-X bus Master/Target: The board supports 32-bit or 64-bit PCI-X at 100MHz, 66MHz and 33MHz.

4K Memory Space Required: Base Address Register 0 for access to configuration registers

4M Memory Space Required: Base Address Register 1 for access to Dual Port SRAM.

4M Memory Space Required: Base Address Register 2 for access to the user-programmable FPGA (U7).

Signaling: 3.3V Compliant

INTA#: Interrupt A is used to request an interrupt. Source of interrupt can be from the Digital I/O, or DMA Channels.

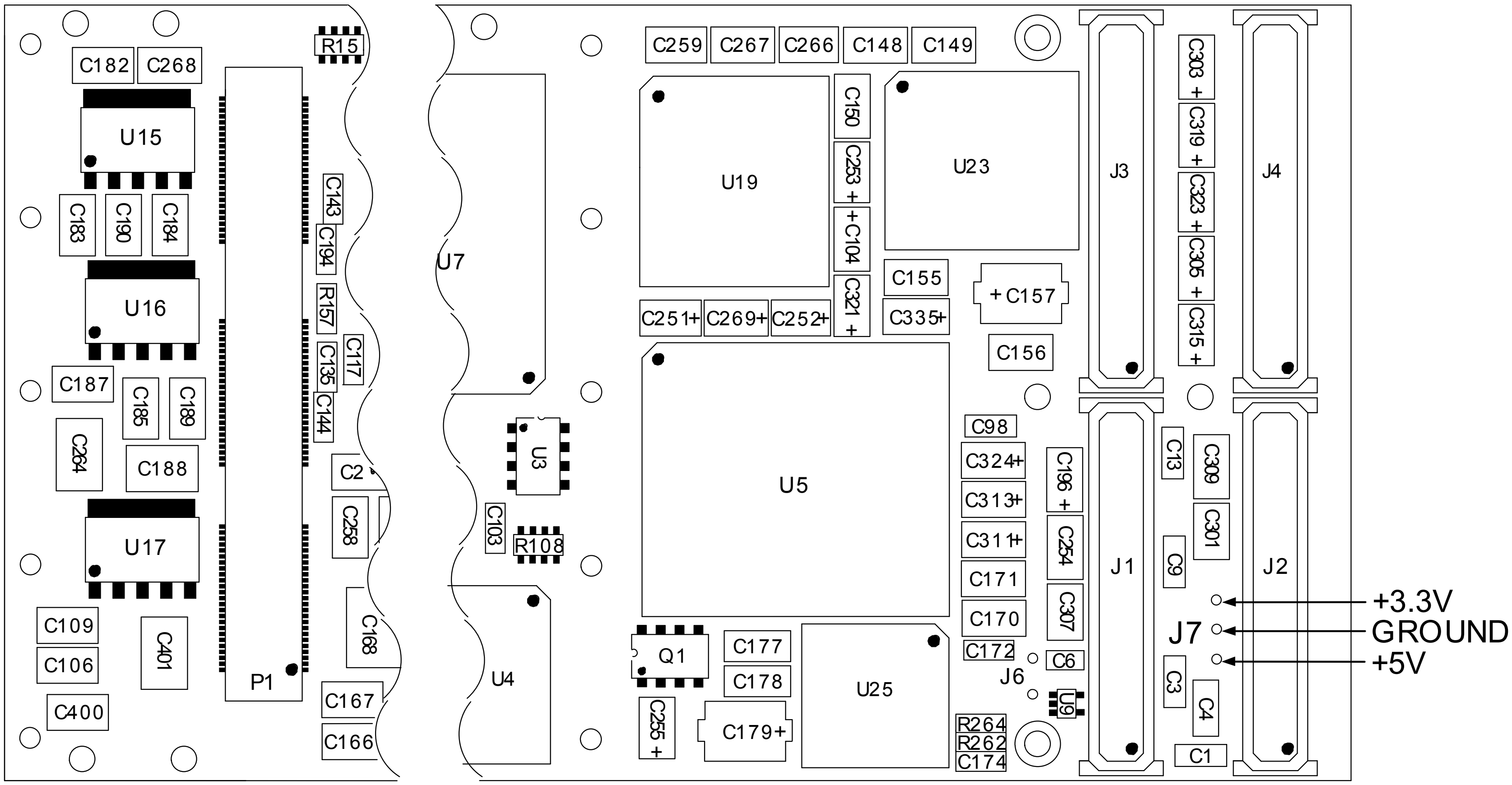
Certificate of Volatility	
Acromag Model PMC-VFX70(E)	Manufacturer: Acromag, Inc. 30765 Wixom Rd Wixom, MI 48393

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: 256K x 72bit	User Modifiable ■ Yes □ No	Function: Data storage for FPGA	Process to Sanitize: Power Down
Type (SRAM, SDRAM, etc.) FPGA based RAM	Size: Variable up to 1.28Mbyte	User Modifiable ■ Yes □ No	Function: Data storage for FPGA	Process to Sanitize: Power Down
Type (SRAM, SDRAM, etc.) SDRAM	Size: 64M x 32bit	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: 32Mbyte	User Modifiable ■ Yes □ No	Function: Storage of Code for FPGA and PowerPC	Process to Sanitize: Clear Flash memory by writing a logic 1 to bit-0 of the Flash Erase Chip Register at PCIBAR2 + 24H
Type(EEPROM, Flash, etc.) Flash	Size: 16Mbit	User Modifiable □ Yes ■ No	Function: Storage of Code for PCI bus 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

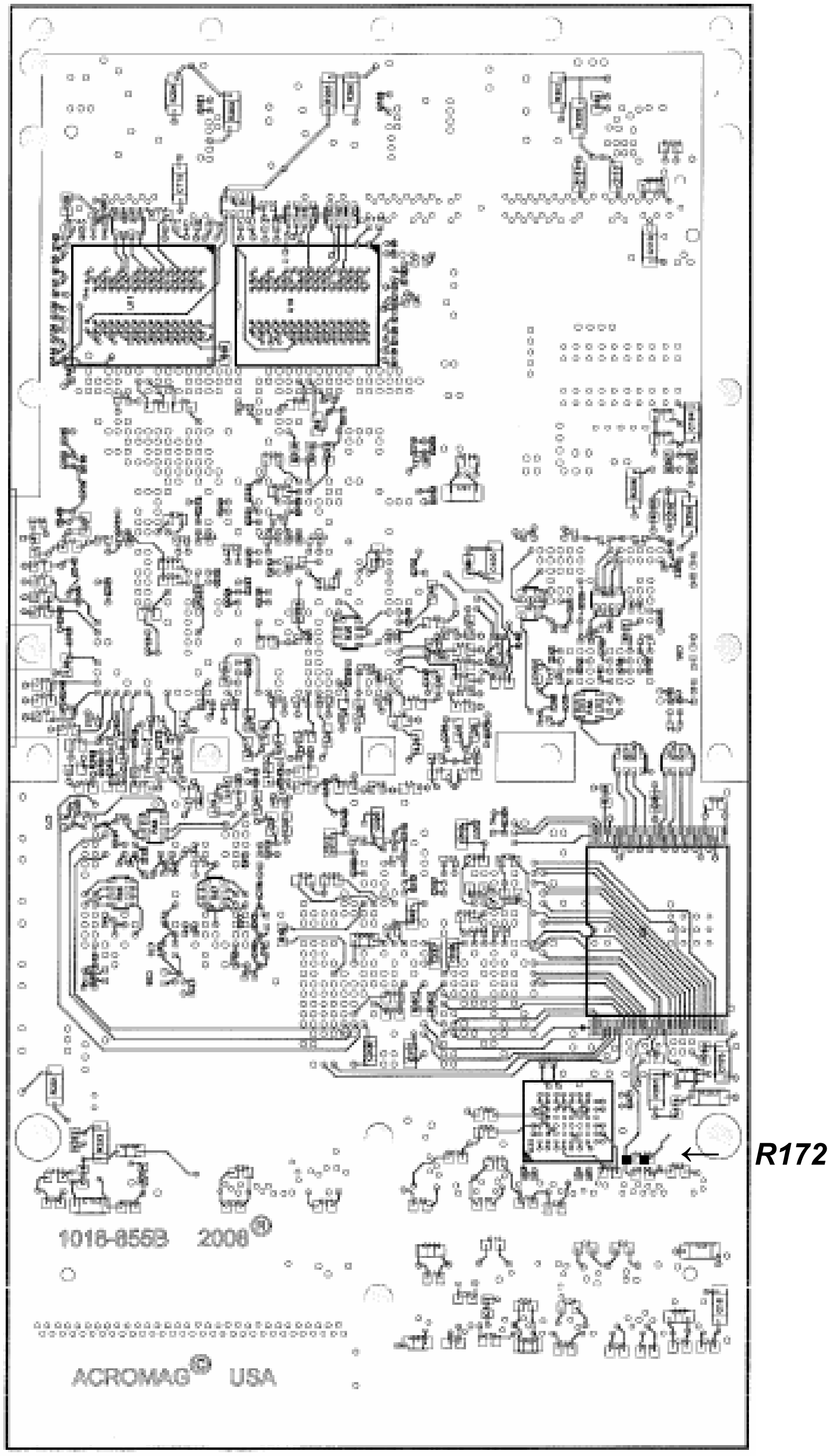
-



J7 EXTERNAL POWER LOCATION
4502-083

In standalone mode, where the card is not plugged into the PCI bus, external power must be provided via the J7 contact holes.

PMC VFX R172 Resistor Location Drawing
4502-088a


R172
1018-8559 2008©
ACROMAG© USA