AP225 12-Bit, 16 Channel Analog Output Module
AP235 16-Bit, 16 Channel Analog Output Module
USER'S MANUAL

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

8501095B

Table of Contents

1. GENERAL INFORMATION

Intended Audience

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

Preface

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

Trademark, Trade Name and Copyright Information

Class A Product Warning

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

Environmental Protection Statement

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

AcroPack Information – All Models

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

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

Ordering Information

The AcroPack ordering options are given in the following table.

Table 1 AP2x5 Models

Model Number	Description	Temp Range
AP225-16E-LF[1]	16-Channel 12-Bit Analog Output	-40°C to 75°C
AP235-16E-LF[1]	16-Channel 16-Bit Analog Output	-40°C to 75°C

Note 1: Applications requiring operating temperatures of 70°C to 75°C will require purchase of the AcroPack Heatsink Accessory AP-CC-01 and minimum airflow of 650LFM. For temperatures below 70°C the module will require a minimal airflow of 650LFM.

AP-CC-01 AcroPack Conduction Cool Kit (See Appendix for installation instructions)

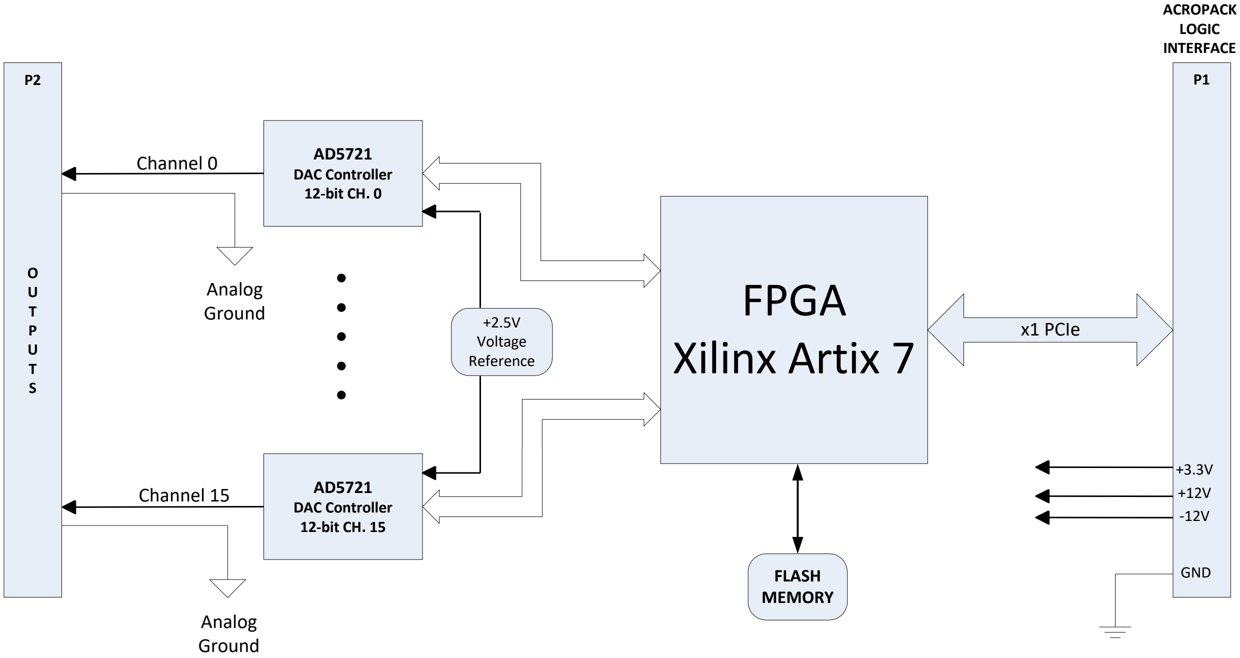


Figure 1 AP225 Block Diagram

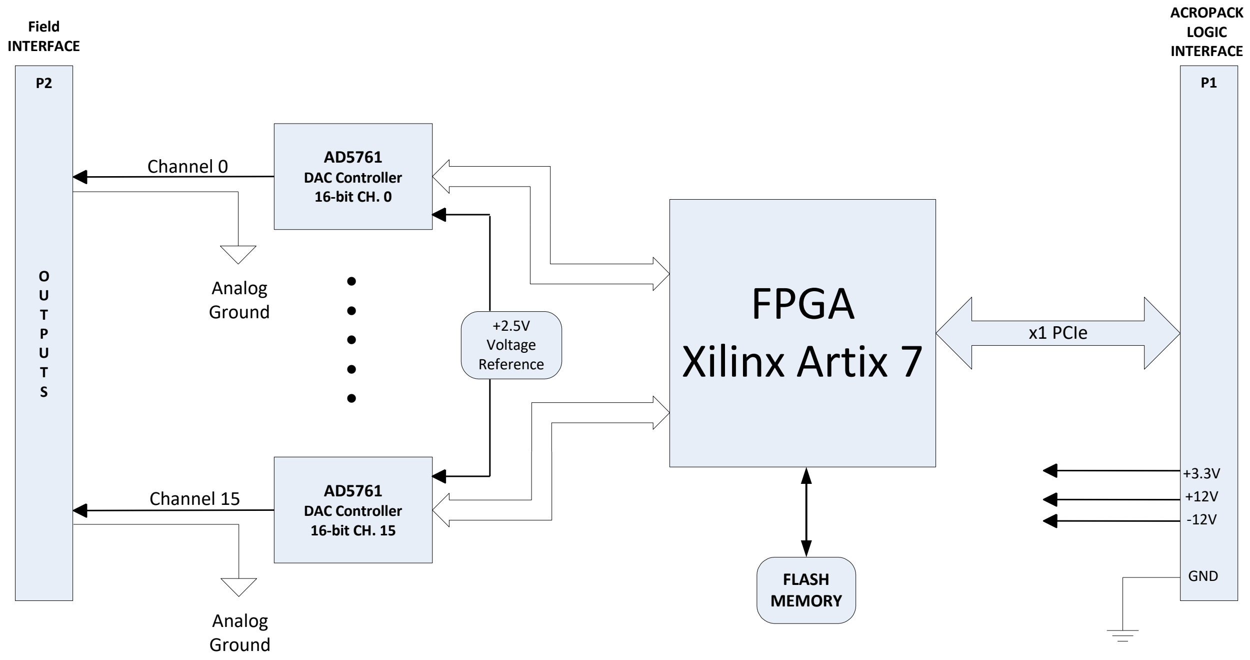


Figure 2 AP235 Block Diagram

Key Features

- **High Channel Count** – Individual control of sixteen analog voltage output channels is provided.
- **Flexible Operating Modes –** Each channel may be independently configured to operate in direct access, single burst, continuous, or streaming (FIFO) mode.
- **Flexible Trigger –** Each channel can be independently configured for software trigger, on-board timer trigger, or external trigger mode. The module can accept an external trigger or be the trigger source for synchronizing with other modules.
- **Flexible Memory Allocation –** The on-board 64K x 16-bit sample memory is shared among the sixteen channels. The amount of memory allocated to each channel is configurable.
- **Continuous Output Mode -** Continuous conversions are implemented by continuously cycling through the waveform memory, from Start Address to End Address, until halted by software. The interval between conversions is controlled by the interval timer. Conversions are initiated by issue of a software or external trigger.
- **FIFO Output Mode -** Each of the channel's FIFOs can be filled/loaded with new data without stopping output waveform generation.
- **Single Conversion Mode -** Conversions are started with the Start Address and can continue until the channels End Address is reached.

- **AP225, 12-Bit Resolution** - Each channel contains its own 12-bit, Digital to Analog Converter (DAC) with 7.5uS output settling time.
- **AP235, 16-Bit Resolution** - Each channel contains its own 16-bit, Digital to Analog Converter (DAC) with 7.5uS output settling time.
- **Software selectable output range** - Provides six voltage ranges:

  0 to 10 Volts, 0 to 5 Volts, +/-10 Volts, +/-5 Volts, +/-3 Volts, - 2.5V to +7.5 Volts
- **Reliable Software Calibration** - Calibration coefficients stored on-board provide the means for accurate software calibration of the module.
- **PCIe Bus** –The PCI Express Generation 1 interface operates at a bus speed of 2.5 Gbps per lane per direction.
- **DMA Operation** –The DMA controller moves data efficiently between the FPGA memory and host memory. This is implemented without use of the host CPU.

Signal Interface Products

This AcroPack Module will mate directly to all Acromag AP carriers. Once connected, the module is accessed via a fifty or sixty-eight pin front panel connector, depending on the chosen carrier.

Cables and termination panels are also available. For optimum performance with the AP2x5 modules, use of the shortest possible length of shielded I/O cable is recommended.

Software Support

The AP2x5 series products require support drivers specific to your operating system. Supported operating systems include: Linux®, Windows®, and VxWorks®.

Windows

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

VxWorks

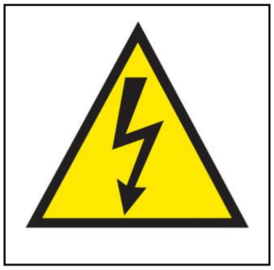
Acromag provides a software product (sold separately) consisting of VxWorks software. This software (Model APSW-API-VXW) is composed of VxWorks (real time operating system) libraries for all AcroPack modules, VPX I/O board products, and PCIe I/O Cards. The software is implemented as a library of "C"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

Linux

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

2. PREPARATION FOR USE

IMPORTANT PERSONAL AND PRODUCT SAFETY CONSIDERATIONS



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

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

UNPACKING AND INSPECTION

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

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

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

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

Card Cage Considerations

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

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

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

Board Installation

Power should be removed from the carrier board when installing AP modules, cables, termination panels, and field wiring. Model AP2x5 digital input boards have no hardware jumpers or switches to configure.

Non-Isolation Considerations

The AP2x5 is non-isolated, since there is electrical continuity between the logic and field I/O grounds. As such, the field I/O connections are not isolated from the system. Care should be taken in designing installations without isolation to avoid noise pickup and ground loops caused by multiple ground connections.

Field I/O Connector

The field I/O interface connector P2 provides a mating interface between the AP2x5 modules and the carrier board.

Table 2 lists pin assignments for each of the AP2x5 field I/O signals. Every other pin of the 100 pin P2 connector is left unconnected.

Table 2 Field I/O Connector Pin Assignments

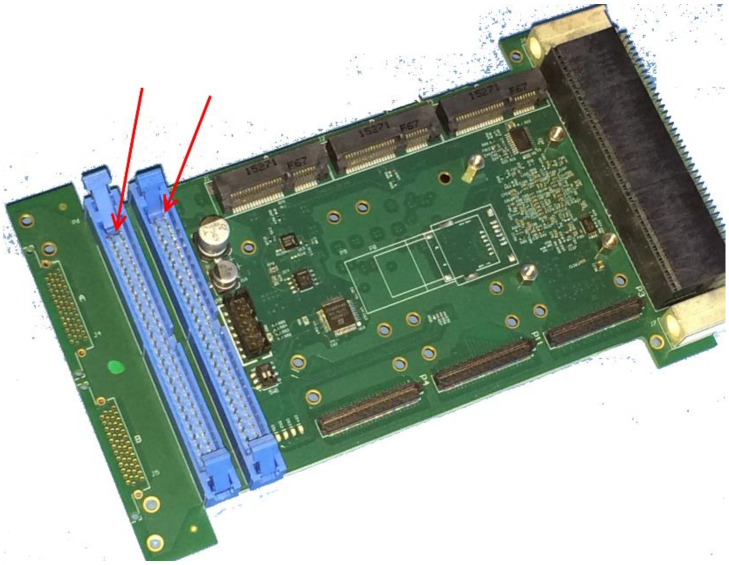
Carrier Connector Ribbon[1]	Carrier Connector 50 pin Champ[2]	Carrier Connector 68 pin Champ	Module P2 Pin Number	Signal Name
1	1	1	2	CH0+
2	26	35	1	Signal Return
			4	Reserved/isolation
			3	Reserved/isolation

Carrier Connector Ribbon[1]	Carrier Connector 50 pin Champ[2]	Carrier Connector 68 pin Champ	Module P2 Pin Number	Signal Name
3	2	2	6	CH1+
4	27	36	5	Signal Return
			8	Reserved/isolation
			7	Reserved/isolation
5	3	3	10	CH2+
6	28	37	9	Signal Return
			12	Reserved/isolation
			11	Reserved/isolation
7	4	4	14	CH3+
8	29	38	13	Signal Return
			16	Reserved/isolation
			15	Reserved/isolation
9	5	5	18	CH4+
10	30	39	17	Signal Return
			20	Reserved/isolation
			19	Reserved/isolation
11	6	6	22	CH5+
12	31	40	21	Signal Return
			24	Reserved/isolation
			23	Reserved/isolation
13	7	7	26	CH6+
14	32	41	25	Signal Return
			28	Reserved/isolation
			27	Reserved/isolation
15	8	8	30	CH7+
16	33	42	29	Signal Return
			32	Reserved/isolation
			31	Reserved/isolation
17	9	9	34	CH8+
18	34	43	33	Signal Return
			36	Reserved/isolation
			35	Reserved/isolation
19	10	10	38	CH9+
20	35	44	37	Signal Return
			40	Reserved/isolation
			39	Reserved/isolation

Carrier Connector Ribbon[1]	Carrier Connector 50 pin Champ[2]	Carrier Connector 68 pin Champ	Module P2 Pin Number	Signal Name
21	11	11	42	CH10+
22	36	45	41	Signal Return
			44	Reserved/isolation
			43	Reserved/isolation
23	12	12	46	CH11+
24	37	46	45	Signal Return
			48	Reserved/isolation
			47	Reserved/isolation
25	13	13	50	CH12+
26	38	47	49	Signal Return
			52	Reserved/isolation
			51	Reserved/isolation
27	14	14	54	CH13+
28	39	48	53	Signal Return
			56	Reserved/isolation
			55	Reserved/isolation
29	15	15	58	CH14+
30	40	49	57	Signal Return
			60	Reserved/isolation
			59	Reserved/isolation
31	16	16	62	CH15+
32	41	50	61	Signal Return
			64	Reserved/isolation
			63	Reserved/isolation
33	17	17	66	Reserved/isolation
34	42	51	65	Signal Return
			68	Reserved/isolation
			67	Reserved/isolation
35	18	18	70	Reserved/isolation
36	43	52	69	Reserved/isolation
			72	Reserved/isolation
			71	Reserved/isolation
37	19	19	74	Reserved/isolation
38	44	53	73	Reserved/isolation
			76	Reserved/isolation
			75	Reserved/isolation

Carrier Connector Ribbon[1]	**Carrier Connector 50 pin Champ**[2]	**Carrier Connector 68 pin Champ**	**Module P2 Pin Number**	**Signal Name**
39	20	20	78	Reserved/isolation
40	45	54	77	Reserved/isolation
			80	Reserved/isolation
			79	Reserved/isolation
41	21	21	82	Ext_Trig_n
42	46	55	81	Reserved/isolation
			84	Reserved/isolation
			83	Reserved/isolation
43	22	22	86	Reserved/isolation
44	47	56	85	Reserved/isolation
			88	Reserved/isolation
			87	Reserved/isolation
45	23	23	90	Reserved/isolation
46	48	57	89	Reserved/isolation
			92	Reserved/isolation
			91	Reserved/isolation
47	24	24	94	EXT_VSS_DAC
48	49	58	93	Signal Return
			96	Reserved/isolation
			95	Reserved/isolation
49	25	25	98	EXT_VDD_DAC
50	50	59	97	Signal Return
			100	Reserved/isolation
			99	Reserved/isolation

Note 1: VPX4500-CC-LF is an example of a carrier that uses the 50-pin ribbon cable connector see image of carrier.



Note 2: APCe7020E-LF is an example of a carrier that uses the 50-pin Champ connector see image of carrier.

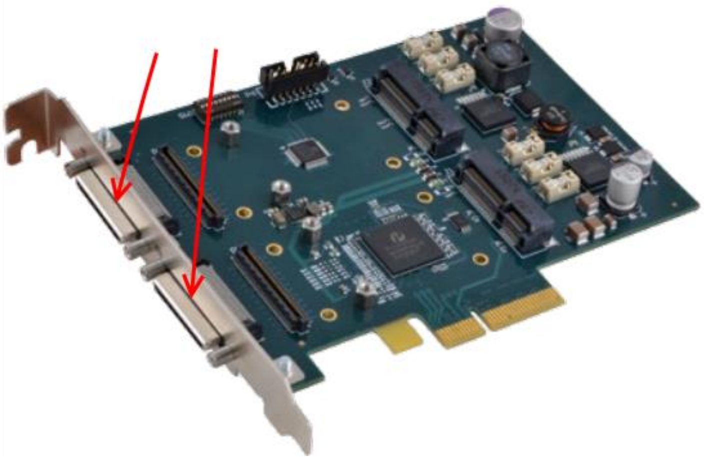


Table 3 Mini-PCIe Connector

Pin #	Name	Pin #	Name
51	**+5V**[2]	52	+3.3V[3]
49	**+12V**[2]	50	GND
47	**-12V**[2]	48	+1.5V
45	**Present**[2]	46	N.C. (LED_WPAN#)[1]
43	GND	44	N.C. (LED WLAN#)[1]
41	+3.3V[3]	42	N.C. (LED_WWAN#)[1]
39	+3.3V[3]	40	GND
37	GND	38	N.C. (USB_D+)[1]

Pin #	Name	Pin #	Name
35	GND	36	N.C. (USB D-)[1]
33	PETp0	34	GND
31	PETn0	32	SMB DATA[4]
29	GND	30	SMB_CLK[4]
27	GND	28	+1.5V
25	PERp0	26	GND
23	PERn0	24	+3.3V[3]
21	GND	22	PERST#
19	**TDI** (UIM C4)[1]	20	N.C. (W DISABLE#)[1]
17	**TDO** (UIM_C8)[1]	18	GND
15	GND	16	UIM VPP
13	RECLK+	14	UIM_RESET
11	REFCLK-	12	UIM CLK
9	GND	10	UIM_DATA
7	CLKREQ#	8	UIM PWR
5	**TCK** (COEX2)[1]	6	+1.5V
3	**TMS** (COEX1)[1]	4	GND
1	N.C. (WAKE#)[1]	2	+3.3V[3]

1. **Note 1:** The following mini-PCIe signals are not used by the AP2x5: USB_D+, USB_D-, WAKE#, LED_WPAN#, LED_WLAN#, LED_WWAN#, W_DISABLE#, COEX1, COEX2, UIM_C4, UIM_C8. The following signals UIM_C4, UIM_C8, COEX2 and COEX1 are repurposed for JTAG.
2. **Note 2:** +5, +12, and -12 Volt power supplies have been assigned to pins that are reserved in the mini-PCIe specification. The Present signal is tied to circuit common on the AP module.
3. **Note 3:** All +3.3Vaux power pins are changed to system +3.3V power.
4. **Note 4:** The SM bus signals SMB_CLK and SMB_DATA will be used to clock a carrier location serial stream from the carrier. These signals are under the control of the AcroPack module. We have a unique controller implemented in a Carrier CPLD. The FPGAs on the AcroPacks use the SM data and clock to retrieve the board location but not with the I2C protocol.

3. PROGRAMMING INFORMATION

This Section provides the specific information necessary to program and operate the AP225 or AP235 module.

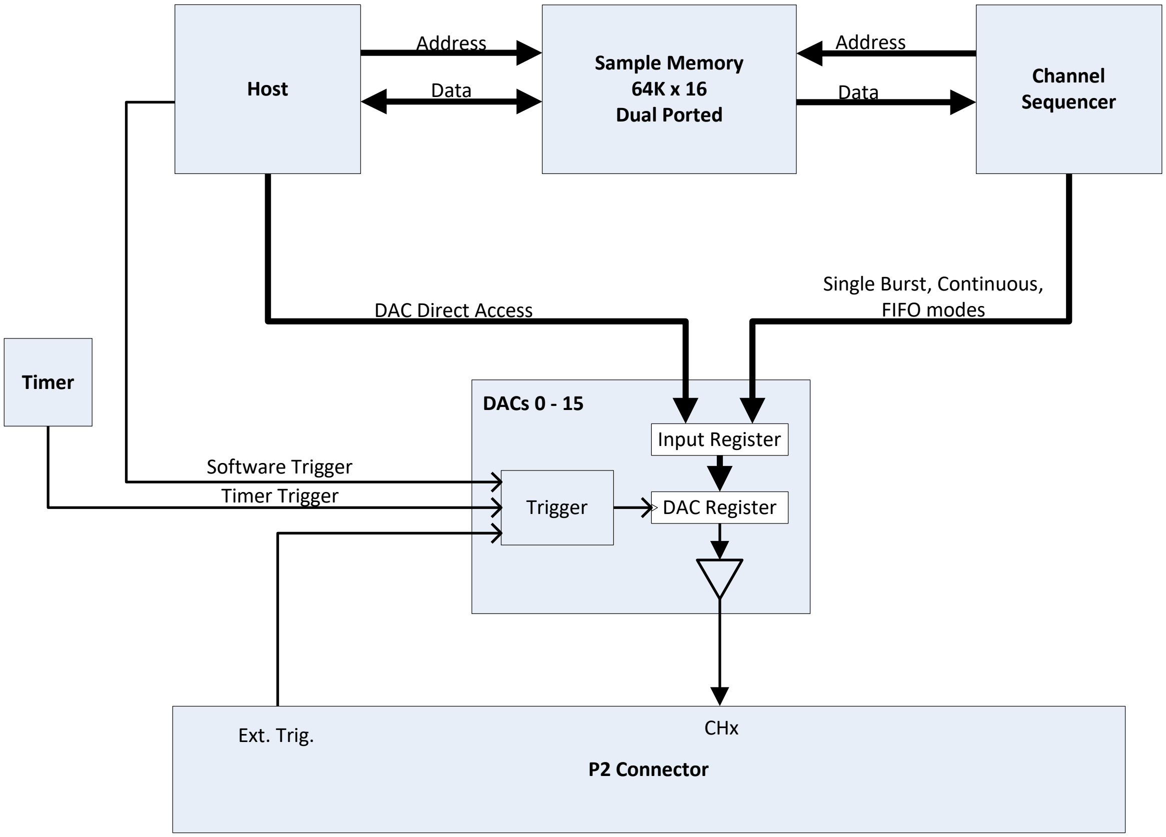


Figure 3 AD5721/61 D/A Converter Interface Block Diagram

Figure 3 shows the two data paths used to write commands to the DACs' input registers. Commands can be written to the DACs' inputs registers from the host or from the channel sequencer. The channel sequencer is used for transferring samples from sample memory to the DACs' input registers independent of host processor. The channel's sequencer is used in Single Burst, Continuous, and FIFO modes. The Sample Memory is dual ported allowing the host and channel sequencer to access the memory simultaneously. There are three trigger sources available to each channel, External, Timer, and Software. Each DAC can be independently configured to use any of the three trigger sources. All DACs configured for simultaneous output mode that have the same trigger source will be synchronous with each other.

PCIe CONFIGURATION ADDRESS SPACE

This AP2X5 series modules are PCI Express Base Specification Revision v2.1 compliant.

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

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

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

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

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

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

CONFIGURATION REGISTERS

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

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

Table 4 Configuration Registers

Reg. Num.	**D31 D24**	**D23 D16**	**D15 D8**	**D7 D0**
0	**Device ID** 0x701C AP225 0x701D AP235		**Vendor ID** 16D5	
1	Status		Command	
2	Class Code=118000			Rev ID=00

3	BIST	Header	Latency	Cache
4:5	64-bit Memory Base Address for Memory Accesses to PCIe interrupt, I/O registers, System Monitor registers, and Flash memory. 1 MB Space **(BAR0)**			
6:10	Not Used			
11	**Subsystem ID** 0x701C AP225 0x701D AP235		**Subsystem Vendor ID** 16D5	
12	Not Used			
13,14	Reserved			
15	Max_Lat	Min_Gnt	Inter. Pin	Inter. Line

This board is allocated a 1M byte block of memory (BAR0), to access the PCIe interrupt, I/O registers, XADC registers, and Flash memory. The PCIe bus decodes 1M bytes for BAR0 for this memory space.

BAR0 MEMORY MAP

The BAR0 memory address space is used to access the devices listed in Table 5. Note that the base address for the board (BAR0) in memory space must be added to the addresses shown to properly access these registers.

Table 5 BAR0 Registers

BAR0 Base Address	Size	Description
0x0000_0000→0x0000_0FFF	4K	CDMA (see PG034)
0x0000_1000→0x0000_1FFF	4K	PCIe AXI Bridge Control (see PG054)
0x0000_2000→0x0000_2FFF	4K	Interrupt Controller
0x0000_3000→0x0000_3FFF	4K	XADC System Monitor (see DS790)
0x0000_4000→0x0000_4FFF	4K	Firmware Revision Register
0x0000_5000→0x0000_5FFF	4K	Flash AXI_QSPI (see pg153)
0x0000_6000→0x0000_6FFF	4K	Slot ID Register
0x0000_7000→0x0000_9FFF	12K	Reserved
0x0000_A000→0x0000_BFFF	8K	Block RAM
0x0000_C000→0x0003_FFFF	212K	Reserved
0x0004_0000→0x0004_0213	1K	DAC Registers
0x0004_0214→0x0005_FFFF	127K	Reserved
0x0006_0000→0x0007_FFFF	128K	Sample Memory

BAR0 Base Address	Size	Description
0x0008_0000→0x000F_FFFF	512K	AXI to PCIe BAR0

CDMA MEMORY MAP

The Central Direct Memory Access (CDMA) controller can access the AXI to PCI bridge (BAR0), and the AD5721/61 DAC peripheral which includes the DAC registers and Sample Memory.

BAR0 Base Address	Size	Description
0x0000_1000→0x0000_1FFF	4K	PCIe AXI Bridge Control (see PG054)
0x0000_A000→0x0000_BFFF	8K	Block RAM
0x0004_0000→0x0004_0213	1K	DAC Registers
0x0004_0214→0x0005_FFFF	127K	Reserved
0x0006_0000→0x0007_FFFF	128K	Sample Memory
0x0008_0000→0x000F_FFFF	512K	AXI to PCIe BAR0

SCATTER GATHER MEMORY MAP

The scatter-gather engine can access the block RAM Memory only. The scatter-gather descriptor list must be located in this memory. The block RAM can hold 128 descriptors.

BAR0 Base Address	Size	Description
0x0000_A000→0x0000_BFFF	8K	Block RAM

AXI-CDMA

The AXI Central Direct Memory Access (CDMA) core is a soft Xilinx Intellectual Property core. The CDMA provides direct memory access between system memory over the PCIe bus and the memory resident on the AP2x5 series module.

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

Scatter Gather is a mechanism that allows for automated DMA transfer scheduling via a pre-programmed instruction list of transfer descriptors. This instruction list is programmed by the user software application into a memory-resident data structure that must be accessible by the AXI CDMA

Scatter Gather interface. This list of instructions is organized into what is referred to as a transfer descriptor chain. Each descriptor has an address pointer to the next descriptor to be processed. The last descriptor in the chain generally points back to the first descriptor in the chain but it is not required. The AXI CDMA Tail Descriptor Pointer register needs to be programmed with the address of the first word of the last descriptor of the chain. When the AXI CDMA executes the last descriptor, and finds that the Tail Descriptor pointer register contents matches the address of the completed descriptor, the Scatter Gather Engine stops descriptor fetching and waits. See the Xilinx AXI Central Direct Memory Access product guide PG034 for additional details for Scatter Gather operations.

Table 6 AXI CDMA Registers

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

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

CDMA Control Register

This register provides software application control of the AXI CDMA.

Table 7 CDMA Control Register (Read/Write) - (BAR0 + 0x0000)

Bit(s)	Function
0	This bit is reserved for future definition and will always return zero.
1	This bit will always return one.

2	Soft reset control for the AXI CDMA core. Setting this bit to a ‘1’ causes the AXI CDMA to be reset. Reset is accomplished gracefully. Committed AXI4 transfers are then completed. Other queued transfers are flushed. After completion of a soft reset, all registers and bits are in the Reset State.	
	0	Reset Not in Progress
	1	Reset in Progress
3	This bit controls the transfer mode of the CDMA. Setting this bit to a ‘1’ causes the AXI CDMA to operate in a Scatter Gather mode. **Note:** This bit must only be changed when the CDMA engine is IDLE (CDMA Status bit-1 = ‘1’). Changing the state of this bit at any other time has undefined results. **Note:** This bit must be set to a 0 then back to 1 by the software application to force the CDMA Scatter Gather engine to use a new value written to the CDMA Current Descriptor Pointer register. **Note:** This bit must be set prior to setting Bit-13 of this CDMA Control register.	
	0	Simple DMA Mode
	1	Scatter Gather Mode
4	Reserved, always write zero.	
5	Key Hole write. Writing 1 to this enables the keyhole write (FIXED address AXI transaction). This value should not be changed when a transfer is in progress. This value should remain constant until all the descriptors are processed (for SG = 1). CDMA shows unexpected behavior if this value is changed in the middle of a transfer.	
	0	Key Hole Write Disabled
	1	Key Hole Write Enabled

6		Cyclic BD Enable. When set to 1, you can use the CDMA in Cyclic Buffer Descriptor (BD) mode without any user intervention. In this mode, the Scatter Gather module ignores the Completed bit of the BD. With this feature, you can use the same BDs in cyclic manner without worrying about any errors. This bit should be set before updating the TAILDESC register. Changing this bit while the transfer is in progress will generate undefined results.
	0	Cyclic BD Disabled
	1	Cyclic BD Enabled
11-7		Reserved
12		Interrupt on Complete Interrupt Enable. When set to '1', it allows an interrupt after completed DMA transfers.
	0	Interrupt on Complete Disabled
	1	Interrupt on Complete Enabled
13		Interrupt on Delay Timer Interrupt Enable. When set to '1', it allows a delayed interrupt out. This is only used with Scatter Gather assisted transfers.
	0	Delayed Interrupt Disabled
	1	Delayed Interrupt Enabled
14		Interrupt on Error Interrupt Enable. When set to '1', it allows an error to generate an interrupt out.
	0	Error Interrupt Disabled
	1	Error Interrupt Enabled
15		Reserved

23-16	Interrupt Threshold value. This field is used to set the Scatter Gather interrupt coalescing threshold. When Interrupt On Complete interrupt events occur, an internal counter counts down from the Interrupt Threshold setting. When the count reaches zero, an interrupt out is generated by the CDMA engine. **Note:** The minimum setting for the threshold is 0x01. A write of 0x00 to this register has no effect. If the CDMA is built with Scatter Gather disabled (Simple Mode Only), the default value of the port is zeros.
31-24	Interrupt Delay Time Out. This value is used for setting the interrupt delay time out value. The interrupt time out is a mechanism for causing the CDMA engine to generate an interrupt after the delay time period has expired. This is used for cases when the interrupt threshold is not met after a period of time, and the CPU desires an interrupt to be generated. Timer begins counting when the CDMA is IDLE (CDMA Status bit-1 = '1'). This generally occurs when the CDMA has completed all scheduled work defined by the transfer descriptor chain (reached the tail pointer) and has not satisfied the Interrupt Threshold count. **Note:** Setting this value to zero disables the delay timer interrupt.

CDMA Status Register

This register provides status of the AXI CDMA.

Table 8 CDMA Status Register (Read/Write) - (BAR0 + 0x0004)

Bit(s)	Function
0	This bit is reserved for future definition and will always return zero.
1	CDMA Idle. Indicates the state of AXI CDMA operations. When set and in Simple DMA mode, the bit indicates the programmed transfer has completed and the CDMA is waiting for a new transfer to be programmed. Writing to the "Bytes to Transfer" register in Simple DMA mode causes the CDMA to start (not Idle). When set and in Scatter Gather mode, the bit indicates the Scatter Gather Engine has reached the tail pointer for the associated channel and all queued descriptors have been processed. Writing to the tail pointer register automatically restarts CDMA Scatter Gather operations.

Bit(s)	**Function**	
	0	Not Idle
	1	CDMA is Idle
2	Reserved	
3	Scatter Gather Included. This bit indicates the AXI CDMA has been implemented with Scatter Gather support included (C_SG_ENABLE = 1). This is used by application software (drivers) to determine if Scatter Gather Mode can be utilized.	
	1	Scatter Gather is included
4	DMA Internal Error. This bit indicates that an internal error has been encountered by the DataMover on the data transport channel. This error can occur if a 0 value Byte to Transfer register is fed to the AXI DataMover, or DataMover has an internal processing error. A Bytes to Transfer register value of 0 only happens if the register is written with zeros (in Simple DMA mode) or a Bytes to Transfer register value of zero is specified in the Control word of a fetched descriptor (Scatter Gather Mode). This error condition causes the AXI CDMA to gracefully halt. The CDMA Status register bit-1 is set to ‘1’when the CDMA has completed shut down. A reset (soft or hard) must be issued to clear the error condition.	
	0	No CDMA Internal Errors
	1	CDMA Internal Error detected. CDMA Engine halts.
5	DMA Slave Error. This bit indicates that an AXI slave error response has been received by the AXI DataMover during an AXI transfer (read or write). This error condition causes the AXI CDMA to gracefully halt. The CDMA Status register bit-1 is set to ‘1’ when the CDMA has completed shut down. A reset (soft or hard) must be issued to clear the error condition.	
	0	No CDMA Slave Errors
	1	CDMA Slave Error detected. CDMA Engine halts.

Bit(s)	Function	
6	DMA Decode Error. This bit indicates that an AXI decode error has been received by the AXI DataMover. This error occurs if the DataMover issues an address that does not have a mapping assignment to a slave device. This error condition causes the AXI CDMA to halt gracefully. The CDMA Status register bit-1 is set to ‘1’ when the CDMA has completed shut down. A reset (soft or hard) must be issued to clear the error condition.	
	0	No CDMA Decode Errors
	1	CDMA Decode Error detected. CDMA Engine halts.
7	Reserved	
8	Scatter Gather Internal Error. This bit indicates that an internal error has been encountered by the Scatter Gather Engine. This error condition causes the AXI CDMA to gracefully halt. The CDMA Status register bit-1 is set to 1 when the CDMA has completed shut down. A reset (soft or hard) must be issued to clear the error condition.	
	0	No Scatter Gather Internal Errors
	1	Scatter Gather Internal Error. CDMA Engine halts.
9	Scatter Gather Slave Error. This bit indicates that an AXI slave error response has been received by the Scatter Gather Engine during an AXI transfer (transfer descriptor read or write). This error condition causes the AXI CDMA to gracefully halt. The CDMA Status register bit-1 is set to 1 when the CDMA has completed shut down. A reset (soft or hard) must be issued to clear the error condition.	
	0	No Scatter Gather Slave Errors
	1	Scatter Gather Slave Error. CDMA Engine halts.

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

Bit(s)	Function	
	0	No Error Interrupt
	1	Error Interrupt Active
15	Reserved	
23-16	Interrupt Threshold Status. This field reflects the current interrupt threshold value in the Scatter Gather Engine.	
31-24	Interrupt Delay Time Status. This field reflects the current interrupt delay timer value in the Scatter Gather Engine.	

CDMA Current Descriptor Pointer Register

Table 9 CDMA Current Descriptor Pointer Register (Read/Write) - (BAR0 + 0x00000008)

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

CDMA Tail Descriptor Pointer Register

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

Table 10 CDMA Tail Descriptor Pointer Register (Read/Write) - (BAR0 + 0x00000010)

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

CDMA Source Address Register

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

If a location in system memory is the source address, it must be set with the AXI aperture base address 0x00080000 + the least significant 19-bits of the system memory address.

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

Table 11 CDMA Source Address Register (Read/Write) - (BAR0 + 0x0018)

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

CDMA Destination Address Register

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

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

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

Table 12 CDMA Destination Address Register (Read/Write) - (BAR0 + 0x0020)

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

CDMA Bytes to Transfer Register

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

Table 13 CDMA Bytes to Transfer Register (Read/Write) - (BAR0 + 0x0028)

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

Scatter Gather Transfer Descriptor Definition

This defines the format and contents of the AXI CDMA Scatter Gather Transfer Descriptors. These are used only by the SG function if the CDMACR.SGMode bit is set to 1. A transfer descriptor consists of eight 32-bit words. The descriptor represents the control and status information needed for a single CDMA transfer plus address linkage to the next sequential descriptor. Each descriptor can define a single CDMA transfer of up to 8,388,607 Bytes of data. A descriptor chain is defined as a series of descriptors that are sequentially linked through the address linkage built into the descriptor format.

The AXI CDMA SG Engine traverses the descriptor chain following the linkage until the last descriptor of the chain has been completed. The relationship and identification of the AXI CDMA transfer descriptor words is shown in Table 14.

Note: Transfer Descriptors must be aligned on sixteen 32-bit word alignment that is 16x4 Bytes = 64 Bytes. Example valid offsets are

0x00, 0x40, 0x80, and 0xC0.

Table 14 Transfer Descriptor

Address Space Offset	Name	Description
0x00	NXTDESC_PNTR	Next Descriptor Pointer
0x04		Reserved
0x08	SA	Source Address

Address Space Offset	Name	Description
0x0C		Reserved
0x10	DA	Destination Address
0x14		Reserved
0x18	CONTROL	Transfer Control
0x1C	STATUS	Status

Transfer Descriptor NXTDESC_PNTR

This word provides the address pointer to the first word of the next transfer descriptor in the descriptor chain.

Table 15 Transfer Descriptor NXTDESC_PNTR (Offset 0x00)

Bit(s)	Function
5 - 0	These bits are reserved and fixed to zeros. This forces the address value programmed in this register to be aligned to 64-byte aligned addresses.
31 - 6	Next Descriptor Pointer. This field is an address pointer (most significant 26 bits) to the first word of the next transfer descriptor to be executed by the CDMA SG Engine. The least-significant 6 bits of this register are appended to this value when used by the SG Engine forcing transfer descriptors to be loaded in memory at 64-byte address alignment.

Transfer Descriptor Source Address

This word provides the starting address for the data read operations for the associated DMA transfer.

Table 16 Transfer Descriptor SA (Offset 0x04)

Bit(s)	Function
31 - 0	Source Address. This value specifies the starting address for data read operations for the associated DMA transfer. The address value can be at any byte offset.

Transfer Descriptor Destination Address

This word provides the starting address for the data write operations for the associated DMA transfer.

Table 17 Transfer Descriptor DA (Offset 0x10)

Bit(s)	Function
31 - 0	Destination Address. This value specifies the starting address for data write operations for the associated DMA transfer. The address value can be at any byte offset.

Transfer Descriptor CONTROL Word

This word provides the starting address for the data write operations for the associated DMA transfer.

Table 18 Transfer Descriptor Control Word (Offset 0x18)

Bit(s)	Function
22 - 0	Bytes to Transfer. This field in the Control word specifies the desired number of bytes to DMA from the Source Address to the Destination Address. A maximum of 8,388,607 bytes of data can be specified by this field for the associated transfer. A value of zero (0) is not allowed and causes a DMA internal error to be set by AXI CDMA.
31 - 23	Reserved

Transfer Descriptor Status Word

This word provides the starting address for the data write operations for the associated DMA transfer.

Table 19 Transfer Descriptor Status Word (Offset 0x1C)

Bit(s)	Function
27 - 0	Reserved
28	DMA Internal Error. This bit indicates that an internal error was encountered by the AXI CDMA DataMover on the data transport channel during the execution of this descriptor. This error can occur if a 0 value BTT (bytes to transfer) is fed to the AXI DataMover or if the DataMover has an internal processing error. A BTT of 0 only happens if the BTT field in the transfer descriptor CONTROL word is programmed with a value of zero. This error condition causes the AXI CDMA to gracefully halt. The CDMASR.IDLE bit is set to 1 when the CDMA has completed shutdown. 0 = No CDMA Internal Errors 1 = CDMA Internal Error detected. CDMA Engine halts at this descriptor

Bit(s)	Function
29	DMA Slave Error. This bit indicates that an AXI slave error response was received by the AXI CDMA DataMover during the AXI transfer (read or write) associated with this descriptor. This error condition causes the AXI CDMA to halt gracefully. 0 = No CDMA Slave Errors 1 = CDMA Slave Error received. CDMA Engine halts at this descriptor.
30	DMA Decode Error. This bit indicates that an AXI decode error was received by the AXI CDMA DataMover. This error occurs if the DataMover issues an address that does not have a mapping assignment to a slave device. This error condition causes the AXI CDMA to halt gracefully. 0 = No CDMA Decode Errors 1 = CDMA Decode Error received. CDMA Engine halts at this descriptor
31	Transfer Completed. This indicates to the software application that the CDMA Engine has completed the transfer as described by the associated descriptor. The software application can manipulate any descriptor with the Completed bit set to 1. 0 = Descriptor not completed 1 = Descriptor completed If the CDMA SG Engine fetches a descriptor, with this bit set to 1, the descriptor is considered a stale descriptor. An SGIntErr is flagged in the AXI CDMA Status register and the AXI CDMA engine halts with no update to the descriptor.

Simple CDMA Programming Example

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

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

3. Write the source address for the transfer to the Source Address register at 0x0020. In this example, the source is the system memory. The following is required:

   a. Given physical address of buffer of 0x0000333012345678

   b. AXIBAR2PCIEBAR_0U <offset 00001208> = 0x00003330

   c. AXIBAR2PCIEBAR_0L <offset 0000120C> = 0x12345678

   d. The least significant 19 bits of this address 0x12345678 must be added to the AXI BAR0 Aperture Base address. The new AXI address is 0x00080000 + 0x0045678 = 0x000C5678. Write the value 0x000C5678 to the CDMA Source Address register at location 0x0020.

4. Write the destination address to the Destination Address register at 0x0018. For this example, the destination will be the sample buffer of the DAC interface. Write 0x0006_0000 to the Destination Address register at 0x0018.

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

6. Poll the CDMA Status register bit-1 (CDMA idle) for logic '1'.

8. When ready for another transfer. Go back to step 1.

AXI-BAR0 Aperture Base Address

The AXI BAR0 aperture base address of 0x00080000 is set as the base address on the AXI bus used to reach system host memory for CDMA transfers.

In Vivado IP Integrator the address map will show that a 512K address space for the AXI BAR0 Aperture Base Address is reserved.

Table 20 AXI BAR0 Aperture Base Address

0x00080000→0x000FFFFF	512K	Window into PCIe Interface AXI BAR0 Aperture Base Address

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

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

The least significant 19 bits of this address 0x56ABCDEF must be added to the AXI BAR0 Aperture Base address. The new AXI address is 0x00080000 + 0x003CDEF = 0x000BCDEF. These values are then appended by the PCIe AXI bridge to give the final PCIe address of the system memory location.

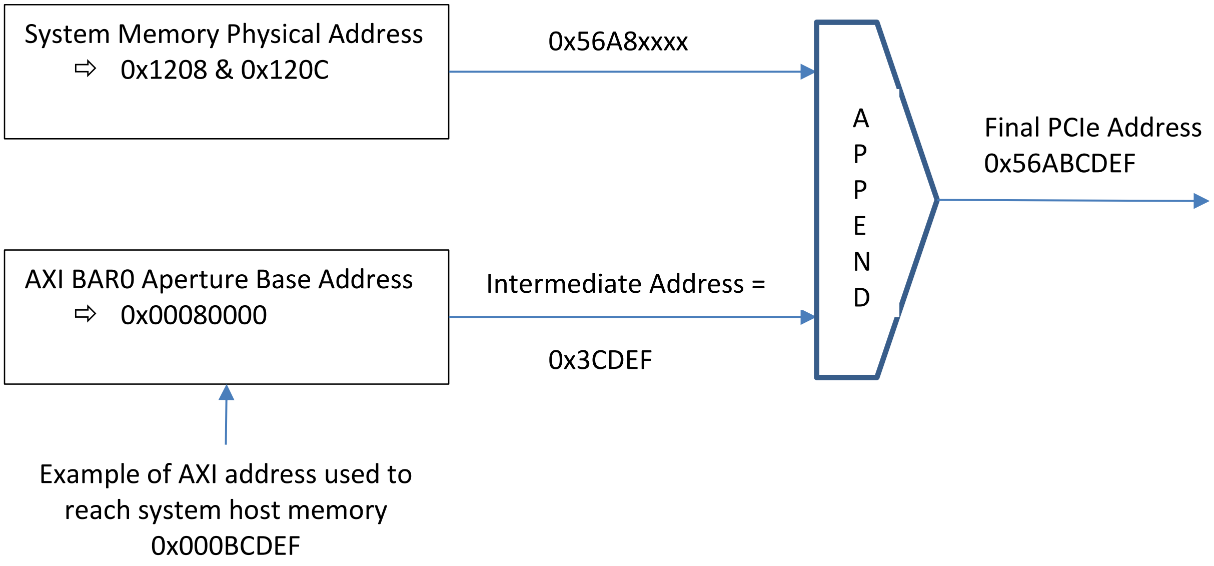


Figure 4 AXI to System Physical Address Translation

INTERRUPT CONTROLLER

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

Table 21 Interrupt Controller Registers

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

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

Interrupt Status Register

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

Table 22 Interrupt Status Register (Read/Write) - (BAR0 + 0x2000)

Bit(s)	Function	
0	When set indicates DAC channel 0 requires service.	
	0	No service required
	1	In single burst mode indicates burst complete In FIFO mode, indicates FIFO is half full
⋮	⋮	
15	When set indicates DAC channel 15 requires service.	
	0	No service required
	1	In single burst mode indicates burst complete In FIFO mode, indicates FIFO is half full
16	When set indicates an AXI CDMA interrupt. See the CDMA section for more information on the source of this interrupt.	
	0	Interrupt Inactive
	1	Interrupt Active
31-17	Reserved	
	0	NA
	1	NA

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

Interrupt Pending Register

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

Table 23 Interrupt Pending Register (Read) - (BAR0 + 0x2004)

Bit(s)	Function	
0	When set indicates DAC channel 0 requires service.	
	0	No Interrupt Pending
	1	Interrupt Pending
⋮	⋮	
15	When set indicates DAC channel 15 requires service.	
	0	No Interrupt Pending
	1	Interrupt Pending
16	When set indicates an AXI CDMA interrupt is pending. See the CDMA section for more information on the source of this interrupt.	
	0	No Interrupt Pending
	1	Interrupt Pending
31-17	Reserved	
	0	NA
	1	NA

Interrupt Enable Register

This is a read/write register. Writing a '1' to a bit in this register enables the corresponding Interrupt Status bit to cause assertion of the interrupt output. This Interrupt Enable bit set to '0' does not inhibit an interrupt condition from being captured. It will still show up in the Interrupt Status register even when not enabled here. To show up in the Interrupt Pending register it needs to be enabled here. Writing a '0' to a bit disables, or masks, the generation of an interrupt output for the corresponding interrupt input signal. Note however, that disabling an interrupt input is not the same as clearing it. Disabling an active interrupt prevents that interrupt from reaching the IRQ output. When it is re-enabled, the interrupt immediately generates a request on the IRQ output. An interrupt must be cleared by writing to the Interrupt Acknowledge Register, as described below. Reading this Interrupt Enable register indicates which interrupt inputs are enabled; where a '1' indicates the input is enabled and a '0' indicates the input is disabled.

Table 24 Interrupt Enable Register (Read/Write) - (BAR0 + 0x2008)

Bit(s)	Function	
0	When set enables DAC channel 0 to generate interrupts.	
	0	Interrupt Disabled
	1	Interrupt Enabled
⋮	⋮	
15	When set enables DAC channel 15 to generate interrupts.	
	0	Interrupt Disabled
	1	Interrupt Enabled
16	When set indicates an AXI CDMA interrupt is enabled. See the CDMA section for more information on the source of this interrupt.	
	0	Interrupt Disabled
	1	Interrupt Enabled
31-17	Reserved	
	0	NA
	1	NA

Interrupt Acknowledge Register

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

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

Table 25 Interrupt Acknowledge Register (Write) - (BAR0 + 0x200C)

Bit(s)	Function
0	Clear DAC channel 0 interrupt request
⋮	⋮
15	Clear DAC channel 15 interrupt request
16	Clear AXI CDMA interrupt request
31-17	Reserved

Set Interrupt Enable Register

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

Table 26 Set Interrupt Enable Register (Write) - (BAR0 + 0x2010)

Bit(s)	Function
0	Set DAC channel 0 interrupt enable
⋮	⋮
15	Set DAC channel 15 interrupt enable
16	Set AXI CDMA interrupt enable
31-17	Reserved

Clear Interrupt Enable Register

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

Table 27 Clear Interrupt Enable Register (Write) - (BAR0 + 0x2014)

Bit(s)	Function
0	Clear DAC channel 0 interrupt enable
⋮	⋮
15	Clear DAC channel 15 interrupt enable
16	Clear AXI CDMA interrupt enable
31-17	Reserved

Master Enable Register

This is a 2-bit, read / write register. The two bits are mapped to the two least significant bits of the location. The least significant bit contains the Master Enable bit and the next bit contains the Hardware Interrupt Enable bit. Writing a ‘1’ to the Master Enable bit enables the IRQ output signal. Writing a ‘0’ to the Master Enable bit disables the IRQ output, effectively masking all interrupt inputs. The Hardware Interrupt Enable bit is a write-once bit. At reset, this bit is reset to ‘0’, allowing the software to write to the Interrupt Status register to generate interrupts for testing purposes, and disabling any hardware interrupt inputs. Writing a ‘1’ to this bit enables the hardware interrupts input and disables software generated inputs. Writing a ‘1’ also disables any further changes to this bit until the device has been reset. Writing 1s or 0s to any other bit location does nothing. When read, this register will reflect the state of the Master Enable and Hardware Interrupt Enable bits. All other bits will read as 0s.

Table 28 Master Enable Register (Read/Write) - (BAR0 + 0x201C)

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

PCIe AXI Bridge Control

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

Table 29 PCIe AXI Bridge Control Registers

BAR0 Base Addr+	Bit(s)	Description
0x0000_1000→ 0x0000_1140	31:0	See Xilinx pg055 Memory Map
0x0000_1144	31:0	PHY Status/Control Register
0x0000_1148→ 0x0000_1204	31:0	See Xilinx pg055 Memory Map (These registers are not used)
0x0000_1208	31:0	Address Translation Register Upper AXIBAR2PCIEBAR_0U
0x0000_120C	31:0	Address Translation Register Lower AXIBAR2PCIEBAR_0L
0x0000_1210	31:0	Address Translation Register Upper AXIBAR2PCIEBAR_1U
0x0000_1214	31:0	Address Translation Register Lower AXIBAR2PCIEBAR_1L

BAR0 Base Addr+	Bit(s)	Description
0x0000_1210→ 0x0000_1FFF	31:0	See Xilinx pg055 Memory Map

PHY Status/Control Register

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

Table 30 PHY Status/Control Register (Read/Write) - (BAR0 + 0x1144)

Bit(s)	Function	
0	Reports the current link rate.	
	0	2.5 GT/s
	1	5.0 GT/s
2-1	Reports the current link width.	
	00	x1
	01	x2
	10	x4
	11	x8
8-3	Reports the current Link Training and Status State Machine state. Encoding is specific to the underlying Integrated Block.	
	x	
	x	
10-9	Reports the current lane reversal mode.	
	00	No reversal
	01	Lanes 1:0 reversed
	10	Lanes 3:0 reversed
	11	Lanes 7:0 reversed
11	Reports the current PHY Link-up state.	
	0	Link down
	1	Link up
15-12	Reserved	
31-16	See Xilinx pg055 PHY Status/Control Register	

AXI Base Address Translation Configuration Register

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

These registers are needed for DMA transfers that move data to the system memory buffer. Two AXI to Host address translation BARs are provided to support scatter-gather DMA operation: AXIBAR2PCIEBAR_0 and

AXIBAR2PCIEBAR_1. It is expected that two host memory regions will be used, one which contains the scatter-gather descriptor list, and the other which contains D/A samples. The memory region holding the scatter-gather descriptor list must be physically contiguous. The memory region holding D/A samples is not required to be contiguous. The location of the system memory buffer is loaded into these registers. For data transfers that cross non-contiguous page boundaries, the scatter-gather descriptor list must include updates to the AXI to Host address translation BARs for each non-contiguous region.

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

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

Example C code:

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

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

iStatus = PCIe7A_DmaGetBuffPhysAddress(iHandle, &u64PhyAddr);

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

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

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

Flash Memory

The Serial flash memory provides 4M bytes of non-volatile memory for storing the FPGA configuration bitstream. The lower portion of the flash address space is used to store the FPGA configuration bitstream. The upper portion is available for general non-volatile / program storage. The serial flash is accessible through the AXI Quad SPI interface. The flash device is Cypress Semiconductor part number S25FL132K0XMFB040.

Table 31 Flash Memory Map (Read/Write)

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

AXI XADC Analog to Digital Converter (System Monitor)

The XADC Analog to Digital Converter is used to monitor the die temperature and supply voltages of the FGPA. The XADC channel sequencer is configured to continuously sample the temperature, Vccint and Vccaux channels. The results from the A/D conversions can be read at the addresses given in column one of Table 32.

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

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

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

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

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

Additional information regarding the XADC can be found in the Xilinx XADC product guide PG019 and the user guide UG480

Table 32 System Monitor Register Map – (BAR0 + 0x32xx)

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

The expected values for V_{CCINT}, $V_{CCINT,}$ and Temperature are shown in Table 33.

Table 33 Expected Operating Parameters

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

DAC Registers

This section describes the AD5721/61 interface shown in Figure 3 AD5721/61 D/A Converter Interface Block Diagram that connects the AXI interface to the sixteen D/A converters and the sample memory that is shared among all sixteen channels. Each of the channels can be independently configured to operate in one of four modes: direct access, single burst, continuous, or FIFO mode. In either single burst, continuous, or FIFO mode, a channel sequencer retrieves samples from the sample memory and writes the sample data to its D/A converter. The portion of sample memory allocated to each channel is programmable. There are six channel specific registers per channel and five registers that affect all channels.

Table 34 AD5721/61 DAC Interface Register Map - (BAR0 + 0x40000)

BAR0 Base Addr +	Bits	Register Description
0x0004_0000	31:0	Channel 0 Start Address
0x0004_0004	31:0	Channel 0 End Address
0x0004_0008	31:0	Channel 0 FIFO
0x0004_000C	31:0	reserved
0x0004_0010	31:0	Channel 0 Control
0x0004_0014	31:0	Channel 0 Status
0x0004_0018	31:0	Channel 0 D/A Direct Access
0x0004_001C	31:0	reserved
0x0004_0020	31:0	**Channel 1 Start Address**
0x0004_0024	31:0	**Channel 1 End Address**
0x0004_0028	31:0	**Channel 1 FIFO**
0x0004_002C	31:0	reserved
0x0004_0030	31:0	**Channel 1 Control**
0x0004_0034	31:0	**Channel 1 Status**
0x0004_0038	31:0	**Channel 1 D/A Direct Access**
0x0004_003C	31:0	reserved
⋮	⋮	⋮
0x0004_01E0	31:0	**Channel 15 Start Address**

BAR0 Base Addr +	Bits	Register Description
0x0004_01E4	31:0	**Channel 15 End Address**
0x0004_01E8	31:0	**Channel 15 FIFO**
0x0004_01EC	31:0	reserved
0x0004_01F0	31:0	**Channel 15 Control**
0x0004_01F4	31:0	**Channel 15 Statu**s
0x0004_01F8	31:0	**Channel 15 D/A Direct Access**
0x0004_01FC	31:0	reserved
0x0004_0200	31:0	Control
0x0004_0204	31:0	Timer Divider
0x0004_0208	31:0	Software Trigger
0x0004_020C	31:0	reserved
⋮	⋮	⋮
0x0005_FFFe	31:0	reserved
0x0006 0000	31:0	**Sample Memory**
⋮	⋮	⋮
0X0007 FFFE	31:0	Sample Memory

Addressing Sample Memory

The Sample Memory is a dual-port memory that provides buffer space for storage of waveforms. It is organized as 64K 16-bit values. To meet varying application requirements, the portion of the sample memory allocated to each channel is configurable. D/A channels that output a static value, or channels that are not used, do not require any of the sample memory. Channels that output a repetitive waveform at a lower frequency require a larger portion of sample memory than a channel with higher frequency output. Channels that output a non-repetitive waveform will require a continuous stream of data from the host. For these channels, a portion of the sample memory may be configured as a FIFO to buffer the data stream. The host processor can directly read and write the sample memory concurrent with the channel sequencer reads. The address offset used by the host processor to access sample memory is a byte address relative to PCI BAR0. The addresses written into the DAC address registers are word addresses relative to the start of host memory, i.e. the address of the first 16-bit sample in memory is 0, the address of the next 16-bit sample is 1. When a channel is configured in FIFO mode, the host processor writes sample data to the appropriate DAC FIFO register address. The channels sequencer will manage the internal FIFO write address pointer that will ultimately determine the location in sample memory where the data will be stored. This address will always be within the Start and End address range configured for the channel.

Channel X Start Address

For channels configured in Single Burst mode, Continuous mode or FIFO mode, this register contains the start address in sample memory of the waveform to be written to the D/A converter. Addressing is relative to the first location in sample memory. A write to the Start Address register will also initialize the hidden FIFO read and write pointers when the channel is configured in FIFO mode.

Table 35 Channel X Start Address Register - (0x00)

Bit(s)	Description
15 - 0	Start address
31 - 16	Reserved

Channel X End Address

For channels configured in Single Burst mode, Continuous mode, or FIFO mode, this register contains the end address in sample memory of the waveform to be written to the D/A converter. Addressing is relative to the first location in sample memory.

Table 36 Channel X End Address Register - (0x04)

Bit(s)	Description
15 - 0	End address
31 - 16	Reserved

Channel X FIFO

For channels configured in FIFO mode, write to this address to add one or two samples to the FIFO. A 32-bit write operation will write two samples, a 16-bit write operation adds one sample. Use the key hole feature of the DMA controller when the FIFO is the destination of the DMA transfer.

Table 37 Channel X FIFO - (0x08)

Bit(s)	Description
15 - 0	Sample 0
31 - 16	Sample 1

Channel X Control Register

The channel specific control and status registers configure each channel's operating mode and trigger mode, and include a FIFO underflow flag clear function.

Table 38 Channel X Control Register - (0x10)

Bit(s)	Description
1 - 0	Operating mode: 00 – stopped 01 – continuous 10 – FIFO 11 – single burst
3 - 2	Trigger source: 00 – software 01 – timer 10 – external
31 - 4	Reserved

Channel X Status Register

The channel specific status registers indicate the mode dependent status of each channel.

Table 39 Channel X Status Register - (0x14)

Bit(s)	Description
0	FIFO empty (read only) 0 – FIFO not empty 1 – FIFO empty

Bit(s)	Description
1	FIFO half full (read only) 0 – FIFO is more than half full 1 – FIFO is half full or less
2	FIFO full (read only) 0 – FIFO is not full 1 – FIFO is full
3	FIFO underflow (write to reset) 0 – underflow has not occurred 1 – an attempt was made to read from an empty FIFO Write '1' to this bit to clear it.
4	Burst Single Complete – (write to reset) this flag indicates the last sample in the burst has been copied to the D/A data register 0 – burst in progress 1 – burst complete Write '1' to this bit to clear it.
5	Busy – (read only) when operating in direct access mode, this bit indicates that a serial transfer to the D/A converter is in progress 0 – not busy 1 – busy
31 – 6	Reserved

Channel X DAC Direct Access

A write to this address issues a command to the DAC. The contents of the DAC Channel registers are transferred to their corresponding converter input buffer serially. This serial data transfer takes 1µs. Thus, a new write of the same DAC register can be performed no sooner than 1µs after the previous write. The Channel X Status register includes a bit to indicate the busy status of the write operation. The channels Status busy bit will be set high upon initiation of a write operation and will remain high until the requested write operation has completed. New write accesses to the same DAC Channel register, should not be initiated unless its write busy status bit is low. Read of the DAC registers is not supported.

Table 40 Channel X DAC Direct Access Register - (0x18)

Bit(s)	Description	
15 - 0	voltage data or control data depending on address bits.	
19 - 16	Address 4-bits	
	0000	No Operation
	0001	**Write to input register** (no DAC output update, input register only written) a trigger is needed to transfer the data from the input register to the DAC register
	0010	**Update DAC register from input register** (Updates DAC output voltage)
	0011	**Write and Update DAC register** (Updates the input register and DAC register without waiting for a trigger)
	0100	**Write to control register** (see control register bit assignment below)
	0101	No Operation
	0110	No Operation
	0111	**Software Data Reset** (Reset to zero scale, midscale, or full scale as specified by PV1 and PV0 bits of control register)
	1000	Reserved
	1001	**Disable daisy-chain functionality** (executing this command tri-states unused SDO pin to save power)
	1010	Not supported (Read input register)
	1011	Not supported (Read DAC register)
	1101	No operation
	1110	No operation
	1111	**Software full reset** (Device set to power up state, output at GND and output buffer is powered down)
20	0 (this bit must be zero)	
31 – 21	Reserved	

When the Write to Control register address is selected (0100) bits 15 – 0 are control register bits as described below.

Bit(s)	Description	
2 - 0	Output Range (Software full reset is also issued when the output range is reconfigured.)	
	000	-10V to +10V
	001	0V to +10V
	010	-5V to +5V
	011	0V to +5V
	100	-2.5V to +7.5V
	101	-3V to +3V
	110	0V to 16V (external power supply required, contact Acromag)
	111	0V to 20V (external power supply required, contact Acromag)
4 – 3	Power-up Voltage	
	00	Zero scale
	01	Mid scale
	10	Full scale
	11	Full scale
5	0	Fixed at 0
6	Thermal shutdown alert	
	0	Die temperature > 150C do not power down
	1	Die temperature > 150C power down
7	Data Format Control Bit When a channel is configured for 0-10 or 0-5 ranges, this bit is ignored and anything written to the DAC is treated as straight binary. See Table 41 for example corresponding codes.	
	0	Straight binary coded or Bipolar Offset Binary coded
	1	Twos complement coded
8	5 % Over-range	

Bit(s)	Description	
	0	5% over-range disable
	1	5% over-range enable
10 – 9	Clear voltage selection	
	00	Zero scale
	01	Mid scale
	10	Full scale
	11	Full scale
15 – 11	Not Used	

Table 41 DAC Data Format

Straight Binary	Decimal Code	Twos Complement
1111	+7	0111
1110	+6	0110
1101	+5	0101
1100	+4	0100
1011	+3	0011
1010	+2	0010
1001	+1	0001
1000	0	0000
0111	-1	1111
0110	-2	1110
0101	-3	1101
0100	-4	1100
0011	-5	1011
0010	-6	1010
0001	-7	1001
0000	-8	1000

Control Register

This control register provides control functions that affect all channels.

Table 42 Control Register - (BAR0 + 0x40200)

Bit(s)	Description
0	Waveform Output Enable 0 – stop waveform output 1 – start waveform output
1	DAC reset 0 – DAC normal operation 1 – sets all outputs to ground, output buffers powered down
2	DAC clear 0 – DAC normal operation 1 – sets all DAC outputs to their pre-configured value (zero-scale, mid-scale, full-scale) setup for each channel
3	Trigger Direction 0 - accepts trigger input from an external source 1 – outputs trigger to external devices
4	Software Reset 0 – normal operation 1 – resets the registers in the AD5721 interface
31 – 5	Reserved

Timer Divider

This 32-bit register controls the period of the internal trigger. The timer is clocked at a rate of 31.25 MHz. The period of the internal trigger is calculated by multiplying the Timer Divider register contents by 32 nano-seconds. The minimum recommended value is 310. With this value the frequency of trigger is 100 KHz, which is the maximum frequency the DACs can operate at and still settle to within 1 LSB accuracy.

Table 43 Control Register - (BAR0 + 0x40204)

Bit(s)	Description
31 - 0	Timer Divider – count of 32 nS clock periods between internal triggers

Software Trigger

Write to this address to trigger the DAC outputs for all channels configured to respond to a software trigger.

Table 44 Software Trigger - (BAR0 + 0x40208)

Bit(s)	Description
31 -0	The data in this write operation is not used

Firmware Revision Register

This is a read only register located at BAR0 + 0x4000. The ASCII code representing the current revision of the MCS firmware file is readable from this location. For example, if the firmware is at revision A then this register will read 0x41 in the least significant byte or B= 0x42, C=0x43, etc.

AXI Quad SPI

The serial flash memory is accessed through the AXI Quad SPI interface. For a thorough understanding of the interface refer to the Xilinx product guide pg153 and the commands section of the flash memory device Cypress Semiconductor part number S25FL132K0XMFB040. Following is a description of the AXI Quad SPI registers. The AXI Quad SPI interface has been configured as standard mode (Quad message not supported), master mode enabled, no FIFOs, and the interrupt output is not connected. The clock frequency ratio is set at 2 resulting in a 31.25 MHz SPI clock. With the interrupt output unconnected, the Quad SPI IP Interrupt Status Register is still useful for reporting error conditions.

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

Quad SPI Software Reset Register

The Software Reset Register (SRR) permits resetting the core independently of other cores in the system. Writing 0x0000_000a to the SRR resets the core register for four AXI clock cycles. Any other write access generates undefined results and results in an error. The bit assignment in the software reset register is shown in and described in Table 45. Any attempt to read this register returns undefined data.

Table 45 Quad SPI Software Reset Register (BAR0 + 0x5040)

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

Quad SPI Control Register

The SPI Control Register (SPICR) allows programmer control over various aspects of the AXI Quad SPI core. The bit assignment in the SPICR is described in Table 46.

Table 46 Quad SPI Control Register (BAR0 + 0x5060)

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

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

Bit(s)	Quad SPI Control Register	
	1	1 = Master transactions disabled. Note: This bit immediately inhibits the transaction. Setting this bit while transfer is in progress would result in unpredictable outcome.
9	LSB first: This bit selects LSB first data transfer format. The default transfer format is MSB first. When set to:	
	0	MSB first transfer format.
	1	LSB first transfer format. Note: In Dual/Quad SPI mode, only the MSB first mode of the core is allowed.
31 - 10	Reserved	

Quad SPI Status Register

The Quad SPI Status Register (SPISR) is a read-only register that provides the status of some aspects of the AXI Quad SPI core to the programmer. The bit assignment in the SPISR is described in Table 47. Writing to the SPISR does not modify the register contents.

Table 47 Quad SPI Status Register (BAR0 + 0x5064)

Bit(s)	Quad SPI Status Register	
0	Receive Register Empty: This bit is set High when the receive register is empty.	
	0	Rx register Is not empty
	1	Rx register is empty
1	Receive Register Full: This bit is set High when the Receive Register is full.	
	0	Rx register not full
	1	Rx register full
2	Transmit Register empty: This bit is set High when the transmit register is empty. This bit goes High as soon as the TX register becomes empty.	
	0	Tx FIFO not empty
	1	Tx FIFO empty

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

Bit(s)	Quad SPI Status Register	
9	Loopback error flag	
	0	Default. The loopback bit in the control register is at default state.
	1	When the SPI command, address, and data bits are set to be transferred in other than standard SPI protocol mode and this bit is set in control register (SPICR).
10	Command error flag	
	0	Default
	1	The first entry in SPI DTR FIFO is incompatible with the command list for the attached flash device
31 - 11	Reserved	

Quad SPI Data Transmit Register

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

If a transfer is in progress, the data in the SPI DTR is loaded into the shift register as soon as the data in the shift register is transferred to the SPI DRR and a new transfer starts. The data is held in the SPI DTR until replaced by a subsequent write. The SPI DTR is described in Table 48.

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

Table 48 Quad SPI Data Transmit Register (BAR0 + 0x5068)

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

Quad SPI Data Receive Register

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

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

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

The power-on reset values for the SPI DRR are unknown. When known data has been written into the SPI DRR during core transactions, the data in this register can be considered for reading. The SPI DRR is described Table 49.

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

Table 49 Quad SPI Data Receive Register (BAR0 + 0x506C)

Bit(s)	Quad SPI Data Receive Register
7 - 0	SPI receive data
31 - 8	Reserved

Quad SPI Slave Select Register

The SPI Slave Select Register (SPISSR) contains an active-low, one-hot encoded slave select vector SS of length one. The bit assignment in the SPISSR is described in Table 50.

Table 50 Quad SPI Slave Select Register (BAR0 + 0x5070)

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

Quad SPI IP Interrupt Status Register

The IP Interrupt Status Register (IPISR) collects all of the events. Bit assignments are described in Table 51. The interrupt register is a read/toggle-on-write register. Writing a 1 to a bit position within the register causes the corresponding bit to toggle. All register bits are cleared on reset.

Table 51 IP Interrupt Status Register (BAR0 + 0x5020)

Bit(s)	IP Interrupt Status Register
0	Mode-fault error. This interrupt is generated if the SS signal goes active while the SPI device is configured as a master. This bit is set immediately on SS going active.
1	Slave mode-fault error. This interrupt is generated if the SS signal goes active while the SPI device is configured as a slave, but is not enabled. This bit is set immediately on SS going active and continually set if SS is active and the device is not enabled.
2	Data transmit register empty. It is set when the last byte of data has been transferred out to the external flash memory. In master mode if this bit is set to 1, no more SPI transfers are permitted.
3	Data transmit register underrun. This bit is set at the end of a SPI element transfer by a one-clock period strobe to the interrupt register when data is requested from an empty transmit register by the SPI core logic to perform a SPI transfer. This can occur only when the SPI device is configured as a slave in standard SPI configuration and is enabled by the SPE bit as set. All zeros are loaded in the shift register and transmitted by the slave in an under-run condition.
4	Data receive register full. This bit is set at the end of the SPI element transfer, when the receive register has been filled by a one-clock period strobe to the interrupt register.
5	Data receive register overrun. This bit is set by a one-clock period strobe to the interrupt register when an attempt to write data to a full receive register is made by the SPI core logic to complete a SPI transfer. This can occur when the SPI device is in either master or slave mode (in standard SPI mode) or if the IP is configured in SPI master mode (dual or quad SPI mode).
6	Reserved
7	Slave select mode. The assertion of this bit is applicable only when the core is configured in slave mode in standard SPI configuration. This bit is set when the other SPI master core selects the core by asserting the slave select line. This bit is set by a one-clock period strobe to the interrupt register. Note: This bit is applicable only in standard SPI slave mode.
31 - 8	Reserved

Serial Flash Write/Read Example

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

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

Write 0x0000000A to address 0x00005040 to reset the interface

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

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

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

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

Read from 0x0000506C to empty the receive data register.

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

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

Read from 0x0000506C to empty the receive data register.

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

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

Read from 0x0000506C to empty the receive data register.

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

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

Read from 0x0000506C to empty the receive data register. The value should be 0x00.

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

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

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

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

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

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

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

Slot ID Register

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

Table 52 Slot ID Register (BAR0 +0x6000)

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

4. USE OF CALIBRATION DATA

Calibration data is provided in the form of calibration coefficients, so the user can adjust and improve the accuracy of the analog output voltage over the uncalibrated state. Each channel's unique offset and gain calibration coefficients are stored in Flash memory. The use of software calibration allows the elimination of hardware calibration potentiometers traditionally used in producing precision analog outputs. A comparison of the uncalibrated and software calibrated performance is shown to illustrate the importance of the software calibration.

Software calibration uses some fairly complex equations. Acromag provides software products (sold separately) to make communication with the board and calibration easy. It relieves you from having to turn the equations of the following sections into debugged software calibration code.

Uncalibrated Performance

The uncalibrated performance is affected by two primary error sources. These are the channel's offset and gain errors. The use of channel specific calibration coefficients to accurately adjust offset and gain is important because the worst case uncalibrated error can be significant (although the typical uncalibrated errors observed may be much less). The maximum uncalibrated error is summarized as follows.

AP235 Model

AD5761 @ -40°C to 85°C:
Linearity Error is +/- 0.003% FSR maximum (i.e. +/-2 LSB).
Offset Error is +/- 0.05% FSR (i.e. 20V SPAN) maximum.
Gain Error is +/- 0.1% FSR maximum.
Total Error +/- 0.153% FSR maximum (+/-98.5 LSB)

AP225 Model

AD5721 @ -40°C to 85°C:
Linearity Error is +/- 0.0122% FSR maximum (i.e. +/-0.5 LSB).
Offset Error is +/- 0.05% FSR (i.e. 20V SPAN) maximum.
Gain Error is +/- 0.1% FSR maximum.
Total Error +/- 0.1512% FSR maximum (+/-6.2 LSB)

Typically, each error component is much less than its maximum and all error components do not reinforce each other. Thus, typical errors are much less than that shown above.

Calibrated Performance

Accurate calibration of the AP235 or AP225 can be accomplished through software control by using calibration coefficients to adjust the analog output voltage. Unique calibration coefficients are stored in Flash memory as 1/16 LSB’s for each specific channel. Once retrieved, the channel's unique offset and gain coefficients can be used to correct the data value sent to the DAC channel to accurately generate the desired output voltage.

The maximum calibrated error combining the linearity and adjusted offset and gain errors:

AP235 Model

AD5761 @ -40°C to 85°C:
Linearity Error is +/-2 LSB
Offset Error is +/-0.0625 LSB
Gain Error is +/-0.0625 LSB
Total Error +/-2.125 LSB (+/- 0.0032% FSR) maximum

AP225 Model

AD5721 @ -40°C to 85°C:
Linearity Error is +/-0.5 LSB
Offset Error is +/-0.0625 LSB
Gain Error is +/-0.0625 LSB
Total Error +/-0.625 LSB (+/- 0.0152% FSR) maximum

Thus, correcting the value programmed to the DAC Channel Register using the stored calibration coefficients provides the means to obtain excellent accuracy.

Data is corrected using a couple of formulas. Equation (1) expresses the ideal relationship between the value (IdealCode) written to the 16-bit DAC to achieve a specified voltage within output range assuming Straight Binary (also called Bipolar Offset Binary) or 2's Complement data format (see Table 53 and 54).

Equation (1):

IdealCode = [IdealSlope × DesiredVoltage] + IdealZeroCode

Where,

Table 53 AP235 Ideal slope and zero values for supported ranges.

Range	***IdealSlope***	***IdealZeroCode Straight Binary(2's Comp)***
-10 to 10V	3276.8	32768 (0)
-5 to 5V	6553.6	32768 (0)
-3 to 3V	10922.67	32768 (0)
-2.5 to 7.5V	6553.6	16384 (-16384)
0 to 10V	6553.6	0 (-32768)
0 to 5V	13107.2	0 (-32768)

Table 54 AP225 Ideal slope and zero values for supported ranges

Range	***IdealSlope***	***IdealZeroCode Straight Binary(2's Comp)***
-10 to 10V	204.8	2048 (0)
-5 to 5V	409.6	2048 (0)
-3 to 3V	682.6	2048 (0)
-2.5 to 7.5V	409.6	1024 (-1024)
0 to 10V	409.6	0 (-2048)
0 to 5V	819.2	0 (-2048)

Using equation (1), one can determine the IdealCode for any desired voltage within the range. For example, if it is desired to output a voltage of +5 Volts for -10 to 10V range, equation (1) returns the result 49,152 for IdealCode for Model AP235. If this value is used to program the DAC output (following conversion to Hex 0xC000), the output value will approach +5 Volts to within the uncalibrated error. This will be acceptable for some applications.

For applications needing better accuracy, the software calibration coefficients should be used to correct the IdealCode into the CorrectedCode required to accurately produce the output voltage. This is illustrated in equation (2):

Equation (2) AP235 Model

$$\mathbf{CorrectedCode}{=}\left(\frac{\mathbf{GainCoef}}{\mathbf{65536{\times}16}}+\mathbf{1}\right)[\mathbf{IdealSlope{\times}DesiredVoltage}]+\mathbf{IdealZeroCode}+\frac{\mathbf{OffsetCoef}}{\mathbf{16}}$$

The GainCoef and OffsetCoef are stored and retrieved from Flash memory at the addresses shown in Table 55. Coefficients are unique to each of the 16 channels. The GainCoef and OffestCoef values are calculated using the following equations at room temperature and then stored in Flash memory. Note that the coefficients are stored as 2's complement to allow use of plus and minus values.

AP235 Model

GainCoef = 65536×16 $\left(\frac{ActualSlope}{IdealSlope} - 1\right)$

OffsetCoef = (ActualZeroCode $-$ IdealZeroCode) $\times$ 16

ActualSlope = $\left(\frac{Code2-Code1}{MeasuredV2-MeasuredV1}\right)$

ActualZeroCode = Code1 $-$ (ActualSlope $\times$ $MeasuredV1$)

Where:
Code1 = 655(0x28F hex)
Code2 = 64880 (0xFD70 hex)

Measured values (MeasuredV2 and MeasuredV1) are taken using data averaging**.**

Equation (3) AP225 Model

$$\mathbf{CorrectedCode}{=}\left(\frac{\mathbf{GainCoef}}{\mathbf{4096{\times}16}}+\mathbf{1}\right)[\mathbf{IdealSlope{\times}DesiredVoltage}]+\mathbf{IdealZeroCode}+\frac{\mathbf{OffsetCoef}}{\mathbf{16}}$$

AP225 Model

GainCoef = 4096×16 $\left(\frac{ActualSlope}{IdealSlope} - 1\right)$

OffsetCoef = (ActualZeroCode $-$ IdealZeroCode) $\times$ 16

ActualSlope = $\left(\frac{Code2-Code1}{MeasuredV2-MeasuredV1}\right)$

ActualZeroCode = Code1 $-$ (ActualSlope $\times$ $MeasuredV1$)

Where:
Code1 = 40 (0x28 hex)
Code2 = 4055 (0xFD7 hex)

Measured values (MeasuredV2 and MeasuredV1) are taken using data averaging.

Flash Memory Map

Calibration coefficients are stored in Flash memory starting at the second from the last sector SA1022. This corresponds to Flash address range of 0x3FE000 to 0x3FEFFF.

Channel 0 coefficient start at 0x3FE000 to 0x3FE017 as shown in Table 55.

Channel 1 coefficients start at 0x3FE100 to 0x3FE117.

Channel 2 coefficients start at 0x3FE200 to 0x3FE217.

Etc.

Channel 15 coefficients start at 0x3FEF00 to 0x3FEF17.

Note that the coefficients are stored as 2's complement to allow use of plus and minus values.

Table 55 Calibration Coefficient Map

Flash Address	Bits	Range	Description
0x3F E000	7:0	-10 to 10V	Channel 0 OffsetCoef LSB
0x3F E001	7:0	-10 to 10V	Channel 0 OffsetCoef MSB
0x3F E002	7:0	-10 to 10V	Channel 0 GainCoef LSB
0x3F E003	7:0	-10 to 10V	Channel 0 GainCoef MSB
0x3F E004	7:0	0 to 10V	Channel 0 OffsetCoef LSB
0x3F E005	7:0	0 to 10V	Channel 0 OffsetCoef MSB
0x3F E006	7:0	0 to 10V	Channel 0 GainCoef LSB
0x3F E007	7:0	0 to 10V	Channel 0 GainCoef MSB
0x3F E008	7:0	-5 to 5V	Channel 0 OffsetCoef LSB

0x3F E009	7:0	-5 to 5V	Channel 0 OffsetCoef MSB
0x3F E00A	7:0	-5 to 5V	Channel 0 GainCoef LSB
0x3F E00B	7:0	-5 to 5V	Channel 0 GainCoef MSB
0x3F E00C	7:0	0 to 5V	Channel 0 OffsetCoef LSB
0x3F E00D	7:0	0 to 5V	Channel 0 OffsetCoef MSB
0x3F E00E	7:0	0 to 5V	Channel 0 GainCoef LSB
0x3F E00F	7:0	0 to 5V	Channel 0 GainCoef MSB
0x3F E010	7:0	-2.5 to 7.5V	Channel 0 OffsetCoef LSB
0x3F E011	7:0	-2.5 to 7.5V	Channel 0 OffsetCoef MSB
0x3F E012	7:0	-2.5 to 7.5V	Channel 0 GainCoef LSB
0x3F E013	7:0	-2.5 to 7.5V	Channel 0 GainCoef MSB
0x3F E014	7:0	-3 to 3V	Channel 0 OffsetCoef LSB
0x3F E015	7:0	-3 to 3V	Channel 0 OffsetCoef MSB
0x3F E016	7:0	-3 to 3V	Channel 0 GainCoef LSB
0x3F E017	7:0	-3 to 3V	Channel 0 GainCoef MSB
0x3F E018	7:0	Reserved	Reserved
⋮	⋮	⋮	⋮
0x3F E0FF	7:0	Reserved	Reserved

The model number formatted as ASCII characters is written to the Flash memory at 0x3FEFF0 to indicate valid calibration data has been written to the Flash

Table 56 Model Number String

Flash Address	Bits	AP225 Model	AP235 Model
0x3F EFF0	7:0	A = 0x41	A = 0x41
0x3F EFF1	7:0	P = 0x50	P = 0x50
0x3F EFF2	7:0	2 = 0x32	2 = 0x32
0x3F EFF3	7:0	2 = 0x32	3 = 0x33
0x3F EFF4	7:0	5 = 0x35	5 = 0x35
0x3F EFF5	7:0	0 (null Character)	0 (null Character)
0x3F EFF6	7:0	Reserved	Reserved
⋮	⋮	⋮	⋮
0x3F EFFF	7:0	Reserved	Reserved

5. SUGGESTED PROGRAMMING FOR STREAMING MODE

Streaming mode is available to support output of non-repetitive waveforms. In this mode, the host processor must keep the DACs supplied with data. The host must manage the transfer of data from host memory to the AP2x5 module. If all sixteen channels were operating at the maximum frequency, the host would have to supply data at 3.2 M bytes/sec. The Central Direct Memory Access (CDMA) controller is included in the FPGA firmware to assist the host processor in moving data from host memory to the AP2x5 on-board sample memory. Each channel will be configured in FIFO mode to buffer the samples written by the host. The on-chip block memory will be used to store a scatter-gather descriptor chain list used by the CDMA controller. Host memory will hold sample data that has been scaled and corrected for gain and offset errors. For this example, it is assumed the host memory used to store the sample data is contiguous, which would allow the addresses in the AXI2PICeBARs to remain constant. This section will describe the suggested programming sequence to support streaming data. The management of the DMA controller will be done within an interrupt handler.

1. Start in a known state by writing the Control Register with the Software Reset bit set to logic '1'.
2. Reset the DACs by writing the Control Register with the DAC reset bit set to logic '1'.
3. Configure each DAC channel by writing to the appropriate Channel Direct Access register. Set the output range, power-up voltage, clear voltage, and data format.
4. Set the initial output voltage for all DACs by writing the Control Register with the DAC clear bit set to logic '1'. This will power up the DAC outputs. The DACs will output the voltage configured with the previous step.
5. Set the FIFO size for each channel by writing the Channel Start Address and Channel End Address registers. If all channels will be outputting data at the same frequency, make all the FIFOs equal size.
6. Configure the operating mode of each channel as FIFO mode, and set the appropriate bits to select the trigger source.
7. Initialize the DMA scatter-gather descriptor chain list in block RAM. Up to sixteen descriptors could be needed each time a transfer is initiated. All the host memory addresses written to the descriptors must take into consideration the address translation that is performed by the PCIe interface. The Next Descriptor Pointer field must be set for each of the sixteen descriptors. The destination address will be the appropriate Channel FIFO register. Set the bytes to transfer field in the descriptor to one half the size of the sample memory allocated to each channel. The source address will be a host memory address where the next set of sample data for each channel

is stored. Write zeroes to the Transfer Descriptor Status Word for each descriptor to indicate that it has not completed.

8. Reset the CDMA by writing the Reset bit in the CDMA Control Register.
9. Poll the CDMA control register until the Reset bit indicates reset not in progress.
10. Configure the CDMA by writing the CMDA Control Register with Tail Pointer Mode enabled, Scatter Gather Mode selected, Key Hole write enabled, and Cyclic BD Disabled.
11. Write the address of the first scatter-gather descriptor to the CDMA Current Descriptor Pointer Register.
12. Write the address of the descriptor set up for channel fifteen to the CDMA Tail Descriptor Pointer Register. This initiates the DMA transfers.
13. Poll the CDMA Status Register until the CDMA idle bit indicates the CMDA is in the idle state. Each of the FIFOs are now half full.
14. Write the Interrupt Enable register with 0xFFFF. This enables each DAC channel to generate interrupts. Since each channel is configured in FIFO mode, an interrupt will be generated when any of the channels' FIFOs becomes half full. Also, note the CDMA interrupt is not enabled.
15. Write the following fields of the Master Enable Register:
    Master IRQ Enable = 1
    Hardware Interrupt Enable = 1
16. Write the Waveform Output Enable bit in the Control Register to start waveform output. The DACs will output the data stored in their FIFOs at the rate of the trigger pulses.
17. Wait for an interrupt from the AP2x5 module.
18. Read the Interrupt Pending Register. Store the value read for later use in the DMA complete interrupt handler. For each DAC channel interrupt bit in the Interrupt Pending Register that is set to a logic '1' set up the scatter-gather descriptor to transfer up to one half of the channel's FIFO size.
19. For each DAC channel interrupt bit in the Interrupt Pending Register that is not set to a logic '1', remove the channel's descriptor from the scatter-gather chain.
20. Write the address of the scatter-gather descriptor of the first channel requiring service to the CDMA Current Descriptor Pointer Register.
21. Write the following fields of the CDMA control register:
    Scatter Gather Mode = 1
    Key Hole Write = 1
    Cyclic BD Enable = 0

Interrupt on Complete Interrupt Enable = 1
Interrupt on Delay Timer = 0
Interrupt on Error Interrupt Enable = 1
Interrupt Threshold Value = number of descriptors to transfer
Interrupt Delay Timeout = 0

22. Write 0x10000 to the Interrupt Enable Register. This disables all DAC channel interrupts and enables the CDMA interrupt.
23. Write the address of the descriptor of the last channel requiring service to the CMDA Tail Descriptor Pointer Register. This will initiate the DMA transfers.
24. Wait for an interrupt from the AP2x5 module.
25. Read the CMDA status register. Check for error bits that are set.
26. Write the Interrupt Acknowledge Register with the saved value from the Interrupt Pending Register from above. This will clear the interrupts for the channels that were serviced by the DMA transfer.
27. Write 0xFFFF to the Interrupt Enable Register to re-enable the DAC interrupts.

See the APSW-API-WIN, APSW-API-VXW, APSW-API-LNX (Windows, VxWorks and Linux) AcroPack sample software code which implement this Data Streaming mode example.

6. THEORY OF OPERATION

This section contains information regarding the design of the board. A description of the basic functionality of the circuitry used on the board is also provided. Refer to Figure 1 AP225 Block Diagram, Figure 2 AP235 Block Diagram, and **Error! Reference source not found.** as you review this material.

Figure 5 AP225 / AP235 FPGA Firmware Block Diagram

PCIe Interface

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

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

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

Clock Generation

All clocks are derived from the 100 MHz PCIe REF_CLK signal. The PCIe interface includes a PLL that generates a 62.5 MHz clock that is used to clock the bus interfaces. The 62.5 MHz clock is further divided to produce a 31.25 MHz clock that is used to clock the timer and the serial interface to the DACs.

32MB Serial Flash

The serial FLASH memory provides 32 megabytes of non-volatile storage that is used for FPGA configuration and data storage. The FLASH device is Spansion part number 25FL132F. The lower 16 megabytes of memory space are dedicated to storage of the FPGA bitstream. Calibration coefficients are stored in the upper half of the FLASH.

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

JTAG Port

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

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

System Monitor (XADC)

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

7. SERVICE AND REPAIR

SERVICE AND REPAIR ASSISTANCE

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

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

PRELIMINARY SERVICE PROCEDURE

Before beginning repair, be sure that all the procedures in Section 2, Preparation for Use, have been followed. Also, refer to the documentation of your carrier board to verify that it is correctly configured. Replacement of the carrier and/or AP module with one that is known to work correctly is a good technique to isolate a faulty board.

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

WHERE TO GET HELP

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

Go to the “Support” tab or your specific model ordering page.

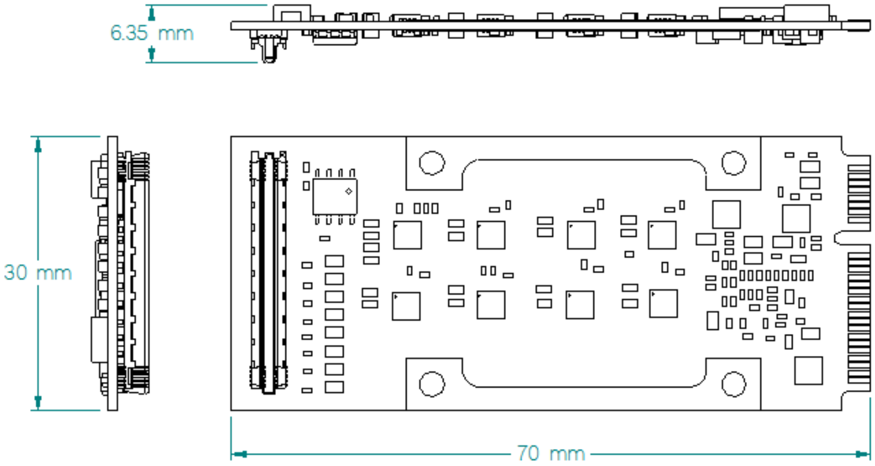
Acromag’s application engineers can also be contacted directly for technical assistance via email, telephone, or FAX through the contact information listed below. Note that an email question can also be submitted from within the Knowledge Base or directly from the “Contact Us” tab. When needed, complete repair services are also available.

- Email: solutions@acromag.com
- Phone: 248-295-0310

8. SPECIFICATIONS

PHYSICAL

Length	70.0 mm (3.150 in)
Width	30.0 mm (1.181 in)
Weight AP225	7.4 g
Weight AP235	7.4 g



POWER

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

AP225E-LF

+3.3 Volts	0.5 A typical, 1 A max.
+12 Volts	85 mA Typical, 275 mA maximum
-12 Volts	50 mA Typical, 200 mA maximum

AP235E-LF

+3.3 Volts	0.5 A typical, 1 A max.
+12 Volts	85 mA Typical, 275 mA maximum
-12 Volts	50mA Typical, 200 mA maximum

PCIe BUS COMPLIANCE

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

Table 57 PCIe Bus Data Rates

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

Note 1: PCIe x1 Gen 1 = 2.5GT/s (with 10-bit encoding we have a 20% loss in possible throughput due to encoding) giving 2.0 G bits/sec or 250M Bytes/sec.
Note 2: With PCIe we have a header for address and read/write command that is sent with every packet. This header is 20 Bytes with data payload of 4 Bytes (for our typical AcroPack). For each 4 Byte data sample, 24 Bytes are sent.
$\frac{250MByte/s}{24\,Bytes}$ = 10.4 M samples/sec or 41.6 M Bytes/sec or 0.332 G bit/sec
Note 3: For our typical AcroPack have measured back to back 4 Byte read operations completing every 1.7usec. A read operation starts with a host read request. The AcroPack must process the read and fetch the data and then generated the completion back to the host. The host then sends a message back that says I got it. This back and forth hand shaking slows down individual reads.
Note 4: DMA Read of 1024 sample takes 100us. Each sample is 4 Bytes. 100us/1024=0.0977us per sample or 4/0.0977us = 40.94Mbyte/s. We use DMA transfers to improve data transfers on the AP341/2 and AP225/235.
Note 5: Simple write operations are just as fast as DMA read operations. Write data is presented to the AcroPack in one transaction. Measured 4-byte back to back write accesses taking place every 100ns.

ENVIRONMENTAL

Operating Temperature .... AP2x5 -40°C to + 75 °C with 650 LFM airflow. Applications requiring operating temperatures above 70°C will require purchase of AcroPack Heatsink Accessory **AP-CC-01.**

Relative Humidity.......................... 5-95% non-condensing

Storage Temperature.................... -55 to +125°C

Non-Isolated................................. The PCIe bus and the AcroPack module commons have a direct electrical connection. As such, the user I/O signals are not isolated from the PCIe bus for these products.

EMC Directives .... AcroPack complies with EMC Directive 2004/108/EC.

Immunity per EN 61000-6-2 .... This is a Class A product
Electrostatic Discharge Immunity (ESD)
per IEC 61000-4-2.
Radiated Field Immunity (RFI)....... per IEC 61000-4-3.
Electrical Fast Transient Immunity (EFT)
per IEC 61000-4-4
Surge Immunity............................ per IEC 61000-4-5.
Conducted RF Immunity (CRFI)..... per IEC 61000-4-6.

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

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

RELIABILITY PREDICTION

AP225-16E

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

Temperature	MTBF (Hours)	MTBF (Years)	Failure Rate (FIT[1])
25°C	1,428,074	163.0	700.2
40°C	803,600	91.7	1,244.4

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

AP235-16E

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

Temperature	MTBF (Hours)	MTBF (Years)	Failure Rate (FIT[1])
25°C	2,313,653	264.1	432.2
40°C	1,257,145	143.5	795.5

[1] FIT is Failures in 109 hours.

Certificate of Volatility

Certificate of Volatility				
Acromag Models: AP225E-LF AP235E-LF	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.) FPGA based RAM	Size: 75 x 36-Kbit	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: 32 Meg x 1bit	User Modifiable ■ Yes □ No	Function: Storage of Code for FPGA	Process to Sanitize: Clear Flash memory by erasing all sectors of the Flash
Type (EEPROM, Flash, etc.) One Time Programmable area in flash device	Size: 3 x 256-byte	User Modifiable □ Yes ■ No	Function: The OTP area has been disabled by writing the lock registers with zeroes.	Process to Sanitize: Not applicable
Acromag Representative				
Name: Russ Nieves	Title: Dir. of Sales and Marketing	Email: rnieves@acromag.com	Office Phone: 248-295-0823	Office Fax: 248-624-9234

9. REVISION HISTORY

Release Date	Version	EGR/DOC	Description of Revision
14 FEB 2018	A	LMP/MJO	Initial release.
04 DEC 2020	B	ENZ/AMM	Inserted MTBF Table FOR AP225 and AP235.