PMC-SOFTDAC-M

16 Channels 500KSPS 16 Bit D/A
WAVEFORM GENERATOR

REFERENCE MANUAL

827-10-000-4000
Version 1.0
September 2006

ALPHI TECHNOLOGY CORPORATION

6202 S. Maple Avenue #120
Tempe, AZ 85283 USA
Tel: (480) 838-2428
Fax: (480) 838-4477

NOTICE

The information in this document has been carefully checked and is believed to be entirely reliable. While all reasonable efforts to ensure accuracy have been taken in the preparation of this manual, ALPHI TECHNOLOGY assumes no responsibility resulting from omissions or errors in this manual, or from the use of information contain herein.

ALPHI TECHNOLOGY reserves the right to make any changes, without notice, to this or any of ALPHI TECHNOLOGY's products to improve reliability, performance, function or design.

ALPHI TECHNOLOGY does not assume any liability arising out of the application or use of any product or circuit described herein; nor does ALPHI TECHNOLOGY convey any license under its patent rights or the rights of others.

Part Number : 827-10-000-4000

TABLE OF CONTENTS

Part Number : 827-10-000-4000

1. GENERAL DESCRIPTION

1.1 INTRODUCTION

The **PMC-SOFTDAC** is a high performance DIGITAL TO ANALOG module. The **PMC-SOFTDAC** outputs 16 channels with a 16-bit resolution at a maximum settling time of 2 μS.

The primary features of the **PMC-SOFTDAC** are as follows:

- 2 μSecond settling time (0 to 5 V)
- Six Programmable Output Ranges per channel
- Unipolar: 0V to 5V, 0V to 10V
- Bipolar Mode: ±5V, ±10V, ±2.5V,–2.5V to 7.5V
- 1LSB Max DNL and INL Over the Industrial Temperature Range
- Glitch Impulse < 2nV-s
- 16-Lead SSOP Package
- Power-On Reset to 0V
- Local 8kx8 Flash EPROM to store local user information
- Two stage buffers
- Global output buffer w/ internal or external triggering
- Automated state machine and buffer RAM to minimize impact to HOST
- Direct memory mapped access to the registers and buffer RAM
- Front panel I/O connectors for all outputs

1.2 FUNCTIONAL DESCRIPTION

1.3 ANALOG OUTPUT

The PMC-SOFTDAC uses 16 Linear LTC 1592 D/A converters.

The Linear LTC1592 are serial input 16-bit multiplying current output DACs that operates from a single 5 Volt supply. These SoftSpan DACs can be software-programmed for either uni-polar or bi-polar mode through a 3-wire SPI interface. In either mode, the voltage output range can also be software-programmed. Two output ranges in uni-polar mode and four output ranges in bi-polar mode are available.

The DACs are accurate to 1LSB over the industrial temperature range in both uni-polar and bi-polar modes. True 16-bit 4-quadrant multiplication is achieved with on-chip four quadrant multiplication resistors.

These devices include an internal deglitcher circuit that reduces the glitch impulse to less than 2nV-s (typ).

The output ranges are programmable using the board control register.

1.3.1 LTC1592 Command Structure

The LTC1592 receive serially a 24-bit input word. The 4 first bits are a command. The next 4 bits are unused. The last 16-bits are the value to be digitized.

The input word is send to one of the D/A when a write access takes place to the corresponding data register. The serialization logic on the SOFTDAC module use the 4 lower bits of the Command Register (0x48) as the 4 command bit, and the 16 bits being written as the value to be digitized.

It is not necessary to write into the command register between 2 data access. Whatever value was already there is used. Conversely, writing into the command register does not generate any access to the D/As.

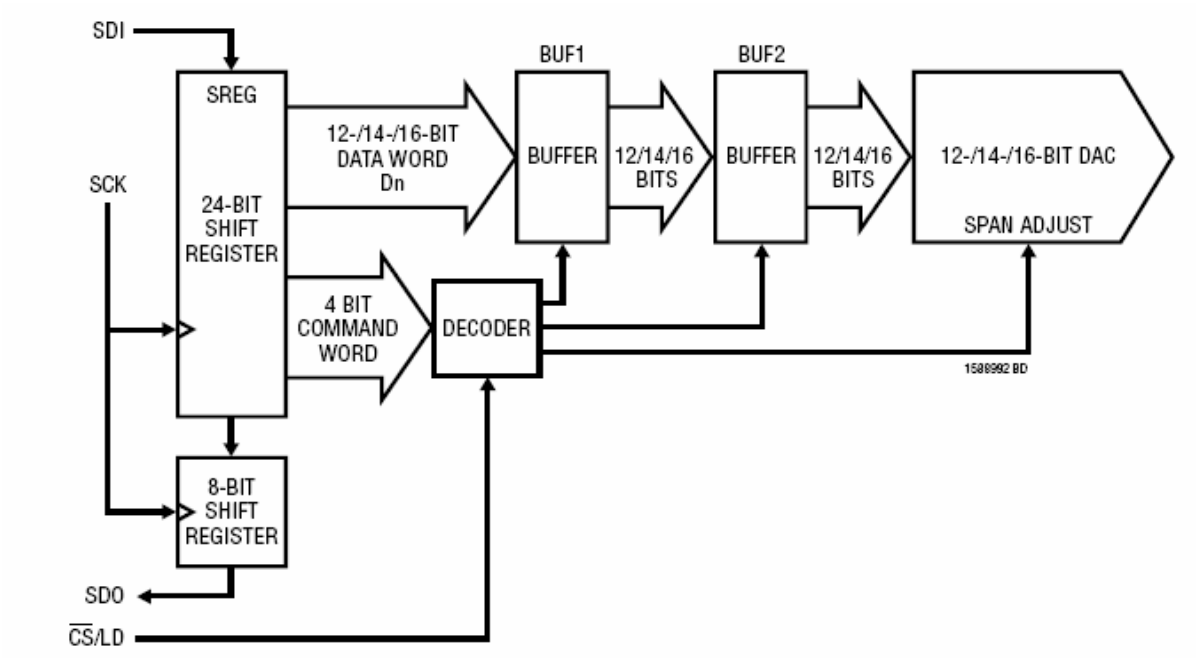

Figure 2.1: LTC1592 Block Diagram

Figure 1.1: Block Diagram

1.4 FUNCTIONAL DESCRIPTION

A functional block diagram of the **PMC-SOFTDAC-M** is presented below in Figure 1-1.

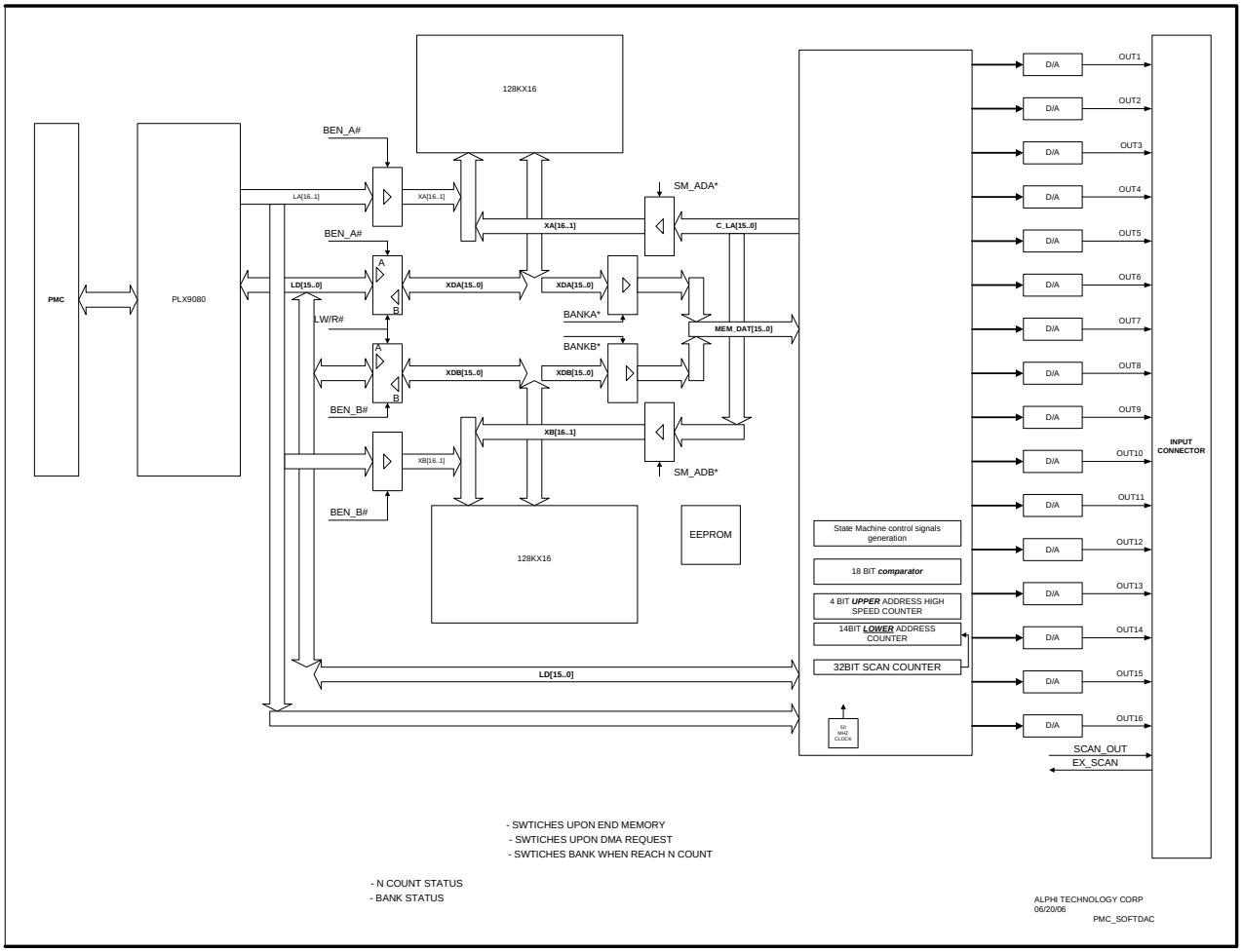

Figure 1.1: Block Diagram

The **PMC-SOFTDAC-M** operates as a slave that is managed by the host processor on the PMC bus. The card contains a state machine which performs most of the work of outputting data, in order to minimize the impact to the HOST system.

The **PMC-SOFTDAC-M** is supported by ALPHI Technology under *Windows XP* by a **Board Support Package** which is supplied with the card. Other documentation supplied with the card will describe this support in full detail.

Also Lab View example are provided

This is provided in a manner consistent across ALPHI Technology platforms.

2. HOST (PMC) SIDE

2.1 PCI CONFIGURATION REGISTERS

The card presents the following configuration values to the CPCI system, based on the values stored in the NVRAM device read by the AMCC PCI interface chip.

Register	Value (Meaning)
Vendor ID	0x13c5 (ALPHI Technology)
Device ID	0x0313 (PMC_SOFTDAC)
Revision ID	0x00
Class Code	0xff0000 (Device does not fit into defined class codes)
Interrupt Line	0xff
Interrupt Pin	A
Multifunction Device	No
Build In Self Test	No
Latency Timer	0x00
Minimum Grant	0x00
Maximum Latency	0x00
Expansion ROM Size	None

Table 2.1: PMC Configuration Registers

2.2 PMC BASE ADDRESS REGISTERS

The card requests base address regions from the PMC system after RESET, based on the values stored in the NVRAM device read by the PLX 9080 PMC interface chip.

The decode addresses of these regions are assigned by the host processor. The **PMC-SOFTDAC-M** uses 2 mapped base address registers. The PLX9080 is normally programmed at the factory to request the following resources from the PCI BIOS:

BAR	From	To	Description	Type
0	0x00000000	0x0000007F	PLX 9080 Operation Registers	MEM
2	0x00000000	0x0003FFFF	RAM A Region	MEM
2	0x00040000	0x0007FFFF	RAM B Region	MEM
2	0x00080000	0x000BFFFF	D/A and locals registers	MEM
2	0x000C0000	0x000FFFFF	EEPROM	MEM

Table 2.2: Base Addresses and Use

NOTE: The PLX9080 has been programmed to request memory above 1 Mbytes.

2.3 PMC OPERATION REGISTERS

2.3.1 The host processor communicates with the PMC-SOFTDAC-M **module via the PLX_9080 pass-through interface. After the base address registers have been programmed by the PMC configurator, incoming PMC I/O or Memory cycles are translated into either accesses to the PLX_9080 chip or passed through to the output registers as described in the next section. The PMC Operation Registers of the PLX_9080 chip are depicted below:**

NAME	PCI Address	31 – 24	23 – 16	15 – 8	7 – 0
PCIIDR	0x00	Device ID		Vendor ID	
PCICR	0x04	Status		Command	
PCISR	0x08	Class Code			Revision ID
PCIREV	0x0C	BIST	Header Type	PCI Latency Timer	Cache Line Size
PCICCR	0x10	PCI Base Address 0 (Memory Access to PLX Registers)			
PCICLSR	0x14	PCI Base Address 1 (I/O Access to PLX Registers)			
PCILTR	0x18	PCI Base Address 2 (Memory Access to DSP SRAM, IP IOSPACE, and other card registers)			
PCIHTR	0x1C	PCI Base Address 3 (Memory Access to DPR)			
PCIBISTR	0x20	Unused PCI Base Address 4			
PCIBAR0	0x24	Unused PCI Base Address 5			
PCIBAR1	0x28	Cardbus CIS Pointer (Not Supported)			
PCIBAR2	0x2C	Subsystem ID		Subsystem Vendor ID	
PCIBAR3	0x30	PCI Base Address for Expansion ROM			
PCIBAR4	0x34	Reserved			
PCIBAR5	0x38	Reserved			
PCICIS	0x3C	Max Latency	Min Grant	Interrupt Pin	Interrupt Line

Table 2.1: PCI Configuration Space

The card presents the following initial configuration values to the PCI system, based on the values stored in the NVRAM device read by the PLX PCI9080 interface chip.

Register	Value (Meaning)
Vendor ID	0x13C5 (ALPHI Technology)
Device ID	0x0313
Revision ID	0x00
Class Code	0xFF0000 (Device does not fit into defined class codes)
Interrupt Line	0xFF
Interrupt Pin	A
Multifunction Device	No
Build In Self Test	No
Latency Timer	0x00
Minimum Grant	0x00
Maximum Latency	0x00
Base Address 0 Size	0x100 Bytes (Memory Access to PLX Registers)
Base Address 1 Size	N/A
Base Address 2 Size	0x000000 – FFFFF(I/O Access and DPR)
Base Address 3 Size	N/A
Expansion ROM Size	None

Table 2.2: PCI Configuration Register Default Values

Table 2.2: PLX_9080 Registers (HOST)

For more information about these registers refer to the PLX_9080 PCI controller manual.

2.4 HOST CONTROL REGION

ADDRESS: $80000 - $BFFFF

Accesses to the following offsets from BAR2 will allow for communication with the card.

NAME	ADDR	DATA	R/W	COMMENTS
INT SAMP CLK	0x80000	DW	R/W	Divisor for Internal Sampling Clock
SM ADDRESS	0x80004	DW	RO	Current Address for the State Machine
LAST ADDR 0	0x80008	DW	R/W	Last Address with Valid Data, Bank 0
LAST ADDR 1	0x8000C	DW	R/W	Last Address with Valid Data, Bank 1
BANK 0 CTRL	0x80010	DWB	R/W	Controls Bank 0
BANK 1 CTRL	0x80011	DWB	R/W	Controls Bank 1
CTRL/STAT 0	0x80012	DWB	R/W	General control and status
CTRL/STAT 1	0x80013	DWB	R/W	General control and status
RESET SAMP CLK	0x80014	W	WS	Reset sampling clock counter
RESET ADDRESS	0x80016	W	WS	Reset address counter
RESET DACS	0x80018	W	WS	Reset all DACs

UPDATE DACS	0x8001a	W	WS	Update all DACs
SWITCH BANKS	0x8001c	W	WS	Switch active bank
DAC01	0x80020	DW	W	Direct DAC output
DAC02	0x80022	DW	W	Direct DAC output
DAC03	0x80024	DW	W	Direct DAC output
DAC04	0x80026	DW	W	Direct DAC output
DAC05	0x80028	DW	W	Direct DAC output
DAC06	0x8002a	DW	W	Direct DAC output
DAC07	0x8002c	DW	W	Direct DAC output
DAC08	0x8002e	DW	W	Direct DAC output
DAC09	0x80030	DW	W	Direct DAC output
DAC10	0x80032	DW	W	Direct DAC output
DAC11	0x80034	DW	W	Direct DAC output
DAC12	0x80036	DW	W	Direct DAC output
DAC13	0x80038	DW	W	Direct DAC output
DAC14	0x8003a	DW	W	Direct DAC output
DAC15	0x8003c	DW	W	Direct DAC output
DAC16	0x8003e	DW	W	Direct DAC output
COMMAND D/A REGISTER	0x80048	DW	R/W	Control output range and operation of each D/A.

Table 2.3: Host Control Region

2.4.1 INT SAMP CLK (Read / Write 32 bits)

ADDRESS: $80000

This register sets the sampling rate of the internal sampling rate generator. The internal sampling rate generator is based on a 36 MHz oscillator on the card. The sampling rate is set by the following formula where N is the contents of this register.

$$Samplingrate = \frac{36000000}{2 + N}$$

Since the maximum sampling rate supported by the DACs on the **PMC-SOFTDAC-M** is 500 KHz, the smallest value for *N* should be 70 ($46 hex).

This register can be accessed in WORD and DWORD modes.

2.4.2 SM ADDRESS (Read Only 24 bits)

ADDRESS: $80004

This register reports the current buffer address of the state machine. It can be cleared by writing to either **RESET ADDRESS** or **SWITCH BANKS**.

The current buffer address is the next offset into the active bank, which will be written to the DACs on the next sample clock. Although the address counter is 24 bits, only the lowest 13 bits are significant at this time.

This register can be accessed in WORD and DWORD modes.

2.4.3 LAST ADDR 0 / LAST ADDR 1 (Read / Write 24 bits)

ADDRESS: $80008 LAST ADDR 0

ADDRESS: $8000C LAST ADDR 1

These registers contain the last valid data point address when the card is operated in state machine mode. Each bank has its own individual last address.

Although these registers and the address counter are 24 bits, only the lowest 13 bits are significant at this time.

Note: If the last address register for the currently active bank is changed while the state machine is active, it is possible that the current address will pass the new last address. In this case, the current address register will continue to increment, and will wrap to 0 after up to 2^{24} sample clocks.

This register can be accessed in WORD and DWORD modes.

2.4.4 BANK 0 CTRL / BANK 1 CTRL (Read / Write 4 lower bits of a byte)

ADDRESS: $80010 BANK 0 CTRL

ADDRESS: $80011 BANK 1 CTRL

These registers can be accessed in BYTE, WORD and DWORD modes.

Byte access : $80010

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
				N/A	INT WHEN DONE	MODE 1	MODE 0

Byte access : $80011

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
				N/A	INT WHEN DONE	MODE 1	MODE 0

INT WHEN DONE Interrupt HOST when bank is done

When this bit is set to 1 and this bank has output its final value, the appropriate **BANK *N* DONE INT** bit is set to 1 and the HOST is interrupted. When this bit is cleared to 0, no interrupt is generated and no bits are set when the bank is done. This bit is cleared to 0 by a board RESET.

MODE 1, MODE 0 What to do when bank is done

The following table describes the available options when a bank is finished being output.

MODE 1	MODE 0	Description
0	0	Play this bank again from address 0 (Default at board RESET)
0	1	Switch to the other bank at address 0
1	0	Disable the state machine and end output at last value in bank
1	1	Disable the state machine, end output at last value in bank, and set the **UNDERFLOW** bit

Table 2.4: Options at Bank Completion

2.4.5 CTRL/STAT 0 (Read / Write 8 bits)

ADDRESS: $80012

This register can be accessed in BYTE, WORD and DWORD modes.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
Not used	Not used	ENABLE STATE MACH	UNDER FLOW	ENABLE EXT SAMP CLOCK	ENABLE INT SAMP CLOCK	ENABLE CLOCK OUTPUT	ACTIVE BANK

ENABLE STATE MACH Enables automatic output from the RAM buffers

When this bit is set to 1, the card will automatically reload the DAC registers from the output buffers at each sampling clock based on the current address and the active bank. When this bit is cleared to 0, no automatic updates will occur.

This bit is cleared to a 0 by a board RESET, and by a negative edge on **FP_RST**, if enabled.

UNDER FLOW Card is reporting an underflow state

This bit can potentially be used as a fault bit when the state machine is used for arbitrary output and the buffers are written alternately by an interrupt routine. If for some reason, the currently playing buffer reaches the end before the other buffer is updated and the mode bits updated to *switch banks* (from *stop output with underflow*), the output will stop and the underflow bit will be set.

The user can directly write this bit into either state.

This bit is cleared to a 0 by a board RESET.

ENABLE EXT SAMP CLOCK, ENABLE INT SAMP CLOCK

These bits determine the source of the sampling clock as demonstrated in the following table.

ENABLE EXT SAMP CLOCK	ENABLE INT SAMP CLOCK	SOURCE
X	1	Internal sampling clock generator

1	0	External sampling clock from **SAMPLING_CLOCK_IN**
0	0	Writes to **UPDATE DACS** generate sampling clock

Table 2.5: Sampling Clock Options

These bits are cleared to a 0 by a board RESET.

ENABLE CLOCK OUTPUT Enables output of selected sampling clock

When this bit is set to a 1, the sampling clock source as determined from the above table is output on the SAMPLING_CLOCK_OUT line of the front panel connector. When this bit is cleared to 0, no sampling clock is output.

This bit is cleared to a 0 by a board RESET.

ACTIVE BANK (Read Only) Reports active bank

This read only bit reports the current active bank for the state machine. The current active bank can be changed by the state machine switching banks automatically upon reaching the end of the bank, or by the user writing to **SWITCH BANKS**.

2.4.6 CTRL/STAT 1 (Read / Write 8 bits)

ADDRESS: $80013

This register can be accessed in BYTE, WORD and DWORD modes.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
Not used	INT SAMPLE CLOCK	INT BANK 1 DONE	INT BANK 0 DONE	N/A	ENABLE SAMPLE CLK INT	Not used	Not used

INT SAMPLE CLOCK Interrupt caused by sampling clock

When set, this bit indicates that an interrupt was generated by the sampling clock triggering an update of the DAC outputs. The user can use this interrupt to write the next outputs to the DAC holding registers prior to the next sample clock.

This bit is cleared by writing a 1 to it. Writing a 0 will not affect the state of this bit.

INT BANK 1 DONE Interrupt caused by Bank 1 completing

When set, this bit indicates that an interrupt was generated by the state machine writing the last value from Bank 1 to the DAC holding registers. The user can use this interrupt to fill Bank 1 with the next outputs when the card is operated alternating between both banks.

This bit is cleared by writing a 1 to it. Writing a 0 will not affect the state of this bit.

INT BANK 0 DONE Interrupt caused by Bank 0 completing

When set, this bit indicates that an interrupt was generated by the state machine writing the last value from Bank 0 to the DAC holding registers. The user can use this interrupt to fill Bank 0 with the next outputs when the card is operated alternating between both banks.

This bit is cleared by writing a 1 to it. Writing a 0 will not affect the state of this bit.

ENABLE SAMPLE CLK INT Enable interrupt on sampling clock

When this bit is set to 1, an interrupt is generated during each sampling clock. When this bit is cleared to 0, no interrupt is generated.

Note that interrupt latency issues restrict the use of this interrupt to fairly slow sampling clock frequencies.

2.4.7 RESET SAMP CLK (Write Strobe)

ADDRESS: $80014

Writing to this location will force the internal sampling clock generator to reload the count from the **INT SAMP CLK** register.

2.4.8 RESET ADDRESS (Write Strobe)

ADDRESS: $80016

Writing to this location will clear the **SM ADDRESS** counter.

NOTE: If this is written while the state machine is active, there will be a phase jump in the output, and additionally, if the switch occurs in the middle of an update, then some outputs may be updated from the old location, and some from the new location for one sample clock.

2.4.9 RESET DACS (Write Strobe)

ADDRESS: $80018

Writing to this location will clear DAC holding registers, and force the DACs to output 0 volts.

2.4.10 UPDATE DACS (Write Strobe)

ADDRESS: $8001A

Writing to this location will generate a manual sample clock pulse. It can be used in a full manual mode to update all DAC outputs simultaneously.

2.4.11 SWITCH BANKS (Write Strobe)

ADDRESS: $8001C

Writing to this location will clear the **SM ADDRESS** counter and switch the active bank.

NOTE: If this is written while the state machine is active, there will be a phase jump in the output, and additionally, if the switch occurs in the middle of an update, then some outputs may be updated from the old bank, and some from the new bank for one sample clock.

2.4.12 DAC (Write Only)

ADDRESS: $80020 - $8003E

These 16 registers will directly write into the holding registers of the DACs. Writes can also force an update of the DAC if the ***AUTO UPDATE DAC*** bit is set. Otherwise, the update will occur on the next sampling clock.

2.4.13 Command Register (Read/Write)

ADDRESS: $80048

Writing in a D/A data register will update the corresponding analog output. Along the data, the content of the command register will be sent to the D/A.

The command register contains the output range information, to allow for different ranges in different channels, either the command register will need to be updated between the data writes, or, once all the channels have been programmed with the proper range.

Each D/A converter can be programmed for different range.

First the Command register is loaded with the desired value, then a write to the selected D/A will send the information to the D/A.

2.4.13.1 Global Update

It is possible to update all the D/A with the same Command Register value by setting bit # 4 to a “1”. Therefore a sole write to any D/A will transfer the Command Register value to all the D/A.

Example: Setting all the D/A output to +/-10v, then immediate transfer of next data to the output.

-write $1B to address $80048

-write any data to D/A #1 at address $80020

2.4.13.2 Immediate Mode Operation

-write $02 to address $80048

The Command Register value 0x02 allows sending data without range information. It should always be used, in the Command Register, after the initial range programming is completed, particularly if the range selection is not the same among the channels.

COMMAND				OPERATION EACH COMMAND IS EXECUTED ON THE RISING EDGE OF CS/LD	Internal Register Status			
					SREG DATA WORD Dn IN INPUT SHIFT REGISTER	BUF1 INPUT BUFFER	BUF2 DAC BUFFER (DAC OUTPUT)	DAC OUTPUT RANGE
C3	C2	C1	C0					
0	0	0	0	Copy Dn in SReg to Buf1. Does not change range.	Dn	Dn	No Change	No Change
0	0	0	1	Copy the Data in Buf1 to Buf2	X	Dn	Dn	No Change
0	0	1	0	Copy Dn in SReg to Buf1 and Buf2 Does not change range.	Dn	Dn	Dn	No Change
0	0	1	1	Reserved (Do Not Use)				
0	1	0	0	Reserved (Do Not Use)				
0	1	0	1	Reserved (Do Not Use)				
0	1	1	0	Reserved (Do Not Use)				
0	1	1	1	Reserved (Do Not Use)				
1	0	0	0	Set Range to 0 to 5V. Copy Dn in SReg to Buf1 and Buf2	Dn	Dn	Dn	5V
1	0	0	1	Set Range to 0 to 10V. Copy Dn in SReg to Buf1 and Buf2	Dn	Dn	Dn	10V
1	0	1	0	Set Range to ±5V. Copy Dn in SReg to Buf1 and Buf2	Dn	Dn	Dn	±5V
1	0	1	1	Set Range to ±10V. Copy Dn in SReg to Buf1 and Buf2	Dn	Dn	Dn	±10V
1	1	0	0	Set Range to ±2.5V. Copy Dn in SReg to Buf1 and Buf2	Dn	Dn	Dn	±2.5V
1	1	0	1	Set Range to -2.5V to 7V. Copy Dn in SReg to Buf1 and Buf2	Dn	Dn	Dn	-2.5V to 7.5V
1	1	1	0	Reserved (Do Not Use)				
1	1	1	1	No Operation	X	No Change	No Change	No Change

Data Word Dn (n = 0 to 15) is the last 16 bits shifted into the input shift register SReg that corresponds to the DAC code.

Table 2.2: LTC1592 Commands

2.5 RAM Buffer Region

There are 8192 output locations available for each DAC channel in each bank. The output buffers can be directly written and read by the HOST.

When the state machine is active, the currently outputting bank is not accessible, but the other bank is accessible. The current bank can be determined by a status bit in the control status register.

The RAM buffers are addressable in WORD and DWORD modes. Most HOST bridge chipsets will compact individual WORD writes into DWORD writes, which are more efficient on a CPCI bus without any additional effort by the customer.

Start	End	Bank	Channel	Start	End	Bank	Channel
0x00000	0x03FFF	0	DAC01	0x40000	0x43FFF	1	DAC01
0x04000	0x07FFF	0	DAC02	0x44000	0x47FFF	1	DAC02
0x08000	0x0BFFF	0	DAC03	0x48000	0x4BFFF	1	DAC03
0x0C000	0x0FFFF	0	DAC04	0x4C000	0x4FFFF	1	DAC04
0x10000	0x13FFF	0	DAC05	0x50000	0x53FFF	1	DAC05
0x14000	0x17FFF	0	DAC06	0x54000	0x57FFF	1	DAC06
0x18000	0x1BFFF	0	DAC07	0x58000	0x5BFFF	1	DAC07
0x1C000	0x1FFFF	0	DAC08	0x5C000	0x5FFFF	1	DAC08
0x20000	0x23FFF	0	DAC09	0x60000	0x63FFF	1	DAC09
0x24000	0x27FFF	0	DAC10	0x64000	0x67FFF	1	DAC10
0x28000	0x2BFFF	0	DAC11	0x68000	0x6BFFF	1	DAC11
0x2C000	0x2FFFF	0	DAC12	0x6C000	0x6FFFF	1	DAC12
0x30000	0x33FFF	0	DAC13	0x70000	0x73FFF	1	DAC13
0x34000	0x37FFF	0	DAC14	0x74000	0x77FFF	1	DAC14
0x38000	0x3BFFF	0	DAC15	0x78000	0x7BFFF	1	DAC15
0x3C000	0x3FFFF	0	DAC16	0x7C000	0x7FFFF	1	DAC16

Table 2.6: RAM Buffer Locations

2.6 INTERRUPTION

The different sources of interruption described above are sent to the Host by setting low the LINTI# line of the PLX9080.

The interrupt will stay active until the host removes the source.

2.7 MODES OF OPERATION

The card can be viewed as operating in one of the following modes.

- State Machine providing Automatic Update and Load on Sampling Clock
- Manual Load with Update on Sampling Clock
- Manual Load and Update

2.7.1 State Machine providing Automatic Update and Load on Sampling Clock

The card contains a state machine capable of automatically loading the DAC holding registers from the RAM buffers on each sample clock. There are two banks of RAM buffer so that one can be updated by the HOST while the other is being output. Full interrupt support to the HOST allows for easy synchronization of the buffer updates.

Additionally, a single RAM buffer can be played out repeatedly until a change in signal is required, at which time the HOST can write the new waveform to the other buffer and set up a switch at the end of the previous one to ensure phase consistency. Separate length registers for each bank help to achieve this functionality.

On each sampling clock, the DACs are updated from the holding registers, then 16 values are read from the active buffer bank into the holding registers for the next sampling clock.

On the first sampling clock after the state machine is enabled, the DAC holding registers will contain zero if the DACs were RESET. The first data point will be output on the second sampling clock. When the state machine is disabled at the end of a bank, the actual last point is output one sampling clock later.

2.7.2 Algorithm for arbitrary output using both buffers

Starting Output

Ensure that the sampling clock is turned off, and that the state machine is turned off. Ensure that BANK 0 is active.

Hook HOST interrupt service routine to the card.

Program both bank length registers, and load the first data to be played into bank 0, and the second data into bank 1. Set bank 0 to switch banks at the end, and to generate interrupts. Set bank 1 to disable state machine and set underflow, and to generate interrupt.

Enable the state machine and the sampling clock. Continue with non-interrupt processing.

Interrupt Routine

Check underflow bit. If it is set, then interrupt latency would have caused gaps in the output. Check and clear the interrupt cause.

Write the next frame of data to the inactive bank.

Set inactive bank to disable state machine and set underflow and to generate interrupt, and set active bank to switch banks at the end and to generate interrupt. This can be done in one WORD write.

Return from interrupt.

Stopping Output

Disable the state machine and the sampling clock. Disconnect the interrupt routine.

2.7.3 Manual Load with Update on Sampling Clock

In this mode, the card will transfer the values in the DAC holding registers to the output on each internal or external sampling clock. A HOST interrupt is used to have the HOST load the next set of data to the DAC holding registers.

Obviously, this mode will make much greater demands of the HOST as it will be interrupted at every sample clock. Sample rates above a few kHz will not be possible due to the needs of the interrupt routine.

2.7.4 Manual Load and Update

This is a purely manual mode of operation, without making use of any timing on the part of the card. The HOST can write the desired values to the DAC holding registers and then write to **UPDATE DACS** to update all 16 DACs at the same time.

2.8 RESET SIGNALS

The **PMC-SOFTDAC-M** is reset when the PMC bus is reset, and when the reset bit of the PLX_9080 issues a board reset.

3. LED INDICATORS

There are four LED indicators visible on the solder side. They are marked with a legend on the board, and they are labeled on the PCB as LD1 to LD4. The LEDs have the following meanings:

LED	Meaning	Signal name
LD1	Bank B is being output.	SM_ADB#
LD2	Bank A is being output.	SM_ADA#
LD3		AEB
LD4	Bank B is being output.	BEN_B#
LD5	Bank A is being output.	BEN_A#
D1	HOST has written to register or RAM.	LHOLDA

Table 3.1 LED Descriptions

4. CONNECTIONS

4.1 ANALOG OUTPUT CONNECTOR (P3)

A 50 pin subminiature D shelled connector is used to route all the analog output signals off the card. ALPHI Technology can supply transition modules and cable assemblies to meet any special requirements. Contact the factory for additional information.

Connectors are manufactured by AMP.

Use	Model
On PC Board	787190-5
Suggested Plug	749111-4

Table 4.1: I/O Connector Model Numbers

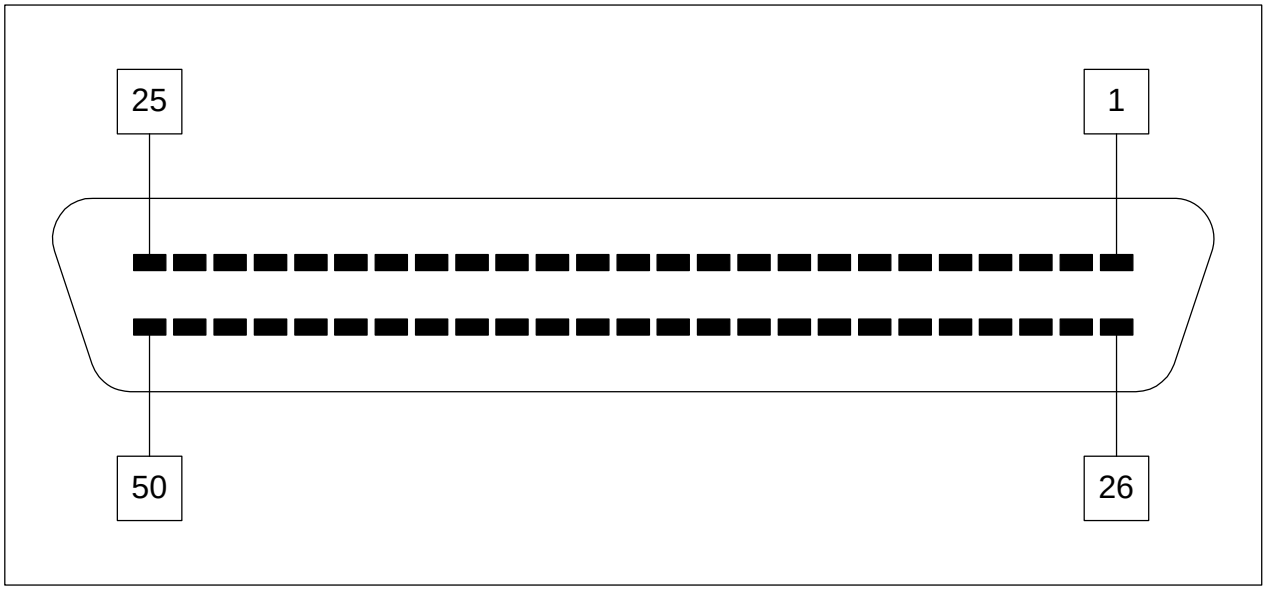

Figure 4.1: Analog Output Connector

Pin	Connection	Pin	Connection
1	OUT01	26	A_GND
2	OUT02	27	A_GND
3	OUT03	28	A_GND
4	OUT04	29	A_GND
5	OUT05	30	A_GND
6	OUT06	31	A_GND
7	OUT07	32	A_GND
8	OUT08	33	A_GND
9	OUT09	34	A_GND
10	OUT10	35	A_GND
11	OUT11	36	A_GND
12	OUT12	37	A_GND
13	OUT13	38	A_GND
14	OUT14	39	A_GND
15	OUT15	40	A_GND
16	OUT16	41	A_GND
17		42	
18		43	
19		44	
20		45	
21		46	
22		47	
23	SAMP_CLK_OUT	48	
24	D_GND	49	D_GND
25	SAMP_CLK_IN	50	

Table 4.2: Analog Output Connector

4.2 32 BIT PMC BUS (P11&P12)

This connector provides the standard PMC signals for all PMC Carrier systems.

4.3 FACTORY USE (P1)

This connector is used at the factory for programming the FPGA.