CPCI-DA16

16 Channel 16 Bit D/A
for 3U *CPCI*™ Systems

REFERENCE MANUAL

748-11-000-4000
Version 1.1
June 1998

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 : 748-11-000-4000

TABLE OF CONTENTS

1. GENERAL DESCRIPTION

1.1 INTRODUCTION

The **CPCI-DA16** is a CPCI expansion card in a 3U form factor. The **CPCI-DA16** allows:

- 16 analog outputs with 10 μS 16 bit converters at +/- 10 Volts
- Texas Instrument Part Number: DAC712UB
- Binary 2's Complement Format
- 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
- Low current outputs option: CPCI-DA16-1 (up to 40ma current drive, Low offset)
- High current outputs options CPCI-DA16-2 (up to 250ma current drive, Higher offset)

1.2 FUNCTIONAL DESCRIPTION

A functional block diagram of the **CPCI-DA16** is presented below in Figure 1-1.

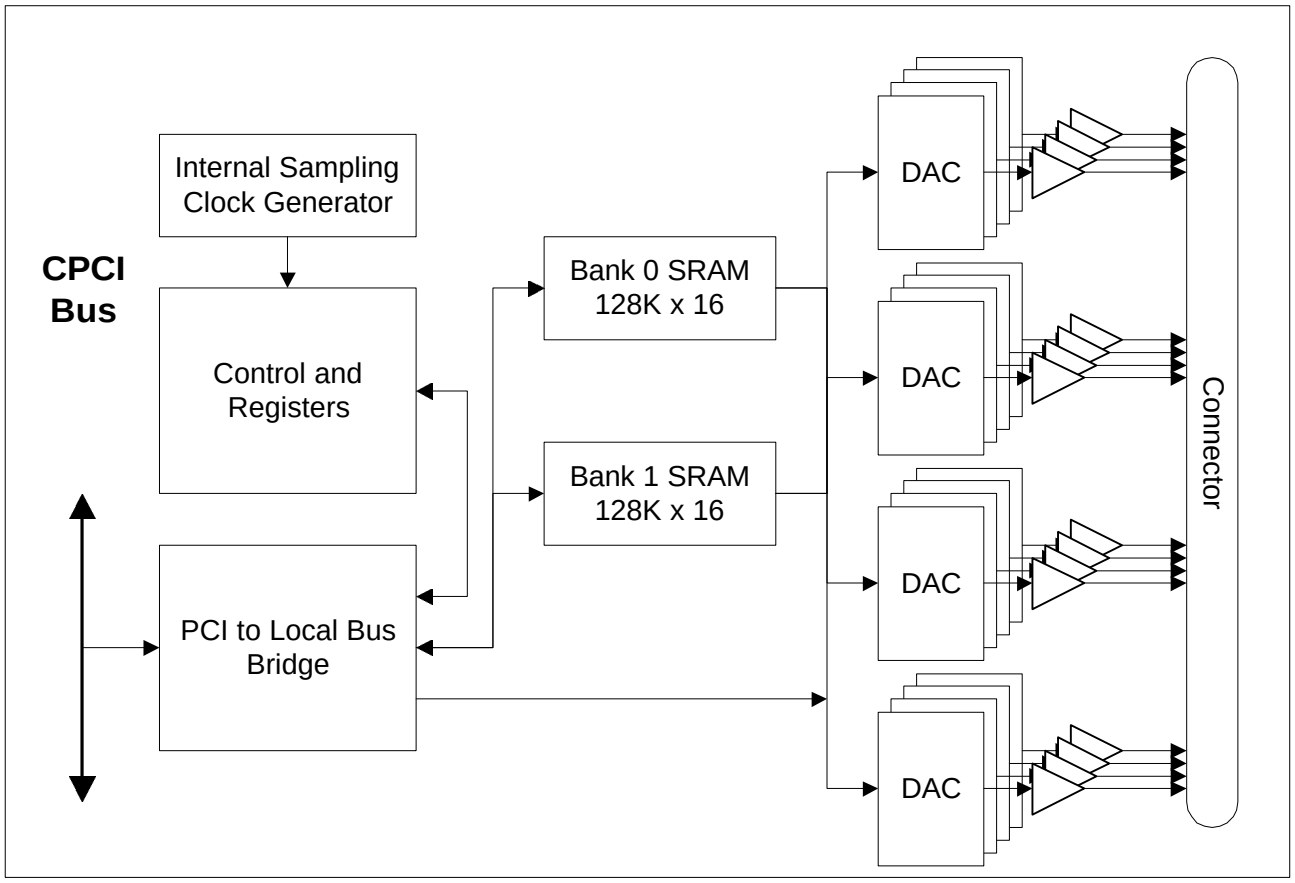

Figure 1.1: Block Diagram

The jumper and the connector placement is depicted in Figure 1-2.

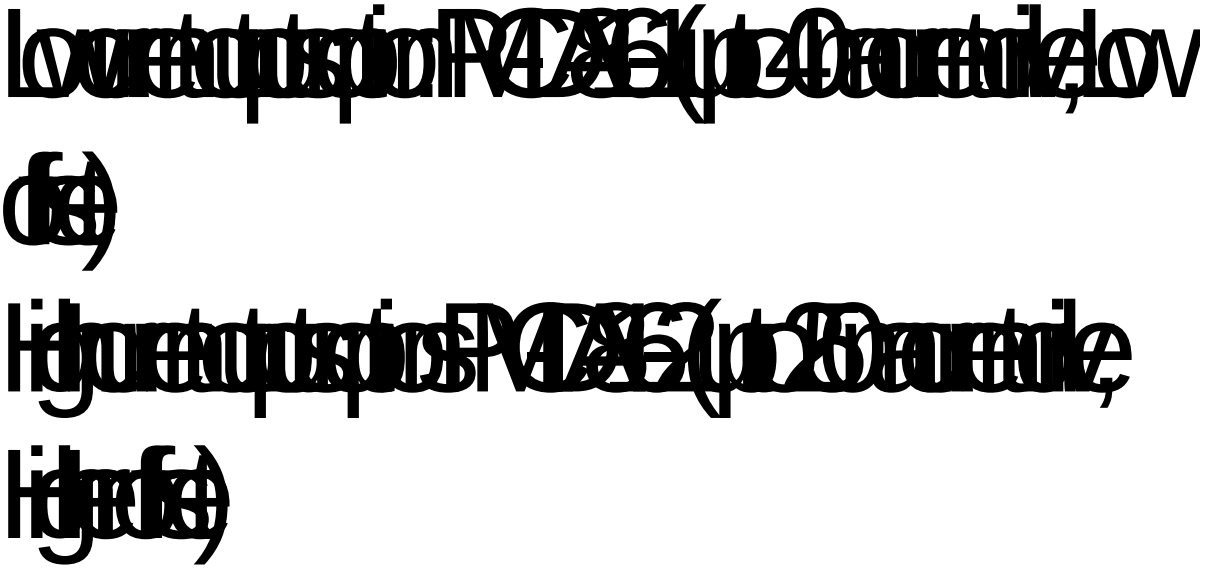

Figure 1.2: Jumper and Connector Locations

The **CPCI-DA16** operates as a slave that is managed by the host processor on the CPCI 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 **CPCI-DA16** is supported by ALPHI Technology under *Windows NT* by a **Board Support Package** which is supplied with the card. Other documentation supplied with the card will describe this support in full detail.

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

1.3 REFERENCE MATERIALS LIST

The reader should refer to the PCI Local Bus Specification for a detailed explanation of the PCI bus architecture and timing requirements. This specification is available from the following source:

PCI Special Interest Group
PO Box 14070
Portland, OR 97214
Tel: (800) 433-5177
Tel: (503) 797-4207
Fax: (503) 234-6762

The reader is also referred to the S5933 PCI Controller data book:

AMCC Applied Micro Circuits Corporation
6195 Lusk Boulevard
San Diego, CA 92121-2793
Tel: (800) 755-2622
http://www.amcc.com

WindowsNT and Windows95 Programming Tools:

BlueWater Systems
144 Railroad Ave. Suite #217
Edmonds, WA 98020
Tel: (206) 771-3610
Fax: (206) 771-2742
E-Mail: info@bluewatersystems.com
Web: http://www.bluewatersystems.com

Texas Instrument

For D/A (DAC712) specifications
Web: http://www.ti.com

2. HOST (CPCI) 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	0x010a (CPCI-DA16)
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: CPCI Configuration Registers

2.2 PCI BASE ADDRESS REGISTERS

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

The decode addresses of these regions are assigned by the host processor. The **CPCI-DA16** uses 3 of the 5 AMCC mapped base address registers. The AMCC is normally programmed at the factory to request the following resources from the CPCI BIOS:

BAR	From	To	Description	Type
0	0x00000000	0x0000003F	AMCC PCI Operation Registers	MEM
1	0x00000000	0x000000FF	Host Control Region	MEM
3	0x00000000	0x0007FFFF	RAM Buffer Region	MEM

Table 2.2: Base Addresses and Use

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

2.3 CPCI OPERATION REGISTERS

The host processor communicates with the **CPCI-DA16** module via the AMCC pass-through interface. After the base address registers have been programmed by the CPCI configurator, incoming CPCI I/O or Memory cycles are translated into either accesses to the AMCC chip or passed through to the output registers as described in the next section. The PCI Operation Registers of the AMCC 5933 chip are depicted below:

Offset	Register Name
0x00	OMB1 Outgoing Mailbox Register 1
0x04	OMB2 Outgoing Mailbox Register 2
0x08	OMB3 Outgoing Mailbox Register 3
0x0C	OMB4 Outgoing Mailbox Register 4
0x10	IMB1 Incoming Mailbox Register 1
0x14	IMB2 Incoming Mailbox Register 2
0x18	IMB3 Incoming Mailbox Register 3
0x1C	IMB4 Incoming Mailbox Register 4
0x20	FIFO Register Port (bi-directional)
0x24	MWAR Master Write Address Register
0x28	MWTC Master Write Transfer Counter
0x2C	MRAR Master Read Address Register
0x30	MRTC Master Read Transfer Counter
0x34	MBEF Mailbox Empty/Full Status
0x38	INTCSR Interrupt Control/Status Register
0x3C	MCSR Bus Master Control/Status Register

Table 2.2: AMCC Registers (HOST)

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

2.4 HOST CONTROL REGION

Accesses to the following offsets from BAR1 will allow for communication with the card.

NAME	ADDR	DATA	R/W	COMMENTS
INT SAMP CLK	0x00	DW	R/W	Divisor for Internal Sampling Clock
SM ADDRESS	0x04	DW	RO	Current Address for the State Machine
LAST ADDR 0	0x08	DW	R/W	Last Address with Valid Data, Bank 0
LAST ADDR 1	0x0C	DW	R/W	Last Address with Valid Data, Bank 1
BANK 0 CTRL	0x10	DWB	R/W	Controls Bank 0
BANK 1 CTRL	0x11	DWB	R/W	Controls Bank 1
CTRL/STAT 0	0x12	DWB	R/W	General control and status
CTRL/STAT 1	0x13	DWB	R/W	General control and status
RESET SAMP CLK	0x14	W	WS	Reset sampling clock counter
RESET ADDRESS	0x16	W	WS	Reset address counter
RESET DACS	0x18	W	WS	Reset all DACs
UPDATE DACS	0x1a	W	WS	Update all DACs

SWITCH BANKS	0x1c	W	WS	Switch active bank
DAC01	0x20	DW	W	Direct DAC output
DAC02	0x22	DW	W	Direct DAC output
DAC03	0x24	DW	W	Direct DAC output
DAC04	0x26	DW	W	Direct DAC output
DAC05	0x28	DW	W	Direct DAC output
DAC06	0x2a	DW	W	Direct DAC output
DAC07	0x2c	DW	W	Direct DAC output
DAC08	0x2e	DW	W	Direct DAC output
DAC09	0x30	DW	W	Direct DAC output
DAC10	0x32	DW	W	Direct DAC output
DAC11	0x34	DW	W	Direct DAC output
DAC12	0x36	DW	W	Direct DAC output
DAC13	0x38	DW	W	Direct DAC output
DAC14	0x3a	DW	W	Direct DAC output
DAC15	0x3c	DW	W	Direct DAC output
DAC16	0x3e	DW	W	Direct DAC output

Table 2.3: Host Control Region

2.4.1 INT SAMP CLK (Read / Write 32 bits)

This register sets the sampling rate of the internal sampling rate generator. The internal sampling rate generator is based on a 32 MHz oscillator on the card. The sampling rate is set by the following formula where N is the contents of this register.

$$SamplingRate = \frac{32000000}{2 + N}$$

Since the maximum sampling rate supported by the DACs on the **CPCI-DA16** is 100 KHz, the smallest value for *N* should be 318.

This register can be accessed in WORD and DWORD modes.

2.4.2 SM ADDRESS (Read Only 24 bits)

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)

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 bits)

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

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)

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
AUTO UPDATE DAC	ENABLE FP_RST	ENABLE STATE MACH	UNDER FLOW	ENABLE EXT SAMP CLOCK	ENABLE INT SAMP CLOCK	ENABLE CLOCK OUTPUT	ACTIVE BANK

AUTO UPDATE DAC ***Update outputs on manual DAC writes***

When this bit is set to 1, a manual update to a DAC register will also immediately update the output voltage. When this bit is cleared to 0, a manual write will only store the value in a holding register, and the output voltage can be updated by manually writing to the update address **UPDATE DACS**, or by the internal or external sampling clocks if enabled.

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

ENABLE FP_RST Allows FP_RST from front panel connector to stop output

When this bit is set to 1, a negative edge at the **FP_RST** line at the front panel connector will clear certain bits in the control status register in order to stop DAC output. When this bit is cleared to a 0, the **FP_RST** line has no effect. The state of this bit does not affect the generation of an interrupt based on the **FP_RST** line.

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

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, and by a negative edge on **FP_RST**, if enabled.

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)

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
INT FP_RST	INT SAMPLE CLOCK	INT BANK 1 DONE	INT BANK 0 DONE	N/A	ENABLE SAMPLE CLK INT	ENABLE FP_RST INT	CURRENT STATE FP_RST

INT FP_RST ***Interrupt caused by FP_RST going low***

When set, this bit indicates that an interrupt was generated by FP_RST going low. FP_RST can also stop the card from outputting values, and reset the DACs to 0 if enabled by ***ENABLE FP_RST***. Otherwise, **FP_RST** can be used as an external interrupt.

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

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.

ENABLE FP_RST INT ***Enable interrupt on FP_RST going low***

When this bit is set to 1, an interrupt is generated by a falling edge of FP_RST. When this bit is cleared to 0, no interrupt is generated. FP_RST can also stop the card from outputting values, and reset the DACs to 0 if enabled by ***ENABLE FP_RST***. Otherwise, **FP_RST** can be used as an external interrupt.

CURRENT STATE FP_RST (Read Only) ***Current state of the FP_RST input***

This read only bit reports the current state of the **FP_RST** line from the front panel connector.

2.4.7 RESET SAMP CLK (Write Strobe)

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)

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)

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

2.4.10 UPDATE DACS (Write Strobe)

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)

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)

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.

2.5 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 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

2.6.1 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.

2.6.2 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.

2.6.3 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.

2.6.4 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.

Alternatively, if the ***AUTO UPDATE DAC*** bit is set, the DACs will update the output voltage at the same time that the holding register is written.

2.7 RESET SIGNALS

The **CPCI-DA16** is reset when the CPCI bus is reset, and when the reset bit of the AMCC 5933 issues a board reset. The **FP_RST** line may be used to force a partial reset in order to stop output and force 0 voltage outputs from the DACs.

3. JUMPER DESCRIPTION

JUMPER	FACTORY SETTING	DESCRIPTION
W1	2-3	Power to the DACs. 1-2: Powered by externally supplied +12 or +15. 2-3: Powered at +12 V from BUS
W2	2-3	Power to the DACs. 1-2: Powered by externally supplied -12 or -15. 2-3: Powered at -12 V from BUS

Table 3.1 Jumper Descriptions

4. LED INDICATORS

There are four LED indicators visible at the CPCI card bracket. They are marked with a legend on the bracket, and they are labeled on the PCB as pairs L1 and L2 where L1 is at the top of the card. The LEDs have the following meanings:

LED	LEGEND	Meaning
L1 LF	S	Sample clock has triggered an output.
L1 RT	W	HOST has written to register or RAM.
L2 LF	1	Bank 1 is being output.
L2 RT	0	Bank 0 is being output.

Table 4.1 LED Descriptions

5. CONNECTIONS

5.1 ANALOG OUTPUT CONNECTOR (P1)

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 5.1: I/O Connector Model Numbers

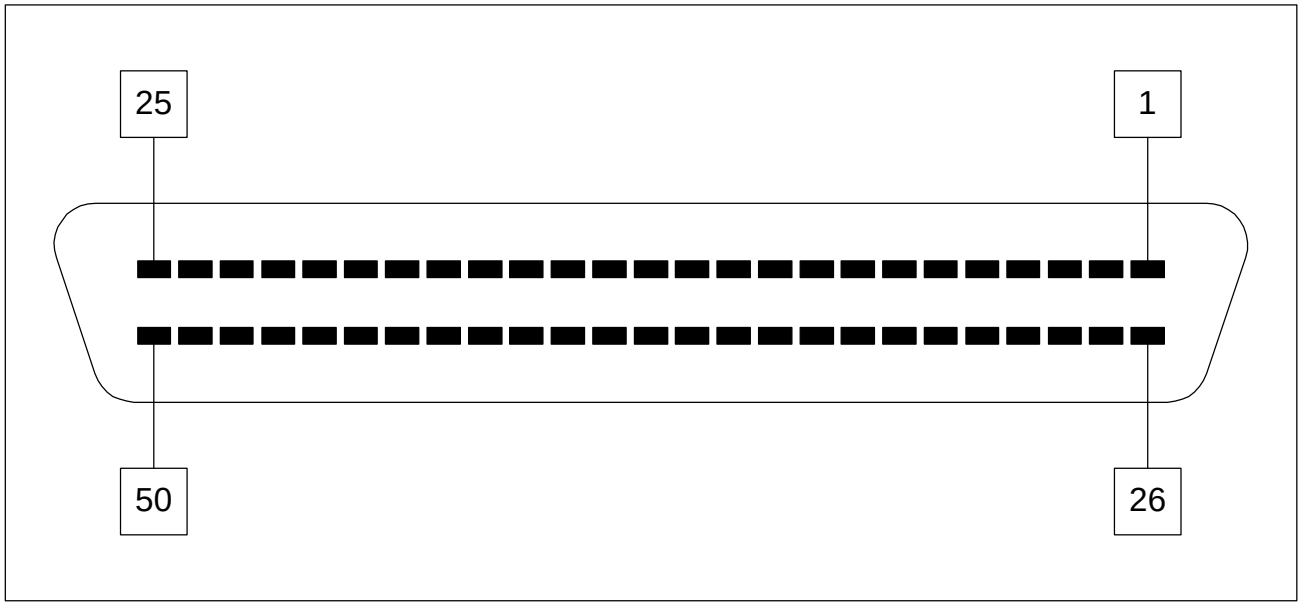

Figure 5.1: Analog Output Connector

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

Table 5.2: Analog Output Connector

5.2 32 BIT CPCI BUS (J1)

This connector plugs into the backplane and provides the standard CPCI signals for all CPCI systems.

5.3 FACTORY USE (P2)

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

5.4 FACTORY USE (P3)

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