

User's Guide

<http://www.omega.com>
e-mail: info@omega.com

PCI-DAS1602/16

Table of Contents

1.0 INTRODUCTION	Page 2
2.0 INSTALLATION	Page 2
2.1 <i>Software Installation</i>	Page 2
3.0 INSTACAL	Page 3
3.1 <i>System Requirements</i>	Page 3
3.2 <i>Running InstaCal</i>	Page 4
3.3 <i>Base I/O Address & Interrupt Level</i>	Page 4
3.4 <i>Testing The Installation</i>	Page 5
3.5 <i>Calibration</i>	Page 5
4.0 HARDWARE CONNECTIONS	Page 6
4.1 <i>Connector Pin Diagram</i>	Page 6
4.2 <i>Connecting Signals to the PCI-DAS1602/16</i>	Page 7
4.3 <i>Analog Input Configurations</i>	Page 7
5.0 FEATURES AND FUNCTIONS OVERVIEW	Page 9
6.0 PROGRAMMING & APPLICATIONS	Page 10
6.1 <i>Programming Languages</i>	Page 10
6.2 <i>Packaged Applications Programs</i>	Page 10
7.0 SELF-CALIBRATION OF THE PCI-DAS1602/16	Page 11
7.1 <i>Analog Input Calibration</i>	Page 11
7.2 <i>Analog Output Calibration</i>	Page 12
8.0 PCI-DAS1602/16 REGISTER DESCRIPTION	Page 13
8.1 <i>BADR0</i>	Page 13
8.2 <i>BADR1</i>	Page 13
8.2.1 <i>Interrupt / ADC FIFO Register</i>	Page 13
8.2.2 <i>ADC Channel MUX And Control Register</i>	Page 16
8.2.3 <i>Trigger Control/Status Register</i>	Page 18
8.2.4 <i>Calibration Register</i>	Page 21
8.2.5 <i>DAC Control/Status Register</i>	Page 23
8.3 <i>BADR2</i>	Page 25
8.3.1 <i>ADC Data Register</i>	Page 25
8.3.2 <i>ADC FIFO Clear Register</i>	Page 25
8.4 <i>BADR3</i>	Page 26
8.4.1 <i>ADC Pacer Clock Data And Control Registers</i>	Page 26
8.4.2 <i>High-Drive Digital I/O Data and Control Registers</i>	Page 27
8.4.3 <i>DAC Pacer Clock Data and Control Registers</i>	Page 29
8.5 <i>BADR4</i>	Page 30
8.5.1 <i>DAC Data Register</i>	Page 30
8.5.2 <i>DAC FIFO Clear Register</i>	Page 32
9.0 ELECTRICAL SPECIFICATIONS	Page 33

1.0 INTRODUCTION

The PCI-DAS1602/16 is a multifunction measurement and control board designed to operate in computers with PCI bus accessory slots. The architecture of the boards is loosely based on the original CIO-DAS16; the standard of ISA bus data acquisition. Much has changed though, and all of it due to improvements in technology.

Higher quality analog components have made 16-bit measurements the standard. Dense packaging technology and custom ASICs allow a far greater range of control over programmable options, such as calibration, triggering, synchronization and data transfer.

Even the connector has changed. New, denser connectors allow up to 100 signal lines where once 37 was the standard.

Because of the improvements in technology, the PCI-DAS1602/16 is easier to install and use than any previous DAS16. There is not a single switch or jumper on the board, so go ahead and install the PCI-DAS1602/16 into your computer then turn your computer on. Welcome to the future!

2.0 INSTALLATION

2.1 Software Installation

2.1.1 Windows 95, 98, NT or above

If you will be installing the Universal Library with your board, insert the Universal Library diskette or CD in an appropriate drive, run the program SETUP.EXE, and follow the installation instructions provided. This program will install both **InstaCal™** (setup and test utility) and the Universal Library. If you are using Windows 95, you will have the option of installing the 16-bit and/or 32-bit libraries. Unless you have a specific reason to use the 16-bit library (e.g. compatibility with an existing program) install the 32-bit version.

If you are not using the Universal Library, insert the disk or CD labeled **InstaCal™** into an appropriate drive, and run SETUP.EXE. The install wizard will now launch and you will then be prompted for additional information. Follow the instructions and, if possible, accept the defaults, especially if this is your first installation. It will be easier for us to assist you in the unlikely event of trouble during your system setup and operation.

2.1.2 UNIVERSAL LIBRARY INSTALLATION OPTIONS

The Universal Library provides example programs for a wide variety of programming languages. If you are installing the Universal Library, an "Installation Options" dialog box will allow you to select which languages' example programs are loaded onto your computer. Select the desired example programs by checking the appropriate box(s).

2.1.3 FILE DEFAULT LOCATION

InstaCal will place all appropriate files in "C:CB" If you change this default location remember where the installed files are placed as you may need to access them later.

2.1.4 INSTALLATION QUESTIONS

At the end of the installation process the installation wizard will ask a series of questions regarding updating your startup files. Unless you have knowledge to the contrary, simply accept the default (YES) when prompted. You will also be asked if you would like to read an updated README file. If possible, please choose yes and take a look at the information in the file. It will include the latest information regarding the software you are installing.

2.1.5 INSTALLATION COMPLETION

After the installation of *InstaCal* is complete you should restart your computer to take advantage of changes made to the system.

2.1.6 DOS and WINDOWS 3.x

Most users are now installing PCI Bus boards in systems with Windows operating systems (e.g., Windows 95, 98 or NT). If you are using Windows 3.x, the setup wizard will automatically install the 16-bit version of the Universal Library and InstaCAL. These versions are compatible with the DOS operating system.

If you need to install the software and do not have access to Windows, you will need to order the special "DOS Only" version of the software. The part number for these products are INSTACAL/DOS and Universal Library/DOS. Please contact the factory if you have any questions regarding these special DOS only versions.

3.0 InstaCal

InstaCal is the Installation, Calibration and Test software supplied with all I/O boards. After installing *InstaCal* you should re-start your computer to take advantage of changes made to the AUTOEXEC and CONFIG files. The PCI-DAS1602/16 does not have to be installed in order for *InstaCal* to run, but must be in order to test or calibrate the board.

3.1 System Requirements

Two versions of *InstaCal* are supplied with the PCI-DAS1602/16. The standard 32-bit version is compatible with Windows 95, 98, NT and greater. For those using older operating systems, the DOS based, 16-bit version is supplied and is compatible with DOS, Windows 3.x and Windows 95 (though we recommend Windows 95 users take advantage of the 32-bit version).

3.2 Running InstaCal

Be sure to restart your computer after the initial installation, and before running InstaCal. Run the *InstaCal* program in order to test your board and configure it for run-time use. By configuring the board, you add information to the configuration file, *cb.cfg*, that is used by the Universal Library as well as third-party data acquisition packages that use the Universal Library to access the board.

Launch *InstaCal* by going to your *Start Menu* then to *Programs*, then to *CB*, and finally choosing *InstaCal* (*InstCAL 16* if you wish to run the 16-bit version). You may also launch the program by going to *START>RUN* and typing *INSCAL32*, or by finding the file named "*INSCAL32.exe*" in your installation directory and double clicking it.

InstaCal will display a dialog box indicating the boards that have been detected in the system. If there are no other boards currently installed by *InstaCal*, then the PCI-DAS1602/16 board will be assigned board number 0. Otherwise it will be assigned the next available board number.

You can now view and change the board configuration by clicking the properties icon or selecting the *Install\Configure* menu.

Once done, exit *InstaCal*. This will update and save the configuration file, *CBI.CFG* in the *C:\CB* directory.

3.3 Base I/O Address & Interrupt Level

The PCI-DAS1602/16 uses a number of addresses and one interrupt. The addresses are allocated by the PCI plug & play procedure and may not be modified. If you have installed ISA bus boards in the past you are familiar with the need to select a base address and interrupt level. On PCI systems this is not of concern to you. It is not up to you to select a base address and ensure that it does not conflict with an installed port. In PCI systems, the operating software and installation software do the selection and checking for you.

The computer BIOS selects and sets the I/O address and interrupt level from the range of available addresses. This address and other information is read by *InstaCAL* and stored in the configuration file *CB.CFG*. This file is accessed by the Universal Library for programmers. Note also that the Universal Library is the I/O board interface for packaged applications such as Labtech Notebook and HP-VEE, therefore the *InstaCal* settings must be made in order for these and other applications to run.

The base address and interrupt level are also stored in the system software. Once *InstaCal* installation software is run, other programming methods such as direct *IN* and *OUT* statements can write and read the PCI-DAS1602/16 registers by reference to the base address and the offset from base address corresponding to the chart of registers located elsewhere in this manual.

But a word of warning is in order here. Direct writes to the addresses simply by reference to the base address of the PCI-DAS1602/16 I/O registers is not advised. Since the addresses assigned by the PCI plug & play software are not under your control, there is no way to guarantee that your program will run in any other computer.

Not only that, but if you install another PCI board in a computer after the PCI-DAS1602/16 addresses have been assigned, those addresses may be moved by the plug & play software when the second board is installed. It is best to use a library such as Universal Library or a program such as HP-VEE to make measurements with your PCI-DAS1602/16.

3.4 Testing The Installation

After you have run the install program and set your address and interrupt with InstaCal, it is time to test the installation. The following section describes the InstaCal procedure to test that your board is properly installed. The procedure has you connect one of the output channels to one of the A/D channels, it then outputs a simple waveform and shows you the waveform monitored on the selected A/D channel.

1. With InstaCal running, select the PCI-DAS1602/16.
2. Select the "TEST" function from the main menu
3. Follow the instructions provided

If you do not receive the expected results:

- a. make certain you have connected the correct pins according to the connector diagram.
- b. go back through the installation procedure and make sure you have installed the board according to the instructions.

If this does not get you to the desired display, please call us (or contact your local distributor) for additional assistance.

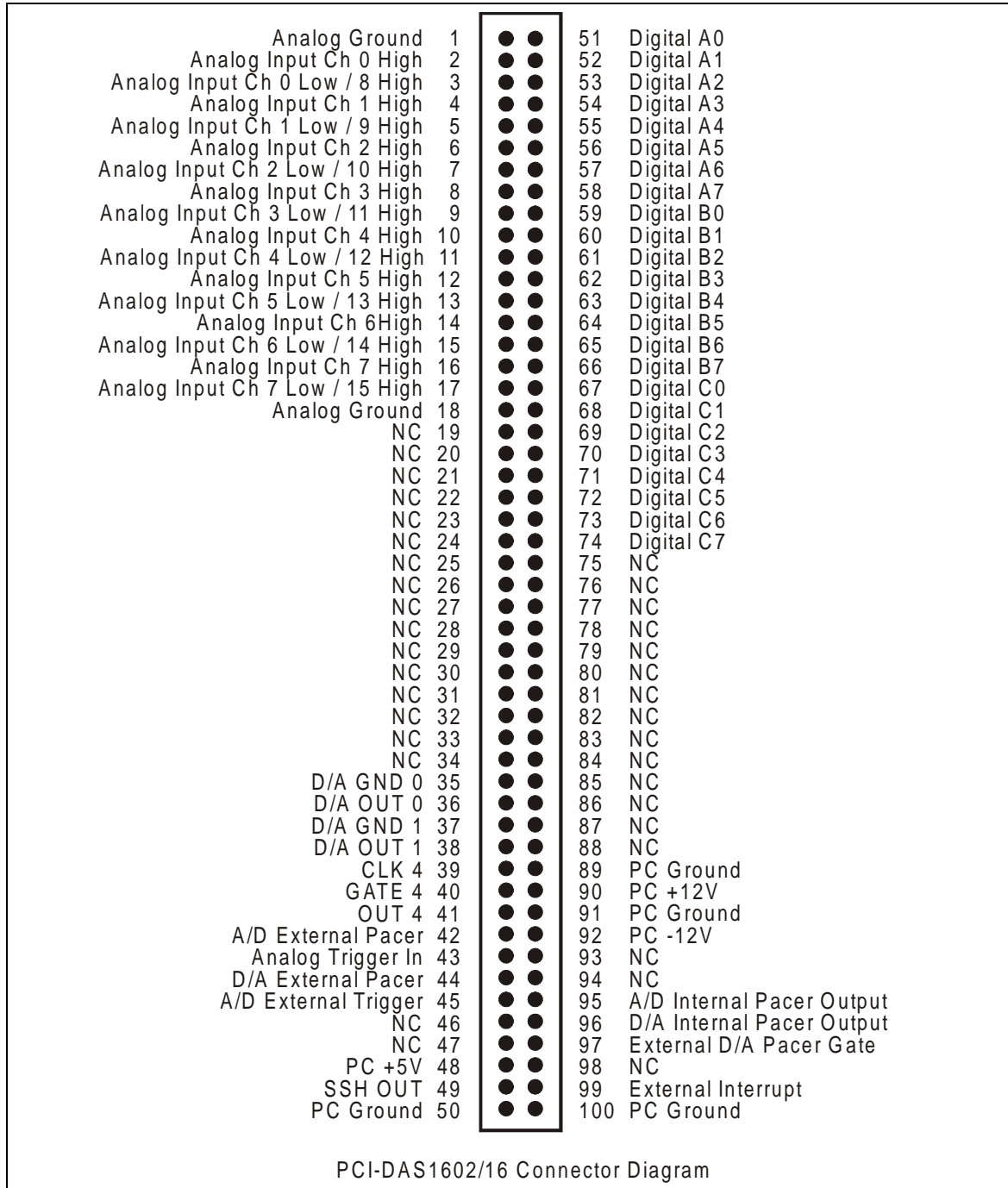
3.5 Calibration

Selecting *CALIBRATE* from the InstaCal main menu runs a fully automated PCI-DAS1602/16 calibration program. The software controlled calibration of the PCI-DAS1602/16 is explained extensively in the section on calibration.

4.0 HARDWARE CONNECTIONS

4.1 Connector Pin Diagram

The PCI-DAS1602/16 employs the new 100 pin connector. Please make accurate notes and pay careful attention to wire connections. In a large system a misplaced wire may create hours of work 'fixing' problems that do not exist before the wiring error is found.



4.2 Connecting Signals to the PCI-DAS1602/16

The 100 pin connector provides a far greater signal density than the traditional 37 pin D type connector. In exchange for that density comes a far more complex cable and mating connector. The C100-FF-2 cable is a pair of 50 pin ribbon cables. At one end they are joined together with a 100 pin connector. From the 100 pin connector designed to mate with the PCI-DAS1602/16 connector, the two 50 pin ribbon cables diverge and are terminated at the other end with standard 50 pin header connectors. A CIO-MINI50 screw terminal board is the ideal way to terminate real word signals and route them into the PCI-DAS1602/16

Analog inputs to the PCI-DAS1600 may be connected in three different configurations. In order of complexity, these are Single Ended, Floating Differential and Differential.

WARNING - PLEASE READ

Here is a good tip. Measure the voltage potential (difference) between the ground signal at the signal source and the PC. Use a volt meter and place the red probe on the PC ground and the black probe on the signal ground. If there is a difference of more than 10 volts, do not connect the CIO-DAS1600 to this signal source because you will not be able to make any reading. A difference more than 30 volts will likely damage the board and possibly the computer.

4.3 Analog Input Configurations

SINGLE ENDED

Single ended inputs are most appropriate in systems where the signal source and the data acquisition board share a common ground. This is a very common scenario and includes almost all systems where the data acquisition system is supplying either the power (+5V and GND) or the excitation (from a D/A). In this case you may take advantage of the PCI-1602/16's 16 channel mode without affecting performance.

Single ended inputs are also useful the signal source is electrically isolated (also referred to as floating) from the data acquisition board's ground (e.g. connecting to the two terminals of a battery). However, the use of differential input configuration will provide better noise immunity than single ended when monitoring an isolated signal source. Unless you absolutely need more than 8 channels, we recommend using the differential input mode for isolated input signals.

Please refer to the next paragraph for connection information for the connection of isolated signals to differential inputs.

FLOATING DIFFERENTIAL

A floating differential input is two wires from the signal source and a 10K ground reference resistor installed at the PCI-DAS1602/16 input. The two signals from the signal source are Signal High (CH# HI) and Signal Low (CH# LO).

The reference resistor is connected between the PCI-DAS1602/16 CH# LO and LLGND pins.

This floating configuration is appropriate when the signal source is floating with respect to ground, (e.g. a battery or 4-20mA transmitter), the lead lengths are long, your system is subject to substantial EMI interference.

WARNING!

Is that signal source really floating? Check it with a voltmeter before risking the PCI-DAS1602/16 and PC! If you can measure a constant voltage between the grounds of the PC and your signal source, it's probably not floating.

DIFFERENTIAL

Proper measurement of a differential signal requires three wires from the signal source. The signals are Signal High (CH# HI), Signal Low (CH# LO) and Signal Ground (LLGND).

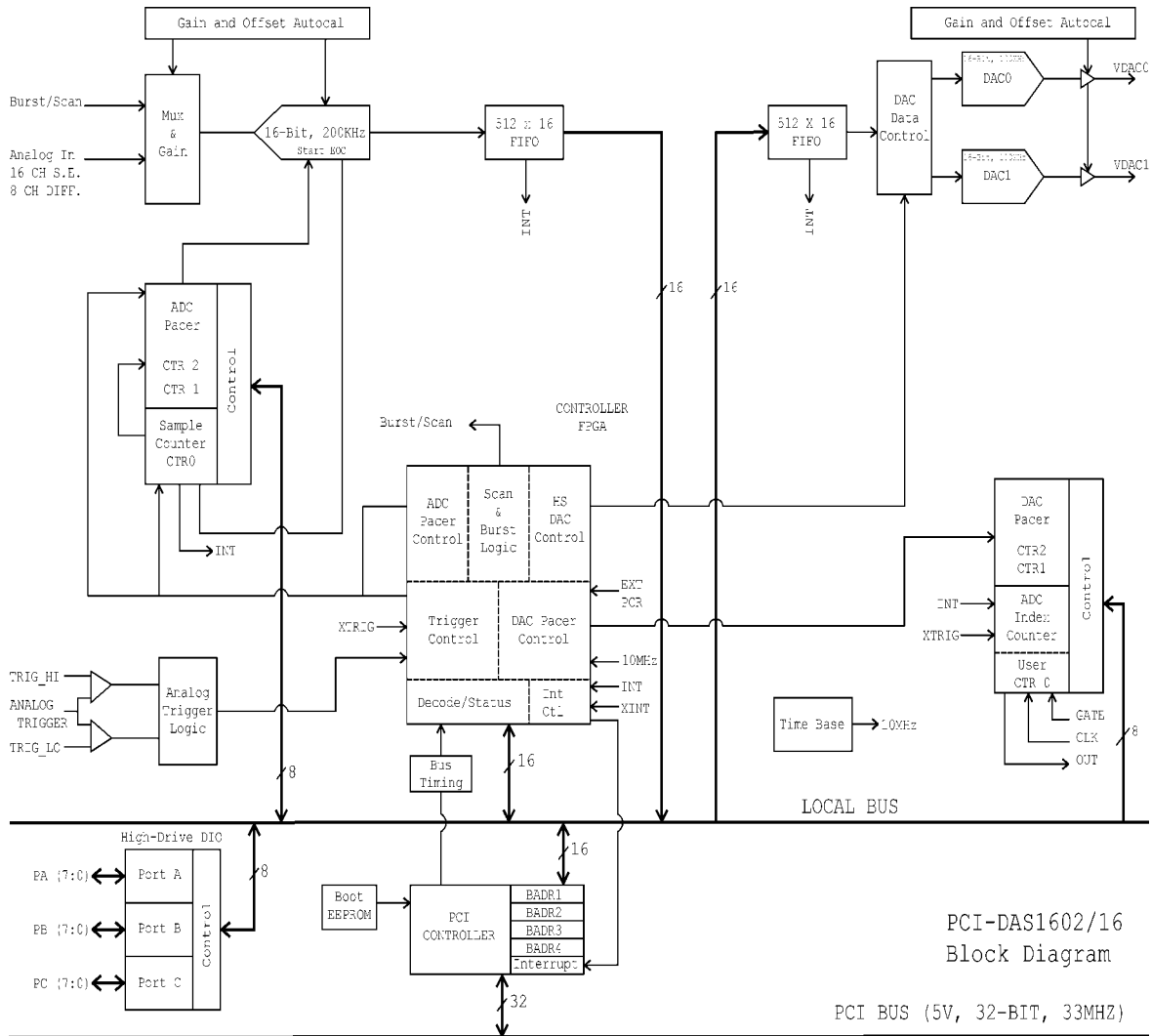
A differential connection allows you to connect the PCI-DAS1602/16 to a signal source with a ground that is different, but not isolated from the PC ground, but less than 10V difference, and still make a true measurement of the signal between CH# HI and CH# LO.

EXAMPLE:

Non-Isolated laboratory instruments with three prong wall plug. There are usually differences in wall GND potentials between outlets.

5.0 Features and Functions Overview

The PCI-DAS1602/16 is a multifunction measurement and control board. The design of the board may be simplified into several blocks containing the major functions of the board. Please take a moment to examine the diagram here.



PCI-DAS1602/16
Block Diagram

PCI BUS (5V, 32-BIT, 33MHZ)

6.0 Programming & Applications

Your PCI-DAS1602/16 is now installed and ready for use. Although the PCI-DAS1602/16 is part of the larger DAS family, there is no correspondence between registers. Software written at the register level for the other DAS's will not work with the PCI-DAS1602/16. This includes any driver or library where the target board is other than a PCI-DAS1602/16.

6.1 Programming Languages

The UniversalLibrary provides complete access to the PCI-DAS1602/16 functions from a range of programming languages; both DOS and Windows. If you are planning to write programs, or would like to run the example programs for Visual Basic or any other language, please turn now to the UniversalLibrary manual.

VIX Components is a set of programming tools based on a DLL interface to Windows languages. A set of VBX, OCX or ActiveX interfaces allows point and click construction of graphical displays, analysis and control structures. Please see the catalog for a complete description of the package.

6.2 Packaged Applications Programs

Many packaged application programs, such as DAS Wizard, Labtech Notebook and HP-VEE now have drivers for the PCI-DAS1602/16. If the package you own does not appear to have drivers for the PCI-DAS1602/16 please fax or e-mail the package name and the revision number from the install disks. We will research the package for you and advise how to obtain PCI-DAS1602/16 drivers.

Some application drivers are included with the Universal Library package, but not with the Application package. If you have purchased an application package directly from the software vendor, you may need to purchase our Universal Library and drivers. Please contact us for more information on this topic.

7.0 Self-Calibration of the PCI-DAS1602/16

The PCI-DAS1602/16 provides self-calibration of the analog source and measure systems thereby eliminating the need for external equipment and user adjustments. All adjustments are made via 8-bit calibration DACs or digital potentiometers referenced to an on-board factory calibrated standard. The PCI-DAS1602/16 is shipped fully-calibrated from the factory with calibration coefficients stored in nvRAM. At run time, these calibration factors are loaded into system memory and are automatically retrieved each time a different DAC/ADC range is specified. The user has the option to recalibrate with respect to the factory-measured voltage standards at any time by simply selecting the "Calibrate" option in InstaCal. Full calibration typically requires less than two minutes and requires no user intervention.

7.1 Analog Input Calibration

A variety of methods are used to calibrate the different elements on the board. The analog front-end has several "knobs" to turn. Offset calibration is performed in both the instrumentation amplifier gain stage and the ADC itself. Front-end gain adjustment is performed only via the ADC reference. This strategy was chosen since the gain tolerance of the in-amp circuit is quite good and there is adequate gain tuning range using only the ADC.

The analog output circuits are calibrated for gain and offset as well. Offset adjustments for the analog output are made in the output buffer section. The tuning range of this adjustment allows for max DAC and output buffer offsets. Gain calibration of the analog outputs are performed via DAC reference adjustments.

Figure 1 below is a block diagram of the analog front-end calibration system:

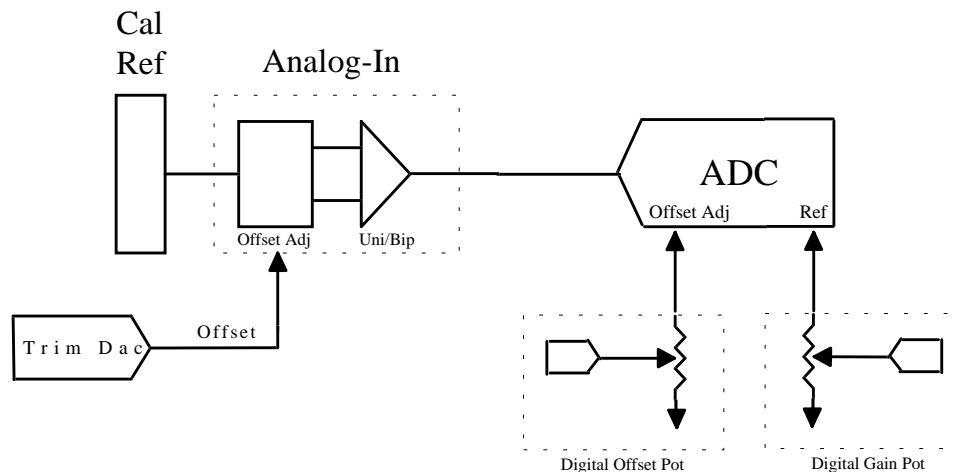


Figure 1

7.2 Analog Output Calibration

The analog output circuits are calibrated for both gain and offset. Coarse and Fine offset adjustments are made in the output buffer section. The tuning range of these adjustments allows for maximum DAC and output buffer offsets. Coarse and Fine gain calibration is performed via adjustments to the DAC reference.

Note that there are no references associated with the DAC calibration - a fully calibrated ADC set to the respective DAC range is used as the measurement system. Sub-ranging of the ADC is used to ensure highly accurate offset adjustments.

The calibration scheme for the Analog Out section is shown in Figure 2 below. This circuit is duplicated for both DAC0 and DAC1 Figure 2

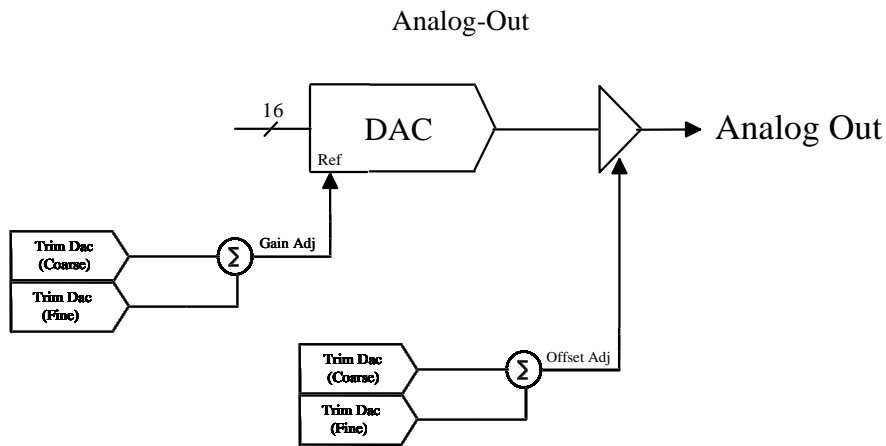


Figure 2

8.0 PCI-DAS1602/16 Register Description

The PCI-DAS1602/16 operation registers are mapped into the PC I/O address space. Unlike its ISA counterpart, this board has several base addresses each corresponding to a reserved block of addresses in I/O space. Of six Base Address Regions (BADR) available in the PCI 2.1 specification, five are implemented in this design and are summarized as follows:

I/O Region	Function	Operations
BADR0	PCI Controller Operation Registers	32-Bit DWORD
BADR1	General Control/Status Registers	16-Bit WORD
BADR2	ADC Data, FIFO Clear Registers	16-Bit WORD
BADR3	Pacer, Counter/Timer and DIO Registers	8-Bit BYTE
BADR4	DAC Data, FIFO Clear Registers	16-Bit WORD

BADRN will likely be different on different machines. Assigned by the PCI BIOS, these Base Address values cannot be guaranteed to be the same even on subsequent power-on cycles of the same machine. All software must interrogate BADR0 at run-time with a READ_CONFIGURATION_DWORD instruction to determine the BADRN values.

Please see the "AMCC S5933 PCI Controller Data Book, Spring 1996" for more information.

8.1 BADR0

BADR0 is reserved for the AMCC S5933 PCI Controller operations. This region supports 32-bit DWORD operations

8.2 BADR1

The I/O region defined by BADR1 contains 5 control and status registers for ADC, DAC, interrupt and Autocal operations. This region supports 16-bit WORD operations.

8.2.1 Interrupt / ADC FIFO Register

BADR1+ 0

Interrupt Control, ADC status. A read/write register.

WRITE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	DAEMCL	ADFLCL	DAEMIE	-	-	-	-	INTCL	EOACL	DAHFLCL	EOAIE	DAHIE	INTE	INT1	INT0

Write operations to this register allow the user to select interrupt sources, enable interrupts, clear interrupts as well as ADC FIFO flags. The following is a description of the Interrupt/ADC FIFO Register:

INT[1:0]

General Interrupt Source selection bits.

INT1	INT0	Source
0	0	External
0	1	End of Channel Scan
1	0	AD FIFO Half Full
1	1	AD FIFO Not Empty

INTE

Enables interrupt source selected via the INT[1:0] bits.

1 = Selected interrupt Enabled. 0 = Selected interrupt Disabled

DAHFIIE

Enables DAC FIFO Half-Full signal as interrupt source. Used for high speed DAC operations.

1 = Enable DAC FIFO Half-Full interrupt. 0 = Disable DAC FIFO Half-Full interrupt

EOAIE

Enables End-of-Acquisition interrupt. Used during FIFO'd ADC operations to indicate that the desired sample size has been gathered.

1 = Enable EOA interrupt. 0 = Disable EOA interrupt

DAHFCCL

A write-clear to reset DAC FIFO Half-Full interrupt status.

1 = Clear DAC FIFO Half-Full interrupt. 0 = No effect.

EOACL

A write-clear to reset EOA interrupt status.

1 = Clear EOA interrupt. 0 = No effect.

INTCLA write-clear to reset **INT[1:0]** selected interrupt status.1 = Clear **INT[1:0]** interrupt 0 = No effect.**DAEMIE**

Enables DAC FIFO Empty signal as an interrupt source.

1 = Enables DAC FIFO Empty interrupt. 0 = Disables DAC FIFO Empty interrupt.

ADFLCL

A write-clear to reset latched ADC FIFO Full status.

1 = Clear ADC FIFO Full latch. 0 = No Effect.

DAEMCL

A write-clear to reset DAEM interrupt status.

1= Clear DAEM interrupt. 0 = No effect.

NOTE: It is not necessary to reset any write-clear bits after they are set.

READ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	DAEMI	LADFUL	ADNE	ADNEI	ADHFI	EOBI	XINTI	INT	EOAI	DAHFI	-	-	-	-	-

Write operations to this register allow you to check status of the selected interrupts and ADC FIFO flags. The following is a description of Interrupt / ADC FIFO Register Read bits:

DAHFI

Status bit of DAC FIFO Half-Full interrupt

1 = Indicates a DAC FIFO Half-Full interrupt has been latched. 0 = Indicates a DAHF interrupt has not occurred.

EOAI

Status bit of ADC FIFO End-of-Acquisition interrupt.

1 = Indicates an EOA interrupt has been latched. 0 = Indicates an EOA interrupt has not occurred.

INT

Status bit of General interrupt selected via **INT[1:0]** bits. This bit indicates that *any* one of these interrupts has occurred.

1 = Indicates a General interrupt has been latched. 0 = Indicates a General interrupt has not occurred.

XINTI

Status bit of External interrupt. External interrupt requires a rising TTL logic level input.

1 = Indicates an External interrupt has been latched. 0 = Indicates an interrupt has not occurred.

EOBI

Status bit ADC End-of-Burst interrupt. Only valid for ADC Burst Mode enabled.

1 = Indicates an EOB interrupt has been latched. 0 = Indicates an EOB interrupt has not occurred.

ADHFI

Status bit of ADC FIFO Half-Full interrupt. Used during REP INSW operations.

1 = Indicates an ADC Half-Full interrupt has been latched. FIFO has been filled with more than 255 samples. 0 = Indicates an ADC Half-Full interrupt has not occurred. FIFO has not yet exceeded 1/2 of its total capacity.

ADNEI

Status bit of ADC FIFO Not-Empty interrupt. Used to indicate ADC conversion complete in single conversion applications.

1 = Indicates an ADC FIFO Not-Empty interrupt has been latched and that one data word may be read from the FIFO. 0 = Indicates an ADC FIFO Not-Empty interrupt has not occurred. FIFO has been cleared, read until empty or ADC conversion still in progress.

ADNE

Real-time status bit of ADC FIFO Not-Empty status signal.

1 = Indicates ADC FIFO has at least one word to be read. 0 = Indicates ADC FIFO is empty.

LADFUL

Status bit of ADC FIFO FULL status. This bit is latched.

1 = Indicates the ADC FIFO has *exceeded* full state. Data may have been lost. 0 = Indicates non-overflow condition of ADC FIFO.

DAEMI

Status bit of DAC FIFO Empty interrupt. Used to indicate that a FIFO'd DAC Operation has completed.

1 = DAC FIFO Empty interrupt condition has occurred. 0 = DAC FIFO Empty interrupt condition has not occurred.

8.2.2 ADC Channel MUX And Control Register

BADR1 + 2

This register sets channel mux HI/LO limits, ADC gain, offset and pacer source.

A Read/Write register.

WRITE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	ADPS1	ADPS0	UNIBIP	SEDIFF	GS1	GS0	CHH8	CHH4	CHH2	CHH1	CHL8	CHL4	CHL2	CHL1

CHL8-CHL1,**CHH8-CHH1**

When these bits are written, the analog input multiplexers are set to the channel specified by CHL8-CHL1. After each conversion, the input multiplexers increment to the next channel, reloading to the "CHL" start channel after the "CHH" stop channel is reached. LO and HI channels are the decode of the 4-bit binary patterns.

GS[1:0]

These bits determine the ADC range as indicated below:

GS1	GS0	Range
0	0	10V
0	1	5V
1	0	2.5V
1	1	1.25V

SEDIFF

Selects measurement configuration for the Analog Front-End.

1 = Analog Front-End in Single-Ended Mode. This mode supports up to 16 channels.

0 = Analog Front-End in Differential Mode. This mode supports up to 8 channels.

UNIBIP

Selects offset configuration for the Analog Front-End.

1 = Analog Front-End Unipolar for selected range

0 = Analog Front-End Bipolar for selected range.

The following table summarizes all possible Offset/Range configurations:

UNIBIP	GS1	GS0	Input Range	Input Gain	Measurement Resolution
0	0	0	±10V	1	305uV
0	0	1	± 5V	2	153uV
0	1	0	±2.5V	4	76uV
0	1	1	±1.25V	8	38uV
1	0	0	0-10V	1	153uV
1	0	1	0-5V	2	76uV
1	1	0	0-2.5V	4	38uV
1	1	1	0-1.25V	8	19uV

ADPS[1:0]

These bits select the ADC Pacer Source. Maximum Internal/External Pacer frequency is 200KHz.

ADPS1	ADPS0	Pacer Source
0	0	SW Convert
0	1	82C54 Counter/Timer
1	0	External Falling
1	1	External Rising

Note: For ADPS[1:0] = 00 case, SW conversions are initiated via a word write to BADR2 + 0. Data is 'don't care.'

READ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	EOC	-	-	-	-	-	-	-	-	-	-	-	-	-	-

EOC

Real-time, non-latched status of ADC End-of-Conversion signal.

1 = ADC DONE

0 = ADC BUSY

8.2.3 Trigger Control/Status Register

BADR1 + 4

This register provides control bits for all ADC trigger modes. A Read/Write register.

WRITE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	COSRC	FFM0	ARM	HMODE	CHL_EN	CLO_EN	XTRCL	PRTRG	BURSTE	TGEN	TGSEL	TGPOL	TS1	TS0

TS[1:0]

These bits select one-of-three possible ADC Trigger Sources:

TS1	TS0	Source
0	0	Disabled
0	1	SW Trigger
1	0	External (Digital)
1	1	External (Analog)

Note: TS[1:0] should be set to 0 while setting up Pacer source and count values.

TGPOL

This bit sets the polarity for the external trigger/gate. Internally, the ADC is triggered on a rising edge or gated on with an active high signal. Use **TGPOL** to condition external trigger/gate for proper polarity.

1 = External trigger/gate input inverted.

0 = External trigger/gate input not inverted.

TGSEL

This bit selects whether external ADC control signal is an edge or a level. Use **TGPOL** signal to create rising edge or high level input.

- 1 = Edge triggered.
- 0 = Level triggered.

TGEN

This bit is used to enable External Trigger/Gate function

- 1 = Selected Trigger Source enabled.
- 0 = Selected Trigger Source has no effect.

Note that external trigger/gate requires proper setting of the **TS[1:0]**,

TGPOL, **TGSEL** and **TGEN** bits.

Example: Application requires use of external falling edge to start acquisition. Set:

- TS1 = 1, TS0 = 0 -> External Digital Trigger
- TGPOL = 1 -> Invert falling edge
- TGSEL = 1 -> Edge Triggered event
- TGEN = 1 -> Enable External Trigger.

Once **TGEN** is set the next falling edge will start a Paced ADC conversion. Subsequent triggers will have no effect until external trigger flop is cleared (**XTRCL**).

BURSTE

bit enables ADC Burst mode. Start/Stop channels are selected via the CHLx, CHHx bits in ADC CTRL/STAT register at BADR1 + 2.

- 1 = Burst Mode enabled
- 0 = Burst Mode disabled

PRTRG

This bit enables ADC Pre-trigger Mode. This bit works with the ARM and FFM0 bits when using Pre-trigger mode.

- 1 = Enable Pre-trigger Mode
- 0 = Disable Pre-trigger Mode

XTRCL

A write-clear to reset the **XTRIG** flip-flop.

- 1 = Clear **XTRIG** status.
- 0 = No Effect.

CHI_EN

These bits select the Analog Trigger/Gate Mode as described in the table below.

CLO_EN

Note that the CHI Threshold is set by DAC1, CLO Threshold is set by DAC0.

HMODE

CHI >= CLO by definition.

CHI_EN	CLO_EN	HMODE	Analog Trigger/Gate Function	Mode
0	0	0	Signal goes HIGH when ATRIG is more positive than CHI. Signal goes low when ATRIG becomes more negative than CLO. Hysteresis level is the difference between CHI and CLO.	Negative Hysteresis
0	0	1	Signal goes HIGH when ATRIG is more negative than CLO. Signal goes low when ATRIG becomes more positive than CHI. Hysteresis level is the difference between CHI and CLO.	Positive Hysteresis
0	1	X	Signal goes high when ATRIG more negative than CLO. CHI has no effect.	Negative Slope
1	0	X	Signal goes high when ATRIG is more positive than CHI. CLO has no effect.	Positive Slope
1	1	X	Signal goes high when within region defined by CHI-CLO. Signal is low outside this region.	Window

ARM, FFM0

These bits work in conjunction the **PRTRG** bit during FIFO'd ADC operations. Note that 1 FIFO = 512 samples.

PRTRG	FFM0	ARM is set...	FIFO Mode	Sample CTR Starts on...
0	0	Via SW when remaining count <1 FIFO ----- Via SW immediately	# Samples >1 FIFO Normal Mode ----- 1/2 FIFO < # Samples < 1 FIFO Normal Mode	ADHF
0	1	Via SW immediately	# Samples <1/2 FIFO Normal Mode	ADC Pacer
1	0	Via SW when remaining count <1 FIFO ----- Via SW after XTRIG has been detected (INDX_GT=1)	# Post-Trigger Samples >1 FIFO Pre-Trigger Mode ----- 1/2 FIFO < # Post-Trigger Samples < 1 FIFO Pre-Trigger Mode	ADHF
1	1	Via SW after XTRIG has been detected (INDX_GT=1)	# Post-Trigger Samples < 1/2 FIFO Pre-Trigger Mode	Via SW after INDX_GT=1

C0SRC

This bit allows the user to select the clock source for user Counter 0.

1 = Internal 10MHz oscillator

0 = External clock source input via *CTROCLK* pin on 100p connector.

READ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	INDX-GT	-	-	-	-	XTRIG	-	-	-	-	-	-	-

XTRIG

1 = External Trigger flip-flop has been set. This bit is write-cleared.

0 = External Trigger flip-flop reset. No trigger has been received.

INDX-GT

1 = Pre Trigger index counter has completed its count

0 = Pre Trigger index counter has not yet been gated on, or has not yet completed its count.

8.2.4 Calibration Register

BADR1 + 6

This register controls all autocal operations. A Write only register.

WRITE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDI	CALEN	CSRC2	CSRC1	CSRC0	SEL08	SEL8402	SEL8800	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0

CD[7:0]

These 8 bits are the D/A code inputs for the analog-front DAC08 offset calibration DAC. Complimentary current outputs of the DAC08 are equal at mid-scale, 7Fh. This should be the default, non-calibrated value.

SEL8800

This bit enables the 8-bit trim DACs for the following circuits:

DAC Channel	Cal Function
0	DAC0 Fine Gain
1	DAC0 Coarse Gain
2	DAC0 Offset
3	DAC1 Offset
4	DAC1 Fine Gain
5	DAC1 Coarse Gain
6	DAC0 Fine Offset
7	DAC1 Fine Offset

SEL8402

This bit enables the 8-bit digital potentiometers (50KOhm) for the following circuits.

Trimmer Channel	Cal Function
0	ADC Gain Cal
1	ADC Offset Cal

SEL08

This bit enables conversions on the 8-bit DAC08 for Analog Front-End offset calibration. Data value is set via **CD[7:0]**.

1 = Offset DAC conversions enabled.

0 = Offset DAC conversions disabled (last value held).

CSRC[2:0]

These bits select the different calibration sources available to the ADC front end.

CSRC2	CSRC1	CSRC0	Cal Source
0	0	0	AGND
0	0	1	7.0V
0	1	0	3.5V
0	1	1	1.75V
1	0	0	0.875V
1	0	1	-10.0V
1	1	0	VDAC0
1	1	1	VDAC1

CALEN

This bit is used to enable Cal Mode.

1 = Selected Cal Source, **CSRC[2:0]**, is fed into Analog Channel 0.

0 = Analog Channel 0 functions as normal input.

SDI

Serial Data In. This bit is used to set serial address/data stream for the DAC8800 TrimDac and 8402 digital potentiometer. Used in conjunction with **SEL8800** and **SEL8402** bits.

8.2.5 DAC Control/Status Register

BADR1 + 8

This register selects the DAC gain/range, Pacer source, trigger and High-Speed Modes. In addition, DAC FIFO status information is available. This is a Read/Write register.

WRITE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	DAC1R1	DAC1R0	DAC0R1	DAC0R0	-	HS1	HS0	DAPS1	DAPS0	START	DACEN	LDAEMCL

LDAEMCL

This is a Write-clear bit to reset the latched EMPTY status flag of the DAC FIFO.

1 = Reset Empty Flag

0 = No Effect.

DACEN

This bit enables the Analog Out features of the board.

1 = DAC0/1 enabled.

0 = DAC0/1 disabled.

START

This bit starts FIFO'd DAC operations. If used with **DAXTRG**, the external trigger signal, the **START** bit is used to arm the operation.

1 = Start/Arm FIFO operations.

0 = Disable FIFO'd DAC operations.

DAPS[1:0]

These bits select the DAC Pacer Source:

DAPS1	DAPS0	Pacer Source
0	0	SW Convert
0	1	Internal 82C54 Programmed via BADR3 + 9, + A
1	0	External Falling Edge
1	1	External Rising Edge

HS[1:0]

These bits select the High-Speed DAC Modes as follows:

HS1	HS0	DAC Mode
0	0	Disabled
0	1	DAC0
1	0	DAC1
1	1	Simultaneous DAC0/1

DACnR[1:0]

These bits select the independent gains/ranges for either DAC0 or DAC1. n=0 for DAC0 and n=1 for DAC1.

DACnR1	DACnR0	Range	LSB Size
0	0	Bipolar 5V	153uV
0	1	Bipolar 10V	305uV
1	0	Unipolar 5V	76uV
1	1	Unipolar 10V	153uV

READ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LDAEM

LDAEM

This is the latched version of the DAC FIFO_EMPTY signal. This bit must be write-write cleared with the **DAEMCL** bit.

1 = DAC FIFO was emptied at some point during FIFO'd operations. Incorrect data may have been clocked into the selected DAC(s).

0 = DAC FIFO did not empty during FIFO'd operations. Status good.

8.3 BADR2

The I/O Region defined by BADR2 contains the ADC Data register and the ADC FIFO clear register.

8.3.1 ADC Data Register

BADR2 + 0 ADC Data register.

WRITE

Writing to this register is only valid for SW initiated conversions. The ADC Pacer source must be set to 00 via the ADPS[1:0] bits. A null write to BADR2 + 0 will begin a single conversion. Conversion status may be determined in two ways. The **EOC** bit in BADR1 + 0 may be polled until true or **ADNEI** (the AD FIFO not-empty interrupt) may be used to signal that the ADC conversion is complete and the data word is present in the FIFO.

READ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

MSB

LSB

AD[15:0]

This register contains the current ADC data word. Data format is dependent upon offset mode:

Bipolar Mode: Offset Binary Coding

0000 h = -FS

7FFFh = Mid-scale (0V)

FFFFh = +FS - 1LSB

Unipolar Mode: Straight Binary Coding

0000 h = -FS (0V)

7FFFh = Mid-scale (+FS/2)

FFFFh = +FS - 1LSB

8.3.2 ADC FIFO Clear Register

BADR2 + 2 ADC FIFO Clear register. A Write-only register. A write to this address location clears the ADC FIFO. Data is don't care. The ADC FIFO should be cleared before all new ADC operations.

8.4 BADR3

The I/O Region defined by BADR3 contains data and control registers for the ADC Pacer, DAC Pacer, Pre/Post-Trigger Counters and High-Drive Digital I/O bytes. The PCI-DAS1602/16 has two 8254 counter/timer devices. These are referred to as 8254A and 8254B and are assigned as shown below:

Device	Counter #	Function
8254A	0	ADC Post-Trigger Sample Counter
8254A	1	ADC Pacer Lower Divider
8254A	2	ADC Pacer Upper Divider
8254B	0	ADC Pre-Trigger Index/UserCounter
8254B	1	DAC Pacer Lower Divider
8254B	2	DAC Pacer Upper Divider

All reads/writes to BADR3 are *byte* operations.

8.4.1 ADC Pacer Clock Data And Control Registers

8254A COUNTER 0 DATA - ADC RESIDUAL SAMPLE COUNTER

BADR3 + 0

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Counter 0 is used to stop the acquisition when the desired number of samples have been gathered. It is gated on when a 'residual' number of conversions remain. Counter 0 will be enabled by use of the **ARM** bit (BADR1 + 4).

Counter 0 is to operated in Mode 0.

8254A COUNTER 1 DATA - ADC PACER DIVIDER LOWER

BADR3 + 1

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

8254A COUNTER 2 DATA - ADC PACER DIVIDER UPPER

BASE + 2

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Counter 1 provides the lower 16 bits of the 32-bit pacer clock divider. Its output is fed to the clock input of Counter 2 which provides the upper 16-bits of the pacer clock divider. The clock input to Counter 1 is a precision 10MHz oscillator source.

Counter 2 output is called the 'Internal Pacer' and can be selected by software to be the ADC Pacer source. Counters 1 & 2 should be configured to operate in 8254 Mode 2.

ADC 8254 CONTROL REGISTER

BADR3 + 3

WRITE ONLY

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

The control register is used to set the operating Modes of 8254 Counters 0,1 & 2. A counter is configured by writing the correct Mode information to the Control Register followed by count written to the specific Counter Register.

The Counters on the 8254 are 16-bit devices. Since the interface to the 8254 is only 8-bits wide, Count data is written to the Counter Register as two successive bytes. First the low byte is written, then the high byte. The Control Register is 8-bits wide. Further information can be obtained on the 8254 data sheet, available from Intel or Harris.

8.4.2 High-Drive Digital I/O Data and Control Registers

The 24 High-Drive DIO lines on the PCI-DAS1602/16 are grouped as three byte-wide I/O ports. Port assignment and functionality is identical to that of the industry standard 8255 Peripheral Interface operating in Mode 0. Please see the Intel or Harris data sheets for more information. Drive capability of each line is -15mA in the TTL HI state and 24mA in the TTL LO state.

HDIO PORT A DATA

BADR3 + 4

PORT A may be configured as an 8-bit I/O channel.

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

HDIO PORT B DATA

BADR3 + 5

PORT B may be configured as an 8-bit I/O channel. Its functionality is identical to that of PORT A.

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

HDIO PORT C DATA

BADR3 + 6

PORT C may be configured as an 8-bit port of either input or output, or it may be split into two independent 4-bit ports of input or output. When split into two 4-bit I/O ports, **D[3:0]** make up the lower nibble, **D[7:4]** comprise the upper nibble. Although it may be split, every write to Port C is a byte operation. Unwanted information must be ANDed out during reads and writes must be ORd with current value of the other 4-bit port.

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

HDIO CONTROL REGISTER

BADR3 + 7

The HDIO Control register is used configure Ports A,B and C as inputs or outputs. Operation is identical to that of the 8255 in Mode 0.

WRITE

7	6	5	4	2	3	1	0
1	0	0	D4	D3	0	D1	D0

Note: Bits 3,5-7 are hardwired to the values shown (Mode 0). Actual writes to these bit positions are "don't care."

The following table summarizes the possible I/O Port configurations for the PCI-DAS1602/16 HDIO:

D4	D3	D1	D0	PORT A	PORT C UPPER	PORT B	PORT C LOWER
0	0	0	0	OUT	OUT	OUT	OUT
0	0	0	1	OUT	OUT	OUT	IN
0	0	1	0	OUT	OUT	IN	OUT
0	0	1	1	OUT	OUT	IN	IN
0	1	0	0	OUT	IN	OUT	OUT
0	1	0	1	OUT	IN	OUT	IN
0	1	1	0	OUT	IN	IN	OUT
0	1	1	1	OUT	IN	IN	IN
1	0	0	0	IN	OUT	OUT	OUT
1	0	0	1	IN	OUT	OUT	IN
1	0	1	0	IN	OUT	IN	OUT
1	0	1	1	IN	OUT	IN	IN
1	1	0	0	IN	IN	OUT	OUT
1	1	0	1	IN	IN	OUT	IN
1	1	1	0	IN	IN	IN	OUT
1	1	1	1	IN	IN	IN	IN

8.4.3 DAC Pacer Clock Data and Control Registers

8254B COUNTER 0 DATA - ADC PRE-TRIGGER INDEX COUNTER

BADR3 + 8

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Counter 0 of the DAC 8254 device is actually used as the ADC Pre-Trigger index counter. This counter serves to mark the boundary between pre- and post-trigger samples when the ADC is operating in Pre-Trigger Mode. The External ADC Trigger flip flop gates Counter 0 on; the ADC FIFO Half-Full signal gates it off. Knowing the desired number of post-trigger samples, software can then calculate how many 1/2 FIFO data packets need to be collected and what corresponding residual sample count needs to be written to BADR3 + 0.

8254B COUNTER 1 DATA - DAC PACER DIVIDER LOWER

BADR3 + 9

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

8254B COUNTER 2 DATA - DAC PACER DIVIDER UPPER

BADR3 + Ah

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Counter 1 provides the lower 16 bits of the 32-bit pacer clock divider. Its output is fed to the clock input of Counter 2 which provides the upper 16-bits of the pacer clock divider. The clock input to Counter 1 is a precision 10MHz oscillator source.

Counter 2's output is called the 'Internal Pacer' and can be selected by software to be the ADC Pacer source. Counters 1 & 2 should be configured to operate in 8254 Mode 2.

8254B CONTROL REGISTER

BADR3 + Bh

WRITE ONLY

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

The control register is used to set the operating Modes of 8254 Counters 0,1 & 2. A counter is configured by writing the correct Mode information to the Control Register, then the proper count data must be written to the specific Counter Register.

The Counters on the 8254 are 16-bit devices. Since the interface to the 8254 is only 8-bits wide, Count data is written to the Counter Register as two successive bytes. First the low byte is written, then the high byte. The Control Register is 8-bits wide. Further information can be obtained on the 8254 data sheet, available from Intel or Harris.

8.5 BADR4

The I/O Region defined by BADR4 contains the shared DAC data register and the DAC FIFO clear register.

8.5.1 DAC Data Register

BADR4 + 0

DAC Data register. A Write-only register.

WRITE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

MSB

LSB

DA[15:0]

These bits represent the DAC data word. Format is dependent upon offset mode as described below:

Bipolar Mode: Offset Binary Coding

0000 h = -FS

7FFFh = Mid-scale (0V)

FFFFh = +FS - 1LSB

Unipolar Mode: Straight Binary Coding

0000 h = -FS (0V)

7FFFh = Mid-scale (+FS/2)

FFFFh = +FS - 1LSB

Paced DAC operations require that the FIFO be loaded with the appropriate data. A REP OUTSW instruction to this address will do this. It is important to note that the FIFO is the shared data source between DAC0 and DAC1. Care must be taken to ensure that DAC0 data always precedes DAC1 data during simultaneous operations. Target DAC selection is made via the **HS[1:0]** bits described earlier.

HS1	HS0	SELECTED DAC(S)	LOCATION #	FIFO DATA
0	0	None	N/A	N/A
0	1	DAC0	0	DAC0
			1	DAC0
			2	DAC0
			3	DAC0
1	0	DAC1	0	DAC1
			1	DAC1
			2	DAC1
			3	DAC1
1	1	DAC0 & DAC1	0	DAC0
			1	DAC1
			2	DAC0
			3	DAC1

NOTE: FIFO location #0 is the first value written to the Cleared DAC FIFO.

