ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Datel PCI-416-G2A* at our website: ***Click HERE***

TABLE OF CONTENTS

Section 1 - Installation and Setup

1.0 Introduction 1-1
1.1 Scope 1-1
1.2 About Microsoft Windows 1-2
1.3 Application Support 1-3
1.4 Suggested Equipment 1-3
1.5 Static Discharge Protection 1-3
1.6 Installation Outline 1-4
1.7 Jumpers and Board Components 1-4
1.8 PCI Slot Selection and Host Compatibility 1-5
1.9 Installation Procedure 1-5
1.9.1 Troubleshooting 1-6

Section 2 - Input/Output Connections and Board Layout

2.0 Analog Input Connector 2-1
2.1 82C55 Digital I/O Port - P2 2-3
2.2 Jumpers 2-4
2.2.1 Board Component Layout 2-4
2.2.2 Jumpers and Board Components 2-4

Section 3 - Analog Modules

3.0 Introduction 3-1
3.1 Physical Characteristics 3-2
3.2 ADM-12A (PCI-416A) 3-4
3.3 ADM-14B (PCI-416B) 3-6
3.4 ADM-12C (PCI-416C) 3-7
3.5 ADM-12D (PCI-416D) 3-9
3.6 ADM-12E (PCI-416E) 3-10
3.7 ADM-12F (PCI-416F) 3-11
3.8 ADM-14G (PCI-416G) 3-13
3.9 ADM-12H (PCI-416H) 3-14
3.10 ADM-8SIM (PCI-416J) 3-15
3.10.1 Input Polarity and Full Scale Range 3-17
3.10.2 Sign Extension 3-17
3.10.3 Production Programming Pins and Miscellaneous Functions 3-18
3.10.4 Channel Addressing 3-18
3.10.5 Timing Diagram 3-19
3.10.6 Calibration 3-20
3.11 Input Amplifier Programmable Gain Selection 3-21
3.12 Repair and Troubleshooting 3-21

Section 4 - PCI-416 Register Descriptions

4.0 Introduction 4-1
4.1 DATEL Defaults 4-1
4.2 Hardware Register Map 4-2
4.3 Programming Guidelines 4-3
4.4 Non-volatile Configuration Memory 4-4
4.5 Register Bit Assignments 4-4

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Section 5 - Using the PCI-416

5.0 Introduction 5-1
5.1 Application Design Strategy 5-3
5.2 FIFO A/D Memory 5-5
5.3 Counter/Timing Systems 5-8
5.3.1 Frame Architecture and Channel Addressing 5-9
5.3.2 Triggering versus A/D Clocking 5-11
5.3.3 Triggers 5-12
5.3.4 External A/D Start Clock Operation 5-13
5.4 A/D Sample Clocking 5-16
5.4.1 Frequency Synthesizer Operation 5-17
5.4.2 Frequency Synthesizer Programming 5-19
5.5 Sample Counting 5-20

Section 6 - Operation Notes

6.0 82C55 Parallel Port 6-1
6.1 Modes 6-1
6.2 Programming the 82C55 6-2
6.3 A/D Calibration 6-2
6.4 Procedure 6-3
6.5 Zero/Offset Adjust 6-4
6.6 A/D Gain Adjustment 6-4
6.7 Universal Coding Chart 6-5
6.8 D/A Calibration 6-5
6.9 Hard Drive Interrupt Conflict Solutions for the PCI-416 6-6
6.10 Policy on Design Information and Modifications 6-8
6.11 PCI-416 Bus Master Operation 6-8
6.12 Analog Signal Recommendations 6-12
6.13 Windows 95 Compatibility, March 1996 6-12

Section 7 - 82C54 Programmable Interval Timer

7.0 General Description 7-1
7.1 82C54 Modes of Operation 7-3
7.2 82C54 Operation in PCI-416 7-5
7.3 82C54 Programming Guide for the PCI-416 7-5

Section 8 - PCI-416SET Software

8.0 Introduction 8-1
8.1 "Front End" to Signal Processing Software 8-1
8.2 Interleaved Operations Using "Start and Exit Mode" 8-1
8.3 PCI-416 Program Design Philosophy 8-2
8.4 Program Outline 8-3
8.5 Configuration Files 8-3
8.6 Command Line Parameters 8-3
8.7 PCI 416 Top Menu 8-4
8.8 Start and Exit Mode 8-6
8.9 Porting PCI-416SRC to Other Operating Systems 8-6
8.10 Additional Notes on PCI-416SET Operation for Version 1.2 8-6

Appendix A - Intel 82C55A Data Sheet

Appendix B - PCI-416 Data Sheet

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SECTION 1

INSTALLATION AND SETUP

1.0 Introduction

Thank you for your interest in DATEL's advanced PCI-416 data acquisition board family for PCI computers. The PCI-416 data sheet, listed in the Appendix of this manual contains a detailed description of the PCI-416. Please refer to it for full specifications, product description, block diagram, ordering guide and applications.

The PCI specification is documented by:

Intel Corp., c/o PCI Special Interest Group
539 SE Division Place
Portland, OR, U.S.A. 97202-1047
Phone: 800-433-5177 or 503-797-4207 Fax: 503-234-6762

The PCI Group tends to change addresses now and then but usually stays in the Intel building and is sometimes listed under the Hillsboro, Oregon Intel address. The PCI Group also has a forum on Compuserve, "GO PCVENH".

A highly recommended book for PCI is "PCI Systems Architecture" by Tom Shanley and Don Anderson, ISBN 1-881609-08-1, published by:

Mindshare Press Inc.
2202 Buttercup Drive
Richardson, TX, U.S.A. 75082
Phone: 214-231-2216 Fax: 214-783-4715

The PCI book is also available from:

Annabooks
P.O. Box 270708
San Diego, CA, U.S.A. 92198-2708
Phone: 800-462-1042 or 619-673-0870 Fax: 619-673-1432 Email: 73204.3405@compuserve.com

Annabooks has numerous other resources devoted to PCI including seminars, trade shows, and PCI directories.

1.1 Scope

This manual contains complete hardware and software programming information for the PCI-416, including installation, configuration, I/O connections and application information. This guide is written in the sequence most users prefer to use the board. This manual is not a repair manual. There are no user-serviceable parts on the PCI-416 (besides the "ADM" pluggable analog module) and any attempts to repair the board without special training or equipment may damage it. All repairs should be referred to DATEL.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PCI-416 users should be familiar with all aspects of developing a high performace data acquisition application. This includes selecting and installing external analog sensors or transducers, wiring and connectors, complete operation and software programming of the host computer. You should have a general knowledge of how data acquisition systems work, possible error sources (both hardware and software) and math operations on the data. If you are planning any high speed signal processing, you should understand data acquisition sampling theory and probably also Digital Signal Processing mathematics. Some knowledge of linear circuits, digital logic and microprocessors will be very helpful. You may need several people with different skills to accomplish your goals.

You will find it helpful to know the "C" high level programming language and some 80X86 Assembly language. Although the PCI-416SET/SRC software will allow you to acquire A/D data to memory or disk files, PCI-416SET will *NOT* perform all conceivable applications by itself. To achieve a specific application, you may need to write your own programs or work with a software programmer. Such a programmer will find DATEL's source code very helpful. PCI-416SRC should be thought of as a comprehensive example program to guide the experienced programmer. You may be able to execute your exact application with only minor changes to this source code.

Under special contracted agreement, DATEL will review requests for custom software when associated with quantity orders. If you wish to pursue this path, make your request with *very clear written specifications.* Most likely, you will have the best success with a local, on-site programmer. This will give you the highest degree of control over your project and probably the quickest results. DATEL will coordinate with your local programmer if needed.

1.2 About Microsoft Windows

At maximum A/D sampling rates, the PCI-416 makes serious demands on the performance of your host PC. The PCI-416 is one of the fastest boards you can obtain for any PC system. And advanced data acquisition is one of the most demanding jobs a PC can do. As of this date, Microsoft's Windows operating environment cannot achieve high speed compared to MS-DOS and any Windows PCI-416 programs would cripple the throughput available on this board. This is largely because of the non-deterministic timing of Windows' message queue, meaning that unpredictable delays occur when running normal Windows programs. Also, Windows uses a considerable amount of memory and high disk activity at random times.

While it is possible to run Microsoft Windows in "locked" mode not accessing the message loop, this is considered bad programmng form and may result in unknown system problems. Therefore, use the PCI-416SET MS-DOS program to realize the full speed available on your PCI-416. Newer Windows 95 and Windows NT versions may offer higher PCI-416 speeds but code which makes no calls to Windows will always be fastest.

If you wish to use a Windows post analysis program such as your favorite spreadsheet or math processing software, use PCI-416SET first in DOS mode to save A/D data to disk file. These PCI-416SET files are integer binary which is acceptable to almost all data processing systems. After collecting the A/D files, run your Windows application to process those A/D files. This method retains the full speed of PCI-416SET while giving you the windowed analysis environment you are familiar with. ***DO NOT*** run PCI-416SET under Windows' "MS-DOS" icon since this will degrade the speed, possibly run out of base memory and may cause other problems.

DATEL also offers an ISA board, the PC-414 family, which shares the same identical analog front ends as the PCI-416 *and* has full Windows software available including LabVIEW®.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.3 Application Support

DATEL's staff of experienced engineers will assist you in selecting and using our board products. These Application Engineers can reached by telephone during normal business hours, by facsimile, postal or electronic mail. We will offer advice on how to operate the board, answer application questions of reasonable duration and we'll review short sections of user-written software. If we do not know the answer to a question, it may take some time to research information.

While we can assist you in many ways, we cannot be responsible for your application. When calling us, please organize your questions so that you can complete the call in less than 10 minutes. If you have longer inquiries, please send them in writing. Be very clear about your problem and the sequence of where you are having difficulty. It's always a good idea to send us written questions even if you plan to call by telephone. This gives us time to prepare your answers.

1.4 Suggested Equipment

To operate the PCI-416, you will need the following:

- PCI-416 board, PCI-416SET software disk, PCI-416 User Manuals.
- Your computer, its PCI software and its documentation.
- Input sensors and cabling.

In addition, you may find the following items helpful:

- A set of electronic tools.
- Digital pulse generator (for external trigger or A/D clock)
- "C" compiler/linker/assembler package (PCI-416SET uses Borland)
- Oscilloscope to verify proper trigger or clock pulses and to observe analog signals
- AC/DC Analog signal generator/calibrator to simulate input signals and do DC calibration.
- In-line coaxial passive filter to reduce AC generator harmonics
- PCI extender board and spare DB-25 connectors
- Various probes, wiring and test leads
- Soldering iron to wire connectors, sensors or cables.

Obviously, these should all be used only by skilled persons.

A suitable source for active PCI extender boards is:
Catalyst Enterprises, Inc., 1439 Torrington Court, San Jose, CA 95120, (408) 268-4145

1.5 Static Discharge Protection

IMPORTANT: Although all inputs and outputs on the PCI-416 have reasonable protection when powered up, many components can be permanently damaged by static discharge when handling the board, thereby voiding the product warrany. Electrostatic charges build up especially in dry indoor air in the winter and with clothing or carpeting with high synthetic fiber content.

Here are a few static discharge protection methods:

- Keep the board in its anti-static protective bag until you are just about to install it. When making jumper changes, keep the board on top of this bag. Place a finger or hand continuously in contact with the bag.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- When first handling the board, touch and continuously hold the rear mounting bracket. This connects to the board's ground plane. Then touch and continuously hold the computer's chassis until the board is inserted in the PCI bus slot. If preferred, connect a grounding alligator cable between the PCI-416 mounting bracket and the PC chassis. *Even if you do this, damage can occur if* ***YOU*** *are not also grounded and you touch the board!*

- It is not sufficient to ground yourself against the computer's chassis then let go and handle the PCI-416. In very dry air, thousands of volts of static potential build up very quickly just by walking or sitting or from your clothing.

- Use a grounding wrist strap

1.6 Installation Outline

The PCI-416 needs these items for successful installation:

- First, make sure your PCI computer is running properly *before* installing the PCI-416.
- Copy any necessary files from your *existing* PCI configuration, *before* you install the PCI-416. It's a good idea to copy *all* important files to backup disks or tape in the unlikely event of a system failure.
- Change any jumpers required on the PCI-416 board
- Remove AC power, *and using anti-static protection,* select a PCI bus slot and insert the board.

After these are successfully done, you should run DATEL's PCI-416SET program to verify board operation.

Before beginning installation, study your host computer documentation carefully. Installing the PCI-416 must not disrupt the normal operation of your host PC. If your development environment differs from the actual end application, be aware of these differences. Use the MSD utility and write down all interrupt, DMA and I/O assignments.

The PCI-416 uses *only* +5Vdc power and makes no connection to any +3.3Vdc power systems in your host computer. The PCI-416 will not operate in PCI systems which have *only* +3.3V power but works properly for systems which have *both* +5V and +3.3V. Refer to the PCI specifications for the two different sections of the PCI bus connector for +5V and +3.3V power.

Caution: The A/D converter(s) normally run very warm. Mount the boards vertically if possible and insure adequate airflow. If the board is horizontal (as in a "tower" computer), consider mounting a small fan near the A/D section if the main fan is sufficient.

1.7 Jumpers and Board Components

M1 - Pluggable A/D module. Each unique module is indentified by an "ADM-" model number. See Section 3.
J1 - The DB-25 analog input and trigger input connector.
JP2 - Close for ±5V D/A output. Open JP2 for ±10V D/A output.
R29 - D/A Full Scale calibration adjustment
R28 - D/A Offset calibration adjustment
SG2 - This solder gap will normally be preconfigured. It must be changed only if a different A/D module is installed. When SG2 is open, serial device programming information is daisy-chained through the A/D module. When closed this information bypasses the A/D module.
SG2 open - For models J, K, and L.
SG2 closed - For all other models.
SG1 - This solder gap is not used and is always closed.
JP1 - Not used, don't care

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

U5 - Dual Inline Package switch. See the Command Register descriptions. The module ID (switches 1-4) should normally never be changed. Switch 8 connects a 51 ohm input load to the External Trigger input (J1-13).
CR7 - Diagnostic LED lamp controlled by bit 8 in the A/D Channel Address Register
P2 - 82C55 Parallel Port connector
P3 - On board logic programming port. ***Make no connection!***

D/A Analog Output Range Selection

Output Range	Jumper JP2
±5V	Closed
±10V	Open

1.8 PCI Slot Selection and Host Compatibility

CAUTION: ***PCI bus slots are different !!*** Your computer will probably have *both* ISA (16-bit PC/AT) and PCI slots. Do not force the PCI-416 into a PC/AT slot. The PCI-416 only works in an PCI slot. If you have a choice, position the PCI-416 away from other boards, switching power supplies or disk drives. This will slightly improve the noise performance although you may not detect any difference.

DATEL has also encountered several systems with both PCI and EISA buses. In cases where the EISA bus is derived from the PCI controller (and does not connect directly to the host CPU), both the EISA bus and PCI bus may be slower than normal although it still functions properly. Each system must be tested to determine the actual speed available. Some of these systems degrade the speed by having the PCI bus controller inject extra wait states to the CPU for time to arbitrate with the EISA bus.

1.9 Installation Procedure

WARNING: Disconnect the PC's AC power cable from the wall socket or remove this cable from the PC before working on the computer's interior. *Do not merely shut of the main power switch.* If AC power is present with the computer housing open, you risk **PERSONAL INJURY or DEATH.**

The PC's interior generally is not dangerous, even if the AC power is still applied. But remove the AC cable just to be sure. Be especially careful to *avoid dropping tools or fasteners inside the PC* or making momentary contact to internal components. This may void the product warranty. If you are unfamiliar working with circuit boards and other computer hardware, *get experienced assistance.*

1. Copy DATEL's PCI files to your host hard drive.

2. Open the PC's housing to gain access to the PCI bus connections. Refer to your computer's documentation to do this. Save all fasteners and *remember the sequence that the fasteners were removed.*

3. Find a spare PCI bus slot and insert the PCI-416 directly into the bus connectors. Attach the rear mounting bracket.

4. Reattach the chassis housing and reconnect the AC power cord. Apply power to the computer and boot the system. Verify that the computer boots normally.

5. Run DATEL's PCI-416SET program. Refer to the PCI-416SET description in this manual.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.9.1 Troubleshooting

The PCI-416 is very reliable when installed correctly. Advanced data acquisition is one of the most demanding tasks for high end PC's and is certainly an application that was not considered when most PC's were designed. DATEL is responsible for your PCI-416 board. We cannot be responsible for your computer however lots of advice is listed below.

This information is intended for users who clearly know the limits of their knowledge and can get experienced assistance when they are stuck. The key to troubleshooting is to thoroughly understand what you are doing, make only one change at a time, do things in a logical sequence and keep accurate notes of what is happening. You should be patient, well-organized and persistent to get the most out of these suggestions.

This troubleshooting information cannot possibly include all conceivable sources of difficulty. But it will give you a start and perhaps other areas to consider. Also, this information is not presented in any particular logical order since there are many approaches to troubleshooting. You should read this entire section before deciding where to start. If you have properly performed all the installation steps above and trouble persists, try these fixes:

1. Make sure the **PCI connector** on the PCI-416 is fully seated into the bus sockets. But do not apply so much force that you damage the motherboard or form hairline cracks in the copper etch. Rock the board gently to get it to seat or start with one end slightly offset then gradually work in the full connector.

2. There are a very large number of connections which all must make good contact. Remove the board and use a mild solvent (such as clinical grade isopropyl alcohol) on the PC edgeboard connector. You may also try a "tuner lubricant" (**contact cleaner**) sparingly to remove possible oxide since isopropanol won't cut oxides. Gently wipe off excess lubricant residue. Do not rub so much that you degrade the thin gold plating on the connector fingers. The edgeboard connectors on the PCI-416 and on all computer boards have a limited insertion life. Therefore, once the board is running properly, avoid removing the board too often. Always store the board in its antistatic bag to prevent accumulation of dirt, residue or connector oxidation.

   Once you have the board working under PCI-416SET, **defective input connections** can still cause problems. If necessary, do continuity tests all the way from the input signal PC board etch on the PCI-416 to the signal source. You should understand analog input characteristics thoroughly. For example, the analog low connection is ultimately connected to the PC power ground and may also connect to AC earth ground. This can cause bad data or damage if you connect to inputs with a different grounding system.

3. Some brands of low cost motherboards include lower quality connectors. Sometimes all connections are not thoroughly tested during manufacture. Try **another slot.** PCI is a very high speed bus. DATEL has noticed that certain combinations of boards in various PCI slots may change the transmission characteristics of the bus even though they are all supposed to follow PCI electrical standards. And mixed controller or transceiver chips drive the bus in different ways depending on whether the PCI-416 is closer or further away from the PCI controller chips on the motherboard.

   Remove all unneeded boards (and their software drivers) which are not absolutely necessary to the system. Do you really need that LAN board during A/D sampling? Can you live without your CD-ROM drive for awhile? DATEL has noticed particular problems with SCSI disk controllers. But this is usually fixed by reassigning DMA or interrupts. Try getting the PCI-416 running with very few boards then add more boards plus their drivers and continually retest the system.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When you remove and reinstall boards, turn off the power fully each time. ***DO NOT insert boards while the power is on!*** And allow a few seconds for the power to completely die away.

Many sections of the PC must initialize properly in the correct sequence for the system to work. Occasionally, the system may not boot successfully. When in doubt, press the **hardware reset button** (or turn off power for a full 5 seconds). Don't just try to reboot using CTRL-ALT-DEL.

4. Try different **combinations of interrupts**. Microsoft's MSD.EXE utility reports most (but not all) possible conflicts. Some hosts initialize these systems in sequential or slot order and a break in device numbering or slot occupancy may cause problems. Other systems make strange assumptions about device assignments, won't initialize if they think the device is already set up, use unknown initialization sequences, may not load properly or can't find the files (but won't tell you).

   Do not run PCI-416SET under the Windows MS-DOS prompt.

5. Examine the **CONFIG.SYS and AUTOEXEC.BAT** files very closely. If you do not thoroughly understand all aspects of these files, get experienced assistance. Rearrange the sequence of device initialization in these files if necessary. Look for files not in the correct directories, improper PATH assignments, etc. Delete any lines that are not necessary including **EMM386.EXE and SMARTDRV**. You should leave HIMEM.SYS loaded and if you are using an equivalent memory manager, try deleting it and going back to HIMEM.SYS, at least temporarily. You probably do not have a memory problem if you don't get error messages. If in doubt, reload files from the original diskettes.

   DATEL once had a difficult problem with an AUTOEXEC.BAT file. The bug was traced to a missing carriage return character at the end of the last line. The bug could not be seen with a text editor. Extremely simple, apparently obvious problems can stop the system.

6. One of the better tests is to have **two or more of everything**. Try both PCI-416's in two separate computers. Then swap boards if you still have a problem. All PCI-416's are thoroughly tested at DATEL before shipment. However, failures are possible during transportation or from other causes. Your board may be **retested** at any time by DATEL. If you have two identical boards, test both of them in exactly the same way.

   For applications running two or more PCI-416's *simultaneously* verify proper operation first with only one board installed at a time. Once all boards have been tested singly, then try multiple boards.

7. You should be experienced with setting up and running high performance PC's including thorough knowledge of how MS-DOS works, file systems, and peripherals. Your system should be well-maintained with **no file errors**. Run the SCANDISK utility or 3rd party disk checkers such as the Norton utilities. If you use a **disk compression** system, run its diagnostic frequently. (Disk compressors make the disk system *even slower* for saving A/D data). Try running the system without the disk compressor (although this is rarely a problem). **Defragment** your disk periodically. You should be thoroughly familiar with Microsoft's MSD.EXE configuration utility and MEM.EXE. And you should be running recent versions of the DOS and all other files. Also kill any files with **illegal filenames.**

8. Be realistic about how much **remote debugging** can be done over the phone and FAX. Even if DATEL has your same brand of computer, we probably will have a different mix of peripherals, configuration and initialization files. If you want our help, we will need clear copies of your CONFIG.SYS and AUTOEXEC.BAT files and revision levels of everything. We cannot provide much help unless you are reasonably **knowledgeable about high end PC's** and understand basic electronic troubleshooting..

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9. All DATEL diagnostic assistance revolves around **PCI-416SET**. If you are writing your own code, we must see that PCI-416SET is running properly *before* discussing problems with your software. PCI-416SET is used by a many customers in different applications. The program is relatively high performance and carefully maintained, and while it is not perfect, bugs are vigorously documented, pursued and fixed.

   If you have written a large section of special PCI-416 code, you should find a local software programmer for assistance, even if that code is derived from PCI-416SRC. We can successfully diagnose small sections of user-written code in languages we support.

10. Obviously, your copy of PCI-416SET and all other system files should be *perfect*. If there is the slightest doubt, don't hesitate to **reload everything** (yes, including MS-DOS) from the original diskettes. Your MS-DOS version should be 5.00 or later. Make sure all the disks are flawless. New copies of PCI-416SET/SRC are available from DATEL. This is lots of work but should be seriously considered as a last resort. Just because the system boots up with no errors does not mean that a file is not defective or the memory image is not corrupted.

11. All high end PC's are totally dependent on their BIOS code. Fortunately, most major **BIOS ROM** suppliers (Award, etc.) offer reputable, well-tested code. The PCI-416 uses some of the more recent BIOS extensions for extended memory access, finding the PCI configuration, etc. BIOS errors are unlikely but possible. The BIOS revision is shown very briefly during bootup and can also be seen during the "Setup" screen, often invoked with a special key sequence *before* bootup. Copy down the revision levels and serial numbers if you call DATEL. We probably will not have your same configuration but these items are needed when calling the BIOS supplier. MSD also shows BIOS information.

    As of this date, no known BIOS problems have appeared. If you write your own code, be aware of subtle BIOS differences. Some functions return with unspecified registers or buffers containing undefined data. They are within spec but do not operate as expected. Remember to thoroughly save then set all needed registers before a function call. Also be aware of possible documentation errors by cross-checking several references.

12. High performance PC's are very complex systems designed by numerous people over many years and different locations. The engineers all read the design documentation at different revision levels and sometimes not in their native language. Other board manufacturers can make design mistakes occasionally or don't follow the rules for registers, bus loading and timing and BIOS calls. If you add a board which causes the failure after success with PCI-416SET, sometimes it is impossible to get a full reason for the failure.

13. Your AC **power supply** should be stable and within specification. Is there enough power available for all the boards *and peripherals?* Use an oscilloscope (be careful!!) to look for spikes on the AC line, sometimes indicating a power suppy near its capacity. Similarly, high RF fields or electromagnetic pulses may cause problems in your PC and analog inputs. Also look at the DC power lines inside your PC. They should be clean with mimimal switching noise. Check for loose, corroded or deficient ground and power connections. Examine all flat cables, clean and reseat them if necessary.

14. If you have trouble with fast **DMA** transfers, see if the system works with slower program (status) mode or if you decrease the CPU speed. If lower speed succeeds, you may need faster DRAMS. DATEL has also encountered slow DRAM controllers (not the memory chips themselves).

15. Go away for awhile and think very clearly about your application. The unconscious mind of a good engineer will sometimes continue to think about possible solutions. Occasionally it is more than coincidence that a problem is solved when you are not at the computer.

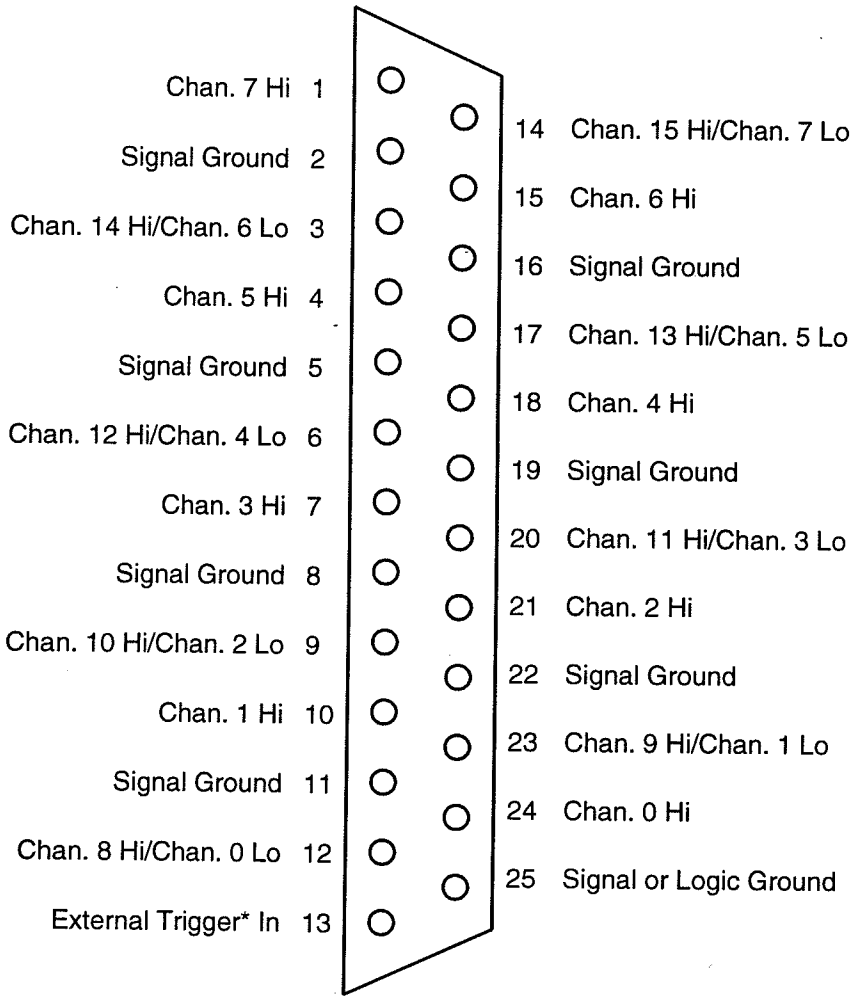
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SECTION 2

INPUT/OUTPUT CONNECTIONS AND BOARD LAYOUT

2.0 Analog Input Connector

PCI-416 models use the DB-25S rear panel connector. Pin numbering is shown as viewed from the rear panel:



Channel addresses use the notation, single-ended/differential.

Figure 2.0 DB-25S Connector

Some models may use SMA connectors, listed following:

Five miniature coaxial threaded "SMA" connectors are mounted on the rear plate of some PCI-416 models. The four bottom connectors are for the analog inputs. For models that use a subset of these four, the other inputs are not connected. The top connector is for the external trigger input. An SMA-to-BNC male to male adapter cable, one meter long, (model 61-7342340) is available from DATEL. One cable is required per channel.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

For 50 ohm terminations to reduce line reflections and standing wave errors, use an external 50 ohm in-line shunt, immediately adjacent to the SMA connector. Note that the 50 ohm termination will reduce the signal amplitude seen by the PCI-416. The remote driving source must also be 50 ohm impedance.

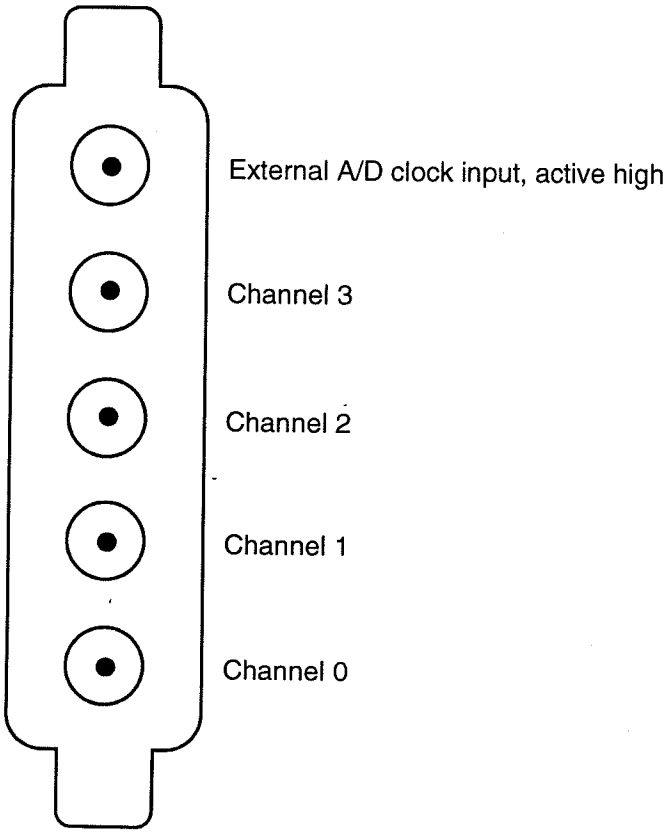


Figure 2.1 Rear Mounting Plate

SMA Connectors

DATEL offeres an SMA-to-BNC coaxial cable, model number 61-7342340, for input connections. If you prefer to fabricate your cables, the following two manufacturers offer the SMA male coaxial fitting. Their local distributors are listed in the EEM directory. These fittings are not available from DATEL.

E.F. Johnson	142-0361-006
Pomona	4859

Pasternak Enterprises, Irvine, CA (714) 261-1920 offers cables and fittings.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.1 82C55 Digital I/O Port - P2

24-bit digital data is presented on one 50-pin female flat cable header connector. Several other functions appear on this connector including the DAC analog output, Marker input, and Analog Trigger input. Refer to 82C55 programming information in Section 6.2 and Appendix A. Except for the analog connections, the pinout of this port is compatible with several popular stand-alone digital I/O boards for the PC. A small amount of +5V dc power may be extracted from this connector to power external circuits.

Port A, Bit 7	1	2	Logic Ground
Port A, Bit 6	3	4	Logic Ground
Port A, Bit 5	5	6	Logic Ground
Port A, Bit 4	7	8	Logic Ground
Port A, Bit 3	9	10	Logic Ground
Port A, Bit 2	11	12	Logic Ground
Port A, Bit 1	13	14	Logic Ground
Port A, Bit 0	15	16	Logic Ground
Port C, Bit 7	17	18	Logic Ground
Port C, Bit 6	19	20	Logic Ground
Port C, Bit 5	21	22	Logic Ground
Port C, Bit 4	23	24	Logic Ground
Port C, Bit 3	25	26	Logic Ground
Port C, Bit 2	27	28	Logic Ground
Port C, Bit 1	29	30	Logic Ground
Port C, Bit 0	31	32	Logic Ground
Port B, Bit 7	33	34	Logic Ground
Port B, Bit 6	35	36	A/D Clock Out
Port B, Bit 5	37	38	Logic Ground
Port B, Bit 4	39	40	Internal Trigger* Out
Port B, Bit 3	41	42	Logic Ground
Port B, Bit 2	43	44	Marker* Input
Port B, Bit 1	45	46	Analog Trigger Input
Port B, Bit 0	47	48	Analog Ground
+5V pwr out (.5A max)	49	50	DAC Analog Output

"*" means low true

Figure 2.1 82C55 Digital I/O Port - P2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.2 Jumpers

There are several jumpers on the board which may require changing before installation. Jumpers consist of a fixed set of upright gold-plated pins mounted on the circuit board and a removable "Berg" plug which slides off vertically. The plug is "connected" (circuit "closed") if two pins are both inserted inside the plug. If the plug is to be disconnected, it can be conveniently stored on a *single* pin by mounting it *sideways*. This will avoid losing the plug. Jumpers offer superior electrical characteristics in a very small area when software configuration cannot be used or would offer inferior performance. These are listed following.

2.2.1 Board Component Layout

CAUTION: Handle the board only with antistatic protection. Most items are well labelled using the PC board silk screening. The following diagram shows important components in case the screening is not clear.

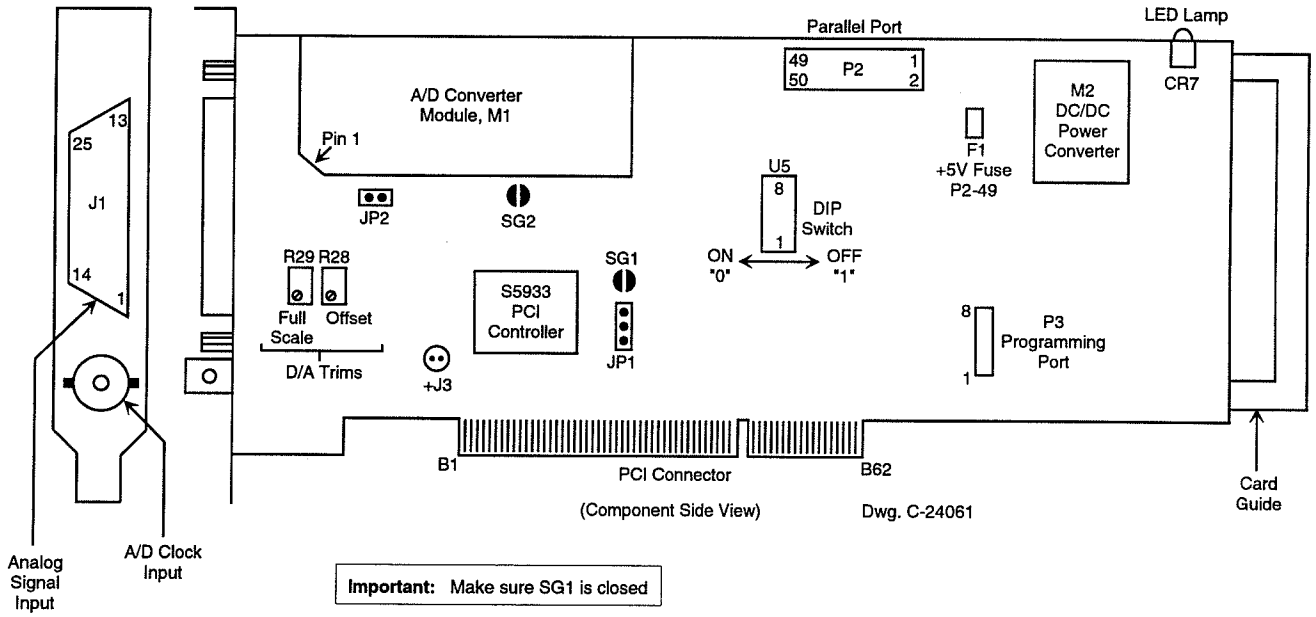


Figure 2.2 Board Layout (drawing not to scale)

2.2.2 Jumpers and Board Components

M1 - Pluggable A/D module. Each unique module is indentified by an "ADM-" model number. See Section 3.
J1 - The DB-25 analog input and trigger input connector.
JP2 - Close for ±5V D/A output. Open JP2 for ±10V D/A output.
R29 - D/A Full Scale calibration adjustment
R28 - D/A Offset calibration adjustment
SG2 - This solder gap will normally be preconfigured. It must be changed only if a different A/D module is installed. When SG2 is open, serial device programming information is daisy-chained through the A/D module. When closed this information bypasses the A/D module.

SG2 open - For models J, K, and L.
SG2 closed - For all other models.

SG1 - This solder gap is not used and is always closed.
JP1 - Not used, don't care
U5 - Dual Inline Package switch. See the Command Register descriptions. The module ID (switches 1-4) should normally never be changed. Switch 8 connects a 51 ohm input load to the External Trigger input (J1-13).
CR7 - Diagnostic LED lamp controlled by bit 8 in the A/D Channel Address Register
P2 - 82C55 Parallel Port connector
P3 - On board logic programming port. ***Make no connection!***

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SECTION 3

ANALOG MODULES

3.0 Introduction

The "ADM" series modules are the analog sections for DATEL's PC-414, PCI-416, PC-430, PC-415, DVME-614, and DVME-630 series high performance data acquisition boards. DATEL designs and manufactures these modules using DATEL standard data acquisition microcircuits and other components.

All modules consist of one or more DATEL sampling S/H-A/D converters, multiplexers, input buffers, and other circuits. The channel address sequencing logic (CA0..CA3) is always external to the module and is limited to 4 address bits (16 input channels max.). All modules share an identical signal pinout. Therefore, one module may be exchanged with any other module to provide different A/D front-ends on the A/D carrier board. The same software runs on the base-board - with the exception of sample rate adjustments for different modules. All modules may be ordered from DATEL, Inc. as separate model numbers, e.g. ADM-12A, ADM-14G, etc.

The A/D converter connects to the outside world either through an amplifier on the module or directly. There are no other active analog input circuits on the main carrier board. The module's digital outputs are always buffered but are not tri-stated. The only critical area of timing is the A/D start input. Although the modules are designed to feed a FIFO memory on the carrier board, the outputs may be used in other ways to suit your application. The A/D digital outputs can drive up to 4 TTL loads without buffering.

The ADM-12F, ADM-8SIM, and ADM-14G modules use multiple simultaneously triggered A/D converters. There is no channel multiplexer, the single channel and multiple channel rates are the same. Local PAL logic decodes whether one or all A/D's are triggered simultaneously. This logic also enables each of the A/D outputs alternately.

The various modules available are:

ADM-12A	4 channels, 12 bit, Simultaneous Sample/Hold (SS&H), 1.5 MHz (single channel), 1 MHz (multi-channel).
ADM-14B	4 channels, 14 bit, 500 kHz (single channel), 333 kHz (multi-channel).
ADM-12C	4 channels, 12 bit, 1.6 MHz (single channel), 500 kHz (muti-channel).
ADM-12D	1 channel, 12 bit, 5 MHz.
ADM-12E	16S/8D channels, 12 bit, 2MHz (single channel), 500 kHz (multi-channel).
ADM-12F	2 channels, 12 bit, 2 MHz per channel SS&H - A/D converter on each channel.
ADM-14G	2 channels, 14 bit, 1 MHz per channel SS&H - A/D converter per channel.
ADM-12H	1 channel, 12 bit, 10 MHz - needs external A/D start source to achieve 10 MHz.
ADM-8SIM-1 ("J" models)	8 channel simultaneous 12-bit A/D, 500 kHz per channel, ±5V or ±10V input.
ADM-8SIM-2	Same as ADM-8SIM-1 except 0 to +2.5V input.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The A/D board model number indicates which ADM module is used, e.g. a "B" type uses the ADM-14B, an "F" carrier uses the ADM-12F, etc.

DATEL is continually designing new ADM modules - different A/D speeds, channel densities, resolution, isolation, etc. Contact DATEL for special requirements.

3.1 Physical Characteristics

All ADM modules are designed to mount vertically on carrier boards. The A/D gain and offset adjustment potentiometers are then easily accessible when the card is installed in the host computer. Customers planning to use these modules in their own applications should obtain the DATEL board manuals and study the base boards carefully.

All ADM modules are constructed as follows:

Construction	2 or 4 layer, epoxy printed circuit board.
Outline Dimensions	2.0" x 4.0" x 0.375" (5.08cm x 10.16cm x 0.953cm).
Board Thickness	0.0625" (0.159cm)
Mounting Pins	Dual row, 76 pin, pins 1, 38, 39, and 78 are not normally installed.
Pin Dimensions	0.25" (0.64cm) square, gold plated, male.
Pin Height (Above PCB)	0.3175" (0.806cm), component side.
Clearance (Solder side)	0.15" (0.38cm)
Pin Spacing	0.10" (0.25cm)
Row Spacing	1.8" (4.57cm)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The standard ADM module pinout is illustrated in Figure 3.0.

Not Installed	39	38	Not Installed
Not Used	40	37	+15V dc In (250 mA max.)
A/D Bit 15 (MSB) Out	41	36	Power Common
A/D Bit 14 Out	42	35	Power Common
A/D Bit 13 Out	43	34	-15V dc In (250 mA max.)
A/D Bit 12 Out	44	33	Not Used
A/D Bit 11 Out	45	32	Not Used
A/D Bit 10 Out	46	31	Key Way
A/D Bit 9 Out	47	30	Not Used
A/D Bit 8 Out	48	29	Not Used
A/D Bit 7 Out	49	28	Not Used
A/D Bit 6 Out	50	27	Not Used
A/D Bit 5 Out	51	26	Not Used (+10V ref., ADM-12A)
A/D Bit 4 Out	52	25	Not Used
A/D Bit 3 Out	53	24	Not Used
A/D Bit 2 Out	54	23	Not Used
A/D Bit 1 Out	55	22	Not Used
A/D Bit 0 Out	56	21	Not Used
A/D Data Enable* ①	57	20	Not Used
A/D Data Enable* ①	58	19	Analog Return
Note 9 Not Used	59	18	Ch15 Hi/7 Lo In
Note 9 Not Used	60	17	Ch14 Hi/6 Lo In
ACQ In ⑥	61	16	Ch13 Hi/5 Lo In
A/D Start Conversion In ③	62	15	Ch12 Hi/4 Lo In
A/D Start Convert Out	63	14	Ch11 Hi/3 Lo In
A/D End of Conversion* Out	64	13	Ch10 Hi/2 Lo In
Digital Reset* In	65	12	Ch9 Hi/1 Lo In
Simultaneous Sample/Hold In ⑤	66	11	Ch8 Hi/0 Lo In
Channel Address 0 ④	67	10	Analog Return
Channel Address 1	68	9	Ch7 Hi In
Channel Address 2	69	8	Ch6 Hi In
Channel Address 3	70	7	Ch5 Hi In
MUX Enable* In ①	71	6	Ch4 Hi In
Digital Ground	72	5	Ch3 Hi In
Digital Ground	73	4	Ch2 Hi In
+5V dc In (500 mA max.)	74	3	Ch1 Hi In
+5V dc In (500 mA max.)	75	2	Ch0 Hi In
Not Installed	76	1	Not Installed

Figure 3.0 ADM Module Pinout

Notes:

1. Pins 57 and 58 were intended to select or de-select (tri-state) the A/D data outputs for applications with more than 16 channels, i.e. another analog module sharing the data bus. The logic for this has never been implemented. Many DATEL A/D's used in these modules are tri-stated. Most modules include buffers for sign extension and therefore are not tri-stated. On DATEL carrier boards, (PC-414, PCI-416, PC-430, DVME-630, and DVME-614), pins 57, 58, and 71 are connected to ground.

2. The analog inputs are shown as single-ended/differential. For single-ended, inputs are referred to analog return (pins 10 and 19). Most inputs exhibit a very high input impedance (100 Megohms).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3. The A/D Start Conversion input requires controlled pulse width for each A/D module type. 125 nanoseconds is adequate for most modules. Conversion starts on the leading edge of A/D Start Convert Out (pin 63). The trailing edge of this convert output is typically used for a short internal acquisition delay and indicates the exact time that the internal Sample/Hold goes into hold mode.

4. Pins 67 through 70 are the channel address inputs. Allow adequate settling time after changing the channel address before starting the next A/D conversion. This delay depends on the module - see individual specifications later.

5. The SSHSEL input is used by the ADM-12A only to place all S/H's in hold mode ("1") or sample mode ("0"). This signal is typically driven by End of Scan (EOS) or ACQuire signal on the carrier board.

   On the carrier boards, this signal is set by a bit in the local Command Register.

6. ACQ is decoded by ADM-12F & G only. It is necessary to allow sufficient A/D conversion time and to avoid premature FIFO transfers for these high speed, dual channel models.

   Pin 61 (high) is used to inhibit the first EOC on the F and G modules so that the first sample is ignored and does not get loaded into the FIFO. This is necessary because of the multiplexed data loading of the FIFO.

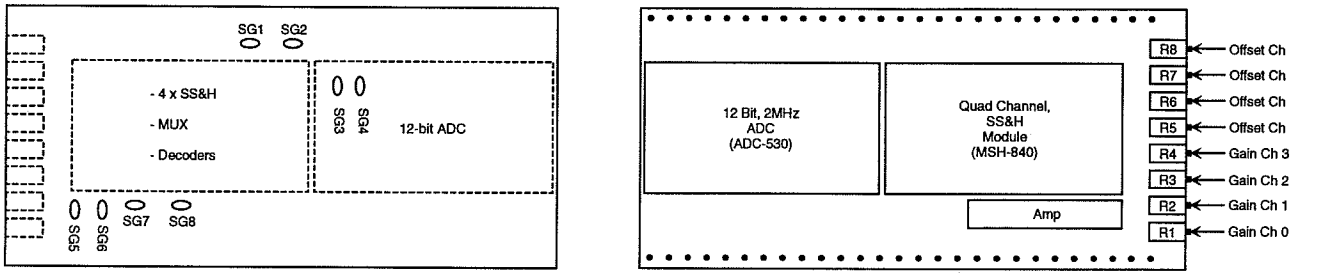
7. Some older ADM-12A modules use pin 26 as a +10V reference. Do not connect.

8. See the latest board data sheet for ADM specifications.

9. Pins 59 and 60 are used for internal ADM testing on newer ADM's.

3.2 ADM-12A (PCI-416A)

The ADM-12A is a quad channel, simultaneous sample/hold (SS&H) A/D module. It contains two DATEL hybrid components. The MSH-840 is a quad SS&H featuring an acquisition time of 900 nanoseconds maximum. A four channel multiplexer allows individual S/H outputs to be selected. The ADC-530 is a 12-bit A/D converter with a maximum conversion time of 350 nanoseconds. When the module goes from sample into hold mode all four analog input channels are held simultaneously - common strobe signal. Using the channel address lines (CA0..3) each S/H channel is sequentially digitized by the ADC-530. After a channel advance, sufficient MSH-840 multiplexer settling must be allowed before starting the next conversion.



(a) Solder Side **(b) Component Side**

Figure 3.1 ADM-12A Layout

Options such as input voltage range selections and A/D data coding are configured by various solder gaps on the non-component side of the board. There is also a x10 gain selection on channels 3 and 4 only.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Solder Gap	Configuration	Solder Gap Status
SG1	Gain = x10 (Channel 3)	Open
SG2	Gain = x10 (Channel 4)	Open
SG3	±10V	Closed Standard
SG4	0 +10V	Open
SG5	Zero Extension	Open
SG6	Sign Extension	Closed
SG7	2's Complement Binary	Closed
SG8	Offset Binary	Open

Table 3.0 ADM-12A Solder Gap Selections

ADM-12A Timing

The following timing diagram illustrates ADM-12A operation in simultaneous sample/hold (SS&H) mode.

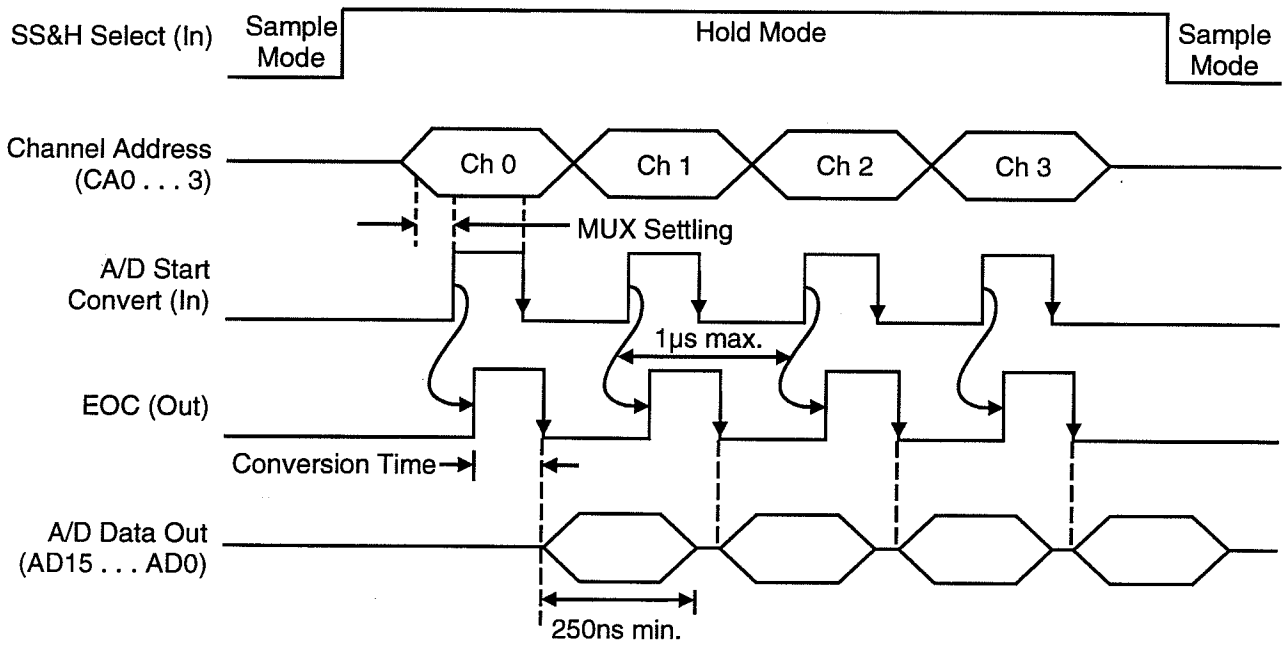


Figure 3.2 ADM-12A SS&H Timing

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3 ADM-14B (PCI-416B)

Quad channel, sequential sample hold module with 14 bit ADC resolution. It uses DATEL's ADS-927, 14 bit, 500 kHz sampling A/D converter. The sample hold modulus is internal to the ADC to maximize performance. The analog inputs may be sampled sequentially or sampling may continue on one channel at higher speed (no multiplexer settling) - controlled by address lines CA0..3.

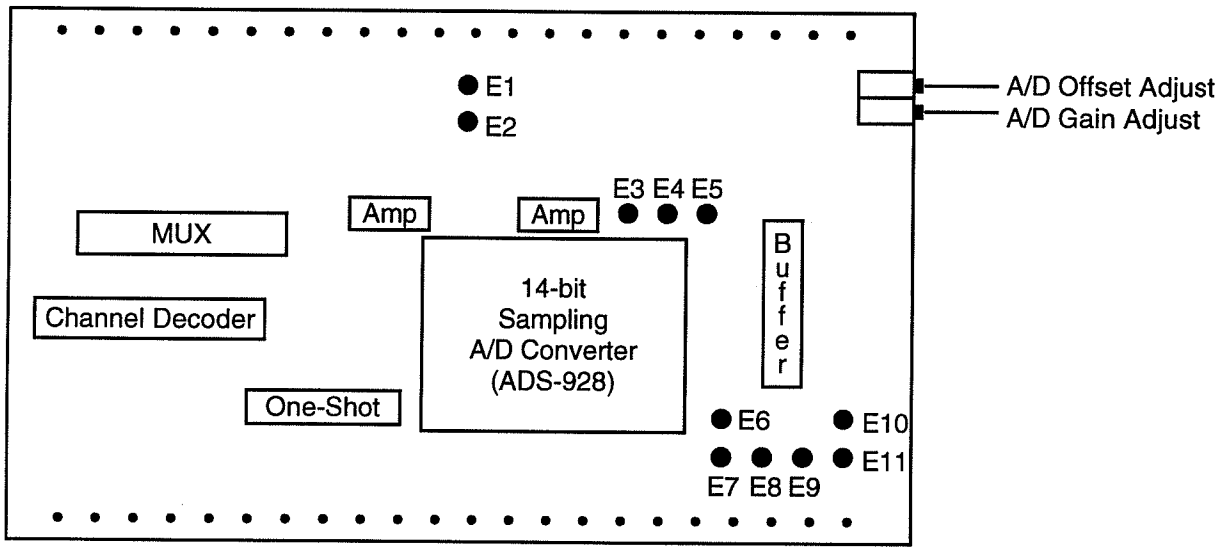


Figure 3.3 ADM-14B Layout (component side)

Options such as input voltage range selections and A/D data coding are configured by jumpers on the component side of the board as follows:

Jumper	Selection
E4-E5	±10V (standard)
E1-E2, E4-E5	±5V
E1-E2, E3-E4	0 to +10V
E7-E8, E9-E11	2's Complement (standard)
E6-E7, E10-E11	Offset Binary or Unipolar

Table 3.1 ADM-14B Jumper Selections

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ADM-14B Timing

The following timing diagram illustrates ADM-14B timing.

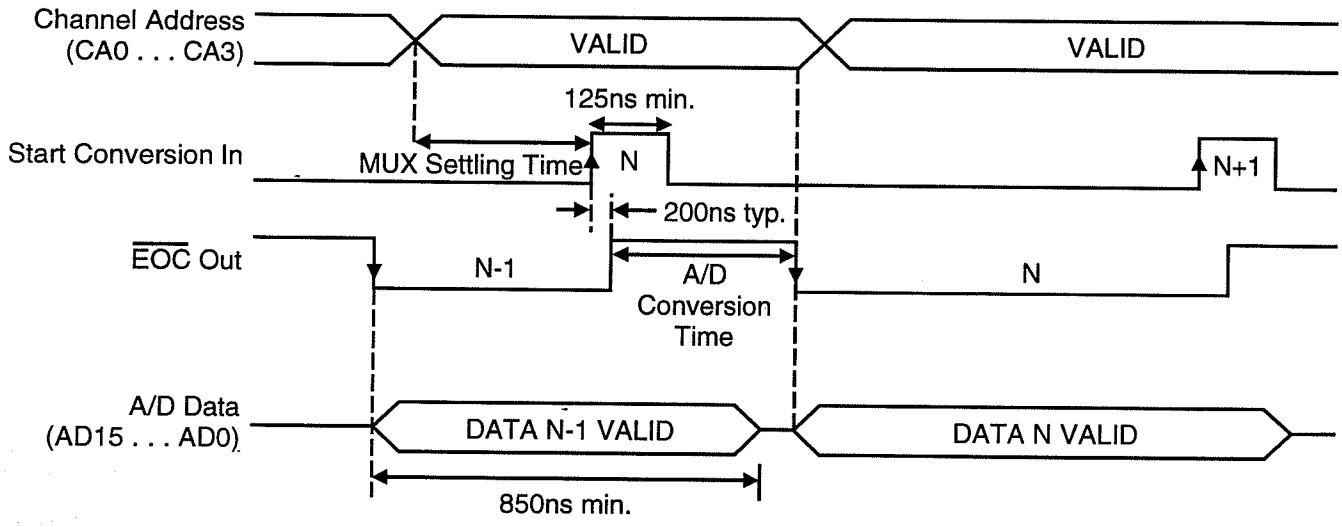


Figure 3.4 ADM-14B Timing

3.4 ADM-12C (PCI-416C)

Quad channel, 12 bit A/D module. Uses DATEL's ADS-117 sampling A/D converter (internal sample/hold). Channel sampling may continue at 500 kHz on one channel only. Multi-channel sampling can be done at 333 kHz to allow sufficient multiplexer settling.

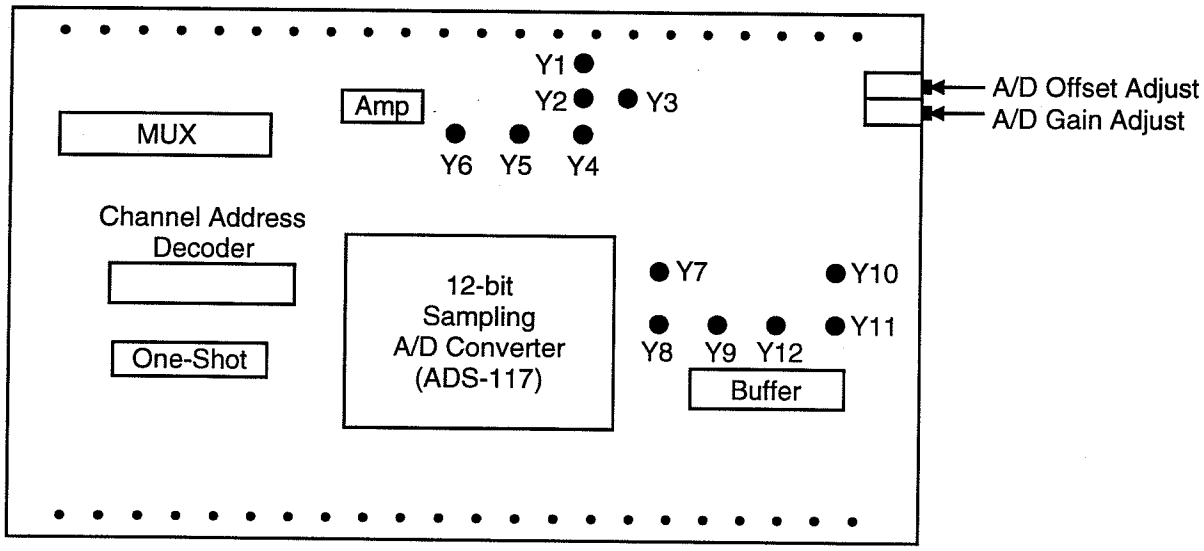


Figure 3.5 ADM-12C Layout (component side)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Options such as input voltage range selections and A/D data coding are configured by jumpers on the component side of the board as follows:

Jumper	Selection
Y1-Y3, Y4-Y5	±10V (standard)
Y1-Y2, Y4-Y5	±5V
Y1-Y2, Y5-Y6	0 to 10V
Y7-Y8, Y10-Y11	Unipolar or Offset Binary
Y8-Y9, Y11-Y12	2's Complement (standard)

Table 3.2 ADM-12C Jumper Selections

ADM-12C Timing

The following timing diagram illustrates ADM-12C operation.

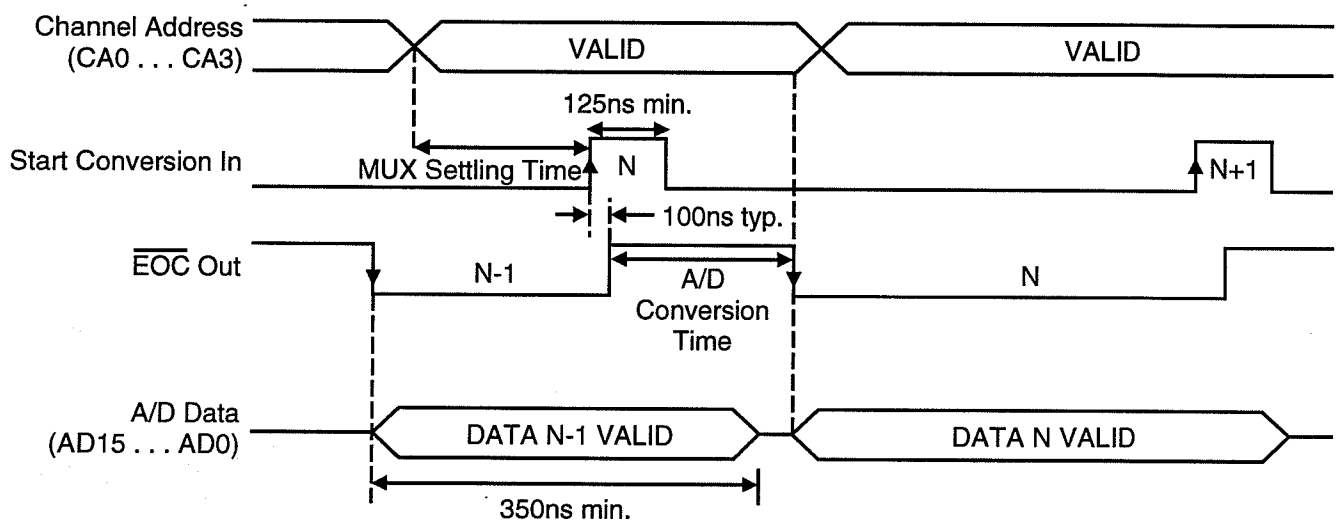
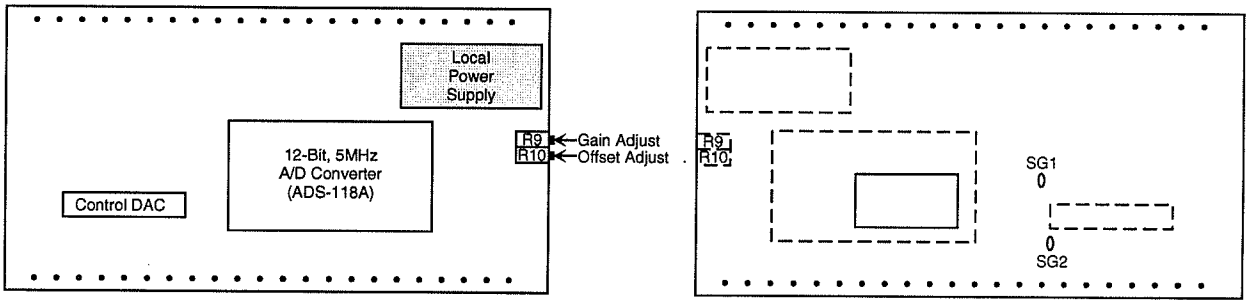


Figure 3.6 ADM-12C Timing

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 ADM-12D (PCI-416D)

Single channel, 12 bit A/D module. Uses DATEL's ADS-118A sampling A/D converter (internal sample/hold). Channel sampling may continue at up to 5 MHz.



(a) Solder Side **(b) Component Side**

Figure 3.7 ADM-12D Layout

The A/D data coding is configured by solder gaps on the component side of the board as follows:

Solder Gap	Status	Selection
SG1 & SG2	Closed	Straight Binary
SG1 & SG2	Open	2's Complement (standard)

Table 3.3 ADM-12D Solder Gap Selections

ADM-12D Timing

Figure 3.8 illustrates ADM-12D timing. Because it is a single channel module the channel address is always locked on 0.

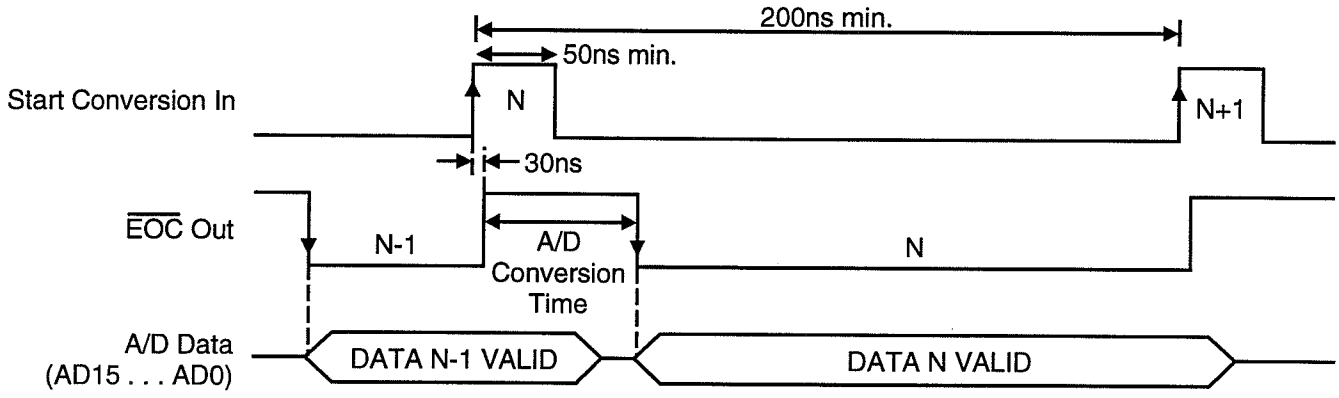
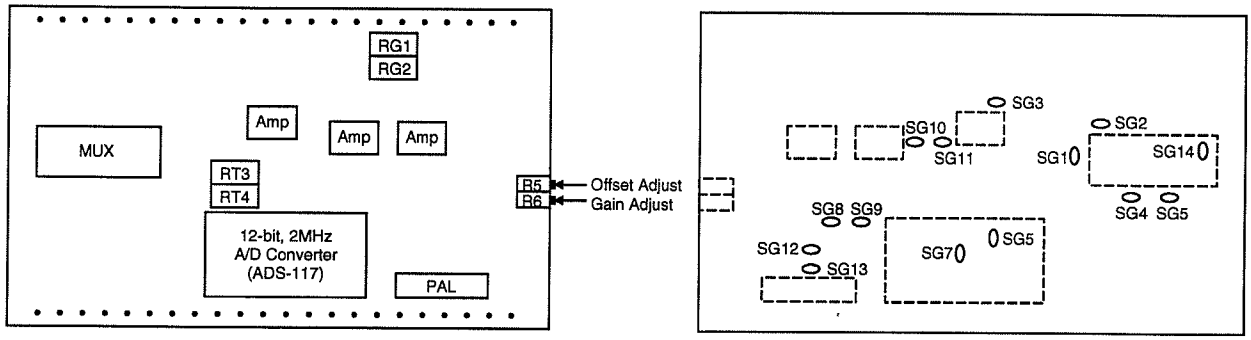


Figure 3.8 ADM-12D Timing

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6 ADM-12E (PCI-416E)

16 single-ended/8 differential input channels, 12 bit A/D. Uses DATEL's ADS-117 sampling A/D converter (internal sample/hold). Single channel sampling may continue at up to 2 MHz, multi-channel sampling at 500 kHz (to allow sufficient multiplexer settling).



(a) Solder Side **(b) Component Side**

Figure 3.9 ADM-12E Layout

The A/D data coding is configured by solder gaps on the component side of the board as follows:

Solder Gap	Selection
SG6, SG8, SG10	–10V to +10V (standard)
SG6, SG9, SG11	–5V to +5V
SG7, SG9, SG11	0 to 10V
SG1, SG3, SG5	Single Ended (standard)
SG2, SG4	Differential
Open SG12	2's Complement
SG12	Straight Binary

Table 3.4 ADM-12E Solder Gap Selections

PCI-416E: The input instrumentation amplifier gain is resistor programmable for gains from x1 to x100. The gain resistor designations are RG1, RG2, and RG3 - refer to Figure 3.5. These are configured with RG1 in parallel with RG2 and RG3 in series with the combination. The total gain resistance may be calculated from the following equation:

$$\text{RGAIN (total)} = 4000/(\text{Gain} - 1)\ \text{Ohms},$$
$$\text{where RGAIN (total)} = [1/(1/\text{RG1} + 1/\text{RG2})] + \text{RG3}$$

The recommended procedure is to short out RG3 by connecting a wire across its pads. This forces RG3 = 0 in the above equation. RG1 should be a low drift, high precision resistor (0.1% tolerance) and RG2 is used to fine tune the gain.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ADM-12E Timing

The following timing diagram illustrates ADM-12E operation.

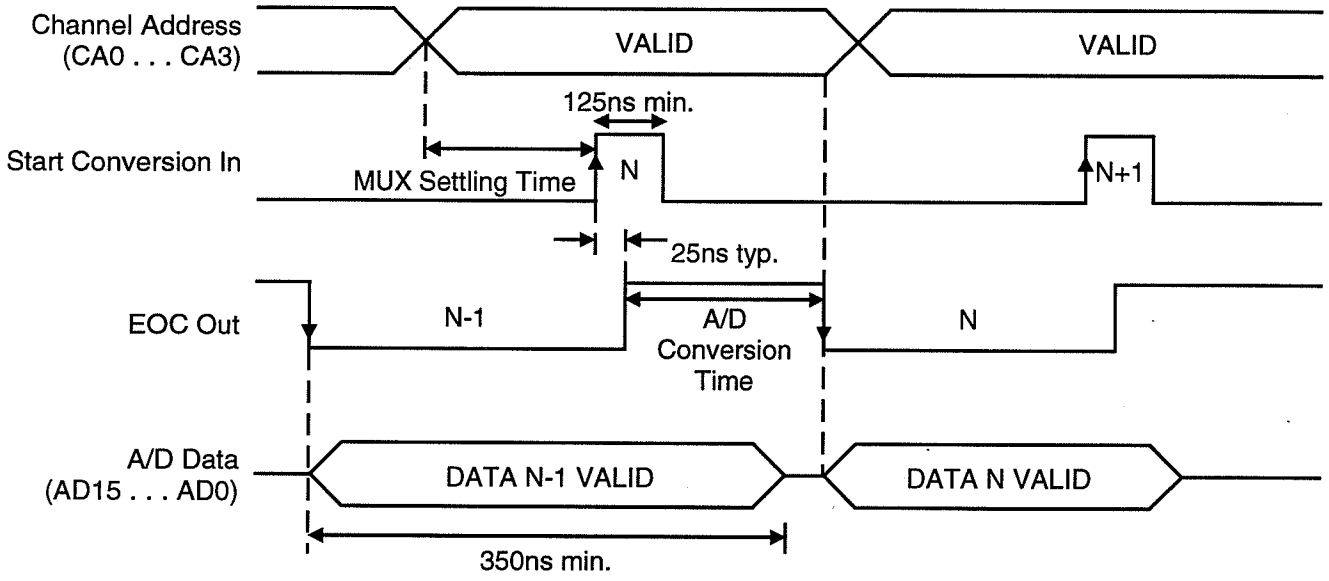
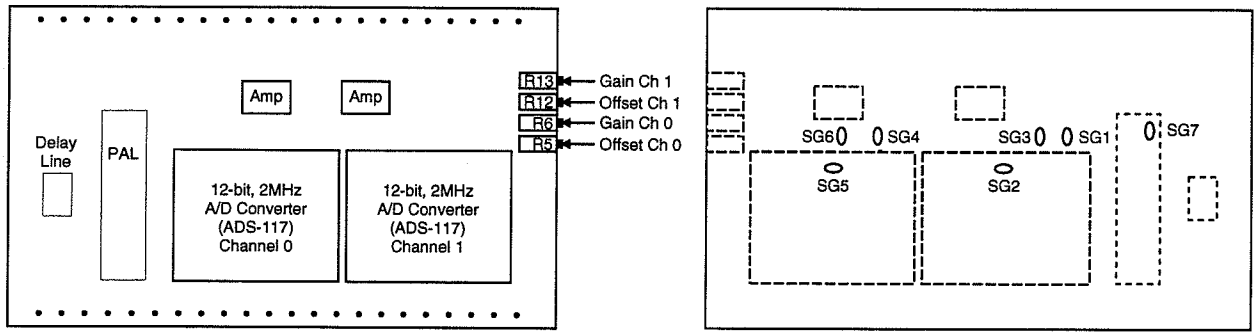


Figure 3.10 ADM-12E Timing

3.7 ADM-12F (PCI-416F)

2 single-ended input channels, SS&H with 12 bit sampling ADC one each channel. Uses two of DATEL's ADS-117 sampling A/D converters (internal sample/hold). Single channel sampling may continue at up to 2 MHz. Because there is a converter on each channel dual channel simultaneous sampling also runs at up to 2 MHz per channel. Both ADC's are strobed simultaneously. Local logic on the module (in the PAL) enables each channel alternately - onto the output lines (AD0..15). Running at full speed (2 MHz per channel) the aggregate system throughput is 4 MHz. The ACQuire signal is decoded to synchronize sample transfer into the FIFO.



(a) Solder Side **(b) Component Side**

Figure 3.11 ADM-12F Layout

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The input voltage ranges and A/D data coding are configured by solder gaps on the component side of the board as follows:

Solder Gap	Selection
SG1 & SG4	±5V Input
SG2 & SG5	0 to 10V Input
SG7 (closed)	Unipolar Straight Binary Out
SG7 (open)	2's Complement Coding (standard)

Table 3.5 ADM-12F Solder Gap Selections

Note: Dual channel, SS&H mode is selected by writing 2 into the Channel Address Register (BASE + 2). All other registers should be configured for single channel acquisition.

Channel Selected	Address Code CA1	CA0
Channel 0 only	0	0
Channel 1 only	0	1
Channels 0, 1 Triggered simultaneously	1	X

Table 3.6 Channel Addressing

Notes:
1. Channel address bits 2 and 3 are not used. Do not use auto-increment mode in the Command Register.
2. In simultaneous mode, channel 0 is loaded into the FIFO first. The data output sequence is: chan 0, chan 1, chan 0, . . . For example, at 1 MHz sample rate, the FIFO load rate is 2 MHz.

ADM-12F Timing

To select dual channel, simultaneous sample/hold mode the channel address must be locked on 2. The timing is as follows:

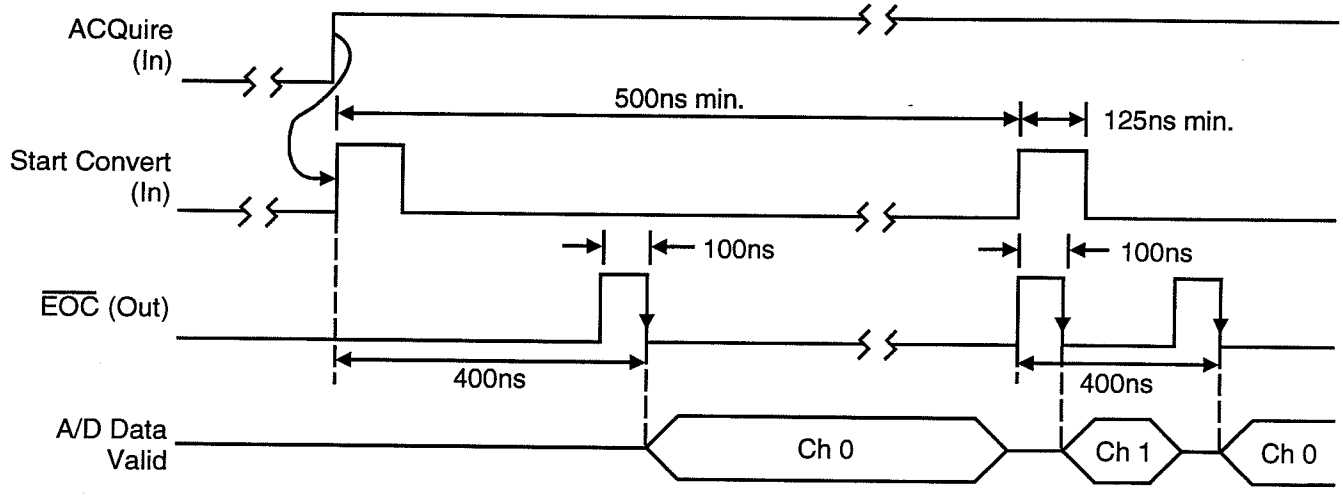


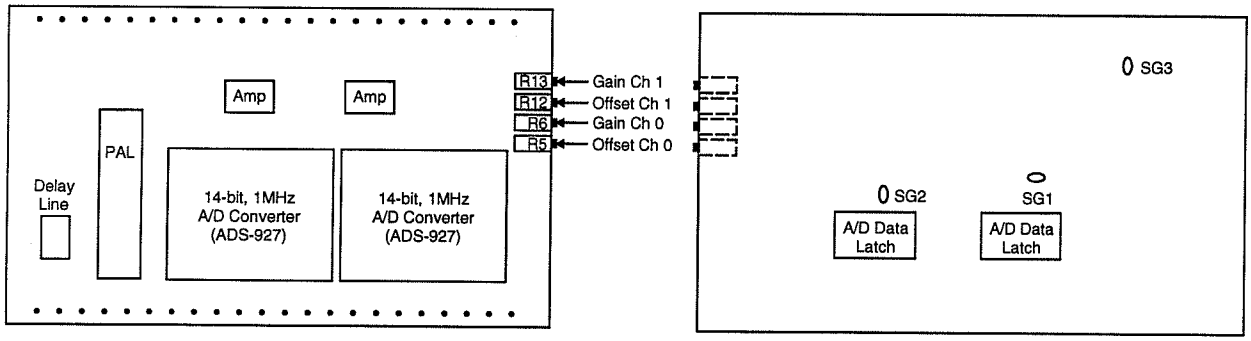
Figure 3.12 ADM-12F SS&H Timing

In single channel mode the first EOC in each pair is inhibited - one sample per Start Convert input.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.8 ADM-14G (PCI-416G)

2 single-ended input channels, SS&H with 14 bit sampling ADC one each channel. Uses two of DATEL's ADS-927 sampling A/D converters (internal sample/hold). Single channel sampling may continue at up to 1 MHz. Because there is a converter on each channel dual channel simultaneous sampling also runs at up to 1 MHz per channel. Both ADC's are strobed simultaneously. Local logic on the module (in the PAL) enables each channel alternately - onto the output lines (AD0..15). Running at full speed (1 MHz per channel) the aggregate system throughput is 2 MHz. The ACQuire signal is decoded to synchronize data transfer into the FIFO.



(a) Solder Side **(b) Component Side**

Figure 3.13 ADM-14G Layout

Note: Dual channel, SS&H mode is selected by writing 2 into the Channel Address Register (BASE + 2). All other registers should be programmed for single channel acquisition.

The input voltage ranges and A/D data coding are configured by solder gaps on the component side of the board as follows:

Solder Gap	Selection
SG1	0 to 10V (ADS-917 special)
SG1 & SG2	±5V Input (standard)
SG3	Unipolar, Straight Binary Coding (2's complement normally - SG3 open

Table 3.7 ADM-14G Solder Gap Selections

Channel Selected	Address Code CA1	CA0
Channel 0 only	0	0
Channel 1 only	0	1
Channels 0, 1 Triggered simultaneously	1	X

Table 3.8 Channel Addressing

Notes:
1. Channel address bits 2 and 3 are not used. Do not use auto-increment mode in the Command Register.
2. In simultaneous mode, channel 0 is loaded into the FIFO first. The data output sequence is: chan 0, chan 1, chan 0, . . . For example, at 1 MHz sample rate, the FIFO load rate is 2 MHz.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ADM-14G Timing

To select dual channel simultaneous sample/hold mode the channel address must be set to 2. The timing is as follows:

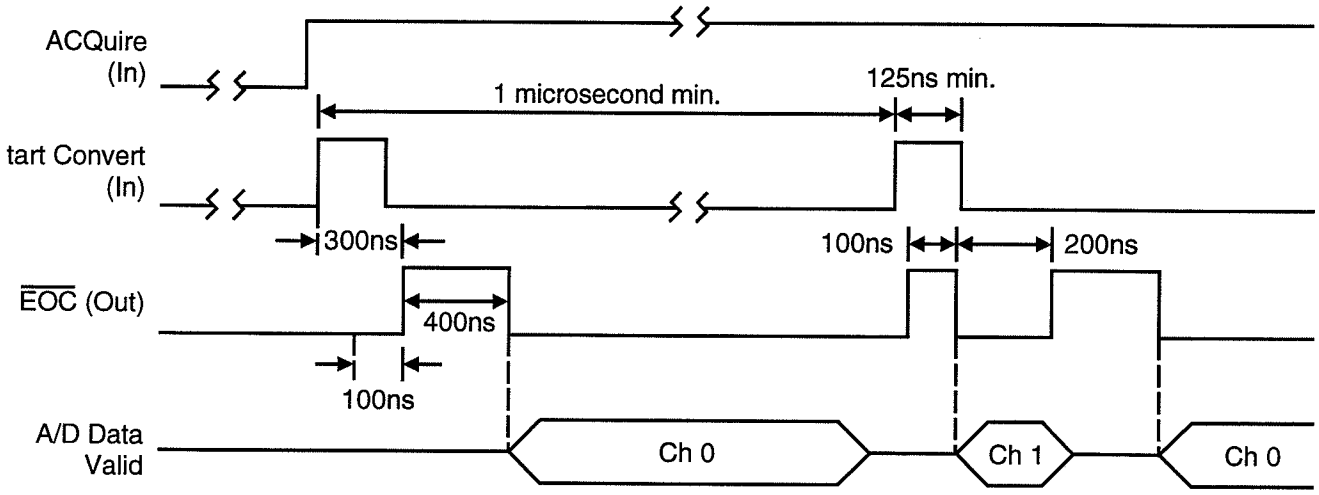
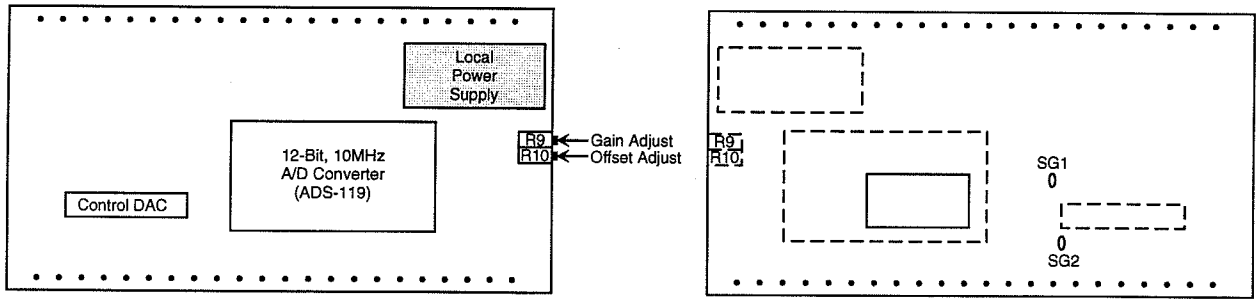


Figure 3.14 ADM-14G SS&H Timing

In single channel mode the first EOC in each pair is inhibited.

3.9 ADM-12H (PCI-416H)

Single channel, 12 bit A/D module. Uses DATEL's ADS-119 sampling A/D converter (internal sample/hold). Channel sampling may continue at up to 10 MHz.



(a) Solder Side **(b) Component Side**

Figure 3.15 ADM-12H Layout

The A/D data coding is configured by solder gaps on the component side of the board as follows:

Solder Gap	Status	Selection
SG1 & SG2	Closed	Straight Binary
SG1 & SG2	Open	2's Complement (standard)

Table 3.9 ADM-12H Solder Gap Selections

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ADM-12H Timing

Figure 3.16 illustrates ADM-12H timing. Because it is a single channel module the channel address is always locked on 0.

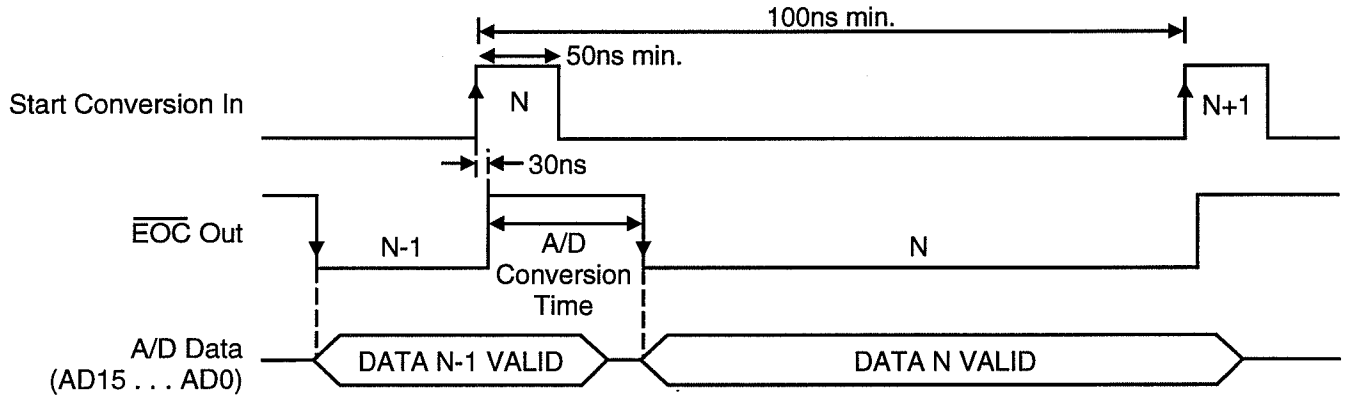


Figure 3.16 ADM-12H Timing

3.10 ADM-8SIM (PCI-416J)

The ADM-8SIM is an 8-channel single-ended simultaneous analog input module. Eight separate surface mount 12-bit A/D converters are used with all converters triggered in parallel. By software, a single channel only may also be sampled. All A/D converters are pretrimmed over the operating temperature range and no adjustments are required. Two versions of this module are offered. The ADM-8SIM-1 has a bipolar input and the ADM-8SIM-2 is unipolar as listed in the following chart:

Model	Full Scale Input Range
ADM-8SIM-1	±5V or ±10V
ADM-8SIM-2	0 to +2.5V

Table 3.10 "J" Series analog modules

November 1995: The ADM-8SIM-2 is presently not available.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The module includes several hardware programmable features. If you wish to modify these parameters, you should be skilled working with miniature electronic assemblies. Please refer to the component layout drawing.

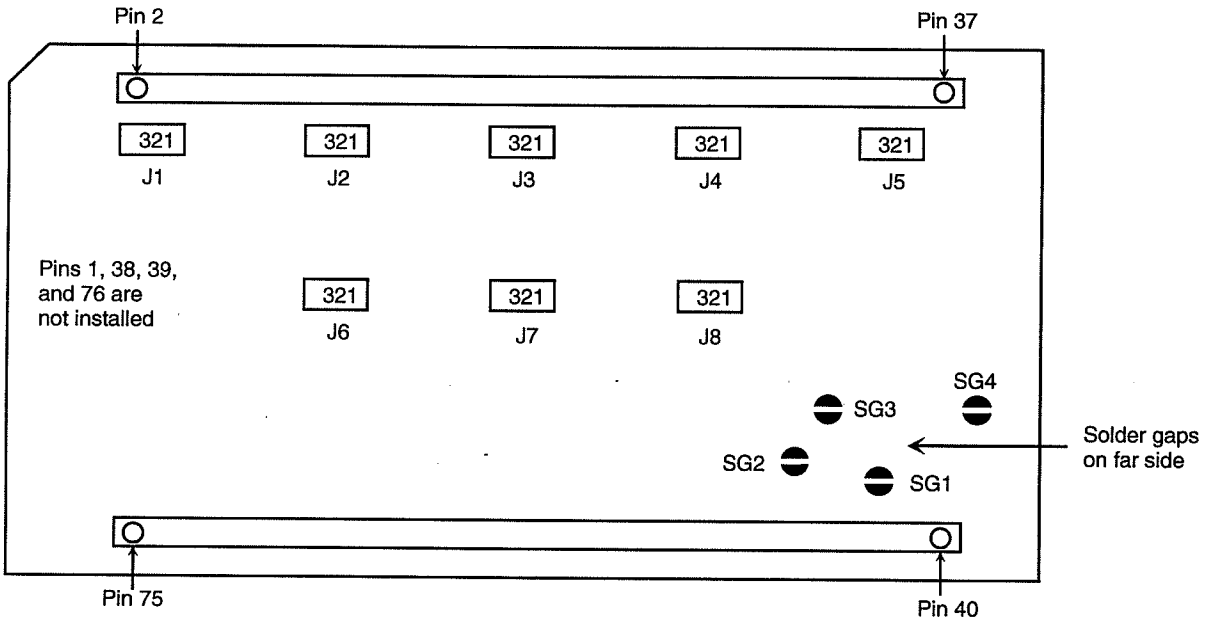


Figure 3.17 ADM-8SIM Component Layout
(Component side shown)

In early 1996, the "J" models were redesigned to accomodate the 16-channel "L" model. The new models use the Assembly number **C-24271.** The PC board is labeled **PC-24274.** The new models offer short cycled channel addressing (see table below). The new layout also adds another solder gap, **SG4** shown in the layout drawing below and additional range selection jumpers for each channel on the ADM-12L. *SG4 does not appear in the existing ADM-8SIM.*

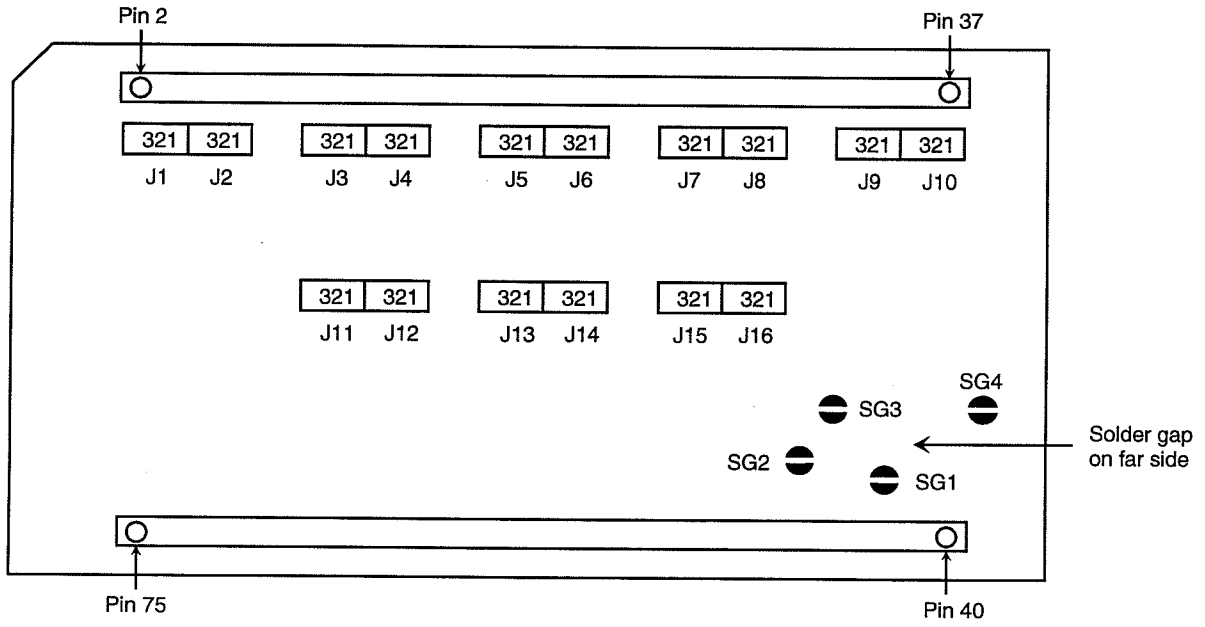


Figure 3.18 ADM-12J/L Component Layout

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.10.1 Input Polarity and Full Scale Range

On the bipolar model ADM-8SIM-1 **only,** the input can be selected between ±5V and ±10V *individually for each channel.* A series of eight jumpers on the ADM module are used to make this selection. Jumper J1 corresponds to A/D channel 0, J2 for channel 1, etc.

Model	Jumpers J1 to J8	Input Range
ADM-8SIM-1	connect pins 1-2, open 2-3	-5 to +5V
ADM-8SIM-1	connect pins 2-3, open 1-2	-10 to +10V
ADM-8SIM-2	connect pins 2-3, open 1-2	0 to +2.5V

Table 3.11 ADM-8SIM Input Range Selection

3.10.2 Sign Extension

The upper data bits from the ADM module (bits 12-15) can be set to logic ZERO (ground) or can be *sign-extended* such that data bits 12 through 15 follow the state of the most significant A/D bit 11. Sign extension is required when converting A/D binary data to an integer data type for arithmetic software operations. These changes require opening and closing *solder gaps.* A solder gap is an area on the printed circuit board where close adjacent areas of copper etch may be bridged with solder, thereby opening or closing an electrical switch. If you make these changes, open one gap and close the other - *do not leave them both open or both closed.* Otherwise you may damage the module.

If you wish to disconnect a solder gap, you will probably need a *solder wick* (section of braided copper strapping) or a *solder sucker* (solder removal tool). Note that if you forget to set these solder gaps in the desired manner, the data can always be modified later by software. The solder gaps have no effect on the least significant 12 bits of A/D data and you will not lose any data.

CAUTION: Excessive heat will damage the PC board and may void the warranty. Do not change the solder gaps unless you have very good soldering skills.

Data Bits 12-15	Solder Gap 2	Solder Gap 3
Sign-extended (echoes A/D MSB, bit 11)	Closed	Open
Logic Zero	Open	Closed

Table 3.12 A/D Most Significant Bit Sign Extension

Input Range	15 <bits> 13	12	11 <bits> 0
Bipolar	S or "0"	S	MSB data LSB
Unipolar	S or "0"	MSB	data LSB

Table 3.13 A/D Data Format

"S" means sign extension. "MSB" is the A/D most significant bit. LSB is the least significant bit. Unipolar converters are 12 data bits resolution. Bipolar models are 11 data bits and one polarity bit.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.10.3 Production Programming Pins and Miscellaneous Functions

Pins 26 through 30 and pin 60 on the ADM-8SIM is used in production to initialize the local programmable logic. *Do not connect these pins.* Normally, these pins are not connected on all other DATEL A/D carrier boards. Pin 61 (the ACQuire input) must be HIGH (logic "ONE") to enable the module. Also, pin 66 (Simultaneous Sample/Hold input) is not used on the ADM-8SIM. The A/D End of Conversion output (EOC*, pin 64) produces eight output pulses in multichannel mode. Refer to the timing diagram.

3.10.4 Channel Addressing

The ADM-8SIM will collect all 8 channels simultaneously or only one channel at a time. Bit 3 in the address code controls single channel or simultaneous mode. If bit 3 = 0, the lower bits select a single channel address. If bit 3 = 1, the module operates in simultaneous mode and the lower 3 bits are "don't care". In single channel mode, the channel address must be written once before starting A/D conversion.

In multichannel simultaneous sampling, all 8 channels are acquired at exactly the same time from a single A/D start pulse. Each A/D output then strobes its data through to the FIFO memory on the carrier board. This digital multiplexing is in sequential channel order, address 0 to 7, producing eight End of Conversion pulses from the module. There is no provision to short cycle the system to less than 8 channels. If you need fewer channels, collect all the data then decimate it to discard unwanted channels.

If you run in single channel mode, there is no requirement to wait for an input settling delay. You may take data immediately. Unlike analog multiplexer designs, each A/D converter remains connected full time to its input channel and does not suffer input switching delays. The host carrier board's Channel Address register controls this addressing as follows:

Address Mode	Addressing
Simultaneous sampling on 8 channels	Use address code 8
Single channel only	Use address code 0 to 7

Table 3.14 Channel Addressing

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Feb. 1, 1996 - ADM-8SIM modules are being modified to allow channel address short cycling. This is controlled by Solder Gap 4. The older, existing address method is retained if SG4 is closed (factory default). For newer ADM-8SIM's (with SG4):

SG4	Address	Channels Converted
Closed (default)	0 to 7	Single channel, 0 to 7
	8	0 to 7, simul. sampling
Open	8	Channel 0 only
	9	0 - 1 simul.
	10	0 - 2 simul.
	11	0 - 3 simul.
	12	0 - 4 simul.
	13	0 - 5 simul.
	14	0 - 6 simul.
	15	0 - 7 simul.

Channel 0 is always output first.

Table 3.15 Short Cycled Addressing

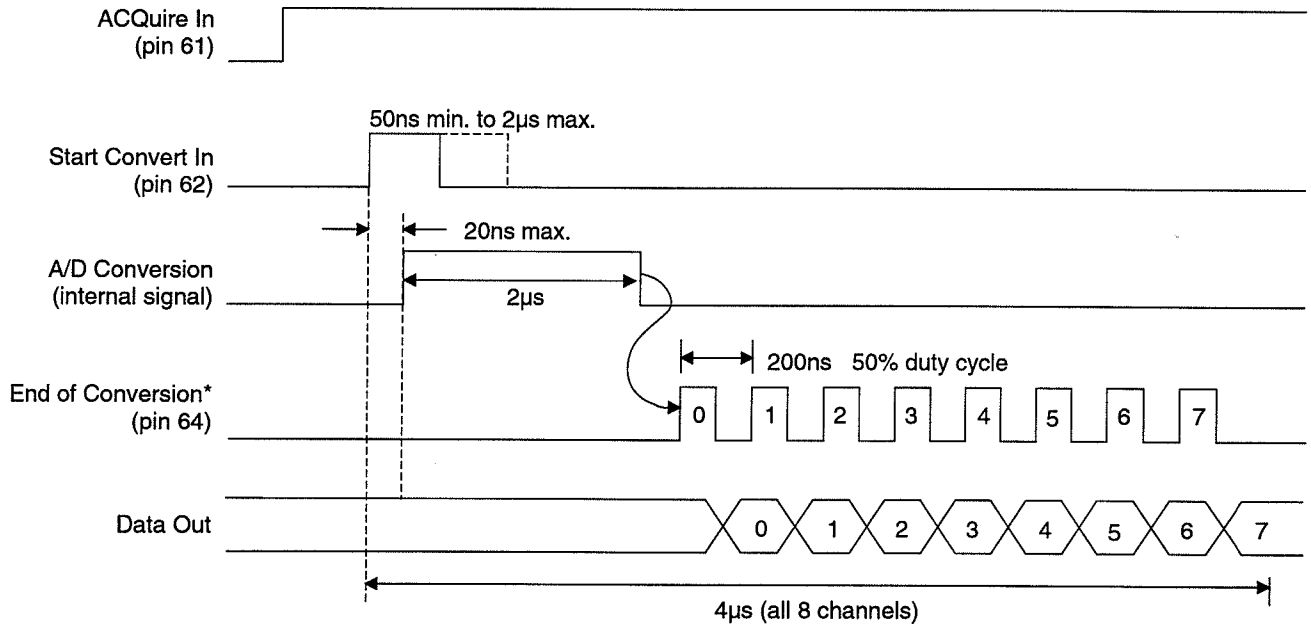
For all addressing modes, leave the command register autoincrement bit OFF.

3.10.5 Timing Diagram

The output timing of the ADM-8SIM module in eight channel simultaneous A/D conversion is designed to be fed to a FIFO memory or other sequentially loading device. The net data rate is 2 megasamples per second (250kHz times 8). The timing is conventional with the exception that the falling edge of EOC* digitally multiplexes the next A/D data output onto the data bus. Therefore, if you are designing your own system built around the ADM-8SIM, allow a small delay of about 50 nanoseconds before reading the data after EOC*. Otherwise if you read too quickly after EOC*, you risk loading old data. Typically, propagation delays in user logic will completely absorb this delay (but test to be sure). On DATEL's A/D carrier boards, an EOC shaping one shot or equivalent logic after the EOC* applies the correct delay before loading the FIFO.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

In single channel mode (address bit 3 = 1), only one EOC* pulse is presented and the sample-to-sample timing is 2.5 microseconds (400kHz sample rate). It is possible by software to very quickly change the digital channel address and collect randomly-addressed single channel samples. Be sure the address is switched just after collecting the previous sample and before the next conversion. Since all eight A/D's remain continuously connected to their analog inputs and no analog settling is required, multichannel data appears at the 400kHz rate.



Timing is not to scale. Allow 50ns (typ.) propogation delay after the falling edge of EOC* before reading data.

The ADM-12L issues sixteen 200ns clocks with identical timing.

Figure 3.19 ADM-8SIM Timing Diagram (8 channel mode)

3.10.6 ADM-8SIM Calibration

The ADM-8SIM is precalibrated with guaranteed performance over it's operating temperature range. This characteristic is excellent for signal inputs where absolute calibration is less important than relative amplitudes and very good linearity. If you wish to remove the small residual offset and gain errors, either use an external calibrating preamplifier or use an external analog multiplexer which periodically samples ground or zero reference and nearly full scale known reference inputs. Use the data collected from these calibration channels and correct all signal data in software.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.11 Input Amplifier Programmable Gain Selection

Input amplifier gains are selectable on both the PCI-416A and E modules to accommodate low level, analog input signals.

PCI-416A: The gain on input channels J3 and J4 is selectable as times 1 or times 10 by soldering selected gaps on the analog module (see Figure 3.1a). This task should be done by experienced personnel and extreme caution should be exercised to avoid damaging the module. Note that any damage inflicted in this respect may void the warranty.

The following are the selections:

Channel 2 (J3) gain = x10 Solder SG1
Channel 3 (J4) gain = x 10 Solder SG2
(Note: The default setting is for x1, with no gaps soldered.)

PCI-416E: The input instrumentation amplifier gain is resistor programmable for gains from x1 to x100. The gain resistor designations are RG1, RG2, and RG3 - refer to Figure 3.9. These are configured with RG1 in parallel with RG2 and RG3 in series with the combination. Total gain resistance may be calculated from the following equations:

$$R_{GAIN}\text{ (total) in Ohms} = 4000/(\text{Gain} - 1)$$
$$\text{where } R_{GAIN}\text{ (total)} = [1/(1/RG1 + 1/RG2)] + RG3$$

The recommended procedure is to short out RG3 by connecting a wire across its pads. This forces RG3 = 0 in the above equation. RG1 should be a low drift, high precision resistor (0.1% tolerance) and RG2 is used to fine tune the gain. Observe longer settling delays after switching channels for higher gains.

3.12 Repair and Troubleshooting

This manual is not intended for repairs. There are no user-serviceable parts on the PCI-416 other than those discussed in previous sections. This board was manufactured using special automated procedures and precision custom test fixtures. Repair attempts by users without special knowledge may damage the board and void the warranty.

DATEL endeavors to keep a small inventory of boards at its home office. However, this inventory cannot be guaranteed. Users with critical downtime requirements should keep a spare board and retest it periodically. Store it in its anti-static cover.

All service difficulties should be referred to the Mansfield, MA, USA home office. The PCI-416 is fully guaranteed within the warranty period. Normal labor and materials charges will apply beyond the warranty period. Returns to the factory for any reason will require a Return Material Authorization (RMA), obtained by contacting the factory before shipment. Products shipped without an RMA will not be accepted. All returns must have transportation charges prepaid.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SECTION 4

PCI-416 REGISTER DESCRIPTIONS

4.0 Introduction

The PCI bus controller for DATEL's PCI-416 is an Applied Micro Circuits Corp. S5933. Please read the S5933 Controller manual (included with this User Manual) very carefully, especially regarding PCI Configuration Space and the Configuration Registers (Section 3). The information below will only make sense if you read the controller book first. You should also be thoroughly familiar with the PCI BIOS specifications and a general overview of PCI bus operations listed in the previous references.

The PCI system contains some unusual concepts for users familiar with simpler register-controlled systems. The actual address map (whether in memory or I/O space) is built during the PCI initialization sequence. The PCI-416 states to the BIOS, via the S5933 registers, how large the address reservation must be and what type (memory or I/O). The PCI BIOS then decides where the absolute physical addresses are as it allocates each PCI board. User software must determine these actual addresses before any access to the board.

The S5933 responds to PCI bus cycles according to the Base Address Registers (BADR0-5) on the S5933 controller. The PCI-416 uses the first five (BADR0 through BADR4) of the six possible registers in the configuration space map. The first register (BADR0) is devoted exclusively to S5933 controller operations. The other four registers are assigned to data acquisition functions using the S5933 Pass-Thru mode according to the hardware register map below.

The size and type (memory or I/O) but not the address of regions 1 through 4 are determined from initialization by the external serial EPROM memory on the PCI-416 board. However, the actual base address of each region is set by the host computer PCI BIOS calls at boot-up time. In effect, this provides a way for the host computer to reposition PCI devices (such as the PCI-416) in memory or I/O regions compatible with the rest of the computer. Since it is possible to reprogram the EPROM from the PCI bus, the system could initialize to some other defaults. Normally, the user should never do this or have to worry about it. Reprogramming the EPROM incorrectly from the PCI bus could lock up the board and require service from DATEL.

The second Base Address Register (BADR1) serves a special function. This register acts as a selector to a specific Pass Thru register. The desired register address within the region is accessed by writing the address to the Pass-Thru Address Latch (ARL) which is mapped to BADR1.

4.1 DATEL Defaults

The EPROM on the PCI-416 initializes the S5933 with the following defaults:

Device ID code: 0416h
Revision ID Register: 0 (this code will cycle with new revisions)
Vendor ID code: 11FBh (DATEL's unique ID)
Access mode: I/O

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.2 Hardware Register Map

All addresses given below are *byte addresses* even though all access is 32-bit wide DWORDS. Note that word (16-bit) or byte (8-bit) registers are *right justified* and occupy the lease signficant bits in the doubleword. Unused higher order bits are "don't care". Always use 32-bit DWORD access even though some registers are only 8 or 16 bits. Although the default is I/O mapping, the user should not assume this and should always test the board with software when first setting up the board. In particular, the S5933 can also be mapped as either base or extended memory and there are bits to limit extended memory to below 1 Mb or allow it to exceed 1 Mb.

BADR0: S5933 PCI controller operation registers

BADR1: Pass-Thru Address Register Latch (ARL)

BADR2: Read/Write general register

If ARL=0,	Command Register (write only)
If ARL=4,	Sample Counter (write only)
If ARL=8,	Channel address register (write)
	Clear A/D FIFO memory (read)
If ARL=12,	A/D convert enable (write only)
If ARL=16,	PLL register (write only)

BADR3: High speed read of status register or A/D FIFO data

If ARL=0,	Status register (read only)
If ARL=4,	A/D FIFO data (read only)

BADR4: Low speed devices, 82C54 and 82C55, Analog Output

82C54 Programmable counter-timer (read/write)

If ARL=0,	Counter 0
If ARL=4,	Counter 1
If ARL=8,	Counter 2
If ARL=12,	Control register

82C55 Programmable parallel port (read/write)

If ARL=16,	Port A
If ARL=20,	Port B
If ARL=24,	Port C
If ARL=28,	Control register

Analog Output (D/A Channel)

If ARL=32,	D/A converter register (write only)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 Programming Guidelines

Controlling the S5933 requires strict adherence to the sequence of operations, data formats and protocols used. Programmers should thoroughly understand concepts of indirection, that is obtaining addresses from registers after first making selections in other registers. While setting up the PCI-416 is more involved than simpler systems, full PCI bus speed is available after initialization.

Study DATEL's commented source code thoroughly to understand the programming sequence. For write-only registers, it is often useful to "remember" what was written to that register so the system can determine what state it is in. This is done with a "shadow" register which is a memory location that stores the last data written to a physical register. The write-only registers will not accept bit operations but often only one bit must be changed. To manipulate a bit in a shadow register, test the bit, change it if necessary then write the entire shadow value out to the corresponding real register.

Debugger advice: If you try developing PCI-416 code from a debugger, be aware of how registers work and how some debuggers operate. Most PCI-416 registers are either read only or write only. Although they may be mapped in memory or I/O space, they do not act like normal memory locations. Some registers contain both read and write operations *at the same address*. But in fact, they are separate registers. Also, many registers have empty fields where there are no physical bits.

Although the PCI-416 defaults as I/O access, if these registers are mapped as memory, some debuggers do a blind readback for any values written to memory (including a memory-mapped PCI-416 register). In an effort to be helpful, the debugger compares the value read back from what was originally written. If the values do not agree, the debugger will report an error. Since the register is write-only, the read back value will be meaningless or may be from another read-only register with a different value. However the write operation usually succeeded and the error may be ignored.

Most debuggers default to byte operations only. You must explicitly tell the debugger to do 32-bit DWORD operations. If this is not available on your debugger, most will accept in-line assembly instructions which *will* do the DWORD access.

BADR1 is used to write the address of the desired on-board register. Pass-thru Base Address Registers 2-4 are then used to access a specific register.

Example: Write "1024" to the sample counter

1. Use the PCI_BIOS_PRESENT function to be sure the computer can access PCI.

2. Use the PCI-416 Vendor and Device ID codes to call the FIND_PCI_DEVICE and FIND_PCI_CLASS_CODE BIOS functions to determine the device and bus number for the PCI-416.

3. Call the READ_CONFIG_DWORD BIOS function passing the DEVICE and BUS NUMBER above to read the BADR's located in the PCI-416's configuration space. The BADR's contain the address of each region. The *least significant bit* of each BADR specifies whether the address is memory-mapped or I/O space where LSB=0 for memory and LSB = 1 for I/O.

Assume after the BIOS assignments, the following mapping resulted:

Eg: BADR1 = 0xFFC1 corresponds to I/O address 0xFFC0
BADR2 = 0xFFD1 corresponds to I/O address 0xFFD0
BADR3 = 0xE0000 corresponds to memory address 0xE0000

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4. To write "1024" to the sample counter, write "4" to the ARL located at BADR1 which is now located at I/O address 0xFFC0.

The "C" instruction for this is: **outpd (0xFFC0, 4L);**

(***outpd*** is a small assembly language function. Refer to DATEL's software for the source code to *outpd*).

Note the "L" qualifier to make the argument 32-bits wide.

Then write the count value "1024" to BADR2 which is now located at I/O address 0xFFD0.

The "C" instruction for this is: **outpd (0xFFD0, 1024L);**

Full details of PCI-416 programming are in DATEL's source code.

4.4 Non-volatile Configuration Memory -

The S5933 has numerous internal registers which must be initialized at power up. These include the Vendor ID, Base Address Register initialization, etc. Some of these registers *cannot* be initialized by the PCI bus therefore a local non-volatile memory automatically loads these registers when power is first applied. In addition, other critical bits (such as the Bus Master Enable in the PCI Command Register) must be initialized properly by the host.

4.5 Register Bit Assignments

IMPORTANT: *Access all registers using 32-bit instructions* ***only****, even if the register is 8 or 16 bits wide. Smaller registers (8 or 16 bits) are right justified starting at bit zero. The upper bits are not used.*

Command Register (write only)
BADR2, ARL = 0

16	15,14	13-11	10-8	7	6	5	4	3	2	1,0
Bus Master Enbl	ADM Ctrl 1 0	MUX Sel CBA	PLL Reg 2 1 0	Intr Enbl	Pre Trig	Mrkr Sel	Auto Incr	Scan Sel	Intr Sel	Trig Sel 1 0

Bits 31-17 are not used. Note that the Bus Master Enable (bit 16) maps into the upper word.

Trigger Mode Select - Bits 1, 0

00 = Internal trigger
01 = External digital trigger
10 = External analog trigger
11 = Not used

Mode 01 starts A/D sampling on the falling edge of the external digital trigger.

Mode 10 starts A/D sampling when the external analog trigger input falls *below* the voltage level set by the D/A converter. A small acquisition delay of a few microseconds occurs from threshold crossing. This trigger function is latched so that the external analog trigger does not have to be held below the D/A level (useful for spike triggers).

Note that *the trigger is not the A/D sample clock.* Read the trigger sections of the manual.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A/D Interrupt Source Select - Bit 2

0 = FIFO Half Full interrupt 1 = End of Scan ACQuire flag interrupt.

End of Scan (EOS) occurs when the A/D sample counter has reached zero and A/D sampling has temporarily stopped while waiting for the next trigger. Note that if the EOS interrupt mode is selected (and the host computer does *not* repeatedly poll the FIFO half full flag), there is no automatic protection against FIFO overflow, even though the application is "interrupt driven". The user must poll the FIFO half full flag and transfer half blocks until EOS occurs then collect the last half block remaining in the FIFO.

The FIFO half full interrupt is more reliable to assure that half full is not missed. However, mapping the HF to an interrupt limits throughput in the very fastest applications because of extra interrupt processing overhead.

Scan Select - Bit 3

0 = Scan mode disabled 1 = Scan mode enabled

The Scan mode is used *only* for the PCI-416A ADM-12A module with simultaneous sample/hold. This bit enables a scan counter so that multiple A/D starts occur with one A/D clock pulse. This bit controls the "SSH" line on the ADM module (pin 66).

Channel Address Auto Increment - Bit 4

0 = No channel advance (single channel mode)
1 = The channel address automatically advances after each sample.

The A/D Channel Address Register is located at BADR2, ARL=8. This bits controls whether the channel address counter is allowed to increment with each sample. Note that the channel address also resets to zero at EOS. Since the channel address register is write-only, user software must keep track of the channel address by software, either by counting blocks or by maintaining a software "shadow register".

For newer A/D-per-channel ADM modules, reset bit 4 to zero and refer to channel address coding in the ADM section.

Marker Bit Select - Bit 5

0 = Marker bit disabled 1 = Marker bit enabled

See the description in the PCI-416 data sheet. If enabled, the marker sets A/D data bit 15 or bit 31 to logic ONE for one A/D clock whenever an external Marker input is received. This may be used to tag data.

Pretrigger Mode - Bit 6

0 = Pretrigger disabled 1 = Pretrigger enabled

If enabled, the A/D sample counter will delay count down to zero until an external trigger is received. When the trigger occurs, A/D sampling continues without delay however the sample counter now starts its countdown. This allows collection of a fixed number of postrigger samples. If the A/D clock is properly set up, the A/D will start and samples will fill the FIFO when Pretrigger is set to logic ONE.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A/D Interrupt Enable - Bit 7

0 = Interrupts disabled 1 = A/D interrupts enabled

Refer to Bit 2 for the interrupt source.

Phase Lock Loop Register Select - Bits 10, 9, 8

These bits define which counter register in the frequency synthesizer is being accessed.

Bit 10	Bit 09	Bit 08	Register
0	0	0	Counter A, bits 3-0
0	0	1	Counter A, bits 7-4
0	1	0	Counter N, bits 3-0
0	1	1	Counter N, bits 7-4
1	0	0	Counter N, bits 11-8
1	0	1	Reference bits, 3-0
1	1	0	Reference bits, 7-4
1	1	1	Reference bits, 11-8

Counter N is normally loaded with a number from 200 to 400. Counter A is loaded with a number from 0 to 127. The reference must be set to 4000.

Clock Source Multiplexer Select - Bits 13, 12, 11

These bits select the A/D clock source and rate which is then further divided by either the frequency synthesizer or the 82C54 programmable counter or the discrete divider.

Bit 13	Bit 12	Bit 11	Function
0	0	0	20-40 MHz in 2.5 KHz increments. The rate is programmed by the frequency synthesizer register.
0	0	1	10-20 MHz in 1.25 KHz increments. The rate is programmed by the frequency synthesizer register with the synthesizer output divided by two.
0	1	0	5-10 MHz in 625 Hz increments. The rate is programmed by the frequency synthesizer register with the synthesizer output divided by four.
0	1	1	From 82C54 timer with synthesizer as clock source. 10 MHz max. input to 82C54. Minimum output divisor is 2. Program the rate via the frequency synthesizer.
1	0	0	From external clock input, TTL logic levels. Max input is 40 MHz using frequency synthesizer or 10 MHz using the 82C54.
1	0	1	Fixed 20 MHz crystal clock direct to the A/D.
1	1	0	Fixed 10 MHz crystal clock direct to the A/D.
1	1	1	From 82C54 timer with 10 MHz crystal clock source. The minimum output divisor is 2.

A/D Module Control - Bits 15, 14

Set these bits to "00" for normal use. *Do not use any other codes.* Other states are used for internal testing. Bit 14 controls pin 60 on the ADM module. Bit 15 controls pin 59.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PCI Bus Master Control - Bit 16

0 = Disable Bus Mastering 1= Enable Bus Mastering

This Bus Mastering bit allows the PCI-416 to acquire control of the PCI bus and initiate high speed block transfers. High speed mode *also* requires programming the PCI-416 S5933 controller to operate in high speed mode. The PCI bus *slot* used for the PCI-416 *must* accept bus mastering to use this mode.

See Bus Master Operation, Section 6.11

Sample Counter Data Register (write only)
BADR2, ARL = 4

31 -24	23 - 0
not used	MSB 24-bit count value LSB

The 24-bit sample count is loaded into this register. This determines the number of samples collected with each trigger. This function counts the number of A/D End of Conversion (EOC) pulses, not the number of scans or frames.

Note that the sample counter can be disabled so that sampling continues "forever". This is done by starting the A/D in the pretrigger mode, enable the A/D then do *not* send a trigger. If you still want to know how many samples were taken, in your software, count the number of FIFO blocks transferred. In order to retain high speed, do not count individual samples.

IMPORTANT: *Load one less than the desired number of samples into this register. To collect 256 samples, load 255. The minimum count is two.*

A/D Channel Address Register (write only)
BADR2, ARL = 8

31 - 9	8	7 - 4	3 - 0
not used	LED	Scan count	Start Channel Address

Bits 31 - 9 are not used.

Bit 8 controls the on-board LED lamp. "0" = dark, "1"= lit. This lamp is intended for diagnostics.

The Scan count field (bits 7-4) is used only by the PCI-416A board. The Scan count field sets the number of samples which will be collected with one A/D clock. The minimum count is three.

The Channel Address field (bits 3-0) sets the starting address for the input channel address multiplexer (for those boards with a MUX - models B, C, and E). In autoincrement mode, this address will automatically sequence in modulo counter style according to the channel capacity of the analog module. The channel address will wrap around to zero when the last channel is reached. For example, the channel sequence for 4-channel boards is 0, 1, 2, 3, 0, 1, ... For A/D-per-channel boards without a MUX, refer to Section 3.

Note that this register is write only and you cannot read it back. To determine what address is used, count blocks in software or periodically reset the address. If you use the external A/D sample clock, be aware that a random glitch would start a conversion and advance the channel. The channel address is automatically reset to zero at the EOS which a frame completes.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FIFO Memory Reset Register (read only)
BADR2, ARL = 8

31 - 0
don't care

The returned value should be ignored.

Any PCI bus reset or any read of this register resets the FIFO to empty. All A/D data which was not previously read from the FIFO will be lost. The value returned from this register is meaningless. If the A/D converter was still running at the time of the reset, the FIFO will immediately start filling again. Reading this register also resets the latched FIFO full flag.

A/D Convert Enable Register (write only)
BADR2, ARL = 12

15	14 - 0
WCNV	don't care

Bits 31-16 and 14-0 are not used.

0 = A/D sampling is disabled.
1 = A/D sampling is enabled and conversion will start when the trigger occurs.

Bit 15 enables A/D conversions as soon as a trigger occurs. After bit 15 is set, the first occurance of the trigger will set the ACQ flipflop which gates A/D clocks through to the converter, enables A/D FIFO writes and starts the sample countdown. Sampling then stops when the sample count reaches zero unless the sample counter has been disabled. The next trigger will repeat this process and will continue until the A/D is disabled by writing zero into bit 15.

Note that several conditions must happen for valid A/D data to successfully enter the FIFO. The FIFO must be have room to accept data, the A/D clock must be running, this WCNV bit must be set to "ONE" then a trigger must occur.

In pretrigger mode, A/D sampling starts immediately with bit 15 and does not wait for a trigger.

Phase Lock Loop Data Register (write only)
BADR2, ARL = 16

31 - 4	3 - 0
not used	MSB PLL data LSB

The Frequency Synthesizer consists of three internal programmable registers - one 8-bit and two 12-bit. Data access to each of these registers uses a 4-bit bus. Form the whole 8-bit or 12-bit data value then load it in multiple 4-bit nybbles while rotating right by 4-bits each time. Each 4-bit nybble is selected by bits 10-8 in the Command Register.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Status Register (read only)
BADR3, ARL = 0

15-13	12	11 - 8	7	6	5	4	3	2	1,0
FIFO Flags F H E	Arm F/F	Mod ID 3 2 1 0	Ana. Trig Stat	Pre-Trig	Mrkr Sel	ACQ F/F	Auto Incr	Scan Sel	Trig Sel 1 0

Bits 31-16 are not used.

Echo Flags - Bits 6, 5, 3-0

These bits echo the state written to corresponding bits of the same name in the Command Register and may be used to confirm proper register loading. *Note that the bit positions are relocated.*

ACQuire Flag - Bit 4

0 = A/D scan is not in progress or the scan is done. The sample count was reached.
1 = A/D scan is in progress. The sample count has not been reached.

The ACQuisition status bit indicates that A/D clocks are being gated through to the A/D start input and that sample counting is in progress. Note that for very fast host code the ACQ flag has a slight delay when it first goes to "1". Therefore the initial "0" state would give a false indication that the scan is complete. If you write your own code, use a short delay or poll on the FIFO empty bit to determine when sampling has started. This is also good advice if you use a delayed external trigger.

In the case of the Pretrigger mode, ACQ is delayed until an external trigger. However, sampling will start immediately when the pretrigger bit (CMD6) is set and the A/D is enabled.

The ACQuire bit can set a PCI bus interrupt if enabled by Command bit 2.

Analog Trigger Status - Bit 7

This bit indicates whether an external signal is above or below the DAC comparator reference level. This bit is the raw comparator output. It does not indicate A/D sampling activity. Use the ACQ bit to see if the A/D is running.

0 = The analog trigger input is higher than the threshold level set by the DAC.
1 = The analog trigger input is lower than the threshold level set by the DAC.

The analog trigger is useful if you wish to inhibit A/D sampling from a "noisy" input which does not yet have valid signals (e.g., receivers). The purpose is to avoid filling memory, disk or tape with erroneous signals.

This bit may also be used for diagnostics on the analog trigger function. Also be aware that there is no internal connection between the analog trigger input and the A/D signal channels. Note that analog triggering occurs on high to low voltage transitions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Analog Module Type - Bits 11-8

These bits echo the state of the DIP switch U5, switches 1-4. A host program such as PCI-416SET, reads these bits to determine the type of analog module installed. Note that there is no coding on the module to set these bits. If a module type is changed, the DIP switches must be manually changed. These bits also set local hardware timing for the modules. Failure to change the bits will not disable the system however it may interfere with the PCI-416SET program and possibly give wrong data.

The U5 DIP switch module coding is as follows:

	MSB			LSB
	S1	S2	S3	S4
	—	—	—	—
Status Bit	11	10	09	08

DIP switch orientation - With the PCI connector to the lower left, the DIP switch is located as shown below:

8	51 Ohm trigger input termination
7	Not used
6	Not used
5	Not used
4	Module code 0, status 8 (LSB)
3	Module code 1, status 9
2	Module code 2, status 10
1	Module code 3, status 11 (MSB)

ON="0" OFF="1"

Some switches may have different orientation for "ON" or "OFF".

Switch 8 grounds a 51 ohm input resistor on the trigger input. "ON" means the 51 ohm resistor is installed. This is to reduce line reflections and false triggering for trigger inputs which can drive 51 ohms.

The A/D modules are in sequential order as follows:

A = 0000	H = 0111
B = 0001	J = 1000
C = 0010	K = 1001
D = 0011	L = 1010
E = 0100	M = 1011
F = 0101	N = 1100
G = 0110	

Note that there is no "I" model to avoid model number confusion.

For example, a PCI-416D, using an ADM-12D analog module uses the following:

SW1 = ON, SW2 = ON, SW3 = OFF, SW4 = OFF

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ARM Flag - Bit 12

0 = ARM not enabled. A/D sampling is inhibited
1 = A/D started sampling but no sample countdown yet

The ARM flag is set when the Pretrigger bit (Command 6) is set to "1". As long as the A/D sample clock and clock MUX systems have started ticking and the A/D converter is enabled, A/D samples will start filling the FIFO. When an external trigger occurs, the ACQ flag will transition from "0" to "1", and the sample counter will start to count down. When the counter reaches zero, ACQ will reset to "0" and the A/D will stop sampling. The entire system is designed for pretrigger sampling where data is collected into a ring buffer.

A convenient algorithm to read the FIFO in ring buffer mode is to start the pretrigger system and begin polling the FIFO half full flag (or use a half full interrupt). Block transfer one half the FIFO at highest speed at each half full signal. Each time the half full is polled, test the ACQ bit=1 to see if an external trigger has occurred and the sample counter has started to count down. Continue collecting half blocks.

When the ACQ returns to zero, the sample count is complete and the A/D has stopped sampling. If the FIFO is still half full, transfer a half block. If the FIFO does not reach half full after the ACQ returns to zero, test the FIFO empty flag and read single samples as long as it is not empty. This avoids overreading empty FIFO data and collects all the data.

FIFO A/D Memory Status Flags - Bits 15, 14, 13

Bit 15: 0 = FIFO is full <latched bit>, 1 = FIFO is not full
Bit 14: 0 = FIFO is half full or greater, 1 = FIFO is less than half full.
Bit 13: 0 = FIFO is empty, 1 = FIFO is not empty

These bits indicate the status of the A/D FIFO memory. Observe the *reverse coding* of these bits. Also, half full (HF) is true when the FIFO contains one half the capacity *PLUS ONE*. For example, when HF is first set to zero, the 4K FIFO contains 2048 samples (not 2047).

The half full bit can generate a PCI bus interrupt if enabled.

Note: The FIFO full flag (FF) is latched to catch transient overflow errors. If the full flag is set, any additional A/D data attempting to write to the FIFO will be lost. Therefore data is not necessarily lost if full but there is no guarantee that it was not lost.

The full flag will remain set even if samples are read and the FIFO is no longer full. The latch is cleared using the FIFO Reset Register.

A/D FIFO Data Register (read only)
BADR3, ARL = 4

31 - 16	15 - 0
Sample or Channel N+1	Sample or Channel N

This register contains two samples of integer A/D data. Data is in two formats depending on the number of channels. For single channel applications, earlier data is in the lower word. For multichannel applications, channel numbers alternate with even channels in the lower word, odd channels in the upper word, assuming the channel address register started at channel zero.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Each A/D sample is right justified within its 16-bit field as follows:

12-bit A/D data:

31 - 28	27	26 - 16 (upper A/D word)
15 - 12	11	10 - 0 (lower A/D word)

Sign extension	MSB/polarity	A/D LSB's

In the lower A/D word, for unipolar input A/D's, bits 15 - 12 are set to ZERO and bit 11 is the A/D Most Significant Bit. For bipolar input A/D's, bit 11 indicates polarity and bits 15 -12 echo bit 11 so that arithmetic operations can quickly test the polarity in bit 15. The same format is used for the upper A/D word. Therefore, the A/D is either a 12-bit unipolar converter or 11-bits and sign for bipolar.

14 -bit A/D's use the same strategy:

31 - 30	29	28 - 16 (upper A/D word)
15 - 14	13	12 - 0 (lower A/D word)

Sign extension	MSB/polarity	A/D LSB's

16-bit A/D's completely fill each word.

Note that there is no single-sample 32-bit wide mode.

82C54 Programmable Timer (Read/write)

IMPORTANT: *The 82C54 registers require delays between each access. See the 82C54 programming section.*

BADR4, ARL = 0

31 - 8	7 - 0	
not used	Timer 0 data	(Cascaded to Timer 1)

BADR4, ARL = 4

31 - 8	7 - 0	
not used	Timer 1 data	(Internal trigger)

BADR4, ARL = 8

31 - 8	7 - 0	
not used	Timer 2 data	(Internal A/D Start Clock)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BADR4, ARL = 12

31 - 8	7 - 0
not used	Control Register

Timer Usage

	Clock Input	Output
Timer0	10 MHz Osc.	To Timer1 clock input
Timer1	From Timer0 out	Internal Trigger
Timer2	From digital MUX	To digital MUX

Under control of the Command Register through the digital MUX, Timer2 divides either the local 10 MHz crystal oscillator or the frequency synthesizer output to generate A/D clocks. The digital MUX also contains a high speed divider. All three 82C54 gates are continuously enabled.

82C55 Parallel I/O Controller

IMPORTANT: *The 82C55 registers require delays between each access. See the 82C55 programming section.*

BADR4, ARL = 16

31 - 8	7 - 0
not used	Port A data register

BADR4, ARL = 20

31 - 8	7 - 0
not used	Port B data register

BADR4, ARL = 24

31 - 8	7 - 0
not used	Port C data register

BADR4, ARL = 28

31 - 8	7 - 0
not used	Port Control Register

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D/A Converter Data Register (write only)
BADR4, ARL = 32

31 - 12	11 - 0
not used	MSB D/A analog data LSB

Integer data for the 12-bit D/A converter channel is written into this register. The analog output will be valid immediately after the write within the settling time of the DAC. The data format is either straight or offset binary.

Note: Two's complement coding is not used.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SECTION 5

USING THE PCI-416

5.0 Introduction

There are three general methods to operate the PCI-416. They are:

1. Use DATEL's PCI-416SET menu-driven Setup and Configuration program. This *MS-DOS* based software fully exercises the board and saves A/D data to either memory and/or disk. The full speed of PCI bus is available using fast PCI bus master transfers to huge extended memory. DATEL's PCI-416NT software for Windows NT operates very similarly to PCI-416SET but at a lower speed.

2. Use PCI-416SET in the Start and Exit mode with data feeding a processing program you have written.

3. Write your own program from scratch, probably using sections of code from PCI-416SRC (the source code to PCI-416SET).

Now let's review the tradeoffs for each option:

Method 1 - PCI-416SET Data Acquisition Software

The first method, PCI-416SET, does not require you to write any software. Refer to the PCI-416SET information in the software section of this manual. A/D data is collected at high speed and you have full access to all on-board hardware. Output is always limited to *disk files*. PCI-416SET does many things but it cannot possibly do all conceivable applications. You cannot run any other concurrent program therefore data cannot be passed by memory buffer to that program.

PCI-416SET also does not have any built-in graphics display of data. The philosophy is that many users already have favorite, powerful data display software and do not wish to learn a new package. Output files from PCI-416SET are compatible to any popular spreadsheet or data base package, to mathematical analysis software and to scientific signal processing or plotting software. You could also write simple BASIC or C programs to process the A/D files and build your own graphics from the Borland or Microsoft graphics libraries.

When using PCI-416SET, you must still understand how the PCI-416 operates to be successful. You must still study this manual. Although PCI-416SET hides many register details, you must still learn what the registers do.

If you decide to write your own software, run PCI-416SET just to verify the board and become familiar with the hardware. If you a have a suspected hardware problem, we will ask you to verify operation under PCI-416SET.

Method 2 - PCI-416SET Start and Exit Mode

This option combines the best of both worlds. Use the menus from PCI-416SET then send data to your own program. This allows the PCI-416 to run concurrently with your application and can run the PCI-416 in non-stop streaming mode. And it saves a lot of unnecessary program development. You do not need to change any code inside PCI-416SET since Start and Exit is a built-in menu mode. You can even run Start and Exit mode completely unattended in the "Autorun" command line mode so that a user does not have to enter any menu items.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Start and Exit mode sets up the board, programs all registers, starts the A/D sampling then it exits, leaving the A/D running. Immediately after Start and Exit quits, your program collects data directly from the PCI-416 FIFO data register. You are free to choose how to be notified that data is ready (interrupts or status) and how to do the transfer (PCI Bus Mastering or programmed I/O block move) and where to send the data. You can use routines already in PCI-416SRC to do most of this work. Since data is sent to the FIFO first, there is a little time for your program to load and run after Start and Exit quits. If your program takes enough time to run that the FIFO overflowed, simply reset the FIFO without stopping the A/D and resume collecting data. You can also set up the system where Start and Exit runs then the A/D does not start sampling until an external trigger is received. This avoids *any* loss of data.

Method 3 - Develop Your Own Program

Writing your own code obviously takes the most effort but gives you the highest degree of configuration and control over the PCI-416. We will not spend much time with this since comprehensive code examples exist (i.e., PCI-416SRC). Study these heavily commented source files thoroughly and extract as much code as you need. These systems are written mostly in "C" and some Assembly. For other languages, use these sources as a general guide to the suggested sequence of operation. One enormous advantage of this source code is that you can see how it operates and make changes based on what you can see.

Modern full screen symbolic debuggers such as Borland, will let you watch the inner workings of these programs to simplify the job of making changes. Please be aware that DATEL will answer brief questions about these sources. However, extended custom applications generally are not offered by DATEL except under special agreement.

Obviously you should be a competent programmer to write your own code. Using the PCI-416 with large amounts of memory involves issues of PCI controller programming, protected CPU mode, descriptors, extended memory access and management. These are beyond the scope of this manual. Consult recent texts on these subjects.

Interleaved Applications

The Start and Exit mode offers concurrent interleaved processing while A/D sampling continues simultaneously. The basic strategy is simple. Collect a block of A/D data from the FIFO and then, while the next block is filling the FIFO, do some other tasks *in incremental fashion* fast enough to catch the next FIFO block and prevent the FIFO from overflowing. Since these "other tasks" can range from disk I/O to 3D graphics to DSP, many other system issues are at work. These include memory allocation, queueing other controller boards, etc. These topics are beyond the scope of this manual. The important point however is that the "other host tasks" must be broken up into small enough increments that they can be partially processed *in between A/D FIFO block transfers without dropping samples.*

About Windows

Windows poses severe performance penalties when the message queue is accessed. These delays are not apparent for slower A/D boards. But in the megahertz region, Windows will drop data. You may run Windows in the "locked" mode where the message queue is not accessed during data acquisition. But this is not considered good Windows programming form and may have other unknown side effects. ***Do not* run PCI-416SET under the MS-DOS icon in Windows.** See DATEL's PC-414 ISA A/D board and PC-414LV LabVIEW software for very complete Windows A/D systems, all sharing the same analog front ends with the PCI-416.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.1 Application Design Strategy

Before getting into details of programming, develop an overall strategy for how the A/D data will be used, system timing architecture, CPU mode and how memory will be set up. The most important of these is to work out overall system timing. If timing is carefully analyzed, you can determine where performance limits may occur before writing any code. The PCI-416 is faster than any peripheral and may be faster than some implementations of PCI bus. So the PCI-416 generally will not limit your performance.

The big issues for application design are:

- How will the data be used, how will it get there?
- Is data being saved? How? What format? Is it saved *during* data acquisition or all at the end *after the A/D has stopped?*
- What kind of events and data processing occur simultaneously? What is the time sequence of each task?
- What are the sizes and types of data structures - buffer lengths, bytes per file, etc. How are they managed?
- Where does hardware triggering and A/D clocking come from? How does it relate to starting the data acquisition software?

A most important concept here is ***no lost data***. In fact, we want *positive overflow detection* (the PCI-416 FIFO flags give this) and perhaps even, tagging of the first invalid data frame.

Users often look at an application and realize they want to do several tasks simultaneously, such as A/D sampling and FFT screen graphics. However there is only one CPU in the system and it can only do one thing at a time. All multiple task operations are really rapidly switched in a single-CPU environment. It is not sufficient to magically say, "somehow the CPU does all this together". You must design your application so that tasks and data records are split into regular small groups which are processed in round-robin fashion such that samples are not lost and everything gets processed in sequence.

Each programmer writes code in his/her own way, however it is helpful to write down the sequence of events at the top level in the simplest way. Resist the urge to start coding as long as possible. Clearly flag areas where simultaneous processes occur, such as A/D collection plus FFT graphics. Keep breaking each task down into finer parts until you have enough detail to start coding. Answer all unknown timing sequence or data and I/O management type questions *before* you start coding.

Determine the sizes of data structures, buffers and memory allocation. Too small a buffer forces small, frequent inefficient transfers with lots of loop overhead. Data blocks that are too large cause dropped samples. Usually the size of half the FIFO is the correct size of a block transfer. Compute the time required to fill the largest buffers. For disk I/O, you must measure your system over a large number of samples since each disk block may have a significant range of transfer times. You may need a high speed system timer to test the real throughput. The PCI-416 includes an I/O port which can be written to as a speed diagnostic. Connect an unused outport bit to an oscilloscope to flag loops in your code.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chain of Buffers

Another concept you need to be clear about is *linked data structures*. This is not linked lists as seen in programming texts. What we mean is data that is passed through a number of processes and different buffers on its way to the end usage.

For example, assume you wish to plot A/D FFT graphics. The first transfer is from the FIFO memory to an intermediate memory buffer in the host. If this is multichannel data, it must be demultiplexed ("unraveled") into separate buffers for each channel. Then each channel buffer is converted from integer A/D values to floating point. But floating point is typically 4 bytes per sample versus 2 bytes for integer data. So the data is converted *then moved* to another buffer. Next we perform an FFT on the data and the output remains as floating point so it probably remains *in place* (it doesn't move). But we're not done yet. Next we need pixel data to plot our graphics and we convert back to an integer format *compatible with our graphics display*. This pixel array is moved to the display adapter's *video RAM*.

Now let's go back and examine the design issues. Since many items are occurring simultaneously, some *swapped buffers* or *circular buffers* may be used to keep the data in temporal order. Also we may be memory limited so the same memory may be reallocated as one function completes then another starts. And certain memory, such as video RAM, may be very slow. Here are the important design issues:

- Keep all data *in **both** channel number and temporal sequence*. Generally this means the data buffers or files will have channel addresses 0, 1, 2, ... 7, 0, 1, 2 ... (note the channel address wrapping around using modulo arithmetic). And the oldest data will be first (lowest) in the data buffer.

- In the aggregate, downstream processes must be *faster* than upstream to avoid sequence overrun.

- Manage resources, remembering to deallocate memory for the next process. Make sure your buffer addressing, especially indexes and offsets, is exact.

- Keep processes insulated from each other to avoid data contamination and memory leaks and then eventual system halt.

- Make sure no one process delays so long that other processes are starved for data. If you are polling for the FIFO half full flag, data is not getting flushed to disk and the screen isn't being updated. Make sure a polling loop does not cause a system lockup if the flag never appears. Consider periodically polling the keyboard to exit.

- Write your code mostly in higher level language but don't hesitate to use Assembly language where you must have speed. If you are not skillful writing tight mixed code, get experienced help. High performance data acquisition is some of the most demanding software that is written. *You cannot obtain maximum performance on the PCI-416 using **ONLY** high level code.* Some of it (at least the FIFO block transfers) *must* be in assembly language for the highest available speed.

- Be aware of the data format needed by the next downstream process.

- And most important, *don't lose data.* Flag an overflow right away and handle the error so as to minimize the loss of data and aid diagnosis of what went wrong. Be sure to mark the *location* of the lost data. This may be an address or sample number.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.2 FIFO A/D Memory

The First-In, First-Out (FIFO) A/D memory on the PCI-416 offers extremely high speed, automated addressing controls and very compact board area. The FIFO A/D memory should be thought of as a temporary holding place for the A/D data until it can be transferred to the PC's system memory. The A/D tends to run at a constant clock rate or a repeating periodic clock rate whereas the host computer system is most efficient if data is transferred in high speed blocks of multiple samples. The time between these block bursts is available to the host PC to process the A/D data or for other tasks. If this time collapses to zero, the system has reached it's speed limit. The FIFO decouples these block bursts from the precise timing of the A/D. The FIFO acts as a spring or shock absorber between these two timing systems.

FIFO Flags

The FIFO memory contains three output signals which appear in the status register. The **Empty** flag says there is no more data in the FIFO and the entire FIFO capacity is available to be filled. The **Half Full** flag indicates that the FIFO contains at least one half the FIFO capacity *plus one sample.* For a 4K FIFO, the half full flag is asserted when there are *2049* (not 2048) or more samples loaded. Half full will still be asserted when the FIFO is full. The **Full** flag indicates that the FIFO is filled to capacity and cannot accept any further data. If the FIFO is not read or reset, the next A/D sample will be lost. Note the *reverse coding* of the FIFO flags in the status register. These flags track the capacity status, whether data is being filled from the A/D or emptied by PCI bus. If both reading and writing of the FIFO occur simultaneously, these flags may rapidly toggle around any one of these conditions. The FIFO may be emptied by either reading all samples or reading the FIFO reset register. If the A/D is still running after emptying the FIFO, it will start filling again with the next A/D sample.

Standard Method

After starting A/D sampling, the standard method of using the on-board FIFO is to *wait* until the FIFO is half full then do a block transfer as fast as possible to the host memory. This method offers the fastest total system speed and avoids loss of data from FIFO overflow because we have some time (about half a FIFO's worth) to respond to the half full and start a block read. Since the half full signal *guarantees* that at least half the FIFO is filled with data, the block transfer size should be one half the FIFO's total capacity or less. This will also prevent overreading the FIFO past empty (see discussion below). The block may read by bus mastering (fastest speed) or by a 32-bit string block move (easier to program).

The FIFO is designed to continue simultaneously collecting A/D samples during the block transfer and it keeps all data in proper temporal sequence. User software should do the transfer as soon after the half full signal occurs but obviously no later than when the FIFO is full.

Common FIFO Memory Misconceptions

Question: Why not just read samples from the FIFO as soon as the empty flag says there is data?

Answer: You can certainly do this but it is very slow because if you follow this procedure, you must poll the empty flag *with each sample read.* Doing the block transfer-at-half-full method avoids this time-consuming polling. The empty flag says there is data in the FIFO but does not say *how many* samples there are.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

One exception to this is if you are reading small blocks of less than half the FIFO and you *know* positively how many samples there are because you used the PCI-416's sample counter (or you externally counted A/D clocks). Here the algorithm is simple: Set up everything then wait for the ACQuire signal to indicate that a *counted* block of data is in the FIFO. Now read a block of known size out of the FIFO. You should test the empty flag again immediately after this block read to be sure the FIFO is now truly empty. Also if you misprogrammed the sample counter so that not *enough* samples were collected, the only indication of overreading the FIFO is that the last "samples" are all 1's (0xFFFF). See below.

If you decide to try this counted block method using the ACQuire signal, always make sure the FIFO is empty before starting A/D conversion. Either read all previous samples until the empty flag appears or clear the FIFO (all old samples will be lost).

Question: FIFO programming mostly seems to be built around transferring the FIFO in half blocks. Doesn't this leave the other FIFO half unused? Isn't it being "wasted"?

Answer: The half block transfer technique is mainly for *non-stop, "streaming" continuous* sampling (the most demanding application). You are certainly free to use all the FIFO if you know your sampling will stop before the FIFO is filled *OR* you don't care about samples which spill over.

The "unused" half FIFO serves a very useful purpose. As the system speed increases, the FIFO is gradually more filled with each transfer because there is a small delay between the half full signal and the actual start of the block transfer. And during this start up delay, the A/D is rapidly stuffing the FIFO. At very high speeds, each time the host comes back to get a block, it finds the half full flag already set (i.e., already more than half full). We don't care how much greater than half full the FIFO is *as long as it never overflows.*

At the highest speeds, the FIFO will reach almost full most of the time even though transfers are done in half FIFO sizes. Therefore the whole FIFO is most certainly used when demanding maximum performance.

Also if you wish to transfer blocks *smaller* than one half the FIFO, you will simply get the half full signals more frequently. But you will not lose data as long as the FIFO never overflows. If you use smaller block sizes, the setup instructions to transfer the block happen more often and so the whole operation is slightly less efficient since no data is transferred while preparing the setup.

Question: Okay, continuing that idea, why don't we just read more of the FIFO when it is more than half full?

Answer: Because there is no way to tell how *much* the FIFO is over half full and if you read the FIFO down past empty, you get meaningless information. It is possible to test the empty flag after each transfer and avoid reading the FIFO. But then you must test *every* sample and this takes much longer than doing a fast block transfer. Doing a half full block burst is the fastest method which avoids overreading and overflow.

Question: I have been running the system in streaming mode, collecting half-FIFO blocks. But when I stop the A/D, it might not half-fill the FIFO and some data would be stuck in the FIFO. How do I get it out using the half-block technique?

Answer: For the *last* group of data, you must switch to a different algorithm. First, you must have some indication that the A/D is no longer sampling. This can be the ACQuire signal you test with each block indicating that the total sample count was reached. Or it can be some external gate signal which you periodically poll. After the A/D stops, test the FIFO empty flag. Keep reading samples until the empty is true. As long as the A/D continues running, your host program will always get as much data as it needs. Some applications may not care about the last samples after the A/D stops.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Question: What happens to the FIFO when it overflows?

Answer: The FIFO will always deliver all the oldest samples *first*. Any data greater than the FIFO capacity will be lost. This means the most recent A/D samples attempting to write to an already-full FIFO will be dropped. If the A/D is still running when you empty an overflowed FIFO, you will get new data but you cannot tell where you lost the data. So the other big reason to prevent FIFO overflow is that you will lose track of the sequence of recently collected data. It's best not to let the FIFO unintentionally overflow.

Question: What happens if I try to read the FIFO all the way past empty?

Answer: You will get good samples down to the last one then all logic ONE's (0FFFF hex) from an empty FIFO. The empty flag is not latched. If the A/D is still running but you persist in reading faster than the A/D rate, you will get both empty samples and good ones. Remember that an all ONE's sample looks like a valid A/D sample at full scale.

You can also verify that no data was lost by plotting the output from a smooth sine wave input well below the Nyquist frequency limit. Look for discontinuities in the reconstructed data. For very large files which you cannot examine by eye, write a simple program which tests the absolute value of the number of counts of change between adjacent samples. This is a maximum as the sine wave crosses the zero axis. If you show hundreds or thousands of counts of change, you probably lost data. Examine the reconstructed output at the location where this differential became excessive.

Question: Sometimes when I read the FIFO half block and test it right after I did the block transfer, the empty flag says I didn't get all the samples.

Answer: These are samples that were collected *during* the block transfer. If the A/D was still running during the half block burst, these samples are stored in the FIFO in temporal order and they will be the first samples transferred during the *next* block transfer. This uses part of that extra half of the FIFO.

Question: But doesn't that mean that the FIFO will gradually overflow because each time, there are extra A/D samples I didn't read?

Answer: No! Those extra A/D samples simply mean that the half full signal will happen a little sooner next time. You will always get all the data as long as you never let the FIFO overflow.

Question: I want to get at sample N buried part way in the FIFO. Why can't I read it directly?

Answer: Because the FIFO has no addressing controls. The FIFO is not like standard read/write memory with random addressing. To read a specific sample, you must read the full block out to system memory then use standard offset addressing to read your sample. Also there is no tagging of the samples to indicate channel address. This is why sample counting and not dropping any samples is so important.

The FIFO data has no address tag encoded in the data. If you are having trouble figuring out if your data is correct, set several channels to known DC voltages. This will mark the data stream. After you have proved that your software is perfect, restore normal signals.

Question: Should I use interrupts to collect the half FIFO block?

Answer: The FIFO half full signal is both a pollable status register bit, and a maskable selectable PCI bus interrupt. Interrupts add a small processing delay therefore the highest possible speed is always done by using bus master transfers.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Consider what happens when the interrupt occurs. The system state must be saved by pushing multiple registers onto the stack in orderly fashion. Now the interrupt must be "processed". But what does this mean? Do we set a system flag somewhere (which ultimately must be polled *again)*? Or do we start the block transfer *within the interrupt service routine* (what happens if a higher priority task interrupts)? If the transfer is set up in advance as a PCI bus master burst, all of this sequence will be automatic. *After* the burst completes, the PCI controller indicates that the cycle is complete and the system should be prepared for the next burst transfer. Study the PCI-416SRC source code for details.

Question: I want all data from one channel put in a single buffer but the data from the FIFO is channel multiplexed.

Answer: The FIFO system has no hardware provision to "unravel" data by splitting it into channels. This must be done on the host side in software. DATEL's smart A/D-DSP boards (PC-430 series) can do such buffer separation on the fly. If you think about it, this is the kind of operation that can only be done by a CPU. There are many variables - channel count, number of samples per buffer, buffer addresses in the host (base or extended memory) and whether the data is presented in circular (ring) buffer format or as long linear buffers. Users often wish to combine other operations here such as decimation or averaging. So it makes sense to do this with a CPU.

You can also operate the PCI-416 in single channel mode by locking on one channel (set the autoincrement bit to single channel). For the A/D-per-channel analog modules (F, G, J, K, L, etc.), refer to the address coding.

Question: What's the best method to detect FIFO overflow?

Answer: The FIFO overflow signal is *latched* so you can test it whenever it's convenient. The signal is cleared by resetting the FIFO. If you test this flag with each block, you can identify an overflow as soon as it happens and stop the system if necessary. If you wait until all blocks have been read (many frames), you cannot tell *which* block had the overflow. But the full flag polling will take slightly less overall time.

5.3 Counter/Timing Systems

Please review the block diagram shown in the data sheet at this time. The PCI-416 has a series of clock systems and counters which do the following functions:

- A 24-bit **Sample Counter** collects up to 16,777,216 samples.
- A **Trigger Rate** timer determines how often a "frame" of samples is collected.
- A **Sample Rate** timer system sets the A/D start clock rate.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

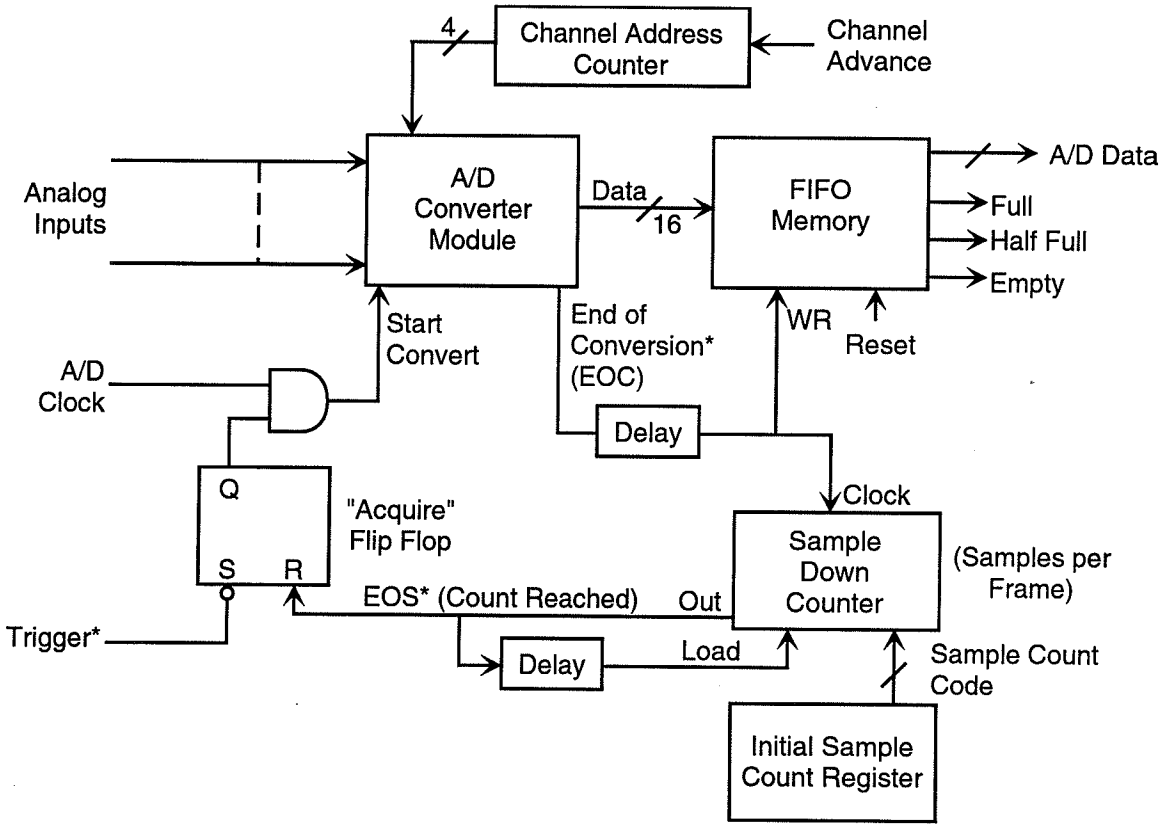


Figure 5.0 Trigger and Sample Counter

Now let's review the philosophy and strategy behind these systems and how they work together. The PCI-416 is designed with a high degree of automation using a local very high speed controller such that, once programmed, the board will keep running unattended for awhile without constant intervention by the PCI host PC. This strategy is ideal for multiple task applications where A/D sampling occurs *simultaneously and concurrently* with other processes such as disk writes, DSP math processing or graphics screen writes. The PCI-416 does this by freeing the host for other tasks during A/D sampling.

The PCI-416 collects data in **frames.** A frame is a finite set of A/D samples with the length (the number of A/D samples per frame) determined by the **Sample Counter**, set by the Sample Count Register. **A trigger starts each frame.** The rate of this trigger in frames per second is set by the **Trigger Rate** system. The time between each sample is set by the **A/D sample rate timer** which generates the A/D sample clock**.** Note very carefully that *the trigger is not the A/D sample clock.*

Once the frame length is set, each new trigger will always collect the same number of A/D samples until the host PCI PC stops and reprograms the system. Once programmed, the PCI-416 can either run once to collect N number of A/D samples then stop. Or it can do repeating frames, each with N A/D samples. Or the system can run ***forever*** in non-stop, streaming applications with the frame concept disabled.

5.3.1 Frame Architecture and Channel Addressing

The frame length sample counter can be set up to 16 megasamples or the sample counter can be *disabled* such that an infinite number of samples are collected. The composition of any one frame can be a single channel or can be multiple channels. If the system is single channel only (by disabling the automatic channel advance using Command Register bit 4), it will stay locked on that channel unless there is time for the host PC to reprogram the channel address after each frame.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If the frame is multichannel using one of the single-A/D-MUX modules, the channel address will automatically increment ("*autoincrement*" mode) with each A/D conversion (but also refer to the channel addressing information for the "F, G and J" modules). The first channel sampled will be the address *first* loaded into the **Channel Address Register**. After the channel address reaches the highest channel, the address will **wrap** modulo style back to channel zero. For example, a 4-channel system will have an A/D data stream with addresses - 0, 1, 2, 3, 0, 1, 2, ... Notice the automatic reset back to channel zero each time. An 8-channel system would rotate around channels 0 to 7, 0 to 7, etc. And 16-channel models go 0-15 0-15, forever.

Combining the number of samples per frame with the channel capacity causes *modulo division* of the sample stream. Therefore, if you want the channel addresses to wrap at orderly boundaries, **always make the samples per frame an exact integer multiple of the channel capacity.** And remember to start at channel zero or program the channel address register at the desired starting address..

For example, using our 4-channel system above, starting at channel zero, *each trigger* will collect a frame of 4 samples, addressed 0-3. The resulting stream into the FIFO memory is as follows:

(trigger) channels 0-3
(trigger) channels 0-3
(trigger) channels 0-3
(trigger) channels 0-3
and so on ...

Each sample is converted according to the A/D start clock rate. Therefore **the trigger acts as a gate to the constantly-running A/D clock**.

IMPORTANT: If command bit 4 = 1, the channel address *automatically resets to zero* after each **frame**. This is caused by the ACQuire signal being reset and it occurs regardless of whether the channel address automatically wrapped. At the same time, the frame length sample counter *automatically reloads* after each frame is collected. If you want to start at some channel other than zero, you must reprogram the channel address *after* the previous frame completes and *before* the next trigger.

Now let's make some changes starting with a frame length of 8:

(trigger) channels 0-3, 0-3
(trigger) channels 0-3, 0-3
etc.

The frame has 8 samples with *each channel sampled twice.* The channel address *automatically* resets from 3 to 0 within the frame and also at the end of the frame. Notice also that any one channel is collected two times *at the A/D clock rate* then there is a delay until the next trigger starts a new frame.

Now some more examples. Let's *disable* the sample counter:

(trigger) channels 0-3, 0-3, 0-3 , ... "forever"

This will collect an infinite number of samples, all spaced by the A/D clock. Your system simply has to collect them all without losing data. Use the FIFO flags to collect each block. These flags will also warn you if data is dropped. Host software stops data collection when all desired samples have been taken. The "forever" mode is done by selecting the "pretrigger" bit (Command 6 = 1) then *do not* send a trigger. See the Command register description.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Or, if you inhibit the channel advance and stay locked on one channel for example:

(trigger) channels 2, 2, 2, 2, 2, ... "forever"

And if needed, you can set the frame length to some large number:

(trigger) channels 2, 2, 2, 2, ... 2 <1 million samples>
Reprogram the channel address.
(trigger) channels 3, 3, 3, 3, ... 3 <1 million samples>
Etc.

If you want to start at some channel other than zero, use the following 4-channel example:

Set channel to 1
(trigger) channels 1,2,3 and wait for ACQuire flag (causes reset to channel 0)
Set channel to 1
(trigger) channels 1,2,3 and wait for ACQuire flag (causes reset to channel 0)
Etc.

This technique obviously takes a little extra time to reset the channel address each time. The ACQuire flag is detected by polling the status register or by interrupt. For some multiplexer type analog modules (especially ADM-12E with some gain added), additional settling time must be allowed before starting a new frame. Do this by selecting a somewhat lower A/D sample rate.

You may also achieve the same result by simply collecting all four channels then discarding every channel 0 sample. This increases the size of the saved data but simplifies the programming.

In some data acquisition literature, a group of all available channels sampled together is called a ***SCAN***. If more or less than the channel capacity is collected with each trigger, that is no longer considered a scan, which is why the more general term *frame* is used.

IMPORTANT: The frame length has nothing to do with the FIFO size. These two systems operate independently. Frames are counted on the input side by the sample counter. The FIFO contains internal address logic which counts FIFO samples and issues empty, half and full signals. In the host-side software, whatever is used to do block transfers (counted string move instructions or PCI bus master transfer) uses its own word-count mechanism. The combination of all these counting systems may mean that blocks are moved before frames finish or vice versa. Therefore each block transfer out of the FIFO to the host may deliver full frames only or one partial frame, according to your software. *No data will be lost as long as the FIFO never overflows.*

5.3.2 Triggering versus A/D Clocking

Both the A/D clock and the trigger can be derived from internal or external sources. If both are internal, they will have a fixed phase relationship. The first A/D sampling occurs at a very small but fixed time after the trigger set by a few logic propagation delays. If either the trigger or the A/D clock is *external*, it is likely that they will be out of phase. This can mean that the next A/D sample will start at an undetermined delay after the trigger but in no case will this delay exceed the A/D sample interval. Obviously for low A/D rates, this delay can equal the long sample delay. The problems here are the possibility of long delays and the unknown aspect of the delays.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If you use an external trigger and require a known maximum delay before the first A/D sample, also use an *external* A/D clock which is phase-related to the trigger.

IMPORTANT: **DO NOT USE THE TRIGGER FRAME RATE AS THE A/D SAMPLE RATE.**

Do not set the frame length to one then use the trigger as the A/D start clock. The reason for this is that some logic delays are required to set the system up for the next trigger. If the trigger is run at very high speed, A/D samples may be dropped or skewed. If you need very high speed A/D sampling, program the sample counter (or disable it in pretrigger mode) then use the external A/D clock input.

5.3.3 Triggers

Referring to the system block diagram and the PCI-416 register descriptions, the system consists of a 40MHz local oscillator, several programmable dividers and a frequency synthesizer section. Either internal or external triggers are selected by Command bits 1 and 0. Internal triggers are derived from the 10 MHz oscillator divided down by two 16-bit dividers connected in series. These dividers are Timer 0 (first) then Timer 1 (cascaded from Timer 0) of the 82C54 divider.

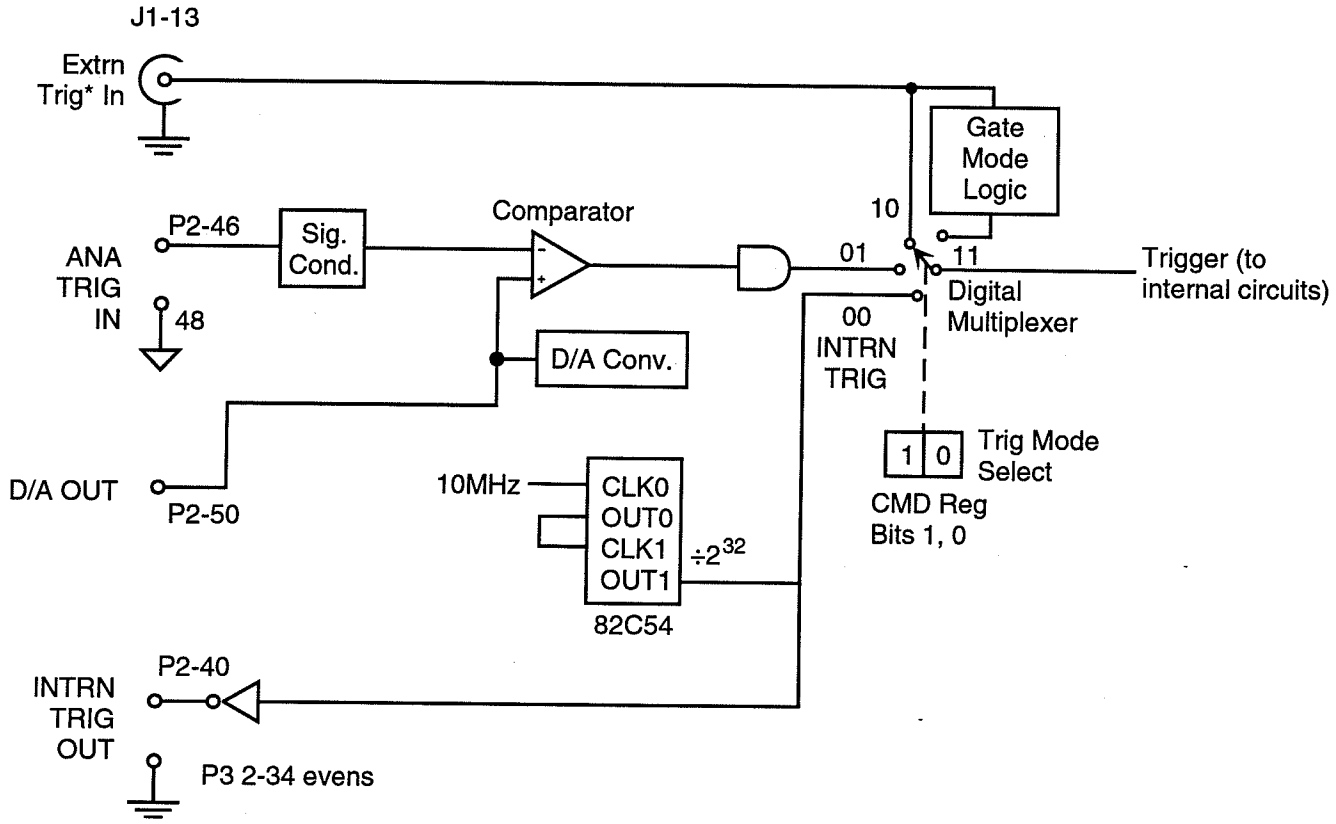


Figure 5.1 Trigger Systems

Programming Guidelines for the 82C54 Timer/Counter:

- Refer to the detailed section on the 82C54 for full details. The 82C54 can be programmed in many ways. However only a few modes are useful for the PCI-416.

- The 82C54 requires software delays between program access to its registers so that command states can propagate through the chip. See the source code examples for suggested delays. These delays are especially important with high speed CPU's. If you fail to add these delays, the PCI-416 may work improperly.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- The 82C54 is a *byte-programmed* device requiring "char" data types in "C" but loaded as 32-bit words. Its data registers are internally 16 bits wide but must be programmed with two successive 8-bit writes with delays between each access. Read the 82C54 section for details on this byte sequence.

- For the internal trigger, *both* Timer 0 and 1 *must* be programmed. You may use either timer to do the amount of binary division you need. *Each* timer requires two 8-bit writes to load its data register (4 bytes total). The resulting trigger rate is:

Internal Trigger rate = 10MHz / [(Timer 0 divisor) X (Timer 1 divisor)]

Failure to program *both* timers will inhibit triggers and perhaps cause undefined operation. These two timers run in cascade.

- The internal trigger either runs in **Continuous** (repeating) trigger mode or **Single** trigger mode. Single mode sends only *one* trigger. These modes are as follows:

For Continuous trigger, use 82C54 **Mode 2.**
For Single trigger, use 82C54 **Mode 5.**

- In mode 2 (Continuous trigger), ***the minimum count value is 2*** (a count value of one is *prohibited*). Therefore each 16 bit divider gives at least a divisor of 2 and in cascade, **the maximum internal trigger rate is 2.5 MHz** (10 MHz / 4). Remember, this is the trigger rate, not the A/D clock rate.

- The 82C54 control word byte register contains several bit-encoded fields which select the mode, the timer/counter to be loaded, read/write sequence and BCD/binary (see the 82C54 section for details). All bits must be assembled properly then written all together at one time. You cannot write to individual bits randomly. This "bit building" is sometimes done in several program variables then assembled in a "shadow register" variable. See the source code for examples.

- The maximum input frequency to the 82C54 is 10MHz. The on-board circuits to the 82C54 never exceed this frequency however it is possible using the external clock input.

Digital Trigger

Command bits 1 and 0 also select three other modes - external digital trigger, external gate trigger and external analog trigger.

The external **digital** trigger (Command bits 1,0 = 01) accepts a TTL digital input pulse on the rear panel DB-25 analog input connector, pin 13. This connection applies to all other external triggers.

The digital trigger occurs on the ***falling*** edge and must be 100 nanoseconds minimum width. *The maximum width should not exceed the expected frame length.* Otherwise, this may interfere with setup for the following frame.

5.3.4 External A/D Start Clock Operation

Before selecting the external A/D sample clock mode, enable external A/D clocking with Command Bits 1, 0 = 01. The start clock logic must see the first falling edge to logic "0" for proper initialization. If the system is triggered while the start clock input is at logic "0", the logic must fully cycle around so that a falling edge is detected, meaning that the first rising edge A/D sample will be ignored.

After the falling edge to logic "0", the sample clock input may be held at logic "0" indefinitely until the first A/D sample is needed. The external clock may also run continuously.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

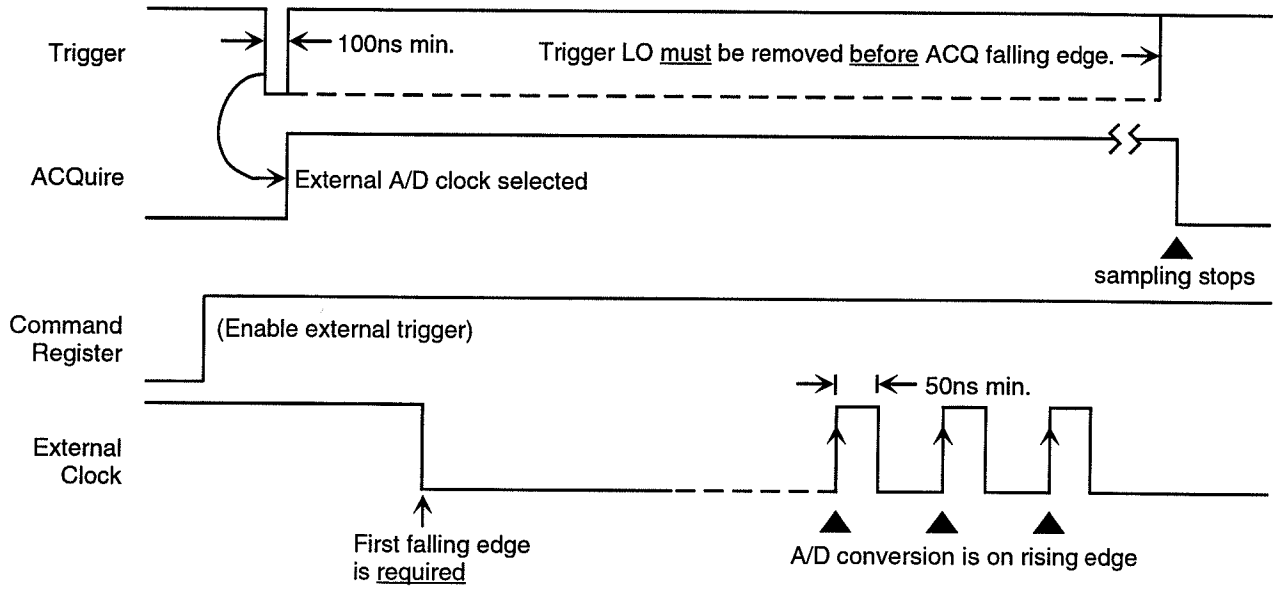


Figure 5.2 External A/D Start Clock and Trigger Timing

Gate Trigger

The external **gate** trigger (Command bits 1,0 = 11) is a variation of the digital trigger. A/D samples will be collected *as long as this gate is low* and stop when the trigger returns high. Use a sharp (TTL) falling edge to start the gate. If you combine external triggering with internal A/D clocking, you may get one A/D sample more or less than expected.

Analog Trigger

The external **analog** trigger (Command bits 1,0 = 10) is compared against a data value written into the D/A converter. This analog input is connected to P2, pin 46. Note that this input is ***not*** internally connected to the A/D signal input. If you want to trigger on the signal input to the A/D converter, you must connect P2-46 to the A/D input externally. The analog trigger occurs as the external signal crosses the reference voltage in the ***positive-going*** direction. If you need to reverse this phasing, use an external inverting amplifier on this analog trigger input.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

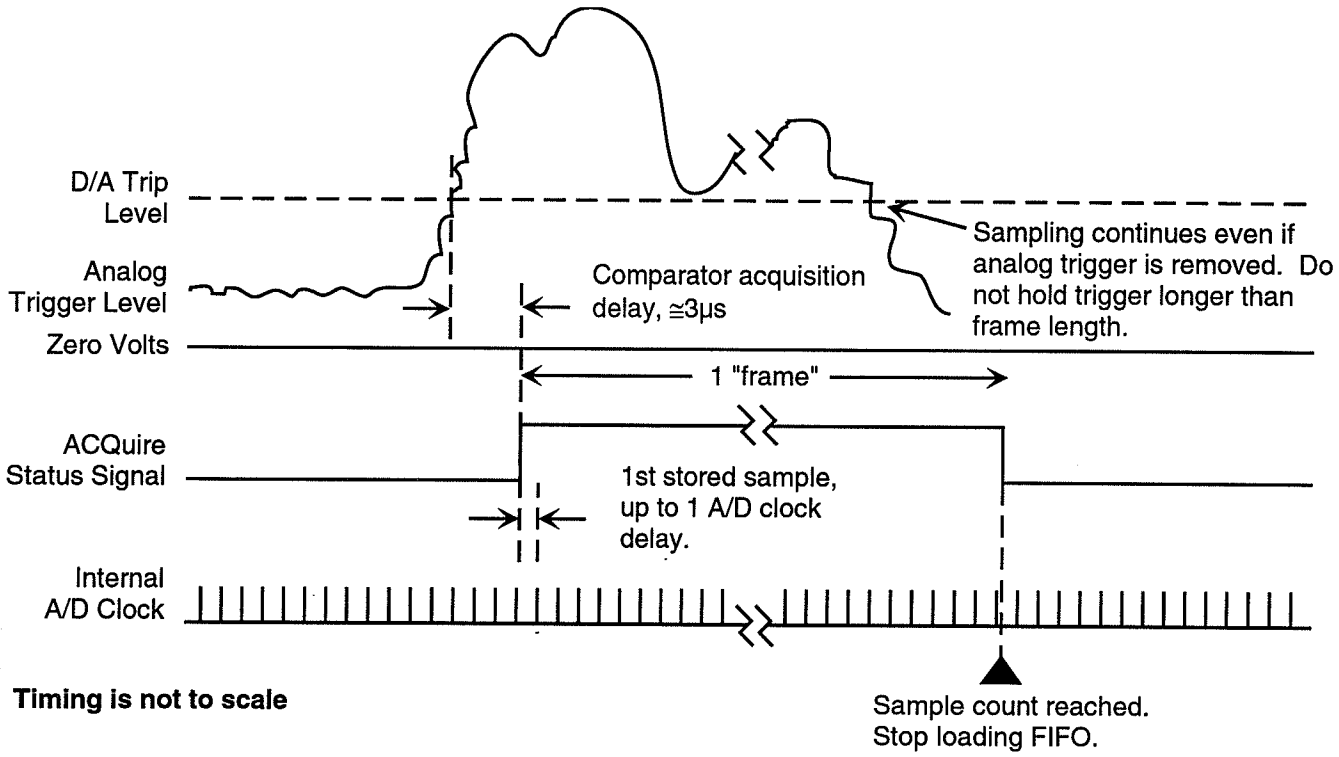


Figure 5.3 External Analog Triggering

The analog comparator and DAC together may have a small offset value of a few millivolts, therefore consider setting a lower or higher level to be sure to catch the first samples after the level passage. The comparator output can be monitored as bit 16 in the status register.

After the analog trigger, A/D sampling remains active as long as the frame counter keeps the ACQuire gate open. Like the digital trigger, this will cause problems if the frame length is shorter than the analog trigger. If you use the analog trigger to start a frame from a noisy input signal, be aware that a noise spike can cause a false start. Also, the actual A/D sampling starts after a short acquisition delay of about 2-3 microseconds through the analog comparator. Keep the trip level near the trigger signal to reduce this delay. Like the external digital triggers, A/D sampling can start up to one sampling interval *after* the trigger crosses the comparator threshold if you are using an internal A/D clock. A *single* analog trigger cannot start multiple frames. Each new frame is started by a trigger.

NOTE: There is no "one shot" function on this analog trigger. Make sure the analog trigger reverts to a "non-triggering" status (below the threshold) ***before*** the frame count reaches zero. If the analog trigger is still active when the frame count ends, the A/D will continue sampling indefinitely. A *workaround* for this anomaly is to count FIFO blocks externally and stop the system by software.

"Data Level" (Software) Triggering

As a system strategy, if you cannot tolerate these small acquisition and triggering delays, consider collecting data continuously in a *circular ring buffer* in host memory. If enough CPU time is available, test each sample in software and mark the address offset where the first sample exceeds a threshold value. Then fill the ring buffer and stop writing to it when the circular address wraps around and the first (oldest) triggered sample is reached. Data level triggering also allows you to *adaptively* trigger on several levels at different points in the sampling session.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.4 A/D Sample Clocking

The Command Register bits 13-11 determine the A/D clock source. Review the register section for details. The A/D sampling can be derived from four main choices. They are:

- From the on-board programmable frequency synthesizer with several ranges.
- Fixed frequencies directly from the crystal oscillator or divided down by the 82C54 programmable counter.
- From the frequency synthesizer plus the 82C54 divider in cascade.
- From an external A/D sample clock.

The frequency synthesizer offers several ranges with high resolution. The crystal oscillator gives slightly lower overall harmonic noise but does not have as many frequency choices. The 82C54 uses **Counter 2** as a 16-bit binary divider to any input clocks and the same programming guidelines listed above for this device should be observed. Remember the programming delays needed for the 82C54.

82C54 Counter 2 should be programmed with ***Mode 2*** by loading the control word register with **B4** hex (180 decimal). The 16-bit divisor value in mode 2 is given by:

Divisor = InputClkFreq / OutputFreq

where the Input Clock Frequency is either 10 MHz or the Frequency Synthesizer Output divided by 4. This is an integer operation and obviously any division with non-integer results will truncate the remainder. Therefore only certain frequencies are available with the oscillator input. This 16-bit divisor *must* be divided into an upper byte and lower byte before loading into the 82C54 as follows:

MSB = Integer part of (Divisor/256)
LSB = Divisor modulo 256 (remainder part)

The *minimum* divisor in mode 2 is *two*. (This is for the whole divisor, not each byte portion). Therefore the highest A/D frequency in the crystal oscillator mode is 5 MHz. For example, using the crystal oscillator, assume you need a sample rate of 1 MHz. This is:

Divisor = 10 MHz/1 MHz = 10. The MSB is 10/256 = 0 and the LSB is 10.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Notice that the MSB must be loaded, even though it is zero. The diagram below indicates how the A/D clock system works. Note that you must program several fields in the Command Register, in the Frequency Synthesizer and in the 82C54 to use different parts of this system.

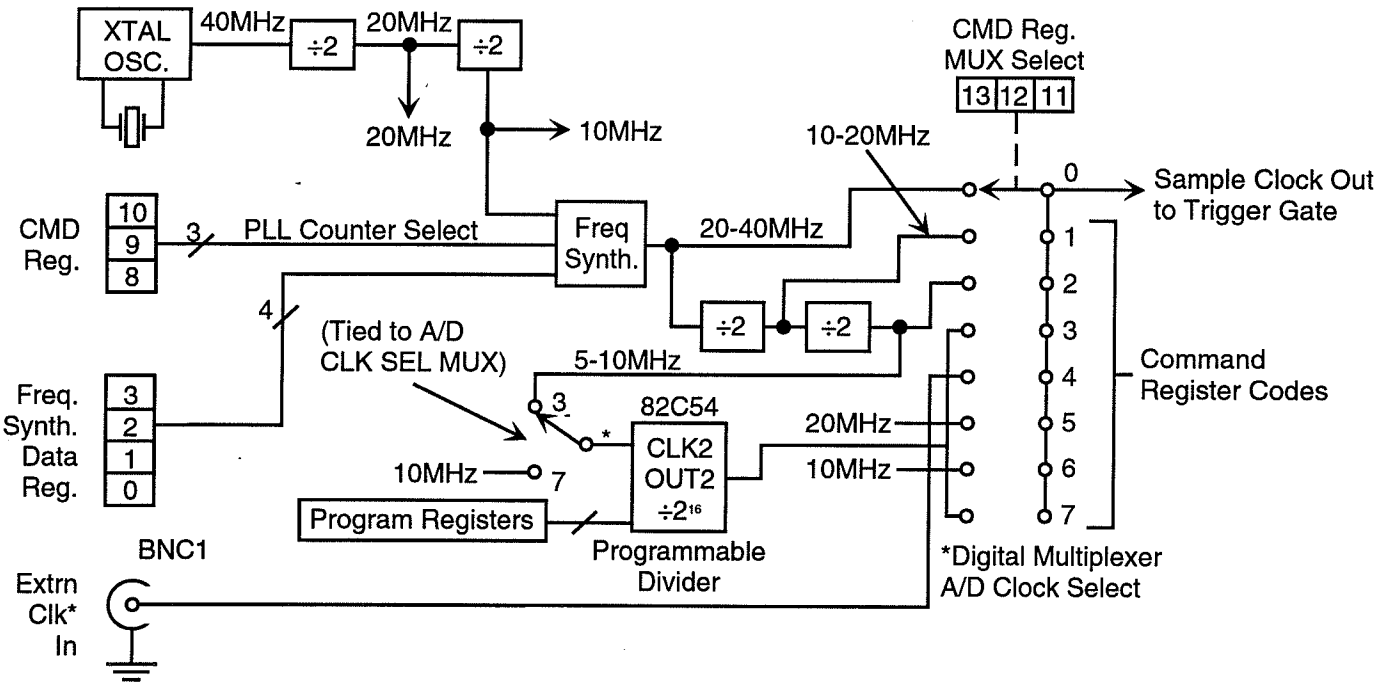


Figure 5.4 A/D Sample Clock System

5.4.1 Frequency Synthesizer Operation

The frequency synthesizer section generates an output clock frequency in the range of 20MHz to 40MHz - which is further divided by a 2 bit counter to give 10 - 20MHz and 5 - 10MHz clock rates. This clock drives the A/D clock directly or the 82C54 timer clock inputs (max. 10MHz allowed here) - selection is made by writing bits 13-11 in the Command Register. The synthesizer consists of a reference oscillator an programmable divider, a digital phase detector, Voltage Controlled Oscillator (VCO), a frequency down converter (prescaler), and feedback divide counters. The linearized digital phase detector enhances transfer function linearity. The loop is in lock (stable output signal) when the reference and fed back signals are of the same phase and frequency. When this happens is determined by programming the R, N, and A counters to give you the desired output frequency. Host software accesses the synthesizer over a 4 bit data bus via the Frequency Synthesizer Register at BADR2, ARL = 16 (Write only).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

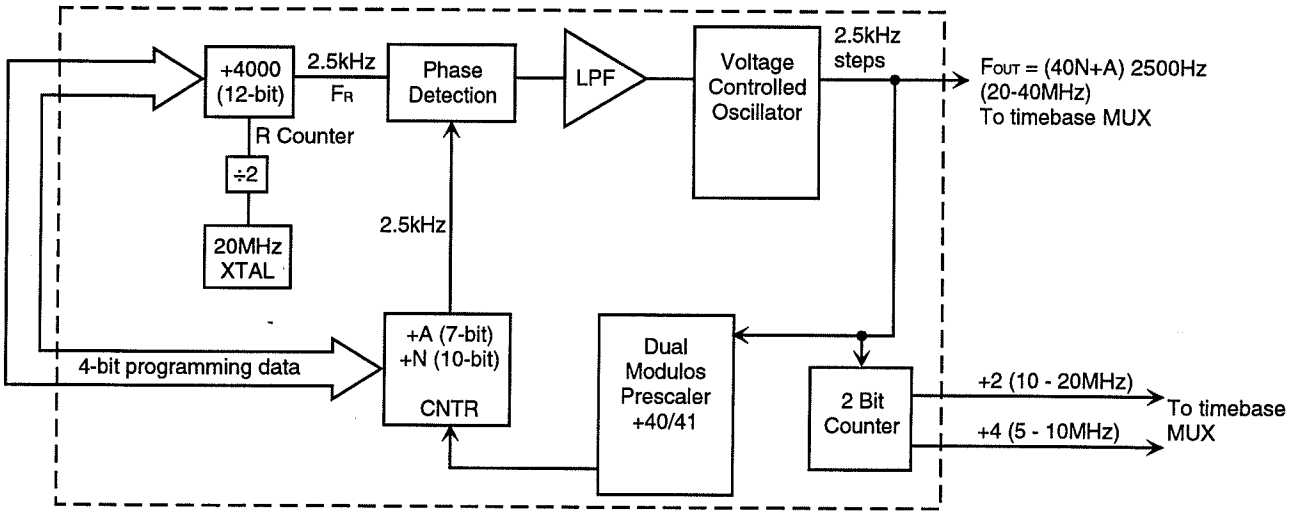


Figure 5.5 Frequency Synthesizer Block Diagram

The input is driven by a 10MHz clock, i.e. 20MHz oscillator divided by 2. The input clock is immediately divided by 4000 (this must be set by host software) by the Reference Counter to give a reference frequency of 2.5kHz. The reference frequency dictates the output frequency resolution in 2.5kHz steps. For the lower ranges, these steps are 1250kHz or 625Hz.

R is the input clock divide ratio - always set to 4000 by host software.

The synthesizer output frequency can be calculated by the formula:

$$FREQout = (40N + A) \times 2500Hz$$

N is the value loaded into a 10 bit counter, the range is 200 to 400.

A is loaded into a 7 bit counter with a range of 0 to 127.

R, N, and A data are loaded by host software to the 4 bit Frequency Synthesizer Register at BADR2, ARL = 16. The individual selections are made by writing bits 10, 9, and 8 in the Command Register as follows:

Command Bits	10	9	8	
	0	0	0	Counter A, bits 0 . . 3
	0	0	1	Counter A, bits 4 . . 7
	0	1	0	Counter N, bits 0 . . 3
	0	1	1	Counter N, bits 4 . . 7
	1	0	0	Counter N, bits 8, 9
	1	0	1	Counter R, bits 0 . . 3
	1	1	0	Counter R, bits 4 . . 7
	1	1	1	Counter R, bits 8 . . 11

When programming the Frequency Synthesizer Register the Command Register MUST be written with the appropriate code before EACH access.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example #1: Set the synthesizer output to 25,000,000 (25MHz)

Calculate N and A. R is fixed at 4000.

The output frequency = (40N + A) x 2500.

(40N + A) x 2500 = 25,000,000

40N + A = 10,000

Setting A = 0, 40N = 10,000, N = 250

Therefore, R = 4000, N = 250, A = 0

Example #2: Set the synthesizer output to 25,005,000Hz (25.005MHz)

Calculate N and A. R is fixed at 4000.

The output frequency = (40N + A) x 2500

(40N + A) x 2500 = 25,005,000

40N + A = 10,002

Cannot set A = 0 otherwise N = 250.05

Set N = 249, A = 10,002 - (40 x 249)

= 10,002 - 9960

= 42

Therefore, R = 4000, N = 249, A = 42

5.4.2 Frequency Synthesizer Programming

Loading each of R, N, and A is a two step procedure:

1. Write the Command Register to select the counter

2. Write the counter data to the Frequency Synthesizer Register at BASE + 6

Because the Frequency Synthesizer Register is only 4 bits wide it must be written 3 times for each of counters R (12 bits) and N (10 bits), twice for counter A (7 bit).

Example #1: 25MHz output

From Section 5.4.1 we already calculated R, N, and A to be:

R = 4000 (FA0 hex) N = 250 (FA hex) A = 0

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Because the Frequency Synthesizer Register is only 4 bits wide (1 nybble), these numbers must be divided for multiple 4-bit accesses:

R (bits 0 . . 3) = 0	R (bits 4 . . 7) = A hex	R (bits 8 . . 11) = F hex
N (bits 0 . . 3) = A hex	N (bits 4 . . 7) = F hex	N (bits 8 . . 11) = 0
A (bits 0 . . 3) = 0		

Example #2: 25.005MHz output

From Section 5.4.1 we already calculated R, N, and A to be:

R = 4000 (FA0 hex)	N = 249 (F9 hex)	A = 42 (2A hex)

Dividing each of these into 4 bit nybbles gives us:

R (bits 0 . . 3) = 0	R (bits 4 . . 7) = A hex	R (bits 8 . . 11) F hex
N (bits 0 . . 3) = 9 hex	N (bits 4 . . 7) = F hex	N (bits 8 . . 11) = 0
A (bits 0 . . 3) = A	A (bits 4 . . 7) = 2	

5.5 Sample Counting

The PCI-416 sample counting system collects a single frame up to 16 million samples long or repeating frames each up to 16M samples, with every frame started by a trigger. This system offers a high degree of automated, unattended running which is adaptable to most applications. When a frame completes sampling, the system automatically reloads the frame count into the sample counter and will wait until the next trigger to start the next frame. The system can also run "forever" or can be set up for pretrigger ring buffering. About the only mode the system cannot do is repeating frames started by only one single trigger. You must have a trigger to start every frame.

Some system notes:

- The internal A/D clock runs continuously and sample loading to the FIFO is gated with ACQuire.
- The channel address counter (for MUXed A/D modules) is gated with ACQuire and EOC.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP