
COMPREHENSIVE SERVICES
We offer competitive repair and calibration services, as well as easily accessible documentation and free downloadable resources.
SELL YOUR SURPLUS
We buy new, used, decommissioned, and surplus parts from every NI series. We work out the best solution to suit your individual needs.
Sell For Cash
Get Credit
Receive a Trade-In Deal
OBSOLETE NI HARDWARE IN STOCK & READY TO SHIP
We stock New, New Surplus, Refurbished, and Reconditioned NI Hardware.
APEX WAVES
Bridging the gap between the manufacturer and your legacy test system.
1-800-915-6216
www.apexwaves.com
sales@apexwaves.com
All trademarks, brands, and brand names are the property of their respective owners.
Request a Quote
CLICK HERE
AT-GPIB

AT-MIO-16F-5
User Manual

Multifunction I/O Board for the PC AT/EISA

February 1994 Edition

Part Number 320266-01

National Instruments Corporate Headquarters
6504 Bridge Point Parkway
Austin, TX 78730-5039
(512) 794-0100
Technical support fax: (800) 328-2203
(512) 794-5678

Branch Offices:
Australia (03) 879 9422, Austria (0662) 435986, Belgium 02/757.00.20, Canada (Ontario) (519) 622-9310,
Canada (Québec) (514) 694-8521, Denmark 45 76 26 00, Finland (90) 527 2321, France (1) 48 14 24 24,
Germany 089/741 31 30, Italy 02/48301892, Japan (03) 3788-1921, Netherlands 03480-33466, Norway 32-848400,
Spain (91) 640 0085, Sweden 08-730 49 70, Switzerland 056/20 51 51, U.K. 0635 523545

Limited Warranty

The AT-MIO-16F-5 is warranted against defects in materials and workmanship for a period of one year from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Trademarks

Warning Regarding Medical and Clinical Use of National Instruments Products

National Instruments products are not designed with components and testing intended to ensure a level of reliability suitable for use in treatment and diagnosis of humans. Applications of National Instruments products involving medical or clinical treatment can create a potential for accidental injury caused by product failure, or by errors on the part of the user or application designer. Any use or application of National Instruments products for or involving medical or clinical treatment must be performed by properly trained and qualified medical personnel, and all traditional medical safeguards, equipment, and procedures that are appropriate in the particular situation to prevent serious injury or death should always continue to be used when National Instruments products are being used. National Instruments products are NOT intended to be a substitute for any form of established process, procedure, or equipment used to monitor or safeguard human health and safety in medical or clinical treatment.

Contents

Figures

Tables

About This Manual

This manual describes the mechanical and electrical aspects of the AT-MIO-16F-5 board and contains information concerning its operation and programming. The AT-MIO-16F-5 is a high-performance, multifunction analog, digital, and timing I/O board for the IBM PC AT and compatibles and EISA personal computers (PCs).

Organization of This Manual

The *AT-MIO-16F-5 User Manual* is organized as follows:

- Chapter 1, *Introduction*, describes the AT-MIO-16F-5; lis``ts the contents of your AT-MIO-16F-5 kit, the optional software, and optional equipment; and explains how to unpack the AT-MIO-16F-5.
- Chapter 2, *Configuration and Installation*, describes the AT-MIO-16F-5 jumper configuration, installation of the AT-MIO-16F-5 in the PC, signal connections to the AT-MIO-16F-5, and cable wiring.
- Chapter 3, *Theory of Operation*, contains a functional overview of the AT-MIO-16F-5 and explains the operation of each functional unit making up the AT-MIO-16F-5.
- Chapter 4, *Programming*, describes in detail the address and function of each of the AT-MIO-16F-5 registers. This chapter also includes important information about programming the AT-MIO-16F-5.
- Chapter 5, *Calibration Procedures*, discusses the calibration procedures for the AT-MIO-16F-5 analog input and analog output circuitry.
- Appendix A, *Specifications*, lists the specifications for the AT-MIO-16F-5.
- Appendix B, *I/O Connector*, shows the pinout and signal names for the AT-MIO-16F-5 50-pin I/O connector, including a description of each connection.
- Appendix C, *AMD Data Sheet*, contains the *AMD Am9513A System Timing Controller* (Advanced Micro Devices, Inc.) data sheet. This controller is used on the AT-MIO-16F-5.
- Appendix D, *Customer Communication*, contains forms for you to complete to facilitate communication with National Instruments concerning our products.
- The *Index* alphabetically lists topics covered in this manual, including the page where the topic can be found.

Conventions Used in This Manual

The following conventions are used in this manual:

italic	Italic text denotes emphasis, a cross reference, or an introduction to a key concept.
PC	PC refers to the IBM PC AT and compatibles, and to EISA personal computers.

Related Documentation

The following document contains information that you may find helpful as you read this manual:

- The *IBM Personal Computer AT Technical Reference* manual

You may also want to consult the following Advanced Micro Devices manual if you plan to program the Am9513A Counter/Timer used on the AT-MIO-16F-5:

- The *Am9513A/Am9513 System Timing Controller* technical manual

Customer Communication

We appreciate communicating with the people who use our products. We are also very interested in hearing about the applications you develop using our products. To make it easy for you to communicate with us, this manual contains forms for to you complete. These forms are located in Appendix D, *Customer Communication*, at the back of this manual.

Chapter 1
Introduction

This chapter describes the AT-MIO-16F-5; lists the contents of your AT-MIO-16F-5 kit, the optional software, and optional equipment; and explains how to unpack the AT-MIO-16F-5.

The AT-MIO-16F-5 is a high-performance multifunction analog, digital, and timing I/O board for the PC. The AT-MIO-16F-5 has a 5 µsec, 12-bit, sampling ADC; 16 single-ended or 8 differential channels (expandable with AMUX-64T multiplexer board); programmable gains of 0.5, 1, 2, 5, 10, 20, 50 and 100; a guaranteed maximum rate of at least 200 ksamples/sec; a 256-word A/D FIFO buffer to obtain the highest possible data acquisition rate; and internal or external A/D timing. The AT-MIO-16F-5 also has two double-buffered, multiplying, 12-bit DACs; unipolar and bipolar voltage output; an onboard DAC reference voltage of 10 V; internal timer and external signal update capability for waveform generation; onboard I/O hardware auto calibration circuitry; eight digital I/O lines able to sink up to 24 mA of current; three independent 16-bit counter/timers for frequency counting, event counting, and pulse output applications; timer-generated interrupts; high-performance RTSI bus interface; four triggers for system-level timing; and full PC I/O channel DMA capability with both analog input and analog output. If additional analog inputs are required, you can use the AMUX-64T multiplexer board. This four-to-one multiplexer can process 64 single-ended or 32 differential inputs. Up to four AMUX-64Ts can be cascaded to obtain 256 single-ended inputs.

The AT-MIO-16F-5, with its multifunction analog, digital, and timing I/O, can be used in many applications for automation of machine and process control, level monitoring and control, instrumentation, electronic testing, and various other functions. The multichannel analog input can be used for such functions as signal and transient analysis, data logging, and chromatography. The two analog output channels can be used for such functions as machine and process control, analog function generation, 12-bit resolution voltage source, and programmable signal attenuation. The eight TTL-compatible digital I/O lines can be used for machine and process control, intermachine communication, and relay switching control. The three 16-bit counter/timers can be used for such functions as pulse and clock generation, timed control of laboratory equipment, and frequency, event, and pulse-width measurement. With all of these functions on one board, laboratory processes can be automatically monitored and controlled.

The AT-MIO-16F-5 is interfaced to the National Instruments RTSI bus. With this bus, National Instruments AT Series boards can send timing signals to each other. The AT-MIO-16F-5 can send signals from the onboard counter/timer to another board, or another board can control single and multiple A/D conversions on the AT-MIO-16F-5.

Detailed specifications for the AT-MIO-16F-5 are listed in Appendix A.

AT-MIO-16F-5 Versus AT-MIO-16

As the next step in the National Instruments MIO board line, the AT-MIO-16F-5 incorporates functional improvements and additions to the older AT-MIO-16 boards. Because the AT-MIO-16F-5 is a superset of the AT-MIO-16 line of boards, any function on the AT-MIO-16 is available on the AT-MIO-16F-5. The following is a listing of the additional functions of the AT-MIO-16F-5.

- 200 kHz Throughput
- 5 µsec Settling at all Gains
- 512-Long Channel-Gain Memory
- 256-Deep ADC FIFO
- DMA to Analog Output
- DAC Update Error Signal
- 0.5 LSB rms Analog Output Noise (DC - 500 kHz)
- Full Acquisition Modes Using the EXTCONV* Signal at the I/O Connector
- Higher Effective Resolution (with Dither and Averaging)
- Software Calibratable Analog Input and Analog Output with Onboard Voltage Reference
- More Gains: 0.5, 1, 2, 5, 10, 20, 50, 100
- Software Configurable Analog Input
- Expanded Timebase Resolution (200 nsec)
- Software Pretriggering
- Software and Hardware Acquisition Gating
- Selectable Counter for Waveform Generation
- 4 µsec Analog Output Settling
- Dedicated DAC Update Signal at the I/O Connector
- Multiple Rate Data Acquisition with Channel Scanning
- AI SENSE is not grounded in the differential analog input configuration.

AT-MIO-16F-5 Hardware Changes from the AT-MIO-16

The hardware changes between the AT-MIO-16F-5 and the AT-MIO-16 are listed in the following table.

Table 1-1. AT-MIO-16F-5 Hardware Differences from the AT-MIO-16

Hardware Differences		
	AT-MIO-16	AT-MIO-16F-5
I/O Connector		
Pin 38	STARTTRIG*	EXTTRIG*
Pin 39	STOPTRIG	EXTGATE*
Pin 44	SOURCE2	EXTDACUPDATE*
RTSI Switch		
A4RCV	STOPTRIG	DACUPTRIG*

What Your Kit Should Contain

The contents of the AT-MIO-16F5 kit (part number 776441-01) are listed as follows.

Kit Component	Part Number
AT-MIO-16F5 board	180985-01
AT-MIO-16F-5 User Manual	320266-01
NI-DAQ software for DOS/Windows/LabWindows, with manuals	776250-01
NI-DAQ Software Reference Manual for DOS/Windows/LabWindows	320498-01
NI-DAQ Function Reference Manual for DOS/Windows/LabWindows	320499-01

If your kit is missing any of the components, contact National Instruments.

Your AT-MIO-16F5 is shipped with the NI-DAQ software for DOS/Windows/LabWindows. NI-DAQ has a library of functions that can be called from your application programming environment. These functions include routines for analog input (A/D conversion), buffered data acquisition (high-speed A/D conversion), analog output (D/A conversion), waveform generation, digital I/O, counter/timer, SCXI, RTSI, and self-calibration. NI-DAQ maintains a consistent software interface among its different versions so you can switch between platforms with minimal modifications to your code. NI-DAQ comes with language interfaces for Professional BASIC, Turbo Pascal, Turbo C, Turbo C++, Borland C++, and Microsoft C for DOS; and Visual Basic, Turbo Pascal, Microsoft C with SDK, and Borland C++ for Windows. NI-DAQ software is on high-density 5.25 in. and 3.5 in. diskettes.

Optional Software

This manual contains complete instructions for directly programming the AT-MIO-16F-5. Normally, however, you should not need to read the low-level programming details in the user manual because the NI-DAQ software package for controlling the AT-MIO-16F-5 is included with the board. Using NI-DAQ is quicker and easier than and as flexible as using the low-level programming described in Chapter 4, *Programming*.

You can use the AT-MIO-16F-5 with LabVIEW for Windows or LabWindows for DOS. LabVIEW and LabWindows are innovative program development software packages for data acquisition and control applications. LabVIEW uses graphical programming, whereas LabWindows enhances Microsoft C and QuickBASIC. Both packages include extensive libraries for data acquisition, instrument control, data analysis, and graphical data presentation.

Part numbers for these software packages are listed in the following table.

Software	Part Number
LabVIEW for Windows	776670-01
LabWindows	
Standard package	776473-01
Advanced Analysis Library	776474-01
Standard package with the Advanced Analysis Library	776475-01

Optional Equipment

Equipment	Part Number
CB-50 I/O connector block (50 screw terminals)	
with 0.5-m type NB1 cable	776164-01
with 1.0-m type NB1 cable	776164-02
AT Series RTSI bus cables for	
2 boards	776249-02
3 boards	776249-03
4 boards	776249-04
5 boards	776249-05
AMUX-64T analog multiplexer board	
with 0.2-m ribbon cable	776366-02
with 0.5-m ribbon cable	776366-05
with 1.0-m ribbon cable	776366-10
with 2.0-m ribbon cable	776366-20

The AT-MIO-16F-5 I/O connector is a 50-pin, male, ribbon cable header. The manufacturer part numbers for this header are as follows:

- Electronic Products Division/3M (part number 3596-5002)
- T&B/Ansley Corporation (part number 609-5007)

The mating connector for the AT-MIO-16F-5 is a 50-position, ribbon socket connector, polarized, with strain relief. National Instruments uses a polarized (keyed) connector to prevent inadvertent misconnection to the AT-MIO-16F-5. Recommended manufacturer part numbers for this mating connector are as follows:

- Electronic Products Division/3M (part number 3425-7650)
- T&B/Ansley Corporation (part number 609-5041CE)

Recommended manufacturer part numbers for the standard ribbon cable (50-conductor, 28 AWG, stranded) that can be used with these connectors are:

- Electronic Products Division/3M (part number 3365/50)
- T&B/Ansley Corporation (part number 171-50)

Unpacking

Your AT-MIO-16F-5 board is shipped in an antistatic plastic package to prevent electrostatic damage to the board. Several components on the board can be damaged by electrostatic discharge. To avoid such damage in handling the board, take the following precautions:

- Touch the plastic package to a metal part of your PC chassis before removing the board from the package.
- Remove the board from the package and inspect the board for loose components or any other sign of damage. Notify National Instruments if the board appears damaged in any way. *Do not* install a damaged board into your computer.

Chapter 2
Configuration and Installation

This chapter describes the AT-MIO-16F-5 jumper configuration, installation of the AT-MIO-16F-5 board in the PC, signal connections to the AT-MIO-16F-5 board, and cable wiring.

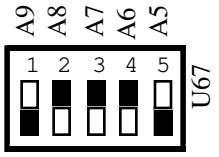
Board Configuration

The AT-MIO-16F-5 contains 13 jumpers and one DIP switch to configure the AT bus interface and analog output settings. The DIP switch is used to set the base I/O address. Four jumpers are used as interrupt and DMA channel selectors. Six of the remaining nine jumpers are used to configure the analog output circuitry. The jumpers are shown in the parts locator diagram in Figure 2-1. Jumpers W8 through W13 configure the analog output circuitry. Jumpers W1 and W2 select the clock signal used by both the Am9513A Counter/Timer and the clock pin on the RTSI bus. Jumpers W4, W5, W6, and W7 select the DMA channel and the interrupt level. Jumper W3 is used for initial calibration functions and should not be changed, so it is removed from the board before shipping.

AT Bus Interface

The AT-MIO-16F-5 is configured at the factory to a base I/O address of 220 hex, to use DMA Channel 6 and Channel 7, and to use interrupt level 10. These settings (shown in Table 2-1) are suitable for most systems. However, if your system has other hardware at this base I/O address, DMA channel, or interrupt level, you need to change these factory settings on the AT-MIO-16F-5 (as described in the following pages) or on the other hardware.

Table 2-1. AT Bus Interface Factory Settings

Base I/O Address	Hex 220	 (The shaded portion indicates the side of the base address switch that is pressed down.)
DMA Channel	DMA A = DMA Channel 6 DMA B = DMA Channel 7 (factory setting)	**W4:** R6: A-B A6: A-B **W4:** R7: B-C A7: B-C **W7:** no jumpers
Interrupt Level	Interrupt level 10 selected (factory setting)	**W5:** Row 10 **W6:** no jumpers

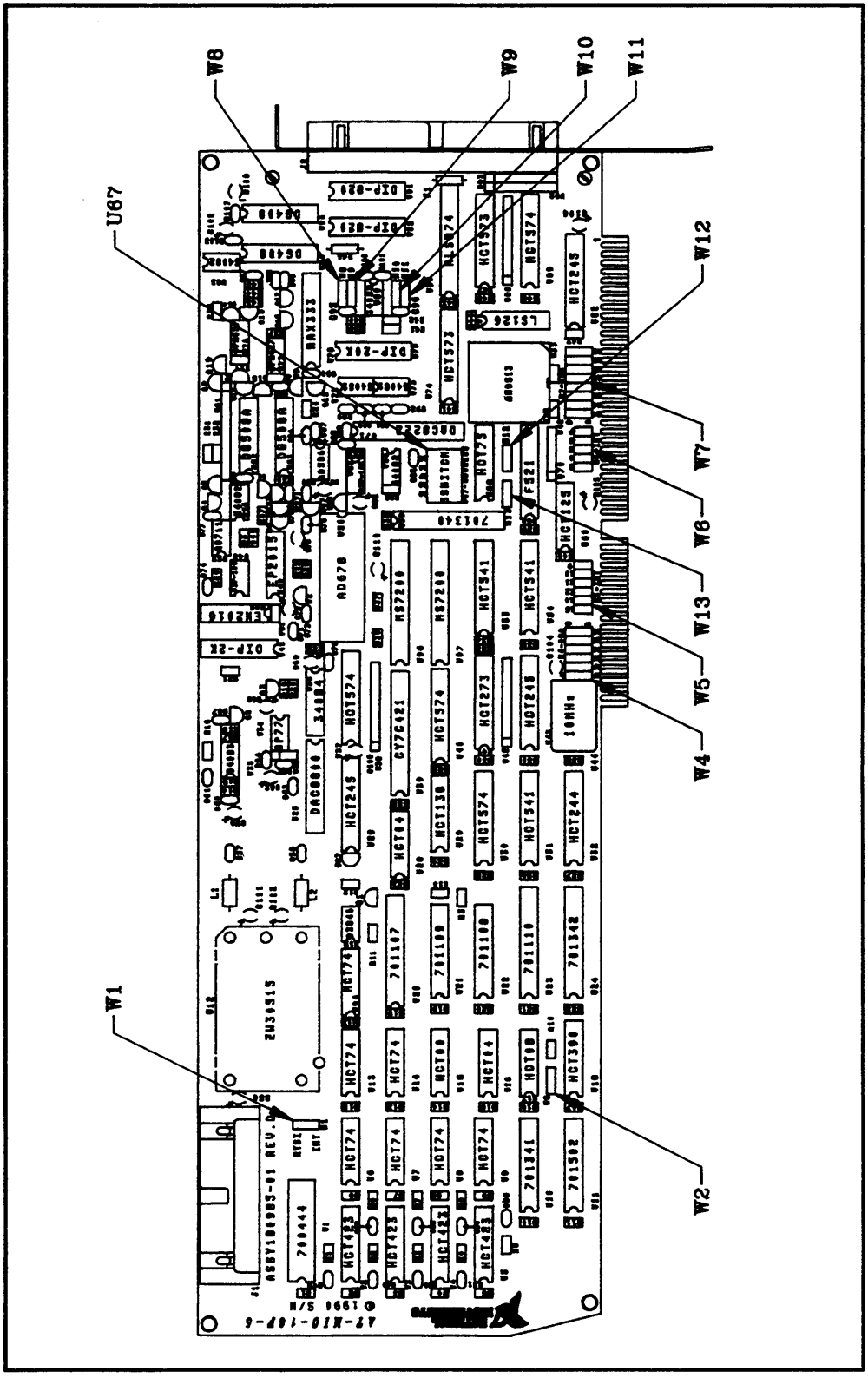


Figure 2-1. AT-MIO-16F-5 Parts Locator Diagram

Base I/O Address Selection

The base I/O address for the AT-MIO-16F-5 is determined by the switches at position U67 (see Figure 2-1). The switches are set at the factory for the base I/O address 220 hex. This factory setting is used by National Instruments software packages as the default base I/O address value for the AT-MIO-16F-5. The AT-MIO-16F-5 uses the base I/O address space 220 hex through 23F hex with the factory setting.

Note: Verify that this space is not already used by other equipment installed in your computer. If any equipment in your computer uses this base I/O address space, change the base I/O address of the AT-MIO-16F-5 or of the other device. If you change the AT-MIO-16F-5 base I/O address, make a corresponding change to any software packages you use with the AT-MIO-16F-5. Table 2-2 lists the default settings of other National Instruments products for the PC. For more information about the I/O address of your PC, refer to the technical reference manual for your computer.

Each switch in U67 corresponds to one of the address lines A9 through A5. Press the side marked OFF to select a binary value of 1 for the corresponding address bit . Press the ON side of the switch to select a binary value of 0 for the corresponding address bit. Figure 2-2 shows two possible switch settings. The shaded portion indicates the side of the switch that is pressed down.

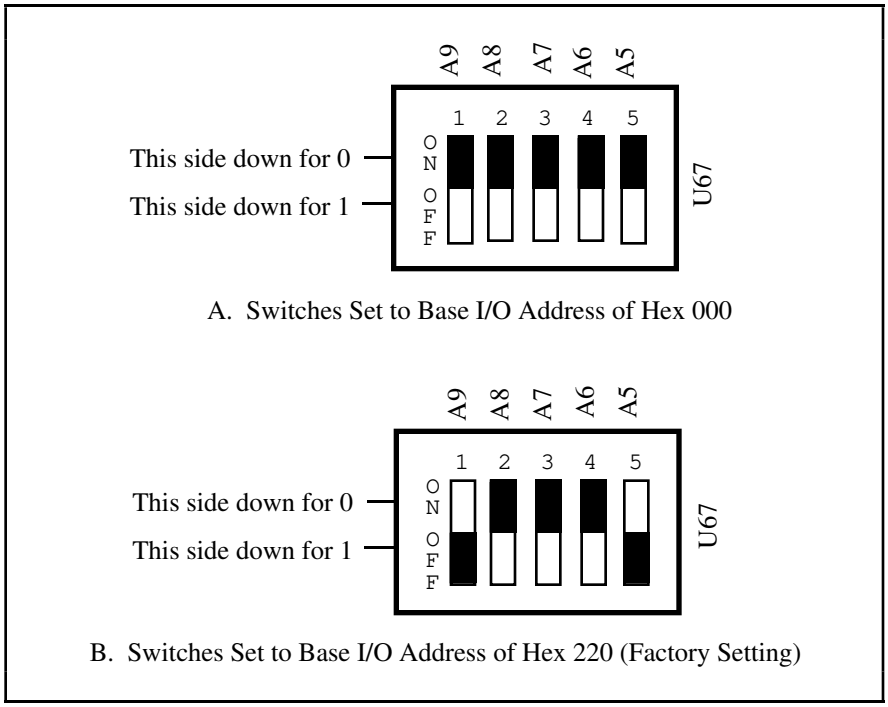


Figure 2-2. Example Base I/O Address Switch Settings

The least significant five bits of the address (A4 through A0) are decoded by the AT-MIO-16F-5 to select the appropriate AT-MIO-16F-5 register. To change the base I/O address, remove the plastic cover on U67; press each switch to the desired position; verify that each switch is pressed down all the way; and replace the plastic cover. Make a note of the new AT-MIO-16F-5 base I/O address for use when configuring the AT-MIO-16F-5 software (a form is included for you in Appendix D).

Table 2-3 lists the possible switch settings, the corresponding base I/O address, and the base I/O address space used for that setting.

Table 2-2. Default Settings of National Instruments Products for the PC

Board	DMA Channel	Interrupt Level	Base I/O Address
AT-A2150	None*	None*	120 hex
AT-AO-6/10	Channel 5	Lines 11, 12	1C0 hex
AT-DIO-32F	Channels 5, 6	Lines 11, 12	240 hex
AT-DSP2200	None*	None*	120 hex
AT-GPIB	Channel 5	Line 11	2C0 hex
AT-MIO-16	Channels 6, 7	Line 10	220 hex
AT-MIO-16D	Channels 6, 7	Lines 5, 10	220 hex
AT-MIO-16F-5	Channels 6, 7	Line 10	220 hex
AT-MIO-16X	None*	None*	220 hex
AT-MIO-64F-5	None*	None*	220 hex
GPIB-PCII	Channel 1	Line 7	2B8 hex
GPIB-PCIIA	Channel 1	Line 7	02E1 hex
GPIB-PCIII	Channel 1	Line 7	280 hex
Lab-PC	Channel 3	Line 5	260 hex
PC-DIO-24	None	Line 5	210 hex
PC-DIO-96	None	Line 5	180 hex
PC-LPM-16	None	Line 5	260 hex
PC-TIO-10	None	Line 5	1A0 hex

* These settings are software configurable and are disabled at startup time.

Table 2-3. Switch Settings with Corresponding Base I/O Address and Base I/O Address Space

Switch Setting A9 A8 A7 A6 A5					Base I/O Address (hex)	Base I/O Address Space Used (hex)
0	0	0	0	0	000	000 - 01F
0	0	0	0	1	020	020 - 03F
0	0	0	1	0	040	040 - 05F
0	0	0	1	1	060	060 - 07F
0	0	1	0	0	080	080 - 09F
0	0	1	0	1	0A0	0A0 - 0BF
0	0	1	1	0	0C0	0C0 - 0DF
0	0	1	1	1	0E0	0E0 - 0FF
0	1	0	0	0	100	100 - 11F
0	1	0	0	1	120	120 - 13F
0	1	0	1	0	140	140 - 15F
0	1	0	1	1	160	160 - 17F
0	1	1	0	0	180	180 - 19F
0	1	1	0	1	1A0	1A0 - 1BF
0	1	1	1	0	1C0	1C0 - 1DF
0	1	1	1	1	1E0	1E0 - 1FF
1	0	0	0	0	200	200 - 21F
1	0	0	0	1	220	220 - 23F
1	0	0	1	0	240	240 - 25F
1	0	0	1	1	260	260 - 27F
1	0	1	0	0	280	280 - 29F
1	0	1	0	1	2A0	2A0 - 2BF
1	0	1	1	0	2C0	2C0 - 2DF
1	0	1	1	1	2E0	2E0 - 2FF
1	1	0	0	0	300	300 - 31F
1	1	0	0	1	320	320 - 33F
1	1	0	1	0	340	340 - 35F
1	1	0	1	1	360	360 - 37F
1	1	1	0	0	380	380 - 39F
1	1	1	0	1	3A0	3A0 - 3BF
1	1	1	1	0	3C0	3C0 - 3DF
1	1	1	1	1	3E0	3E0 - 3FF

Note: Base I/O address values 000 through 0FF hex are reserved for system use. Base I/O address values 100 through 3FF hex are available on the I/O channel.

DMA Channel Selection

The DMA channels used by the AT-MIO-16F-5 are selected by jumpers on W4 and W7 (see Figure 2-1). The AT-MIO-16F-5 is set at the factory to use DMA Channel 6 and Channel 7 for dual DMA mode. These are the default DMA channels used by the AT-MIO-16F-5 software handler. Verify that these DMA channels are not also used by equipment already installed in your computer. If any device uses DMA Channel 6 and/or Channel 7, change the DMA channel used by either the AT-MIO-16F-5 or the other device. (Unless the appropriate DMA modes have been enabled on the AT-MIO-16F-5 through software, the DMA channels are by default in the high-impedance state at startup.) The DMA channels supported by the AT-MIO-16F-5 hardware are Channel 0 through Channel 3 and Channel 5 through Channel 7. If the AT-MIO-16F-5 is used in an AT-type computer, only DMA Channels 5 through 7 should be used since these are the only 16-bit channels. If the board is used in an EISA computer, all of the channels are 16-bit and can be used. The AT-MIO-16F-5 *does not* use and *cannot* be configured to use the 8-bit DMA channels on the PC I/O channel.

Each DMA channel consists of two signal lines as shown in Table 2-4.

Table 2-4. DMA Channels for the AT-MIO-16F-5

DMA Channel	DMA Acknowledge	DMA Request
0	DACK0 (A0)	DRQ0 (R0)
1	DACK1 (A1)	DRQ1 (R1)
2	DACK2 (A2)	DRQ2 (R2)
3	DACK3 (A3)	DRQ3 (R3)
5	DACK5 (A5)	DRQ5 (R5)
6	DACK6 (A6)	DRQ6 (R6)
7	DACK7 (A7)	DRQ7 (R7)

Two jumpers must be installed to select a single DMA channel. The DMA acknowledge and DMA request lines selected must have the same number suffix for proper operation. When you use dual DMA mode, the lower rows of W6 are used for DMA A and the upper two rows of W6 are used for DMA B. Figure 2-3 displays the jumper positions for selecting DMA Channel 6 and Channel 7. In this setting, DMA A uses DMA Channel 6, and DMA B uses DMA Channel 7.

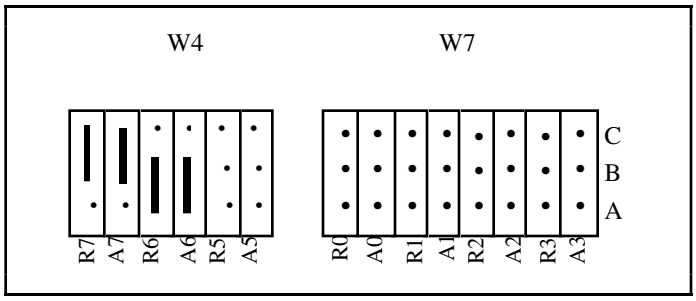


Figure 2-3. DMA Jumper Settings for DMA Channels 6 and 7 (Factory Setting)

If you want to use only one DMA channel, programming the DMA mode in Command Register 2 places DMA B in the high-impedance state.

If you do not want to use DMA for AT-MIO-16F-5 transfers, disabling DMAEN in Command Register 2 places DMA Channel A and DMA Channel B in the high-impedance state.

Interrupt Selection

The AT-MIO-16F-5 board can connect to any one of the eleven interrupt lines of the PC I/O channel. The interrupt line is selected by a jumper on one of the double rows of pins located above the I/O slot edge connector on the AT-MIO-16F-5 (refer to Figure 2-1). To use the interrupt capability of the AT-MIO-16F-5, select an interrupt line and place the jumper in the appropriate position to enable that particular interrupt line.

The AT-MIO-16F-5 can share interrupt lines with other devices. (Unless the appropriate interrupt modes have been enabled on the AT-MIO-16F-5 through software, the interrupt line is by default in the high impedance state at startup.) The interrupt lines supported by the AT-MIO-16F-5 hardware are IRQ3, IRQ4, IRQ5, IRQ6, IRQ7, IRQ9, IRQ10, IRQ11, IRQ12, IRQ14, and IRQ15.

Note: *Do not* use interrupt line 6 or 14. Interrupt line 6 is used by the diskette drive controller, and interrupt line 14 is used by the hard disk controller on most PCs.

Once you have selected an interrupt level, place the interrupt jumper on the appropriate pins to enable the interrupt line.

The interrupt jumper set is W5 and W6. The default interrupt line is IRQ10, which is selected by placing the jumper on the pins in row 10. Figure 2-4 shows the default interrupt jumper setting IRQ10. To change to another line, remove the jumper from IRQ10 and place it on the new pins.

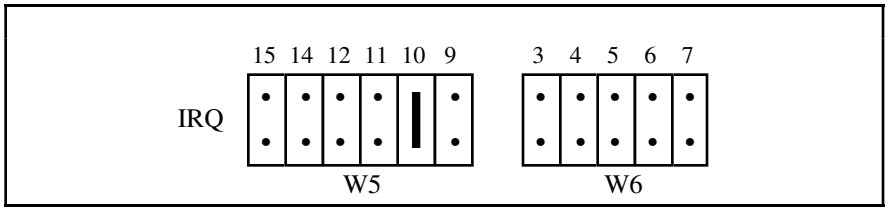


Figure 2-4. Interrupt Jumper Setting IRQ10 (Factory Setting)

If you do not want to use interrupts for AT-MIO-16F-5 transfers, then disabling INTEN, DMATCINTEN, and CMPLINTEN in Command Register 2 places the selected interrupt in the high impedance state.

Analog I/O Jumper Settings

The AT-MIO-16F-5 is shipped from the factory with the following configuration:

- ±10 V analog output range with internal reference selected
- Two's complement analog output coding
- AT-MIO-16F-5 clock signal set to 10 MHz

Table 2-5 lists all the available analog I/O jumper configurations for the AT-MIO-16F-5 and notes the factory settings. Other analog I/O configurations are selected through software and are detailed in the following pages.

Table 2-5. Analog I/O Jumper Settings

	Configuration	Jumper/Switch Settings
Am9513A and RTSI Bus Clock Select	AT-MIO-16F-5 clock signal = 10 MHz (factory setting) AT-MIO-16F-5 clock signal = RTSI clock signal AT-MIO-16F-5 and RTSI clock signals both = 10 MHz	**W1:** B-C **W2:** B-C **W1:** A-B **W2:** A-B **W1:** A-B **W2**: B-C
DAC0 Reference	Internal (factory setting) External	**W11:** A-B **W11:** B-C
DAC1 Reference	Internal (factory setting) External	**W8:** A-B **W8:** B-C
DAC0 Output Polarity - Digital Format	Unipolar - straight binary Bipolar - two's complement (factory setting)	**W10:** A-B **W13:** B-C **W10:** B-C **W13:** A-B
DAC1 Output B-C **Digital Format**	Unipolar - straight binary **Polarity - W12:** (factory setting)	**W9:** A-B **W12:** B-C Bipolar - two's complement **W9:** A-B

Analog Input Configuration

The analog input section of the AT-MIO-16F-5 is fully software-configurable. You can select different analog input configurations by setting the appropriate bits in the command registers as described in Chapter 4, *Programming*. The following paragraphs describe each of the analog input categories in detail.

Input Mode

The AT-MIO-16F-5 offers three different input modes: non-referenced single-ended (NRSE) input, referenced single-ended (RSE) input, and differential (DIFF) input. The single-ended input configurations use 16 channels. The DIFF input configuration uses 8 channels. These configurations are described in Table 2-6.

Table 2-6. Input Configurations Available for the AT-MIO-16F-5

Configuration	Description
DIFF	Differential configuration. Provides 8 differential inputs with the negative (-) input of the instrumentation amplifier tied to the multiplexer output of Channels 8 through 15.
RSE	Referenced Single-Ended configuration. Provides 16 single-ended inputs with the negative (-) input of the instrumentation amplifier referenced to analog ground.
NRSE	Non-Referenced Single-Ended configuration. Provides 16 single-ended inputs with the negative (-) input of the instrumentation amplifier tied to AISENSE and *not* connected to ground.

While reading the following paragraphs, you may find it helpful to refer to *Analog Input Signal Connections* later in this chapter, which contains diagrams showing the signal paths for the three configurations.

DIFF Input (8 Channels)

DIFF input means that each input signal has its own reference, and the difference between each signal and its reference is measured. The signal and its reference are assigned an input channel. With this input configuration, the AT-MIO-16F-5 can monitor eight different analog input signals. This configuration is selected via software. (See Command Register 1 and Table 4-2 in Chapter 4.) The results of this configuration are as follows:

- Channels 0 through 7 are tied to the positive (+) input of the instrumentation amplifier.
- Channels 8 through 15 are tied to the negative (-) input of the instrumentation amplifier.

- Multiplexer control is configured to control eight input channels.
- AI SENSE is left unconnected.

Considerations in using the DIFF input configuration are discussed under *Signal Connections* later in this chapter. Figure 2-16 shows a schematic diagram of this configuration.

RSE Input (16 Channels)

RSE input means that all input signals are referenced to a common ground point that is also tied to the analog input ground of the AT-MIO-16F-5 board. The negative (-) input of the differential input amplifier is tied to the analog ground. This configuration is useful when measuring floating signal sources. See *Types of Signal Sources* later in this chapter for more information. With this input configuration, the AT-MIO-16F-5 can monitor 16 different analog input signals. This configuration is selected via software. (See Command Register 1 and Table 4-2 in Chapter 4.) The results of this configuration are as follows:

- The negative (-) input of the instrumentation amplifier is tied to the instrumentation amplifier signal ground.
- Multiplexer outputs are tied together into the positive (+) input of the instrumentation amplifier.
- Multiplexer control is configured to control 16 input channels.
- AI SENSE is left unconnected.

Considerations in using the RSE configuration are discussed under *Signal Connections* later in this chapter. Figure 2-18 shows a schematic diagram of this configuration.

NRSE Input (16 Channels)

NRSE input means that all input signals are referenced to the same common mode voltage, but this common mode voltage can float with respect to the analog ground of the AT-MIO-16F-5 board. This common mode voltage is subsequently subtracted out by the input instrumentation amplifier. This configuration is useful when measuring ground-referenced signal sources. See *Types of Signal Sources* later in this chapter for more information. With this input configuration, the AT-MIO-16F-5 can measure 16 different analog input signals. This configuration is selected via software. (See Command Register 1 and Table 4-2 in Chapter 4.) The results of this configuration are as follows:

- AI SENSE is tied into the negative (-) input of the instrumentation amplifier.
- Multiplexer outputs are tied together into the positive (+) input of the instrumentation amplifier.
- Multiplexer control is configured to control 16 input channels.

Considerations in using the NRSE configuration are discussed under *Signal Connections* later in this chapter. Figure 2-19 shows a schematic diagram of this configuration.

Input Polarity and Input Range

The AT-MIO-16F-5 has two polarities: unipolar input and bipolar input. Unipolar input means that the input voltage range is between 0 and V_{ref} where V_{ref} is some positive reference voltage. Bipolar input means that the input voltage range is between $-V_{ref}$ and $+V_{ref}$. The AT-MIO-16F-5 has one input range of 10 V. An input range of 20 V is achieved by using the gain of 0.5. Polarity and range settings are selected by writing to registers on the AT-MIO-16F-5.

Considerations for Selecting Input Ranges

Input polarity/range selection depends on the expected input range of the incoming signal. A large input range can accommodate a large signal variation but sacrifices voltage resolution. Choosing a smaller input range increases voltage resolution but can result in the input signal going out of range. For best results, the input range should be matched as closely as possible to the expected range of the input signal. For example, if the input signal is guaranteed never to swing below 0 V, a unipolar input is best. In this configuration, however, if the signal does swing negative, inaccurate readings occur.

Software-programmable gain on the AT-MIO-16F-5 increases overall flexibility by matching input signal ranges to those accommodated by the AT-MIO-16F-5 ADC. The AT-MIO-16F-5 board has gains of 0.5, 1, 2, 5, 10, 20, 50 and 100 and is well suited to a wide variety of signal levels. With the proper gain setting, the full resolution of the ADC can be used to measure the input signal. Table 2-7 shows the overall input range and precision according to the input range configuration and gain used.

Table 2-7. Actual Range and Measurement Precision Versus Input Range Selection and Gain

Range Configuration	Gain	Actual Input Range	Precision*
0 to +10 V	0.5	0 to +20.0 V†	4.88 mV
	1.0	0 to +10.0 V	2.44 mV
	2.0	0 to +5.0 V	1.22 mV
	5.0	0 to +2.0 V	488.00 µV
	10.0	0 to +1.0 V	244.00 µV
	20.0	0 to +0.5 V	122.00 µV
	50.0	0 to +0.2 V	48.80 µV
	100.0	0 to 100.0 mV	24.40 µV

(continues)

Table 2-7. Actual Range and Measurement Precision Versus Input Range Selection and Gain (continued)

Range Configuration	Gain	Actual Input Range	Precision*
-5 to +5 V	0.5	-10.00 to +10.00 V	4.88 mV
	1.0	-5.00 to +5.00 V	2.44 mV
	2.0	-2.50 to +2.50 V	1.22 mV
	5.0	-1.00 to +1.00 V	488.00 µV
	10.0	-0.50 to +0.50 V	244.00 µV
	20.0	-0.25 to +0.25 V	122.00 µV
	50.0	-100.00 to 100.00 mV	48.80 µV
	100.0	-50.00 to +50.00 mV	24.40 µV

* The value of 1 LSB of the 12-bit ADC, that is, the voltage increment corresponding to a change of 1 count in the ADC 12-bit count.
† 0 to +20 V is the effective range. Signals greater than +12 V with respect to the AT-MIO-16F-5 AGND will saturate internal components and result in inaccurate data.
Note: See specifications in Appendix A for absolute maximum ratings.

Analog Output Configuration

You can select different analog output configurations by using the jumper settings shown in Table 2-5. The following paragraphs describe each of the analog output configurations in detail.

Internal and External Reference

Each DAC can be connected to the AT-MIO-16F-5 internal reference of 10 V or to the external reference signal connected to the EXTREF pin on the I/O connector. This signal applied to EXTREF must be between -10 V and +10 V. Both channels need not be configured the same way.

External Reference Selection

You select the external reference signal for each analog output channel by setting the following jumpers:

Analog Output Channel 0:	W11	B - C	External reference signal connected to DAC 0 reference input.
Analog Output Channel 1:	W8	B - C	External reference signal connected to DAC 1 reference input.

This configuration is shown in Figure 2-5.

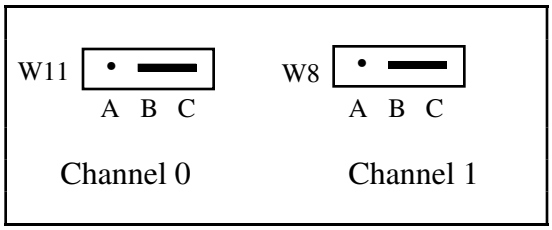


Figure 2-5. External Reference Configuration

Internal Reference Selection (Factory Setting)

You set the onboard 10 V reference for each analog output channel by setting the following jumpers:

Analog Output Channel 0:	W11	A - B	10 V onboard reference connected to DAC 0 reference input.
Analog Output Channel 1:	W8	A - B	10 V onboard reference connected to DAC 1 reference input.

This configuration is shown in Figure 2-6.

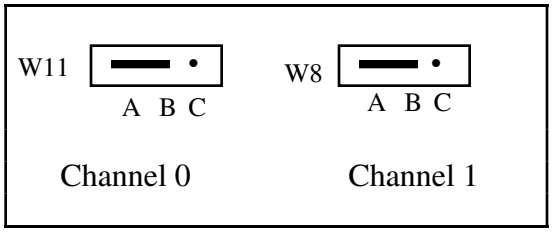


Figure 2-6. Internal Reference Configuration (Factory Setting)

Analog Output Polarity Selection

Each analog output channel can be configured for either unipolar or bipolar output. A unipolar configuration has a range of 0 to V_{ref} at the analog output. A bipolar configuration has a range of $-V_{ref}$ to $+V_{ref}$ at the analog output. V_{ref} is the voltage reference used by the DACs in the analog output circuitry and can either be the 10 V onboard reference or an externally supplied reference between -10 V and +10 V. Both channels need not be configured the same way; however, at the factory, both channels are configured for bipolar output.

Bipolar Output Selection (Factory Setting)

You select the bipolar output configuration for each analog output channel by setting the following jumpers:

Analog Output Channel 0: W10 B - C

Analog Output Channel 1: W9 B - C

This configuration is shown in Figure 2-7.

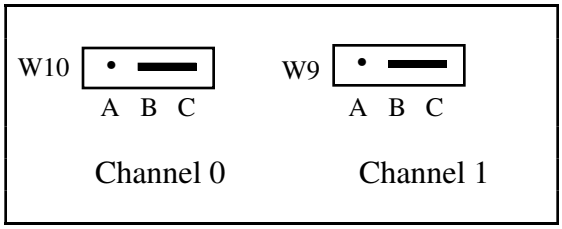


Figure 2-7. Bipolar Output Configuration (Factory Setting)

When you use the bipolar configuration, you must select whether to write straight binary or two's complement to the DAC. In straight binary mode, data values written to the analog output channel range from 0 to 4095 decimal (0 to 0FFF hex). In two's complement mode, data values written to the the analog output channel range from -2048 to +2047 decimal (F800 to 07FF hex).

Straight Binary Mode

The data value written to each analog output channel is interpreted as a straight binary number when the following jumpers are set:

Analog Output Straight Binary for Channel 0: W13 B - C

Analog Output Straight Binary for Channel 1: W12 B - C

This configuration is shown in Figure 2-8.

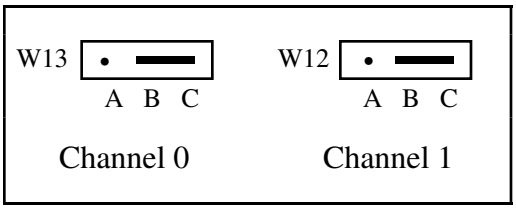


Figure 2-8. Straight Binary Mode

Two's Complement Mode (Factory Setting)

The data value written to each analog output channel is interpreted as a two's complement number when the following jumpers are set.

Analog Output Two's Complement for Channel 0: W13 A - B

Analog Output Two's Complement for Channel 1: W12 A - B

This configuration is shown in Figure 2-9.

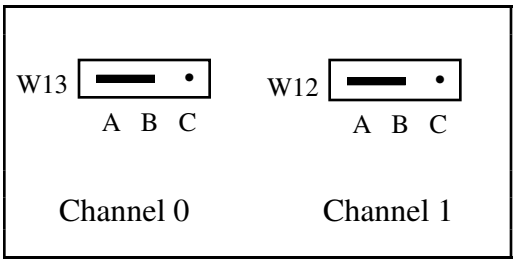


Figure 2-9. Two's Complement Mode (Factory Setting)

Unipolar Output Selection

You select the unipolar output configuration for each analog output channel by setting the following jumpers:

Analog Output Channel 0: W10 A - B

Analog Output Channel 1: W9 A - B

Notice that the straight binary format should be used when in unipolar output mode.

This configuration is shown in Figure 2-10.

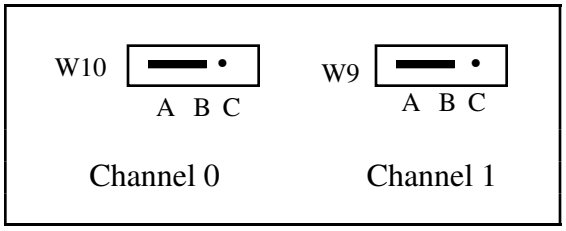


Figure 2-10. Unipolar Output Configuration

RTSI Bus Clock Selection

When multiple AT Series boards are connected via the RTSI bus, you may want to have all the boards use the same 10-MHz clock. This arrangement is useful for applications that require counter/timer synchronization between boards. Each AT Series board with a RTSI bus interface has an onboard 10-MHz oscillator. Thus, one board can drive the RTSI bus clock signal, and the other boards can receive this signal or disconnect from it.

The configuration for jumpers W1 and W2 control whether a board drives the onboard 10-MHz oscillator onto the RTSI bus, receives the RTSI bus clock, or disconnects from the RTSI bus clock. This clock source, whether local or RTSI signal, is then divided by 10 and used as the Am9513A frequency source.

The jumper selections are as follows:

Table 2-8. Configurations for RTSI Bus Clock Selection

Configuration	W1/W2
Disconnect board from RTSI bus clock; use local oscillator	W1: B-C (factory setting) W2: B-C
Receive RTSI bus clock signal	W1: A-B W2: A-B
Drive RTSI bus clock signal with local oscillator	W1: A-B W2: B-C

Figures 2-11, 2-12, and 2-13 show the jumper positions for each of the configurations previously described.

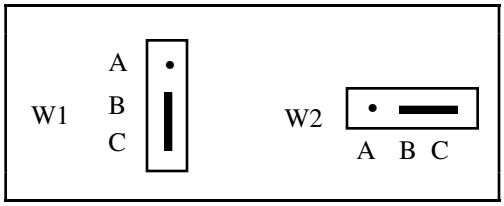


Figure 2-11. Disconnected from RTSI Bus Clock; Use Onboard Oscillator (Factory Setting)

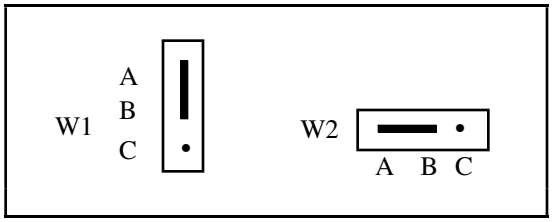


Figure 2-12. Receives RTSI Clock Signal

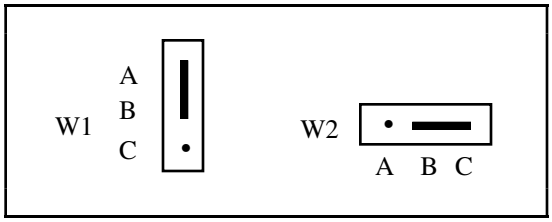


Figure 2-13. Drives RTSI Bus Clock Signal with Onboard Oscillator

Hardware Installation

The AT-MIO-16F-5 can be installed in any available 16-bit expansion slot (AT Series) in your computer. The AT-MIO-16F-5 *does not* work if installed in an 8-bit expansion slot (PC Series). After you have made any necessary changes, verified, and recorded the switches and jumper settings (a form is included in Appendix D), you are ready to install the AT-MIO-16F-5. The following are general installation instructions, but consult the user manual or technical reference manual of your PC for specific instructions and warnings.

1. Turn off your computer.
2. Remove the top cover or access port to the I/O channel.
3. Remove the expansion slot cover on the back panel of the computer.
4. Insert the AT-MIO-16F-5 into a 16-bit slot. Do not force the board into place.
5. Screw the mounting bracket of the AT-MIO-16F-5 to the back panel rail of the computer.
6. Check the installation.
7. Replace the cover.

The AT-MIO-16F-5 board is installed and ready for operation.

Signal Connections

This section describes input and output signal connections to the AT-MIO-16F-5 board via the AT-MIO-16F-5 I/O connector, and includes specifications and connection instructions for the signals given on the AT-MIO-16F-5 I/O connector.

Warning: Connections that exceed any of the maximum ratings of input or output signals on the AT-MIO-16F-5 can result in damage to the AT-MIO-16F-5 board and to the PC. Maximum input ratings for each signal are given in this chapter under the discussion of that signal. National Instruments is not liable for any damages resulting from such signal connections.

Figure 2-14 shows the pin assignments for the AT-MIO-16F-5 I/O connector.

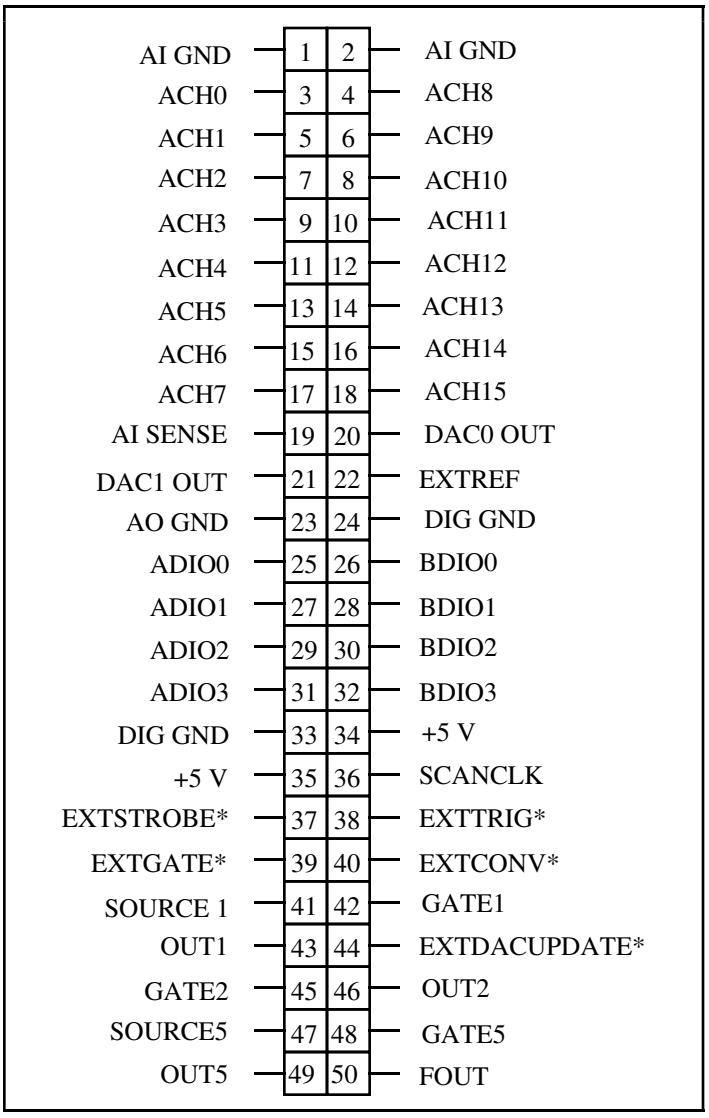


Figure 2-14. AT-MIO-16F-5 I/O Connector

Signal Connection Descriptions

Pins	Signal Names	Reference	Descriptions
1, 2	AIGND	N/A	Analog Input Ground – These pins are the reference point for single-ended measurements and the bias current return point for differential measurements.

(continues)

Pins	Signal Names	Reference	Descriptions (continued)
3 to 18	ACH <0..15>	AI GND	Analog Input Channels 0 through 15 – In differential mode, the input is configured for 8 channels. In single-ended mode, the input is configured for 16 channels.
19	AI SENSE	AI GND	Analog Input Sense – This pin serves as the reference node when the board is in NRSE configuration.
20	DAC0 OUT	AO GND	Analog Channel 0 Output – This pin supplies the voltage output of analog output channel 0.
21	DAC1 OUT	AO GND	Analog Channel 1 Output – This pin supplies the voltage output of analog output channel 1.
22	EXTREF	AO GND	External Reference – This is the external reference input for the analog output circuitry.
23	AOGND	N/A	Analog Output Ground – The analog output voltages are referenced to this node.
24,33	DIG GND	N/A	Digital Ground – This pin supplies the reference for the digital signals at the I/O connector as well as the +5 VDC supply.
25, 27, 29, 31	ADIO <0..3>	DIG GND	Digital I/O port A signals.
26, 28, 30, 32	BDIO <0..3>	DIG GND	Digital I/O port B signals.
34,35	+5 V	DIG GND	+5 VDC Source – This pin is fused for up to 1 A of +5 V supply.
36	SCAN CLK	DIG GND	Scan Clock – This pin pulses once for each A/D conversion. The low-to-high edge indicates when the input signal can be removed from the input or switched to another signal.
37	EXTSTROBE*	DIG GND	External Strobe – Writing to the EXTSTROBE Register results in a 200-500 nsec low pulse on this pin.

(continues)

Pins	Signal Names	Reference	Descriptions (continued)
38	EXTTRIG*	DIG GND	External Trigger – In posttrigger data acquisition sequences, a high-to-low edge on EXTTRIG* initiates the sequence. In pretrigger applications, the first high-to-low edge of EXTTRIG* initiates conversions while the second high-to-low edge initiates the sequence.
39	EXTGATE*	DIG GND	External Gate – When EXTGATE* is low, A/D conversions are inhibited. When EXTGATE* is high, A/D conversions are enabled.
40	EXTCONV*	DIG GND	External Convert – A high-to-low edge on EXTCONV* causes an A/D conversion to occur. If EXTGATE* or EXTCONV* is low, or INTGATE in Command Register 1 is high, conversions are inhibited.
41	SOURCE1	DIG GND	SOURCE1 – This pin is from the Am9513A Counter 1 signal.
42	GATE1	DIG GND	GATE1 – This pin is from the Am9513A Counter 1 signal.
43	OUT1	DIG GND	OUTPUT1 – This pin is from the Am9513A Counter 1 signal.
44	EXTDACUPDATE*	DIG GND	External DAC Update – If selected, a high-to-low edge on EXTDACUPDATE* results in the ouput DACS being updated with the value written to them.
45	GATE2	DIG GND	GATE2 – This pin is from the Am9513A Counter 2 signal.
46	OUT2	DIG GND	OUTPUT2 – This pin is from the Am9513A Counter 2 signal.
47	SOURCE5	DIG GND	SOURCE5 – This pin is from the Am9513A Counter 5 signal.
48	GATE5	DIG GND	GATE5 – This pin is from the Am9513A Counter 5 signal.

(continues)

Pins	Signal Names	Reference	Descriptions (continued)
49	OUT5	DIG GND	OUT5 – This pin is from the Am9513A Counter 5 signal.
50	FOUT	DIG GND	Frequency Output – This pin is from the Am9513A FOUT signal.

The signals on the connector can be classified as analog input signals, analog output signals, digital I/O signals, digital power connections, or timing I/O signals. Signal connection guidelines for each of these groups are given in the following section.

Analog Input Signal Connections

Pins 1 through 19 of the I/O connector are analog input signal pins. Pins 1 and 2 are AI GND signal pins. AI GND is an analog input common signal that is routed directly to the ground tie point on the AT-MIO-16F-5. These pins can be used for a general analog power ground tie point to the AT-MIO-16F-5 if necessary. Pin 19 is the AI SENSE pin. In NRSE mode, this pin is connected internally to the negative (-) input of the AT-MIO-16F-5 instrumentation amplifier. In the DIFF and RSE modes, this signal is left unconnected.

Pins 3 through 18 are ACH<15:0> signal pins. These pins are tied to the 16 analog input channels of the AT-MIO-16F-5. In single-ended mode, signals connected to ACH<15:0> are routed to the positive (+) input of the AT-MIO-16F-5 instrumentation amplifier. In differential mode, signals connected to ACH<7:0> are routed to the positive (+) input of the AT-MIO-16F-5 instrumentation amplifier, and signals connected to ACH<15:8> are routed to the negative (-) input of the AT-MIO-16F-5 instrumentation amplifier.

Warning: Exceeding the differential and common mode input ranges results in distorted input signals. Exceeding the maximum input voltage rating can result in damage to the AT-MIO-16F-5 board and to the PC. National Instruments is not liable for any damages resulting from such signal connections.

Connection of analog input signals to the AT-MIO-16F-5 depends on the configuration of the AT-MIO-16F-5 analog input circuitry and the type of input signal source. With the different AT-MIO-16F-5 configurations, the AT-MIO-16F-5 instrumentation amplifier can be used in different ways. Figure 2-15 shows a diagram of the AT-MIO-16F-5 instrumentation amplifier.

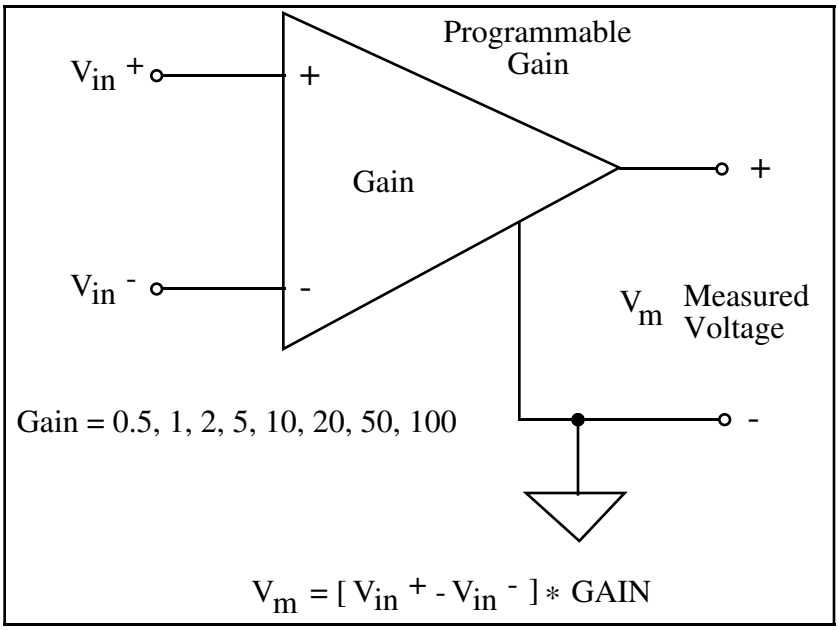


Figure 2-15. AT-MIO-16F-5 Instrumentation Amplifier

The AT-MIO-16F-5 instrumentation amplifier applies gain and common-mode voltage rejection, and presents high-input impedance to the analog input signals connected to the AT-MIO-16F-5 board. Signals are routed to the positive (+) and negative (-) inputs of the instrumentation amplifier through input multiplexers on the AT-MIO-16F-5. The instrumentation amplifier converts two input signals to a signal that is the difference between the two input signals multiplied by the gain setting of the amplifier. The amplifier output voltage is referenced to the AT-MIO-16F-5 ground. The AT-MIO-16F-5 ADC measures this output voltage when it performs A/D conversions.

All signals must be referenced to ground, either at the source device or at the AT-MIO-16F-5. If you have a floating source, the AT-MIO-16F-5 should reference the signal to ground by using the RSE mode or the DIFF input configuration with bias resistors. If you have a grounded source, the AT-MIO-16F-5 should not reference the signal to ground. The board avoids this reference by using the DIFF or NRSE configurations.

Types of Signal Sources

When configuring the input mode of the AT-MIO-16F-5 and making signal connections, you must first determine whether the signal source is floating or ground-referenced. These two types of signals are described in the following sections.

Floating Signal Sources

A floating signal source is one that is not connected in any way to the building ground system but rather has an isolated ground reference point. Some examples of floating signal sources are outputs of transformers, thermocouples, battery-powered devices, optical isolator outputs, and isolation amplifiers. The ground reference of a floating signal must be tied to the

AT-MIO-16F-5 analog input ground in order to establish a local or onboard reference for the signal. Otherwise, the measured input signal varies or appears to float. An instrument or device that provides an isolated output falls into the floating signal source category.

Ground-Referenced Signal Sources

A ground-referenced signal source is one that is connected in some way to the building system ground and is therefore already connected to a common ground point with respect to the AT-MIO-16F-5 board, assuming that the PC AT is plugged into the same power system. Non-isolated outputs of instruments and devices that plug into the building power system fall into this category.

The difference in ground potential between two instruments connected to the same building power system is typically between 1 mV and 100 mV but can be much higher if power distribution circuits are not properly connected. If grounded signal source is measured improperly, this difference may show up as an error in the measurement. The connection instructions for grounded signal sources below are designed to eliminate this ground potential difference from the measured signal.

Input Configurations

The AT-MIO-16F-5 can be configured for one of three input modes: NRSE, RSE, or DIFF. The following sections discuss the use of single-ended and differential measurements, and considerations for measuring both floating and ground-referenced signal sources. Table 2-9 summarizes the recommended input configuration for both types of signal sources.

Table 2-9. Recommended Input Configurations for Ground-Referenced and Floating Signal Sources

Type of Signal	Recommended Input Configuration
Ground-Referenced (non-isolated outputs, plug-in instruments)	DIFF NRSE
Floating (batteries, thermocouples, isolated outputs)	DIFF with bias resistors RSE

Differential Connection Considerations (DIFF Configuration)

Differential connections are those in which each AT-MIO-16F-5 analog input signal has its own reference signal or signal return path. These connections are available when the AT-MIO-16F-5 is configured in the DIFF mode. Each input signal is tied to the positive (+) input of the instrumentation amplifier; and its reference signal, or return, is tied to the negative (-) input of the instrumentation amplifier.

When the AT-MIO-16F-5 is configured for differential input, each signal uses two multiplexer inputs–one for the signal and one for its reference signal. Therefore, with a differential

configuration, only eight analog input channels are available. Differential input connections should be used when any of the following conditions are present:

- Input signals are low-level (less than 1 V).
- Leads connecting the signals to the AT-MIO-16F-5 are greater than 15 ft.
- Any of the input signals require a separate ground reference point or return signal.
- The signal leads travel through noisy environments.

Differential signal connections reduce picked-up noise and increase common mode signal and noise rejection. With these connections, input signals can float within the common mode limits of the input instrumentation amplifier.

Differential Connections for Ground-Referenced Signal Sources

Figure 2-16 shows how to connect a ground-referenced signal source to an AT-MIO-16F-5 board configured for DIFF input. The AT-MIO-16F-5 analog input circuitry must be configured for DIFF input to make these types of connections. Configuration instructions are included in Chapter 4, *Programming.*

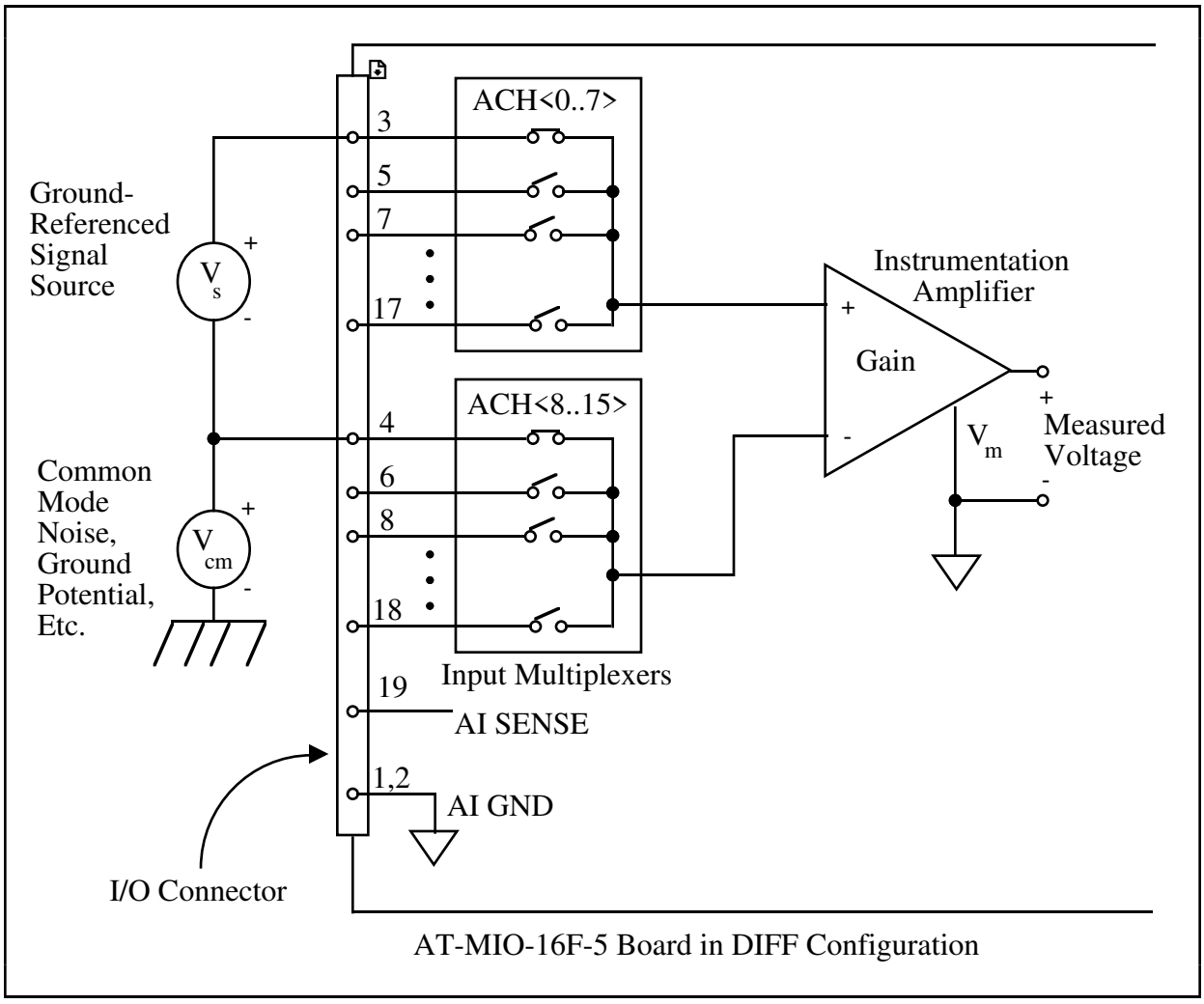


Figure 2-16. Differential Input Connections for Ground-Referenced Signals

With this type of connection, the instrumentation amplifier rejects both the common mode noise in the signal and the ground potential difference between the signal source and the AT-MIO-16F-5 ground (shown as V_{cm} in Figure 2-16).

Differential Connections for Non-Referenced or Floating Signal Sources

Figure 2-17 shows how to connect a floating signal source to an AT-MIO-16F-5 board configured for DIFF input. The AT-MIO-16F-5 analog input circuitry must be configured for DIFF input to make these types of connections. Configuration instructions are included in Chapter 4, *Programming*.

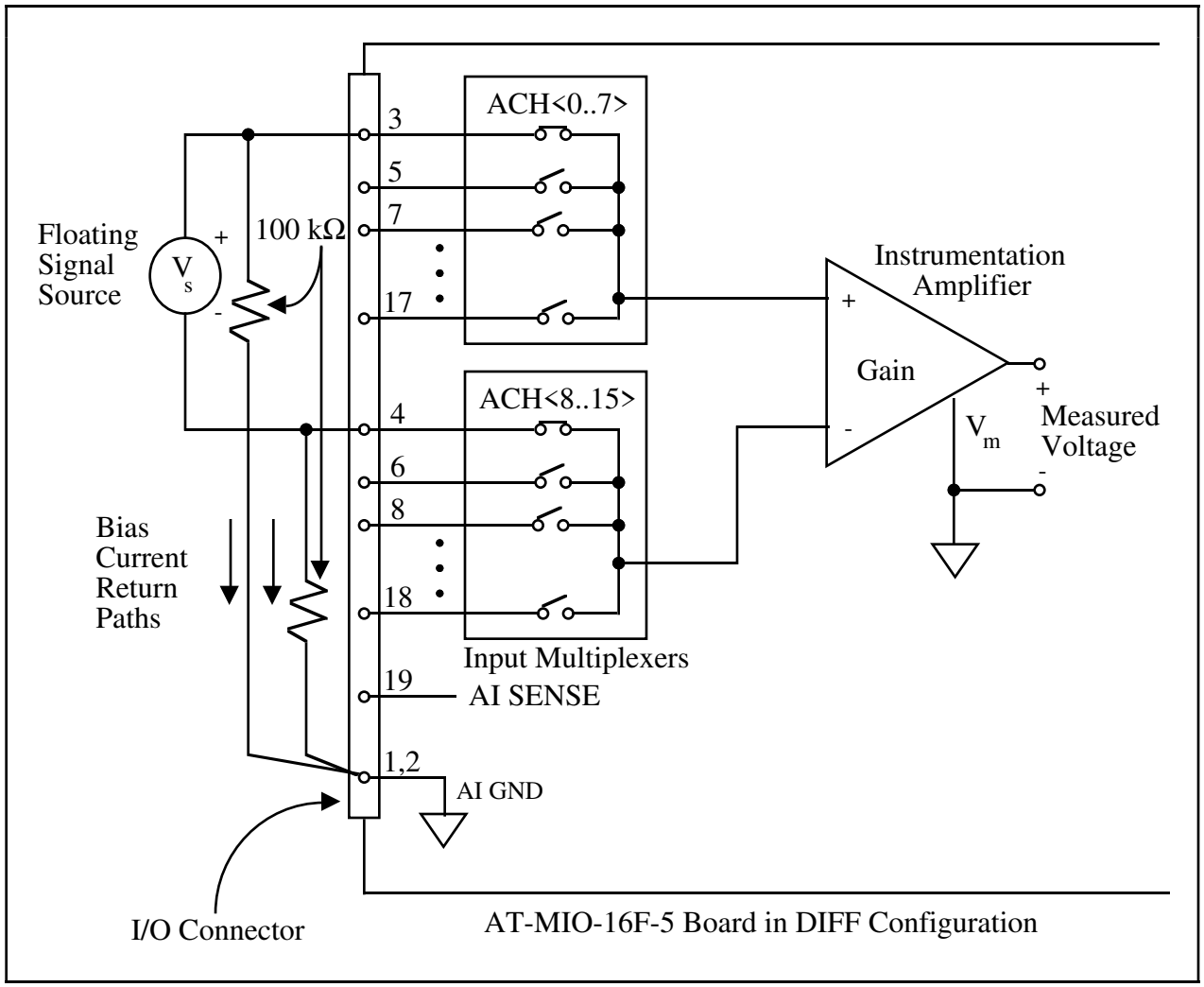


Figure 2-17. Differential Input Connections for Non-Referenced Signals

The 100 kΩ resistors shown in Figure 2-17 create a return path to ground for the bias currents of the instrumentation amplifier. If a return path is not supplied, the instrumentation amplifier bias currents charge up stray capacitances, resulting in uncontrollable drift and possible saturation in the amplifier. Typically, values from 100 kΩ to 1 MΩ are used.

A resistor from each input to ground, as shown in Figure 2-17, supplies bias current return paths for an AC-coupled input signal. This solution, although necessary for AC-coupled signals, lowers the input impedance of the analog input channel. In addition, the input offset current of the instrumentation amplifier contributes a DC offset voltage at the input. The amplifier has a maximum input offset current of ±100 pA and a negligible offset current drift. Multiplied by the 100 kΩ resistor, this current contributes a maximum offset voltage of 10 μV, which is insignificant in most applications. However, the use of larger-valued bias resistors could result in significant offset error.

If the input signal is DC-coupled, then only the resistor connecting the negative (-) signal input to ground is needed. This connection does not lower the input impedance of the analog input channel.

Single-Ended Connection Considerations

Single-ended connections are those in which all AT-MIO-16F-5 analog input signals are referenced to one common ground. The input signals are tied to the positive (+) input of the instrumentation amplifier, and their common ground point is tied to the negative (-) input of the instrumentation amplifier.

When the AT-MIO-16F-5 is configured for single-ended input, 16 analog input channels are available. Single-ended input connections can be used when all input signals meet the following criteria:

- Input signals are high-level (greater than 1 V).
- Leads connecting the signals to the AT-MIO-16F-5 are less than 15 ft.
- All input signals share a common reference signal (at the source).

If any of the preceding criteria are not met, using differential input connections is recommended for greater signal integrity.

The AT-MIO-16F-5 can be software-configured for two different types of single-ended connections: RSE configuration and NRSE configuration. The RSE configuration is used for floating signal sources; in this case, the AT-MIO-16F-5 provides the reference ground point for the external signal. The NRSE configuration is used for ground-referenced signal sources; in this case, the external signal supplies its own reference ground point and the AT-MIO-16F-5 should not supply one.

Single-Ended Connections for Floating Signal Sources (RSE Configuration)

Figure 2-18 shows how to connect a floating signal source to an AT-MIO-16F-5 board configured for single-ended input. The AT-MIO-16F-5 analog input circuitry must be configured for RSE input to make these types of connections. Configuration instructions are included in Chapter 4, *Programming.*

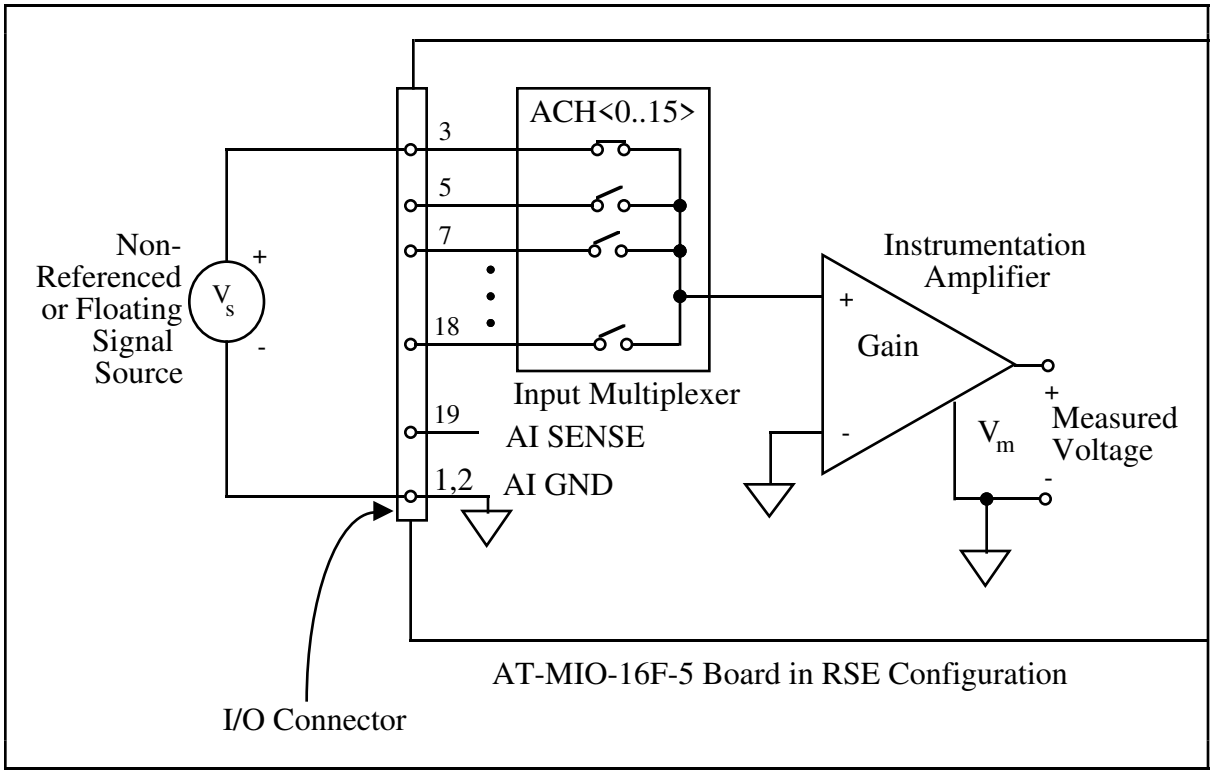


Figure 2-18. Single-Ended Input Connections for Non-Referenced or Floating Signals

Single-Ended Connections for Grounded Signal Sources (NRSE Configuration)

If a grounded signal source is to be measured with a single-ended configuration, then the AT-MIO-16F-5 must be configured in the NRSE input configuration. The signal is connected to the positive (+) input of the AT-MIO-16F-5 instrumentation amplifier and the signal local ground reference is connected to the negative (-) input of the AT-MIO-16F-5 instrumentation amplifier. The ground pont of the signal should therefore be connected to the AISENSE pin. Any potential difference between the AT-MIO-16F-5 ground and the signal ground appears as a common mode signal at both the positive (+) and negative (-) inputs of the instrumentation amplifier and this difference is rejected by the amplifier. On the other hand, if the input circuitry of the AT-MIO-16F-5 is referenced to ground, such as in the RSE configuration, this difference in ground potentials appears as an error in the measured voltage.

Figure 2-19 shows how to connect a grounded signal source to an AT-MIO-16F-5 board configured for single-ended input. The AT-MIO-16F-5 analog input circuitry must be configured for NRSE configuration to make these types of signals. Configuration instructions are included in Chapter 4, *Programming*.

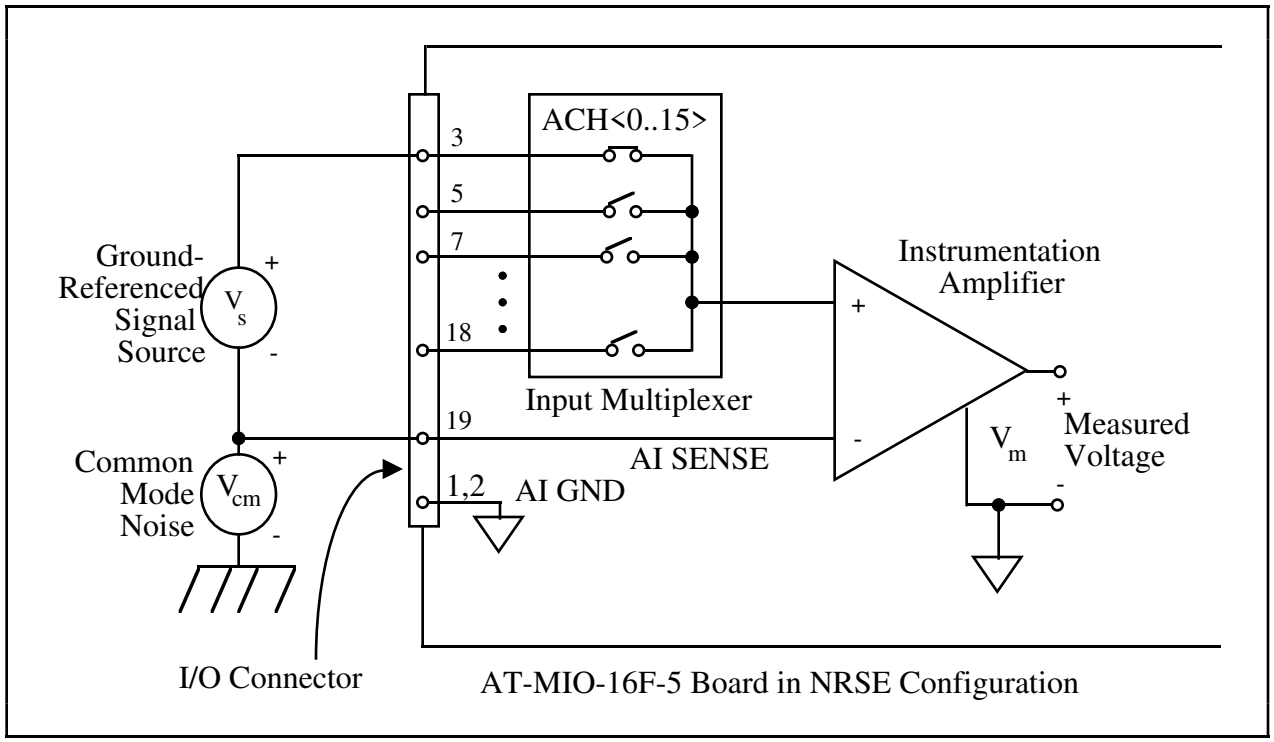


Figure 2-19. Single-Ended Input Connections for Ground-Referenced Signals

Common Mode Signal Rejection Considerations

Figures 2-16 and 2-19, located earlier in this chapter, show connections for signal sources that are already referenced to some ground point with respect to the AT-MIO-16F-5. In these cases, the instrumentation amplifier can reject any voltage caused by ground potential differences between the signal source and the AT-MIO-16F-5. In addition, with differential input connections, the instrumentation amplifier can reject common mode noise pickup in the leads connecting the signal sources to the AT-MIO-16F-5.

The common mode input range of the AT-MIO-16F-5 instrumentation amplifier is defined as the magnitude of the greatest common mode signal that can be rejected. The instrumentation amplifier can reject common mode signals as long as V^+_{in} and V^-_{in} are both in the range ±12 V. Thus the common mode input range for the AT-MIO-16F-5 depends on the size of the differential input signal ($V_{diff} = V^+_{in} - V^-_{in}$). The exact formula for the permissible common mode input range is as follows:

$$V_{cm\text{-}max} = \pm (12\ V - \frac{V_{diff}}{2})$$

Thus, with a differential voltage of 10 V, the maximum possible common mode voltage would be +7 V. The common mode voltage is measured with respect to the AT-MIO-16F-5 ground and can be calculated by the following formula:

$$V_{cm\text{-}actual} = (V^{+}_{in} + V^{-}_{in})/2$$

where V^{+}_{in} is the signal at the positive (+) input of the instrumentation amplifier and V^{-}_{in} is the signal at the negative (-) input of the instrumentation amplifier. Both V^{+}_{in} and V^{-}_{in} are measured with respect to AIGND.

Analog Output Signal Connections

Pins 20 through 23 of the I/O connector are analog output signal pins.

Pins 20 and 21 are the DAC0 OUT and DAC1 OUT signal pins. DAC0 OUT is the voltage output signal for analog output Channel 0. DAC1 OUT is the voltage output signal for analog output Channel 1.

Pin 22, EXTREF, is the external reference input for both analog output channels. Each analog output channel must be configured individually for external reference selection in order for the signal applied at the external reference input to be used by that channel. Analog output configuration instructions are included under the *Analog Output Configuration* section earlier in this chapter.

The following ranges and ratings apply to the EXTREF input:

Useful input voltage range:	±10 V peak with respect to AO GND
Absolute maximum ratings:	±25 V peak with respect to AO GND

Pin 23, AO GND, is the ground reference point for both analog output channels and for the external reference signal.

Figure 2-20 shows how to make analog output connections and the external reference input connection to the AT-MIO-16F-5 board.

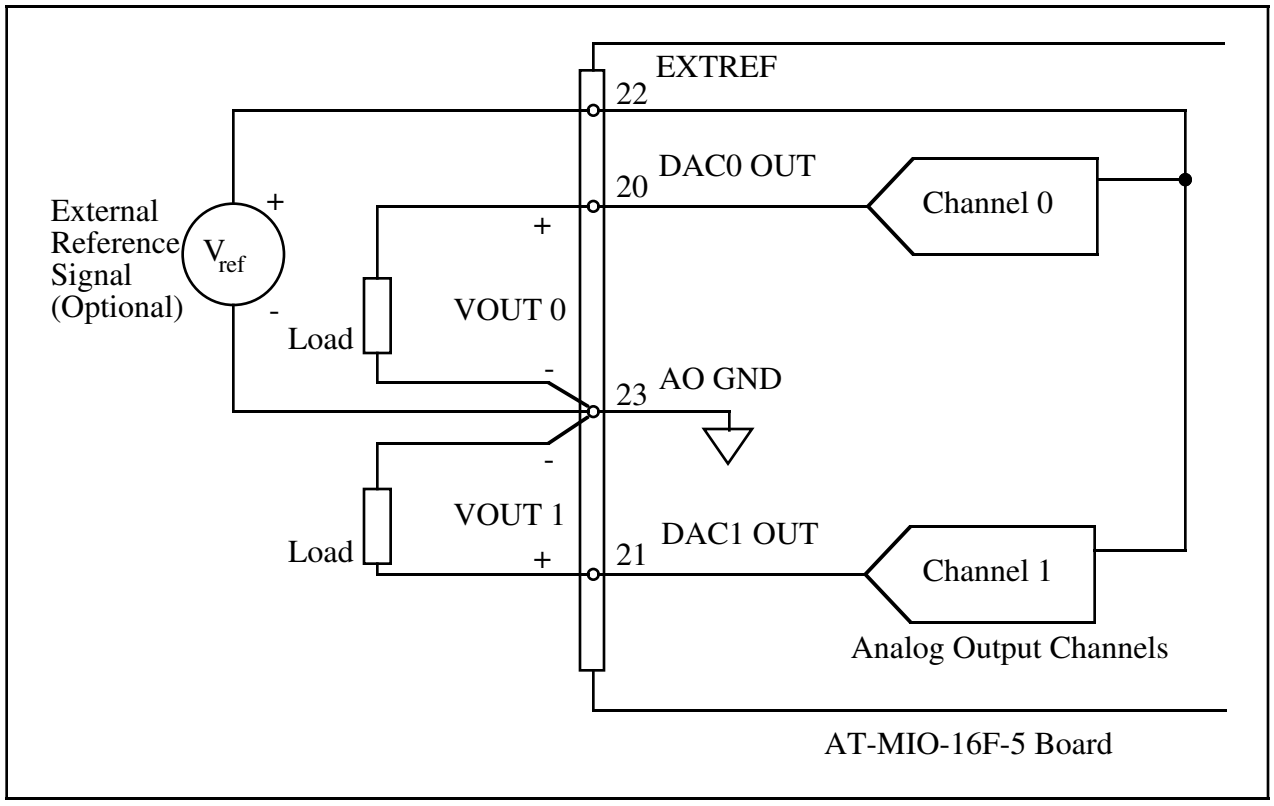


Figure 2-20. Analog Output Connections

The external reference signal can be either a DC or an AC signal. This reference signal is multiplied by the DAC code to generate the output voltage.

Digital I/O Signal Connections

Pins 24 through 32 of the I/O connector are digital I/O signal pins.

Pins 25, 27, 29, and 31 are connected to the digital lines ADIO<3..0> for digital I/O port A. Pins 26, 28, 30, and 32 are connected to the digital lines BDIO<3..0> for digital I/O port B. Pin 24, DIG GND, is the digital ground pin for both digital I/O ports. Ports A and B can be programmed individually to be inputs or outputs.

The following specifications and ratings apply to the digital I/O lines.

Absolute maximum voltage	
Input rating	5.5 V with respect to DIG GND

Digital input specifications (referenced to DIG GND):

V_{IH} input logic high voltage	2 V minimum
V_{IL} input logic low voltage	0.8 V maximum
I_{IH} input current load, logic high input voltage	40 µA maximum
I_{IL} input current load, logic low input voltage	-120 µA maximum

Digital output specifications (referenced to DIG GND):

V_{OH} output logic high voltage	2.4 V minimum
V_{OL} output logic low voltage	0.5 V maximum
I_{OH} output source current, logic high	2.6 mA maximum
I_{OL} output sink current, logic low	24 mA maximum

With these specifications, each digital output line can drive 11 standard TTL loads and over 50 LS TTL loads.

Figure 2-21 depicts signal connections for three typical digital I/O applications.

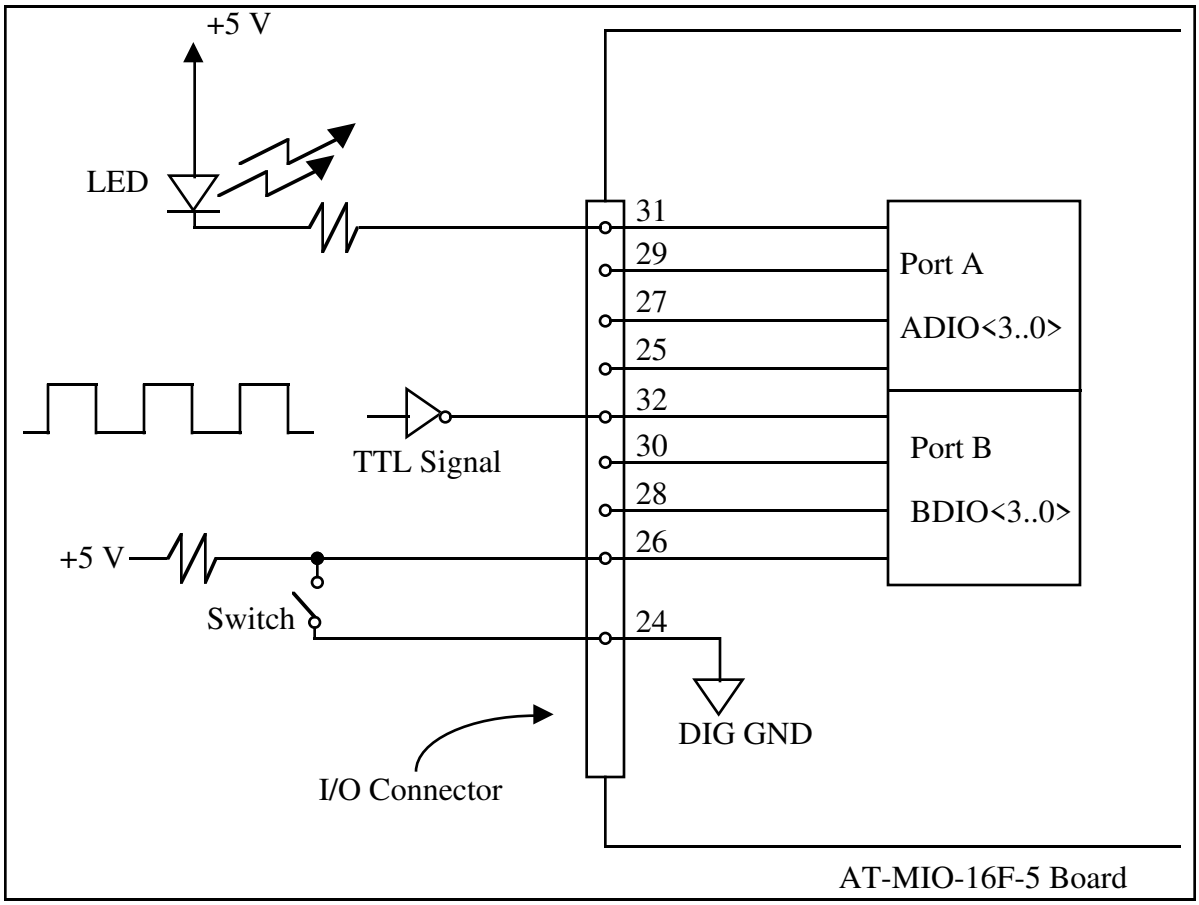


Figure 2-21. Digital I/O Connections

In Figure 2-21, port A is configured for digital output, and port B is configured for digital input. Digital input applications include receiving TTL signals and sensing external device states such as the state of the switch in Figure 2-21. Digital output applications include sending TTL signals and driving external devices such as the LED shown in Figure 2-21.

Power Connections

Pins 34 and 35 of the I/O connector provide +5 V from the PC power supply. These pins are referenced to DIG GND and can be used to power external digital circuitry.

Power Rating	1.0 A at +5 V ± 10%, fused

Warning: *Under no circumstances* should these +5 V power pins be connected directly to analog or digital ground or to any other voltage source on the AT-MIO-16F-5 or any other device. Doing so can damage the AT-MIO-16F-5 and the PC. National Instruments is not liable for damage resulting from such a connection.

Timing Connections

Pins 36 through 50 of the I/O connector are connections for timing I/O signals. Pins 36 through 40 and pin 44 carry signals used for data acquisition timing and analog output triggering. These signals are explained in *Data Acquisition Timing Connections* later in this chapter. Pins 41 through 50 carry general-purpose timing signals and analog output provided by the onboard Am9513A Counter/Timer. These signals are explained in *General-Purpose Timing Signal Connections* later in this chapter.

Data Acquisition and Analog Output Timing Connections

The data acquisition and analog output timing signals are SCANCLK, EXTSTROBE*, EXTTRIG*, EXTGATE*, EXTCONV*, and EXTDACUPDATE*.

SCANCLK Signal

SCANCLK is an output signal that generates a low-to-high edge whenever an A/D conversion begins. SCANCLK pulses only when scanning is enabled on the AT-MIO-16F-5. SCANCLK is normally low and pulses high for approximately 4 µsecs after the A/D conversion begins. The low-to-high edge can be used to clock external analog input multiplexers. The SCANCLK signal is driven by one CMOS TTL gate.

EXTSTROBE Signal

A low pulse is generated on the EXTSTROBE* pin when the External Strobe Register is loaded (see *External Strobe Register* in Chapter 4, *Programming*). Figure 2-22 shows the timing for the EXTSTROBE* signal.

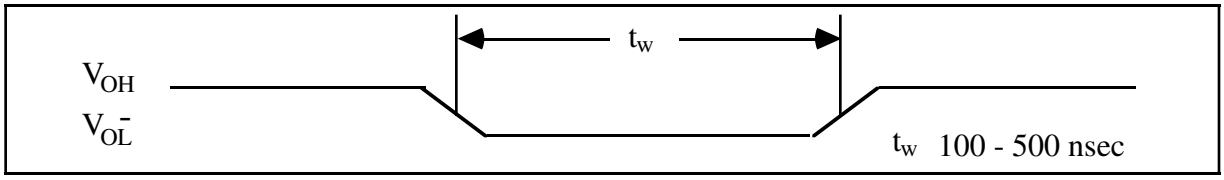


Figure 2-22. EXTSTROBE* Signal Timing

The pulse is typically between 200 nsec and 500 nsec in width. The EXTSTROBE* signal can be used by an external device to latch signals or trigger events. The EXTSTROBE* signal is an HCT signal.

EXTCONV Signal

A/D conversions can be externally triggered with the EXTCONV* pin. Applying an active low pulse to the EXTCONV* signal initiates an A/D conversion. Figure 2-23 shows the timing requirements for the EXTCONV* signal.

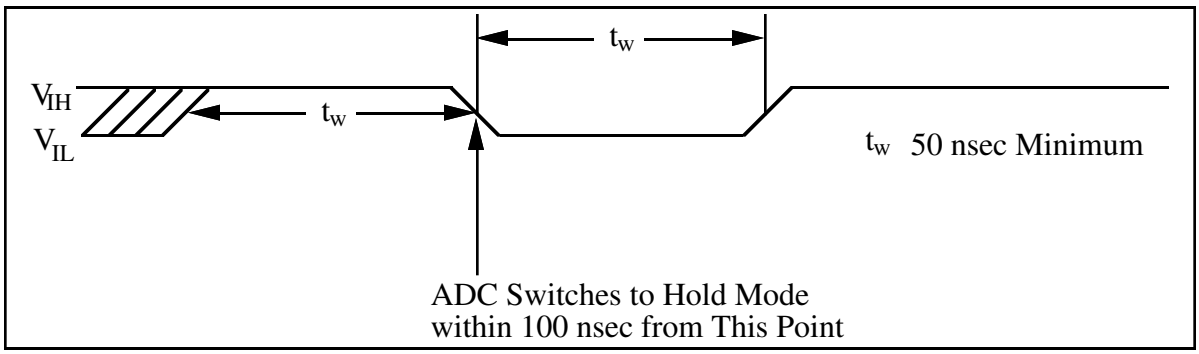


Figure 2-23. EXTCONV* Signal Timing

The minimum allowed pulse width is 50 nsec. The ADC switches to hold mode within 100 nsec of the high-to-low edge. This hold mode delay time is a function of temperature and does not vary from conversion to consecutive conversion. There is no maximum pulse width limitation. EXTCONV* should be high for at least one conversion period before going low. The EXTCONV* signal is one HCT load and is pulled up to +5 V through a 10 kΩ resistor.

Note: EXTCONV* is also driven by the output of Counter 3 of the Am9513A Counter/Timer. This counter is also referred to as the sample-interval counter. The output of Counter 3 must be disabled to a high-impedance state if A/D conversions are to be controlled by pulses applied to the EXTCONV* pin. If Counter 3 is used to control A/D conversions, its output signal can be monitored at the EXTCONV* pin.

EXTTRIG* Signal

Any data acquisition sequence can be initiated by an external trigger applied to the EXTTRIG* pin. Applying a falling edge to the EXTTRIG* pin starts the sample and sample-interval counters, thereby initiating a data acquisition sequence. Figure 2-24 shows the timing requirements for the EXTTRIG* signal.

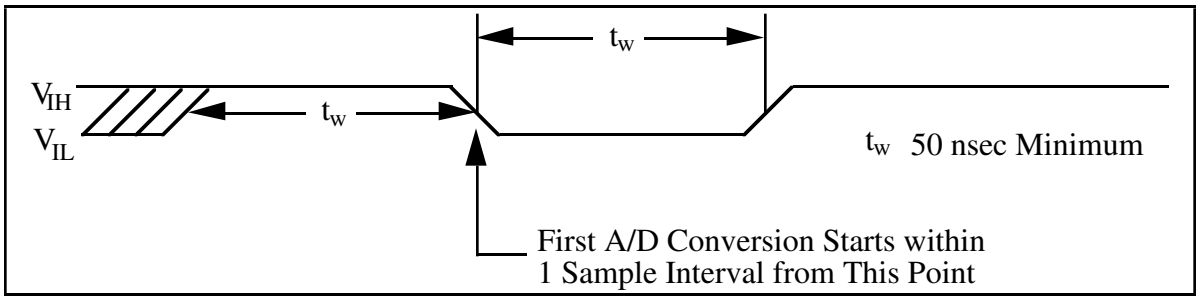


Figure 2-24. EXTTRIG* Signal Timing

The EXTTRIG* pin is also used during AT-MIO-16F-5 pretriggered data acquisition operations. In pretriggered mode, data is acquired but no sample counting occurs until a falling edge is applied to the EXTTRIG* pin. A second falling edge causes the sample counter to then start counting conversions. The acquisition then completes when the sample counter decrements to zero. This mode acquires data both before and after a hardware trigger is received.

The minimum pulse width allowed is 50 nsec. The first A/D conversion starts within one sample interval from the high-to-low edge. The sample interval is controlled by Counter 3 or EXTCONV*.

There is no maximum pulse width limitation; however, EXTTRIG* should be high for at least 50 nsec before going low. The EXTTRIG* signal is one HCT load and is pulled up to +5 V through a 10 kΩ resistor.

Note: The EXTTRIG* signal is logically ANDed with the internal DAQSTART signal. If a data acquisition sequence is to be initated with an internal trigger, EXTTRIG* must be high at both the I/O connector and the RTSI switch. If EXTTRIG* is low, the sequence will not be triggered.

EXTGATE Signal

EXTGATE* is an input signal used for hardware gating. EXTGATE* controls A/D conversion pulses. If EXTGATE* is low, no A/D conversion pulses occur. If EXTGATE* is high, conversions take place if programmed and otherwise enabled.

EXTDACUPDATE* Signal

The analog output DACs on the AT-MIO-16F-5 can be updated using either internal or external signals. The DACs can be updated externally by using the EXTDACUPDATE* signal from the I/O connector. This signal updates the DACs when the WGEN bit in Command Register 2 is set and A4RCV is disabled.

The analog output DACs are updated by the high-to-low edge of the applied pulse. Figure 2-25 shows the timing requirements for the EXTDACUPDATE* signal.

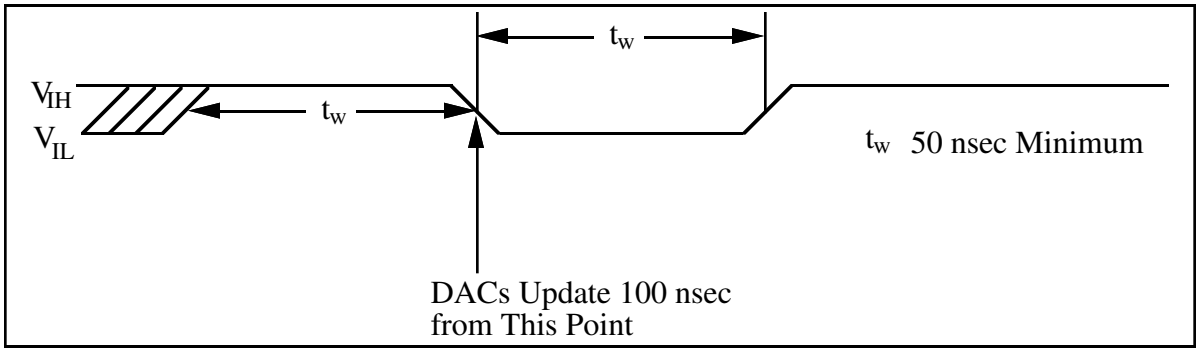


Figure 2-25. EXTDACUPDATE* Signal Timing

The minimum pulse width allowed is 50 nsec. The DACs are updated within 100 nsec of the high-to-low edge. There is no maximum pulse width limitation. EXTDACUPDATE* should be high for at least 50 nsec before going low. The EXTDACUPDATE* signal is one HCT load and is pulled up to +5V through a 10 kΩ resistor.

General-Purpose Timing Signal Connections

The general-purpose timing signals include the GATE and OUT signals for the Am9513A Counters 1, 2, and 5, SOURCE signals for Counters 1 and 5, and the FOUT signal generated by the Am9513A. Counters 1, 2, and 5 of the Am9513A Counter/Timer can be used for general-purpose applications, such as pulse and square wave generation, event counting, pulse-width, time-lapse, and frequency measurements. For these applications, SOURCE and GATE signals can be directly applied to the counters from the I/O connector. The counters are programmed for various operations.

The Am9513A Counter/Timer is described briefly in Chapter 3, *Theory of Operation*. For detailed programming information, consult the Am9513A data sheet in Appendix C. For detailed applications information, consult the technical manual *The Am9513A/Am9513 System Timing Controller* published by Advanced Micro Devices, Inc.

Pulses and square waves can be produced by programming Counter 1, 2, or 5 to generate a pulse signal at its OUT output pin or to toggle the OUT signal each time the counter reaches the terminal count.

For event counting, one of the counters is programmed to count rising or falling edges applied to any of the Am9513A SOURCE inputs. The counter value can then be read to determine the number of edges that have occurred. Counter operation can be gated on and off during event counting.

Figure 2-26 shows connections for a typical event-counting operation where a switch is used to gate the counter on and off.

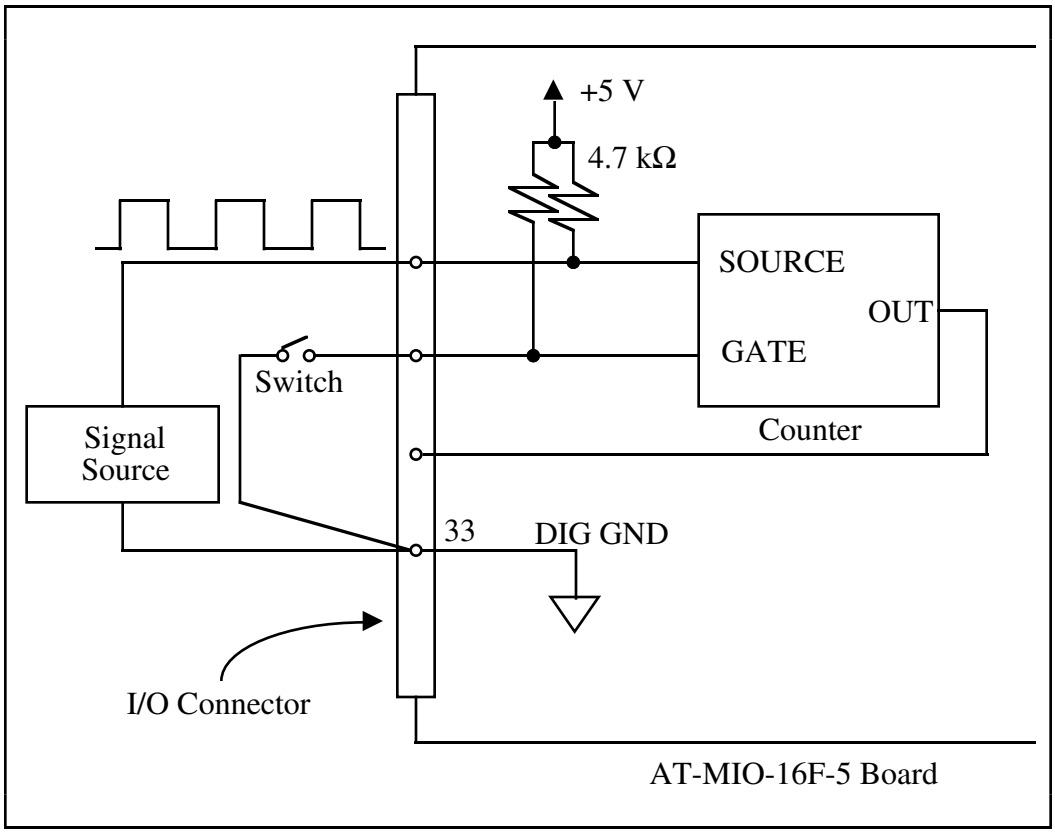


Figure 2-26. Event-Counting Application with External Switch Gating

To perform pulse-width measurement, a counter is programmed to be level-gated. The pulse to be measured is applied to the counter GATE input. The counter is programmed to count while the signal at the GATE input is either high or low. If the counter is programmed to count an internal timebase, then the pulse width is equal to the counter value multiplied by the timebase period.

For time-lapse measurement, a counter is programmed to be edge-gated. An edge is applied to the counter GATE input to start the counter. The counter can be programmed to start counting after receiving either a high-to-low edge or a low-to-high edge. If the counter is programmed to count an internal timebase, then the time lapse since receiving the edge is equal to the counter value multiplied by the timebase period.

To measure frequency, a counter is programmed to be level-gated and the rising or falling edges are counted in a signal applied to a SOURCE input. The gate signal applied to the counter GATE input is of some known duration. In this case, the counter is programmed to count either

rising or falling edges at the SOURCE input while the gate is applied. The frequency of the input signal is then the count value divided by the known gate period. Figure 2-27 shows the connections for a frequency measurement application. A second counter can also be used to generate the gate signal in this application.

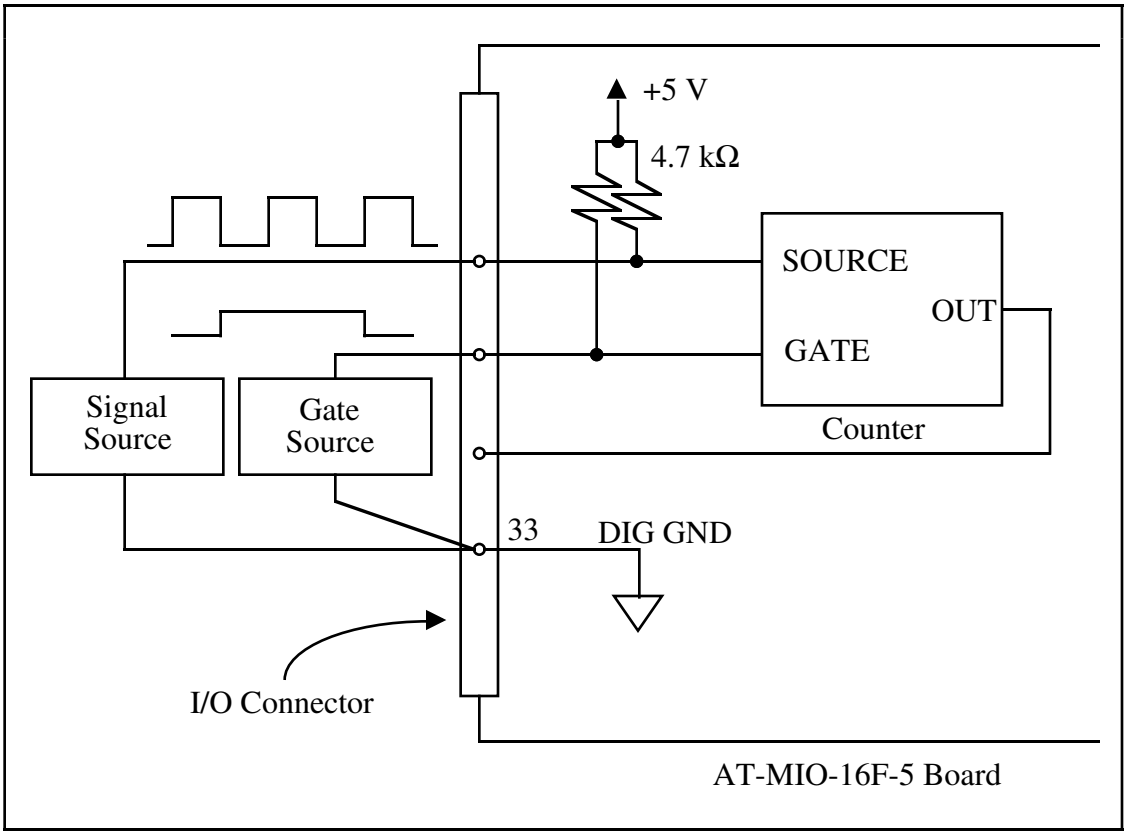


Figure 2-27. Frequency Measurement Application

Two or more counters can be concatenated by tying the OUT signal from one counter to the SOURCE signal of another counter. The counters can then be treated as one 32-bit or 48-bit counter for most counting applications.

The signals for Counters 1, 2, and 5, and the FOUT output signal are tied directly from the Am9513A input and output pins to the I/O connector. In addition, the GATE, SOURCE, and OUT1 pins are pulled up to +5 V through a 4.7 kΩ resistor. The input and output ratings and timing specifications for the Am9513A signals are given as follows:

Absolute maximum voltage input rating	-0.5 V to +7.0 V with respect to DIG GND

Am9513A digital input specifications (referenced to DIG GND):

V_{IH} input logic high voltage	2.2 V minimum
V_{IL} input logic low voltage	0.8 V maximum
Input load current	±10 µA maximum

Am9513A digital output specifications (referenced to DIG GND):

V_{OH} output logic high voltage	2.4 V minimum
V_{OL} output logic low voltage	0.4 V maximum
I_{OH} output source current, at V_{OH}	200 µA maximum
I_{OL} output sink current, at V_{OL}	3.2 mA maximum
Output current, high-impedance state	±25 µA maximum

Figure 2-28 shows the timing requirements for the GATE and SOURCE input signals and the timing specifications for the OUT output signals of the Am9513A.

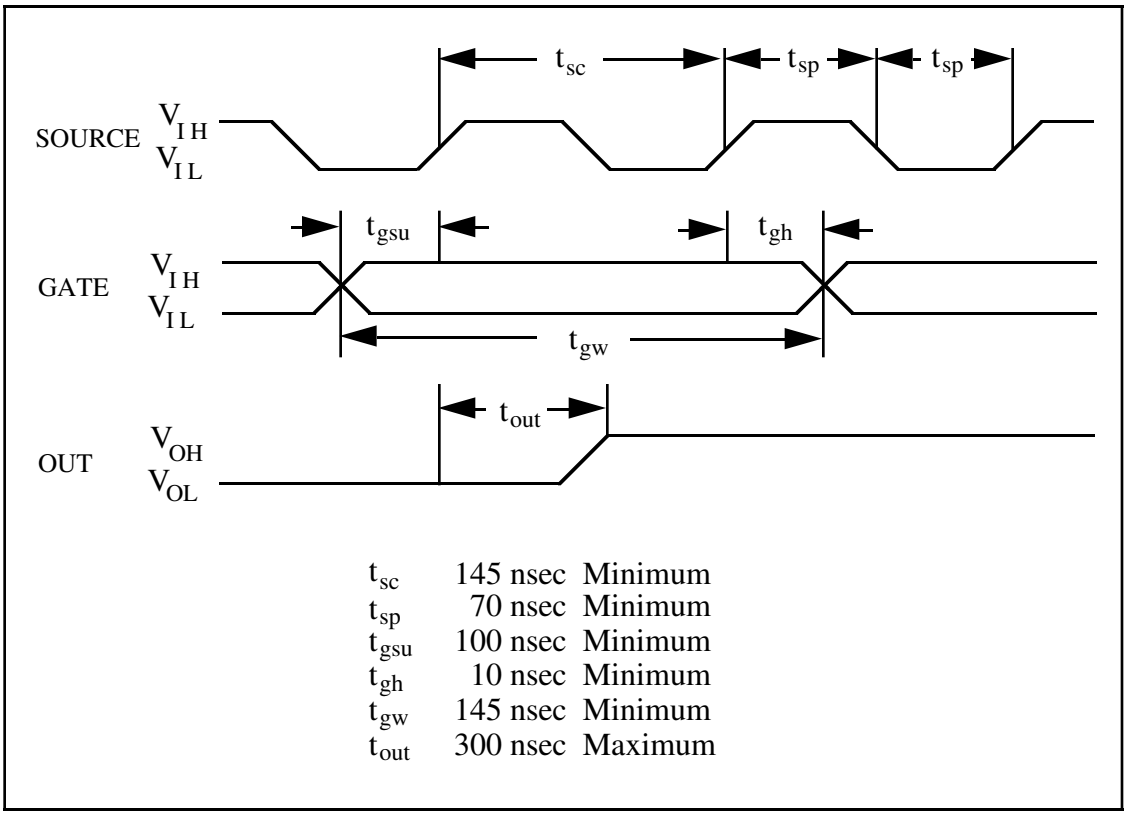


Figure 2-28. General-Purpose Timing Signals

The GATE and OUT signal transitions in Figure 2-28 are referenced to the rising edge of the SOURCE signal. This timing diagram assumes that the counters are programmed to count rising edges. The same timing diagram, with the source signal inverted and referenced to the falling edge of the source signal, applies to the case in which the counter is programmed to count falling edges.

The signal applied at a SOURCE input can be used as a clock source by any of the Am9513A counter/timers and by the Am9513A frequency division output FOUT. The signal applied to a SOURCE input must not exceed a frequency of 6 MHz for proper operation of the Am9513A. The Am9513A counters can be individually programmed to count rising or falling edges of signals applied at any of the Am9513A SOURCE or GATE input pins.

In addition to the signals applied to the SOURCE and GATE inputs, the Am9513A generates five internal timebase clocks from the clock signal supplied by the AT-MIO-16F-5. This clock signal is selected by the W5 jumper and then divided by 10. The factory default value is 1 MHz into the Am9513A (10-MHz clock signal on the AT-MIO-16F-5). The five internal timebase clocks can be used as counting sources, and these clocks have a maximum skew of 75 nsec

between them. The SOURCE signal shown in Figure 2-28 represents any of the signals applied at the SOURCE inputs, GATE inputs, or internal timebase clocks. See the Am9513A data sheet in Appendix C for further details.

Specifications for signals at the GATE input are referenced to the signal at the SOURCE input or one of the Am9513A internally generated signals. Figure 2-28 shows the GATE signal referenced to the rising edge of a source signal. The gate must be valid (either high or low) at least 100 nsec before the rising or falling edge of a source signal for the gate to take effect at that source edge (as shown by t_{gsu} and t_{gh} in Figure 2-28). Similarly, the gate signal must be held for at least 10 nsec after the rising or falling edge of a source signal for the gate to take effect at that source edge. The gate high or low period must be at least 145 nsec in duration. If an internal timebase clock is used, the gate signal cannot be synchronized with the clock. In this case, gates applied close to a source edge take effect either on that source edge or on the next one. This arrangement results in an uncertainty of one source clock period with respect to unsynchronized gating sources.

Signals generated at the OUT output are referenced to the signal at the SOURCE input or to one of the Am9513A internally generated clock signals. Figure 2-28 shows the OUT signal referenced to the rising edge of a source signal. Any OUT signal state changes occur within 300 nsec after the source signal rising or falling edge.

Field Wiring Considerations

Accuracy of measurements made with the AT-MIO-16F-5 can be seriously affected by environmental noise if proper considerations are not taken into account when running signal wires between signal sources and the AT-MIO-16F-5 board. The following recommendations apply mainly to analog input signal routing to the AT-MIO-16F-5 board, though they are applicable for signal routing in general.

Noise pickup can be minimized and measurement accuracy maximized by doing the following:

- Use individually shielded, twisted-pair wires to connect analog input signals to the AT-MIO-16F-5. With this type of wire, the signals attached to the CH+ and CH- inputs are twisted together and then covered with a shield. This shield is then connected only at one point to the signal source ground. This kind of connection is required for signals traveling through areas with large magnetic fields or high electromagnetic interference.

- Use differential analog input connections to reject common mode noise.

The following recommendations apply for all signal connections to the AT-MIO-16F-5:

- Separate AT-MIO-16F-5 signal lines from high-current or high-voltage lines. These lines are capable of inducing currents in or voltages on the AT-MIO-16F-5 signal lines if they run in parallel paths at a close distance. Reduce the magnetic coupling between lines by separating them by a reasonable distance if they run in parallel, or by running the lines at right angles to each other.

- Do not run AT-MIO-16F-5 signal lines through conduits that also contain power lines.
- Protect AT-MIO-16F-5 signal lines from magnetic fields caused by electric motors, welding equipment, breakers, or transformers by running the AT-MIO-16F-5 signal lines through special metal conduits.

Cabling Considerations

National Instruments has a cable termination accessory, the CB-50, for use with the AT-MIO-16F-5 board. This kit includes a terminated 50-conductor flat ribbon cable and a connector block. Signal I/O leads can be attached to screw terminals on the connector block and thereby connected to the AT-MIO-16F-5 I/O connector.

The CB-50 can be used for prototyping an application or in situations where AT-MIO-16F-5 interconnections are frequently changed. However, once a final field wiring scheme has been developed, you may want to develop your own cable. This section contains information and guidelines for design of such a cable.

In making your own cabling, you may decide to shield your cables. The following guidelines can help:

- For the analog input signals, shielded twisted-pair wires for each analog input pair yield the best results, assuming that differential inputs are used. Tie the shield for each signal pair to the ground reference at the source.
- The analog lines, pins 1 through 23, should be routed separately from the digital lines, pins 24 through 50.
- When using a cable shield, use separate shields for the analog and digital halves of the cable. Failure to do so results in noise from switching digital signals coupling into the analog signals.

Chapter 3
Theory of Operation

This chapter contains a functional overview of the AT-MIO-16F-5 board and explains the operation of each functional unit making up the AT-MIO-16F-5.

Functional Overview

The block diagram in Figure 3-1 is a functional overview of the AT-MIO-16F-5 board.

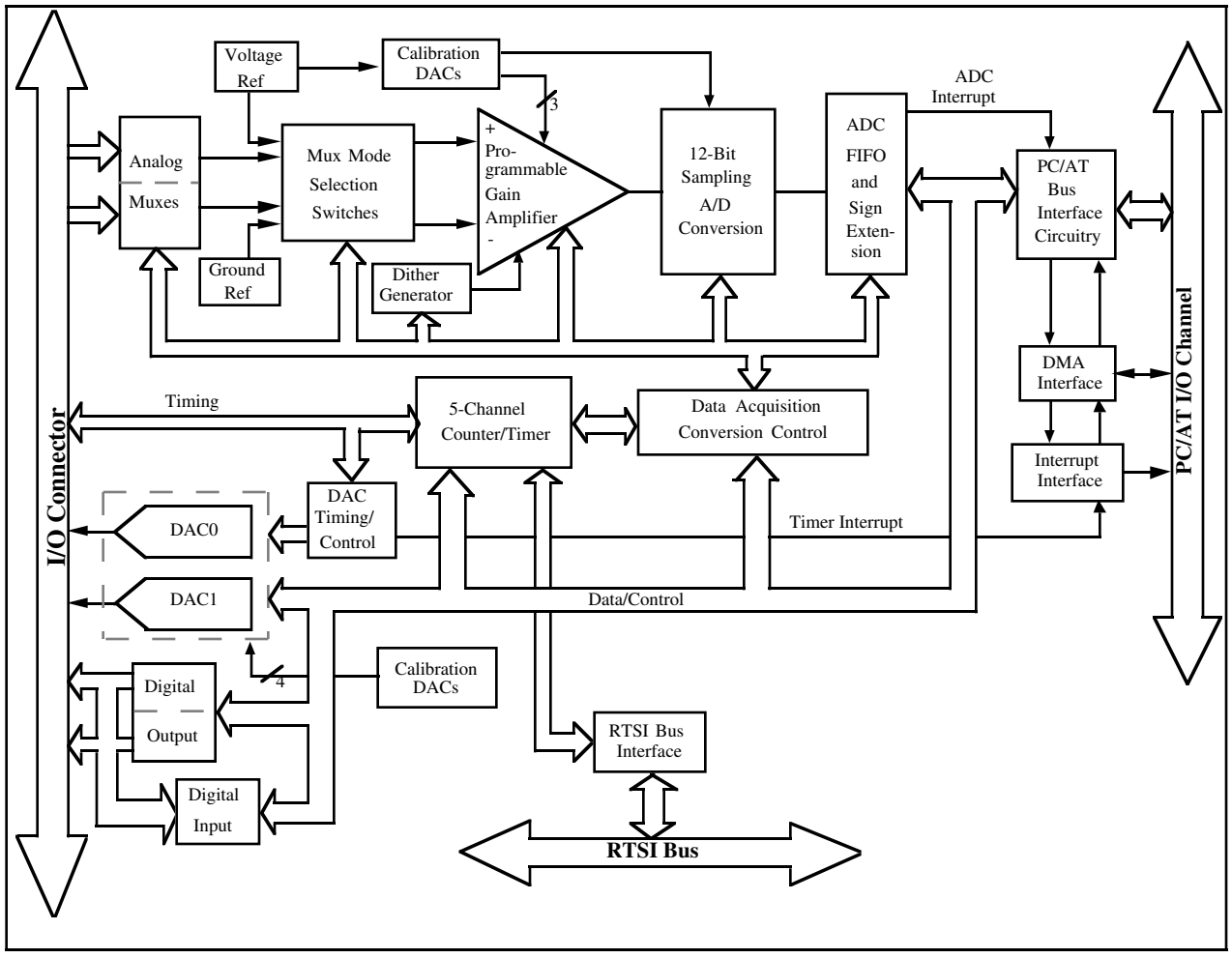


Figure 3-1. AT-MIO-16F-5 Block Diagram

The following are the major components making up the AT-MIO-16F-5 board:

- PC I/O channel interface circuitry
- Analog input and data acquisition circuitry
- Analog output circuitry
- Digital I/O circuitry
- Timing I/O circuitry
- RTSI bus interface circuitry

The internal data and control buses interconnect the components. The theory of operation of each of these components is explained in the remainder of this chapter.

PC I/O Channel Interface Circuitry

The AT-MIO-16F-5 board is a full-size, 16-bit PC I/O channel adapter. The PC I/O channel consists of a 24-bit address bus, a 16-bit data bus, a DMA arbitration bus, interrupt lines, and several control and support signals. The components making up the AT-MIO-16F-5 PC I/O channel interface circuitry are shown in Figure 3-2.

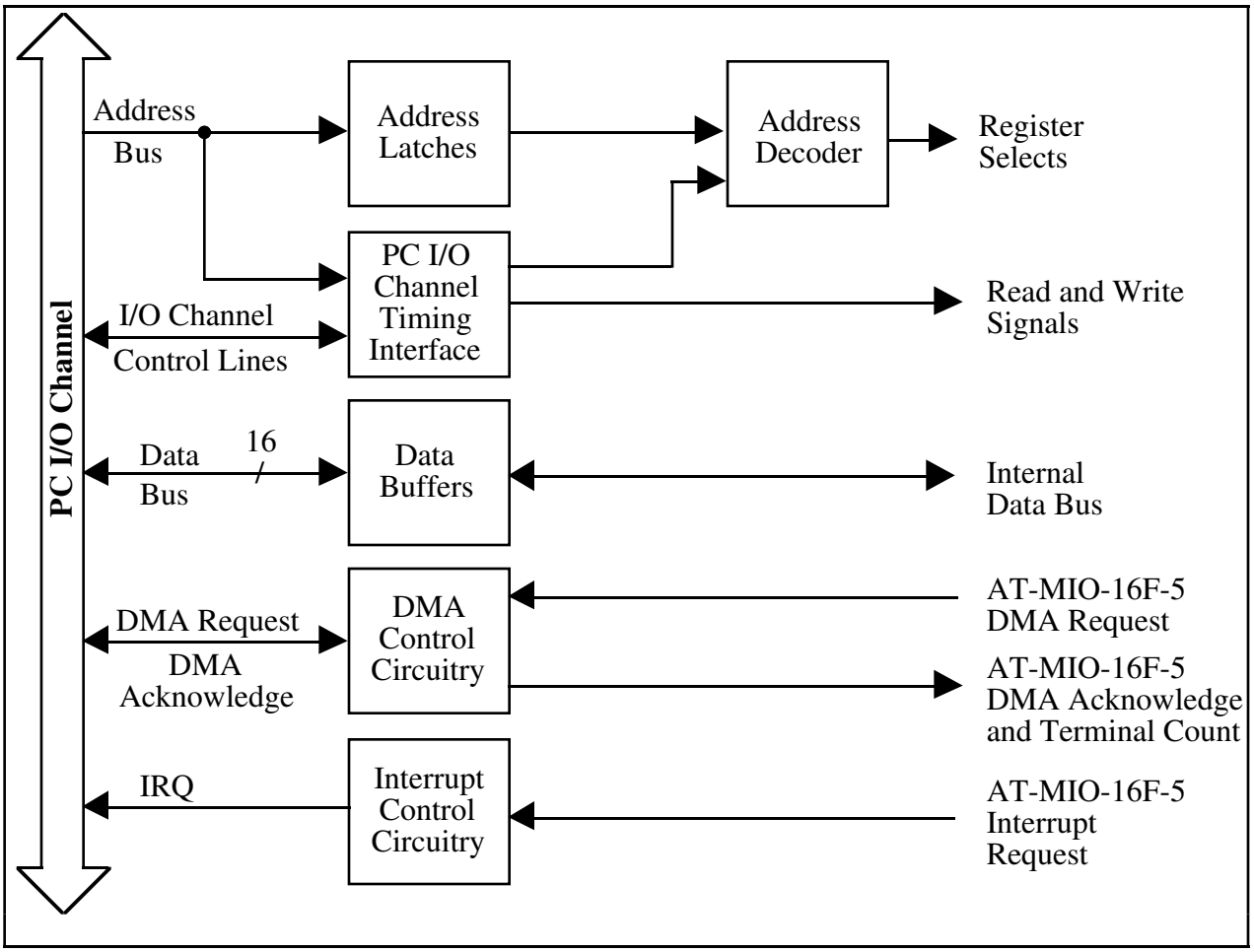


Figure 3-2. PC I/O Channel Interface Circuitry Block Diagram

The PC I/O channel interface circuitry consists of address latches, address decoder circuitry, data buffers, PC I/O channel interface timing signals, interrupt circuitry, and DMA arbitration circuitry. The PC I/O channel interface circuitry generates the signals necessary to control and monitor the operation of the AT-MIO-16F-5 multiple function circuitry.

The PC I/O channel has 24 address lines; the AT-MIO-16F-5 uses 10 of these lines to decode the board address. Therefore, the board address range is 000 to 3FF hex. SA5 through SA9 are used to generate the board enable signal. SA0 through SA4 are used to select onboard registers. These address lines are latched by the address latches at the beginning of an I/O transfer. The latched address lines send the same address to the address-decoding circuitry during the entire I/O transfer cycle. The address-decoding circuitry generates the register select signals that identify which AT-MIO-16F-5 register is being accessed. The data buffers control the direction of data transfer on the bidirectional data lines based on whether the transfer is a read or write.

The PC I/O channel interface timing signals are used to generate read and write signals and to define the transfer cycle. A transfer cycle can be either an 8-bit or a 16-bit data I/O operation. The AT-MIO-16F-5 returns signals to the PC I/O channel to indicate when the board has been accessed, when the board is ready for another transfer, and the data bit size of the current I/O transfer.

The interrupt control circuitry routes any enabled interrupt requests to the selected interrupt request line. The AT-MIO-16F-5 board can share the interrupt line with other devices because the interrupt requests are tri-state output signals. Eleven interrupt request lines are available for use by the AT-MIO-16F-5: IRQ3, IRQ4, IRQ5, IRQ6, IRQ7, IRQ9, IRQ10, IRQ11, IRQ12, IRQ14, and IRQ15. Five different interrupts can be generated by the AT-MIO-16F-5 in the following cases:

- When a single A/D conversion can be read from the A/D FIFO memory.
- When the A/D FIFO is half-full.
- When a data acquisition operation completes, including when either an OVERFLOW or an OVERRUN error occurs.
- When a DMA terminal count pulse is received.
- When a falling edge signal is detected on the DAC update signal.

Each one of these interrupts is individually enabled and cleared. See Chapter 4, *Programming*, for more information about programming with interrupts.

The DMA control circuitry generates DMA requests whenever an A/D measurement is available from the FIFO and when the DACs are updated using one of the methods described previously, if the DMA transfer is enabled. The DMA circuitry supports full PC I/O channel 16-bit DMA transfers. DMA Channels 5, 6, and 7 of the PC I/O channel are available for such transfers. With the DMA circuitry, either single-channel transfer mode or dual-channel transfer mode can be selected for DMA transfer.

Analog Input and Data Acquisition Circuitry

The AT-MIO-16F-5 handles 16 channels of analog input with software-programmable gain and 12-bit A/D conversion. In addition, the AT-MIO-16F-5 contains data acquisition circuitry for automatic timing of multiple A/D conversions and includes advanced options such as external triggering, gating, and clocking. Figure 3-3 shows a block diagram of the analog input and data acquisition circuitry.

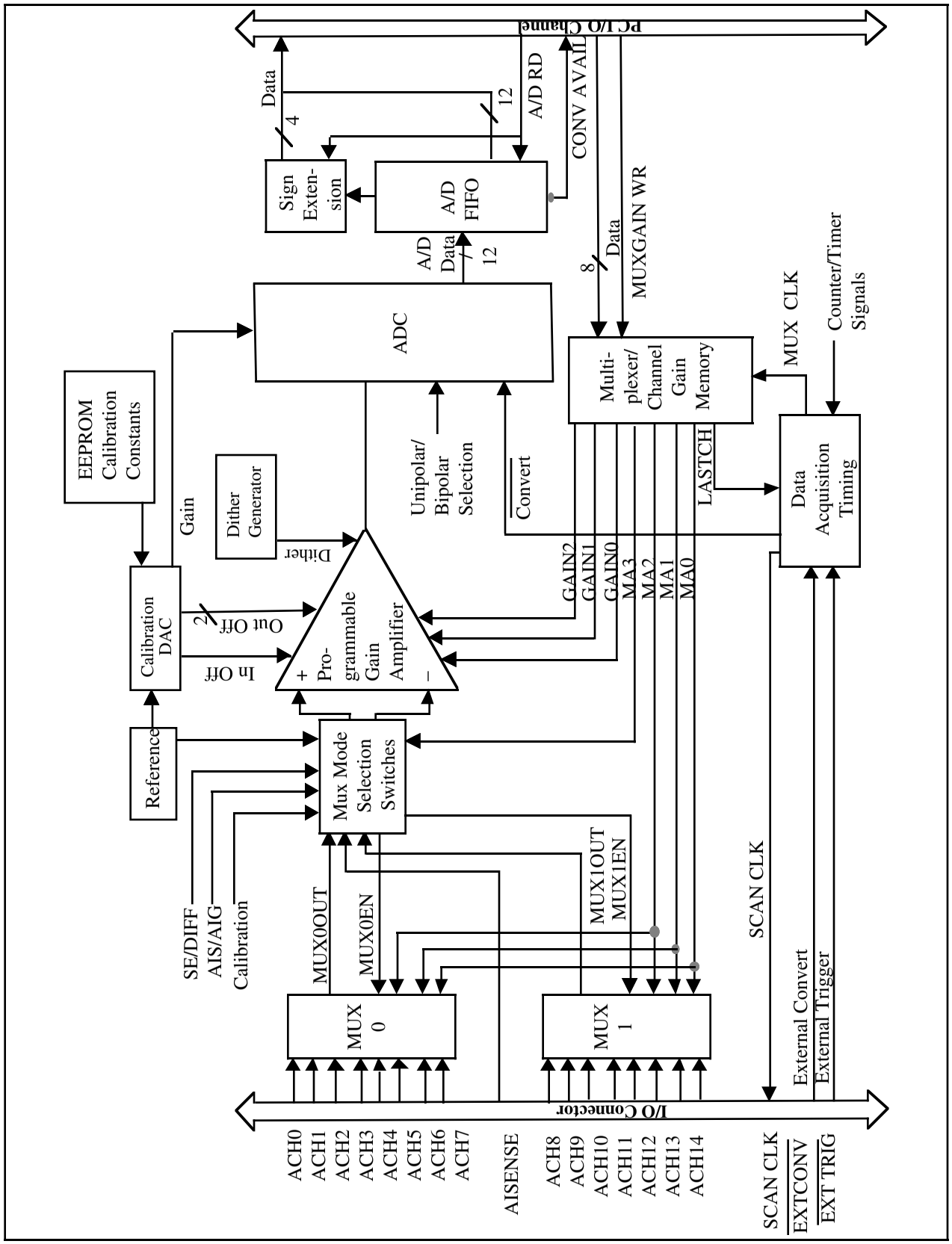


Figure 3-3. Analog Input and Data Acquisition Circuitry Block Diagram

Analog Input Circuitry

The analog input circuitry consists of an input multiplexer, multiplexer mode selection switches, a software-programmable gain instrumentation amplifier, calibration hardware, a 12-bit sampling ADC, and a 16-bit, 256-word deep FIFO with a 16-bit sign extension option.

The input multiplexer consists of two CMOS analog input multiplexers and has 16 analog input channels. Multiplexer MUX0 is connected to analog input Channel 0 through Channel 7. Multiplexer MUX1 is connected to analog input Channel 8 through Channel 15. Analog input overvoltage protection is ±25 V powered on and ±15 V powered off.

The multiplexer mode selection switches are controlled through Command Register 1 and configure the analog input channels as 16 single-ended inputs or 8 differential inputs. When single-ended mode is selected, the outputs of the two multiplexers are tied together and routed to the positive (+) input of the instrumentation amplifier. The negative (-) input of the instrumentation amplifier is tied to the AT-MIO-16F-5 ground for measuring non-referenced single-ended input signals or to the analog return of the input signals via the AI SENSE input on the I/O connector for measuring ground-referenced single-ended input signals. When differential mode is selected, the output of MUX0 is routed to the positive (+) input of the instrumentation amplifier, and the output of MUX1 is routed to the negative (-) input of the instrumentation amplifier. See Table 4-2 in Chapter 4, *Programming*, for more information on input configuration.

The instrumentation amplifier fulfills two purposes on the AT-MIO-16F-5 board. It converts a differential input signal into a single-ended signal with respect to the AT-MIO-16F-5 ground for input common-mode signal rejection as described in Appendix A, *Specifications*. With this conversion the input analog signal can be extracted from any common mode voltage or noise before being sampled and converted. The instrumentation amplifier also applies gain to the input signal, amplifying an input analog signal before sampling and conversion to increase measurement resolution and accuracy. Software selectable gains of 0.5, 1, 2, 5, 10, 20, 50 and 100 are available through the AT-MIO-16F-5 instrumentation amplifier.

The dither circuitry, when enabled, adds approximately 0.5 LSB rms of white Gaussian noise to the signal to be converted by the ADC. This addition is useful for applications involving averaging to increase the resolution of the AT-MIO-16F-5 to more than 12 bits, as in calibration. In such applications, which are often lower frequency in nature, noise modulation is decreased and differential linearity is improved by the addition of the dither. For high-speed 12-bit applications not involving averaging, dither should be disabled because it only adds noise. Enabling and disabling of the dither circuitry is accomplished through software (see Chapter 4, *Programming*).

When taking DC measurements, such as when calibrating the board, enable dither (see Chapter 4, *Programming*) and average about 1,000 points to take a single reading. This process removes the effects of 12-bit quantization and reduces measurement noise, resulting in improved resolution. Dither, or additive white noise, has the effect of forcing quantization noise to become a zero-mean random variable rather than a deterministic function of input. For more information on the effects of dither, see "Dither in Digital Audio" by John Vanderkooy and Stanley P. Lipshitz, *Journal of the Audio Engineering Society*, Vol. 35, No. 12, Dec. 1987.

Measurement reliability is assured through the use of the onboard calibration circuitry of the AT-MIO-16F-5. This circuitry uses an internal, stable, 5 V reference that is measured at the factory against a higher accuracy reference; then its value is stored in the EEPROM on the AT-MIO-16F-5. A storage map of the EEPROM can be found in Chapter 5, *Calibration*

Procedures. With this stored reference value, the AT-MIO-16F-5 board can be recalibrated at any time under any number of different environmental conditions in order to remove errors caused by time and temperature drift. The EEPROM stores calibration constants that can be read and then written to calibration DACs that adjust input offset, output offset, and gain errors associated with the analog input section. When the AT-MIO-16F-5 leaves the factory, locations 52 through 63 of the EEPROM are protected and cannot be overwritten. Locations 0 through 51 are unprotected and can be used to store alternate calibration constants for the differing conditions under which the board is used.

Selection of the analog input channel and gain settings is controlled by the mux-channel-gain memory. With the mux-channel-gain memory, four channel-address bits are available to the input multiplexers and multiplexer mode selection circuitry that selects the analog input channels, and three gain control bits are available to the instrumentation amplifier. Each set of four-channel bits has its own corresponding three gain-selection bits. Operation of the mux-channel-gain memory is explained in more detail in *Data Acquisition Timing Circuitry* later in this chapter.

The ADC is a 12-bit, sampling, successive approximation ADC. With the 12-bit resolution, the converter can resolve its input range into 4,096 different steps. This resolution also generates a 12-bit digital word that represents the value of the input voltage level with respect to the converter input range. The ADC has two input ranges that are software selectable on the AT-MIO-16F-5 board: -5 V to +5 V, or 0 V to +10 V. The ADC on the AT-MIO-16F-5 is guaranteed to convert at a rate of at least 200 ksamples/sec.

When an A/D conversion is complete, the ADC clocks the result into the FIFO. The FIFO is 16 bits wide and 256 words deep. This FIFO serves as a buffer to the ADC and is beneficial for two reasons. Any time an A/D conversion is complete, the value is saved in the FIFO for later reading, and the ADC is free to start a new conversion. Secondly, the FIFO can collect up to 256 A/D conversion values before any information is lost; thus software or DMA has extra time (256 times the sample interval) to catch up with the hardware. If more than 256 values are stored in the FIFO without the FIFO being read from, an error condition called FIFO overflow occurs and A/D conversion information is lost.

The FIFO generates a signal that indicates when it contains a single A/D conversion value. The FIFO also generates a signal that indicates when it is half-filled with A/D conversion data. These signals can be used to generate a DMA or interrupt request signal. Sign-extension circuitry at the FIFO output adds four most significant bits (bits 15 through 12) to the 12-bit ADC output (bits 11 through 0) to produce a 16-bit result. This 16-bit word is then shifted into the A/D FIFO. The sign-extension circuitry is software-programmable to generate either straight binary numbers or two's complement numbers. In straight binary mode, bits 15 through 12 are always 0 and result in a range of 0 to 4095. In two's complement mode, the most significant bit of the 12-bit ADC result, bit 11, is extended to bits 15 through 12, resulting in a range of -2048 to +2047.

Data Acquisition Timing Circuitry

A data acquisition operation refers to the process of taking a sequence of A/D conversions with the sample interval (the time between successive A/D conversions) carefully timed. The data acquisition timing circuitry consists of various clocks and timing signals. Three types of data acquisition are available with the AT-MIO-16F-5 board: single-channel data acquisition, multiple-channel data acquisition with continuous scanning, and multiple-channel data acquisition with interval scanning. All data acquisition operations work with pretrigger and posttrigger modes. Pretriggering acquires data after a software or hardware trigger is applied. When a second trigger is applied, the normal data acquisition sequence is initiated. Posttriggering is a normal data acquisition sequence that can be initiated by a software or hardware trigger.

Scanned data acquisition uses the mux-channel-gain memory to automatically switch between analog input channels and gains during data acquisition. Continuous scanning cycles through the mux-channel-gain memory without any delays between cycles. Interval scanning assigns a time between the starts of consecutive scan sequences. If only one scan sequence is in the mux-channel-gain memory, the circuitry stops at the end of the sequence and waits the necessary interval time before starting the scan sequence again. If multiple scan sequences are in the mux-channel-gain memory, the circuitry stops at the end of each scan sequence and waits the necessary interval time before starting the next scan sequence in memory. When the end of the scan list is reached, the circuitry stops and waits the necessary interval time before sequencing through the channel-gain list again.

Data acquisition timing consists of signals that initiate a data acquisition operation, initiate individual A/D conversions, gate the data acquisition operation, and generate scanning clocks. The sources for these signals can be supplied by timers on the AT-MIO-16F-5 board, by signals connected to the AT-MIO-16F-5 I/O connector, or by signals from other AT Series boards connected to the RTSI bus.

Single A/D conversions can be initiated by applying an active-low pulse to the EXTCONV* input on the I/O connector or by writing to the Start Convert Register on the AT-MIO-16F-5 board. During data acquisition, the onboard sample-interval counter (Counter 3 of the Am9513A Counter/Timer) generates active-low pulses that initiate A/D conversions. External control of the sample interval is possible by applying a stream of pulses at the EXTCONV* input. In this case, you have complete external control over the sample interval and the number of A/D conversions performed. All data acquisition operations are functional with external signals to control conversions.

The sample-interval timer is a 16-bit down counter that can be used with the six internal timebases of the Am9513A to generate sample intervals from 0.4 µsec to 6 sec (see *Timing I/O Circuitry* later in this chapter). The sample-interval timer can also use any of the external clock inputs to the Am9513A as a timebase. During data acquisition, the sample interval counts down at the rate given by the internal timebase or external clock. Each time the sample-interval timer reaches zero, it generates an active low pulse and reloads with the programmed sample-interval count. This operation continues until data acquisition halts.

Data acquisition can be controlled by the onboard sample counter. This counter is loaded with the number of samples to be taken during a data acquisition operation. The sample counter can be 16-bit for counts up to 65,535 or 32-bit for counts up to $(2^{32} - 1)$. If a 16-bit counter is needed, Counter 4 of the Am9513A Counter/Timer is used. If more than 16 bits are needed, Counter 4 is concatenated with Counter 5 of the Am9513A to form a 32-bit counter. The sample counter decrements its count each time the sample-interval counter generates an A/D conversion pulse, and the sample counter stops the data acquisition process when it counts down to 0.

The data acquisition process can be initiated via software or by applying an active low pulse to the EXTTRIG* input on the AT-MIO-16F-5 I/O connector. These triggers start the sample-interval and sample counters. The sample-interval counter then manages the data acquisition process until the sample counter reaches zero.

The sample counter can be triggered in the same way as the data acquisition sequence; externally with the EXTTRIG* input on the AT-MIO-16F-5 I/O connector or internally via software. The counter can also be programmed so that is does not begin counting the A/D conversion pulses until a second falling edge signal occurs on EXTTRIG*. With pretriggering, A/D conversion samples can be collected both before and after a hardware or software trigger is received.

Single-Channel Data Acquisition

During single-channel data acquisition, the mux-channel-gain memory is set up to select the analog input channel and gain before data acquisition is initiated. These channel and gain settings remain constant during the entire data acquisition process; therefore, all A/D conversion data is read from a single channel.

Multiple-Channel (Scanned) Data Acquisition

Multiple-channel data acquisition is performed by enabling scanning during data acquisition. Multiple-channel scanning is controlled by the mux-channel-gain memory.

The mux-channel-gain memory consists of 512 words of memory. Each word of memory contains a multiplexer address (4 bits) for input analog channel selection, a gain setting (3 bits), a bit for synchronizing scanning sequences of different rates, and a bit indicating if the entry is the last in the scan sequence. (In interval scanning, a scan list can consist of any number of scan sequences.) Whenever a mux-channel-gain memory location is selected, the channel and gain control bits contained in that memory location are applied to the analog input circuitry. For scanning operations, the mux counter steps through successive locations in the mux-channel-gain memory at a rate determined by the scan clock. With the mux-channel-gain memory, therefore, an arbitrary sequence of channels with a separate gain setting for each channel can be clocked through during a scanning operation.

A SCANCLK signal is generated from the sample-interval counter. This signal pulses once at the beginning of each A/D conversion and is supplied at the I/O connector. During multiple-channel scanning, the mux-channel-gain memory location pointer is incremented repeatedly, thereby sequencing through the mux-channel-gain memory and automatically selecting new channel and gain settings during data acquisition. The MUXMEMCLK signal is generated from the SCANCLK and generates the pulses that increment the location pointer. In single-channel acquisition mode MUXMEMCLK is disabled and in multiple-channel acquisition mode MUXMEMCLK is enabled. MUXMEMCLK can be identical to SCANCLK, incrementing the mux counter once after every A/D conversion, or it can also be generated by dividing SCANCLK by Counter 1 of the Am9513A Counter/Timer. With this method, the location pointer can be incremented once every *N* A/D conversions so that *N* conversions can be performed on a single channel and gain selection before switching to the next channel and gain selection.

Data Acquisition Rates

The acquisition and channel selection hardware function so that in the channel scanning mode, the next channel in the mux-channel-gain memory is selected immediately after the conversion process has begun on the previous channel. With this method, the input multiplexers and the instrumentation amplifier can settle to the new value within the specified conversion time of the AT-MIO-16F-5, which is 5 µsec maximum.

Analog Output Circuitry

The AT-MIO-16F-5 has two channels of 12-bit D/A output. Unipolar or bipolar output and internal or external reference voltage selection are available with each analog output channel. Figure 3-4 shows a block diagram of the analog output circuitry.

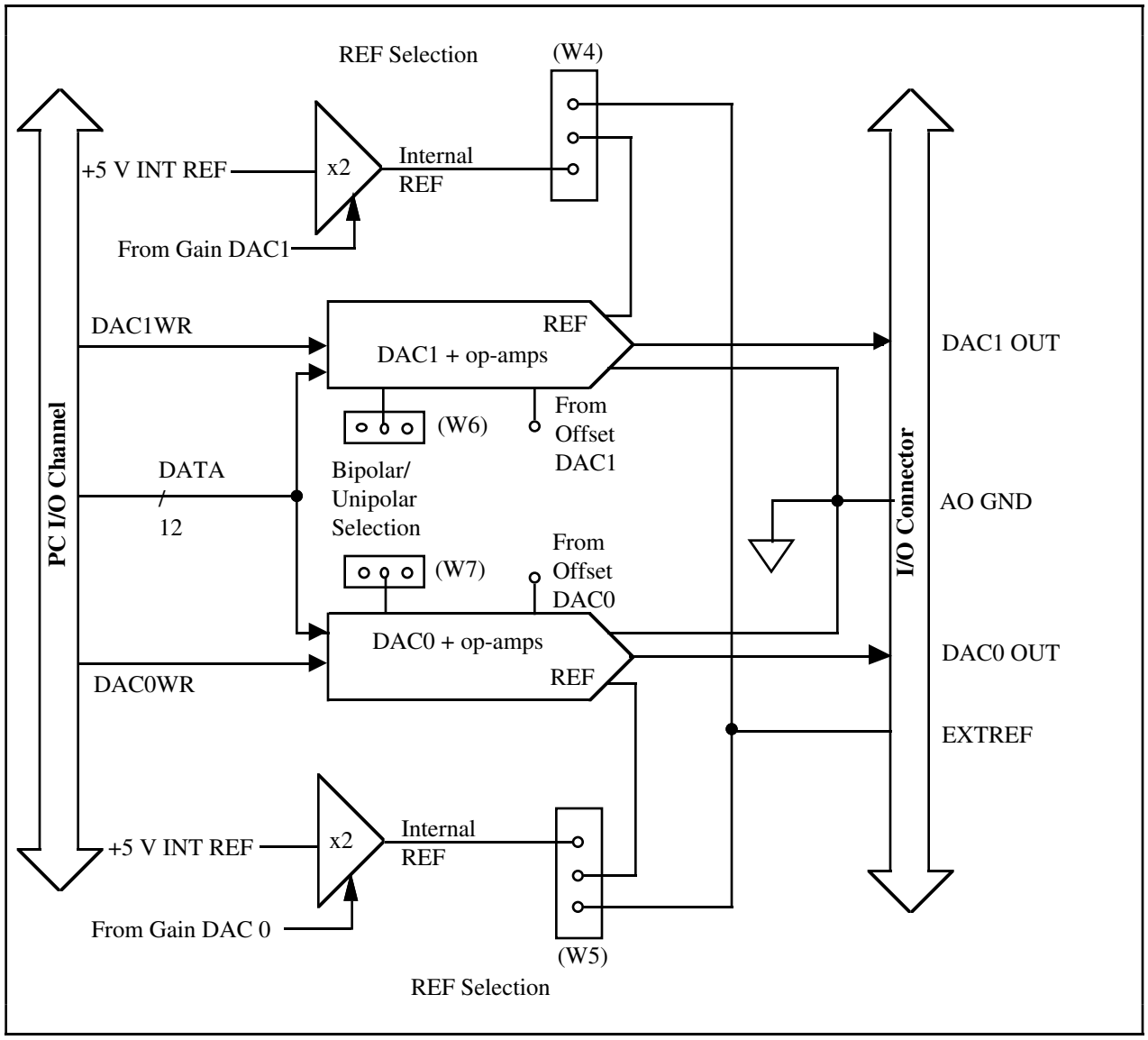


Figure 3-4. Analog Output Circuitry Block Diagram

Each analog output channel contains a 12-bit DAC, output operational amplifiers (op-amps), reference selection jumpers, unipolar/bipolar output selection jumpers, and output data coding jumpers.

The DAC in each analog output channel generates a current proportional to the input voltage reference (V_{ref}) multiplied by the digital code loaded into the DAC. Each DAC can be loaded with a 12-bit digital code by writing to registers on the AT-MIO-16F-5 board. The output op-amps convert the DAC current output to a voltage output on the AT-MIO-16F-5 I/O connector DAC0 OUT and DAC1 OUT pins.

The analog output of the DACs is updated to reflect the loaded 12-bit digital code in one of the following three ways:

- Immediately when the 12-bit code is written to the DACs.
- When an active low pulse is detected on the DACUPTRIG* signal with the WGEN bit set in Command Register 2.
- When the Update Register is strobed with the WGEN bit set in Command Register 2.

The AT-MIO-16F-5 incorporates onboard calibration circuitry to individually adjust the gain and offset for each analog output channel. The startup calibration process is accomplished through means of retrieving constants stored in EEPROM on the AT-MIO-16F-5 and writing them to the calibration DAC. The board is calibrated at the factory and these calibration values are stored in unmodifiable locations in the EEPROM (see Figure 5-1). The board can also be recalibrated at the user's discretion and these new calibration constants can be stored in one of five user slots in the EEPROM. The EEPROM constants written to the calibration DAC can either be factory-calibrated values, or user-defined values to accommodate differing testing situations. A map of the EEPROM locations can be found in Chapter 5, *Calibration Procedures.*

The DAC output op-amps can be jumper-configured to generate either a unipolar voltage output or a bipolar voltage output range. A unipolar output has an output voltage range of 0 to $+V_{ref}$ - 1 LSB V. A bipolar output has an output voltage range of $-V_{ref}$ to $+V_{ref}$ -1 LSB V. For unipolar output, 0 V output corresponds to a digital code word of 0. For bipolar output, the form of the digital code input is software-selectable from Command Register 2. If straight binary form is selected, 0 V output corresponds to a digital code word of 2048. If two's complement form is selected, 0 V output corresponds to a digital code word of 0. One LSB is the voltage increment corresponding to an LSB change in the digital code word. For unipolar output, 1 LSB = $(V_{ref})/4096$. For bipolar output, 1 LSB = $(V_{ref})/2048$.

The voltage reference source for each DAC is jumper-selectable and can be supplied either externally at the EXTREF input or internally. The external reference can be either a DC or an AC signal. If an AC reference is applied, the analog output channel acts as a signal attenuator, and the AC signal appears at the output attenuated by the digital code divided by 4096 (unipolar output) or 2048 (bipolar output). The internal reference is an amplified version of the internal 5 V signal supplied in the input offset section. Using the internal reference supplies an output voltage range of 0 V to 9.9976 V in steps of 2.44 mV for unipolar output and an output voltage range of -10 V to +9.9951 V in steps of 4.88 mV for bipolar output. Gain calibration for the DACs applies only to the internal reference, not the external reference. Offset calibration can be applied to both references.

Note: Each DAC presents an impedance of 11 kΩ (unipolar mode) or 7 kΩ (bipolar mode) to ground at the EXTREF input when the external reference option is selected.

Digital I/O Circuitry

The AT-MIO-16F-5 has eight digital I/O lines. These lines are divided into two ports of four lines each and are located at pins ADIO<3..0> and BDIO<3..0> on the I/O connector. Figure 3-5 shows a block diagram of the digital I/O circuitry.

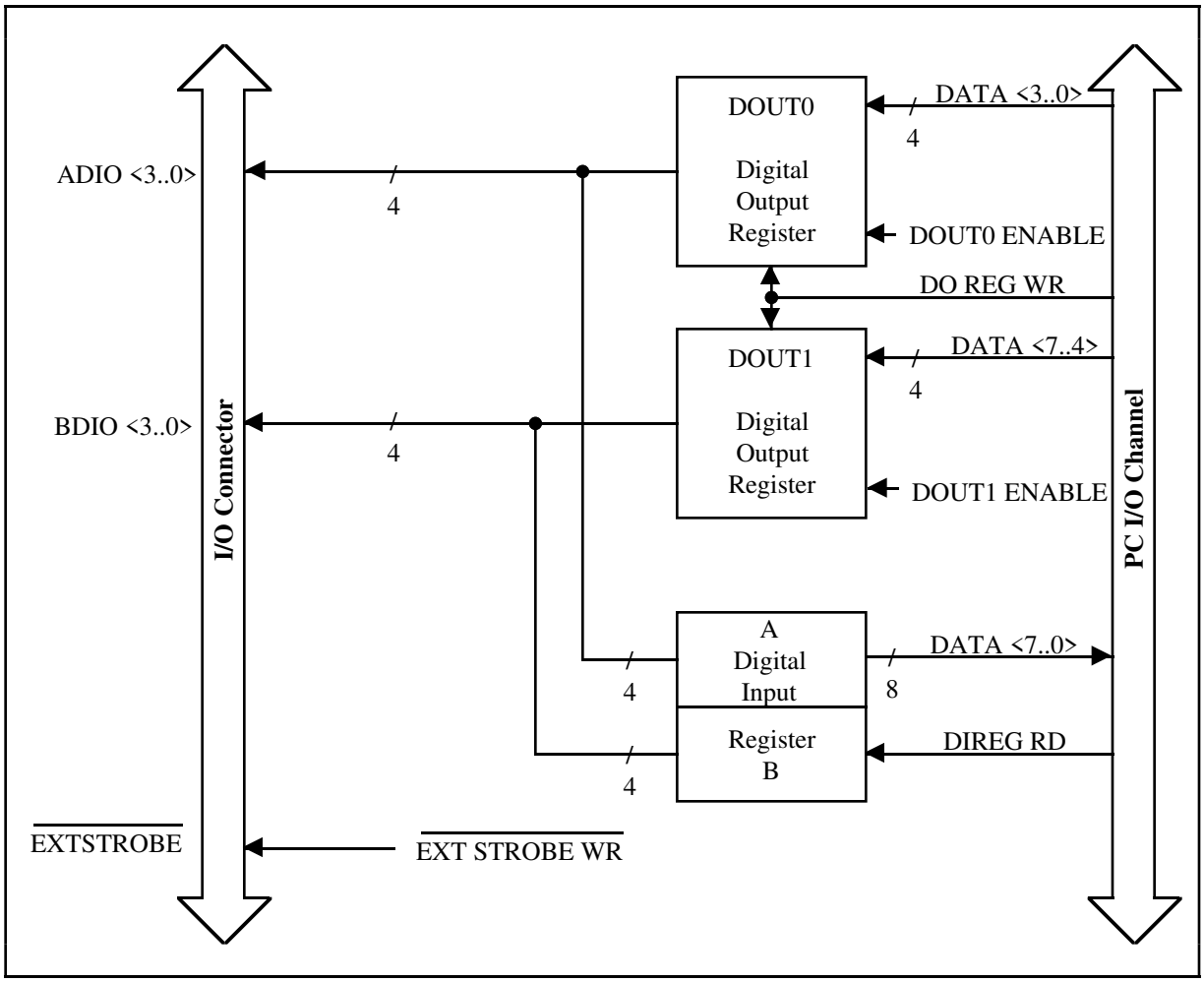


Figure 3-5. Digital I/O Circuitry Block Diagram

The digital I/O lines are controlled by the Digital Output Register and monitored by the Digital Input Register. The Digital Output Register is an 8-bit register that contains the digital output values for both ports 0 and 1. When port 0 is enabled, bits <3..0> in the Digital Output Register are driven onto digital output lines ADIO<3..0>. When port 1 is enabled, bits <7..4> in the Digital Output Register are driven onto digital output lines BDIO<3..0>.

Reading the Digital Input Register returns the state of the digital I/O lines. Digital I/O lines ADIO<3..0> are connected to bits <3..0> of the Digital Input Register. Digital I/O lines BDIO<3..0> are connected to bits <7..4> of the Digital Input Register. When a port is enabled, the Digital Input Register serves as a read-back register, returning the digital output value of the port. When a port is not enabled, reading the Digital Input Register returns the state of the digital I/O lines driven by an external device.

Both the digital input and output registers are TTL-compatible. The digital output ports, when enabled, are capable of sinking 24 mA of current and sourcing 2.6 mA of current on each digital I/O line. When the ports are not enabled, the digital I/O lines act as high-impedance inputs.

The external strobe signal EXTSTROBE*, shown in Figure 3-5, is a general-purpose strobe signal. Writing to an address location on the AT-MIO-16F-5 board generates an active low

200 nsec pulse on this output pin. EXTSTROBE* is not necessarily part of the digital I/O circuitry but is shown here because it can be used to latch digital output from the AT-MIO-16F-5 into an external device.

Timing I/O Circuitry

The AT-MIO-16F-5 uses an Am9513A Counter/Timer for data acquisition timing and for general-purpose timing I/O functions. An onboard oscillator is used to generate the 10-MHz clock. Figure 3-6 shows a block diagram of the timing I/O circuitry.

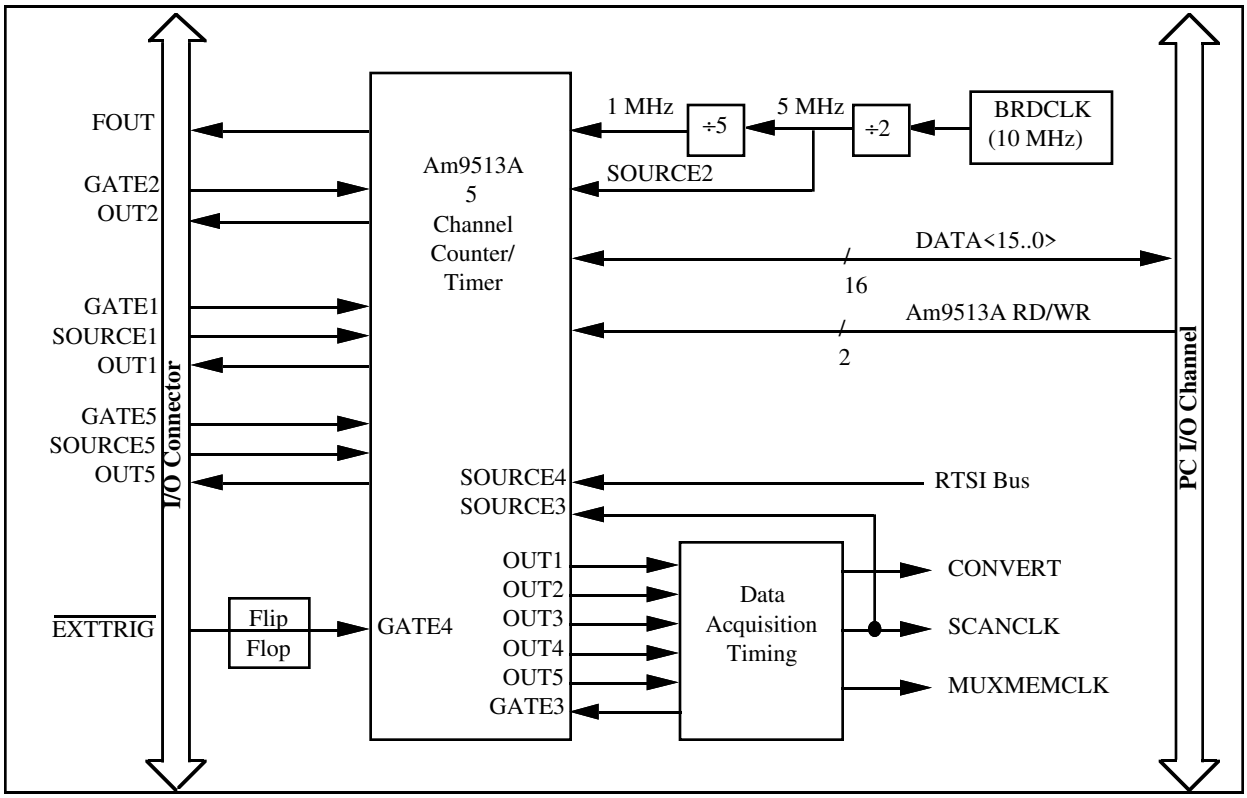


Figure 3-6. Timing I/O Circuitry Block Diagram

The Am9513A contains five, independent, 16-bit counter/timers, a 4-bit frequency output channel, and five internally-generated timebases. The five counter/timers can be programmed to operate in several useful timing modes. The programming and operation of the Am9513A are presented in detail in Appendix C, *AMD Data Sheet*.

The Am9513A clock input is one-tenth the BRDCLK frequency selected by the W1 and W2 jumpers. The factory default for BRDCLK is 10 MHz, which generates a 1-MHz clock input to the Am9513A. The Am9513A uses this clock input plus a BRDCLK divided-by-two input at Source 2 to generate six internal timebases. These timebases can be used as clocks by the counter/timers and by the frequency output channel. When BRDCLK is 10 MHz, the six internal timebases normally used for AT-MIO-16F-5 timing functions are 5 MHz, 1 MHz, 100 kHz,

10 kHz, 1 kHz, and 100 Hz. The 16-bit counters in the Am9513A can be diagrammed as shown in Figure 3-7.

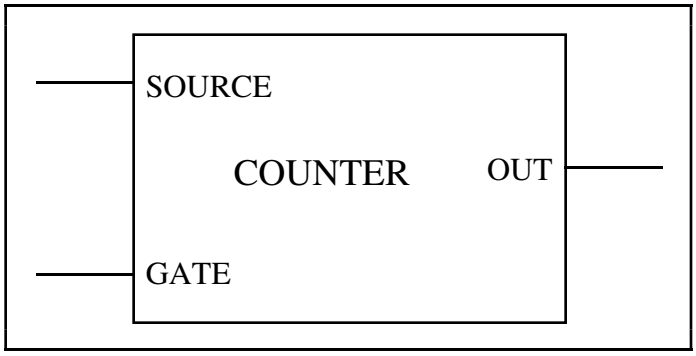


Figure 3-7. Counter Block Diagram

Each counter has a SOURCE input pin, a GATE input pin, and an output pin labeled OUT. The Am9513A counters are numbered 1 through 5, and their GATE, SOURCE, and OUT pins are labeled GATE *N*, SOURCE *N*, and OUT *N*, where *N* is the counter number.

For counting operations, the counters can be programmed to use any of the five internal timebases, any of the five GATE and five SOURCE inputs to the Am9513A, and the output of the previous counter (Counter 4 uses Counter 3 output, and so on). A counter can be configured to count either falling or rising edges of the selected input.

With the counter GATE input, counter operation can be gated. Once a counter is configured for an operation through software, a signal at the GATE input can be used to start and stop counter operation. The five gating modes available with the Am9513A are as follows:

- No gating
- Level gating active high
- Level gating active low

- Low-to-high edge gating
- High-to-low edge gating

A counter can also be active high level gated by a signal at GATE *N*+1 and GATE *N*-1, where *N* is the counter number.

The counter generates timing signals at its OUT output pin. The OUT output pin can also be set to a high-impedance state or a grounded output state. The counters generate two types of output signals during counter operation: terminal count pulse output and terminal count toggle output. Terminal count is often referred to as TC. A counter reaches TC when it counts up or down and rolls over. In many counter applications, the counter reloads from an internal register when it reaches TC. In TC pulse output mode, the counter generates a pulse during the cycle that it reaches TC and reloads. In TC toggle output mode, the counter output changes state after it reaches TC and reloads. In addition, the counters can be configured for positive logic output or negative (inverted) logic output for a total of four possible output signals generated for one timing mode.

The GATE and OUT pins for Counters 1, 2, and 5 and SOURCE pins for Counters 1 and 5 of the onboard Am9513A are located on the AT-MIO-16F-5 I/O connector. A falling edge signal on the EXTTRIG* pin of the I/O connector or writing to the STARTDAQ register sets the flip-flop output signal connected to the GATE4 input of the Am9513A and can be used as an additional gate input. This mode is also used in the pretrigger data acquisition mode. The flip-flop output connected to GATE4 is cleared when the sample counter reaches TC, when an overflow or overrun occurs, or when the A/D Clear Register is written to. An overrun is defined as an error generated when the ADC cannot keep up with the conversion speed it was programmed for.

The Am9513A SOURCE5 pin is connected to the AT-MIO-16F-5 RTSI switch, which means that a signal from the RTSI trigger bus can be used as a counting source for the Am9513A counters.

The Am9513A OUT1, OUT2, and OUT5 pins can be used in several different ways. If waveform generation is enabled, an active low pulse on the output of the counter selected through the RTSI switch updates the analog output on the two DACs. The counter outputs can also be used to trigger interrupt and DMA requests. If the proper mode is selected in Command Register 2, an interrupt or DMA request occurs when a falling edge signal is detected on the selected DAC update signal.

Counters 3 and 4 of the Am9513A are dedicated to data acquisition timing, and therefore are not available for general-purpose timing applications. Signals generated at OUT3 and OUT4 are sent to the data acquisition timing circuitry. GATE3 is controlled by the data acquisition timing circuitry. OUT3 is internally connected to EXTCONV* so that when internal data acquisition sequences (OUT3) are used, EXTCONV* should be disconnected or tri-stated. For the same reason, if external data acquisition sequences (EXTCONV*) are used, OUT3 should be programmed to the high-impedance state.

Counter 5 is sometimes used by the data acquisition timing circuitry and concatenated with Counter 4 to form a 32-bit sample counter. The SCANCLK signal is connected to the SOURCE3 input of the Am9513A, and OUT1 is sent to the data acquisition timing circuitry. This allows Counter 1 to be used to divide the SCANCLK signal for generating the MUXMEMCLK signal (see *Data Acquisition Timing Circuitry* earlier in this chapter).

Counter 2 is sometimes used by the data acquisition timing circuitry to assign a time interval to each cycle through the scan sequence programmed in the mux-channel-gain memory. This mode is called interval channel scanning. See *Multiple-Channel (Scanned) Data Acquisition* earlier in this chapter.

The Am9513A 4-bit programmable frequency output channel is located at the I/O connector FOUT pin. Any of the five internal timebases and any of the counter SOURCE or GATE inputs can be selected as the frequency output source. The frequency output channel divides the selected source by its 4-bit programmed value and makes the divided down signal available at the FOUT pin.

RTSI Bus Interface Circuitry

The AT-MIO-16F-5 is interfaced to the National Instrument RTSI bus. The RTSI bus has seven trigger lines and a system clock line. All National Instruments AT Series boards with RTSI bus connectors can be wired together inside the PC and share these signals. A block diagram of the RTSI bus interface circuitry is shown in Figure 3-8.

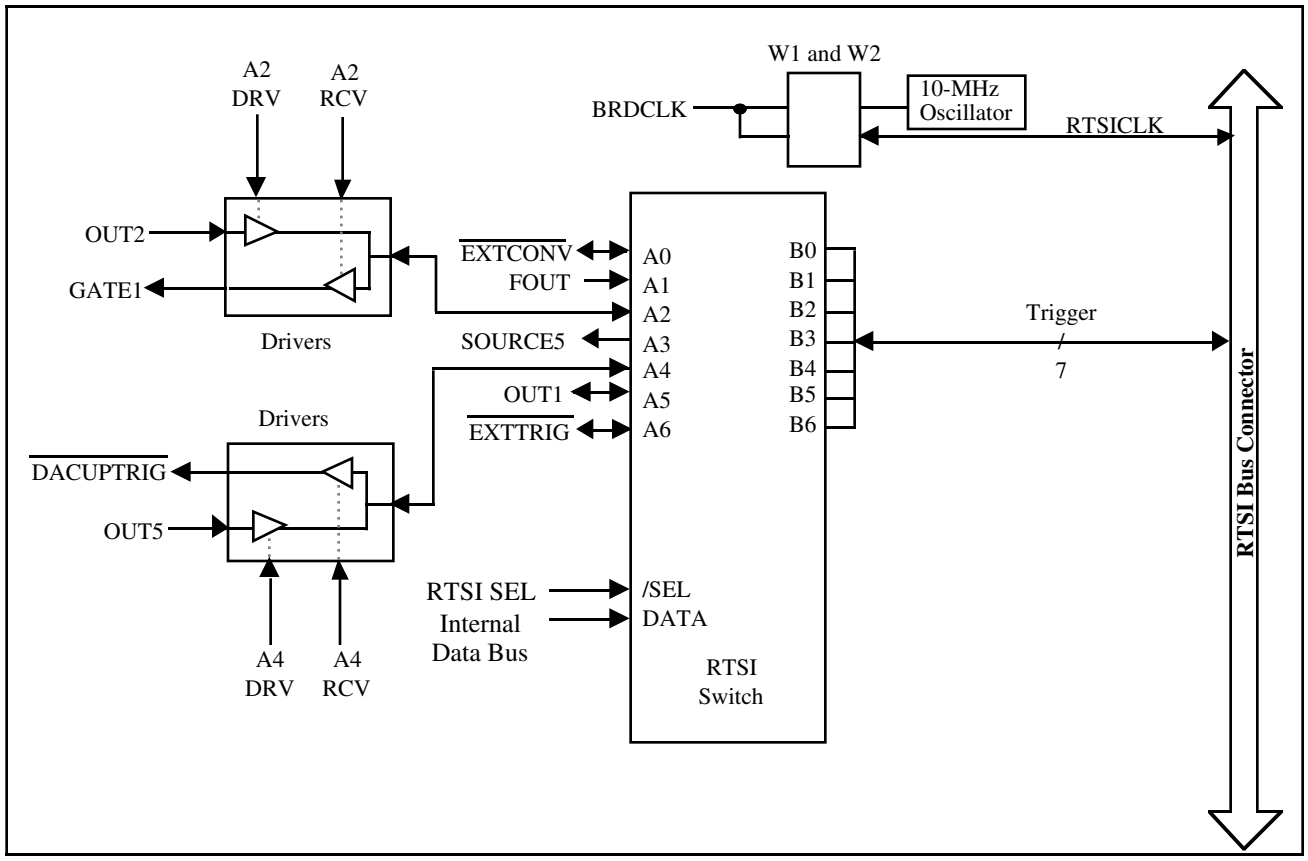


Figure 3-8. RTSI Bus Interface Circuitry Block Diagram

The RTSICLK line can be used to source a 10-MHz signal across the RTSI bus or to receive another clock signal from another AT board connected to the RTSI bus. BRDCLK is the system clock used by the AT-MIO-16F-5. The W1 and W2 jumpers select how these clock signals are routed.

The RTSI switch is a National Instruments custom-integrated circuit that acts as a 7x7 crossbar switch. Pins B<6..0> are connected to the seven RTSI bus trigger lines. Pins A<6..0> are connected to seven signals on the board. The RTSI switch can drive any of the signals at pins A<6..0> onto any one or more of the seven RTSI bus trigger lines and can drive any of the seven trigger line signals onto any one or more of the pins A<6..0>. With this capability, a signal interconnection scheme is completely flexible for any AT Series board sharing the RTSI bus. The RTSI switch is programmed via its chip select and data inputs.

On the AT-MIO-16F-5 board, nine signals are connected to pins A<6..0> of the RTSI switch with the aid of additional drivers. The signals GATE1, OUT1, OUT2, SOURCE5, OUT5, and FOUT are shared with the AT-MIO-16F-5 I/O connector and Am9513A Counter/Timer. The EXTCONV* and EXTTRIG* signals are shared with the I/O connector and the data acquisition timing circuitry. The DACUPTRIG* signal is used to update the two DACs on the AT-MIO-16F-5. With these onboard interconnections, AT-MIO-16F-5 general-purpose and data acquisition timing can be controlled over the RTSI bus as well as externally, and the AT-MIO-16F-5 and the I/O connector can send timing signals to other AT boards connected to the RTSI bus.

Chapter 4
Programming

This chapter describes in detail the address and function of each of the AT-MIO-16F-5 registers. This chapter also includes important information about programming the AT-MIO-16F-5.

Note: If you plan to use a programming software package such as NI-DAQ or LabWindows 2.0 with your AT-MIO-16F-5 board, you do not need to read this chapter.

Register Map

The register map for the AT-MIO-16F-5 is shown in Table 4-1. This table gives the register name, the register address, the type of the register (read only, write only, or read and write) and the size of the register in bits.

Table 4-1. AT-MIO-16F-5 Register Map

Register Name	Offset Address (Hex)	Type	Size
Configuration and Status Register Group:			
Command Register 1	Base address + 0	Write only	16-bit
Command Register 2	Base address + 2	Write only	16-bit
Status Register	Base address + 0	Read only	16-bit
Event Strobe Register Group:			
Start Convert Register	Base address + 8	Write only	8-bit
Start DAQ Register	Base address + A	Write only	8-bit
A/D Clear Register	Base address + C	Write only	8-bit
External Strobe Register	Base address + E	Write only	8-bit
DMA TC INT Clear Register	Base address + 16	Write only	16-bit
Analog Output Register Group:			
DAC0 Register	Base address + 10	Write only	16-bit
DAC1 Register	Base address + 12	Write only	16-bit
DAC Update INT Clear Register	Base address + 14	Write only	8-bit
DAC Update Register	Base Address + E	Read only	8-bit

(continues)

Table 4-1. AT-MIO-16F-5 Register Map (continued)

Register Name	Offset Address (Hex)	Type	Size
Analog Input Register Group:			
MUXMEMLD Register	Base address + 4	Write only	8-bit
MUXMEMCLR Register	Base address + 5	Write only	8-bit
MUXMEM Register	Base address + 6	Write only	16-bit
A/D FIFO Register	Base address + 16	Read only	16-bit
Counter/Timer (Am9513A) Register Group:			
Am9513A Data Register	Base address + 18	Read and write	16-bit
Am9513A Command Register	Base address + 1A	Write only	16-bit
Am9513A Status Register	Base address + 1A	Read only	16-bit
Digital I/O Register Group:			
Digital Input Register	Base address + 1C	Read only	16-bit
Digital Output Register	Base address + 1C	Write only	16-bit
RTSI Switch Register Group:			
RTSI Switch Shift Register	Base address + 1E	Write only	8-bit
RTSI Switch Strobe Register	Base address + 1F	Write only	8-bit

Register Sizes

Two different transfer sizes for read and write operations are available on the PC: byte (8-bit) and word (16-bit). Table 4-1 shows the size of each AT-MIO-16F-5 register. For example, reading the A/D FIFO Register requires a 16-bit (word) read operation at the selected address, whereas writing to the RTSI Strobe Register requires an 8-bit (byte) write operation at the selected address.

Register Description

Table 4-1 divides the AT-MIO-16F-5 registers into seven different register groups. A bit description of each of the registers making up these groups is included later in this chapter.

The Configuration and Status Register Group controls the overall operation of the AT-MIO-16F-5 hardware. The Event Strobe Register Group is a group of registers that, when written to, generate some event on the AT-MIO-16F-5 board. The registers in the Analog Output Register Group access the AT-MIO-16F-5 DACs. With the Analog Input Register Group, ADC output can be read. The Counter/Timer Register Group consists of the three registers of the onboard Am9513A Counter/Timer chip. The registers in the Digital I/O Register Group access the onboard digital input and output lines. The registers in the RTSI Switch Register Group control the onboard RTSI switch.

Register Description Format

The remainder of this register description chapter discusses each of the AT-MIO-16F-5 registers in the order shown in Table 4-1. Each register group is introduced, followed by a detailed bit description of each register. The individual register description gives the address, type, word size, and bit map of the register, followed by a description of each bit.

The register bit map shows a diagram of the register with the MSB (bit 15 for a 16-bit register, bit 7 for an 8-bit register) shown on the left, and the LSB (bit 0) shown on the right. A square is used to represent each bit. Each bit is labeled with a name inside its square. An asterisk (*) after the bit name indicates that the bit is inverted (negative logic).

In many of the registers, several bits are labeled with an X, indicating *don't care* bits. When a register is read, these bits may appear set or cleared but should be ignored because they have no significance.

The bit map field for some write-only registers states *not applicable, no bits used*. Writing to these registers generates a strobe in the AT-MIO-16F-5. These strobes are used to cause some onboard event to occur. For example, they can be used to clear the analog input circuitry or to start a data acquisition operation. The data is ignored when writing to these registers; therefore, any bit pattern suffices.

Configuration and Status Register Group

General control and monitoring of the AT-MIO-16F-5 hardware is accomplished through use of three registers making up the Configuration and Status Register Group. Command Registers 1 and 2 contain bits that control operation of several different pieces of the AT-MIO-16F-5 hardware. The Status Register can be used to read the state of different pieces of the AT-MIO-16F-5 hardware.

Bit descriptions of the three registers making up the Configuration and Status Group are given on the following pages.

Command Register 1

Command Register 1 contains 16 bits that control AT-MIO-16F-5 serial device access, data acquisition mode selection, and analog input configuration.

Address: Base address + 0 (hex)

Type: Write only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
EEPROMCS	SDATA	SCLK	CALDACLD	0	DITHER	INTGATE	CAL

7	6	5	4	3	2	1	0
DAQEN	SCANEN	SCN2	SCANDIV	16*/32CNT	AIS/AIG	SE/DIFF	UNIP/BIP

Bit	Name	Description
15	EEPROMCS	EEPROM Chip Select – This bit enables and disables the chip select of the on-board EEPROM used to store calibration constants. When EEPROMCS is set, the chip select signal to the EEPROM is enabled. Before EEPROMCS is brought high, SCLK should first be pulsed high to initialize the EEPROM circuitry.
14	SDATA	Serial Data – This bit is used to transmit a single bit to both the EEPROM and the calibration DAC.
13	SCLK	Serial Clock – A low-to-high transition of this bit clocks data into the EEPROM (when EEPROMCS is set) and the calibration DAC. If EEPROMCS is cleared, toggling SCLK does not affect the EEPROM.
12	CALDACLD	Calibration DAC Load – Pulsing CALDACLD high loads the calibration DAC with the bits clocked in by SCLK.
11	0	Reserved Bit – This bit must always be set to zero.
10	DITHER	Dither – When this bit is set, 0.5 LSB of White Gaussian Noise is added to the selected analog input signal. By enabling DITHER and using averaging, greater input resolution can be achieved.
9	INTGATE	Internal Gate – When this bit is set, no A/D conversions take place. INTGATE can be used as a software gating tool, or to inhibit random conversions during setup operations.

Bit	Name	Description
8	CAL	Calibration – This bit controls the analog input configuration switches. CAL is used to disconnect the input multiplexers from the instrumentation amplifier during a calibration procedure so that known internal reference signals can be routed to the amplifier. See Table 4-2.
7	DAQEN	Data Acquisition Enable – This bit enables and disables a data acquisition operation that is controlled by the onboard sample-interval and sample counters. If DAQEN is set, a software (STARTDAQ) or hardware (EXTTRIG*) trigger starts the counters (assuming that the counters are programmed and enabled), thereby initiating a data acquisition operation. If DAQEN is cleared, software and hardware triggers are ignored.
6	SCANEN	Scan Enable – This bit enables and disables multiple-channel scanning during data acquisition. If SCANEN is set, alternate analog input channels are sampled during data acquisition under control of the mux-channel-gain memory. If SCANEN is cleared, a single analog input channel is sampled during the entire data acquisition operation.
5	SCN2	Scan Mode 2 – This bit selects the data acquisition scanning mode used when scanning multiple A/D channels. If SCN2 is cleared and SCANEN is set, continuous channel scanning is used. In this mode, scan sequences are repeated with no delays between cycles. If SCN2 is set and SCANEN is set, interval channel scanning is used. In this mode, scan sequences occur during a programmed time interval, called a *scan interval*. One cycle of the scan sequence occurs during each scan interval.
4	SCANDIV	Scan Divide – This bit enables and disables division of the Mux-Counter Clock during data acquisition. The Mux-Counter Clock controls sequencing of the mux-channel-gain memory. If SCANDIV is set, the Mux-Counter Clock is controlled by Counter 1 of the Am9513A Counter/Timer. If SCANDIV is cleared, the Mux-Counter Clock generates one pulse per conversion.
3	16*/32CNT	16 or 32 bit sample count – This bit selects the count resolution for the number of A/D conversions to be performed in a data acquisition operation. If 16*/32CNT is cleared, a 16-bit count mode is selected and Counter 4 of the Am9513A Counter/Timer controls conversion counting. If 16*/32CNT is set, a 32-bit count mode is selected and Counter 4 is concatenated with Counter 5 to control conversion counting. A 16-bit count mode can be used if the number of A/D sample conversions to be performed is less than 65,537. A 32-bit count mode should be used if the number of A/D sample conversions to be performed is greater than or equal to 65,537.

Bit	Name	Description
2	AIS/AIG	Analog Input Sense/Analog Input Ground – This bit configures the analog input section for RSE or NRSE mode when CAL is disabled. When CAL is enabled, this bit controls the reference signal connected to the positive(+) side of the instrumentation amplifier. See Table 4-2.
1	SE/DIFF	Single-Ended/Differential – This bit configures the analog input section for single-ended or differential mode. See Table 4-2.
0	UNIP/BIP	Unipolar/Bipolar – This bit configures the ADC for unipolar or bipolar mode. In unipolar mode, values read from the A/D FIFO are in straight binary format. For bipolar mode, the FIFO values are automatically sign extended.

Table 4-2. Input Configuration

Input Mode	Bit Map			Effect	
	CAL	SE/DIFF	AIS/AIG	Inst Amp(+)	Inst Amp(-)
DIFF	0	0	*X*	Channels 0 to 7	Channels 8 to 15
RSE	0	1	0	Channels 0 to 15	AIGND
NRSE	0	1	1	Channels 0 to 15	AISENSE
Offset Calibration	1	0	0	AIGND	AIGND
Gain Calibration	1	0	1	Internal +5 V Ref.	AIGND
DAC Offset Calibration	1	1	0	AIGND	Channels 8 to 15
DAC Gain Calibration	1	1	1	Internal +5 V Ref.	Channels 8 to 15

Note: *X* indicates a *don't care* bit.

Command Register 2

Command Register 2 contains 16 bits that control AT-MIO-16F-5 interrupt and DMA modes, digital output drivers, and waveform generation modes.

Address: Base address + 2 (hex)

Type: Write only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
DIOPBEN	DIOPAEN	WGEN	INTEN	DMAEN	DMATCINTEN	CMPLINTEN	DMAINTB2

7	6	5	4	3	2	1	0
DMAINTB1	DMAINTB0	ADCFIFOREQ	RETRIG	A4RCV	A4DRV	A2RCV	A2DRV

Bit	Name	Description
15	DIOPBEN	Digital I/O Port B Enable – This bit enables and disables driving of the 4-bit digital output port B by the Digital Output Register. If DIOPBEN is set, the Digital Output Register drives the digital lines. If DIOPBEN is cleared, the Digital Output Register drivers are set to a high-impedance state; therefore an external device can drive the digital lines.
14	DIOPAEN	Digital I/O Port A Enable – This bit enables and disables driving of the 4-bit digital output port A by the Digital Output Register. If DIOPAEN is set, the Digital Output Register drives the digital lines. If DIOPAEN is cleared, the Digital Output Register drivers are set to a high-impedance state; therefore an external device can drive the digital lines.
13	WGEN	Waveform Generation Enable – This bit selects the update method for the DAC outputs. When WGEN is cleared, both DAC0 and DAC1 are updated when either DAC is written to. If WGEN is set, both DACs are updated when an active low pulse is detected on the DACUPTRIG* or when the DAC Update Register is strobed. With the DACUPTRIG* method, the DACs can be updated by any choice of OUT1, OUT2, OUT5 from the Am9513A counter/timer chip, any other signal from another board connected to the RTSI switch, or the EXTDACUPDATE* signal at the I/O connector.

Bit	Name	Description
12	INTEN	Interrupt Enable – This bit, along with the appropriate mode bits, enables and disables interrupts generated from the AT-MIO-16F-5 board. To generate a specific interrupt, INTEN or a specific interrupt enable bit must be set. See Table 4-3 and *Interrupt Programming* later in this chapter.
11	DMAEN	DMA Enable – This bit enables and disables the generation of DMA requests. DMA requests can be generated from A/D conversions as well as from DAC updates. If DMAEN is cleared, no DMA requests are generated from the AT-MIO-16F-5 board. To generate a specific DMA transfer, DMAEN and a specific DMA enable bit must be set. See Table 4-3 and *Programming DMA Operations* later in this chapter.
10	DMATCINTEN	DMA Terminal Count Interrupt Enable – This bit enables and disables generation of an interrupt when a DMA terminal count pulse is received from the DMA controller in the PC AT. If DMATCINTEN is set, an interrupt request is generated when the DMA controller transfer count register decrements from zero to FFFF (hex). The interrupt request is serviced by writing to the DMA TC INT Clear Register. When DMATCINTEN is cleared, no DMA terminal count interrupts are generated. See also *Programming DMA Operations* later in this chapter.
9	CMPLINTEN	DAQ Complete Interrupt Enable – This bit enables and disables generation of an interrupt when a data acquisition sequence completes. If CMPLINTEN is set, an interrupt request is generated when the data acquisition operation completes. The interrupt request is serviced by writing to the ADCLEAR Register. When CMPLINTEN is cleared, completion of a data acquisition sequence does not generate an interrupt. A data acquisition sequence ends by running its course or when an error condition occurs such as OVERRUN or OVERFLOW.
8	DMAINTB2	DMA Interrupt Bit 2 – see Table 4-3 for mode descriptions. Also see *Programming DMA Operations,* and *Interrupt Programming* later in this chapter.
7	DMAINTB1	DMA Interrupt Bit 1 – see Table 4-3 for mode descriptions. Also see *Programming DMA Operations,* and *Interrupt Programming* later in this chapter.
6	DMAINTB0	DMA Interrupt Bit 0 – see Table 4-3 for mode descriptions. Also see *Programming DMA Operations,* and *Interrupt Programming* later in this chapter.

Bit	Name	Description
5	ADCFIFOREQ	ADC FIFO Request – This bit controls the ADC FIFO Interrupt and DMA Request mode when enabled. When ADCFIFOREQ is cleared, ADC interrupt/DMA requests are generated when a single conversion is in the FIFO. When ADCFIFOREQ is set, ADC interrupt/DMA requests are generated when the ADC FIFO is half-full. In both cases, the request is removed when the ADC FIFO is less than half-full.
4	RETRIG	Retrigger – This bit controls the retrigger method of the acquisition circuitry. When RETRIG is set, the data acquisition circuitry will not retrigger until the DAQCOMP bit in the Status Register is cleared by strobing the ADCLEAR Register. When RETRIG is clear, retriggering may occur at any time after the previous sequence has ended.
3	A4RCV	A4 Receive – This bit controls a driver that allows the DACUPTRIG*(DAC Update Trigger) signal to be driven from pin A4 of the RTSI switch. If A4RCV is set, pin A4 of the RTSI switch drives the DACUPTRIG* signal. If A4RCV is cleared, the DACUPTRIG* signal is driven by the EXTDACUPDATE* signal.
2	A4DRV	A4 Drive – This bit controls a driver that allows the OUT5 signal to drive pin A4 of the RTSI switch. If A4DRV is set, pin A4 of the RTSI switch is driven by OUT5. If A4DRV is cleared, pin A4 is not driven.
1	A2RCV	A2 Receive – This bit controls a driver that allows the GATE1 signal to be driven from pin A2 of the RTSI switch. If A2RCV is set, pin A2 of the RTSI switch drives the GATE1 signal. If A2RCV is cleared, the GATE1 signal is not driven by the RTSI switch.
0	A2DRV	A2 Drive – This bit controls a driver that allows the OUT2 signal to drive pin A2 of the RTSI switch. If A2DRV is set, pin A2 of the RTSI switch is driven by OUT2. If A2DRV is cleared, pin A2 is not driven.

Table 4-3. DMA and Interrupt Modes

Interface Mode					Mode Description
INTEN	DMAEN	DMAINTB2	DMAINTB1	DMAINTB0	
0	0	*X*	*X*	*X*	No interrupts or DMA
0	1	0	0	0	Analog input DMA (Channel A)
0	1	0	0	1	Analog output DMA (Channel A to DAC 0)
0	1	0	1	0	Analog output DMA (Channel A to DAC 1)
0	1	0	1	1	Analog output DMA (Channel A to DACs 0/1)
0	1	1	0	0	Analog input DMA (Channels A/B)
0	1	1	0	1	Analog I/O DMA (Channel A in and Channel B to DAC 0)
0	1	1	1	0	Analog I/O DMA (Channel A in and Channel B to DAC 1)
0	1	1	1	1	Analog I/O DMA (Channel A in and Channel B to DACs 0/1)
1	0	0	0	0	No interrupts or DMA
1	0	0	0	1	ADCFIFO interrupt
1	0	0	1	0	DACUP interrupt
1	0	0	1	1	ADCFIFO and DACUP interrupts
1	0	1	0	0	No interrupts or DMA
1	0	1	0	1	No interrupts or DMA
1	0	1	1	0	No interrupts or DMA
1	0	1	1	1	No interrupts or DMA
1	1	0	0	0	Analog input DMA (Channel A) with DACUP INT
1	1	0	0	1	Analog output DMA (Channel A to DAC 0) with ADCFIFOINT
1	1	0	1	0	Analog output DMA (Channel A to DAC 1) with ADCFIFOINT
1	1	0	1	1	Analog output DMA (Channel A to DACs 0/1) with ADCFIFOINT
1	1	1	0	0	Analog input DMA (Channels A/B) with DACUP INT
1	1	1	0	1	No interrupts or DMA
1	1	1	1	0	Analog output DMA (Channels A/B to DACs 0/1)*
1	1	1	1	1	Analog output DMA (Channels A/B to DACs 0/1) with ADCFIFOINT*

Note: *X* indicates a *don't care* bit.

* In this analog output mode, the channels are defined to be synchronous, that is, they must operate concurrently. If one channel stops, the other channel also stops.

Status Register

The Status Register contains 16 bits of AT-MIO-16F-5 hardware status information, including interrupt, analog input status, and data acquisition progress.

Address: Base address + 0 (hex)

Type: Read only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
DAQCOMP	DAQPROG	FIFOHF*	FIFOEF*	DMATC	DMATCA	DMATCB	OVERFLOW*

7	6	5	4	3	2	1	0
OVERRUN*	DACUP	DACUPERR	FACTCAL*	PROMOUT	EEPROMCD*	MUXMEMFF*	MUXMEMEF*

Bit	Name	Description
15	DAQCOMP	Data Acquisition Complete – This bit reflects the status of the data acquisition termination interrupt. If DAQCOMP is set and either OVERFLOW* or OVERRUN* is clear, the current interrupt is due to an error condition. If DAQCOMP is set and neither OVERFLOW* nor OVERRUN* is clear, the data acquisition operation has completed without error. When DAQCOMP becomes set, if ADCFIFOREQ in Command Register 2 is also set, enabled interrupt or DMA requests are generated until the ADC FIFO is empty. DAQCOMP is cleared by writing to the A/D Clear Register (ADCLEAR).
14	DAQPROG	Data Acquisition Progress – This bit indicates whether a data acquisition operation is in progress. If DAQPROG is set, a data acquisition operation is in progress. If DAQPROG is cleared, the data acquisition operation has completed.
13	FIFOHF*	FIFO Half-Full Flag – This bit reflects the state of the A/D FIFO. If the appropriate conversion interrupts are enabled (see Table 4-3) and FIFOHF* is clear, the current interrupt indicates at least 128 A/D conversions are available in the A/D FIFO. To clear the interrupt, read the A/D FIFO until FIFOHF* becomes set. If FIFOHF* is set, less than 128 A/D conversions are available in the A/D FIFO.

Bit	Name	Description
12	FIFOEF*	FIFO Empty Flag – This bit reflects the state of the A/D FIFO. If FIFOEF* is set, one or more A/D conversion results can be read from the A/D FIFO. If the appropriate conversion interrupts are enabled (see Table 4-3) and FIFOEF* is set, the current interrupt indicates that A/D conversion data is available in the A/D FIFO. To clear the interrupt, the FIFO must be read until it is empty. If FIFOEF* is cleared, the A/D FIFO is empty and no conversion interrupt request is asserted.
11	DMATC	DMA Terminal Count – This bit is high if either DMATCA, DMATCB, or both are high.
10	DMATCA	DMA Terminal Count Channel A – DMATCA reflects the status of the DMA process on the selected DMA Channel A. When the DMA operation is finished, DMATCA goes high and remains high until cleared using the DMA TC INT Clear Event Strobe Register. If DMATCINTEN is set, an interrupt is generated when DMATCA goes high.
9	DMATCB	DMA Terminal Count Channel B – DMATCB reflects the status of the DMA process on the selected DMA Channel B. When the DMA operation is finished, DMATCB goes high and remains high until cleared using the DMA TC INT Clear Event Strobe Register. If DMATCINTEN is set, an interrupt is generated when DMATCB goes high.
8	OVERFLOW*	Overflow – This bit indicates whether the A/D FIFO has overflowed during a sample run. OVERFLOW* is an error condition that occurs if the FIFO fills up with A/D conversion data and A/D conversions continue. If OVERFLOW* is clear, A/D conversion data has been lost because of FIFO overflow. If OVERFLOW* is set, no overflow has occurred. If OVERFLOW* occurs during a data acquisition operation, the data acquisition is terminated immediately. This bit can be reset by writing to the A/D Clear Register.
7	OVERRUN*	Overrun – This bit indicates whether an A/D conversion was initiated before the previous A/D conversion was complete. OVERRUN* is an error condition that can occur if the data acquisition sample interval is too small (sample rate is too high). If OVERRUN* is clear, one or more conversions were skipped. If OVERRUN* is set, no overrun condition has occurred. If OVERRUN* occurs during a data acquisition operation, the data acquisition is terminated immediately. This bit can be reset by writing to the A/D Clear Register.

Bit	Name	Description
6	DACUP	DAC Update – This bit reflects the status of the DAC update. DACUP is cleared by writing to the DAC Update INT Clear Register. DACUP is set whenever a falling edge on DACUPTRIG* at the RTSI switch is detected. This condition generates an interrupt or DMA request only if the proper interrupt mode is selected in Command Register 1 according to Table 4-3.
5	DACUPERR	DAC Update Error – This bit reflects an error condition during DAC waveform generation operations to the analog output circuitry (WGEN set in Command Register 2). If an update (DACUPTRIG*) occurs before a new value can be written to the DAC, DACUPERR is set indicating a value being updated to the DAC twice.
4	FACTCAL*	Factory Calibration – When this bit is set, the board has been factory-calibrated and the calibration constants have been written to the upper 11 locations in the EEPROM. Any further attempt to modify these locations has no effect. If FACTCAL* is low, then all locations of the EEPROM may be written to. (This mode is not recommended, as locations 0 through 51 are available with FACTCAL* set.)
3	PROMOUT	EEPROM Output – This bit reflects the value of the data shifted out of the EEPROM using SCLK with EEPROMCS enabled.
2	EEPROMCD*	EEPROM Chip Deselect – This reflects the status of the EEPROM chip select pin. Because protection circuitry surrounds the EEPROM, having EEPROMCS enabled in Command Register 1 does not necessarily result in the EEPROM being enabled. If EEPROMCD* is low after a mode has been shifted into the EEPROM, then an error occurred in shifting in an unsupported mode. To initialize EEPROMCD*, EEPROMCS must be brought low while SCLK is pulsed high.
1	X	Don't care bits.
0	MUXMEMEF*	MUX Memory Empty Flag – If this bit is clear, the mux-channel-gain memory is empty and can be written to. If MUXMEMEF* is set, the mux-channel-gain memory is not empty.

The Event Strobe Register Group

The Event Strobe Register Group consists of five registers that, when written to, cause the occurrence of certain events on the AT-MIO-16F-5 board, such as clearing flags and starting A/D conversions.

Bit descriptions of the five registers making up the Event Strobe Register Group are given on the following pages.

Start Convert Register

Writing to the Start Convert Register location initiates an A/D conversion.

Address: Base address + 8 (hex)

Type: Write only

Word Size: 8-bit

Bit Map: Not applicable, no bits used

Note: A/D conversions can be initiated in one of two ways: by writing to the Start Convert Register or by detecting an active-low signal on the EXTCONV* signal. The EXTCONV* signal is connected to pin 40 on the I/O connector, to OUT3 of the Am9513A, and to the A0 pin of the RTSI bus switch. If EXTCONV* is driven low by any one of these sources, it prevents the Start Convert Register from initiating an A/D conversion. If the Start Convert Register is to initiate A/D conversions, the OUT3 signal should be initialized to a high-impedance state, any signal connected to pin 40 of the I/O connector should be in a high-impedance or high state, and the A0 pin of the RTSI bus switch should be configured as an input pin.

Start DAQ Register

Writing to the Start DAQ Register location initiates a multiple A/D conversion data acquisition operation.

Note: Several other pieces of AT-MIO-16F-5 circuitry must be set up before a data acquisition run can occur. See *Programming Considerations* later in this chapter.

Address: Base address + A (hex)

Type: Write only

Word Size: 8-bit

Bit Map: Not applicable, no bits used

Note: Multiple A/D conversion data acquisition operations can be initiated in one of two ways: by writing to the Start DAQ Register or by detecting an active-low signal on the EXTTRIG* signal. The EXTTRIG* signal is connected to pin 38 on the I/O connector and to the A6 pin of the RTSI bus switch. If EXTTRIG* is driven low by either of these sources, it prevents the Start DAQ Register from initiating a multiple A/D conversion data acquisition operation. If the Start DAQ Register is to initiate multiple A/D conversions, any signal connected to pin 38 of the I/O connector should be in a high-impedance or high state and the A6 pin of the RTSI bus switch should not be driven low.

A/D Clear Register

Writing to the A/D Clear Register location clears the data acquisition circuitry. Writing to the A/D Clear Register initiates the following events:

- Any data acquisition operation in progress is canceled.
- The A/D FIFO is emptied.
- The overrun flag is cleared.
- The overflow flag is cleared.
- Any pending FIFOEF* interrupt is cleared.
- The DAQCOMP bit in the Status Register is cleared.
- The mux-channel-gain memory is reset to start at the beginning of the list.

Address: Base address + C (hex)

Type: Write only

Word Size: 8-bit

Bit Map: Not applicable, no bits used

Note: If the mux-channel-gain memory already contains valid information and no new values are to be added before restarting the data acquisition sequence, the MUXMEMLD Register should be strobed following an ADCLEAR strobe.

External Strobe Register

Writing to the External Strobe Register location generates an active low, approximately 200 to 500 nsec strobe pulse at the EXTSTROBE output at the I/O connector. This pulse can be useful for several applications, including generating external general-purpose triggers and latching data into external devices (for example, from the digital output port).

Address: Base address + E (hex)

Type: Write only

Word Size: 8-bit

Bit Map: Not applicable, no bits used

DMA TC INT Clear Register

Writing to the DMA TC INT Clear Register clears the interrupt request asserted when a DMA terminal count pulse is detected. DMA TC INT Clear also clears both DMATCA and DMATCB in the Status Register.

Address: Base address + 16 (hex)

Type: Write only

Word Size: 16-bit

Bit Map: Not applicable, no bits used

Analog Output Register Group

The registers making up the Analog Output Register Group load the two analog output channels. DAC0 controls analog output Channel 0. DAC1 controls analog output Channel 1. These DACs are written to individually, and the analog output can be updated immediately or each time an active low pulse is detected on the DACUPTRIG* signal or the DAC Update Register is strobed with the WGEN bit set in Command Register 2.

Bit descriptions of the four registers making up the Analog Output Register Group are given on the following pages.

DAC0 Register

Writing to DAC0 loads the corresponding analog output channel DAC. The voltages generated by the analog output channels are updated either immediately or when an active low pulse occurs on DACUPTRIG* or the DAC Update Register is strobed. The update method is selected by the WGEN bit in Command Register 2.

Address: Base address + 10 (hex) loads DAC0

Type: Write only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
				MSB											LSB

Bit	Name	Description
15 - 12	X	Don't care bits.
11 - 0	D<11..0>	These twelve bits are loaded into the DAC and update the voltage generated by the analog output channel in one of three ways, immediately, upon a DACUPTRIG* pulse, or with a strobe of the DAC Update Register. See Table 4-6 and Table 4-7, both of which map digital values to output voltage.

DAC1 Register

Writing to DAC1 loads the corresponding analog output channel DAC. The voltages generated by the analog output channels are updated either immediately or when an active low pulse occurs on DACUPTRIG* or the DAC Update Register is strobed. The update method is selected by the WGEN bit in Command Register 2.

Address: Base address + 12 (hex) loads DAC1

Type: Write only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
				MSB											LSB

Bit	Name	Description
15 - 12	X	Don't care bits.
11 - 0	D<11..0>	These twelve bits are loaded into the DAC and update the voltage generated by the analog output channel in one of three ways, immediately, upon a DACUPTRIG* pulse, or with a strobe of the DAC Update Register. See Table 4-6 and Table 4-7, both of which map digital values to output voltage.

DAC Update INT Clear Register

Writing to the DAC Update INT Clear Register clears the DACUP and DACUPERR bits after a DACUPTRIG* pulse is detected. Clearing DACUP when interrupt or DMA mode is enabled clears the respective interrupt or DMA request.

Address: Base address + 14 (hex)

Type: Write only

Word Size: 8-bit

Bit Map: Not applicable, no bits used

The analog output DACs can be updated internally and externally in the waveform generation mode (WGEN set in Command Register 2) through the control of A4RCV. If A4RCV is enabled, internal updating is selected and any signal from the RTSI switch controls the updating interval. If OUT2 is to be used for updating the DACs, then A2DRV must also be enabled. If OUT5 is to be used, then A4DRV must be enabled as well. If A4RCV is disabled, then external updating is selected and the EXTDACUPDATE* signal from pin 44 of the I/O connector is used for updating.

In both cases, a falling edge on the selected signal triggers the updating mechanism. This trigger also sets the DACUP bit in the Status Register and generates an interrupt or DMA request if so enabled.

DAC Update Register

Reading from the DAC Update Register with waveform generation enabled updates both DAC0 and DAC1 simultaneously with the previously written values.

Address: Base address + E (hex)

Type: Read only

Word Size: 8-bit

Bit Map: Not applicable, no bits used.

Analog Input Register Group

The four registers making up the Analog Input Register Group control the analog input circuitry and can be used to read the A/D FIFO.

Bit descriptions for the registers making up the Analog Input Register Group are given on the following pages.

MUXMEMLD Register

Writing to the MUXMEMLD Register loads the mux-channel-gain memory.

Address: Base address + 4 (hex)

Type: Write only

Word Size: 8-bit

Bit Map: Not applicable, no bits used.

Writing to the MUXMEMLD Register loads the mux-channel-gain memory values and applies the first channel-gain value to the analog input circuitry. After the final write to the mux-channel-gain memory, writing to the MUXMEMLD Register loads the first channel-gain value. Writing to the MUXMEMLD Register again loads the second channel-gain value, and so on.

Strobing the ADCLEAR Register resets the mux-channel-gain memory to the first value, but does not load the value. It does not clear the memory of any values written to it prior to the ADCLEAR strobe. After an ADCLEAR strobe, the MUXMEMLD Register should be strobed to load the first value. Using this method, a scanned data acquisition can be initiated from any location in the mux-channel-gain memory.

MUXMEMCLR Register

Writing to the MUXMEMCLR Register clears all information in the mux-channel-gain memory.

Address: Base address + 5 (hex)

Type: Write only

Word Size: 8-bit

Bit map: Not applicable, no bits used.

Before the mux-channel-gain memory is written to, it must be cleared of its information and reset to its initialized state. Writing to the MUXMEMCLR Register accomplishes this. Once this operation occurs, old channel-gain values are cleared and not recoverable. At this point, the mux-channel-gain memory is ready to be filled with valid information.

MUXMEM Register

The MUXMEM Register controls the multiplexer and gain settings, and can contain up to 512 channel and gain settings for use in scanning sequences.

Address: Base address + 6 (hex)

Type: Write only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	GHOST_CONV

7	6	5	4	3	2	1	0
MA3	MA2	MA1	MA0	GAIN2	GAIN1	GAIN0	LASTONE

Bit	Name	Description
15 - 9	0	These bits should be left clear for proper board operation.
8	GHOST_CONV	This bit is used to synchronize conversions in time for multiple rate channel scanning. When this bit is set with any channel/gain value, the conversion occurs on the selected channel, but the value is not saved in the A/D FIFO. In addition, if the sample counter is programmed to count samples from Source 4, conversions with the GHOST_CONV bit set are not counted. When the GHOST_CONV bit is clear, conversions occur normally and are saved in the A/D FIFO.
7 - 4	MA<3..0>	This 4-bit field controls the multiplexer address setting of the input multiplexers, thereby allowing the analog input channel to be selected. In single-ended mode, only one analog input channel is selected. In differential mode, two analog input channels are selected. The following table shows the analog input channel selected for either mode.

MA<3..0>	Selected Analog Input Channels	
	Single-Ended	**Differential**
		(+) (-)
0000	0	0 and 8
0001	1	1 and 9
0010	2	2 and 10
0011	3	3 and 11
0100	4	4 and 12
0101	5	5 and 13
0110	6	6 and 14
0111	7	7 and 15
1000	8	0 and 8
1001	9	1 and 9
1010	10	2 and 10
1011	11	3 and 11
1100	12	4 and 12
1101	13	5 and 13
1110	14	6 and 14
1111	15	7 and 15

3 - 1 GAIN<2..0> This 3-bit field controls the gain setting of the input instrumentation amplifier. The following gains can be selected on the AT-MIO-16F-5 board:

GAIN<2..0>	Actual Gain
000	0.5
001	1
010	2
011	5
100	10
101	20
110	50
111	100

0 LASTONE This bit should be set in the last entry of the scan sequence loaded into the mux-channel-gain memory. More than one occurrence of the LASTONE bit is possible in the mux-channel-gain memory list for the interval scanning mode. In other words, there can be multiple scan sequences in one memory list.

Writing to the mux-channel-gain memory must be preceded with a write to the MUXMEMCLR Register and followed by a write to the MUXMEMLD Register. Writing to the MUXMEM Register following a MUXMEMCLR automatically sequences into the memory list. Writing can continue until the end of the mux-channel-gain list is reached, or the memory becomes full. A full mux memory is indicated by the MUXMEMFF* bit in the Status Register. After the final write to the mux memory, the MUXMEMLD Register should be strobed to load the first channel-gain value. At this point the mux-channel-gain memory is primed and does not need to be accessed again until a new channel-gain sequence is initiated. Conversions, either by EXTCONV* or Counter 3 of the Am9513A Counter/Timer, automatically sequence through the mux memory. When the end of the mux memory is detected, it is reset to the first value in the list. Strobing the ADCLEAR Register also resets the mux memory to the first value in the list without destroying old channel-gain values. A strobe of the MUXMEMLD Register is still necessary to load the first channel-gain value.

A/D FIFO Register

Reading the A/D FIFO Register returns the oldest A/D conversion value stored in the A/D FIFO. Whenever the A/D FIFO is read, the value read is removed from the A/D FIFO, thereby leaving space for another A/D conversion value to be stored. Values are stored into the A/D FIFO by the ADC whenever an A/D conversion is complete.

The A/D FIFO is emptied when all values it contains are read. The Status Register should be read before the A/D FIFO Register is read. If the A/D FIFO contains one or more A/D conversion values, the FIFOEF* bit is set in the Status Register and the A/D FIFO Register can be read to retrieve a value. If the FIFOEF* bit is cleared, the A/D FIFO is empty, in which case reading the A/D FIFO Register returns meaningless information. If the FIFOHF* flag is clear in the Status Register, the A/D FIFO is half-full with conversion data.

The values returned by reading the A/D FIFO Register are available in two different binary formats: straight binary, which generates only positive numbers, or two's complement binary, which generates both positive and negative numbers. The binary format used is determined by the UNIP/BIP bit in Command Register 1. The bit pattern returned for either format is given as follows:

Address: Base address + 16 (hex)

Type: Read only

Word Size: 16-bit

Bit Map: Straight binary mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
				MSB											LSB

Bit	Name	Description
15 - 0	D<15..0>	These bits are the straight binary result of a 12-bit A/D conversion. The most significant four bits are set to 0 in order to return a 16-bit result. Values read, therefore, range from 0 to 4095 decimal (0000 to 0FFF hex). Straight binary mode is useful for unipolar analog input readings because all values that are read reflect a positive polarity input signal.

Bit Map: Two's complement binary mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D11*	D11*	D11*	D11*	D11*	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
				MSB											LSB

Bit	Name	Description
15 - 0	D<15..0>	These bits are the two's complement result of a 12-bit A/D conversion. Bit D11 is inverted and extended out to bits D12 through D15. Values read, therefore, range from -2048 to +2047 decimal (F800 to 7FF hex). Two's complement mode is useful for bipolar analog input readings because the values read reflect the polarity of the input signal.

Am9513A Counter/Timer Register Group

The three registers making up the Am9513A Counter/Timer Register Group access the onboard Counter/Timer. The Am9513A controls onboard data acquisition timing as well as general-purpose timing for the user.

The Am9513A registers described here are the Am9513A Data Register, the Am9513A Command Register, and the Am9513A Status Register. The Am9513A contains 18 additional internal registers. These internal registers are accessed through the Am9513A Data Register. A detailed register description of all Am9513A registers is included in Appendix C, *AMD Data Sheet*.

Bit descriptions for the Am9513A Counter/Timer Register Group registers are given in the following pages.

Am9513A Data Register

With the Am9513A Data Register, any of the 18 internal registers of the Am9513A can be written to or read from. The Am9513A Command Register must be written to in order to select the register to be accessed by the Am9513A Data Register. The internal registers accessed by the Am9513A Data Register are as follows:

- Counter Mode Registers for Counters 1, 2, 3, 4, and 5
- Counter Load Registers for Counters 1, 2, 3, 4, and 5
- Counter Hold Registers for Counters 1, 2, 3, 4, and 5
- The Master Mode Register
- The Compare Registers for Counters 1 and 2

All these registers are 16-bit registers. Bit descriptions for each of these registers are included in the Am9513A data sheet in Appendix C.

Address: Base address + 18 (hex)

Type: Read and write

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Bit	Name	Description
15 - 0	D<15..0>	These 16 bits are loaded into the Am9513A Internal Register currently selected. See Appendix C, *AMD Data Sheet*, for the detailed bit descriptions of the 18 registers accessed through the Am9513A Data Register.

Am9513A Command Register

The Am9513A Command Register controls the overall operation of the Am9513A Counter/Timer and controls selection of the internal registers accessed through the Am9513A Data Register.

Address: Base address + 1A (hex)

Type: Write only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	C7	C6	C5	C4	C3	C2	C1	C0

Bit	Name	Description
15 - 8	1	These bits must always be set when writing to the Am9513A Command Register.
7 - 0	C<7..0>	These eight bits are loaded into the Am9513A Command Register. See Appendix C, *AMD Data Sheet*, for the detailed bit description of the Am9513A Command Register.

Am9513A Status Register

The Am9513A Status Register contains information about the output pin status of each counter in the Am9513A.

Address: Base address + 1A (hex)

Type: Read only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
X	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
X	X	OUT5	OUT4	OUT3	OUT2	OUT1	BYTEPTR

Bit	Name	Description
15 - 6	X	Don't care bits.
5 - 1	OUT<5..1>	Each of these five bits returns the logic state of the associated counter output pin. For example, if OUT4 is set, then the output pin of Counter 4 is at a logic high state.
0	BYTEPTR	This bit represents the state of the Am9513A Byte Pointer Flip-Flop. This bit has no significance for AT-MIO-16F-5 operation because the Am9513A should always be used in 16-bit mode on the AT-MIO-16F-5.

Digital I/O Register Group

The two registers making up the Digital I/O Register Group monitor and control the AT-MIO-16F-5 digital I/O lines. The Digital Input Register returns the digital state of the eight digital I/O lines. A pattern written to the Digital Output Register is driven onto the digital I/O lines when the digital output drivers are enabled (see the description for Command Register 2).

Bit descriptions for the registers making up the Digital I/O Register Group are given on the following pages.

Digital Input Register

The Digital Input Register, when read, returns the logic state of the eight AT-MIO-16F-5 digital I/O lines.

Address: Base address + 1C (hex)

Type: Read only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
X	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
BDIO3	BDIO2	BDIO1	BDIO0	ADIO3	ADIO2	ADIO1	ADIO0

Bit	Name	Description
15 - 8	X	Don't care bits.
7 - 4	BDIO<3..0>	These four bits represent the logic state of the digital lines BDIO<3..0>.
3 - 0	ADIO<3..0>	These four bits represent the logic state of the digital lines ADIO<3..0>.

Digital Output Register

Writing to the Digital Output Register controls the eight AT-MIO-16F-5 digital I/O lines. The Digital Output Register controls both ports A and B. When either digital port is enabled, the pattern contained in the Digital Output Register is driven onto the lines of the digital port.

Address: Base address + 1C (hex)

Type: Write only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
BDIO3	BDIO2	BDIO1	BDIO0	ADIO3	ADIO2	ADIO1	ADIO0

Bit	Name	Description
15 - 8	0	Reserved Bits. These bits must always be set to 0.
7 - 4	BDIO<3..0>	These four bits control the digital lines BDIO<3..0>. The bit DIOPBEN in Command Register 2 must be set for BDO<3..0> to be driven onto the digital lines BDIO<3..0>.
3 - 0	ADIO<3..0>	These four bits control the digital lines ADIO<3..0>. The bit DIOPAEN in Command Register 2 must be set for ADO<3..0> to be driven onto the digital lines ADIO<3..0>.

The RTSI Switch Register Group

With the two registers making up the RTSI Switch Register Group, the AT-MIO-16F-5 RTSI switch can be programmed for routing of signals on the RTSI bus trigger lines to and from several AT-MIO-16F-5 signal lines. The RTSI switch is programmed by shifting a 56-bit routing pattern into the RTSI switch and then loading the internal RTSI Switch Control Register. The routing pattern is shifted into the RTSI switch by writing one bit at a time to the RTSI Switch Shift Register. The RTSI Switch Control Register is then loaded by writing to the RTSI Switch Strobe Register.

Bit descriptions for the registers making up the RTSI Switch Register Group are given on the following pages.

RTSI Switch Shift Register

The RTSI Switch Shift Register is written to in order to load the RTSI switch internal 56-bit Control Register with routing information for switching signals to and from the RTSI bus trigger lines. The RTSI Switch Shift Register is a 1-bit register and must be written to 56 times to shift the 56 bits into the internal register.

Address: Base address + 1E (hex)

Type: Write only

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	RSI

Bit	Name	Description
7 - 1	0	Reserved Bits. These bits must always be set to 0.
0	RSI	The name of this bit stands for RTSI switch serial input. This bit is the serial input to the RTSI switch. Each time the RTSI Switch Shift Register is written to, the value of this bit is shifted into the RTSI switch. See *Programming the RTSI Switch* later in this chapter for more information.

RTSI Switch Strobe Register

The RTSI Switch Strobe Register is written to in order to load the contents of the RTSI Switch Shift Register into the RTSI Switch Control Register, thereby updating the RTSI switch routing pattern. The RTSI Switch Strobe Register is written to after shifting the 56-bit routing pattern into the RTSI Switch Shift Register.

Address: Base address + 1F (hex)

Type: Write only

Word Size: 8-bit

Bit Map: Not applicable, no bits used

Programming Considerations

This section contains programming instructions for operating the circuitry on the AT-MIO-16F-5 board. Programming the AT-MIO-16F-5 involves writing to and reading from the various registers on the board. The programming instructions list the sequence of steps to take. The instructions are language-independent; that is, they instruct you to write a value to a given register, to set or clear a bit in a given register, or to detect whether a given bit is set or cleared without presenting the actual code.

Register Programming Considerations

Several write-only registers on the AT-MIO-16F-5 contain bits that control a number of independent pieces of the onboard circuitry. In the instructions for setting or clearing bits, specific register bits should be set or cleared without changing the current state of the remaining bits in the register. However, writing to these registers affects all register bits simultaneously. You cannot read these registers to determine which bits have been set or cleared in the past; therefore, you should maintain a software copy of the write-only registers. This software copy can then be read to determine the status of the write-only registers. To change the state of a single bit without disturbing the remaining bits, set or clear the bit in the software copy and write the software copy to the register.

Initializing the AT-MIO-16F-5 Board

The AT-MIO-16F-5 hardware must be initialized for the AT-MIO-16F-5 circuitry to operate properly. To initialize the AT-MIO-16F-5 hardware, complete these steps:

1. Write 0 to Command Register 1 (16-bit write).
2. Write 0 to Command Register 2 (16-bit write).
3. Write 0 to the MUXMEMCLR Register (8-bit write).
4. Initialize the Am9513A (see the following section).
5. Write 0 to the A/D Clear Register.
6. Write 0 to the DAC Update INT Clear Register (8-bit write).
7. Write 0 to the DMA TC INT Clear Register (16-bit write).

This sequence leaves the AT-MIO-16F-5 circuitry in the following state:

- DMA disabled.
- All interrupts disabled.
- Dither disabled.

- Calibration disabled.
- Outputs of counter/timers in high-impedance state.
- Analog input circuitry initialized.
- A/D FIFO cleared.

Initializing the Am9513A

Use the following sequence to initialize the Am9513A Counter/Timer. All writes are 16-bit operations. All values are given in hexadecimal.

1. Issue a master reset by writing FFFF to the Am9513A Command Register.
2. Set up Am9513A 16-bit mode by writing FFEF to the Am9513A Command Register.
3. Point to the Am9513A Master Mode Register by writing FF17 to the Am9513A Command Register.
4. Load the master mode value into the Am9513A Master Mode Register by writing F000 to the Am9513A Data Register.
5. To initialize all five counters – for *ctr* = 1 to 5, follow these steps:
    a. Write FF00 + *ctr* to the Am9513A Command Register to select the Counter Mode Register.
    b. Write 0004 to the Am9513A Data Register to store the counter mode value.
    c. Write FF08 + *ctr* to the Am9513A Command Register to select the Counter Load Register.
    d. Write 3 to the Am9513A Data Register to store an inactive count value in the Counter Load Register.
6. Load all counters with their Counter Load Register values by writing FF5F to the Am9513A Command Register.

After this sequence of writes, the Am9513A Counter/Timer is in the following state:

- 16-bit mode is enabled.
- BCD scaler division is selected.
- The FOUT signal is turned off.

- All counter OUT output pins are set to high-impedance output state.
- All counters are loaded with a non-terminal count value.

For additional details concerning the Am9513A Counter/Timer, see Appendix C, *AMD Data Sheet*.

Note: If a data acquisition operation is to be executed *and* Counter 4 of Am9513A is not to be used, then write 0000 to the Am9513A Data Register (instead of 0004) when *ctr* = 4. Writing 0000 to the Am9513A Data Register causes the output of Counter 4 to be low, and therefore prevents improper termination of the data acquisition operation.

Initializing the Analog Output Circuitry

The AT-MIO-16F-5 powers up with the analog output circuitry at an unknown voltage. For most applications, the analog output circuitry should be initialized to 0 V. To do this, write 0 to the DAC Register (16-bit write) for that channel.

Programming the Analog Input Circuitry

Single Conversions using the SCONVERT or EXTCONV* Signal

Programming the analog input circuitry to obtain a single A/D conversion involves the following sequence of steps:

1. Select analog input channel and gain.
2. Initiate an A/D conversion.
3. Read the A/D conversion result.

In addition, it is good practice to initialize the board before beginning a data acquisition operation and then check the Status Register to make sure the board is in a known state.

1. Select analog input channel and gain.

The analog input channel and gain are selected by writing to the MUXMEM Register followed by a write to the MUXMEMLD Register. Bits 7 through 4 select the analog input channel and bits 3 through 1 control the gain. See the MUXMEM Register bit description earlier in this chapter for analog input channel and gain bit patterns. Set up the bits as given in the MUXMEM Register bit description and write to the MUXMEM Register. Remember that the mux memory should be initialized first with a write to the MUXMEMCLR Register.

Once the MUXMEM Register is set up, it needs to be written to only when the analog input channel or gain settings need to be changed.

2. Initiate an A/D conversion.

An A/D conversion can be initiated in one of two ways: a software-generated pulse or a hardware pulse.

To initiate a single A/D conversion through software, write 0 to the SCONVERT Register.

To initiate a single A/D conversion through hardware, apply an active low pulse to the EXTCONV* pin on the AT-MIO-16F-5 I/O connector. See *Data Acquisition Timing Connections* in Chapter 2, *Configuration and Installation*, for EXTCONV* signal specifications.

Once an A/D conversion is initiated, the ADC automatically stores the result in the A/D FIFO at the end of its conversion cycle.

3. Read the A/D conversion result.

A/D conversion results are available when FIFOEF* is set in the Status Register and can be obtained by reading the A/D FIFO Register.

To read the A/D conversion result, use the following steps:

a. Read the Status Register (16-bit read).

b. If the FIFOEF* bit is set (bit 13), read the A/D FIFO Register to obtain the result.

Reading the A/D FIFO Register removes the A/D conversion result from the A/D FIFO and clears the FIFOEF* bit if no more values remain in the FIFO. The binary modes of the A/D FIFO output are explained in *A/D FIFO Output Binary Formats*.

The FIFOEF* bit indicates whether one or more A/D conversion results are stored in the A/D FIFO. If the FIFOEF* bit is not set, the A/D FIFO is empty and reading the A/D FIFO Register returns meaningless data. Once an A/D conversion is initiated, the FIFOEF* bit is set approximately 5 µsec after initiating the conversion, indicating that the data conversion result can be read from the FIFO.

An A/D FIFO overflow condition occurs if more than 256 conversions are initiated and stored in the A/D FIFO before the A/D FIFO Register is read. If this condition occurs, the OVERFLOW* bit is clear in the Status Register to alert you that one or more A/D conversion results have been lost because of FIFO overflow. Writing to ADCLEAR resets this error flag.

An ADC overrun condition occurs if an attempt is made to start a new conversion while the previous conversion is being completed. If this condition occurs, the OVERRUN* bit is clear in the Status Register to indicate the error condition that an invalid operation occurred. Writing to ADCLEAR resets this error flag.

A/D FIFO Output Binary Formats

The A/D conversion result can be returned from the A/D FIFO as a two's complement or straight binary value depending on the input mode set by the UNIP/BIP bit in Command Register 1. If the analog input circuitry is configured for the input range 0 to +10 V(UNIP/BIP enabled), straight binary format is implemented. Straight binary format returns numbers between 0 and 4095 (decimal) when the A/D FIFO Register is read. If the analog input circuitry is configured

for the input ranges -5 to +5 V or -10 to +10 V (UNIP/BIP disabled), two's complement format is used. Two's complement format returns numbers between -2048 and +2047 (decimal) when the A/D FIFO Register is read. Table 4-4 shows input voltage versus A/D conversion value for straight binary format and 0 to +10V input range. Table 4-5 shows input voltage versus A/D conversion value for two's complement format for both -5 to +5 V and -10 to +10 V input ranges.

Table 4-4. Straight Binary Mode A/D Conversion Values

Input Voltage (Gain = 1)	**A/D Conversion Result Range: 0 to 10 V**	
	Decimal	**Hex**
0	0	0000
2.5	1024	0400
5.0	2048	0800
7.5	3072	0C00
9.9976	4095	0FFF

To convert from the A/D FIFO value to the input voltage measured, use the following formula:

$$V = \frac{\text{A/D Count}}{4096} * \frac{10\ V}{\text{Gain}}$$

Table 4-5. Two's Complement Mode A/D Conversion Values

	A/D Conversion Result			
Input Voltage	**Range: -5 to +5 V (Gain = 1)**		**Range: -10 to +10 V (Gain = 0.5)**	
	Decimal	**Hex**	**Decimal**	**Hex**
-10.0	—	—	-2048	F800
-5.0	-2048	F800	-1024	FC00
-2.5	-1024	FC00	-512	FE00
0	0	0000	0	0000
2.5	1024	0400	512	0200
4.9976	2047	07FF	—	—
5.0	—	—	1024	0400
9.9951	—	—	2047	07FF

To convert from the A/D FIFO value to the input voltage measured, use the appropriate formula as follows:

$$\pm 5\text{ V Range} = \frac{\text{A/D Count}}{2048} * \frac{5\text{ V}}{\text{Gain}}$$

$$\pm 10\text{ V Range} = \frac{\text{A/D Count}}{2048} * \frac{10\text{ V}}{\text{Gain}}$$

Clearing the Analog Input Circuitry

The analog input circuitry can be cleared by writing to ADCLEAR. This operation leaves the analog input circuitry in the following state:

- Analog input error flags OVERFLOW and OVERRUN are cleared.
- Pending interrupt requests are cleared.
- A/D FIFO is emptied.
- DAQCOMP flag in the Status Register is cleared.

Empty the FIFO before starting any A/D conversions. This action guarantees that the A/D conversion results read from the FIFO are the results from the initiated conversions, not left over results from previous conversions.

Posttrigger Data Acquisition Sequence

The following programming sequences for sample counts less than 65,537 leave the data acquisition circuitry in a retriggerable state. The sample-interval and sample counters are reloaded at the end of the data acquisition to prepare for another data acquisition operation. The counters do not need reprogramming, and the next data acquisition operation starts when another trigger is received.

Programming multiple A/D conversions on a single channel requires the following programming steps:

1. Select analog input channel and gain.
2. Program the sample-interval counter.
3. Program the sample counter.
4. Clear the A/D circuitry.
5. Enable the data acquisition operation.
6. Apply a trigger.
7. Service the data acquisition operation.

Each of these programming steps is explained further.

1. Select analog input channel and gain.

The analog input channel and gain are selected by writing to the MUXMEM Register followed by a write to the MUXMEMLD Register. Bits 7 through 4 select the analog input channel and bits 3 through 1 control the gain. See the MUXMEM Register bit description earlier in this chapter for analog input channel and gain bit patterns. Set up the bits as given in the MUXMEM Register bit description and write to the MUXMEM Register. Remember that the mux memory should be initialized first with a write to the MUXMEMCLR Register.

Once the MUXMEM Register is set up, it needs to be written to only when the analog input channel or gain settings need to be changed.

2. Program the sample-interval counter.

Counter 3 of the Am9513A Counter/Timer is used as the sample-interval counter. Counter 3 can be programmed to generate an active low pulse once every *N* counts. *N* is referred to as the sample interval, that is, the time between successive A/D conversions. *N* can be between 2 and 65,536. One count is equal to the period of the timebase clock used by the counter. The following clocks are available internal to the Am9513A: 5-MHz, 1-MHz, 100-kHz, 10-kHz, 1-kHz, and 100-Hz. In addition, the sample-interval timer can use signals connected to any of the Am9513A SOURCE input pins.

To program the sample-interval counter, use the following programming sequence. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF03 to the Am9513A Command Register to select the Counter 3 Mode Register.

b. Write the mode value to the Am9513A Data Register to store the Counter 3 mode value. Use one of the following mode values (Am9513A counter mode information can be found in Appendix C):

   8225 – Selects 5-MHz clock (from SOURCE2 pin)
   8B25 – Selects 1-MHz clock
   8C25 – Selects 100-kHz clock
   8D25 – Selects 10-kHz clock
   8E25 – Selects 1-kHz clock
   8F25 – Selects 100-Hz clock
   8525 – Selects signal at SOURCE5 input as clock (counts the rising edge of the signal, 6 MHz maximum)

c. Write FF0B to the Am9513A Command Register to select the Counter 3 Load Register.

d. Write 2 to the Am9513A Data Register to store the Counter 3 load value.

e. Write FF44 to the Am9513A Command Register to load Counter 3.

f. Write FFF3 to the Am9513A Command Register to step Counter 3 down to 1.

g. Write the desired sample interval to the Am9513A Data Register to store the Counter 3 load value:

   - If the sample interval is between 2 and FFFF (65,535 decimal), write the sample interval to the Am9513A Data Register.

   - If the sample interval is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.

h. Write FF24 to the Am9513A Command Register to arm Counter 3.

After you complete this programming sequence, Counter 3 is configured to generate A/D conversion pulses as soon as application of a trigger causes it to be enabled.

3. Program the sample counter.

Counters 4 and 5 of the Am9513A Counter/Timer are used as the sample counter. The sample counter tallies the number of A/D conversions initiated by Counter 3 and stops Counter 3 when the desired sample count is reached. If the desired sample count is 65,536 or less, only Counter 4 needs to be used, making Counter 5 available for general-purpose timing applications. If the desired sample count is greater than 65,536, both Counters 4 and 5 must be used.

Sample Counts 2 through 65,536.

To program the sample counter for sample counts up to 65,536, use the following programming sequence. The minimum permitted sample count is 2. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF04 to the Am9513A Command Register to select the Counter 4 Mode Register.

b. Write 1025 to the Am9513A Data Register to store the Counter 4 mode value.

c. Write FF0C to the Am9513A Command Register to select the Counter 4 Load Register.

d. Write the sample count value to the Am9513A Data Register to store the Counter 4 load value:

   - If the sample count is between 2 and FFFF (65,535 decimal), write the sample count to the Am9513A Data Register.

   - If the sample count is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.

e. Write FF48 to the Am9513A Command Register to load Counter 4.

f. Write FFF4 to the Am9513A Command Register to decrement Counter 4.

g. Write FF28 to the Am9513A Command Register to arm Counter 4.

h. Clear the 16*/32 CNT bit in Command Register 1 to notify the hardware that only Counter 4 will be used as the sample counter.

After you complete this programming sequence, Counter 4 is configured to count A/D conversion pulses generated by Counter 3 and turns off the data acquisition operation when Counter 4 decrements to 0.

Sample Counts Greater than 65,536.

To program the sample counter for sample counts greater than 65,536, use the following programming sequence to concatenate Counter 4 to Counter 5. The lower 16 bits of the sample count are stored in Counter 4, and the upper 16 bits of the sample count are stored in Counter 5. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF04 to the Am9513A Command Register to select the Counter 4 Mode Register.

b. Write 1025 to the Am9513A Data Register to store the Counter 4 mode value.

c. Write FF0C to the Am9513A Command Register to select the Counter 4 Load Register.

d. Write the least significant 16 bits of the sample count value minus 1 to the Am9513A Data Register to store the Counter 4 load value.

   - If the least significant 16 bits are all 0, write FFFF.

e. Write FF48 to the Am9513A Command Register to load Counter 4.

f. Write 0 to the Am9513A Data Register to store 0 into the Load Register for Counter 4 reloading.

g. Write FF28 to the Am9513A Command Register to arm Counter 4.

h. Write FF05 to the Am9513A Command Register to select the Counter 5 Mode Register.

i. Write 25 to the Am9513A Data Register to store the Counter 5 mode value.

j. Write FF0D to the Am9513A Command Register to select the Counter 5 Load Register.

k. Take the most significant 16 bits of the sample count and do the following:

   - If the least significant 16 bits of the sample count are all 0 or all 0 except for a 1 in the least significant bit, write the most significant 16 bits to the Am9513A Data Register to store the Counter 5 load value.

   - Otherwise, add 1 to the most significant 16 bits of the sample count and write that value to the Am9513A Data Register to store the Counter 5 load value.

l. Write FF70 to the Am9513A Command Register to load and arm Counter 5.

m. Set the 16*/32 CNT bit in Command Register 1 to notify the hardware that both Counters 4 and 5 will be used as the sample counter.

After you complete this programming sequence, Counter 4 is configured to count A/D conversion pulses generated by Counter 3, and Counter 5 decrements every time Counter 4 reaches 0. The data acquisition operation is terminated when Counter 4 and Counter 5 reach 0.

4. Clear the A/D circuitry.

Before you start the data acquisition operation, the A/D FIFO must be emptied to clear out any old A/D conversion results. This emptying must be done after the counters are programmed in case any spurious edges were caused while programming the counters. Write 0 to the A/D Clear Register to empty the FIFO, then write 0 to the MUXMEMLD Register to load the first channel-gain value.

5. Enable the data acquisition operation.

To enable the data acquisition operation such that A/D conversions begin when a trigger is received, set the DAQEN bit in Command Register 1. Also, make sure INTGATE in Command Register 1 and EXTGATE* at the I/O connector are disabled.

6. Apply a trigger.

Once set up by the preceding steps, the data acquisition operation is initiated when a trigger is received. A trigger can be initiated through software or through hardware.

To initiate the data acquisition operation through software, write 0 to the Start DAQ Register (16-bit write). Make sure EXTTRIG* is not pulled low at the I/O connector or the RTSI switch.

To initiate the data acquisition operation through hardware, apply an active low pulse to the EXTTRIG* pin on the AT-MIO-16F-5 I/O connector. See *Data Acquisition Timing Connections* in Chapter 2 for EXTTRIG* signal specifications.

Once the trigger is applied, Counter 3 generates pulses initiating A/D conversions once every sample interval until the sample counter reaches 0.

7. Service the data acquisition operation.

Once the data acquisition operation is started by application of a trigger, the operation must be serviced by reading the FIFO every time an A/D conversion result becomes available. To do this, perform the following sequence until the data acquisition has completed:

a. Read the Status Register (16-bit read).

b. If OVERRUN* or OVERFLOW* are clear, then the data acquisition sequence has been halted because one of these error conditions has occurred. Clear the A/D circuitry by writing to ADCLEAR and determine the cause of the error. OVERRUN* and OVERFLOW* are explained in step 3 of *Programming the Analog Input Circuitry* earlier in this chapter.

c. If the FIFOEF* bit is set , read the A/D FIFO Register to obtain the result.

Interrupts or DMA can also be used to service the data acquisition operation. These topics are discussed later in this chapter.

Controlling Posttrigger Acquisition Sequences with the EXTCONV* Signal

Using the EXTCONV* signal from the I/O connector to control multiple A/D conversions is exactly like a posttrigger data acquisition sequence except you should not program Counter 3, the sample-interval counter. This counter should be left in the high impedance state (see *Posttrigger Data Acquisition Sequence* earlier in this chapter). Conversions are generated by the falling edge of the EXTCONV* signal. Even though EXTCONV* may be pulsing, conversions do not begin until after an active low pulse on STARTDAQ or the EXTTRIG* signal. Conversions are automatically halted irrespective of the EXTCONV* signal when the sample counter reaches zero.

Pretrigger Data Acquisition Sequence

In this case, the sample counter starts counting when a second falling edge signal is applied to the EXTTRIG* input on the AT-MIO-16F-5 I/O connector or a write to the STARTDAQ Register. The sample counter should be programmed for active high-level gating on Gate 4. The data acquisition operation is initiated by writing to the STARTDAQ Register or by a falling edge

on the EXTTRIG* signal. The sample counter does not begin counting samples until this second trigger condition. To perform this operation, complete the following steps:

1. Select analog input channel and gain.
2. Program the sample-interval counter.
3. Program the sample counter.
4. Clear the A/D circuitry.
5. Enable the data acquisition operation.
6. Apply a trigger.
7. Service the data acquisition operation.

1. Select analog input channel and gain.

The analog input channel and gain are selected by writing to the MUXMEM Register followed by a write to the MUXMEMLD Register. Bits 7 through 4 select the analog input channel and bits 3 through 1 control the gain. See the MUXMEM Register bit description earlier in this chapter for analog input channel and gain bit patterns. Set up the bits as given in the MUXMEM Register bit description and write to the MUXMEM Register. Remember that the mux memory should be initialized first with a write to the MUXMEMCLR Register.

Once the MUXMEM Register is set up, it needs to be written to only when the analog input channel or gain settings need to be changed.

2. Program the sample-interval counter.

Counter 3 of the Am9513A Counter/Timer is used as the sample-interval counter. Counter 3 can be programmed to generate an active low pulse once every *N* counts. *N* is referred to as the sample interval, that is, the time between successive A/D conversions. *N* can be between 2 and 65,536. One count is equal to the period of the timebase clock used by the counter. The following clocks are available internal to the Am9513A: 5-MHz, 1-MHz, 100-kHz, 10-kHz, 1-kHz, and 100-Hz. In addition, the sample-interval timer can use signals connected to any of the Am9513A SOURCE input pins.

To program the sample-interval counter, use the following programming sequence. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF03 to the Am9513A Command Register to select the Counter 3 Mode Register.

b. Write the mode value to the Am9513A Data Register to store the Counter 3 mode value. Use one of the following mode values (Am9513A counter mode information can be found in Appendix C):

   8225 – Selects 5-MHz clock (from SOURCE2 pin)
   8B25 – Selects 1-MHz clock
   8C25 – Selects 100-kHz clock
   8D25 – Selects 10-kHz clock
   8E25 – Selects 1-kHz clock
   8F25 – Selects 100-Hz clock
   8525 – Selects signal at SOURCE5 input as clock (counts the rising edge of the signal, 6 MHz maximum)

c. Write FF0B to the Am9513A Command Register to select the Counter 3 Load Register.

d. Write 2 to the Am9513A Data Register to store the Counter 3 load value.

e. Write FF44 to the Am9513A Command Register to load Counter 3.

f. Write FFF3 to the Am9513A Command Register to step Counter 3 down to 1.

g. Write the desired sample interval to the Am9513A Data Register to store the Counter 3 load value:

   - If the sample interval is between 2 and FFFF (65,535 decimal), write the sample interval to the Am9513A Data Register.
   - If the sample interval is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.

h. Write FF24 to the Am9513A Command Register to arm Counter 3.

After you complete this programming sequence, Counter 3 is configured to generate A/D conversion pulses as soon as application of a trigger causes it to be enabled.

3. Program the sample counter.

Counters 4 and 5 of the Am9513A Counter/Timer are used as the sample counter. The sample counter tallies the number of A/D conversions initiated by Counter 3 and stops Counter 3 when the desired sample count is reached. If the desired sample count is 65,536 or less, only Counter 4 needs to be used, making Counter 5 available for general-purpose timing applications. If the desired sample count is greater than 65,536, both Counters 4 and 5 must be used.

Sample Counts 2 through 65,536.

To program the sample counter for sample counts up to 65,536, use the following programming sequence. The minimum permitted sample count is 2. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF04 to the Am9513A Command Register to select the Counter 4 Mode Register.

b. Write 9025 to the Am9513A Data Register to store the Counter 4 mode value.

c. Write FF0C to the Am9513A Command Register to select the Counter 4 Load Register.

d. Write the sample count value to the Am9513A Data Register to store the Counter 4 load value:

   - If the sample count is between 2 and FFFF (65,535 decimal), write the sample count to the Am9513A Data Register.
   - If the sample count is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.

e. Write FF48 to the Am9513A Command Register to load Counter 4.

f. Write FFF4 to the Am9513A Command Register to decrement Counter 4.

g. Write FF28 to the Am9513A Command Register to arm Counter 4.

h. Clear the 16*/32 CNT bit in Command Register 1 to notify the hardware that only Counter 4 will be used as the sample counter.

After you complete this programming sequence, Counter 4 is configured to count A/D conversion pulses generated by Counter 3 and turns off the data acquisition operation when Counter 4 decrements to 0.

Sample Counts Greater than 65,536.

To program the sample counter for sample counts greater than 65,536, use the following programming sequence to concatenate Counter 4 to Counter 5. The lower 16 bits of the sample count are stored in Counter 4, and the upper 16 bits of the sample count are stored in Counter 5. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF04 to the Am9513A Command Register to select the Counter 4 Mode Register.

b. Write 9025 to the Am9513A Data Register to store the Counter 4 mode value.

c. Write FF0C to the Am9513A Command Register to select the Counter 4 Load Register.

d. Write the least significant 16 bits of the sample count value minus 1 to the Am9513A Data Register to store the Counter 4 load value.

   - If the least significant 16 bits are all 0, write FFFF.

e. Write FF48 to the Am9513A Command Register to load Counter 4.

f. Write 0 to the Am9513A Data Register to store 0 into the Load Register for Counter 4 reloading.

g. Write FF28 to the Am9513A Command Register to arm Counter 4.

h. Write FF05 to the Am9513A Command Register to select the Counter 5 Mode Register.

i. Write 25 to the Am9513A Data Register to store the Counter 5 mode value.

j. Write FF0D to the Am9513A Command Register to select the Counter 5 Load Register.

k. Take the most significant 16 bits of the sample count and do the following:

- If the least significant 16 bits of the sample count are all 0 or all 0 except for a 1 in the least significant bit, write the most significant 16 bits to the Am9513A Data Register to store the Counter 5 load value.
- Otherwise, add 1 to the most significant 16 bits of the sample count and write that value to the Am9513A Data Register to store the Counter 5 load value.

l. Write FF70 to the Am9513A Command Register to load and arm Counter 5.

m. Set the 16*/32 CNT bit in Command Register 1 to notify the hardware that both Counters 4 and 5 will be used as the sample counter.

After you complete this programming sequence, Counter 4 is configured to count A/D conversion pulses generated by Counter 3, and Counter 5 decrements every time Counter 4 reaches 0. The data acquisition operation is terminated when both Counters 4 and 5 reach 0.

4. Clear the A/D circuitry.

Before you start the data acquisition operation, the A/D FIFO must be emptied to clear out any old A/D conversion results. This emptying must be done after the counters are programmed in case any spurious edges were caused while programming the counters. Write 0 to the A/D Clear Register to empty the FIFO, then write 0 to the MUXMEMLD Register to load the first channel-gain value.

5. Enable the data acquisition operation.

To enable the data acquisition operation such that A/D conversions begin when a trigger is received, set the DAQEN bit in Command Register 1. Also, make sure INTGATE in Command Register 1 and EXTGATE* at the I/O connector are disabled.

6. Apply a trigger.

Once set up by the preceding steps, a trigger can be initiated through software or hardware.

To initiate the data acquisition operation through software, write 0 to the STARTDAQ Register (16-bit write). Make sure EXTTRIG* is not pulled low at the I/O connector or the RTSI switch.

To initiate the data acquisition operation through hardware, apply an active low pulse to the EXTTRIG* pin on the AT-MIO-16F-5 I/O connector. See *Data Acquisition Timing Connections* in Chapter 2 for EXTTRIG* signal specifications.

Once the first trigger is applied, Counter 3 generates pulses initiating A/D conversions once every sample interval, but these conversions are not counted by the sample counter. Counting begins after the application of a second hardware or software trigger condition and continues until the sample counter reaches 0. A/D conversion data stored before receipt of the EXTTRIG* or STARTDAQ signal are pretrigger samples.

7. Service the data acquisition operation.

Once the data acquisition operation is started by application of a trigger, the operation must be serviced by reading the FIFO every time an A/D conversion result becomes available. To do this, perform the following sequence until the data acquisition has completed:

a. Read the Status Register (16-bit read).

b. If OVERRUN* or OVERFLOW* are clear, then the data acquisition sequence has been halted because one of these error conditions is true. Clear the A/D circuitry by writing to ADCLEAR and determine the cause of the error. OVERRUN* and OVERFLOW* are explained in step 3 of *Programming the Analog Input Circuitry* earlier in this chapter.

c. If the FIFOEF* bit is set, read the A/D FIFO Register to obtain the result.

Interrupts or DMA can also be used to service the data acquisition operation. These topics are discussed later in this chapter.

Controlling Pretrigger Data Acquisition Sequences with the EXTCONV* Signal

Using the EXTCONV* signal to control pretrigger multiple A/D conversions is exactly like a pretrigger data acquisition sequence except step 2 in the programming sequence should be omitted and Counter 3 should be left in the high-impedance state. Even though EXTCONV* may be pulsing, conversions do not begin until the application of the first active low pulse of EXTTRIG* or STARTDAQ. Conversions are generated by the high-to-low edge of the EXTCONV* signal. The sample counter does not begin counting the conversions until a second application of the EXTTRIG* or STARTDAQ signals. Conversions are halted when the sample counter expires.

Programming Data Acquisition Sequences with Channel Scanning

The preceding data acquisition programming sequences program the AT-MIO-16F-5 for multiple A/D conversion on a single input channel. The AT-MIO-16F-5 can also be programmed for scanning multiple analog input channels and switching gain settings during the data acquisition

operation. The sequence of A/D channels and gain settings, called the *scan sequence*, is programmed into the mux-channel-gain memory.

There are two types of multiple A/D conversions with channel scanning: continuous channel scanning and interval channel scanning. Continuous channel scanning cycles through the scan sequence in the mux-channel-gain memory and repeats the scan sequence until the sample counter terminates the data acquisition. There is no delay between the cycles of the scan sequence. Continuous channel scanning can be thought of as a *round-robin* approach to scanning multiple channels.

Interval channel scanning gives each scan sequence a programmed time interval called a *scan interval*. Each cycle of the scan sequence begins at the time interval determined by the scan interval. If the sample-interval counter is programmed for the minimum time required to complete an A/D conversion, interval channel scanning can be thought of as a *pseudo-simultaneous* scanning of multiple channels; that is, all channels in the scan sequence are read as quickly as possible at the beginning of each scan interval.

Posttrigger Data Acquisition with Continuous Channel Scanning

To program continuous scanning of multiple A/D conversions, use the following programming steps:

1. Set up the analog channel and gain selection sequence.
2. Program the sample-interval counter.
3. Program the sample counter.
4. Clear the A/D circuitry and reset the mux counter.
5. Enable the scanning data acquisition operation.
6. Apply a trigger.
7. Service the data acquisition operation.

Setting the SCANEN bit in Command Register 1 enables scanning during multiple A/D conversions. This bit must be set regardless of the type of scanning used; otherwise, only a single channel is scanned. In addition, a channel and gain scan sequence must be stored in the mux-channel-gain memory.

1. Set up the analog channel and gain selection sequence.

During a scanning data acquisition operation, a selected number of locations in the mux-channel-gain memory are stepped through by means of the acquisition circuitry. A new mux-channel-gain memory location is selected after each A/D conversion. The first conversion is performed on the first channel and gain setting in the memory. The second conversion is performed on the second channel and gain setting, and so on. The last entry written to the mux-channel-gain memory must have the LASTONE bit set. This bit marks the end of the scan sequence. After the last conversion is performed, the scan sequence starts over. If there are *X* entries in the mux-channel-gain memory, every *X*th conversion in the data collected is performed on the same channel and gain setting.

Multiple conversions can be performed on each entry in the mux-channel-gain memory before incrementing to the next entry in the scan sequence. If the SCANDIV bit in Command Register 1 is set, the mux-channel-gain memory increments to the next entry when an active low pulse is detected on the Am9513A Counter/Timer OUT1 signal. If the SCANDIV bit is cleared, the mux-channel-gain memory is incremented to the next entry after every conversion.

The mux-channel-gain memory must be loaded with the desired scan sequence before data acquisition begins. To load the mux-channel-gain memory, perform the following write operations where *X* is the number of entries in the scan sequence:

- Write 0 to the MUXMEMCLR Register.
- For $i = 0$ to $X-1$, use the following steps:
  a. Write the desired analog channel selection and gain setting to the MUXMEM Register (this loads the mux-channel-gain memory at location *i*).
  b. If $i = X-1$, also set the LASTONE bit when writing to the MUXMEM Register.
- Write 0 to the MUXMEMLD Register.

2. Program the sample-interval counter.

Counter 3 of the Am9513A Counter/Timer is used as the sample-interval counter. Counter 3 can be programmed to generate an active low pulse once every *N* counts. *N* is referred to as the sample interval, that is, the time between successive A/D conversions. *N* can be between 2 and 65,536. One count is equal to the period of the timebase clock used by the counter. The following clocks are available internal to the Am9513A: 5-MHz, 1-MHz, 100-kHz, 10-kHz, 1-kHz, and 100-Hz. In addition, the sample-interval timer can use signals connected to any of the Am9513A SOURCE input pins.

To program the sample-interval counter, use the following programming sequence. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF03 to the Am9513A Command Register to select the Counter 3 Mode Register.

b. Write the mode value to the Am9513A Data Register to store the Counter 3 mode value. Use one of the following mode values (Am9513A counter mode information can be found in Appendix C):

   8225 – Selects 5-MHz clock (from SOURCE2 pin)
   8B25 – Selects 1-MHz clock
   8C25 – Selects 100-kHz clock
   8D25 – Selects 10-kHz clock
   8E25 – Selects 1-kHz clock
   8F25 – Selects 100-Hz clock
   8525 – Selects signal at SOURCE5 input as clock (counts the rising edge of the signal, 6 MHz maximum)

c. Write FF0B to the Am9513A Command Register to select the Counter 3 Load Register.

d. Write 2 to the Am9513A Data Register to store the Counter 3 load value.

e. Write FF44 to the Am9513A Command Register to load Counter 3.

f. Write FFF3 to the Am9513A Command Register to step Counter 3 down to 1.

g. Write the desired sample interval to the Am9513A Data Register to store the Counter 3 load value:

   - If the sample interval is between 2 and FFFF (65,535 decimal), write the sample interval to the Am9513A Data Register.
   - If the sample interval is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.
   - If the sample interval is greater than 10000.

h. Write FF24 to the Am9513A Command Register to arm Counter 3.

After you complete this programming sequence, Counter 3 is configured to generate A/D conversion pulses as soon as application of a trigger causes it to be enabled.

3. Program the Sample Counter.

Counters 4 and 5 of the Am9513A Counter/Timer are used as the sample counter. The sample counter tallies the number of A/D conversions initiated by Counter 3 and stops Counter 3 when the desired sample count is reached. If the desired sample count is 65,536 or less, only Counter 4 needs to be used, making Counter 5 available for general-purpose timing applications. If the desired sample count is greater than 65,536, both Counters 4 and 5 must be used.

Sample Counts 2 through 65,536.

To program the sample counter for sample counts up to 65,536, use the following programming sequence. The minimum permitted sample count is 2. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF04 to the Am9513A Command Register to select the Counter 4 Mode Register.

b. Write 1025 to the Am9513A Data Register to store the Counter 4 mode value.

c. Write FF0C to the Am9513A Command Register to select the Counter 4 Load Register.

d. Write the sample count value to the Am9513A Data Register to store the Counter 4 load value:

   - If the sample count is between 2 and FFFF (65,535 decimal), write the sample count to the Am9513A Data Register.

   - If the sample count is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.

e. Write FF48 to the Am9513A Command Register to load Counter 4.

f. Write FFF4 to the Am9513A Command Register to decrement Counter 4.

g. Write FF28 to the Am9513A Command Register to arm Counter 4.

h. Clear the 16*/32 CNT bit in Command Register 1 to notify the hardware that only Counter 4 will be used as the sample counter.

After you complete this programming sequence, Counter 4 is configured to count A/D conversion pulses generated by Counter 3 and turns off the data acquisition operation when Counter 4 decrements to 0.

Sample Counts Greater than 65,536.

To program the sample counter for sample counts greater than 65,536, use the following programming sequence to concatenate Counter 4 to Counter 5. The lower 16 bits of the sample count are stored in Counter 4, and the upper 16 bits of the sample count are stored in Counter 5. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF04 to the Am9513A Command Register to select the Counter 4 Mode Register.

b. Write 1025 to the Am9513A Data Register to store the Counter 4 mode value.

c. Write FF0C to the Am9513A Command Register to select the Counter 4 Load Register.

d. Write the least significant 16 bits of the sample count value minus 1 to the Am9513A Data Register to store the Counter 4 load value.

   - If the least significant 16 bits are all 0, write FFFF.

e. Write FF48 to the Am9513A Command Register to load Counter 4.

f. Write 0 to the Am9513A Data Register to store 0 into the Load Register for Counter 4 reloading.

g. Write FF28 to the Am9513A Command Register to arm Counter 4.

h. Write FF05 to the Am9513A Command Register to select the Counter 5 Mode Register.

i. Write 25 to the Am9513A Data Register to store the Counter 5 mode value.

j. Write FF0D to the Am9513A Command Register to select the Counter 5 Load Register.

k. Take the most significant 16 bits of the sample count and do the following:

   - If the least significant 16 bits of the sample count are all 0 or all 0 except for a 1 in the least significant bit, write the most significant 16 bits to the Am9513A Data Register to store the Counter 5 load value.

   - Otherwise, add 1 to the most significant 16 bits of the sample count and write that value to the Am9513A Data Register to store the Counter 5 load value.

l. Write FF70 to the Am9513A Command Register to load and arm Counter 5.

m. Set the 16*/32 CNT bit in Command Register 1 to notify the hardware that both Counters 4 and 5 will be used as the sample counter.

The only restriction on the sample count in the scanning mode is that the sample count should be programmed as a multiple of the number of entries in the mux-channel-gain memory. Thus, if three channels are to be scanned and five samples per channel are desired, the sample count should be programmed as 15.

After you complete this programming sequence, Counter 4 is configured to count A/D conversion pulses generated by Counter 3, and Counter 5 increments every time Counter 4 reaches 0. The data acquisition operation is terminated when both Counters 4 and 5 reach 0 and the last entry in the mux-channel-gain memory is served.

4. Clear the A/D circuitry.

Before starting the data acquisition operation, the FIFO should be emptied to clear out any old conversion results. Emptying must be done after the counters are programmed in case any spurious edges were caused while programming the counters. Write 0 to ADCLEAR to empty the FIFO, then write 0 to the MUXMEMLD Register to load the first channel-gain value.

5. Enable the scanning data acquisition operation.

To enable the scanning data acquisition operation such that A/D conversions begin when a trigger is received, set the DAQEN bit and the SCANEN bit in Command Register 1. Also, make sure INTGATE* in Command Register 1 and EXTGATE* at the I/O connector are disabled.

6. Apply a trigger.

Once set up by the preceding steps, the data acquisition operation is initiated when a trigger is received. A trigger can be initiated through software or through hardware.

To initiate the data acquisition operation through software, write 0 to the Start DAQ Register (16-bit write). Make sure EXTTRIG* is not pulled low at the I/O connector or the RTSI switch.

To initiate the data acquisition operation through hardware, apply an active low pulse to the EXTTRIG* pin on the AT-MIO-16F-5 I/O connector. See *Data Acquisition Timing Connections* in Chapter 2 for EXTTRIG* signal specifications.

Once the trigger is applied, Counter 3 generates pulses initiating A/D conversions once every sample interval until the sample counter reaches 0.

7. Service the data acquisition operation.

Once the data acquisition operation is started by application of a trigger, the operation must be serviced by reading the FIFO every time an A/D conversion result becomes available. To do this, perform the following sequence until the data acquisition has completed:

a. Read the Status Register (16-bit read).

b. If OVERRUN* or OVERFLOW* are clear, then the data acquisition sequence has been halted because one of these error conditions has cccurred. Clear the A/D circuitry by writing to ADCLEAR and determine the cause of the error. OVERRUN* and OVERFLOW* are explained in step 3 of *Programming the Analog Input Circuitry.*

c. If the FIFOEF* bit is set, read the A/D FIFO Register to obtain the result.

Interrupts or DMA can also be used to service the data acquisition operation. These topics are discussed later in this chapter.

Controlling Posttrigger Channel Scanning with the EXTCONV* Signal

Using the EXTCONV* signal to control multiple A/D conversions is exactly like posttrigger data acquisition with channel scanning except step 2 in the programming sequence should be omitted and Counter 3 should be left in the high-impedance state. Conversions are generated by the falling edge of the EXTCONV* signal. Even though EXTCONV* may be pulsing, conversions do not begin until after an active low pulse on STARTDAQ or the EXTTRIG* signal. Conversions are automatically halted when the sample counter reaches zero, irrespective of the EXTCONV* signal .

Pretrigger Data Acquisition with Continuous Channel Scanning

Programming continuous scanning of multiple A/D conversion (round robin) involves the following programming steps:

1. Set up the analog channel and gain selection sequence.
2. Program the sample-interval counter.
3. Program the sample counter.
4. Clear the A/D circuitry and reset the mux counter.
5. Enable the scanning data acquisition operation.
6. Apply a trigger.
7. Service the data acquisition operation.

Setting the SCANEN bit in Command Register 1 enables scanning during multiple A/D conversions. This bit must be set regardless of the type of scanning used; otherwise, only a single channel is scanned. In addition, a channel and gain scan sequence must be stored in the mux-channel-gain memory.

1. Set up the analog channel and gain selection sequence.

During a scanning data acquisition operation, a selected number of locations in the mux-channel-gain memory are stepped through by means of the acquisition circuitry. A new mux-channel-gain memory location is selected after each A/D conversion. The first conversion is performed on the first channel and gain setting in the memory. The second conversion is performed on the second channel and gain setting, and so on. The last entry written to the mux-channel-gain memory must have the LASTONE bit set. This bit marks the end of the scan sequence. After the last conversion is performed, the scan sequence starts over. If there are *X* entries in the mux-channel-gain memory, every *X*th conversion in the data collected is performed on the same channel and gain setting.

Multiple conversions can be performed on each entry in the mux-channel-gain memory before incrementing to the next entry in the scan sequence. If the SCANDIV bit in Command Register 1 is set, the mux-channel-gain memory increments to the next entry when an active low pulse is detected on the Am9513A Counter/Timer OUT1 signal. If the SCANDIV bit is cleared, the mux-channel-gain memory is incremented to the next entry after every conversion.

The mux-channel-gain memory must be loaded with the desired scan sequence before data acquisition begins. To load the mux-channel-gain memory, perform the following write operations where *X* is the number of entries in the scan sequence:

- Write 0 to the MUXMEMCLR Register.
- For $i = 0$ to X-1, use the following steps:
  a. Write the desired analog channel selection and gain setting to the MUXMEM Register (this loads the mux-channel-gain memory at location *i*).
  b. If $i = X$-1, also set the LASTONE bit when writing to the MUXMEM Register.
- Write 0 to the MUXMEMLD Register.

2. Program the sample-interval counter.

Counter 3 of the Am9513A Counter/Timer is used as the sample-interval counter. Counter 3 can be programmed to generate an active low pulse once every *N* counts. *N* is referred to as the sample interval, that is, the time between successive A/D conversions. *N* can be between 2 and 65,536. One count is equal to the period of the timebase clock used by the counter. The following clocks are available internal to the Am9513A: 5-MHz, 1-MHz, 100-kHz, 10-kHz, 1-kHz, and 100-Hz. In addition, the sample-interval timer can use signals connected to any of the Am9513A SOURCE input pins.

To program the sample-interval counter, use the following programming sequence. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF03 to the Am9513A Command Register to select the Counter 3 Mode Register.

b. Write the mode value to the Am9513A Data Register to store the Counter 3 mode value. Use one of the following mode values (Am9513A counter mode information can be found in Appendix C):

8225 – Selects 5-MHz clock (from SOURCE2 pin)
8B25 – Selects 1-MHz clock
8C25 – Selects 100-kHz clock
8D25 – Selects 10-kHz clock
8E25 – Selects 1-kHz clock
8F25 – Selects 100-Hz clock
8525 – Selects signal at SOURCE5 input as clock (counts the rising edge of the signal, 6 MHz maximum)

c. Write FF0B to the Am9513A Command Register to select the Counter 3 Load Register.

d. Write 2 to the Am9513A Data Register to store the Counter 3 load value.

e. Write FF44 to the Am9513A Command Register to load Counter 3.

f. Write FFF3 to the Am9513A Command Register to step Counter 3 down to 1.

g. Write the desired sample interval to the Am9513A Data Register to store the Counter 3 load value:

- If the sample interval is between 2 and FFFF (65,535 decimal), write the sample interval to the Am9513A Data Register.
- If the sample interval is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.
- If the sample interval is greater than 10000.

h. Write FF24 to the Am9513A Command Register to arm Counter 3.

After you complete this programming sequence, Counter 3 is configured to generate A/D conversion pulses as soon as application of a trigger causes it to be enabled.

3. Program the sample counter

Counters 4 and 5 of the Am9513A Counter/Timer are used as the sample counter. The sample counter tallies the number of A/D conversions initiated by Counter 3 and stops Counter 3 when the desired sample count is reached. If the desired sample count is 65,536 or less, only Counter 4 needs to be used, making Counter 5 available for general-purpose timing applications. If the desired sample count is greater than 65,536, both Counters 4 and 5 must be used.

Sample Counts 2 through 65,536.

To program the sample counter for sample counts up to 65,536, use the following programming sequence. The minimum permitted sample count is 2. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF04 to the Am9513A Command Register to select the Counter 4 Mode Register.

b. Write 9025 to the Am9513A Data Register to store the Counter 4 mode value.

c. Write FF0C to the Am9513A Command Register to select the Counter 4 Load Register.

d. Write the sample count value to the Am9513A Data Register to store the Counter 4 load value:

- If the sample count is between 2 and FFFF (65,535 decimal), write the sample count to the Am9513A Data Register.
- If the sample count is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.

e. Write FF48 to the Am9513A Command Register to load Counter 4.

f. Write FFF4 to the Am9513A Command Register to decrement Counter 4.

g. Write FF28 to the Am9513A Command Register to arm Counter 4.

h. Clear the 16*/32 CNT bit in Command Register 1 to notify the hardware that only Counter 4 will be used as the sample counter.

After you complete this programming sequence, Counter 4 is configured to count A/D conversion pulses generated by Counter 3 and turns off the data acquisition operation when Counter 4 decrements to 0.

Sample Counts Greater than 65,536.

To program the sample counter for sample counts greater than 65,536, use the following programming sequence to concatenate Counter 4 to Counter 5. The lower 16 bits of the sample count are stored in Counter 4, and the upper 16 bits of the sample count are stored in Counter 5. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF04 to the Am9513A Command Register to select the Counter 4 Mode Register.

b. Write 9025 to the Am9513A Data Register to store the Counter 4 mode value.

c. Write FF0C to the Am9513A Command Register to select the Counter 4 Load Register.

d. Write the least significant 16 bits of the sample count value minus 1 to the Am9513A Data Register to store the Counter 4 load value.

   - If the least significant 16 bits are all 0, write FFFF.

e. Write FF48 to the Am9513A Command Register to load Counter 4.

f. Write 0 to the Am9513A Data Register to store 0 into the Load Register for Counter 4 reloading.

g. Write FF28 to the Am9513A Command Register to arm Counter 4.

h. Write FF05 to the Am9513A Command Register to select the Counter 5 Mode Register.

i. Write 25 to the Am9513A Data Register to store the Counter 5 mode value.

j. Write FF0D to the Am9513A Command Register to select the Counter 5 Load Register.

k. Take the most significant 16 bits of the sample count and do the following:

- If the least significant 16 bits of the sample count are all 0 or all 0 except for a 1 in the least significant bit, write the most significant 16 bits to the Am9513A Data Register to store the Counter 5 load value.
- Otherwise, add 1 to the most significant 16 bits of the sample count and write that value to the Am9513A Data Register to store the Counter 5 load value.

l. Write FF70 to the Am9513A Command Register to load and arm Counter 5.

m. Set the 16*/32 CNT bit in Command Register 1 to notify the hardware that both Counters 4 and 5 will be used as the sample counter.

The only restriction on the sample count in the scanning mode is that the sample count should be programmed as a multiple of the number of entries in the mux-channel-gain memory. Thus, if three channels are to be scanned and five samples per channel are desired, the sample count should be programmed as 15.

After you complete this programming sequence, Counter 4 is configured to count A/D conversion pulses generated by Counter 3, and Counter 5 increments every time Counter 4 reaches 0. The data acquisition operation is terminated when both Counters 4 and 5 reach 0 and the last entry in the mux-channel-gain memory is served. Counter 4 begins counting A/D conversion pulses when a second falling edge signal is received on the EXTTRIG* input or STARTDAQ is written to a second time. A/D conversion data stored before receipt of the EXTTRIG* or STARTDAQ signal are pretrigger samples.

4. Clear the A/D circuitry.

Before starting the data acquisition operation, the FIFO should be emptied to clear out any old conversion results. Emptying must be done after the counters are programmed in case any spurious edges were caused while programming the counters. Write 0 to ADCLEAR to empty the FIFO, then write 0 to the MUXMEMLD Register to load the first channel-gain value.

5. Enable the scanning data acquisition operation.

To enable the scanning data acquisition operation such that A/D conversions begin when a trigger is received, set the DAQEN bit and the SCANEN bit in Command Register 1. Also, make sure INTGATE* in Command Register 1 and EXTGATE* at the I/O connector are disabled.

6. Apply a trigger.

Once set up by the preceding steps, a trigger can be initiated through software or hardware.

To initiate the data acquisition operation through software, write 0 to the STARTDAQ Register (16-bit write). Make sure EXTTRIG* is not pulled low at the I/O connector or the RTSI switch.

To initiate the data acquisition operation through hardware, apply an active low pulse to the EXTTRIG* pin on the AT-MIO-16F-5 I/O connector. See *Data Acquisition Timing Connections* in Chapter 2 for EXTTRIG* signal specifications.

Once the first trigger is applied, Counter 3 generates pulses initiating A/D conversions once every sample interval, but these conversions are not counted by the sample counter. Counting begins after the application of a second hardware or software trigger condition and continues until the sample counter reaches 0.

7. Service the data acquisition operation.

Once the data acquisition operation is started by application of a trigger, the operation must be serviced by reading the FIFO every time an A/D conversion result becomes available. To do this, perform the following sequence until the data acquisition has completed:

a. Read the Status Register (16-bit read).

b. If OVERRUN* or OVERFLOW* are clear, then the data acquisition sequence has been halted because one of these error conditions has occurred. Clear the A/D circuitry by writing to ADCLEAR and determine the cause of the error. OVERRUN* and OVERFLOW* are explained in step 3 of *Programming the Analog Input Circuitry.*

c. If the FIFOEF* bit is set , read the A/D FIFO Register to obtain the result.

Interrupts or DMA can also be used to service the data acquisition operation. These topics are discussed later in this chapter.

Controlling Pretrigger Channel Scanning with the EXTCONV* Signal

Using the EXTCONV* signal to control pretrigger multiple A/D conversions is exactly like a pretrigger data acquisition with channel scanning except step 2 in the programming sequence should be omitted and Counter 3 of the Am9513A Counter/Timer should be left in the high-impedance state instead of programmed. As in the internal pretrigger data acquisition mode, conversions begin upon the application of the first active low pulse of EXTTRIG* or STARTDAQ. Conversions are generated by the high-to-low edge of the EXTCONV* signal. Even though EXTCONV* may be pulsing, conversions do not begin until the application of the first active low pulse of EXTTRIG* or STARTDAQ. Conversions are generated by the high-to-low edge of the EXTCONV* signal. The sample counter does not begin counting the conversions until a second application of the EXTTRIG* or STARTDAQ signals. Conversions are halted when the sample counter expires.

Posttrigger Data Acquisition with Interval Channel Scanning

Programming scanned multiple A/D conversions (pseudo-simultaneous) with a scan interval involves the following programming steps:

1. Set up the analog channel and gain selection sequence.
2. Program the sample-interval counter.
3. Program the sample counter.
4. Program the scan-interval counter.
5. Clear the A/D circuitry and reset the mux counter.
6. Enable the scanning data acquisition operation.
7. Apply a trigger.
8. Service the data acquisition operation.

Setting the SCANEN bit in Command Register 1 enables scanning during multiple A/D conversions. This bit must be set regardless of the type of scanning used; otherwise, only a single channel is scanned. In addition, a channel and gain scan sequence must be stored in the mux-channel-gain memory.

Setting the SCN2 bit in Command Register 1 enables the use of a scan interval during multiple A/D conversions. The scan-interval counter gives each cycle through the scan sequence a time interval. The scan-interval counter begins counting at the start of the scan sequence programmed into the mux-channel-gain memory. When the scan sequence completes, the next cycle through the scan sequence does not begin until the scan-interval counter has reached its terminal count. Be sure that the scan-interval counter allows enough time for all conversions in a scan sequence to occur so that conversions are not missed.

1. Set up the analog channel and gain selection sequence.

During a scanning data acquisition operation, a selected number of locations in the mux-channel-gain memory are stepped through by means of the acquisition circuitry. A new mux-channel-gain memory location is selected after each A/D conversion. The first conversion is performed on the first channel and gain setting in the memory. The second conversion is performed on the second channel and gain setting, and so on. The last entry for each scan sequence written to the mux-channel-gain memory must have the LASTONE bit set. This bit marks the end of a scan sequence. If only one scan sequence is in the mux-channel-gain memory, the circuitry stops after the last conversion is performed and waits the necessary interval time before starting the scan sequence over. If multiple scan sequences are in the mux-channel-gain memory, the circuitry stops at the end of each scan sequence and waits the necessary interval time before starting the next scan sequence in memory. When the end of the scan list is reached, the circuitry stops and waits the necessary interval time before resetting to the beginning of the mux-channel-gain memory and starting the first scan sequence again.

Multiple conversions can be performed on each entry in the mux-channel-gain memory before incrementing to the next entry in the scan sequence. If the SCANDIV bit in Command Register 1 is set, the mux-channel-gain memory increments to the next entry when an active low pulse is detected on the Am9513A Counter/Timer OUT1 signal. If the SCANDIV bit is cleared, the mux-channel-gain memory is incremented to the next entry after every conversion.

The mux-channel-gain memory must be loaded with the desired scan sequences before data acquisition begins. To load the mux-channel-gain memory, perform the following write operations where *X* is the total number of entries in the scan list:

- Write 0 to the MUXMEMCLR Register.
- For $i = 0$ to X-1, use the following steps:

  a. Write the desired analog channel selection and gain setting to the MUXMEM Register (this loads the mux-channel-gain memory at location i).

  b. If this entry is the last entry in a scan sequence, also set the LASTONE bit when writing to the MUXMEM Register.

- Write 0 to the MUXMEMLD Register.

2. Program the sample-interval counter.

Counter 3 of the Am9513A Counter/Timer is used as the sample-interval counter. Counter 3 can be programmed to generate an active low pulse once every *N* counts. *N* is referred to as the sample interval, that is, the time between successive A/D conversions. *N* can be between 2 and 65,536. One count is equal to the period of the timebase clock used by the counter. The following clocks are available internal to the Am9513A: 5-MHz, 1-MHz, 100-kHz, 10-kHz, 1-kHz, and 100-Hz. In addition, the sample-interval timer can use signals connected to any of the Am9513A SOURCE input pins.

To program the sample-interval counter, use the following programming sequence. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF03 to the Am9513A Command Register to select the Counter 3 Mode Register.

b. Write the mode value to the Am9513A Data Register to store the Counter 3 mode value. Use one of the following mode values (Am9513A counter mode information can be found in Appendix C):

   8225 – Selects 5-MHz clock (from SOURCE2 pin)
   8B25 – Selects 1-MHz clock
   8C25 – Selects 100-kHz clock
   8D25 – Selects 10-kHz clock
   8E25 – Selects 1-kHz clock
   8F25 – Selects 100-Hz clock
   8525 – Selects signal at SOURCE5 input as clock (counts the rising edge of the signal, 6 MHz maximum)

c. Write FF0B to the Am9513A Command Register to select the Counter 3 Load Register.

d. Write 2 to the Am9513A Data Register to store the Counter 3 load value.

e. Write FF44 to the Am9513A Command Register to load Counter 3.

f. Write FFF3 to the Am9513A Command Register to step Counter 3 down to 1.

g. Write the desired sample interval to the Am9513A Data Register to store the Counter 3 load value:

   - If the sample interval is between 2 and FFFF (65,535 decimal), write the sample interval to the Am9513A Data Register.
   - If the sample interval is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.
   - If the sample interval is greater than 10000.

h. Write FF24 to the Am9513A Command Register to arm Counter 3.

After you complete this programming sequence, Counter 3 is configured to generate A/D conversion pulses as soon as application of a trigger causes it to be enabled.

3. Program the sample counter.

Counters 4 and 5 of the Am9513A Counter/Timer are used as the sample counter. The sample counter tallies the number of A/D conversions initiated by Counter 3 and stops Counter 3 when the desired sample count is reached. If the desired sample count is 65,536 or less, only Counter 4 needs to be used, making Counter 5 available for general-purpose timing applications. If the desired sample count is greater than 65,536, both Counters 4 and 5 must be used.

Sample Counts 2 through 65,536.

To program the sample counter for sample counts up to 65,536, use the following programming sequence. The minimum permitted sample count is 2. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF04 to the Am9513A Command Register to select the Counter 4 Mode Register.

b. Write 1025 to the Am9513A Data Register to store the Counter 4 mode value.

c. Write FF0C to the Am9513A Command Register to select the Counter 4 Load Register.

d. Write the sample count value to the Am9513A Data Register to store the Counter 4 load value:

   - If the sample count is between 2 and FFFF (65,535 decimal), write the sample count to the Am9513A Data Register.
   - If the sample count is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.

e. Write FF48 to the Am9513A Command Register to load Counter 4.

f. Write FFF4 to the Am9513A Command Register to decrement Counter 4.

g. Write FF28 to the Am9513A Command Register to arm Counter 4.

h. Clear the 16*/32 CNT bit in Command Register 1 to notify the hardware that only Counter 4 will be used as the sample counter.

After you complete this programming sequence, Counter 4 is configured to count A/D conversion pulses generated by Counter 3 and turns off the data acquisition operation when Counter 4 decrements to 0.

Sample Counts Greater than 65,536.

To program the sample counter for sample counts greater than 65,536, use the following programming sequence to concatenate Counter 4 to Counter 5. The lower 16 bits of the sample count are stored in Counter 4, and the upper 16 bits of the sample count are stored in Counter 5. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF04 to the Am9513A Command Register to select the Counter 4 Mode Register.

b. Write 1025 to the Am9513A Data Register to store the Counter 4 mode value.

c. Write FF0C to the Am9513A Command Register to select the Counter 4 Load Register.

d. Write the least significant 16 bits of the sample count value minus 1 to the Am9513A Data Register to store the Counter 4 load value.

   - If the least significant 16 bits are all 0, write FFFF.

e. Write FF48 to the Am9513A Command Register to load Counter 4.

f. Write 0 to the Am9513A Data Register to store 0 into the Load Register for Counter 4 reloading.

g. Write FF28 to the Am9513A Command Register to arm Counter 4.

h. Write FF05 to the Am9513A Command Register to select the Counter 5 Mode Register.

i. Write 25 to the Am9513A Data Register to store the Counter 5 mode value.

j. Write FF0D to the Am9513A Command Register to select the Counter 5 Load Register.

k. Take the most significant 16 bits of the sample count and do the following:

- If the least significant 16 bits of the sample count are all 0 or all 0 except for a 1 in the least significant bit, write the most significant 16 bits to the Am9513A Data Register to store the Counter 5 load value.
- Otherwise, add 1 to the most significant 16 bits of the sample count and write that value to the Am9513A Data Register to store the Counter 5 load value.

l. Write FF70 to the Am9513A Command Register to load and arm Counter 5.

m. Set the 16*/32 CNT bit in Command Register 1 to notify the hardware that both Counters 4 and 5 will be used as the sample counter.

The only restriction on the sample count in the scanning mode is that the sample count should be programmed as a multiple of the number of entries in the mux-channel-gain memory. Thus, if three channels are to be scanned and five samples per channel are desired, the sample count should be programmed as 15.

After you complete this programming sequence, Counter 4 is configured to count A/D conversion pulses generated by Counter 3, and Counter 5 increments every time Counter 4 reaches 0. The data acquisition operation is terminated when both Counters 4 and 5 reach 0 and the last entry in the mux-channel-gain memory is served.

4. Program the scan-interval counter.

Counter 2 of the Am9513A Counter/Timer is used as the scan-interval counter. Counter 2 can be programmed to generate a pulse once every *N* counts. *N* is referred to as the scan interval, which is the time between successive scan sequences programmed into the mux-channel-gain memory. *N* can be between 2 and 65,536. One count is equal to the period of the timebase clock used by the counter. The following clocks are available internal to the Am9513A: 5-MHz, 1-MHz, 100-kHz,
10-kHz, 1-kHz, and 100-Hz. In addition, the scan-interval timer can use signals connected to any of the Am9513A SOURCE input pins.

To program the scan-interval counter, use the following programming sequence. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF02 to the Am9513A Command Register to select the Counter 2 Mode Register.

b. Write the mode value to the Am9513A Data Register to store the Counter 2 mode value. Use one of the following mode values:

8225 – Selects 5-MHz clock (Counter 2 Source signal)
8B25 – Selects 1-MHz clock
8C25 – Selects 100-kHz clock
8D25 – Selects 10-kHz clock
8E25 – Selects 1-kHz clock
8F25 – Selects 100-Hz clock
8*n*25 – Selects signal at SOURCE*n* input as clock where $1 \leq n \leq 5$ (counts the rising edge of the signal, 6 MHz maximum)

c. Write FF0A to the Am9513A Command Register to select the Counter 2 Load Register.

d. Write 2 to the Am9513A Data Register to store the Counter 2 load value.

e. Write FF42 to the Am9513A Command Register to load Counter 2.

f. Write FFF2 to the Am9513A Command Register to step Counter 2 down to 1.

g. Entries stored in the mux-channel-gain memory should be scanned once during a scan interval. The following condition must be satisfied:

scan interval ≥ sample interval ∗ *x*, where *x* is the number of entries in the scan sequence.

Write the desired scan interval to the Am9513A Data Register to store the Counter 2 load value:

- If the scan interval is between 2 and FFFF (65,535 decimal), write the scan interval to the Am9513A Data Register.
- If the scan interval is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.

h. Write FF22 to the Am9513A Command Register to arm Counter 2.

After you complete this programming sequence, Counter 2 is configured to assign a time interval to scan sequences once the trigger to enable A/D conversions is detected.

5. Clear the A/D circuitry.

Before starting the data acquisition operation, the FIFO should be emptied to clear out any old conversion results. Emptying must be done after the counters are programmed in case any spurious edges were caused while programming the counters. Write 0 to ADCLEAR to empty the FIFO, then write 0 to the MUXMEMLD Register to load the first channel-gain value.

6. Enable the scanning data acquisition operation.

To enable the scanning interval data acquisition operation such that A/D conversions begin when a trigger is received, set the DAQEN, SCANEN and SCN2 bits in Command Register 1. Make sure INTGATE* in Command Register 1 and EXTGATE* at the I/O connector are disabled.

7. Apply a trigger.

Once set up by the preceding steps, the data acquisition operation is initiated when a trigger is received. A trigger can be initiated through software or through hardware.

To initiate the data acquisition operation through software, write 0 to the Start DAQ Register (16-bit write). Make sure EXTTRIG* is not pulled low at the I/O connector or the RTSI switch.

To initiate the data acquisition operation through hardware, apply an active low pulse to the EXTTRIG* pin on the AT-MIO-16F-5 I/O connector. See *Data Acquisition Timing Connections* in Chapter 2 for EXTTRIG* signal specifications.

Once the trigger is applied, Counter 3 generates pulses initiating A/D conversions once every sample interval until the sample counter reaches 0.

8. Service the data acquisition operation.

Once the data acquisition operation is started by application of a trigger, the operation must be serviced by reading the FIFO every time an A/D conversion result becomes available. To do this, perform the following sequence until the data acquisition has completed:

a. Read the Status Register (16-bit read).

b. If OVERRUN* or OVERFLOW* are clear, then the data acquisition sequence has been halted because one of these error conditions has occurred. Clear the A/D circuitry by writing to ADCLEAR and determine the cause of the error. OVERRUN* and OVERFLOW* are explained in step 3 of *Programming the Analog Input Circuitry.*

c. If the FIFOEF* bit is set , read the A/D FIFO Register to obtain the result.

Interrupts or DMA can also be used to service the data acquisition operation. These topics are discussed later in this chapter.

Controlling Posttrigger Interval Channel Scanning with the EXTCONV* Signal

Using the EXTCONV* signal to round robin data acquisition sequences is exactly like a posttrigger interval channel scanning sequence except step 2 in the programming sequence should be omitted and Counter 3 should be left in the high impedance state. Conversions are generated from the falling edge of the EXTCONV* signal. Even though EXTCONV* may be

pulsing, conversions do not begin until after an active low pulse on STARTDAQ or the EXTTRIG* signal is detected. Conversions are automatically halted when the sample counter reaches zero, irrespective of the EXTCONV* signal.

Pretrigger Data Acquisition with Interval Channel Scanning

Programming scanned multiple A/D conversions (pseudo-simultaneous) with a scan interval involves the following programming steps:

1. Set up the analog channel and gain selection sequence.
2. Program the sample-interval counter.
3. Program the sample counter.
4. Program the scan-interval counter.
5. Clear the A/D circuitry and reset the mux counter.
6. Enable the scanning data acquisition operation.
7. Apply a trigger.
8. Service the data acquisition operation.

Setting the SCANEN bit in Command Register 1 enables scanning during multiple A/D conversions. This bit must be set regardless of the type of scanning used; otherwise, only a single channel is scanned. In addition, a channel and gain scan sequence must be stored in the mux-channel-gain memory.

Setting the SCN2 bit in Command Register 2 enables the use of a scan interval during multiple A/D conversions. The scan-interval counter gives each cycle through the scan sequence a time interval. The scan-interval counter begins counting at the start of the scan sequence programmed into the mux-channel-gain memory. When the scan sequence completes, the next cycle through the scan sequence does not begin until the scan-interval counter has reached its terminal count. Be sure that the scan-interval counter allows enough time for all conversions in a scan sequence to occur so that conversions are not missed.

1. Set up the analog channel and gain selection sequence.

During a scanning data acquisition operation, a selected number of locations in the mux-channel-gain memory are stepped through by means of the acquisition circuitry. A new mux-channel-gain memory location is selected after each A/D conversion. The first conversion is performed on the first channel and gain setting in the memory. The second conversion is performed on the second channel and gain setting, and so on. The last entry for each scan sequence written to the mux-channel-gain memory must have the LASTONE bit set. This bit marks the end of a scan sequence. If only one scan sequence is in the mux-channel-gain memory, the circuitry stops after the last conversion is performed and waits the necessary interval time before starting the scan sequence over. If multiple scan sequences are in the mux-channel-gain memory, the circuitry stops at the end of each scan sequence and waits the necessary interval time before starting the next scan sequence in memory. When the end of the scan list is reached, the circuitry stops and

waits the necessary interval time before resetting to the beginning of the mux-channel-gain memory and starting the first scan sequence again.

Multiple conversions can be performed on each entry in the mux-channel-gain memory before incrementing to the next entry in the scan sequence. If the SCANDIV bit in Command Register 1 is set, the mux-channel-gain memory increments to the next entry when an active low pulse is detected on the Am9513A Counter/Timer OUT1 signal. If the SCANDIV bit is cleared, the mux-channel-gain memory is incremented to the next entry after every conversion.

The mux-channel-gain memory must be loaded with the desired scan sequences before data acquisition begins. To load the mux-channel-gain memory, perform the following write operations where *X* is the total number of entries in the scan list:

- Write 0 to the MUXMEMCLR Register.
- For $i = 0$ to X-1, use the following steps:
  a. Write the desired analog channel selection and gain setting to the MUXMEM Register (this loads the mux-channel-gain memory at location *i*).
  b. If this entry is the last entry in a scan sequence, also set the LASTONE bit when writing to the MUXMEM Register.
- Write 0 to the MUXMEMLD Register.

2. Program the sample-interval counter.

Counter 3 of the Am9513A Counter/Timer is used as the sample-interval counter. Counter 3 can be programmed to generate an active low pulse once every *N* counts. *N* is referred to as the sample interval, that is, the time between successive A/D conversions. *N* can be between 2 and 65,536. One count is equal to the period of the timebase clock used by the counter. The following clocks are available internal to the Am9513A: 5-MHz, 1-MHz, 100-kHz, 10-kHz, 1-kHz, and 100-Hz. In addition, the sample-interval timer can use signals connected to any of the Am9513A SOURCE input pins.

To program the sample-interval counter, use the following programming sequence. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF03 to the Am9513A Command Register to select the Counter 3 Mode Register.

b. Write the mode value to the Am9513A Data Register to store the Counter 3 mode value. Use one of the following mode values (Am9513A counter mode information can be found in Appendix C):

8225 – Selects 5-MHz clock (from SOURCE2 pin)
8B25 – Selects 1-MHz clock
8C25 – Selects 100-kHz clock
8D25 – Selects 10-kHz clock
8E25 – Selects 1-kHz clock
8F25 – Selects 100-Hz clock
8525 – Selects signal at SOURCE5 input as clock (counts the rising edge of the signal, 6 MHz maximum)

c. Write FF0B to the Am9513A Command Register to select the Counter 3 Load Register.

d. Write 2 to the Am9513A Data Register to store the Counter 3 load value.

e. Write FF44 to the Am9513A Command Register to load Counter 3.

f. Write FFF3 to the Am9513A Command Register to step Counter 3 down to 1.

g. Write the desired sample interval to the Am9513A Data Register to store the Counter 3 load value):

- If the sample interval is between 2 and FFFF (65,535 decimal), write the sample interval to the Am9513A Data Register.
- If the sample interval is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.
- If the sample interval is greater than 10000.

h. Write FF24 to the Am9513A Command Register to arm Counter 3.

After you complete this programming sequence, Counter 3 is configured to generate A/D conversion pulses as soon as application of a trigger causes it to be enabled.

3. Program the sample counter.

Counters 4 and 5 of the Am9513A Counter/Timer are used as the sample counter. The sample counter tallies the number of A/D conversions initiated by Counter 3 and stops Counter 3 when the desired sample count is reached. If the desired sample count is 65,536 or less, only Counter 4 needs to be used, making Counter 5 available for general-purpose timing applications. If the desired sample count is greater than 65,536, both Counters 4 and 5 must be used.

Sample Counts 2 through 65,536.

To program the sample counter for sample counts up to 65,536, use the following programming sequence. The minimum permitted sample count is 2. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF04 to the Am9513A Command Register to select the Counter 4 Mode Register.

b. Write 9025 to the Am9513A Data Register to store the Counter 4 mode value.

c. Write FF0C to the Am9513A Command Register to select the Counter 4 Load Register.

d. Write the sample count value to the Am9513A Data Register to store the Counter 4 load value:

   - If the sample count is between 2 and FFFF (65,535 decimal), write the sample count to the Am9513A Data Register.

   - If the sample count is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.

e. Write FF48 to the Am9513A Command Register to load Counter 4.

f. Write FFF4 to the Am9513A Command Register to decrement Counter 4.

g. Write FF28 to the Am9513A Command Register to arm Counter 4.

h. Clear the 16*/32 CNT bit in Command Register 1 to notify the hardware that only Counter 4 will be used as the sample counter.

After you complete this programming sequence, Counter 4 is configured to count A/D conversion pulses generated by Counter 3 and turns off the data acquisition operation when Counter 4 decrements to 0.

Sample Counts Greater than 65,536.

To program the sample counter for sample counts greater than 65,536, use the following programming sequence to concatenate Counter 4 to Counter 5. The lower 16 bits of the sample count are stored in Counter 4, and the upper 16 bits of the sample count are stored in Counter 5. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF04 to the Am9513A Command Register to select the Counter 4 Mode Register.

b. Write 9025 to the Am9513A Data Register to store the Counter 4 mode value.

c. Write FF0C to the Am9513A Command Register to select the Counter 4 Load Register.

d. Write the least significant 16 bits of the sample count value minus 1 to the Am9513A Data Register to store the Counter 4 load value.

   - If the least significant 16 bits are all 0, write FFFF.

e. Write FF48 to the Am9513A Command Register to load Counter 4.

f. Write 0 to the Am9513A Data Register to store 0 into the Load Register for Counter 4 reloading.

g. Write FF28 to the Am9513A Command Register to arm Counter 4.

h. Write FF05 to the Am9513A Command Register to select the Counter 5 Mode Register.

i. Write 25 to the Am9513A Data Register to store the Counter 5 mode value.

j. Write FF0D to the Am9513A Command Register to select the Counter 5 Load Register.

k. Take the most significant 16 bits of the sample count and do the following:

   - If the least significant 16 bits of the sample count are all 0 or all 0 except for a 1 in the least significant bit, write the most significant 16 bits to the Am9513A Data Register to store the Counter 5 load value.

   - Otherwise, add 1 to the most significant 16 bits of the sample count and write that value to the Am9513A Data Register to store the Counter 5 load value.

l. Write FF70 to the Am9513A Command Register to load and arm Counter 5.

m. Set the 16*/32 CNT bit in Command Register 1 to notify the hardware that both Counters 4 and 5 will be used as the sample counter.

The only restriction on the sample count in the scanning mode is that the sample count should be programmed as a multiple of the number of entries in the mux-channel-gain memory. Thus, if three channels are to be scanned and five samples per channel are desired, the sample count should be programmed as 15. Also modify the sequence by replacing 1025 (hex) with 9025 (hex) in step b both for counts greater than and less than 65536 (dec) as the value to store in the Counter 4 mode register.

After you complete this programming sequence, Counter 4 is configured to count A/D conversion pulses generated by Counter 3, and Counter 5 increments every time Counter 4 reaches 0. The data acquisition operation is terminated when both Counters 4 and 5 reach 0 and the last entry in the mux-channel-gain memory is served. Counter 4 begins counting A/D conversion pulses when a second falling edge signal is received on the EXTTRIG* input or STARTDAQ is written to for a second time. A/D conversion data stored before receipt of the EXTTRIG* or STARTDAQ signal are pretrigger samples.

4. Program the scan-interval counter.

Counter 2 of the Am9513A Counter/Timer is used as the scan-interval counter. Counter 2 can be programmed to generate a pulse once every *N* counts. *N* is referred to as the scan interval, which is the time between successive scan sequences programmed into the mux-channel-gain memory. *N* can be between 2 and 65,536. One count is equal to the period of the timebase clock used by the counter. The following clocks are available internal to the Am9513A: 5-MHz, 1-MHz, 100-kHz,
10-kHz, 1-kHz, and 100-Hz. In addition, the scan-interval timer can use signals connected to any of the Am9513A SOURCE input pins.

To program the scan-interval counter, use the following programming sequence. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF02 to the Am9513A Command Register to select the Counter 2 Mode Register.

b. Write the mode value to the Am9513A Data Register to store the Counter 2 mode value. Use one of the following mode values:

   8225 – Selects 5-MHz clock (Counter 2 Source signal)
   8B25 – Selects 1-MHz clock
   8C25 – Selects 100-kHz clock
   8D25 – Selects 10-kHz clock
   8E25 – Selects 1-kHz clock
   8F25 – Selects 100-Hz clock
   8525 – Selects signal at SOURCE5 input as clock (counts the rising edge of the signal, 6 MHz maximum)

c. Write FF0A to the Am9513A Command Register to select the Counter 2 Load Register.

d. Write 2 to the Am9513A Data Register to store the Counter 2 load value.

e. Write FF42 to the Am9513A Command Register to load Counter 2.

f. Write FFF2 to the Am9513A Command Register to step Counter 2 down to 1.

g. Entries stored in the mux-channel-gain memory should be scanned once during a scan interval. The following condition must be satisfied:

   scan interval ≥ sample interval ∗ *x*, where *x* is the number of entries in the scan sequence.

   Write the desired scan interval to the Am9513A Data Register to store the Counter 2 load value:

   - If the scan interval is between 2 and FFFF (65,535 decimal), write the scan interval to the Am9513A Data Register.

   - If the scan interval is 10000 (65,536 decimal), write 0 to the Am9513A Data Register.

h. Write FF22 to the Am9513A Command Register to arm Counter 2.

After you complete this programming sequence, Counter 2 is configured to assign a time interval to scan sequences once the trigger to enable A/D conversions is detected.

5. Clear the A/D circuitry.

Before starting the data acquisition operation, the FIFO should be emptied to clear out any old conversion results. Emptying must be done after the counters are programmed in case any spurious edges were caused while programming the counters. Write 0 to ADCLEAR to empty the FIFO, then write 0 to the MUXMEMLD Register to load the first channel-gain value.

6. Enable the scanning data acquisition operation.

To enable the scanning interval data acquisition operation such that A/D conversions begin when a trigger is received, set the DAQEN, SCANEN and SCN2 bits in Command Register 1. Also, make sure INTGATE* in Command Register and EXTGATE* at the I/O connector are disabled.

7. Apply a trigger.

Once set up by the preceding steps, a trigger can be initiated through software or hardware.

To initiate the data acquisition operation through software, write 0 to the STARTDAQ Register (16-bit write). Make sure EXTTRIG* is not pulled low at the I/O connector or the RTSI switch.

To initiate the data acquisition operation through hardware, apply an active low pulse to the EXTTRIG* pin on the AT-MIO-16F-5 I/O connector. See *Data Acquisition Timing Connections* in Chapter 2 for EXTTRIG* signal specifications.

Once the first trigger is applied, Counter 3 generates pulses initiating A/D conversions once every sample interval, but these conversions are not counted by the sample counter. Counting begins after the application of a second hardware or software trigger condition and continues until the sample counter reaches 0.

8. Service the data acquisition operation.

Once the data acquisition operation is started by application of a trigger, the operation must be serviced by reading the FIFO every time an A/D conversion result becomes available. To do this, perform the following sequence until the data acquisition has completed:

a. Read the Status Register (16-bit read).

b. If OVERRUN* or OVERFLOW* are clear, then the data acquisition sequence has been halted because one of these error conditions has occurred. Clear the A/D circuitry by writing to ADCLEAR and determine the cause of the error. OVERRUN* and OVERFLOW* are explained in step 3 of *Programming the Analog Input Circuitry.*

c. If the FIFOEF* bit is set , read the A/D FIFO Register to obtain the result.

Interrupts or DMA can also be used to service the data acquisition operation. These topics are discussed later in this chapter.

Controlling Pretrigger Interval Channel Scanning with the EXTCONV* Signal

Using the EXTCONV* signal to control pretrigger multiple A/D conversions is exactly like a pretrigger multiple channel data acquisition sequence except step 2 in the programming sequence should be omitted and Counter 3 of the Am9513A Counter/Timer should be left in the high-impedance state instead of programmed. As in the internal pretrigger data acquisition mode, conversions begin upon the application of the first active-low pulse of EXTTRIG* or STARTDAQ. Conversions are generated by the high-to-low edge of the EXTCONV* signal. Even though EXTCONV* may be pulsing, conversions do not begin until the application of the first active low pulse of EXTTRIG* or STARTDAQ. Conversions are generated by the high-to-low edge of the EXTCONV* signal. The sample counter does not begin counting the conversions until a second application of the EXTTRIG* or STARTDAQ signals. Conversions are halted when the sample counter expires.

Resetting the Hardware after a Data Acquisition Operation

After a data acquisition operation is complete, if no errors occurred and the sample count was less than or equal to 10,000 hex, then the AT-MIO-16F-5 is left in the same state as it was at the beginning of the data acquisition operation. The counters do not need to be reprogrammed; another data acquisition operation begins when a trigger is received. If the next data acquisition operation requires the counters to be programmed differently, the Am9513A counters that were used must be disarmed and reset.

Resetting Counter 2

To reset Counter 2, use the following programming sequence. All writes are 16-bit operations. All values given are hexadecimal.

1. Write FFC2 to the Am9513A Command Register to disarm Counter 2.

2. Write FF02 to the Am9513A Command Register to select the Counter 2 Mode Register.

3. Write 4 to the Am9513A Data Register to store the Counter 2 mode value such that counter output becomes high-impedance.

4. Write FF0A to the Am9513A Command Register to select the Counter 2 Load Register.

5. Write 3 to the Am9513A Data Register to store non-terminal count value in the Counter 2 Load Register.

6. Write FF42 to the Am9513A Command Register to load Counter 2.

7. Write FF42 to the Am9513A Command Register a second time to load Counter 2 again to guarantee that Counter 2 is not left in a terminal count state.

Resetting Counter 3

To reset Counter 3, use the following programming sequence. All writes are 16-bit operations. All values given are hexadecimal.

1. Write FFC4 to the Am9513A Command Register to disarm Counter 3.

2. Write FF03 to the Am9513A Command Register to select the Counter 3 Mode Register.

3. Write 4 to the Am9513A Data Register to store the Counter 3 mode value such that counter output becomes high-impedance.

4. Write FF0B to the Am9513A Command Register to select the Counter 3 Load Register.

5. Write 3 to the Am9513A Data Register to store non-terminal count value in the Counter 3 Load Register.

6. Write FF44 to the Am9513A Command Register to load Counter 3).

7. Write FF44 to the Am9513A Command Register a second time to load Counter 3 again to guarantee that Counter 3 is not left in a terminal count state.

Resetting Counter 4

To reset Counter 4, use the following programming sequence. All writes are 16-bit operations. All values given are hexadecimal.

1. Write FFC8 to the Am9513A Command Register to disarm Counter 4.

2. Write FF04 to the Am9513A Command Register to select the Counter 4 Mode Register.

3. Write 4 to the Am9513A Data Register to store the Counter 4 mode value such that counter output becomes high-impedance. If Counter 4 is not to be used during the next data acquisition operation, write 0 to the Am9513A Data Register to drive the output low.

4. Write FF0C to the Am9513A Command Register to select the Counter 4 Load Register.

5. Write 3 to the Am9513A Data Register to store non-terminal count value in the Counter 4 Load Register.
6. Write FF48 to the Am9513A Command Register to load Counter 4.
7. Write FF48 to the Am9513A Command Register a second time to load Counter 4 again to guarantee that Counter 4 is not left in a terminal count state.

Resetting Counter 5

To reset Counter 5, use the following programming sequence. All writes are 16-bit operations. All values given are hexadecimal.

1. Write FFD0 to the Am9513A Command Register to disarm Counter 5.
2. Write FF05 to the Am9513A Command Register to select the Counter 5 Mode Register.
3. Write 4 to the Am9513A Data Register to store the Counter 5 mode value such that counter output becomes high-impedance.
4. Write FF0D to the Am9513A Command Register to select the Counter 5 Load Register.
5. Write 3 to the Am9513A Data Register to store non-terminal count value in the Counter 5 Load Register.
6. Write FF50 to the Am9513A Command Register to load Counter 5.
7. Write FF50 to the Am9513A Command Register a second time to load Counter 5 again to guarantee that Counter 5 is not left in a terminal count state.

After resetting the counters, write 0 to the A/D Clear Register to clear all error conditions and to empty the A/D FIFO.

Programming the Analog Output Circuitry

The voltage at the analog output circuitry output pins (pins DAC0 OUT and DAC1 OUT on the AT-MIO-16F-5 I/O connector) is controlled by loading the DAC in the analog output channel with a 12-bit digital code. This DAC is loaded by writing the digital code to the DAC0 and DAC1 Registers. Writing to the DAC0 Register controls the voltage at the DAC0 OUT pin on the AT-MIO-16F-5 I/O connector. Writing to the DAC1 Register controls the voltage at the DAC1 OUT pin. The analog output on pins DAC0 OUT and DAC1 OUT can be updated in one of three ways: immediately when DAC0 or DAC1 is written, when an active low pulse is detected on the DACUPTRIG* signal, or when the DAC Update Register is strobed. The WGEN bit in Command Register 2 selects which update method is used.

Updating the DACs using timer waveform generation (WGEN in Command Register 2) can be handled using either interrupts or DMA requests. Upon the application of a falling edge signal to the DACUPTRIG* signal, both DACs are updated and DACUP in the Status Register is set and a DMA or INT interrupt request is generated, if enabled. DACUPTRIG* can be connected to selected internal signals on the RTSI bus with A4RCV enabled or the external signal EXTDACUPDATE* with A4RCV disabled. If interrupts are enabled with the proper mode bits in Command Register 2, an interrupt occurs when DACUP is set. In interrupt mode, DACUP

must be cleared using the DAC Update INT Clear Register before exiting the interrupt routine. This clears the interrupt request. Using DMA to service DAC transfers is selected using the DMAEN and DMAINTB2 through DMAINTB0 bits in Command Register 2. DMA requests are generated when DACUP is set. When the DMA controller acknowledges the request, DACUP is automatically cleared.

The DACUPERR signal indicates an update error when using timer waveform generation. If DACUP is set when another update occurs, DACUPERR becomes asserted. DACUPERR occurs in interrupt mode if DACUP is not cleared in the interrupt routine. In interrupt and DMA modes, DACUPERR occurs for rates above the maximum rate of the DMA controller or interrupt handling capabilities. DACUPERR is cleared by writing to the DAC Update INT Clear Register.

The output voltage generated from the digital code depends on the configuration, unipolar or bipolar, of the associated analog output channel. This configuration is determined by configuration jumpers on the AT-MIO-16F-5 board. Configuration bits in Command Register 2 determine if the digital code written to the DACs is in straight binary form or in a two's complement form. See *Analog Input Configuration* in Chapter 2 for more information. Table 4-6 shows the output voltage versus digital code for a unipolar analog output configuration. Table 4-7 shows the voltage versus digital code for a bipolar analog output configuration.

The formula for the voltage output versus digital code for a unipolar analog output configuration is as follows:

$$V_{out} = V_{ref} * \frac{\text{(digital code)}}{4096}$$

where V_{ref} is the reference voltage applied to the analog output channel. The digital code in the above formula is a decimal value ranging from 0 to 4095.

Table 4-6. Analog Output Voltage Versus Digital Code (Unipolar Mode)

Digital Code		**Voltage Output**	
Decimal	**Hex**	$\mathbf{V_{ref = 10\ V}}$	$\mathbf{V_{out}}$
0	0	0	0 V
1	1	$\frac{Vref}{4096}$	2.44 mV
1024	0400	$\frac{Vref}{4}$	2.5 V
2048	0800	$\frac{Vref}{2}$	5 V
3072	0C00	$\frac{Vref * 3}{4}$	7.5 V
4095	0FFF	$\frac{Vref * 4095}{4096}$	9.9976 V

The formula for the voltage output versus digital code for a bipolar analog output configuration in straight binary form is as follows:

$$V_{out} = V_{ref} * \frac{(digital\ code - 2048)}{2048}$$

where V_{ref} is the reference voltage applied to the analog output channel. The digital code in the preceding formula is a decimal value ranging from 0 to 4095.

The formula for the voltage output versus digital code for a bipolar analog output configuration in two's complement form is as follows:

$$V_{out} = V_{ref} * \frac{(digital\ code)}{2048}$$

where V_{ref} is the positive reference voltage applied to the analog output channel. The digital code in the preceding formula is a decimal value ranging from -2048 to +2047.

Table 4-7. Analog Output Voltage Versus Digital Code (Bipolar Mode)

Digital Code				**Voltage Output**	
Straight Binary		**Two's Complement**			
Decimal	**Hex**	**Decimal**	**Hex**	**Reference = V_{ref}**	**Reference = 10V**
0	0	-2048	F800	V_{ref}	-10 V
1	1	-2047	F801	$\frac{Vref * (-2047)}{2048}$	-9.9951 V
1024	0400	-1024	FC00	$\frac{-Vref}{2}$	-5 V
2047	07FF	-1	FFFF	$\frac{-Vref}{2048}$	-4.88 mV
2048	0800	0	0	0	0 V
2049	0801	1	1	$\frac{Vref}{2048}$	4.88 mV
3072	0C00	1024	0400	$\frac{Vref}{2}$	5 V
4095	0FFF	2047	07FF	$\frac{Vref * 2047}{2048}$	9.9951 V

External DAC Updating

To use EXTDACUPDATE* from the I/O connector to update the analog output DACs in timer waveform generation mode, the following sequence of programming steps must be followed:

1. Clear the analog output circuitry.
2. Enable external updating.
3. Enable the waveform generation mode.
4. Service update requests.

1. Clear the analog output circuitry.

This involves clearing the DACUP, DACUPERR, and DMATCA or DMATCB bits in the Status Register. To do this, write to the DAC Update INT Clear and DMA TC INT Clear Event Strobe Registers.

2. Enable external updating.

Disable A4RCV in Command Register 2. This routes the EXTDACUPDATE* signal to the DACUPTRIG* signal.

3. Enable the waveform generation mode.

Enable WGEN in Command Register 2. This allows the analog output DACs to be updated when a falling edge is detected on the DACUPTRIG* signal.

4. Service update requests.

The most general method, and least reliable, is programmed I/O, or polling the Status Register until DACUP is set. When DACUP becomes set, this indicates that the next values to the DACs can be written. These new values will not actually be updated on the analog output pins until the next falling edge of EXTDACUPDATE*. After detecting the DACUP signal, it must be cleared by writing to the DAC Update INT Clear Register.

The second method is to use interrupt I/O. To use this mode, INTEN and the appropriate mode bits 2 through 0 must be enabled in Command Register 2 before WGEN is enabled. When DACUP is set, an interrupt is generated. Upon entering the interrupt routine, DACUP should be checked to confirm the source of the interrupt. If DACUP is set, the next values to the DACs can be written. Before exiting the interrupt routine, DACUP should be cleared using the DAC Update INT Clear Register. See *Interrupt Programming* later in this chapter. If DACUPERR is set, an error has occurred.

The most efficient waveform generation method is to use DMA transfers. In this case, before WGEN is enabled, the DMA controller on the AT computer must be programmed appropriately and DMAEN and the necessary mode bits 2 through 0 must be enabled. When WGEN is enabled, transfers are automatically handled by the DMA controller. See *Programming DMA Operations* later in this chapter. If DACUPERR is set, an error has occurred.

Internal DAC Updating

To use internal signals from the RTSI switch to update the analog output DACs in timer waveform generation mode, the following sequence of programming steps must be followed:

1. Clear the analog output circuitry.

2. Program the update interval counter.

3. Program the RTSI switch.

4. Enable internal updating.

5. Enable the waveform generation mode.

6. Service update requests.

1. Clear the analog output circuitry.

This involves clearing the DACUP, DACUPERR, and DMATCA or DMATCB bits in the Status Register. To do this, write to the DAC Update Clear and DMA TC INT Clear Event Strobe Registers.

2. Program the update interval counter.

Select the appropriate counter (1, 2, or 5) from the Am9513A Counter/Timer to be used for updating the DACs. Active low pulsing and no gating should be part of the mode programmed. To program the update-interval counter, use the following programming sequence. All writes are 16-bit operations. All values given are hexadecimal.

a. Write FF0X to the Am9513A Command Register to select the Counter X Mode Register.

b. Write the mode value to the Am9513A Data Register to store the Counter X mode value. Use one of the following mode values (Am9513A counter mode information can be found in Appendix C):

   0225 – Selects 5-MHz clock (from SOURCE2 pin)
   0B25 – Selects 1-MHz clock
   0C25 – Selects 100-kHz clock
   0D25 – Selects 10-kHz clock
   0E25 – Selects 1-kHz clock
   0F25 – Selects 100-Hz clock
   0525 – Selects signal at SOURCE5 input as clock (counts the rising edge of the signal, 6 MHz maximum)

c. Write (FF08 + X) to the Am9513A Command Register to select the Counter X Load Register.

d. Write the desired update interval to the Am9513A Data Register to store the Counter X load value.

e. Write the following value to the Am9513A Command Register to load and arm Counter X:

FF61 – Load and arm Counter 1
FF62 – Load and arm Counter 2
FF64 – Load and arm Counter 3
FF68 – Load and arm Counter 4
FF70 – Load and arm Counter 5

After you complete this programming sequence, Counter X is configured to generate active-low pulses as soon as the load/arm counter command is written.

3. Program the RTSI switch.

Select the desired signal at the RTSI switch to be used for updating the DACs. OUT1, OUT2, and OUT5 are the most logical signals to use. To route these update signals, the A side pin of the RTSI switch must be internally routed to the B side, or trigger side. Select a trigger line that is not being used. Then the signal must be routed from the selected B side trigger line to the A4 pin on the RTSI switch. All of this is done in one programming sequence by shifting a 56-bit value to the RTSI switch. See *RTSI Bus Trigger Line Programming Considerations* later in this chapter.

Notice that if OUT5 is to be used for updating, it does not need to be routed across the RTSI switch. In this case only is it sufficient to enable A4DRV to drive pin A4 of the RTSI switch with OUT5.

4. Enable internal updating.

Enable A4RCV in Command Register 2. This routes the selected internal signal from A4 of the RTSI switch to the DACUPTRIG* signal.

5. Enable the waveform generation mode.

Enable WGEN in Command Register 2. This allows the analog output DACs to be updated when a falling edge is detected on the DACUPTRIG* signal.

6. Service update requests.

The most general method, and least reliable, is programmed I/O, or polling the Status Register until DACUP is set. When DACUP becomes set, this indicates that the next values to the DACs can be written. These new values will not actually be updated on the analog output pins until the next falling edge of EXTDACUPDATE*. After detecting the DACUP signal, it must be cleared by writing to the DAC Update INT Clear Register.

The second method is to use interrupt I/O. To use this mode, INTEN and the appropriate mode bits 2 through 0 must be enabled in Command Register 2 before WGEN is enabled. When DACUP is set, an interrupt is generated. Upon entering the interrupt routine, DACUP should be checked to confirm the source of the interrupt. If DACUP is set, the next values to the DACs can be written. Before exiting the interrupt routine, DACUP should be cleared using the DAC Update INT Clear Register. See *Interrupt Programming* later in this chapter. If DACUPERR is set, an error has occurred.

The most efficient waveform generation method is to use DMA transfers. In this case, before WGEN is enabled, the DMA controller on the AT computer must be programmed appropriately and DMAEN and the necessary mode bits 2 through 0 must be enabled. When WGEN is enabled, transfers are automatically handled by the DMA controller. See *Programming DMA Operations* later in this chapter. If DACUPERR is set, an error has occurred.

Programming the Digital I/O Circuitry

The digital input circuitry is controlled and monitored using the Digital Input Register, the Digital Output Register, and the two bits DIOPAEN and DIOPBEN in Command Register 2. See the register bit descriptions earlier in this chapter for more information.

To enable digital output port A, set the DIOPAEN bit in Command Register 2. To enable digital output port B, set the DIOPBEN bit in Command Register 2. When a digital output port is enabled, the contents of the Digital Output Register are driven onto the digital lines corresponding to that port. The digital output for both ports A and B are updated by writing the desired pattern to the Digital Output Register.

In order for an external device to drive the digital I/O lines, the input ports must be enabled. Clear the DIOPAEN bit in Command Register 2 if an external device is driving digital I/O lines ADIO<3..0>. Clear the DIOPBEN bit in Command Register 2 if an external device is driving digital I/O lines BDIO<3..0>. The Digital Input Register can then be read to monitor the state of the digital I/O lines as driven by the external device.

The logic state of all eight digital I/O lines can be read from the Digital Input Register. If the digital output ports are enabled, the Digital Input Register serves as a read-back register; that is, you can determine how the AT-MIO-16F-5 is driving the digital I/O lines by reading the Digital Input Register.

If any digital I/O line is not driven, it floats to an indeterminate value. If more than one device is driving any digital I/O line, the voltage at that line can also be indeterminate. In these cases, the digital line has no meaningful logic value, and reading the Digital Input Register can return either 1 or 0 for the state of the digital line.

Programming the Am9513A Counter/Timer

Counters 1, 2, and 5 of the Am9513A Counter/Timer are available for general-purpose timing applications. The programmable frequency output pin FOUT is also available as a timing signal source. These applications and a general description of the Am9513A Counter/Timer are included in *Data Acquisition Timing Connections* in Chapter 2. *Timing I/O Circuitry* in Chapter 4 explains how the Am9513A is used on the AT-MIO-16F-5 board.

Initialization of the Am9513A as required by the AT-MIO-16F-5 and specific programming requirements for the sample-interval and sample counters are given earlier in this chapter. For general programming details for Counters 1, 2 and 5, and the programmable frequency output, refer to the Am9513A data sheet included in Appendix C.

In programming the Master Mode Register, keep the following considerations in mind:

- The Am9513A must be used in 16-bit bus mode.
- The scaler control should be set to BCD division for correct operation of the clocks as described in *Programming Multiple A/D Programming Conversions on a Single Input Channel*.

RTSI Bus Trigger Line Programming Considerations

The RTSI switch connects signals on the AT-MIO-16F-5 to the seven RTSI bus trigger lines. The RTSI switch has seven pins labeled A<6..0> connected to AT-MIO-16F-5 signals and seven pins labeled B<6..0> connected to the seven RTSI bus trigger lines. Table 4-8 shows the signals connected to each pin.

Table 4-8. RTSI Switch Signal Connections

RTSI Switch Pin	Signal Name	Signal Direction
A Side:		
A0	EXTCONV*	Bidirectional
A1	FOUT	Output
A2	OUT2	Output
A2	GATE1	Input
A3	SOURCE5	Bidirectional
A4	OUT5	Output
A4	DACUPTRIG*	Input
A5	OUT1	Bidirectional
A6	EXTTRIG*	Bidirectional
B Side:		
B0	TRIGGER0	Bidirectional
B1	TRIGGER1	Bidirectional
B2	TRIGGER2	Bidirectional
B3	TRIGGER3	Bidirectional
B4	TRIGGER4	Bidirectional
B5	TRIGGER5	Bidirectional
B6	TRIGGER6	Bidirectional

Figure 3-8 in Chapter 3, *Theory of Operation*, diagrams the AT-MIO-16F-5 RTSI switch connections.

AT-MIO-16F-5 RTSI Signal Connection Considerations

The AT-MIO-16F-5 board has a total of nine signals connected to the seven A-side pins of the RTSI switch. These same signals also appear at the AT-MIO-16F-5 I/O connector. As shown in Table 4-8, two AT-MIO-16F-5 signals are connected to pin A2, and two signals are connected to pin A4. The routing of these signals is further controlled by the bits A4DRV, A4RCV, A2DRV, and A2RCV in Command Register 2.

- To drive the RTSI switch pin A2 with the signal OUT2, set the A2DRV bit in Command Register 2. Otherwise, clear the A2DRV bit.

- To drive the signal GATE1 from pin A2 of the RTSI switch, set the A2RCV bit in Command Register 2. Otherwise, clear the A2RCV bit.

Note: If both the A2DRV and A2RCV bits are set, the GATE1 signal is driven by the signal OUT2. This arrangement is probably not desirable.

- To drive the RTSI switch pin A4 with the signal OUT5, set the A4DRV bit in Command Register 2. Otherwise, clear the A4DRV bit.

- To drive the signal DACUPTRIG* from pin A4 of the RTSI switch, set the A4RCV bit in Command Register 2. Otherwise, clear the A4RCV bit.

Note: If both the A4DRV and A4RCV bits are set, the DACUPTRIG* signal is driven by the signal OUT5.

Programming the RTSI Switch

The RTSI switch can be programmed to connect any of the signals on the A side to any of the signals on the B side and vice versa. To do this, a 56-bit pattern is shifted into the RTSI switch by writing one bit at a time to the RTSI Switch Shift Register and then writing to the RTSI Switch Strobe Register to load the pattern into the RTSI switch.

The 56-bit pattern is made up of two 28-bit patterns, one for side A and one for side B of the RTSI switch. The low-order 28 bits select the signal sources for the B-side pins. The high-order 28 bits select the signal sources for the A-side pins. Each of the 28-bit patterns are made up of seven 4-bit fields, one for each pin. The 4-bit field selects the signal source and the output enable for the pin. Figure 4-1 shows the bit map of the RTSI switch 56-bit pattern.

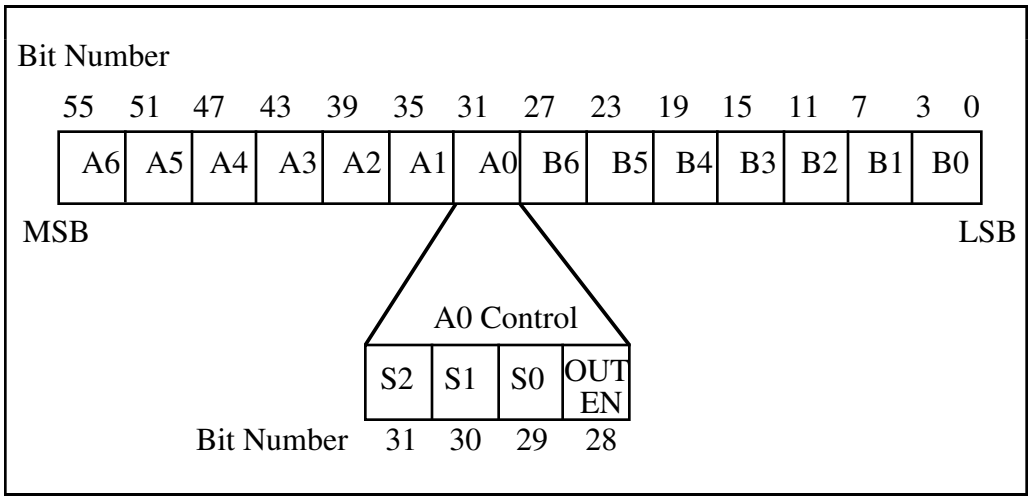


Figure 4-1. RTSI Switch Control Pattern

In Figure 4-1, the fields labeled A6 through A0 and B6 through B0 are the 4-bit control fields for each RTSI switch pin of the same name. The 4-bit control field for pin A0 is shown in Figure 4-1.

The bits labeled S2 through S0 are the signal source selection bits for the pin. One of seven source signals can be selected. Pins A6 through A0 can select any of the pins B6 through B0 as signal sources. Pins B6 through B0 select any of the pins A6 through A0 as signal sources. For example, the pattern 011 for S2 through S0 in the A0 control field selects the signal connected to pin B3 as the signal source for pin A0.

The bit labeled OUTEN is the output enable bit for that pin. If the OUTEN bit is set, the pin is driven by the selected source signal (the pin acts as an output pin). If the OUTEN bit is cleared, the pin is not driven regardless of the source signal selected; instead, the pin can be used as an input pin.

If the preceding A0 control field contains the pattern 0111, the signal connected to pin B3 (Trigger Line 3) appears at pin A0. On the AT-MIO-16F-5 board, this arrangement allows the EXTCONV* signal to be driven by Trigger Line 3. Conversely, if the B4 control field contains the pattern 1011, the signal connected to pin A5 appears at pin B4. This arrangement allows Trigger Line 4 to be driven by the AT-MIO-16F-5 OUT1 signal. In this way, boards connected via the RTSI bus can send signals to each other over the RTSI bus trigger lines.

To program the RTSI switch, complete these steps:

1. Calculate the 56-bit pattern based on the desired signal routing.

   a. Clear the OUTEN bit for all input pins and for all unused pins.

   b. Select the signal source pin for all output pins by setting bits S2 through S0 to the source pin number.

   c. Set the OUTEN bit for all output pins.

2. For i = 0 to 55, follow these steps:

   a. Copy bit i of the 56-bit pattern to bit 0 of an 8-bit temporary variable.

   b. Write the temporary variable to the RTSI Switch Shift Register (8-bit write).

3. Write 0 to the RTSI Switch Strobe Register (8-bit write). This operation loads the 56-bit pattern into the RTSI switch. At this point, the new signal routing goes into effect.

Step 2 can be completed by simply writing the low-order 8 bits of the 56-bit pattern to the RTSI Switch Shift Register, then shifting the 56-bit pattern right once, and repeating this two-step operation a total of 56 times. Only bit 0 of the word written to the RTSI Switch Shift Register is used. The higher-order bits are ignored.

Programming DMA Operations

The AT-MIO-16F-5 can be programmed so that the A/D FIFOEF* generates DMA requests every time one or more A/D conversion values are stored in the A/D FIFO, when the FIFOHF* is low and the FIFO is half-full, and when a falling edge is detected on the DACUPTRIG* signal. There are two DMA modes: single-channel transfer and dual-channel transfer. Single-channel DMA uses only Channel A DMA signals, while dual-channel DMA uses signals for both Channel A and Channel B. The DMA channels are selected by the onboard jumpers (W4 and W7). If DMA Channel B is jumpered to one of the AT DMA channels, it need not be unjumpered in single channel mode since selecting single channel DMA places the control signals for Channel B in the high-impedance state. To program the DMA operation, perform the following steps after the circuitry on the AT-MIO-16F-5 is set up:

1. Set the DMAEN bit in Command Register 2 to enable DMA request generation. Then select the appropriate mode bits 2 through 0 for single-channel or double-channel input, output, or both.

2. Write 0 to the DMA TC INT Clear Register, the DAC Update INT Clear Register,and the A/D Clear Register.

3. Program the DMA controller to service DMA requests from the AT-MIO-16F-5 board. Refer to the *IBM Personal Computer AT Technical Reference* manual for more information on DMA controller programming.

4. If a DMA terminal count is received after the DMA service, write 0 to either the DMA TC INT Clear Register to clear the DMATCA and DMATCB bits in the Status Register.

Once steps 1 through 3 are completed, the DMA controller is programmed to acknowledge requests. If analog input DMA is programmed, the DMA controller automatically reads the A/D FIFO Register whenever an A/D conversion result is available and then stores the result in a buffer in memory. If the DMA controller has been programmed for analog output updating, values from the buffer in memory are automatically written to the DAC upon receipt of a DMA request. If both analog input and output DMA is selected, then the DMA controller reads the FIFO or writes to the DACs depending on which channel requested a DMA transfer.

If single-channel interleaved DMA is selected for writing data to the DACs, then one buffer services both DAC 0 and DAC 1. This is accomplished by interleaving the data in the buffer. The first location in the buffer should hold the first value to be transferred to DAC0, the second should hold the first value to be transferred to DAC1, the third should hold the second value to be transferred to DAC0, and so on.

If dual-channel DMA operation has been selected to service analog input, DMA Channel A and Memory Buffer A (DMA A) are served first. When a DMA terminal count is received, the board automatically switches the DMA operation to DMA Channel B and Memory Buffer B (DMA B). Therefore, the board can collect data to or from one buffer and service data in another buffer simultaneously. If the DMA controller is programmed for auto-reinitialize mode, DMA A and DMA B are continuously served in turn.

If dual-channel DMA operation has been selected to service analog output, Memory Buffer A (DMA Channel A) and Memory Buffer B (DMA Channel B) are serviced concurrently, with Buffer A serving DAC 0 and Buffer B serving DAC 1.

Interrupt Programming

Five different interrupts are generated by the AT-MIO-16F-5 board:

- An interrupt whenever a conversion is available to be read from the A/D FIFO.
- An interrupt whenever the A/D FIFO is more than half-full.
- An interrupt whenever a data acquisition sequence completes.
- An interrupt whenever a DMA terminal count is received.
- An interrupt whenever a falling edge on the DACUPTRIG* pin of the Am9513A is detected.

These five interrupts can be enabled either individually or in any combination. In any of the interrupt modes, it is a good practice to confirm the source of the interrupt through reading the Status Register. If FIFOEF* or FIFOHF* are set, then a conversion interrupt has occurred. Reading from the A/D FIFO Register clears these interrupt conditions. Writing to the A/D Clear Register also clears these conversion interrupts. If DAQCOMP is set then the interrupt results from the completion of a data acquisition operation. This interrupt is cleared by writing to the ADCLEAR Register. If DACUP is set, then a DAC update interrupt has occurred. Writing to the DAC Update INT Clear Register clears this interrupt condition. In the case that WGEN is disabled in Command Register 2, the DACs are not updated and the DACUP signal can be used as a timer interrupt. If DMATCA or DMATCB are set, then a DMA TC INT has occurred on either DMA Channel A or B. Writing to the DMA TC INT Clear Register clears this interrupt condition.

If interrupts are desired on a DMA terminal count and/or completion of a data acquisiton sequence, it is sufficient to enable only the DMATCINTEN and/or CMPLINTEN bit(s) in Command Register 2 without enabling the INTEN bit. On the AT-MIO-16F-5 the INT signal can be tri-stated and can share the same interrupt level as another device.

Chapter 5
Calibration Procedures

This chapter discusses the calibration procedures for the AT-MIO-16F-5 analog input and analog output circuitry.

The AT-MIO-16F-5 is calibrated at the factory before shipment; an onboard EEPROM stores the calibration constants, which must be written to the eight calibration DACs for the board to be properly calibrated. To maintain the 12-bit accuracy of the AT-MIO-16F-5 analog input and analog output circuitry, periodic self-calibration is recommended. This is done under software control. Calibration constants are stored in an onboard EEPROM (see Figure 5-1). Calibration software is included with the board package as part of the utility application and utility library. Using the self-calibration feature eliminates most errors due to drift of offset and gain with time and temperature.

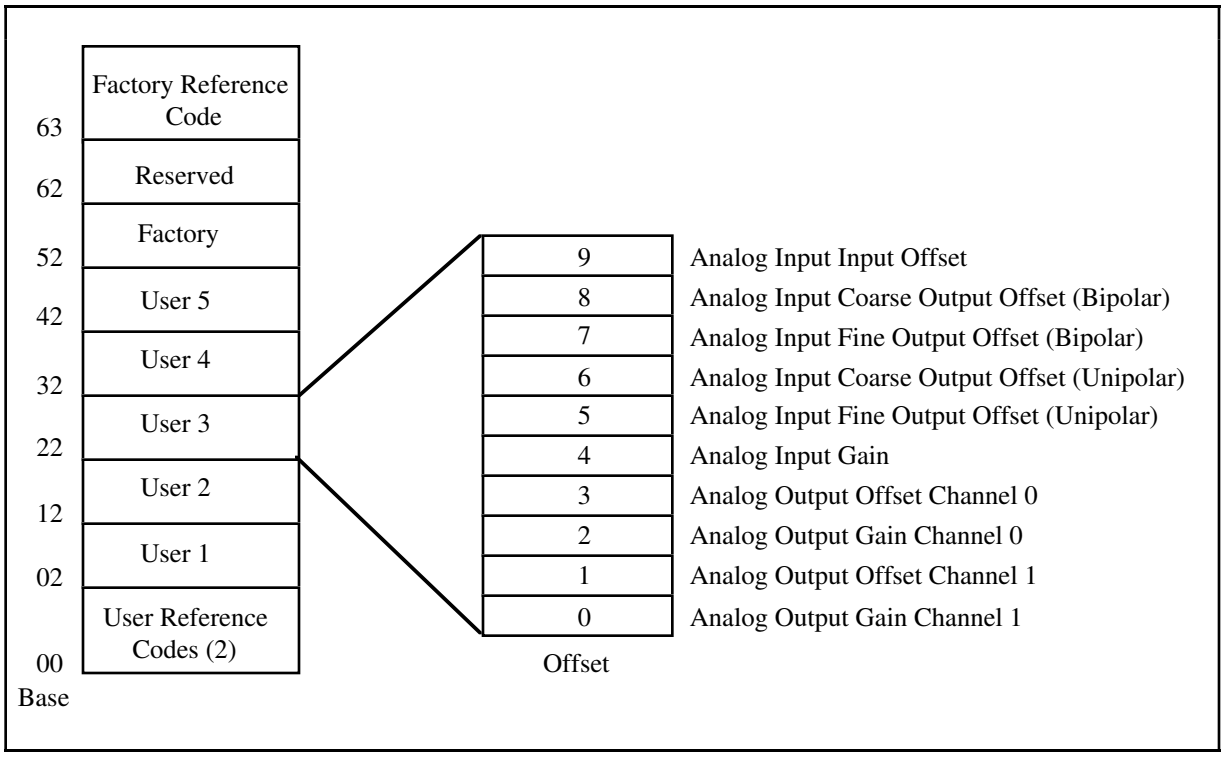


Figure 5-1. EEPROM Map

Factory calibration with the AT-MIO-16F-5 is valid in any analog input configuration, and with the analog outputs set up for a bipolar output range with the internal reference. If you want to use a different analog output configuration, it may be necessary to recalibrate the analog output offsets and gains.

Calibration Equipment Requirements

Normal self-calibration requires no external calibration equipment. However, because the internal voltage reference drifts slightly with time and temperature, it may be necessary to redetermine its value every two or three years, or whenever operating the board at an ambient temperature that is more than 20 ˚C from the temperature at which the reference value was last determined. The value of the reference is initially determined at the factory at a room temperature of 25 ˚C. After the value of the reference is determined, the value should be stored in the EEPROM so that it can be used by the input and output calibration routines. The calibration procedure which determines the reference value is explained in *Reference Calibration* later in this chapter. Locations have been provided in the EEPROM to accommodate user calibration constants (see Figure 5-1).

For best measurement results, the AT-MIO-16F-5 onboard reference needs to be measured to ±0.012% accuracy. According to standard practice, the equipment used to calibrate the AT-MIO-16F-5 should be 10 times as accurate; that is, the equipment should have ±0.001 percent rated accuracy. Practically speaking, calibration equipment with four times the accuracy of the item under calibration is generally considered acceptable. Four times the accuracy of the AT-MIO-16F-5 is 0.003 percent. To redetermine the value of the reference on the AT-MIO-16F-5 board you need the following equipment:

A precision DC voltage source (usually a calibrator):

Voltage: ±6.0 V to ±10.0 V

Accuracy: ±0.001% standard
±0.003% sufficient

It is important to realize that inaccuracy of the internal voltage reference results only in gain error. Offset error is unaffected. If an application can tolerate slight gain inaccuracy, there should not be a need to redetermine the value of the onboard reference.

Calibration DACs

There are eight 8-bit DACs on the AT-MIO-16F-5 that are used for calibration. These DACs are described in Table 5-1.

Table 5-1. Calibration DACs

Analog	DAC	Function	Adjustment Range	Incremental Effect
Input	CALDAC 0	Input-referred offset trim	3.5 mV	-14 µV
	CALDAC 1	Coarse offset trim (gain-independent)	300 LSB	-1.2 LSB
	CALDAC 2	Fine offset trim (gain-independent)	9 LSB	-0.04 LSB
	CALDAC 3	Gain trim	0.9%	-35 ppm
Output	CALDAC 4	Offset trim, channel 0	200 mV (bipolar) 100 mV (unipolar)	-0.78 mV (bipolar) -0.39 mV (unipolar)
	CALDAC 5	Gain trim, channel 0	1.0%	-39 ppm
	CALDAC 6	Offset trim, channel 1	200 mV (bipolar) 100 mV (unipolar)	-0.78 mV (bipolar) -0.39 mV (unipolar)
	CALDAC 7	Gain trim, channel 1	1.0%	-39 ppm

Reference Calibration

The AT-MIO-16F-5 has a stable voltage reference to which gain can be calibrated. The value of this voltage reference is determined through the reference calibration routine, which requires a known external voltage between 6 and 10 volts to be connected differentially on any desired input channel. The routine compares the internal reference to the external one, and uses their measured ratio in conjunction with the known value of the external reference to compute the value of the internal reference. This value is stored as a binary number in the onboard EEPROM for subsequent use by the analog input and output gain calibration routines.

Because the onboard reference is very stable with respect to time and temperature, it is seldom necessary to use the reference calibration routine. Every two or three years should be sufficient, or whenever operating the board at an ambient temperature that is more than 20° C from the temperature at which the reference value was last determined. Factory calibration is performed at approximately 25° C.

Input Calibration

To null out error sources that compromise the quality of measurements, the input calibration routine calibrates the analog input circuitry by adjusting the following potential sources of error:

- Offset error at the input of the instrumentation amplifier
- Offset error at the input of the ADC
- Gain error of the analog input circuitry

Offsets at the input to the instrumentation amplifier contribute gain-dependent error to the analog input system. This offset is multiplied by the gain of the instrumentation amplifier. To calibrate this offset, the routine grounds the inputs of the instrumentation amplifier, measures the input at two different gains, and adjusts CALDAC0 until the measured offset in LSBs is independent of the gain setting.

Offset error at the input of the ADC is the total of the voltage offsets contributed by the circuitry from the output of the instrumentation amplifier to the ADC input (including the ADC's own offsets). To calibrate this offset, the routine grounds the inputs of the instrumentation amplifier, measures the input at two different gains, and adjusts CALDAC1 and CALDAC2 so that the measured offset in LSBs is exactly proportional to the gain setting.

If the three analog input offset DACs are adjusted in this way, there is no significant residual offset error, and reading a grounded channel returns, on average, less than 0.5 LSB, regardless of gain setting.

All the stages up to and including the input of the ADC contribute to the gain error of the analog input circuitry. With the instrumentation amplifier set to a gain of 1, the gain of analog input circuitry is ideally 1. The gain error is the deviation of the gain from 1 and appears as a multiplication of the input voltage being measured. To eliminate this error source, the routine measures the input first with the inputs grounded and then with the inputs connected to the internal voltage reference. It then adjusts CALDAC3 until the measured difference between the two voltages is equal to the value of the reference as stored in the onboard EEPROM. Once the board is calibrated at a gain of 1, there is only a small residual gain error (±0.02% maximum) at the other gains.

Output Calibration

To null out error sources that affect the accuracy of the output voltages generated, the output calibration routine calibrates the analog output circuitry by adjusting the following potential sources of error:

- Analog output offset error
- Analog output gain error

In order to read the analog output voltages, the output calibration routine requires that the AT-MIO-16F-5 analog outputs be "wrapped back" to the analog inputs as follows:

1. DAC0 should be connected to a channel in ACH8-15.

2. DAC1 should be connected to another channel in ACH8-15.

3. AOGND should be connected to the positive sides of the channels selected in steps 1 and 2. Do not tie AOGND to AIGND as this will complete a ground loop, potentially introducing offset calibration errors of several LSBs.

Offset error in the analog output circuitry is the total of the voltage offsets contributed by each component in the circuitry. This error, which is independent of the DAC output voltage, appears as a voltage difference between the desired voltage and the actual output voltage generated. To correct this offset error, the routine writes a value of 0 to each DAC and adjusts CALDAC4 and CALDAC6 until it measures 0 volts between each analog output and AOGND. The routine automatically corrects for any errors in the analog input circuitry.

Gain error in the analog output circuitry is the sum of the gain errors contributed by each component in the circuitry. This error also appears as a voltage difference between the desired voltage and the actual output voltage generated, but it is proportional to the DAC output voltage. The output calibration routine corrects this error by setting each analog output to 5 V and comparing it to the onboard reference. It then sets each output to 0 V and compares it to AIGND. These readings are all combined with the value of the onboard reference, as recovered from the EEPROM, to produce values to load into CALDAC5 and CALDAC7. The process is repeated until the half-scale swing of each analog output is correct according to the value of the onboard reference. This procedure is insensitive to offset, gain, and linearity errors in the analog input circuitry. It should be noted that CALDAC5 and CALDAC7 adjust the gain by varying the values of the internal DAC references. Thus the gain of an analog output channel cannot be adjusted under software control if the channel is using an external reference.

Appendix A
Specifications

This appendix lists the specifications for the AT-MIO-16F-5 board. These are typical at 25° C unless otherwise stated. The operating temperature range is 0° C to 70° C.

Analog Input

Number of input channels	16 single-ended, 8 differential
Analog resolution	12-bit, 1 in 4,096
Relative accuracy (nonlinearity + quantization error, see explanation of specifications)	±1.5 LSB maximum over temperature, ±0.8 LSB typical
Integral nonlinearity	±1 LSB maximum over temperature, ±0.3 LSB typical
Differential nonlinearity	±0.8 LSB maximum (no missing codes over temperature), ±0.2 LSB typical
Differential analog input ranges	±5 V or 0 to +10 V, software-selectable
Common-mode input range	Each input to the instrumentation amplifier should remain within ±12 V of AIGND.
Overvoltage protection	±15 V power-off, ±25 V power-on
Common-mode rejection ratio (DIFF/NRSE mode)	78 dB minimum, 90 dB typical, gain = 0.5 84 dB minimum, 96 dB typical, gain = 1 90 dB minimum, 102 dB typical, gain = 2 94 dB minimum, 110 dB typical, gain ≥ 5
Input bias current	±200 pA
Input offset current (DIFF/NRSE mode)	±100 pA
Input impedance	100 GΩ in parallel with 50 pF

Analog Input (continued)

Gain ranges	0.5, 1, 2, 5, 10, 20, 50 and 100, software-selectable
Gain accuracy (includes adjustment range)	
gain = 1	±0.5%, adjustable to < ±0.01%
gain ≠ 1	±0.5% ±0.02% maximum with gain error adjusted to zero at gain = 1
temperature coefficient	±20 ppm/° C
Input offset voltage (includes adjustment range)	±2.5 mV, adjustable to < ±10 µV
temperature coefficient	±4 µV/° C
Other system offset voltage (includes adjustment range)	±150 LSB, adjustable to < ±0.5 LSB
temperature coefficient	±0.1 LSB/° C
System noise	0.2 LSB rms for gains 0.5 to 50, dither off 0.4 LSB rms for gain 100, dither off 0.5 LSB rms, dither on
Input settling time	5 µsec maximum to ±0.5 LSB for all gains and ranges
Onboard reference	
temperature coefficient	5 ppm/° C maximum
long-term stability	15 ppm/$\sqrt{1000\ \text{hrs}}$

Explanation of Analog Input Specifications

Relative accuracy is a measure of the linearity of an ADC. However, relative accuracy is a tighter specification than a *nonlinearity* specification. Relative accuracy indicates the maximum deviation from a straight line for the analog-input-to-digital-output transfer curve. If an ADC has been calibrated perfectly, then this straight line is the ideal transfer function, and the relative accuracy specification indicates the worst deviation from the ideal that the ADC permits.

A relative accuracy specification of ±1 LSB is roughly equivalent to (but not the same as) a ±1/2 LSB nonlinearity or integral nonlinearity specification because relative accuracy encompasses both nonlinearity and variable quantization uncertainty, a quantity often mistakenly assumed to be exactly ±1/2 LSB. Although quantization uncertainty is ideally ±1/2 LSB, it can be different for each possible digital code and is actually the analog width of each code. Thus, it is more specific to use relative accuracy as a measure of linearity than it is to use what is normally called nonlinearity, because relative accuracy ensures that the *sum* of quantization uncertainty and A/D conversion error does not exceed a given amount.

Integral nonlinearity in an ADC is an often ill-defined specification that is supposed to indicate a converter's overall A/D transfer linearity. The manufacturer of the ADC chip used by National Instruments on the AT-MIO-16F-5 specifies its integral nonlinearity by stating that the analog center of any code will not deviate from a straight line by more than ±1 LSB. This specification is misleading because although a particularly wide code's center may be found within ±1 LSB of the ideal, one of its edges may be well beyond ±1.5 LSB; thus, the ADC would have a relative accuracy of that amount. National Instruments tests its boards to ensure that they meet all three linearity specifications defined in this appendix.

Differential nonlinearity is a measure of deviation of code widths from their theoretical value of 1 LSB. The width of a given code is the size of the range of analog values that can be input to produce that code, ideally 1 LSB. A specification of ±1 LSB differential nonlinearity ensures that no code has a width of 0 LSBs (that is, no missing codes) and that no code width exceeds 2 LSBs.

System noise is the amount of noise seen by the ADC when there is no signal present at the input of the board. The amount of noise that is reported directly (without any analysis) by the ADC is not necessarily the amount of real noise present in the system, unless the noise is considerably greater than 0.5 LSB rms. Noise that is less than this magnitude produces varying amounts of flicker, and the amount of flicker seen is a function of how near the real mean of the noise is to a code transition. If the mean is near or at a transition between codes, the ADC flickers evenly between the two codes, and the noise is seen as very nearly 0.5 LSB. If the mean is near the center of a code and the noise is relatively small, very little or no flicker is seen, and the noise is reported by the ADC as nearly 0 LSB. From the relationship between the mean of the noise and the measured rms magnitude of the noise, the character of the noise can be determined. National Instruments has determined that the character of the noise in the AT-MIO-16F-5 is fairly Gaussian, so the noise specifications given are the amounts of pure Gaussian noise required to produce our readings.

Analog Data Acquisition Rates

Single-Channel Acquisition Rates

The AT-MIO-16F-5 operates at a data acquisition rate of at least 200 ksamples/sec. Permissible data acquisition rates are determined by the minimum A/D conversion time of the system. This minimum conversion time is the sum of the conversion time of the ADC and the settling time of the analog input front end. When data acquisition is performed on a single analog input channel, the time required for the input sample-and-hold amplifier to acquire the input signal and settle to 12-bit accuracy (0.01%) is added to the conversion time of the ADC. The sum of conversion time and acquisition time for the sampling ADC used on the AT-MIO-16F-5 is guaranteed to be less than 5 µsec and is typically 4.6 µsec.

Multiple-Channel Scanning Acquisition Rates

Because of an innovative custom instrumentation amplifier designed for the AT-MIO-16F-5, the maximum multiple-channel scanning acquisition rate is identical to the single channel acquisition rate for all gains. No extra settling time is necessary between channels.

Analog Output

Number of output channels	2
Type of DAC	12-bit, multiplying
Relative accuracy (nonlinearity)	±0.5 LSB maximum, ±0.25 LSB typical
Differential nonlinearity	±1 LSB maximum (monotonic over temperature), ±0.2 LSB typical
Gain error	
internal reference (includes adjustment range)	±1.0%, adjustable to < 0.005%
temperature coefficient	±10 ppm/° C
external reference	±0.1%, not adjustable
temperature coefficient	±5 ppm/° C
Voltage offset	±100 mV bipolar mode, adjustable to < ±0.5 mV
(includes adjustment range)	±50 mV unipolar mode, adjustable to < ±0.3 mV
Output voltage ranges (jumper-selectable)	0 to 10 V, unipolar mode; ±10 V, bipolar mode
Current drive capability	±5 mA
External reference input impedance	11 kΩ unipolar mode, 7 kΩ bipolar mode for each output channel
Output settling time to 0.01%	4 µsec for a 20 V step
Output slew rate	30 V/µsec
Output noise	1 mV rms, DC to 1 MHz

Explanation of Analog Output Specifications

Relative accuracy in a D/A system is the same as nonlinearity because no uncertainty is added due to code width. Unlike an ADC, every digital code in a D/A system represents a specific analog value rather than a range of values. The relative accuracy of the system is therefore limited to the worst-case deviation from the ideal correspondence (a straight line), excepting noise. If a D/A system has been calibrated perfectly, then the relative accuracy specification reflects its worst-case absolute error.

Differential nonlinearity in a D/A system is a measure of deviation of code width from 1 LSB. In this case, code width is the difference between the analog values produced by consecutive digital codes. A specification of ±1 LSB differential nonlinearity ensures that the code width is always greater than 0 LSBs (guaranteeing monotonicity) and is always less than 2 LSBs.

Digital I/O

Compatibility	TTL-compatible
Output current source capability	Can source 2.6 mA and maintain V_{OH} at 2.4 V
Output current sink capability	Can sink 24 mA and maintain V_{OL} at 0.5 V

Timing I/O

Number of channels	3 counter/timers 1 frequency output
Resolution	16-bit for 3 counter/timers, 4-bit for frequency output channel
Base clock available	5-MHz, 1-MHz, 100-kHz, 10-kHz, 1-kHz, 100-Hz
Base clock accuracy	±0.01%
Compatibility	TTL-compatible inputs and outputs. Counter gate and source inputs are pulled up with 4.7 kΩ resistors onboard.
Counter input frequency	6.9 MHz maximum (145 nsec period) with a minimum pulse width of 70 nsec

Power Requirement (from PC I/O Channel)

Power consumption	1.6 A typical at +5 VDC

Physical

Board dimensions	13.3 x 4.5 in.
I/O connector	50-pin male ribbon-cable connector

Operating Environment

Component temperature	0° to 70° C
Relative humidity	5% to 90% noncondensing

Storage Environment

Temperature	-55° to 150° C
Relative humidity	5% to 90% noncondensing

Appendix B
I/O Connector

This appendix shows the pinout and signal names for the AT-MIO-16F-5 50-pin I/O connector, including a description of each connection.

Figure B-1 shows the AT-MIO-16F-5 I/O connector.

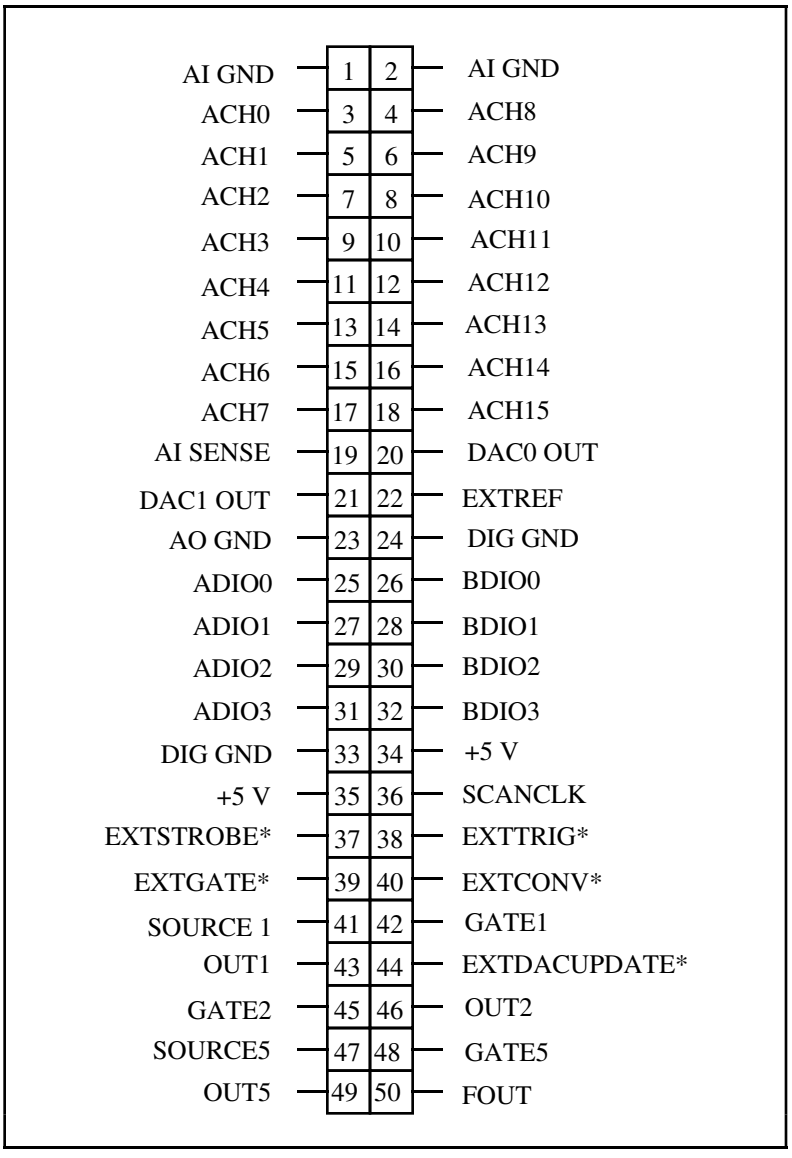


Figure B-1. AT-MIO-16F-5 I/O Connector

Signal Connection Descriptions

Pins	Signal Names	Reference	Descriptions
1, 2	AIGND	N/A	Analog Input Ground – These pins are the reference point for single-ended measurements and the bias current return point for differential measurements.
3 to 18	ACH <0..15>	AI GND	Analog Input Channels 0 through 15 – In differential mode, the input is configured for 8 channels. In single-ended mode, the input is configured for 16 channels.
19	AI SENSE	AI GND	Analog Input Sense – This pin serves as the reference node when the board is in NRSE configuration.
20	DAC0 OUT	AO GND	Analog Channel 0 Output – This pin supplies the voltage output of analog output channel 0.
21	DAC1 OUT	AO GND	Analog Channel 1 Output – This pin supplies the voltage output of analog output channel 1.
22	EXTREF	AO GND	External Reference – This is the external reference input for the analog output circuitry.
23	AOGND	N/A	Analog Output Ground – The analog output voltages are referenced to this node.
24,33	DIG GND	N/A	Digital Ground – This pin supplies the reference for the digital signals at the I/O connector as well as the +5 VDC supply.
25, 27, 29, 31	ADIO <0..3>	DIG GND	Digital I/O port A signals.
26, 28, 30, 32	BDIO <0..3>	DIG GND	Digital I/O port B signals.
34,35	+5 V	DIG GND	+5 VDC Source – This pin is fused for up to 1 A of +5 V supply.
36	SCAN CLK	DIG GND	Scan Clock – This pin pulses once for each A/D conversion. The low-to-high edge indicates when the input signal can be removed from the input or switched to another signal.

(continues)

Pins	Signal Names	Reference	Descriptions (continued)
37	EXTSTROBE*	DIG GND	External Strobe – Writing to the EXTSTROBE Register results in a 200-500 nsec low pulse on this pin.
38	EXTTRIG*	DIG GND	External Trigger – In posttrigger data acquisition sequences, a high-to-low edge on EXTTRIG* initiates the sequence. In pretrigger applications, the first high-to-low edge of EXTTRIG* initiates conversions while the second high-to-low edge initiates the sequence.
39	EXTGATE*	DIG GND	External Gate – When EXTGATE* is low, A/D conversions are inhibited. When EXTGATE* is high, A/D conversions are enabled.
40	EXTCONV*	DIG GND	External Convert – A high-to-low edge on EXTCONV* causes an A/D conversion to occur. If EXTGATE* or EXTCONV* is low, or INTGATE in Command Register 1 is high, conversions are inhibited.
41	SOURCE1	DIG GND	SOURCE1 – This pin is from the Am9513A Counter 1 signal.
42	GATE1	DIG GND	GATE1 – This pin is from the Am9513A Counter 1 signal.
43	OUT1	DIG GND	OUTPUT1 – This pin is from the Am9513A Counter 1 signal.
44	EXTDACUPDATE*	DIG GND	External DAC Update – If selected, a high-to-low edge on EXTDACUPDATE* results in the ouput DACS being updated with the value written to them.
45	GATE2	DIG GND	GATE2 – This pin is from the Am9513A Counter 2 signal.
46	OUT2	DIG GND	OUTPUT2 – This pin is from the Am9513A Counter 2 signal.
47	SOURCE5	DIG GND	SOURCE5 – This pin is from the Am9513A Counter 5 signal.

(continues)

Pins	Signal Names	Reference	Descriptions (continued)
48	GATE5	DIG GND	GATE5 – This pin is from the Am9513A Counter 5 signal.
49	OUT5	DIG GND	OUT5 – This pin is from the Am9513A Counter 5 signal.
50	FOUT	DIG GND	Frequency Output – This pin is from the Am9513A FOUT signal.

Detailed signal specifications are included in Chapter 2, *Configuration and Installation*.

Appendix C
AMD Data Sheet*

This appendix contains the *AMD Am9513A System Timing Controller* (Advanced Micro Devices, Inc.) data sheet. This controller is used on the AT-MIO-16F-5.

Am9513A

System Timing Controller

FINAL

DISTINCTIVE CHARACTERISTICS

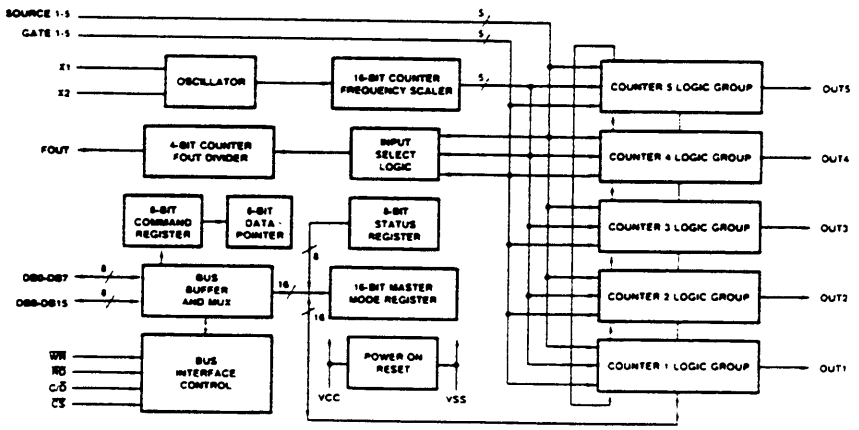
- Five independent 16-bit counters
- High speed counting rates
- Up/down and binary/BCD counting
- Internal oscillator frequency source
- Tapped frequency scaler
- Programmable frequency output
- 8-bit or 16-bit bus interface
- Time-of-day option
- Alarm comparators on counters 1 and 2
- Complex duty cycle outputs
- One-shot or continuous outputs
- Programmable count/gate source selection
- Programmable input and output polarities
- Programmable gating functions
- Retriggering capability
- +5 volt power supply
- Standard 40-pin package
- SMD/DESC qualified

GENERAL DESCRIPTION

The Am9513A System Timing Controller is an LSI circuit designed to service many types of counting, sequencing and timing applications. It provides the capability for programmable frequency synthesis, high resolution programmable duty cycle waveforms, retriggerable digital one-shots, time-of-day clocking, coincidence alarms, complex pulse generation, high resolution baud rate generation, frequency shift keying, stop-watching timing, event count accumulation, waveform analysis, etc. A variety of programmable operating modes and control features allows the Am9513A to be personalized for particular applications as well as dynamically reconfigured under program control.

The STC includes five general-purpose 16-bit counters. A variety of internal frequency sources and external pins may be selected as inputs for individual counters with software selectable active-high or active-low input polarity. Both hardware and software gating of each counter is available. Three-state outputs for each counter provide pulses or levels and can be active-high or active-low. The counters can be programmed to count up or down in either binary or BCD. The host processor may read an accumulated count at any time without disturbing the counting process. Any of the counters may be internally concatenated to form any effective counter length up to 80 bits.

BLOCK DIAGRAM

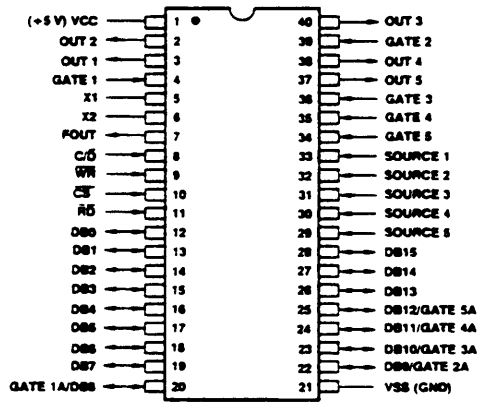


BD003381

Publication # 01731 Rev. F Amendment /0
Issue Date: August 1989

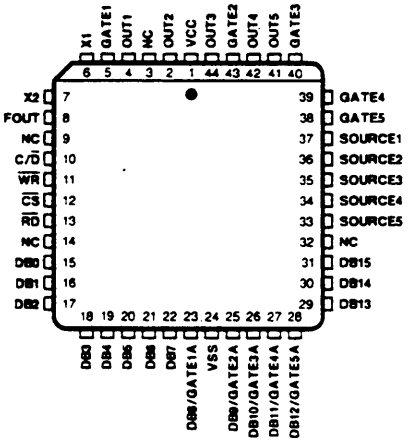
CONNECTION DIAGRAMS
Top View

DIP



CD005212

LCC & PLCC



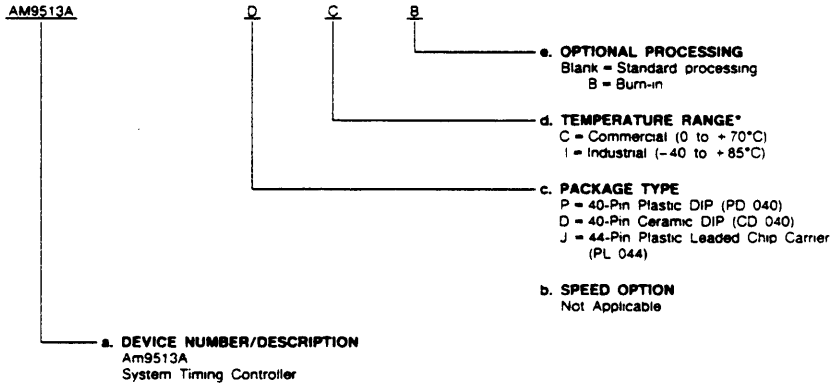
CD009912

Note: Pin 1 is marked for orientation.

2

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM9513A	PC, DC, DCB, DIB, JC

*This device is also available in Military temperature range.

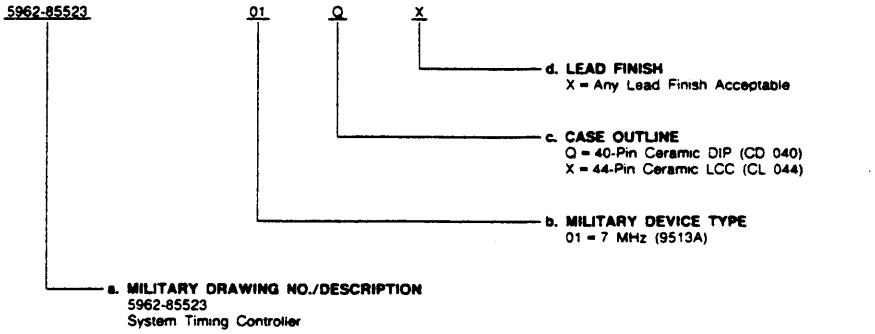
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

ORDERING INFORMATION (continued)

Standard Military Drawing (SMD)/DESC Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. Standard Military Drawing (SMD)/DESC products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for SMD/DESC products is formed by a combination of: **a. Military Drawing Part Number**
b. Device Type
c. Case Outline
d. Lead Finish



Valid Combinations	
5962-8552301	QX, XX

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

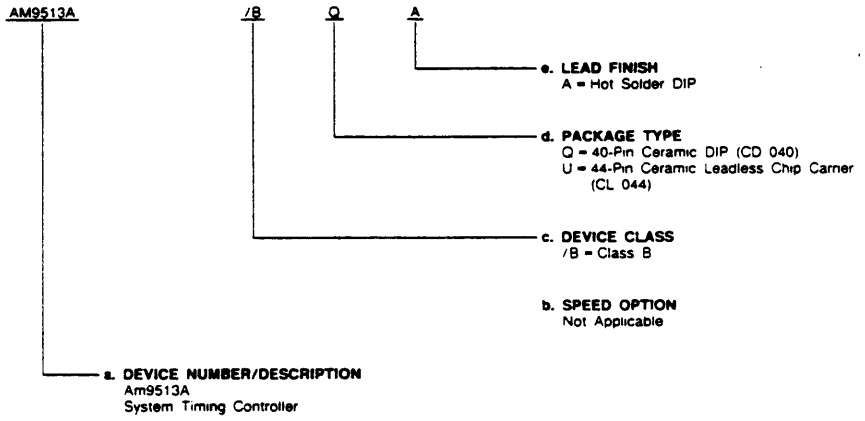
Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

ORDERING INFORMATION (continued)

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
AM9513A	/BQA, /BUA

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

Group A tests consist of
Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

2

PIN DESCRIPTION

Pin No.	Name	I/O	Description
1	VCC		+5 V Power Supply.
21	VSS		Ground.
5, 6	X1, X2	O, I	(Crystal). X1 and X2 are the connections for an external crystal used to determine the frequency of the internal oscillator. The crystal should be a parallel-resonant, fundamental-mode type. An RC or LC or other reactive network may be used instead of a crystal. For driving from an external frequency source, X1 should be left open and X2 should be connected to a TTL source and a pull-up resistor.
7	FOUT	O	(Frequency Out). The FOUT output is derived from a 4-bit counter that may be programmed to divide its input by any integer value from 1 through 16 inclusive. The input to the counter is selected from any of 15 sources, including the internal scaled oscillator frequencies. FOUT may be gated on and off under software control and when off will exhibit a low impedance to ground. Control over the various FOUT options resides in the Master Mode register. After power-up, FOUT provides a frequency that is 1/16 that of the oscillator. The input source on power-up is F1.
4, 39, 36 - 34	GATE1 - GATE5	I	(Gate). The Gate inputs may be used to control the operations of individual counters by determining when counting may proceed. The same Gate input may control up to three counters. Gate pins may also be selected as count sources for any of the counters and for the FOUT divider. The active polarity for a selected Gate input is programmed at each counter. Gating function options allow level-sensitive gating or edge-initiated gating. Other gating modes are available including one that allows the Gate input to select between two counter output frequencies. All gating functions may also be disabled. The active Gate input is conditioned by an auxiliary input when the unit is operating with an external 8-bit data bus. See Data Bus description. Schmitt-trigger circuitry on the GATE inputs allows slow transition times to be used.
33 - 29	SRC1 - SRC5	I	(Source) The Source inputs provide external signals that may be counted by any of the counters. Any Source line may be routed to any or all of the counters and the FOUT divider. The active polarity for a selected SRC input is programmed at each counter. Any duty cycle waveform will be accepted as long as the minimum pulse width is at least half the period of the maximum specified counting frequency for the part. Schmitt-trigger circuitry on the SRC inputs allows slow transition times to be used.
3, 2, 40, 38, 37	OUT1 - OUT5	O	(Counter). Each 3-state OUT signal is directly associated with a corresponding individual counter. Depending on the counter configuration, the OUT signal may be a pulse, a square wave, or a complex duty cycle waveform. OUT pulse polarities are individually programmable. The output circuitry detects the counter state that would have been all bits zero in the absence of a reinitialization. That information is used to generate the selected waveform type. An optional output mode for Counters 1 and 2 overrides the normal output mode and provides a true OUT signal when the counter contents match the contents of an Alarm register.
12 - 19, 20, 22 - 28	DB0 - DB7, DB8 - DB15	I/O	(Data Bus). The 16 bidirectional Data Bus lines are used for information exchanges with the host processor. HIGH on a Data Bus line corresponds to one and LOW corresponds to zero. These lines act as inputs when $\overline{WR}$ and $\overline{CS}$ are active and as outputs when $\overline{RD}$ and $\overline{CS}$ are active. When $\overline{CS}$ is inactive, these pins are placed in a high-impedance state. After power-up or reset, the data bus will be configured for 8-bit width and will use only DB0 through DB7. DB0 is the least significant and DB7 is the most significant bit position. The data bus may be reconfigured for 16-bit width by changing a control bit in the Master Mode register. This is accomplished by writing an 8-bit command into the low-order DB lines while holding the DB13 - DB15 lines at a logic high level. Thereafter, all 16 lines can be used, with DB0 as the least significant and DB15 as the most significant bit position. When operating in the 8-bit data bus environment, DB8 - DB15 will never be driven active by the Am9513A. DB8 through DB12 may optionally be used as additional Gate inputs (see Figure 2). If unused, they should be held HIGH. When pulled LOW, a GATENA signal will disable the action of the corresponding counter N gating. DB13 - DB15 should be held HIGH in 8-bit bus mode whenever $\overline{CS}$ and $\overline{WR}$ are simultaneously active.
10	$\overline{CS}$	I	(Chip Select). The active-low Chip Select input enables Read and Write operations on the data bus. When Chip Select is HIGH, the Read and Write inputs are ignored. The first Chip Select signal after power-up is used to clear the power-on reset circuitry. If Chip Select is tied to ground permanently, the power-on reset circuitry may not function. In such a configuration, the software reset command must be issued following power-up to reset the Am9513A.
11	$\overline{RD}$	I	(Read). The active-low Read signal is conditioned by Chip Select and indicates that internal information is to be transferred to the data bus. The source will be determined by the port being addressed and, for Data Port reads, by the contents of the Data Pointer register. $\overline{WR}$ and $\overline{RD}$ should be mutually exclusive.
9	$\overline{WR}$	I	(Write). The active-low Write signal is conditioned by Chip Select and indicates that data bus information is to be transferred to an internal location. The destination will be determined by the port being addressed and, for Data Port writes, by the contents of the Data Pointer register. $\overline{WR}$ and $\overline{RD}$ should be mutually exclusive.
8	C/$\overline{D}$	I	(Control/Data). The Control/Data signal selects source and destination locations for Read and Write operations on the data bus. Control Write operations load the Command register and the Data Pointer. Control Read operations output the Status register. Data Read and Data Write transfers communicate with all other internal registers. Indirect addressing of the data port is controlled internally by the Data Pointer register.

Signal	Abbreviation	Type	Pins
+ 5 Volts	VCC	Power	1
Ground	VSS	Power	1
Crystal	X1, X2	O, I	2
Read	$\overline{RD}$	Input	1
Write	$\overline{WR}$	Input	1
Chip Select	$\overline{CS}$	Input	1
Control/Data	C/$\overline{D}$	Input	1
Source N	SRC	Input	5
Gate N	GATE	Input	5
Data Bus	DB	I/O	16
Frequency Out	FOUT	Output	1
Out N	OUT	Output	5

Figure 1. Interface Signal Summary

Figure 1 summarizes the interface signals and their abbreviations for the STC.

Package Pin	Data Bus Width (MM14)	
	16 Bits	8 Bits
12	DB0	DB0
13	DB1	DB1
14	DB2	DB2
15	DB3	DB3
16	DB4	DB4
17	DB5	DB5
18	DB6	DB6
19	DB7	DB7
20	DB8	GATE 1A
22	DB9	GATE 2A
23	DB10	GATE 3A
24	DB11	GATE 4A
25	DB12	GATE 5A
26	DB13	(VIH)
27	DB14	(VIH)
28	DB15	(VIH)

Figure 2. Data Bus Assignments

Interface Considerations

All of the input and output signals for the Am9513A are specified with logic levels compatible with those of standard TTL circuits. In addition to providing TTL compatible voltage levels, other output conditions are specified to help configure non-standard interface circuitry. The logic level specifications take into account all worst-case combinations of the three variables that affect the logic level thresholds: ambient temperature, supply voltage and processing parameters. A change in any of these toward nominal values will improve the actual operating margins and will increase noise immunity.

Unprotected open gate inputs of high quality MOS transistors exhibit very high resistances on the order of perhaps 10^{14} ohms. It is easy, therefore, in some circumstances, for charge to enter the gate node of such an input faster than it can be discharged and consequently, for the gate voltage to rise high enough to break down the oxides and destroy the transistor.

All inputs to the Am9513A include protection networks to help prevent damaging accumulations of static charge. The protection circuitry is designed to slow the transitions of incoming current surges and to provide low-impedance discharge paths for voltages beyond the normal operating levels. Note, however, that input energy levels can nonetheless be too high to be successfully absorbed. Conventional design, storage, and handling precautions should be observed so that the protection networks themselves are not overstressed.

Within the limits of normal operation, the input protection circuitry is inactive and may be modeled as a lumped series RC as shown in Figure 3(a). The functionally active input connection during normal operation is the gate of a MOS transistor. No active sources or drains are connected to the inputs so that neither transient nor steady-state currents are impressed on the driving signals other than the charging or discharging of the input capacitance and the accumulated leakage associated with the protection network and the input circuit.

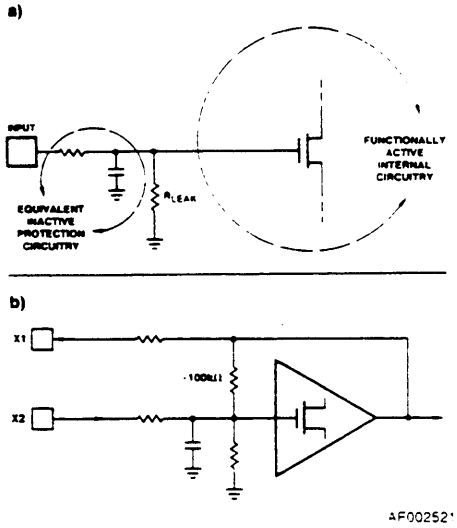


Figure 3. Input Circuitry

The only exception to the purely capacitive input case is the X2 crystal input. As shown in Figure 3(b) an internal resistor connects X1 and X2 in addition to the protection network. The resistor is a modestly high value of more than 100kohms.

Fanout from the driving circuitry into the Am9513A inputs will generally be limited by transition time considerations rather than DC current limitations when the loading is dominated by conventional MOS circuits. In an operating environment, all inputs should be terminated so they do not float and therefore will not accumulate stray static charges. Unused inputs should be tied directly to Ground or VCC, as appropriate. An input in use will have some type of logic output driving it, and termination during operation will not be a problem. Where inputs are driven from logic external to the card containing this chip, however, on-board termination should be provided to protect the chip when the board is unplugged (the input would otherwise float). A pull-up resistor or a simple inverter or gate will suffice.

DETAILED DESCRIPTION

The Am9513A System Timing Controller (STC) is a support device for processor oriented systems that is designed to enhance the available capability with respect to counting and timing operations. It provides the capability for programmable frequency synthesis, high resolution programmable duty cycle waveforms, retriggerable digital timing functions, time-of-day clocking, coincidence alarms, complex pulse generation, high resolution baud rate generation, frequency shift keying, stop-watching timing, event count accumulation, waveform analysis and many more. A variety of programmable operating modes and control features allow the Am9513A to be personalized for particular applications as well as dynamically reconfigured under program control.

The STC includes five general-purpose 16-bit counters. A variety of internal frequency sources and external pins may be selected as inputs for individual counters with software selectable active-high or active-low input polarity. Both hardware and software gating of each counter is available. Three-state outputs for each counter provide either pulses or levels. The counters can be programmed to count up or down in either binary or BCD. The accumulated count may be read without disturbing the counting process. Any of the counters may be internally concatenated to form an effective counter length of up to 80 bits.

The Am9513A block diagrams indicate the interface signals and the basic flow of information. Internal control lines and the internal data bus have been omitted. The control and data registers are all connected to a common internal 16-bit bus. The external bus may be 8- or 16-bits wide; in the 8-bit mode, the internal 16-bit information is multiplexed to the low order data bus pins DB0 through DB7.

An internal oscillator provides a convenient source of frequencies for use as counter inputs. The oscillator's frequency is controlled at the X1 and X2 interface pins by an external reactive network such as a crystal. The oscillator output is divided by the Frequency Scaler to provide several sub-frequencies. One of the scaled frequencies (or one of ten input signals) may be selected as an input to the FOUT divider and then comes out of the chip at the FOUT interface pin.

The STC is addressed by the external system as two locations: a Control port and a Data port. The Control port provides direct access to the Status and Command registers, as well as allowing the user to update the Data Pointer register. The Data port is used to communicate with all other addressable internal locations. The Data Pointer register controls the Data port addressing.

Among the registers accessible through the Data port are the Master Mode register and five Counter Mode registers, one for each counter. The Master Mode register controls the programmable options that are not controlled by the Counter Mode registers.

Each of the five general-purpose counters is 16-bits long and is independently controlled by its Counter Mode register. Through this register, a user can software select one of 16 sources as the counter input, a variety of gating and repetition modes, up or down counting in binary or BCD and active-high or active-low input and output polarities.

Associated with each counter are a Load register and a Hold register, both accessible through the Data port. The Load register is used to automatically reload the counter to any predefined value, thus controlling the effective count period. The Hold register is used to save count values without disturbing the count process, permitting the host processor to read intermediate counts. In addition, the Hold register may be used as a second Load register to generate a number of complex output waveforms.

All five counters have the same basic control logic and control registers. Counters 1 and 2 have additional Alarm registers and comparators associated with them, plus the extra logic necessary for operating in a 24-hour time-of-day mode. For real-time operation, the time-of-day logic will accept 50Hz, 60Hz or 100Hz input frequencies.

Each general counter has a single dedicated output pin. It may be turned off when the output is not of interest or may be configured in a variety of ways to drive interrupt controllers, Darlington buffers, bus drivers, etc. The counter inputs, on the other hand, are specifically not dedicated to any given interface line. Considerable versatility is available for configuring both the input and the gating of individual counters. This not only permits dynamic reassignment of inputs under software control, but also allows multiple counters to use a single input and a single gate pin to control more than one counter. Indeed, a single pin can be the gate for one counter and, at the same time, the count source for another.

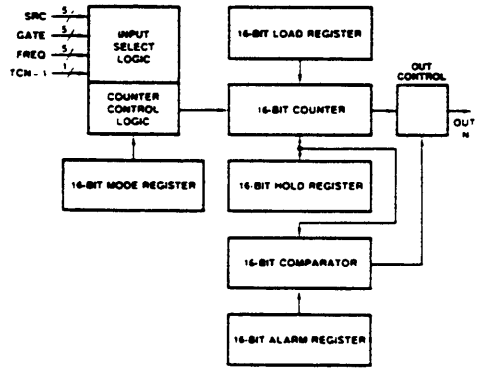


Figure 4. Counter Logic Groups 1 and 2

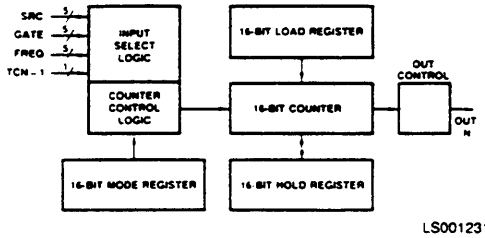


Figure 5. Counter Logic Groups 3, 4 and 5

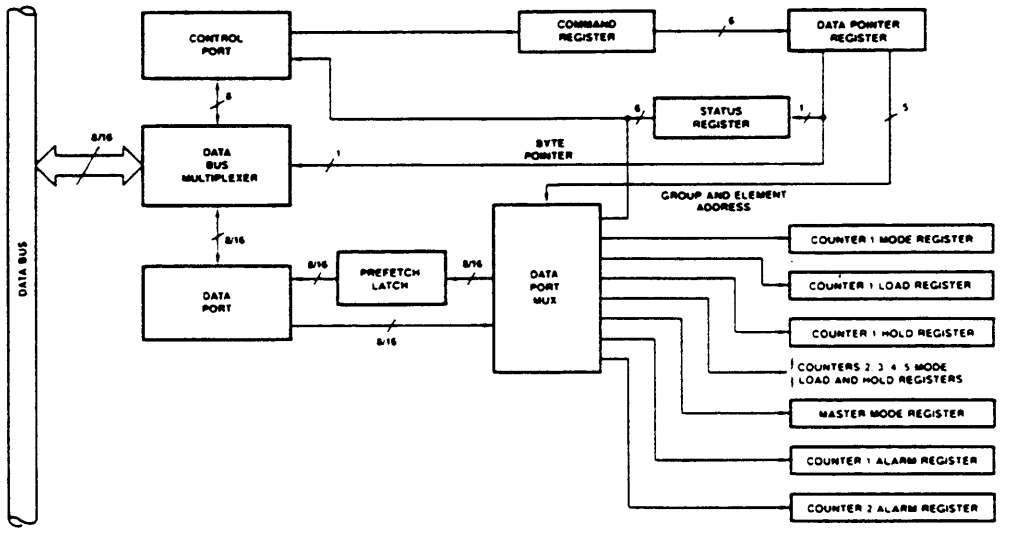


Figure 6. Am9513A Register Access

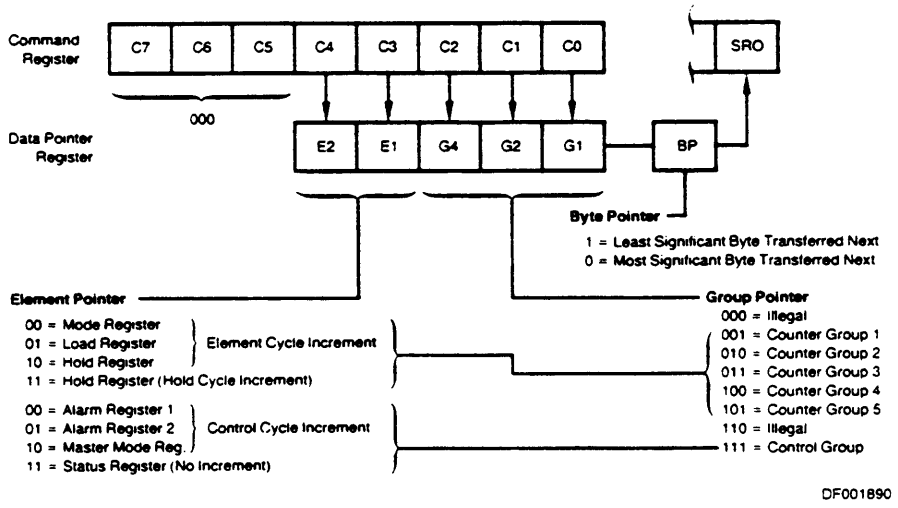


Figure 7. Data Pointer Register

2

	Element Cycle			Hold Cycle
	Mode Register	Load Register	Hold Register	Hold Register
Counter 1	FF01	FF09	FF11	FF19
Counter 2	FF02	FF0A	FF12	FF1A
Counter 3	FF03	FF0B	FF13	FF1B
Counter 4	FF04	FF0C	FF14	FF1C
Counter 5	FF05	FF0D	FF15	FF1D
Master Mode Register = FF17 Alarm 1 Register = FF07 Alarm 2 Register = FF0F Status Register = FF1F				

Notes:
1. All codes are in hex.
2. When used with an 8-bit bus, only the two low order hex digits should be written to the command port; the "FF" prefix should be used only for a 16-bit data bus interface.

Figure 8. Load Data Pointer Commands

Sequencing is enabled by clearing Master Mode bit 14 (MM14) to zero. As shown in Figure 9 several types of sequencing are available depending on the data bus width being used and the initial Data Pointer value entered by command.

When E1 = 0 or E2 = 0 and G4, G2, G1 points to a Counter Group, the Data Pointer will proceed through the Element cycle. The Element field will automatically sequence through the three values 00, 01 and 10 starting with the value entered. When the transition from 10 to 00 occurs, the Group field will also be incremented by one. Note that the Element field in this case does not sequence to a value of 11. The Group field circulates only within the five Counter Group codes.

If E2, E1 = 11 and a Counter Group are selected, then only the Group field is sequenced. This is the Hold cycle. It allows the Hold registers to be sequentially accessed while bypassing the Mode and Load registers. The third type of sequencing is the Control cycle. If G4, G2, G1 = 111 and E2, E1 ≠ 11, the Element Pointer will be incremented through the values 00, 01 and 10, with no change to the Group Pointer.

When G4, G2, G1 = 111 and E2, E1 = 11, no incrementing takes place and only the Status register will be available through the Data port. Note that the Status register can also always be read directly through the Control port.

For all these auto-sequencing modes, if an 8-bit data bus is used, the Byte pointer will toggle after every data transfer to allow the least and most significant bytes to be transferred before the Element or Group fields are incremented.

Prefetch Circuit

To minimize the read access time to internal Am9513A registers, a prefetch circuit is used for all read operations through the Data port. Following each read or write operation through the Data port, the Data Pointer register is updated to point to the next register to be accessed. Immediately following this update, the new register data is transferred to a special prefetch latch at the interface pad logic. When the user performs a subsequent read of the Data port, the data bus drivers are enabled, outputting the prefetched data on the bus. Since the internal data register is accessed prior to the start of the read operation, its access time is transparent to the user. To keep the prefetched data consistent with the Data Pointer, prefetches are also performed after each write to the Data port and after execution of the "Load Data Pointer" command. The following rules should be kept in mind regarding Data port Transfers.

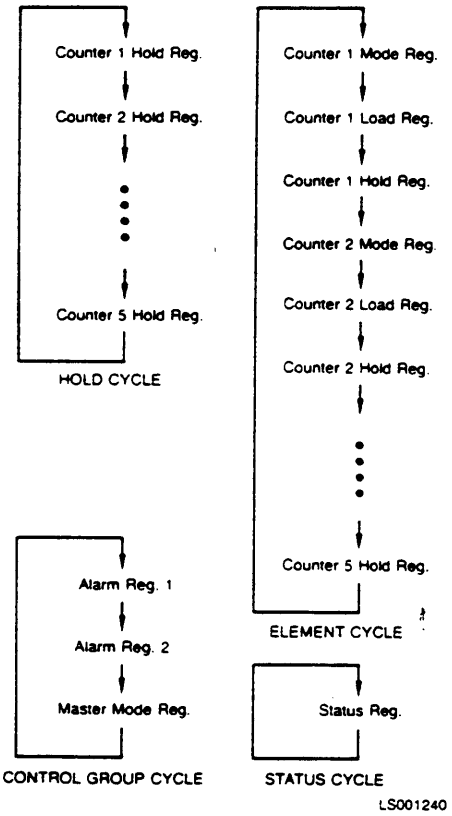


Figure 9. Data Pointer Sequencing

1. The Data Pointer register should always be reloaded before reading from the Data port if a command, other than "Load Data Pointer," was issued to the Am9513A following the last Data port read or write. The Data Pointer does not have to be loaded again if the first Data port transaction after a command entry is a write, since the Data port write will automatically cause a new prefetch to occur.
2. Operating modes N, O, Q, R and X allow the user to save the counter contents in the Hold register by applying an active-going gate edge. If the Data Pointer register had been pointing to the Hold register in question, the prefetched value will not correspond to the new value saved in the Hold Register. To avoid reading an incorrect value, a new "Load Data Pointer" command should be issued before attempting to read the saved data. A Data port write (to another register) will also initiate a prefetch; subsequent reads will access the recently saved Hold register data. Many systems will use the "saving" gate edge to interrupt the host CPU. In systems such as this, the interrupt service routine should issue a "Load Data Pointer" command prior to reading the saved data.

Status Register

The 8-bit read-only Status register indicates the state of the Byte Pointer bit in the Data Pointer register and the state of the

OUT signal for each of the general counters. See Figures 10 and 17. The OUT signals reported are those internal to the chip after the polarity-select logic and just before the three-state interface buffer circuitry. Bits SR6 and SR7 may be 0 or 1.

The Status register OUT bit reflects an active-high or active-low TC output or a TC Toggled output, as programmed in the Output Control Field of the Counter Mode register. That is, it reflects the exact state of the OUT pin. When the low-impedance to Ground Output option (CM2-CM0 = 000) is selected, the Status register will reflect an active-high TC Output. When a high-impedance Output option (CM2-CM0 = 100) is selected, the Status register will reflect an active-low TC output.

For Counters 1 and 2, the OUT pin will reflect the comparator output if the comparators are enabled. The Status register bit and OUT pin are active-high if CM2 = 0 and active-low if CM2 = 1. When the high-impedance option is selected and the comparator is enabled, the status register bit will reflect an active-high comparator output. When the low-impedance to Ground option is selected and the comparator is enabled, the status register bit will reflect an active-low comparator output.

The Status register is normally accessed by reading the Control port (see Figure 6) but may also be read via the Data port as part of the Control Group.

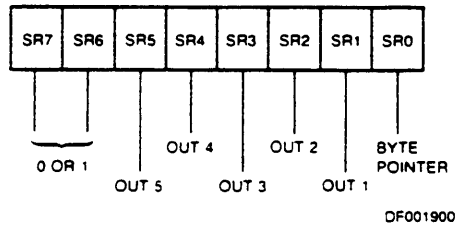


Figure 10. Status Register Bit Assignments

DATA PORT REGISTERS

Counter Logic Groups

As shown in Figures 4 and 5, each of the five Counter Logic Groups consists of a 16-bit general counter with associated control and output logic, a 16-bit Load register, a 16-bit Hold register and a 16-bit Mode register. In addition, Counter Groups 1 and 2 also include 16-bit Comparators and 16-bit Alarm registers. The comparator/alarm functions are controlled by the Master Mode register. The operation of the Counter Mode registers is the same for all five counters. The host CPU has both read and write access to all registers in the Counter Logic Groups through the Data port. The counter itself is never directly accessed.

Load Register

The 16-bit read/write Load register is used to control the effective length of the general counter. Any 16-bit value may be written into the Load register. That value can then be transferred into the counter each time the Terminal Count (TC) occurs. "Terminal Count" is defined as that period of time when the counter contents would have been zero if an external value had not been transferred into the counter. Thus, the terminal count frequency can be the input frequency divided by the value in the Load register. In all operating modes, either the Load or Hold register will be transferred into the counter when TC occurs. In cases where values are being accumulated in the counter, the Load register action can become transparent by filling the Load register with all zeros.

Hold Register

The 16-bit read/write Hold register is dual-purpose. It can be used in the same way as the Load register, thus offering an alternate source for module definition for the counter. The Hold register may also be used to store accumulated counter values for later transfer to the host processor. This allows the count to be sampled while the counting process proceeds without interruption. Transfer of the counter contents into the Hold register is accomplished by the hardware interface in some operating modes or by software commands at any time.

Counter Mode Register

The 16-bit read/write Counter Mode register controls the gating, counting, output and source select functions within each Counter Logic Group. The "Counter Mode Control Options" section of this document describes the detailed control options available. Figure 16 shows the bit assignments for the Counter Mode registers.

Alarm Registers and Comparators

Added functions are available in the Counter Logic Groups for Counters 1 and 2 (see Figure 4). Each contains a 16-bit Alarm register and a 16-bit Comparator. When the value in the counter reaches the value in the Alarm register, the Comparator output will go true. The Master Mode register contains control bits to individually enable/disable the comparators. When enabled, the comparator output appears on the OUT pin of the associated counter in place of the normal counter output. The output will remain true as long as the comparison is true, that is, until the next input causes the count to change. The polarity of the Comparator output will be active-high if the Output Control field of the Counter Mode register is 001 or 010 and active-low if the Output Control field is 101.

MASTER MODE CONTROL OPTIONS

The 16-bit Master Mode (MM) register is used to control those internal activities that are not controlled by the individual Counter Mode registers. This includes frequency control, Time-of-Day operation, comparator controls, data bus width and data pointer sequencing. Figure 11 shows the bit assignments for the Master Mode register. This section describes the use of each control field.

Master Mode register bits MM12, MM13 and MM14 can be individually set and reset using commands issued to the Command register. In addition, they can all be changed by writing directly to the Master Mode register.

After power-on reset or a Master Reset command, the Master Mode register is cleared to an all zero condition. This results in the following configuration:

Time-of-Day disabled
Both Comparators disabled
FOUT Source is frequency F1
FOUT Divider set for divide-by-16
FOUT gated on
Data Bus 8 bits wide
Data Pointer Sequencing enabled
Frequency Scaler divides in binary

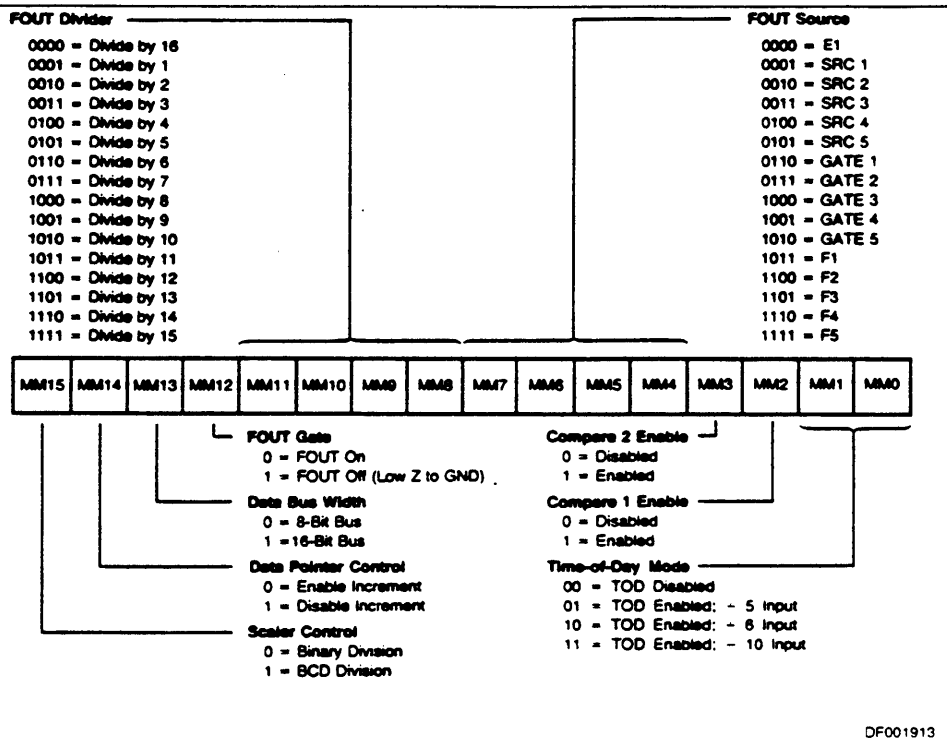


Figure 11. Master Mode Register Bit Assignments

Time-of-Day

Bits MM0 and MM1 of the Master Mode register specify the Time-of-Day (TOD) options. When MM0 = 0 and MM1 = 0, the special logic used to implement TOD is disabled, and Counters 1 and 2 will operate in exactly the same way as Counters 3, 4 and 5. When MM0 = 1 or MM1 = 1, additional counter decoding and control logic is enabled on Counters 1 and 2, which causes their decades to turn over at the counts that generate appropriate 24-hour TOD accumulations. For additional information, see the Time-of-Day chapter in the 9513A System timing controller technical manual.

Comparator Enable

Bits MM2 and MM3 control the Comparators associated with Counters 1 and 2. When a Comparator is enabled, its output is substituted for the normal counter output on the associated OUT1 or OUT2 pin. The comparator output will be active-high if the output control field of the Counter Mode register is 001 or 010 and active-low for a code of 101. Once the compare output is true, it will remain so until the count changes and the comparison therefore goes false.

The two Comparators can always be used individually in any operating mode. One special case occurs when the Time-of-Day option is revoked and both Comparators are enabled. The operation of Comparator 2 will then be conditioned by Comparator 1 so that a full 32-bit compare must be true in order to generate a true signal on OUT2. OUT1 will continue, as usual, to reflect the state of the 16-bit comparison between Alarm 1 and Counter 1.

FOUT Source

Master Mode bits MM4 through MM7 specify the source input for the FOUT divider. Fifteen inputs are available for selection, and they include the five Source pins, the five Gate pins and the five internal frequencies derived from the oscillator. The 16th combination of the four control bits (all zeros) is used to assure that an active frequency is available at the input to the FOUT divider following reset.

FOUT Divider

Bits MM8 through MM11 specify the dividing ratio for the FOUT Divider. The FOUT source (selected by bits MM4 through MM7) is divided by an integer value between 1 and 16, inclusive, and is then passed to the FOUT output buffer. After power-on or reset, the FOUT divider is set to divide-by-16.

FOUT Gate

Master Mode bit MM12 provides a software gating capability for the FOUT signal. When MM12 = 1, FOUT is off and in a low-impedance state to ground. MM12 may be set or cleared in conjunction with the loading of the other bits in the Master Mode register; alternatively, there are commands that allow MM12 to be individually set or cleared directly without changing any other Master Mode bits. After power-up or reset, FOUT is gated on.

When changing the FOUT divider ratio or FOUT source, transient pulses as short as half the period of the FOUT source may appear on the FOUT pin. Turning the FOUT gate on or off can also generate a transient. This should be considered when using FOUT as a system clock source.

Bus Width

Bit MM13 controls the multiplexer at the data bus interface in order to configure the part for an 8-bit or 16-bit external bus. The internal bus is always 16-bits wide. When MM13 = 1, 16-bit data is transferred directly between the internal bus and all 16 of the external bus lines. In this configuration, the Byte Pointer bit in the Data Pointer register remains set at all times. When MM13 = 0, 16-bit internal data is transferred a byte at a time to and from the eight low-order external data bus lines. The Byte Pointer bit toggles with each byte transfer in this mode.

When the Am9513A is set to operate with an 8-bit data bus width, pins DB8 through DB15 are not used for the data bus and are available for other functions. Pins DB13 through DB15 should be tied high. Pins DB8 through DB12 are used as auxiliary gating inputs and are labeled GATE1A through GATE5A respectively. The auxiliary gate pin, GATENA, is logically ANDed with the gate input to Counter N, as shown in Figure 12. The output of the AND gate is then used as the gating signal for Counter N.

Data Pointer Sequencing

Bit MM14 controls the Data Pointer logic to enable or disable the automatic sequencing functions. When MM14 = 1, the contents of the Data Pointer can be changed only directly by entering a command. When MM14 = 0, several types of automatic sequencing of the Data Pointer are available. These are described in the Data Pointer register section of this document.

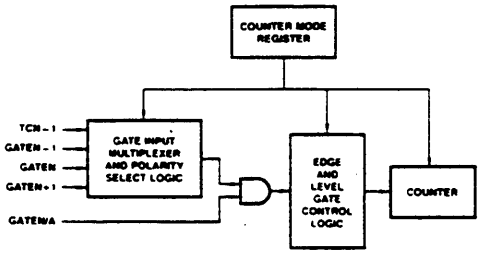
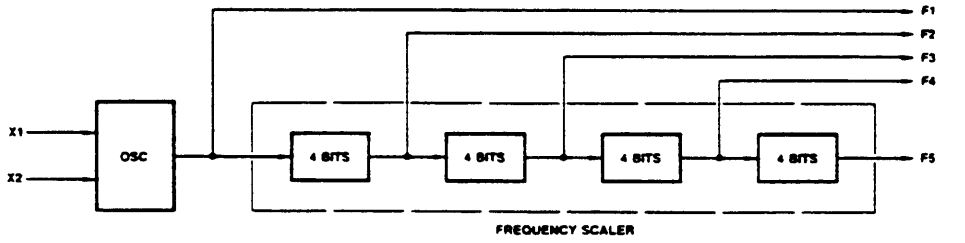


Figure 12. Gating Control

Thus, the host processor, by controlling MM14, may repetitively read/write a single internal location, or may sequentially read/write groups of locations. Bit MM14 can be loaded by writing to the Master Mode register or can be set or cleared by software command.

Scaler Ratios

Master Mode bit MM15 controls the counting configuration of the Frequency Scaler counter. When MM15 = 0, the Scaler divides the oscillator frequency in binary steps so that each subfrequency is 1/16 of the preceding frequency. When MM15 = 1, the Scaler divides in BCD steps so that adjacent frequencies are related by ratios of 10 instead of 16 (see Figure 13).



Frequency	BCD Scaling MM15 = 1	Binary Scaling MM15 = 0
F1	OSC	OSC
F2	F1 ÷ 10	F1 ÷ 16
F3	F1 ÷ 100	F1 ÷ 256
F4	F1 ÷ 1,000	F1 ÷ 4,096
F5	F1 ÷ 10,000	F1 ÷ 65,536

Figure 13. Frequency Scaler Ratios

Counter Mode	A	B	C	D	E	F	G	H	I	J	K	L
Special Gate (CM7)	0	0	0	0	0	0	0	0	0	0	0	0
Reload Source (CM6)	0	0	0	0	0	0	1	1	1	1	1	1
Repetition (CM5)	0	0	0	1	1	1	0	0	0	1	1	1
Gate Control (CM15–CM13)	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE
Count to TC once, then disarm	X	X	X									
Count to TC twice, then disarm							X	X	X			
Count to TC repeatedly without disarming				X	X	X				X	X	X
Gate input does not gate counter input	X			X			X			X		
Count only during active gate level		X			X			X			X	
Start count on active gate edge and stop count on next TC			X			X						
Start count on active gate edge and stop count on second TC									X			X
No hardware retriggering	X	X	X	X	X	X	X	X	X	X	X	X
Reload counter from Load register on TC	X	X	X	X	X	X						
Reload counter on each TC, alternating reload source between Load and Hold registers							X	X	X	X	X	X
Transfer Load register into counter on each TC that gate is LOW, transfer Hold register into counter on each TC that gate is HIGH.												
On active gate edge transfer counter into Hold register and then reload counter from Load register												
Counter Mode	M	N	O	P	Q	R	S	T	U	V	W	X
Special Gate (CM7)	1	1	1	1	1	1	1	1	1	1	1	1
Reload Source (CM6)	0	0	0	0	0	0	1	1	1	1	1	1
Repetition (CM5)	0	0	0	1	1	1	0	0	0	1	1	1
Gate Control (CM15–CM13)	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE
Count to TC once, then disarm		X	X									
Count to TC twice, then disarm							X					
Count to TC repeatedly without disarming					X	X				X		X
Gate input does not gate counter input							X			X		
Count only during active gate level		X			X							
Start count on active gate edge and stop count on next TC			X			X						X
Start count on active gate edge and stop count on second TC												
No hardware retriggering							X			X		X
Reload counter from Load register on TC		X	X		X	X						X
Reload counter on each TC, alternating reload source between Load and Hold registers.												
Transfer Load register into counter on each TC that gate is LOW, transfer Hold register into counter on each TC that gate is HIGH.							X			X		
On active gate edge transfer counter into Hold register and then reload counter from Load register		X	X		X	X						
On active gate edge transfer counter into Hold register, but counting continues												X

Notes: 1. Counter modes M, P, T, U and W are reserved and should not be used.
2. Mode X is available for Am9513A only.

Figure 14. Counter Mode Operating Summary

COUNTER MODE DESCRIPTIONS

Counter Mode register bits CM15–CM13 and CM7–CM5 select the operating mode for each counter (see Figure 14). To simplify references to a particular mode, each mode is assigned a letter from A through X. Representative waveforms for the counter modes are illustrated in Figures 15a through 15v. (Because the letter suffix in the figure number is keyed to the mode, Figures 15m, 15p, 15t, 15u and 15w do not exist.) The figures assume down counting on rising source edges. Those modes which automatically disarm the counter (CM5 = 0) are shown with the $\overline{WR}$ plus entering the required ARM command; for modes which count repetitively (CM5 = 1), the ARM command is omitted. The retriggering modes (N, O, Q and R) are shown with one retrigger operation. Both a TC output waveform and a TC Toggled output waveform are shown for each mode. The symbols L and H are used to represent count values equal to the Load and Hold register contents, respectively. The symbols K and N represent arbitrary count values. For each mode, the required bit pattern in the Counter Mode register is shown; "don't care" bits are marked "X." These figures are designed to clarify the mode descriptions; the Am9513A Electrical Specification should be used as the authoritative reference for timing relationships between signals.

To keep the following mode descriptions concise and to the point, the phrase "source edges" is used to refer to active-going source edges only, not to inactive-going edges. Similarly, the phrase "gate edges" refers only to active-going gate edges. Also, again to avoid verbosity and euphuism, the descriptions of some modes state that a counter is stopped or disarmed "on a TC, inhibiting further counting." As is fully explained in the TC section of this document, for these modes the counter is actually stopped or disarmed following the active-going source edge which drives the counter out of TC. In other words, since a counter in the TC state always counts, irrespective of its gating or arming status, the stopping or disarming of the count sequence is delayed until TC is terminated.

MODE A

Software-Triggered Strobe with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode A, shown in Figure 15a, is one of the simplest operating modes. The counter will be available for counting source edges when it is issued an ARM command. On each TC, the counter will reload from the Load register and automatically disarm itself, inhibiting further counting. Counting will resume when a new ARM command is issued.

MODE B

Software-Triggered Strobe with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode B, shown in Figure 15b, is identical to Mode A except that source edges are counted only when the assigned Gate is active. The counter must be armed before counting can occur. Once armed, the counter will count all source edges which occur while the Gate is active and disregard those edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. On each TC the counter will reload from the Load register and automatically disarm itself, inhibiting further counting until a new ARM command is issued.

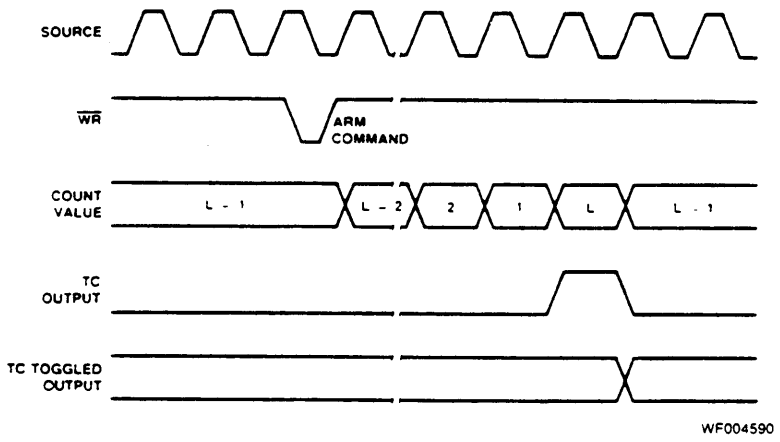


Figure 15a. Mode A Waveforms

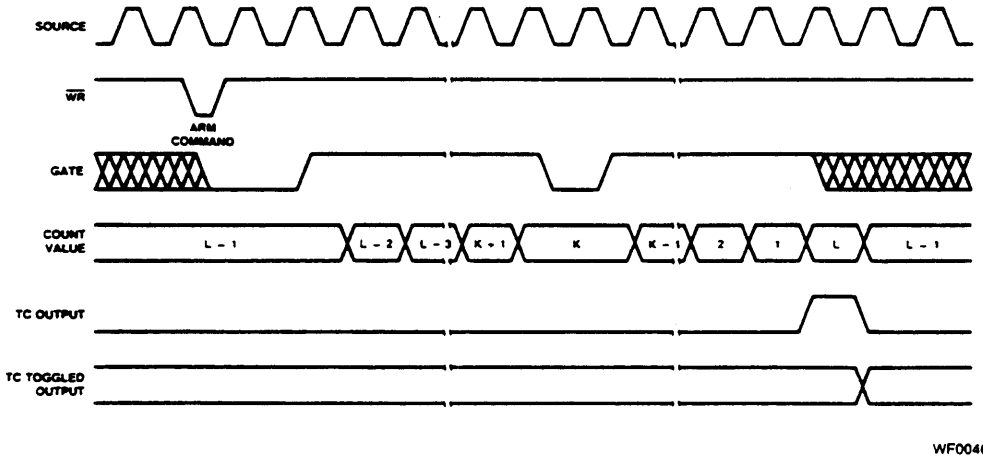


Figure 15b. Mode B Waveforms

MODE C

Hardware-Triggered Strobe

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode C, shown in Figure 15c, is identical to Mode A, except that counting will not begin until a Gate edge is applied to the armed counter. The counter must be armed before application of the triggered Gate edge; Gate edges applied to a disarmed counter are disregarded. The counter will start counting on the first source edge after the triggering Gate edge and will continue counting until TC. At TC, the counter will reload from the Load register and automatically disarm itself. Counting will then remain inhibited until a new ARM command and a new Gate edge are applied in that order. Note that after application of a triggered Gate edge, the Gate input will be disregarded for the remainder of the count cycle. This differs from Mode B, where the Gate can be modulated throughout the count cycle to stop and start the counter.

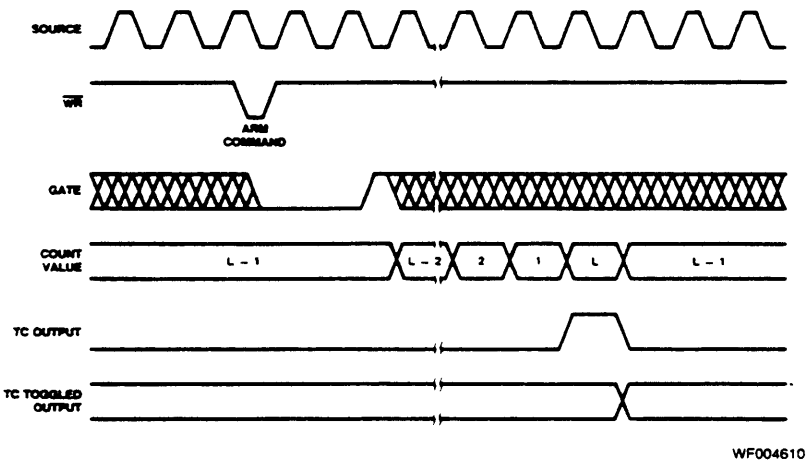


Figure 15c. Mode C Waveforms

MODE D

Rate Generator with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode D, shown in Figure 15d, is typically used in frequency generation applications. In this mode, the Gate input does not affect counter operation. Once armed, the counter will count to TC repetitively. On each TC, the counter will reload itself from the Load register; hence, the Load register value determines the time between TCs. A square wave rate generator may be obtained by specifying the TC Toggled output mode in the Counter Mode register.

MODE E

Rate Generator with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode E, shown in Figure 15e, is identical to Mode D, except the counter will only count those source edges which occur while the Gate input is active. This feature allows the counting process to be enabled and disabled under hardware control. A square wave rate generator may be obtained by specifying the TC Toggled output mode.

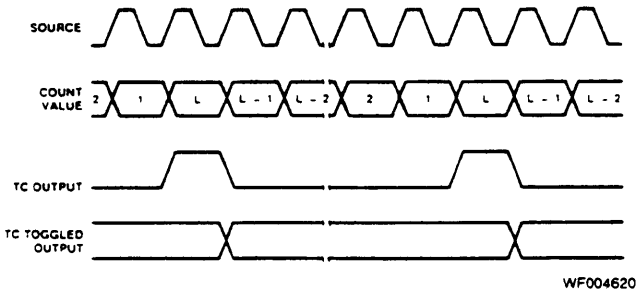


Figure 15d. Mode D Waveforms

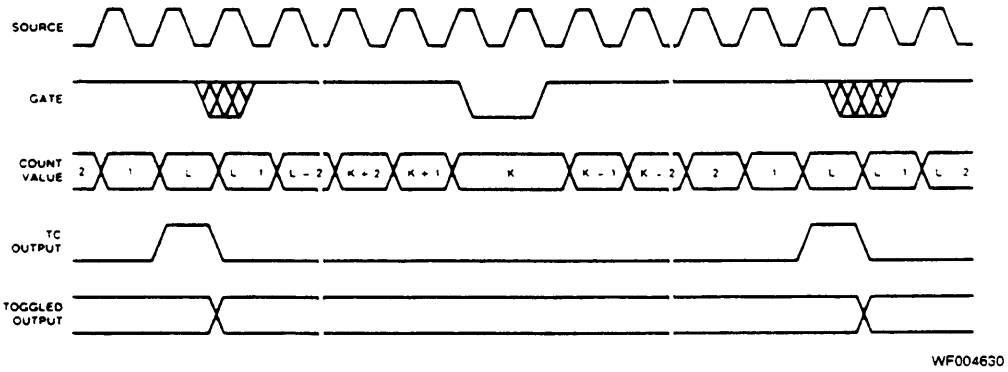


Figure 15e. Mode E Waveforms

MODE F

Non-Retriggerable One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode F, shown in Figure 15f, provides a non-retriggerable one-shot timing function. The counter must be armed before it will function. Application of a Gate edge to the armed counter will enable counting. When the counter reaches TC, it will reload itself from the Load register. The counter will then stop counting, awaiting a new Gate edge. Note that unlike Mode C, a new ARM command is not needed after TC, only a new Gate edge. After application of a triggering Gate edge, the Gate input is disregarded until TC.

MODE G

Software-Triggered Delayed Pulse One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

In Mode G, the Gate does not affect the counter's operation. Once armed, the counter will count to TC twice and then automatically disarm itself. For most applications, the counter will initially be loaded from the Load register either by a LOAD command or by the last TC of an earlier timing cycle. Upon counting to the first TC, the counter will reload itself from the Hold register. Counting will proceed until the second TC, when the counter will reload itself from the Load register and automatically disarm itself, inhibiting further counting. Counting can be resumed by issuing a new ARM command. A software-triggered delayed pulse one-shot may be generated by specifying the TC Toggled output mode in the Counter Mode register. The initial counter contents control the delay from the ARM command until the output pulse starts. The Hold register contents control the pulse duration. Mode G is shown in Figure 15g.

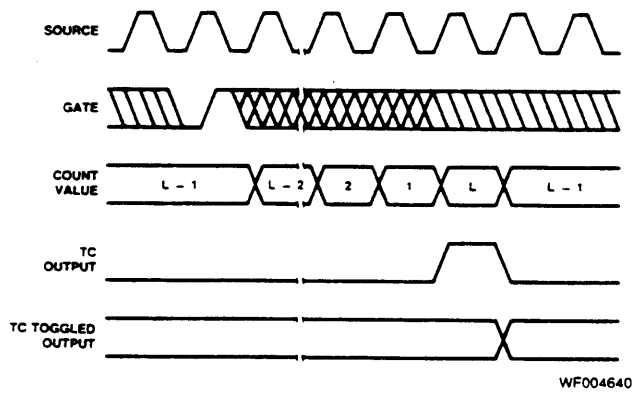


Figure 15f. Mode F Waveforms

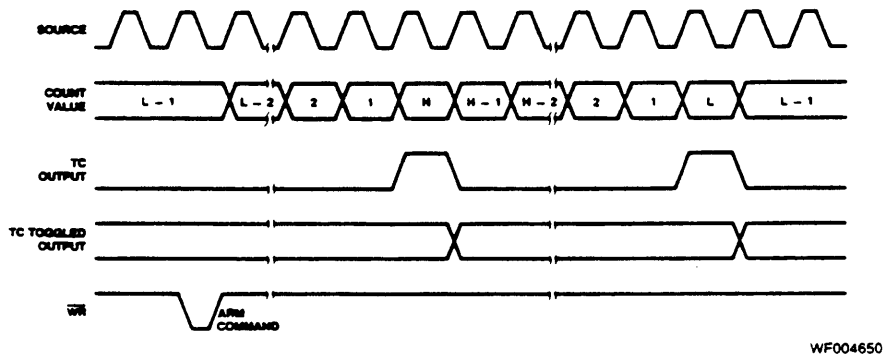


Figure 15g. Mode G Waveforms

MODE H

Software-Triggered Delayed Pulse One-Shot with Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

Mode H, shown in Figure 15h, is identical to Mode G except that the Gate input is used to qualify which source edges are to be counted. The counter must be armed for counting to occur. Once armed, the counter will count all source edges that occur while the Gate is active. This permits the Gate to turn the count process on and off. As with Mode G, the counter will be reloaded from the Hold register on the first TC and reloaded from the Load register and disarmed on the second TC. This mode allows the Gate to control the extension of both the initial output delay time and the pulse width.

MODE I

Hardware-Triggered Delayed Pulse Strobe

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

Mode I, shown in Figure 15i, is identical to Mode G, except that counting will not begin until a Gate edge is applied to an armed counter. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. An armed counter will start counting on the first source edge after the triggering Gate edge. Counting will then proceed in the same manner as in Mode G. After the second TC, the counter will disarm itself. An ARM command and Gate edge must be issued in this order to restart counting. Note that after application of a triggering Gate edge, the Gate input will be disregarded until the second TC. This differs from Mode H, where the Gate can be modulated throughout the count cycle to stop and start the counter.

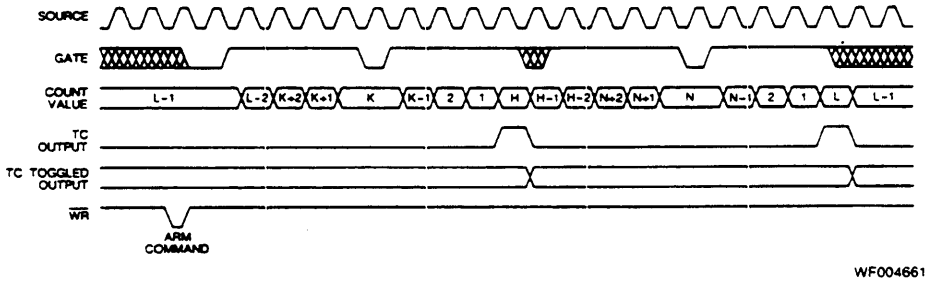


Figure 15h. Mode H Waveforms

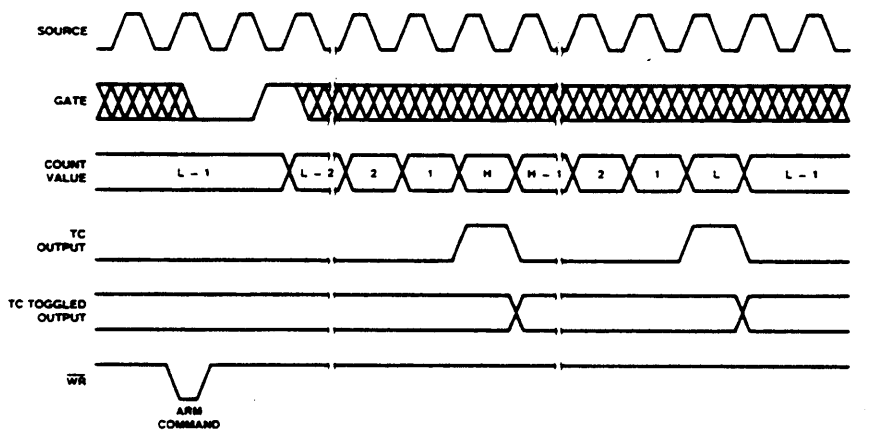


Figure 15i. Mode I Waveforms

MODE J

Variable Duty Cycle Rate Generator with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode J, shown in Figure 15j, will find the greatest usage in frequency generation applications with variable duty cycle requirements. Once armed, the counter will count continuously until it is issued a DISARM command. On the first TC, the counter will be reloaded from the Hold register. Counting will then proceed until the second TC at which time the counter will be reloaded from the Load register. Counting will continue, with the reload source alternating on each TC, until a DISARM command is issued to the counter. (The third TC reloads from the Hold register, the fourth TC reloads from the Load register, etc.) A variable duty cycle output can be generated by specifying the TC Toggled output in the Counter Mode register. The Load and Hold values then directly control the output duty cycle, with high resolution available when relatively high count values are used.

MODE K

Variable Duty Cycle Rate Generator with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode K, shown in Figure 15k, is identical to Mode J, except that source edges are only counted when the Gate is active. The counter must be armed for counting to occur. Once armed, the counter will count all source edges which occur while Gate is active and disregard those source edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. As with Mode J, the reload source used will alternate on each TC, starting with the Hold register on the first TC after any ARM command. When the TC Toggled output is used, this mode allows the Gate to modulate the duty cycle of the output waveform. It can affect both the HIGH and LOW portions of the output waveform.

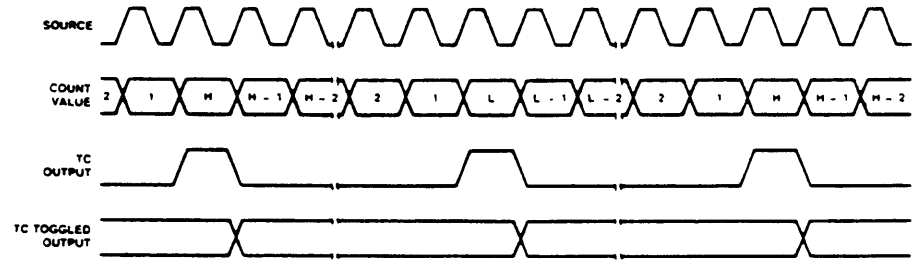


Figure 15j. Mode J Waveforms

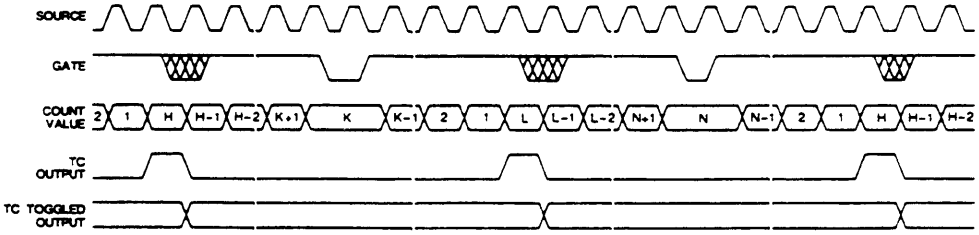


Figure 15k. Mode K Waveforms

2

MODE L

Hardware-Triggered Delayed Pulse One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode L, shown in Figure 15l, is similar to Mode J except that counting will not begin until a Gate edge is applied to an armed counter. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. The counter will start counting source edges after the triggering Gate edge, and counting will proceed until the second TC. Note that after application of a triggering Gate edge, the Gate input will be disregarded for the remainder of the count cycle. This differs from Mode K, where the gate can be modulated throughout the count cycle to stop and start the counter. On the first TC after application of the triggering Gate edge, the counter will be reloaded from the Hold register. On the second TC, the counter will be reloaded from the Load register, and counting will stop until a new gate edge is issued to the counter. Note that unlike Mode K, new Gate edges are required after every second TC to continue counting.

MODE N

Software-Triggered Strobe with Level Gating and Hardware Retriggering

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	0	X	X	X	X	X

Mode N, shown in Figure 15n, provides a software-triggered strobe with level gating that is also hardware retriggerable. The counter must be issued an ARM command before counting can occur. Once armed, the counter will count all source edges which occur while the gate is active and disregard those source edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. After the issuance of the ARM command and the application of an active Gate, the counter will count to TC. Upon reaching TC, the counter will reload from the Load register and automatically disarm itself, inhibiting further counting. Counting will resume upon the issuance of a new ARM command. All active-going Gate edges issued to an armed counter will cause a retrigger operation. Upon application of the Gate edge, the counter contents will be saved in the Hold register. On the first qualified source edge after application of the retriggering gate edge, the contents of the Load register will be transferred into the counter. Counting will resume on the second qualified source edge after the retriggering Gate edge. Qualified source edges are active-going edges which occur while the Gate is active.

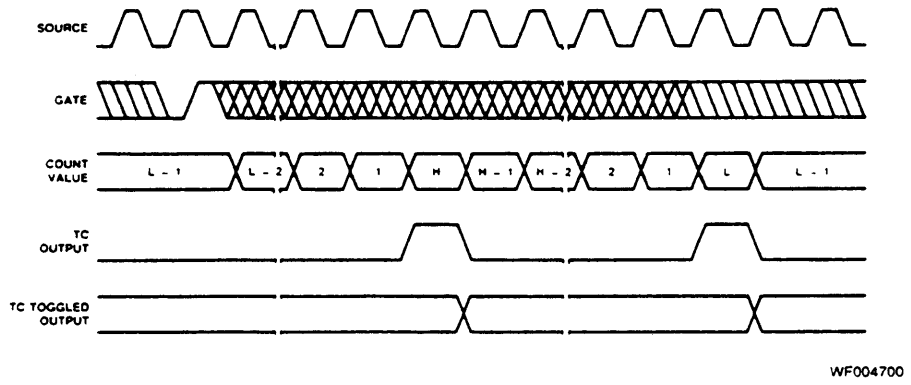


Figure 15l. Mode L Waveforms

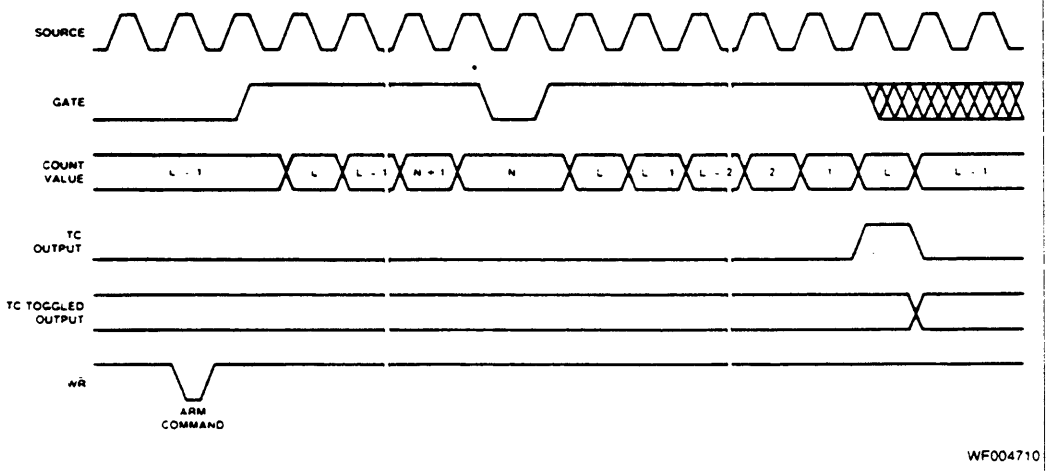


Figure 15n. Mode N Waveforms

MODE O

Software-Triggered Strobe with Edge Gating and Hardware Retriggering

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	0	X	X	X	X	X

Mode O, shown in Figure 15o, is similar Mode N, except that counting will not begin until an active-going Gate edge is applied to an armed counter and the Gate level is not used to modulate counting. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. Irrespective of the Gate level, the counter will count all source edges after the triggering Gate edge until the first TC. On the first TC,the counter will be reloaded from the Load register and disarmed. A new ARM command and a new Gate edge must be applied in that order to initiate a new counting cycle. Unlike Modes C, F, I and L, which disregard the Gate input once counting starts, in Mode O the count process will be retriggered on all active-going Gate edges, including the first Gate edge used to start the counter. On each retriggering Gate edge, the counter contents will be transferred into the Hold register. On the first source edge after the retriggering Gate edge, the Load register contents will be transferred into the counter. Counting will resume on the second-source edge after a retrigger.

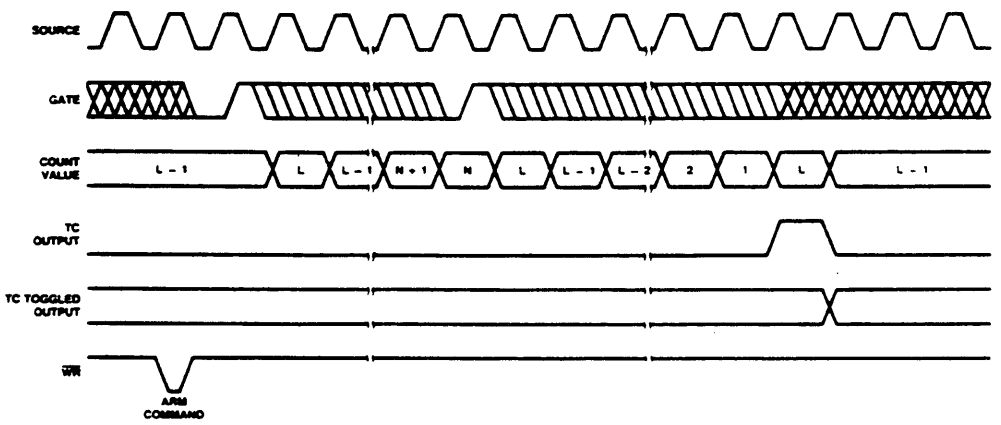


Figure 15o. Mode O Waveforms

MODE Q

Rate Generator with Synchronization (Event Counter with Auto-Read/Reset)

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	1	X	X	X	X	X

Mode Q, shown in Figure 15q, provides a rate generator with synchronization or an event counter with auto-read/reset. The counter must first be issued an ARM command before counting can occur. Once armed, the counter will count all source edges which occur while the Gate is active and disregard those edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. After the issuance of an ARM command and the application of an active Gate, the counter will count to TC repetitively. On each TC,the counter will reload itself from the Load register. The counter may be retriggered at any time by presenting an active-going Gate edge to the Gate input. The retriggering Gate edge will transfer the contents of the counter into the Hold register. The first qualified source edge after the retriggering Gate edge will transfer the contents of the Load register into the Counter. Counting will resume on the second qualified source edge after the retriggering Gate edge. Qualified source edges are active-going edges which occur while the Gate is active.

MODE R

Retriggerable One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	1	X	X	X	X	X

Mode R, shown in Figure 15r, is similar to Mode Q, except that edge gating rather than level gating is used. In other words, rather than use the Gate level to qualify which source edges to count, Gate edges are used to start the counting operation. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. After application of a Gate edge, an armed counter will count all source edges until TC, irrespective of the Gate level. On the first TC, the counter will be reloaded from the Load register and stopped. Subsequent counting will not occur until a new Gate edge is applied. All Gate edges applied to the counter, including the first used to trigger counting, initiate a retrigger operation. Upon application of a Gate edge, the counter contents are saved in the Hold register. On the first source edge after the retriggering Gate edge, the Load register contents will be transferred into the counter. Counting will resume on the second source edge after the retriggering Gate edge.

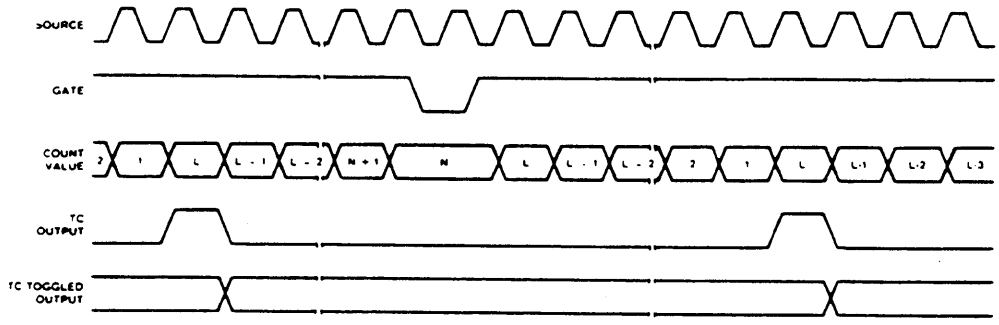


Figure 15q. Mode Q Waveforms

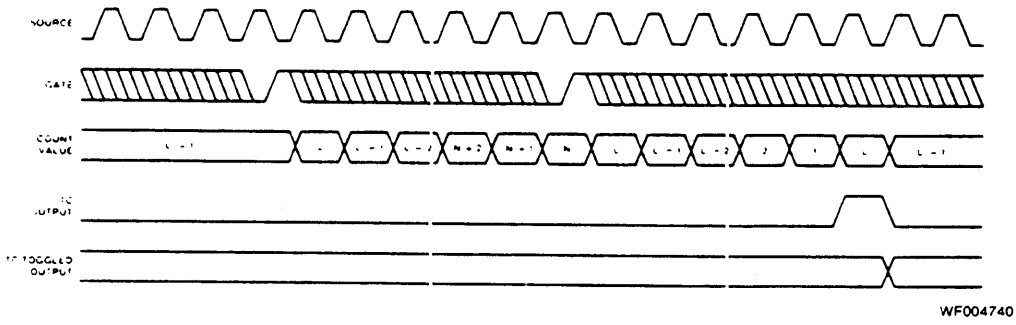


Figure 15r. Mode R Waveforms

MODE S

RELOAD SOURCE

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	0	X	X	X	X	X

In this mode, the reload source for LOAD commands (irrespective of whether the counter is armed or disarmed) and for TC-initiated reloads is determined by the Gate input. The Gate input in Mode S is used only to select the reload source, not to start or modulate counting. When the Gate is Low, the Load register is used; when the Gate is High, the Hold register is used. Note the Low-Load, High-Hold mnemonic convention. Once armed, the counter will count to TC twice and then disarm itself. On each TC, the counter will be reloaded from the reload source selected by the Gate. Following the second TC, an ARM command is required to start a new counting cycle. Mode S is shown in Figure 15s.

MODE V

Frequency-Shift Keying

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	1	X	X	X	X	X

Mode V, shown in Figure 15v, provides frequency-shift keying modulation capability. Gate operation in this mode is identical to that in Mode S. If the Gate is Low, a LOAD command or a TC-induced reload will reload the counter from the Load register. If the Gate is HIGH, LOADs and reloads will occur from the Hold register. The polarity of the Gate only selects the reload source; it does not start or modulate counting. Once armed, the counter will count repetitively to TC. On each TC, the counter will reload itself from the register determined by the polarity of the Gate. Counting will continue in this manner until a DISARM command is issued to the counter. Frequency shift keying may be obtained by specifying a TC Toggled output mode in the Counter Mode register. The switching of frequencies is achieved by modulating the Gate.

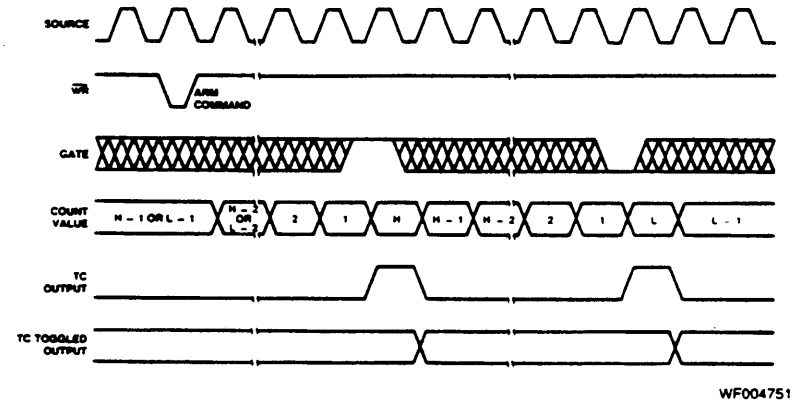


Figure 15s. Mode S Waveforms

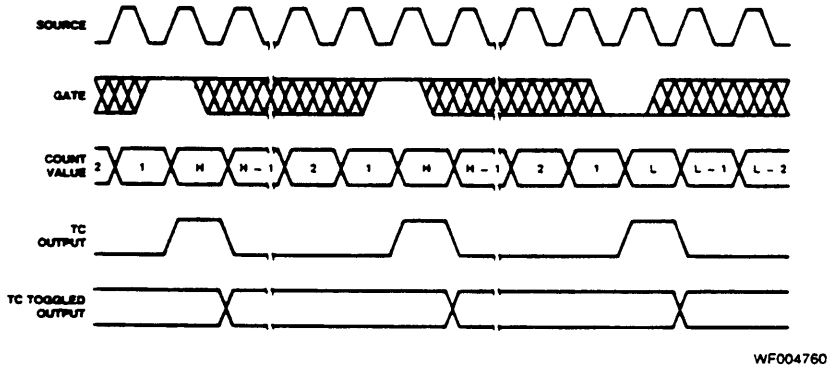


Figure 15v. Mode V Waveforms

2

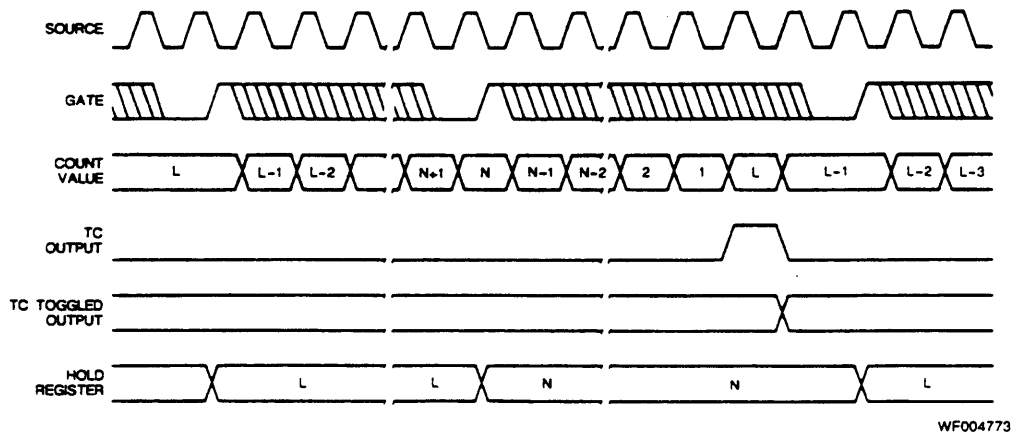


Figure 15x. Mode X Waveforms

MODE X

Hardware Save (available in Am9513A only)

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
Edge			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	1	X	X	X	X	X

Mode X, as shown in Figure 15x, provides a hardware sampling of the counter contents without interrupting the count. A LOAD AND ARM command or a LOAD command followed by an ARM command is required to initialize the counter. Once armed, a Gate edge starts the counting operation; Gate edges applied to a disarmed counter are disregarded. After application of the Triggering Gate edge, the counter will count all qualified source edges until the first TC, irrespective of the gate level. All gate edges applied during the counting sequence will store the current count in the Hold register, but they will not interrupt the counting sequence. On each TC, the counter will be reloaded from the Load register and stopped. Subsequent counting requires a new triggering Gate edge; counting resumes on the first source edge following the triggering Gate edge.

Note: Mode X is only available in the Am9513A devices.

COUNTER MODE CONTROL OPTIONS

Each Counter Logic Group includes a 16-bit Counter Mode (CM) register used to control all of the individual options available with its associated general counter. These options include output configuration, count control, count source and gating control. Figure 16 shows the bit assignments for the Counter Mode registers. This section describes the control options in detail. Note that generally each counter is independently configured and does not depend on information outside its Counter Logic Group. The Counter Mode register should be loaded only when the counter is Disarmed. Attempts to load the Counter Mode register when the counter is armed may result in erratic counter operation.

After power-on reset or a Master Reset command, the Counter Mode registers are initialized to a preset condition. The value entered is 0B00 hex and results in the following control configuration:

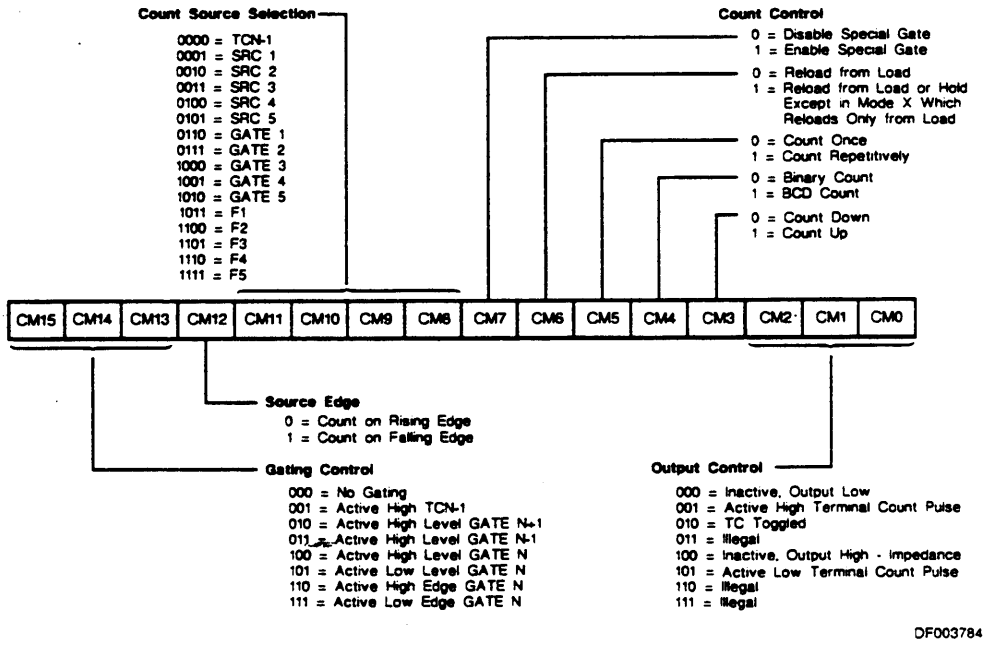
Output low-impedance to ground
Count down
Count binary
Count once
Load register selected
No retriggering
F1 input source selected
Positive-true input polarity
No gating

Output Control

Counter mode bits CM0 through CM2 specify the output control configuration. Figure 17 shows a schematic representation of the output control logic. The OUT pin may be off (a high-impedance state), or it may be inactive with a low-impedance to ground. The three remaining valid combinations represent the active-high, active-low or TC Toggle output waveforms.

One output form available is called Terminal Count (TC) and represents the period in time that the counter reaches an equivalent value of zero. TC will occur on the next count when the counter is at 0001 for down counting, at 9999 (BCD) for BCD up counting or at FFFF (hex) for binary up counting. Figure 18 shows a Terminal Count pulse and an example context that generated it. The TC width is determined by the period of the counting source. Regardless of any gating input or whether the counter is Armed or Disarmed, the terminal count will go active for only one clock cycle. Figure 18 assumes active-high source polarity, counter armed, counter decrementing and an external reload value of K.

The counter will always be loaded from an external location when TC occurs; the user can choose the source location and the value. If a non-zero value is picked, the counter will never really attain a zero state, and TC will indicate the counter state that would have been zero had no parallel transfer occurred.



Note: See Figure 15 for restrictions on Count Control and Gating Control bit combinations.

Figure 16. Counter Mode Register Bit Assignments

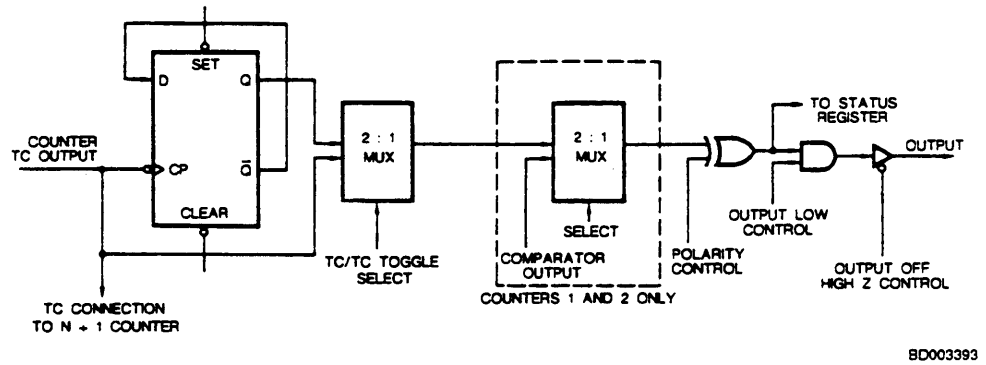


Figure 17. Output Control Logic

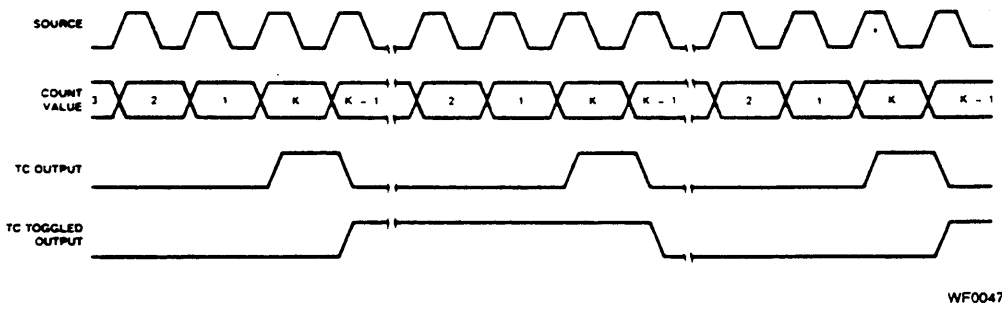


Figure 18. Counter Output Waveforms

The other output form, TC Toggled, uses the trailing edge of TC to toggle a flip-flop to generate an output level instead of a pulse. The toggle output is half the frequency of TC. The TC Toggled output will frequently be used to generate variable duty-cycle square waves in Operating Modes G through K.

In Mode L the TC Toggled output can be used to generate a one-shot function, with the delay to the start of the output pulse and the width of the output pulse separately programmable. With selection of the minimum delay to the start of the pulse, the output will toggle on the second source pulse following application of the triggering Gate edge.

Note that the TC Toggled output form contains no implication about whether the output is active-high or active-low. Unlike the TC output, which generates a transient pulse which can clearly be active-high or active-low, the TC Toggled output waveform only flips the state of the output on each TC. The sole criterion of whether the TC Toggled output is active-high or active-low is the level of the output at the start of the count cycle. This can be controlled by the Set and Clear Output commands. (See Figure 19.)

TC (Terminal Count)

On each Terminal Count (TC), the counter will reload itself from the Load or Hold register. TC is defined as that period of time when the counter contents would have been zero had no reload occurred. Some special conditions apply to counter operation immediately before and during TC.

1. In the clock cycle before TC, an internal signal is generated that commits the counter to go to TC on the next count, and retriggering by a hardware Gate edge (Modes N, O, Q and R) or a software LOAD or LOAD AND ARM command will not extend the time to TC. Note that the "next count" driving the counter to TC can be caused by the application of a count source edge (in level gating modes, the edge must occur while the gate is active, or it will be disregarded), by the application of a LOAD or LOAD AND ARM command (see 2 below) or by the application of a STEP command.
2. If a LOAD or LOAD AND ARM command is executed during the cycle preceding TC, the counter will immediately go to TC. If these commands are issued during TC, the TC state will immediately terminate.
3. When TC is active, the counter will always count the next source edge issued to it, even if it is disarmed or gated off during TC. This means that TC will never be active for longer than one count period and it may, in fact, be shorter if a STEP command or a LOAD or LOAD AND ARM command is applied during TC (see item 2 above). This also means that a counter that is disarmed or stopped on TC is actually disarmed/stopped immediately following TC.

This may cause count sequences different from what a user might expect. Since the counter is always reloaded at the start of TC and since it always counts at the end of TC, the counter contents following TC will differ by one from the reloaded value, irrespective of the operating mode used.

If the reloaded value was 0001 for down counting, 9999 (BCD) for BCD up counting or FFFF (hex) for binary up counting, the count at the end of TC will drive the counter into TC again regardless of whether the counter is gated off or disarmed. As long as these values are reloaded, the TC output will stay active. If a TC Toggled output is selected, it will toggle on each count. Execution of a LOAD, LOAD AND ARM or STEP command with these counter contents will act the same as application of a source pulse, causing TC to remain active and a TC Toggled output to toggle.

Count Control

Counter Mode bits CM3 through CM7 specify the various options available for direct control of the counting process. CM3 and CM4 operate independently of the others and control up/down and BCD/binary counting. They may be combined freely with other control bits to form many types of counting configurations. The other three bits and the Gating Control field interact in complex ways. Bit CM5 controls the repetition of the count process. When CM5 = 1, counting will proceed in the specified mode until the counter is disarmed. When CM5 = 0, the count process will proceed only until one full cycle of operation occurs. This may occur after one or two TC events. The counter is then disarmed automatically. The single or double TC requirement will depend on the state of other control bits. Note that even if the counter is automatically disarmed upon a TC, it always counts the count source edge which generates the trailing TC edge.

When TC occurs, the counter is always reloaded with a value from either the Load register of the Hold register. Bit CM6 specifies the source options for reloading the counter. When CM6 = 0, the contents of the Load register will be transferred into the counter at every occurrence of TC. When CM6 = 1, the counter reload location will be either the Load or Hold Register. The reload location in this case may be controlled externally by using a Gate pin (Modes S and V) or may alternate on each TC (Modes G through L). With alternating sources and with the TC Toggled output selected, the duty cycle of the output waveform is controlled by the relative Load and Hold values and very fine resolution of duty cycles ratios may be achieved.

Bit CM7 controls the special gating functions that allow retriggering and the selection of Load or Hold sources for counter reloading. The use and definition of CM7 will depend

on the status of the Gating Control field and bits CM5 and CM6.

Hardware Retriggering

Whenever hardware retriggering is enabled (Modes N, O, Q, and R), all active-going Gate edges initiate retrigger operations. On application of the Gate edge, the counter contents will be transferred to the Hold register. On the first qualified source edge after application of the retriggering Gate edge, the Load register contents will be transferred into the counter. (Qualified source edges are edges which occur while the counter is gated on and Armed.)

This means that, if level gating is used, the edge occurring on active-going gate transitions will initiate a retrigger. Similarly, when edge gating is enabled, an edge used to start the counter will also initiate a retrigger. The first count source edge applied after the Gate edge will not increment/decrement the counter but retrigger it.

If a LOAD, LOAD AND ARM, or a STEP Command occurs between the retriggering Gate edge and the first qualified source edge, it will be interpreted as a source edge and transfer the Load register contents into the counter. Thereafter, the counter will count all qualified source edges.

When some form of Gating is specified, CM7 controls hardware retriggering. In this case, when CM7 = 0, hardware retriggering does not occur; when CM7 = 1, the counter is retriggered any time an active-going Gate edge occurs. Retriggering causes the counter value to be saved in the Hold register and the Load register contents to be transferred into the counter.

When No Gating is specified, the definition of CM7 changes. In this case, when CM7 = 0, the Gate input has no effect on the counting; when CM7 = 1, the Gate input specifies the source (selecting either the Load or Hold register) used to reload the counter when TC occurs. Figure 14 shows the various available control combinations for these interrelated bits.

Count Source Selection

Counter Mode bits CM8 through CM12 specify the source used as input to the counter and the active edge that is counted. Bit CM12 controls the polarity for all the sources; logic zero counts rising edges and logic one counts falling edges. Bits CM8 through CM11 select 1 of 16 counting sources to route to the counter input. Five of the available inputs are internal frequencies derived from the internal oscillator (see Figure 13 for frequency assignments). Ten of the available inputs are interface pins; five are labeled SRC and five are labeled GATE.

The 16th available input is the TC output from the adjacent lower-numbered counter. (The Counter 5 TC wraps around to the Counter 1 input.) This option allows internal concatenating that permits very long counts to be accumulated. Since all five counters may be concatenated, it is possible to configure a counter that is 80-bits long on one Am9513A chip. When TCN-1 is the source, the count ripples between the connected counters. External connections can also be made, and can use the toggle bit for even longer counts. This is easily accomplished by selecting a TC Toggled output mode and wiring OUTN to one of the SRC inputs.

Gating Control

Counter Mode bits CM15, CM14, CM13 specify the hardware gating options. When "no gating" is selected (000), the counter will proceed unconditionally as long as it is armed. For any other gating mode, the count process is conditioned by the specified gating configuration.

For a code of 100 in this field, counting can proceed only when the pin labeled GATEN associated with Counter N is at a logic high level. When it goes LOW, counting is simply suspended until the Gate goes HIGH again. A code of 101 performs the same function with an opposite active polarity. Codes 010 and 011 offer the same function as 100, but specify alternate input pins as Gating Sources. This allows any of three interface pins to be used as gates for a given counter. On Counter 4, for example, pin 34, pin 35 or pin 36 may be used to perform the gating function. This also allows a single Gate pin to simultaneously control up to three counters. Counters 1 and 5 are considered adjacent when using TCN-1 (001), Gate N + 1 (010) and Gate N – 1 (011) controls.

For codes of 110 or 111 in this field, counting proceeds after the specified active Gate edge until one or two TC events occur. Within this interval, the Gate input is ignored, except for the retriggering option. When repetition is selected, a cycle will be repeated as soon as another Gate edge occurs. With repetition selected, any Gate edge applied after TC goes active will start a new count cycle. Edge gating is useful when implementing a digital single-shot since the gate can serve as a convenient firing trigger.

A 001 code in this field selects the TC (not TOGGLE) output from the adjacent lower-numbered counter as the gate. This is useful for synchronous counting when adjacent counters are concatenated.

COMMAND DESCRIPTIONS

The command set for the Am9513A allows the host processor to customize and manage the operating modes and features for particular applications, to initialize and update both the internal data and control information, and to manipulate operating bits during operation. Commands are entered directly into the 8-bit Command register by writing into the Control port (see Figure 6).

All available commands are described in the following text. Figure 19 summarizes the command codes and includes a brief description of each function. Figure 20 shows all the unused code combinations; unused codes should not be entered into the Command register since undefined activities may occur.

Six of the command types are used for direct software control of the counting process and they each contain a 5-bit S field. In a linear-select fashion, each bit in the S field corresponds to one of five general counters (S1 = Counter 1, S2 = Counter 2, etc.). When an S bit is a one, the specified operation is performed on the counter so designated; when an S bit is a zero, no operation occurs for the corresponding counter. This type of command format has three basic advantages. It saves host software by allowing any combination of counters to be acted on by a single command. It allows simultaneous action on multiple counters where synchronization of commands is important. It allows counter-specific service routines to control individual counters without needing to be aware of the operating context of other counters.

Three of the commands use a 3-bit binary code (N4, N2, N1) to identify the affected counter (a 001 programs counter 1, etc.). Unlike the previously mentioned commands, these commands allow you to program only one counter at a time.

Command Code								Command Description
C7	C6	C5	C4	C3	C2	C1	C0	
0	0	0	E2	E1	G4	G2	G1	Load Data Pointer register with contents of E and G fields. (G ≠ 000, G ≠ 110)
0	0	1	S5	S4	S3	S2	S1	Arm counting for all selected counters
0	1	0	S5	S4	S3	S2	S1	Load contents of specified source into all selected counters
0	1	1	S5	S4	S3	S2	S1	Load and Arm all selected counters*
1	0	0	S5	S4	S3	S2	S1	Disarm and Save all selected counters
1	0	1	S5	S4	S3	S2	S1	Save all selected counters in Hold register
1	1	0	S5	S4	S3	S2	S1	Disarm all selected counters
1	1	1	0	1	N4	N2	N1	Set Toggle out (HIGH) for counter N (001 ≤ N ≤ 101)
1	1	1	0	0	N4	N2	N1	Clear Toggle out (LOW) for counter N (001 ≤ N ≤ 101)
1	1	1	1	0	N4	N2	N1	Step counter N (001 ≤ N ≤ 101)
1	1	1	0	1	0	0	0	Set MM14 (Disable Data Pointer Sequencing)
1	1	1	0	1	1	1	0	Set MM12 (Gate off FOUT)
1	1	1	0	1	1	1	1	Set MM13 (Enter 16-bit bus mode)
1	1	1	0	0	0	0	0	Clear MM14 (Enable Data Pointer Sequencing)
1	1	1	0	0	1	1	0	Clear MM12 (Gate on FOUT)
1	1	1	0	0	1	1	1	Clear MM13 (Enter 8-bit bus mode)
1	1	1	1	1	0	0	0	Enable Prefetch for Write operations (Am9513'A' only)
1	1	1	1	1	0	0	1	Disable Prefetch for Write operations (Am9513'A' only)
1	1	1	1	1	1	1	1	Master reset

*Not to be used for asynchronous operations.

Figure 19. Am9513A Command Summary

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	0	0	0	0
1	1	1	1	0	1	1	0
1	1	1	1	0	1	1	1
0	0	0	X	X	1	1	0
0	0	0	X	X	0	0	0
*1	1	1	1	1	X	X	X

*Unused except when XXX = 111, 001 or 000.

Figure 20. Am9513A Unused Command Codes

Arm Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	0	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be enabled for counting. A counter must be armed before counting can commerce. Once armed, the counting process may be further enabled or disabled using the hardware gating facilities. This command can only arm or do nothing for a given counter; a zero in the S field does not disarm the counter.

ARM and DISARM commands can be used to gate counter operation on and off under software control. DISARM commands entered while a counter is in the TC state will not take effect until the counter leaves TC. This ensures that the counter never latches up in a TC state. (The counter may leave the TC state because of application of a count source edge, execution of a LOAD or LOAD AND ARM command, or execution of a STEP command.)

In modes which alternate reload sources (Modes G – L), the ARMing operation is used as a reset for the logic which determines which reload source to use on the upcoming TC. Following each ARM or LOAD AND ARM command, a counter in one of these modes will reload from the Hold register on the first TC and alternate reload sources thereafter (reload from the Load register on the second TC, the Hold register on the third, etc.).

Load Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	1	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified in the S field, will be loaded with previously entered values. The source of information for each counter will be either the associated Load register or the associated Hold register, as determined by the operating configuration in the Mode register. The Load/Hold contents are not changed. This command will cause a transfer independent of any current operating configuration for the counter. It will often be used as a software retrigger or as counter initialization prior to active hardware gating.

If a LOAD or LOAD AND ARM command is executed during the cycle preceding TC, the counter will go immediately to TC. This occurs because the LOAD operation is performed by generating a pseudo-count pulse internal to the Am9513A, and the Am9513A is expecting to go into TC on the next count pulse. The reload source used to reload the counter will be the same as that which would have been used if the TC were generated by a source edge rather than by the LOAD operation.

Execution of a LOAD or LOAD AND ARM command while the counter is in TC will cause the TC to end. For Armed counters in all modes except S or V, the LOAD source used will be that to be used for the upcoming TC. (The LOADing operation will not alter the selection of reload source for the upcoming TC.) For Disarmed counters in modes except S or V, the reload sources used will be the LOAD register. For modes S or V, the reload source will be selected by the GATE input, regardless of whether the counter is Armed or Disarmed.

Special considerations apply when modes with alternating reload sources are used (Modes G – L). If a LOAD command drives the counter to TC in these modes, the reload source for the next TC will be from the opposite reload location. In other words, the LOAD-generated TC will cause the reload sources to alternate just as a TC generated by a source edge would. Note that if a second LOAD command is issued during the LOAD-generated TC (or during any other TC, for that matter), the second LOAD command will terminate the TC and cause a reload from the source designated for use with the next TC. The second LOAD will not alter the reload source for the next TC since the second LOAD does not generate a TC; reload sources alternate on TCs only, not on LOAD commands.

Load and Arm Counters*

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	1	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified in the S field, will be first loaded and then armed. This command is equivalent to issuing a LOAD command and then an ARM command.

A LOAD AND ARM command which drives a counter to TC generates the same sequence of operations as execution of a LOAD command and then an ARM command. In modes which disarm on TC (Modes A – C and N – O, and Modes G – I and S if the current TC is the second in the cycle), the ARM part of the LOAD AND ARM command will re-enable counting for another cycle. In modes which alternate reload sources (Modes G – L), the ARMing operation will cause the next TC to reload from the HOLD register, irrespective of which reload source the current TC used.

*This command should not be used during asynchronous operations.

Disarm Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be disabled from counting. A disarmed counter will cease all counting independent of other conditions. The only exception to this is that a counter in the TC state will always count once, in order to leave TC, before DISARMing. This count may be generated by a source edge, by a LOAD or LOAD AND ARM command (the LOAD AND ARM command will negate the DISARM command) or by a STEP command. A disarmed counter may be updated using the LOAD command and may be read using the SAVE command. A count process may be resumed using an ARM command. See the ARM command description for further details.

Save Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	0	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by S field, will have their contents transferred into their associated Hold register. The transfer takes place without interfering with any counting that may be underway. This command will overwrite any previous Hold register contents. The SAVE command is designed to allow an accumulated count to be preserved so that it can be read by the host CPU at some later time.

Disarm and Save Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	0	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be disarmed, and the contents of the counter will be transferred into the associated Hold registers. This command is identical to issuing a DISARM command followed by a SAVE command.

Set TC Toggle Output

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	N4	N2	N1

(001 ≤ N ≤ 101)

Description: The initial output level for TC Toggle mode is set (HIGH) for counter N selected by N4, N2, N1 = 001 (Counter 1) thru 101 (Counter 5) respectively. This command conditions the TC Toggle flip-flop (see Figure 17) but does not appear at the counter output unless TC Toggle mode (CM2, CM1, CM0 = 010) is selected.

Clear TC Toggle Output

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	N4	N2	N1

(001 ≤ N ≤ 101)

Description: The initial output level for TC Toggle mode is Cleared (LOW) for counter N selected by N4, N2, N1 = 001 (Counter 1) thru 101 (Counter 5) respectively. This command conditions the TC Toggle flip-flop (see Figure 17) but does not appear at the counter output unless TC Toggle mode (CM2, CM1, CM0 = 010) is selected.

Step Counter

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	0	N4	N2	N1

(001 ≤ N ≤ 101)

Description: Counter N is incremented or decremented by one, depending on its operating configuration. If the Counter Mode register associated with the selected counter has its CM3 bit cleared to zero, this command will cause the counter to decrement by one. If CM3 is set to a logic high, this command will increment the counter by one. The STEP command will take effect even on a disarmed counter.

Load Data Pointer Register

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	0	0	E2	E1	G4	G2	G1

(G4, G2, G1 ≠ 000, ≠ 110)

Description: Bits in the E and G fields will be transferred into the corresponding Element and Group fields of the Data Pointer register as shown in Figure 7. The Byte Pointer bit in the Data Pointer register is set. Transfers into the Data Pointer only occur for G field values of 001, 010, 011, 100, 101 and 111. Values of 000 and 110 for G should not be used. See the "Setting the Data Pointer Register" section of this document for additional details.

Disable Data Pointer Sequencing

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	0	0	0

Description: This command sets Master Mode bit 14 without affecting other bits in the Master Mode register. MM14 controls the automatic sequencing of the Data Pointer register. Disabling the sequencing allows repetitive host processor access to a given internal location without repetitive updating of the Data Pointer. MM14 may also be controlled by loading a full word into the Master Mode register.

Enable Data Pointer Sequencing

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	0	0	0

Description: This command clears Master Mode bit 14 without affecting other bits in the Master Mode register. MM14 controls the automatic sequencing of the Data Pointer register. Enabling the sequencing allows sequential host processor access to several internal locations without repetitive updating of the Data Pointer. MM14 may also be controlled by loading a full word into the Master Mode register. See the "Data Pointer Register" section of this document for additional information on Data Pointer sequencing

Enable 16-Bit Data Bus

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	1	1	1

Description: This command sets Master Mode bit 13 without affecting other bits in the Master Mode register. MM13 controls the multiplexer in the data bus buffer. When MM13 is set, no multiplexing takes place and all 16 external data bus lines are used to transfer information into and out of the STC. MM13 may also be controlled by loading the full Master Mode register in parallel.

Enable 8-Bit Data Bus

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	1	1	1

Description: This command clears Master Mode bit 13 without affecting other bits in the Master Mode register. MM13 controls the multiplexer in the data bus buffer. When MM13 is cleared, the multiplexer is enabled and 16-bit internal information is transferred eight bits at a time to the eight low-order external data bus lines. MM13 may also be controlled by loading the full Master Mode register in parallel.

Gate Off FOUT

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	1	1	0

Description: This command sets Master Mode bit 12 without affecting other bits in the Master Mode register. MM12 controls the output state of the FOUT signal. When gated off, the FOUT line will exhibit a low-impedance to ground. MM12 may also be controlled by loading the full Master Mode register in parallel.

Gate On FOUT

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	1	1	0

Description: This command clears Master Mode bit 12 without affecting other bits in the Master Mode register. MM12 controls the output status of the FOUT signal. When MM12 is cleared, FOUT will become active and will drive out the selected and divided FOUT signal. MM12 may also be controlled by loading the full Master Mode register in parallel. When FOUT is gated on or off, a transient pulse may be generated on the FOUT signal.

Disable Prefetch for Write Operations

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	0	0	1

Description: This command disables the prefetch circuitry during Write operations (it does not affect Read operations). This reduces the write recovery time and allows the user to use block move instructions for initialization of the Am9513A registers. Once prefetch is disabled for writing, an Enable Prefetch for Write or a Reset command is necessary to re-enable the prefetch circuitry for writing.
Note: This command is only available in Am9513A devices; it is an illegal command in the "non-A" Am9513 device.

Enable Prefetch for Write Operations

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	0	0	0

Description: This command re-enables the prefetch circuitry for Write operations. It is used only to terminate the Disable Prefetch Command.
Note: This command is only available in Am9513A devices; it is an illegal command in the "non-A" Am9513 device.

Master Reset

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	1	1	1

Description: The Master Reset command duplicates the action of the power-on reset circuitry. It disarms all counters, enters 0000 in the Master Mode, Load and Hold registers and enters 0B00 (hex) in the Counter Mode registers.

Following either a power-up or software reset, the LOAD command should be applied to all the counters to clear any that may be in a TC state. The Data Pointer register should also be set to a legal value, since reset does not initialize it. A complete reset operation follows.

1. Using the procedure given in the "Command Initiation" section of this document, enter the FF (hex) command to perform a software reset.
2. Using the "Command Initiation" procedure, enter the LOAD command for all counters, opcode 5F (hex).
3. Using the procedure given in the "Setting the Data Pointer Register" section of this document, set the Data Pointer to a valid code. The legal Data Pointer codes are given in Figure 8.

The Master Mode, Counter Mode, Load and Hold registers can now be initialized to the desired values.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
VCC with Respect to VSS ..... −0.5 V to +7.0 V
All Signal Voltages
with Respect to VSS ..... −0.5 V to +7.0 V
Power Dissipation (Package Limitation) ..... 1.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

Military (M) Devices
Temperature (T_C) ..... −55 to +125°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

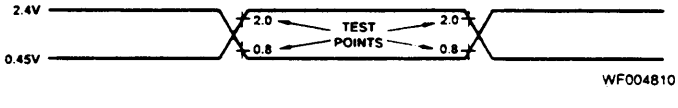
Operating ranges define those limits between which the functionality of the device is guaranteed.

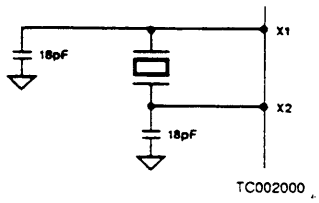
DC CHARACTERISTICS over operating ranges unless otherwise specified.

Parameters	Description		Test Conditions	Min	Max		Units
VIL	Input Low Voltage	All Inputs Except X2		VSS − 0.5	0.8		Volts
		X2 Input		VSS − 0.5	0.8		
VIH	Input High Voltage	All Input Except X2		2.2 V	VCC		Volts
		X2 Input		3.8	VCC		
VITH	Input Hysteresis (SRC and GATE Inputs Only)			0.2			Volts
VOL	Output Low Voltage		IOL = 3.2 mA		0.4		Volts
VOH	Output High Voltage		IOH = −200 µA	2.4			Volts
IIX	Input Load Current (Except X2)		VSS ≤ VIN ≤ VCC		±10		µA
IIX	Input Load Current X2		VSS ≤ VIN ≤ VCC		±100		µA
IOZ	Output Leakage Current (Except X1)		VSS + 0.4 ≤ VOUT ≤ VCC High-Impedance State		±25		µA
ICC	VCC Supply Current (Steady State)				255	275	mA
CIN	Input Capacitance		f = 1 MHz, T_A = +25°C, All pins not under test at 0 V		10*	20*	pF
COUT	Output Capacitance				15*	20*	
CIO	IN/OUT Capacitance				20*	20*	

* Guaranteed by design.

SWITCHING TEST INPUT/OUTPUT WAVEFORMS





Crystal is fundamental mode parallel resonant 32 pF load capacitance less than 100 Ω ESR C_o less than 100 pF

The second and fourth letters designate the reference states of the signals named in the first and third letters respectively, using the following abbreviations.

H = HIGH
L = LOW
V = VALID
X = Unknown or Don't care
Z = High-Impedance

2. Any input transition that occurs before this minimum setup requirement will be reflected in the contents read from the status register.
3. Any input transition that occurs before this minimum setup requirement will act on the counter before the execution of the operation initiated by the write and the counter may be off by one count.
4. Any input transition that occurs after this minimum hold time is guaranteed to not influence the contents read from the status register on the current read operation.
5. Any input transition that occurs after this minimum hold time is guaranteed to be seen by the counter as occurring after the action initiated by the write operation and the counter may be off by one count.
6. This parameter applies to cases where the write operation causes a change in the output bit.
7. The enabled count source is one of F1 – F5, TCN-1 SRC1 – SRC5 or GATE1 – GATE 5, as selected in the applicable Counter Mode register. The timing diagram assumes the counter counts on rising source edges. The timing specifications are the same for falling-edge counting.
8. This parameter applies to edge gating (CM15 – CM13 = 110 or 111) and gating when both CM7 = 1 and CM15 – CM13 ≠ 000. This parameter represents the minimum GATE pulse width needed to ensure that the pulse initiates counting or counter reloading.
9. This parameter applies to both edge and level gating (CM15 – CM13 = 001 through 111 and CM7 = 0). This parameter represents the minimum setup or hold times to ensure that the Gate input is seen at the intended level on the active source edge and the counter may be off by one count.
10. This parameter assumes that the GATENA input is unused (16-bit bus mode) or is tied high. In cases where the GATENA input is used, this timing specification must be met by both the GATE and GATENA inputs.
11. Signals F1 – F5 cannot be directly monitored by the user. The phase difference between these signals will manifest itself by causing counters using two different F signals to count at different times on nominally simultaneous transitions in the F signals. F1 = X2.
12. This timing specification assumes that $\overline{CS}$ is active whenever $\overline{RD}$ or $\overline{WR}$ are active. $\overline{CS}$ may be held active indefinitely.
13. This parameter assumes X2 is driven from an external gate with a square wave.
14. This parameter assumes that the write operation is to the command register.
15. This timing specification applies to single-action commands only (e.g., LOAD, ARM, SAVE, etc.). For double-action commands such as LOAD AND ARM and DISARM AND SAVE, TWHEH minimum = 700 ns.
16. In short data write mode, TWHRL and TWHWL minimum = 1000 ns.
17. This parameter applies to the hardware retrigger/save modes N, O, Q, R, and X (CM7 = 1 and CM15 – CM13 < > 000). This parameter ensures that the gating pulse initiates a hardware retrigger/save operation.
18. This parameter applies to hardware load source select modes S and V (CM7 = 1 and CM15 – CM13 = 000). This parameter represents the minimum hold time to ensure that the GATE input selects the correct load source on the active source edge.

SWITCHING CHARACTERISTICS over **COMMERCIAL** operating ranges unless otherwise specified (Note 1)

Parameters	Description		Figure	Am9513A Min	Am9513A Max	Unit
TAVRL	C/$\overline{D}$ Valid to Read Low		21	25		ns
TAVWH	C/$\overline{D}$ Valid to Write High		21	170		ns
TCHCH	X2 High to X2 High (X2 Period) (Note 13)		22	145		ns
TCHCL	X2 High to X2 Low (X2 High Pulse Width) (Note 13)		22	70		ns
TCLCH	X2 Low to X2 High (X2 Low Pulse Width) (Note 13)		22	70		ns
TDVWH	Data In Valid to Write High		21	80		ns
TEHEH	Count Source High to Count Source High (Source Cycle Time) (Note 7)		22	145		ns
TEHEL TELEH	Count Source Pulse Duration (Note 7)		22	70		ns
TEHFV	Count Source High to FOUT Valid (Note 7)		22		500	ns
TEHGV	Count Source High to Gate Valid (Level Gating Hold Time) (Notes 7, 9, 10)		22	10		ns
TEHRL	Count Source High to Read Low (Set-up Time) (Notes 2, 7)		21	190		ns
TEHWH	Count Source High to Write High (Set-up Time) (Notes 3, 7)		21	-100		ns
TEHYV	Count Source High to Out Valid (Note 7)	TC Output	22		300	ns
		Immediate or Delayed Toggle Output	22		300	
		Comparator Output	22		350	
TFN	FN High to FN + 1 Valid (Note 11)		22		75	ns
TGVEH	Gate Valid to Count Source High (Level Gating Set-up Time) (Notes 7, 9, 10)		22	100		ns
TGVGV	Gate Valid to Gate Valid (Gate Pulse Duration) (Notes 8, 10)		22	145		ns
TGVWH	Gate Valid to Write High (Notes 3, 10)		21	-100		ns
TRHAX	Read High to C/$\overline{D}$ Don't Care		21	0		ns
TRHEH	Read High to Count Source High (Notes 4, 7)		21	0		ns
TRHQX	Read High to Data Out Invalid		21	10		ns
TRHQZ	Read High to Data Out at High-Impedance (Data Bus Release Time)		21		85	ns
TRHRL	Read High to Read Low (Read Recovery Time)		21	1000		ns
TRHSH	Read High to $\overline{CS}$ High (Note 12)		21	0		ns
TRHWL	Read High to Write Low (Read Recovery Time)		21	1000		ns
TRLQV	Read Low to Data Out Valid		21		110	ns
TRLQX	Read Low to Data Bus Driven (Data Bus Drive Time)		21	20		ns
TRLRH	Read Low to Read High (Read Pulse Duration) (Note 12)		21	160		ns
TSLRL	$\overline{CS}$ Low to Read Low (Note 12)		21	20		ns
TSLWH	$\overline{CS}$ Low to Write High (Note 12)		21	170		ns
TWHAX	Write High to C/$\overline{D}$ Don't Care		21	20		ns
TWHDX	Write High to Data In Don't Care		21	20		ns
TWHEH	Write High to Count Source High (Notes 5, 7, 14, 15)		21	550		ns
TWHGV	Write High to Gate Valid (Notes 5, 10, 14)		21	475		ns
TWHRL	Write High to Read Low (Write Recovery Time) (Note 16)		21	1500*		ns
TWHSH	Write High to $\overline{CS}$ High (Note 12)		21	20		ns
TWHWL	Write High to Write Low (Write Recovery Time) (Note 16)		21	1500*		ns
TWHYV	Write High to Out Valid (Notes 6, 14)		21		650	ns
TWLWH	Write Low to Write High (Write Pulse Duration) (Note 12)		21	150		ns
TGVEH2	Gate Valid to Count Source High (Special Gate) (Notes 10, 13, 17)		22	200		ns
TEHGV2	Count Source High to Gate Valid (Special Gate) (Notes 10, 13, 18)		22	80		ns

Notes:

1. Abbreviations used for the switching parameter symbols are given as the letter T followed by four or five characters. The first and third characters represent the signal names on which the measurements start and end. Signal abbreviations used are:

   A (Address) = C/$\overline{D}$
   C (Clock) = X2
   D (Data In) = DB0-DB15
   E (Enabled counter source input) = SRC1 - SRC5, GATE1 - GATE5, F1 - F5,TCN-1
   F = FOUT
   G (Counter gate input) = GATE1 - GATE5, TCN-1
   Q (Data Out) = DB0 - DB15
   R (Read) = $\overline{RD}$
   S (Chip Select) = $\overline{CS}$
   W (Write) = $\overline{WR}$
   Y (Output) = OUT1 - OUT5

SWITCHING CHARACTERISTICS over **MILITARY** operating range (for SMD/DESC and APL Products, Group A, Subgroups 9, 10, 11 are tested unless otherwise noted)

Parameter Symbol	Description		Am9513A Min.	Am9513A Max.	Unit
TAVRL	C/$\overline{D}$ Valid to Read Low		25		ns
TAVWH	C/$\overline{D}$ Valid to Write High		170		ns
TCHCH	X2 High to X2 High (X2 Period) (Note 13)		145		ns
TCHCL	X2 High to X2 Low (X2 High Pulse Width) (Note 13)		70		ns
TCLCH	X2 Low to X2 High (X2 Low Pulse Width) (Note 13)		70		ns
TDVWH	Data In Valid to Write High		80		ns
TEHEH	Count Source High to Count Source High (Source Cycle Time) (Note 7)		145		ns
TEHEL TELEH	Count Source Pulse Duration (Note 7)		70		ns
TEHFV	Count Source High to FOUT Valid (Note 7)			500	ns
TEHGV	Count Source High to Gate Valid (Level Gating Hold Time) (Notes 7, 9, 10)		10		ns
TEHRL	Count Source High to Read Low (Set-up Time) (Notes 2, 7)		190		ns
TEHWH	Count Source High to Write High (Set-up Time) (Notes 3, 7)		-100		ns
TEHYV	Count Source High to Out Valid (Note 7)	TC Output		300	ns
		Immediate or Delayed Toggle Output		300	
		Comparator Output		350	
TFN	FN High to FN + 1 Valid (Note 11)			75	ns
TGVEH	Gate Valid to Count Source High (Level Gating Set-up Time) (Notes 7, 9, 10)		100		ns
TGVGV	Gate Valid to Gate Valid (Gate Pulse Duration) (Notes 8, 10)		145		ns
TGVWH	Gate Valid to Write High (Notes 3, 10)		-100		ns
TRHAX	Read High to C/$\overline{D}$ Don't Care		0		ns
TRHEH	Read High to Count Source High (Notes 4, 7)		0		ns
TRHQX	Read High to Data Out Invalid		10		ns
TRHQZ	Read High to Data Out at High-Impedance (Data Bus Release Time)			85	ns
TRHRL	Read High to Read Low (Read Recovery Time)		1000		ns
TRHSH	Read High to $\overline{CS}$ High (Note 12)		0		ns
TRHWL	Read High to Write Low (Read Recovery Time)		1000		ns
TRLQV	Read Low to Data Out Valid			110	ns
TRLQX	Read Low to Data Bus Driven (Data Bus Drive Time)		20		ns
TRLRH	Read Low to Read High (Read Pulse Duration) (Note 12)		160		ns
TSLRL	$\overline{CS}$ Low to Read Low (Note 12)		20		ns
TSLWH	$\overline{CS}$ Low to Write High (Note 12)		170		ns
TWHAX	Write High to C/$\overline{D}$ Don't Care		20		ns
TWHDX	Write High to Data In Don't Care		20		ns
TWHEH	Write High to Count Source High (Notes 5, 7, 14, 15)		550		ns
TWHGV	Write High to Gate Valid (Notes 5, 10, 14)		475		ns
TWHRL	Write High to Read Low (Write Recovery Time) (Note 16)		1500		ns
TWHSH	Write High to $\overline{CS}$ High (Note 12)		20		ns
TWHWL	Write High to Write Low (Write Recovery Time) (Note 16)		1500		ns
TWHYV	Write High to Out Valid (Notes 6, 14)			650	ns
TWLWH	Write Low to Write High (Write Pulse Duration) (Note 12)		150		ns
TGVEH2	Gate Valid to Count Source High (Special Gate) (Notes 10, 13, 17)		200		ns
TEHGV2	Count Source High to Gate Valid (Special Gate) (Notes 10, 13, 18)		80		ns

Notes:

1. Abbreviations used for the switching parameter symbols are given as the letter T followed by four or five characters. The first and third characters represent the signal names on which the measurements start and end. Signal abbreviations used are:

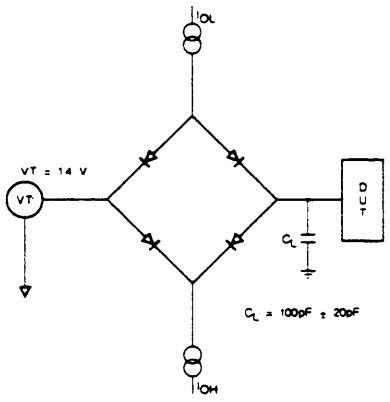
   A (Address) = C/$\overline{D}$
   C (Clock) = X2
   D (Data In) = DB0 - DB15
   E (Enabled counter source input) = SRC1 - SRC5, GATE1 - GATE5, F1 - F5,TCN-1
   F = FOUT
   G (Counter gate input) = GATE1 - GATE5, TCN-1
   Q (Data Out) = DB0 - DB15
   R (Read) = $\overline{RD}$
   S (Chip Select) = $\overline{CS}$
   W (Write) = $\overline{WR}$
   Y (Output) = OUT1 - OUT5

The second and fourth letters designate the reference states of the signals named in the first and third letters respectively, using the following abbreviations.

H = HIGH
L = LOW
V = VALID
X = Unknown or Don't care
Z = High-Impedance

2. Any input transition that occurs before this minimum setup requirement will be reflected in the contents read from the status register.
3. Any input transition that occurs before this minimum setup requirement will act on the counter before the execution of the operation initiated by the write and the counter may be off by one count.
4. Any input transition that occurs after this minimum hold time is guaranteed to not influence the contents read from the status register on the current read operation.
5. Any input transition that occurs after this minimum hold time is guaranteed to be seen by the counter as occurring after the action initiated by the write operation and the counter may be off by one count.
6. This parameter applies to cases where the write operation causes a change in the output bit.
7. The enabled count source is one of F1 – F5, TCN-1 SRC1 – SRC5 or GATE1 – GATE 5, as selected in the applicable Counter Mode register. The timing diagram assumes the counter counts on rising source edges. The timing specifications are the same for falling-edge counting.
8. This parameter applies to edge gating (CM15 – CM13 = 110 or 111) and gating when both CM7 = 1 and CM15 – CM13 ≠ 000. This parameter represents the minimum GATE pulse width needed to ensure that the pulse initiates counting or counter reloading.
9. This parameter applies to both edge and level gating (CM15 – CM13 = 001 through 111 and CM7 = 0). This parameter represents the minimum setup or hold times to ensure that the Gate input is seen at the intended level on the active source edge and the counter may be off by one count.
10. This parameter assumes that the GATENA input is unused (16-bit bus mode) or is tied high. In cases where the GATENA input is used, this timing specification must be met by both the GATE and GATENA inputs.
11. Signals F1 – F5 cannot be directly monitored by the user. The phase difference between these signals will manifest itself by causing counters using two different F signals to count at different times on nominally simultaneous transitions in the F signals. F1 = X2.
12. This timing specification assumes that $\overline{CS}$ is active whenever $\overline{RD}$ or $\overline{WR}$ are active. $\overline{CS}$ may be held active indefinitely.
13. This parameter assumes X2 is driven from an external gate with a square wave.
14. This parameter assumes that the write operation is to the command register.
15. This timing specification applies to single-action commands only (e.g., LOAD, ARM, SAVE, etc.). For double-action commands such as LOAD AND ARM and DISARM AND SAVE, TWHEH minimum = 700 ns.
16. In short data write mode, TWHRL and TWHWL minimum = 1000 ns.
17. This parameter applies to the hardware retrigger/save modes N, O, Q, R, and X (CM7 = 1 and CM15 – CM13 < > 000). This parameter ensures that the gating pulse initiates a hardware retrigger/save operation.
18. This parameter applies to hardware load source select modes S and V (CM7 = 1 and CM15 – CM13 = 000). This parameter represents the minimum hold time to ensure that the GATE input selects the correct load source on the active source edge.

SWITCHING TEST CIRCUIT



TC003853

This test circuit is the dynamic load of a Teradyne J941.

SWITCHING WAVEFORMS

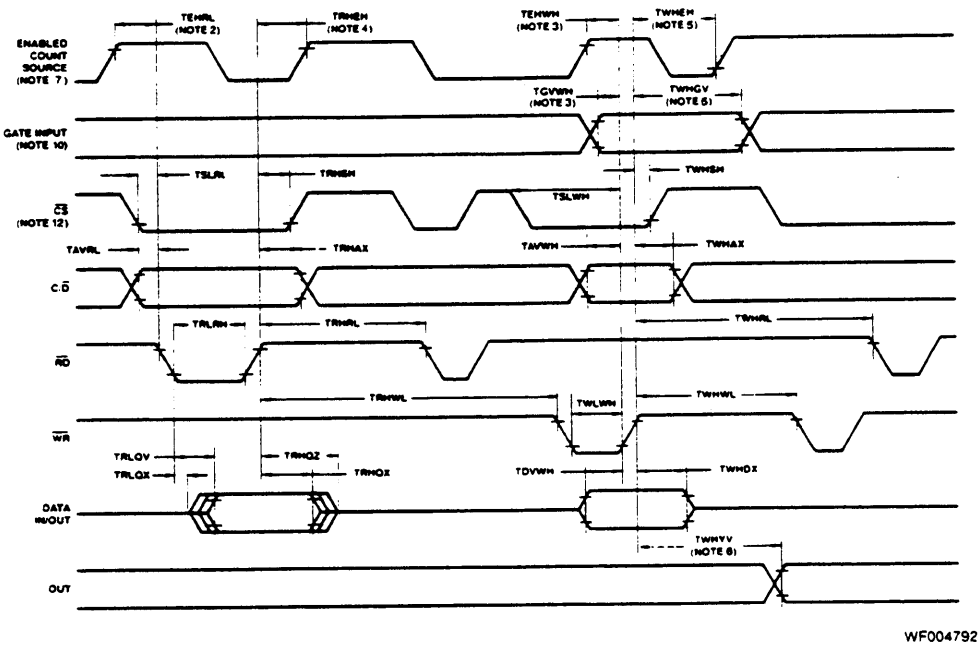


Figure 21. Bus Transfer Switching Waveforms

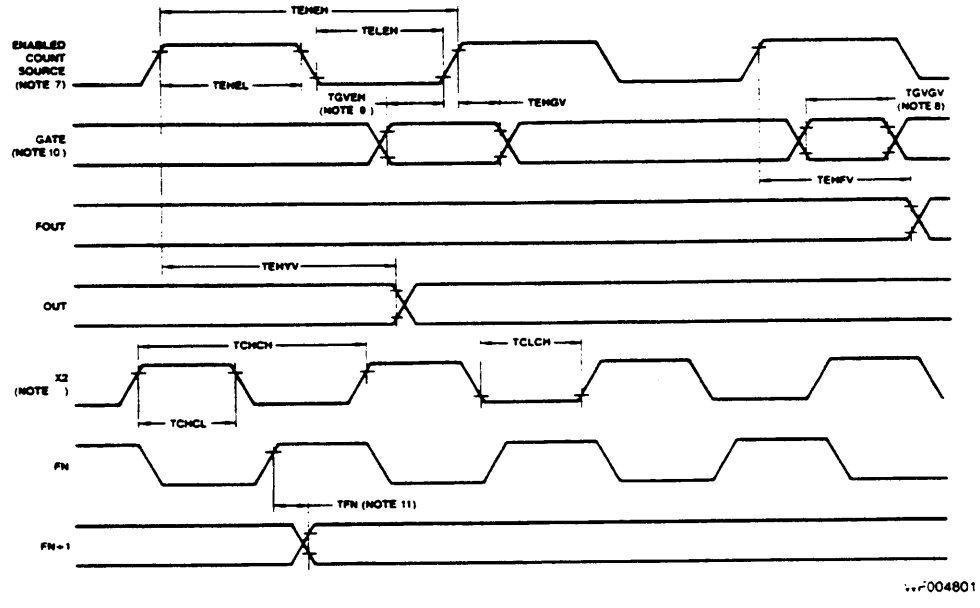


Figure 22. Counter Switching Waveforms

APPENDIX A

Design Hints

1) When a crystal is not being used, X1 and X2 should be connected as shown for TTL input (Figure A1) and no input (Figure A2).

2) Recommended oscillator capacitor values are 18 pF on X1 and X2.

3) Unused inputs should be tied to VCC.

4) The TC output can glitch when the counter is loaded. For this reason this output should not be connected to edge sensitive interrupts. The counter output should be set or cleared after the LOAD command.

5. The two most significant bits of the status register are not specified. They may be zero or one.

6. The mode register should not be modified when the counter is armed.

7. The LOAD and HOLD registers should not be changed during TC.

8. When using the different clocks for different counters be aware that there is a 75 ns skew between F1, F2, F3, F4 and F5.

9. The TC output will remain inactive if programmed to be in the TC TOGGLE mode and the step command is used to increment or decrement the counter. The output will go into TC if programmed to be in the active High or active Low terminal count modes. The only two ways out of TC in this case are:

   —Arming the counter and having an active source connected to it.

   —Issuing another step command.

10) Timing parameters TEHWH and TGVWH are specified as negative. The diagrams in Figure A3 show the relationship between these signals.

11) In mode X the counter will count all qualified source edges until the second (not the first) TC and then stop.

12) A TC can occur when the counters are loaded if the counter was stopped at $FFFF_H$ or 9999_{10} in the count up mode or at count 0001 when counting down. This is because an internal TC is generated which forces TC to be generated on the next count pulse.

13) In modes that alternate the reload source between the load and the hold registers (e.g., mode J), if the counter is disarmed at 0001_H for down counting or 9999_{10} for BCD up counting or $FFFF_H$ for binary up counting and rearmed, the reload source after the first TC will be the load register instead of the hold register. To avoid this, issue a software "dummy" load to the counter immediately after the disarm command.

14) In the down counting mode of the Am9513A, if a 0001 is loaded into the counter and another LOAD COUNTER command is issued, the TC of that counter will go active. If the load register contents are subsequently changed, and the counter armed, the first clock edge will cause the new load register contents to transfer into the counter and the next clock edge will decrement the counter and make it go out of TC.

15) Glitches on $\overline{CS}$ just before the $\overline{RD}$ or $\overline{WR}$ pulse may cause the part to behave incorrectly.

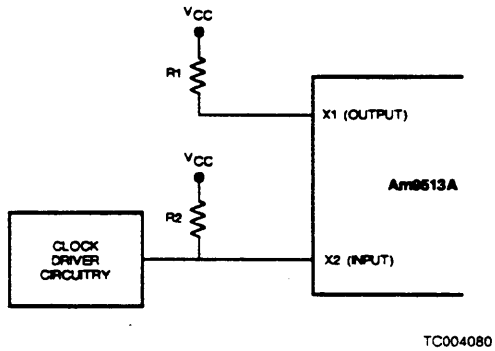
16) Timing parameters TGVEH & TEHGV must not be violated; Figure A4 shows a method.

2

Troubleshooting (Symptom: Solution)

1) Registers not being programmed correctly: Check READ or WRITE recovery time.

2) Setup and hold problems observed in synchronous systems: Try switching from positive edge to negative edge triggering.



R1 = 6.8 kΩ ±10%
R2 is a function of Driver Circuitry to meet
X2 VIH = 3.8 V
X2 VIL = 0.8 V

Figure A1. Crystal Input Configuration

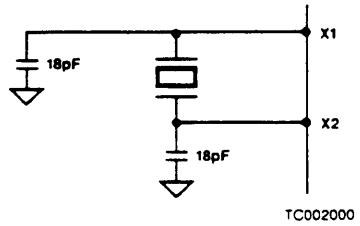


Figure A2. Crystal Input Configuration

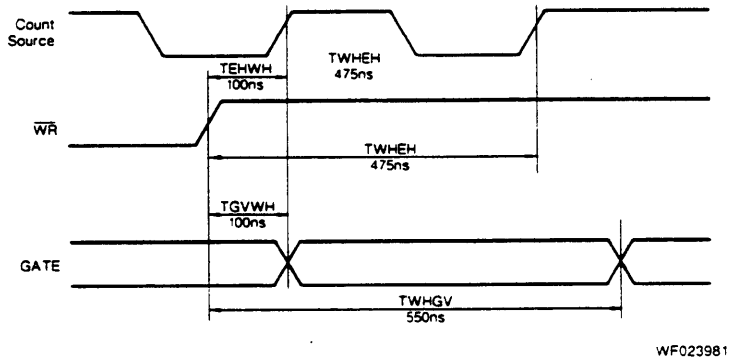


Figure A3. TEHWH/TGVWH Timing Diagram

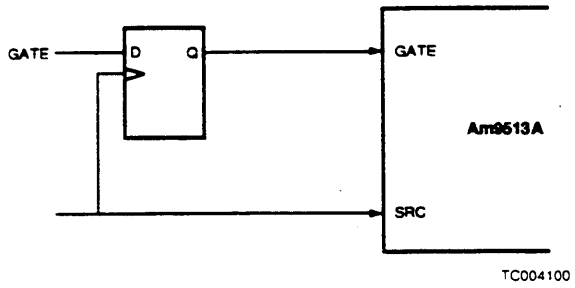


Figure A4. GATE/SRC Configuration Suggestion

Appendix D
Customer Communication

For your convenience, this appendix contains forms to help you gather the information necessary to help us solve technical problems you might have as well as a form you can use to comment on the product documentation. Filling out a copy of the *Technical Support Form* before contacting National Instruments helps us help you better and faster.

National Instruments provides comprehensive technical assistance around the world. In the U.S. and Canada, applications engineers are available Monday through Friday from 8:00 a.m. to 6:00 p.m. (central time). In other countries, contact the nearest branch office. You may fax questions to us at any time.

Corporate Headquarters
(512) 795-8248
Technical support fax: (800) 328-2203
(512) 794-5678

Branch Offices	Phone Number	Fax Number
Australia	(03) 879 9422	(03) 879 9179
Austria	(0662) 435986	(0662) 437010-19
Belgium	02/757.00.20	02/757.03.11
Denmark	45 76 26 00	45 76 71 11
Finland	(90) 527 2321	(90) 502 2930
France	(1) 48 14 24 00	(1) 48 14 24 14
Germany	089/741 31 30	089/714 60 35
Italy	02/48301892	02/48301915
Japan	(03) 3788-1921	(03) 3788-1923
Netherlands	03480-33466	03480-30673
Norway	32-848400	32-848600
Spain	(91) 640 0085	(91) 640 0533
Sweden	08-730 49 70	08-730 43 70
Switzerland	056/20 51 51	056/20 51 55
U.K.	0635 523545	0635 523154

Technical Support Form

Photocopy this form and update it each time you make changes to your software or hardware, and use the completed copy of this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

If you are using any National Instruments hardware or software products related to this problem, include the configuration forms from their user manuals. Include additional pages if necessary.

Name _____

Company _____

Address _____

_____

Fax (___)_____ Phone (___)_____

Computer brand _____ Model _____ Processor _____

Operating system _____

Speed _____ MHz RAM _____ MB Display adapter _____

Mouse _____ yes _____ no Other adapters installed _____

Hard disk capacity _____MB Brand _____

Instruments used _____

National Instruments hardware product model _____ Revision _____

Configuration _____

National Instruments software product _____ Version _____

Configuration _____

The problem is _____

_____

_____

_____

_____

List any error messages _____

_____

_____

_____

_____

The following steps will reproduce the problem _____

_____

_____

AT-MIO-16F-5 Hardware and Software Configuration Form

Record the settings and revisions of your hardware and software on the line to the right of each item. Complete a new copy of this form each time you revise your software or hardware configuration, and use this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

National Instruments Products

- AT-MIO-16F-5 Revision _____
- Base I/O Address of AT-MIO-16F-5 (Factory Setting: hex 0220) _____
- DMA Channels of AT-MIO-16F-5 (Factory Setting: 6 and 7) _____
- Interrupt Level of AT-MIO-16F-5 (Factory Setting: 10) _____
- Input Configuration (DIFF, RSE, or NRSE) _____
- NI-DAQ, LabVIEW, or LabWindows Version _____

Other Products

- Computer Make and Model _____
- Microprocessor _____
- Clock Frequency _____
- Type of Video Board Installed _____
- Operating System _____
- Operating System Version _____
- Programming Language _____
- Programming Language Version _____
- Other Boards in System _____
- Base I/O Address of Other Boards _____
- DMA Channels of Other Boards _____

Documentation Comment Form

National Instruments encourages you to comment on the documentation supplied with our products. This information helps us provide quality products to meet your needs.

Title: **AT-MIO-16F-5 User Manual**

Edition Date: **February 1994**

Part Number: **320266-01**

Please comment on the completeness, clarity, and organization of the manual.

If you find errors in the manual, please record the page numbers and describe the errors.

Thank you for your help.

Name _____

Title _____

Company _____

Address _____

_____

Phone (_____) _____

Mail to: Technical Publications
National Instruments Corporation
6504 Bridge Point Parkway, MS 53-02
Austin, TX 78730-5039

Fax to: Technical Publications
National Instruments Corporation
MS 53-02
(512) 794-5678

Glossary

Prefix	Meaning	Value
p-	pico-	10^{-12}
n-	nano-	10^{-9}
µ-	micro-	10^{-6}
m-	milli-	10^{-3}
k-	kilo-	10^{3}
M-	mega-	10^{6}
G-	giga-	10^{9}
t-	tera-	10^{12}

VDC	volts direct current
A	amperes
ADC	analog-to-digital converter
AWG	American Wire Gauge
BCD	binary-coded decimal
DAC	digital-to-analog converter
dB	decibels
DIFF	differential
DMA	direct memory access
EISA	Extended Industry Standard Architecture
FIFO	first-in-first-out
ft	feet
F	farads
HCT	high-speed CMOS TTL
hex	hexadecimal
Hz	hertz
ksamples	1,000 samples
LSB	least significant bit
M	megabytes of memory
m	meters
MSB	most significant bit
NRSE	non-referenced single-ended
Ω	ohms
ppm	parts per million
RAM	random-access memory
rms	root mean square
RSE	referenced single-ended

RTSI	Real-Time System Integration
sec	seconds
TC	terminal count
TTL	transistor-transistor logic
V	volts
VDC	volts direct current
Vrms	volts, root mean square

Index

Numbers

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X