AT-DIO-32F
User Manual

High-Speed 32-Bit Parallel Digital I/O Interface for the PC

April 1995 Edition

Part Number 320147-01

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, Mexico 95 800 010 0793,
Netherlands 03480-33466, Norway 32-84 84 00, Singapore 2265886, Spain (91) 640 0085, Sweden 08-730 49 70,
Switzerland 056/20 51 51, Taiwan 02 377 1200, U.K. 0635 523545

Limited Warranty

The AT-DIO-32F 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

Introduction to the AT-DIO-32F

The AT-DIO-32F is a high-speed, 32-bit, parallel, digital I/O interface. The AT-DIO-32F is a member of the National Instruments AT Series of PC AT I/O channel expansion boards for the IBM PC AT and compatible computers. These boards are designed for high-performance data acquisition and control for applications in laboratory testing, production testing, and industrial process monitoring and control.

This manual describes the installation, basic programming considerations, and theory of operation for the AT-DIO-32F. Example programs are provided in the C programming language.

Organization of This Manual

The manual is divided into the following chapters and appendixes:

- Chapter 1, *Introduction*, describes the AT-DIO-32F, lists the contents of your AT-DIO-32F kit, and explains how to unpack the AT-DIO-32F kit.
- Chapter 2, *Configuration and Installation*, explains the installation of the AT-DIO-32F board into your computer, signal connections to the AT-DIO-32F board, and cable wiring.
- Chapter 3, *Theory of Operation*, explains the basic operation of the AT-DIO-32F circuitry.
- Chapter 4, *Programming*, describes in detail the address and function of each of the AT-DIO-32F control and status registers. This chapter also includes important information about programming the AT-DIO-32F.
- Appendix A, *Specifications*, lists the specifications for the AT-DIO-32F.
- Appendix B, *I/O Connector and Register Descriptions*, contains a description of the AT-DIO-32F I/O connector and references to the registers of the AT-DIO-32F.
- Appendix C, *Application Notes*, contains the application notes for the AT-DIO-32F board.
- Appendix D, *Intel Data Sheet*, contains the *8254 Programmable Interval Timer* (Intel Corporation) data sheet. This counter/timer device is used on the AT-DIO-32F board.
- Appendix E, *Customer Communication*, contains forms for you to complete to facilitate communications with National Instruments concerning our products.
- The *Glossary* contains an alphabetical list and description of terms used in this manual, including abbreviations, acronyms, metric prefixes, mnemonics, symbols, and terms.
- 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 throughout this manual:

italic	Italic text denotes emphasis, a cross reference, or an introduction to a key concept.
NI-DAQ	NI-DAQ is used throughout this manual to refer to the NI-DAQ software for DOS/Windows/LabWindows unless otherwise noted
PC	PC refers to the IBM PC AT and compatible computers.

Related Documentation

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

- *IBM Personal Computer AT Technical Reference* manual

You may also want to consult the following manual if you plan to program the Intel 8254-2 Counter/Timer used on the AT-DIO-32F:

- *Intel 8254 System Timing Controller* technical manual

Customer Communication

National Instruments wants to receive your comments on our products and manuals. We are interested in the applications you develop with our products, and we want to help if you have problems with them. To make it easy for you to contact us, this manual contains comment and configuration forms for you to complete. These forms are in Appendix E, *Customer Communication*, at the end of this manual.

Chapter 1
Introduction

This chapter describes the AT-DIO-32F, lists the contents of your AT-DIO-32F kit, and explains how to unpack the AT-DIO-32F kit.

The AT-DIO-32F is a high-speed, 32-bit, parallel, digital I/O interface board for the PC. The 32 lines of digital I/O are organized into four 8-bit ports. With the various handshaking options available, the AT-DIO-32F is compatible with a wide range of peripheral devices and other computers. The AT-DIO-32F can be used for interrupt handling and DMA transfers on two DMA channels. Onboard counters can be used for pattern generation. A RTSI bus interface can transfer signals from other AT Series boards to the AT-DIO-32F.

All digital I/O is transferred through a standard, 50-pin, male connector. The pin assignments for this connector are compatible with the DEC DRV11J parallel interface and most standard 32-channel digital I/O applications.

The AT-DIO-32F can be used in a wide range of digital I/O applications. With the AT-DIO-32F, a digital pattern generator can be implemented, or the PC can be interfaced to any of the following:

- Other computers
  - Another IBM PC/XT, PC AT, or compatible computer with a National Instruments PC-DIO-24 or AT-DIO-32F
  - IBM Personal System/2 computer with a National Instruments MC-DIO-24 or MC-DIO-32F
  - Apple Macintosh II computer with a National Instruments NB-DIO-24 or NB-DIO-32F
  - DEC, UNIBUS, or Q-BUS system with a 16-bit interface
  - DEC LSI-11 microcomputer with a 32-bit DRV11J interface
  - Any other computer with an 8-bit, 16-bit, or 32-bit parallel interface
- Centronics-compatible printers and plotters
- Panel meters
- Instruments and test equipment with BCD readouts and/or controls
- Opto-isolated solid-state relays and I/O module mounting racks

With the AT-DIO-32F, the PC can serve as a digital I/O system controller for laboratory testing, production testing, and industrial process monitoring and control.

What Your Kit Should Contain

The contents of the AT-DIO-32F kit (part number 776246-01) are listed as follows.

Kit Component	Part Number
AT-DIO-32F board	180735-01
AT-DIO-32F User Manual	320147-01
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-DIO-32F 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-DIO-32F. 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-DIO-32F 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-DIO-32F 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
Signal Conditioning Accessories		
Cable Adapter Board for Signal Conditioning:		
SC-2052 and 50-conductor cable	0.5 m	776335-02
	1.0 m	776335-12
Optically-Isolated Digital Input:		
SC-2060 and 26-conductor cable	0.2 m	776336-00
	0.4 m	776336-10
Optically-Isolated Digital Output:		
SC-2061 and 26-conductor cable	0.2 m	776336-01
	0.4 m	776336-11
Electromechanical Relay Digital Control:		
SC-2062 and 26-conductor cable	0.2 m	776336-02
	0.4 m	776336-12
General-Purpose Termination Breadboard:		
SC-2072 with 50-conductor cable	0.5 m	776358-02
	1.0 m	776358-12
Digital Signal Conditioning Modules:		
SSR Series mounting rack and 1.0 m cable		
32-channel		776290-32
24-channel		776290-24
16-channel		776290-16
8-channel		776290-08
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
Standard ribbon cable	0.5 m	180524-05
	1.0 m	180524-10
Shielded ribbon cable	0.5 m*	180554-05
	1.0 m*	180554-10
Ribbon cable with edge connection at one end	0.5 m	180723-05
	1.0 m	180723-10

* The AT-DIO-32F is equipped with an EMI shield on the I/O connector that can be used to connect the shield of a shielded ribbon cable to the computer chassis. Shielded ribbon cables are necessary to meet FCC Class A Emission Limits.

Refer to the *Cabling* section in Chapter 2, *Configuration and Installation*, for additional information on cabling and connectors.

Unpacking

Your AT-DIO-32F 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 explains the installation of the AT-DIO-32F board into your computer, signal connections to the AT-DIO-32F board, and cable wiring.

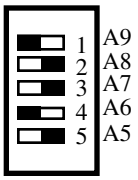
Board Configuration

The AT-DIO-32F contains three jumpers and one DIP switch to configure the AT bus interface and board clock settings. The jumpers are shown in the parts locator diagram in Figure 2-1. Jumper W3 selects the clock signal used by the board and the clock pin on the RTSI bus. Jumper W1 selects the DMA channel, and jumper W2 selects the interrupt enable lines. The DIP switch is used to set the base I/O address.

AT Bus Interface

The AT-DIO-32F is configured at the factory to use a base I/O address of hex 240, to use interrupt lines 11 and 12, to use DMA channels 5 and 6, and to disconnect the board from the RTSI clock. 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, interrupt level, or DMA channel, you need to change these settings on the AT-DIO-32F (as described in the following pages) or on the other hardware. Record your settings in the *AT-DIO-32F Hardware and Software Configuration Form* in Appendix E, *Customer Communication*.

Table 2-1. AT-DIO-32F Factory-Set Jumper and Switch Settings

Base I/O Address	Hex 240 (factory setting)	U61  The black side indicates the side that is pushed down.
DMA Channel	Bank A = Channel 5 Bank B = Channel 6 (factory setting)	**W1:** Upper-right two rows **W1:** Lower-middle two rows
Interrupt Level	Lines 11 and 12 selected (factory setting)	**W2:** Row 4 from left **W2:** Row 3 from left
RTSI Clock	Disconnect board from RTSI clock; use onboard oscillator (factory setting)	**W3:** STANDBY, BRDCLK-OSC

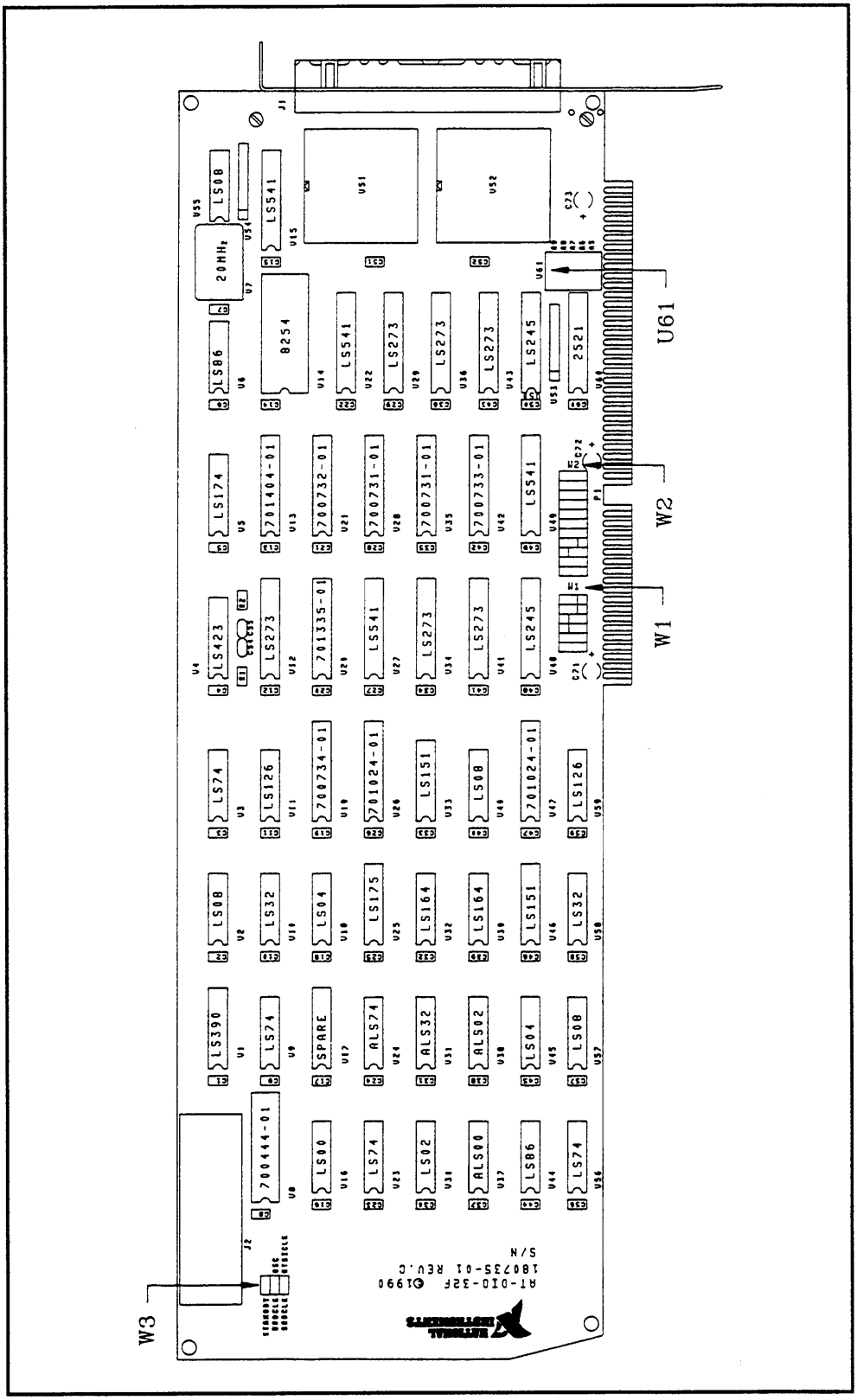


Figure 2-1. AT-DIO-32F Parts Locator Diagram

Base I/O Address Selection

The base I/O address for the AT-DIO-32F is determined by the switches at position U61 (see Figure 2-1). The switches are set at the factory for the base I/O address hex 240. This factory setting is used as the default base I/O address value by National Instruments software packages using the AT-DIO-32F. The AT-DIO-32F uses the base I/O address space hex 240 through 25F 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-DIO-32F or of the other device. If you change the AT-DIO-32F base I/O address, make a corresponding change to any software packages you use with the AT-DIO-32F. 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 U61 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 other 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.

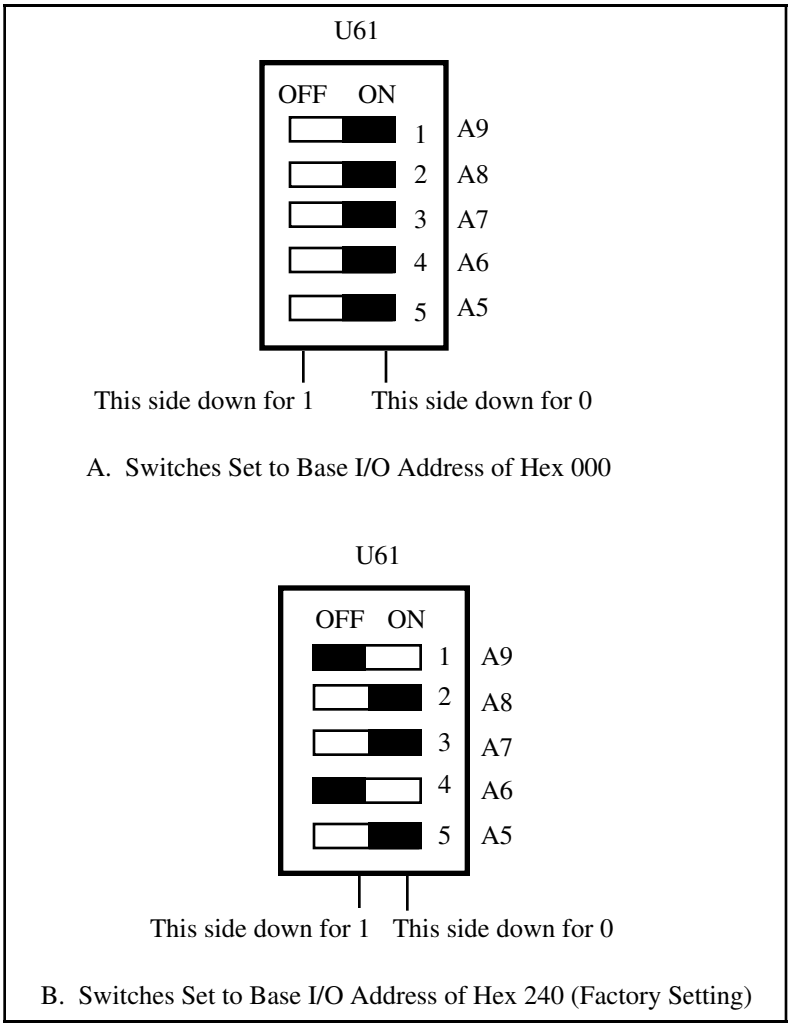


Figure 2-2. Example Base I/O Address Switch Settings

The five LSBs of the address (A4 through A0) are decoded by the AT-DIO-32F to select the appropriate AT-DIO-32F register. To change the base I/O address, remove the plastic cover on U61, press each switch to the desired position, verify that each switch is completely pressed down, and replace the plastic cover. Make a note of the new AT-DIO-32F base I/O address for use when configuring the AT-DIO-32F software (a form is included for you in Appendix E). 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 hex 000 through 0FF are reserved for system use. Base I/O address values hex 100 through 3FF are available on the I/O channel.

DMA Channel Selection

The DMA channel used by the AT-DIO-32F is selected by jumpers on W1 (see Figure 2-1). The AT-DIO-32F is set at the factory to use DMA Channels 5 and 6. These are the default DMA channels used by the AT-DIO-32F software handler. Verify that these DMA channels are not also used by equipment already installed in your computer. If any device uses DMA Channel 5 and/or Channel 6, change the DMA channel used by either the AT-DIO-32F or the other device.

The AT-DIO-32F hardware can only use Channels 5, 6, and 7 as DMA channels. Notice that these are the three available 16-bit channels on the PC I/O channel. The AT-DIO-32F *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-DIO-32F

DMA Channel	DMA Acknowledge	DMA Request
5	DACK5	DRQ5
6	DACK6	DRQ6
7	DACK7	DRQ7

Two jumpers must be installed to select a DMA channel. The switch BANK A is used to select the DMA channel for Group 1, and the switch BANK B is used to select the DMA channel for Group 2. Figure 2-3 displays the jumper positions for selecting DMA Channels 5 and 6. In this setting, Group 1 uses DMA Channel 5, and Group 2 uses DMA Channel 6.

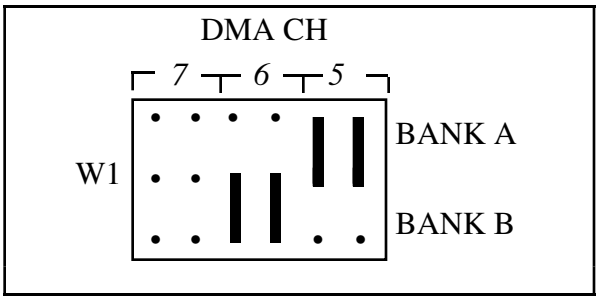


Figure 2-3. DMA Jumper Settings for DMA Channels 5 and 6 (Factory Settings)

If you want to use only one DMA channel, then place the configuration jumpers in the positions shown in Figure 2-4.

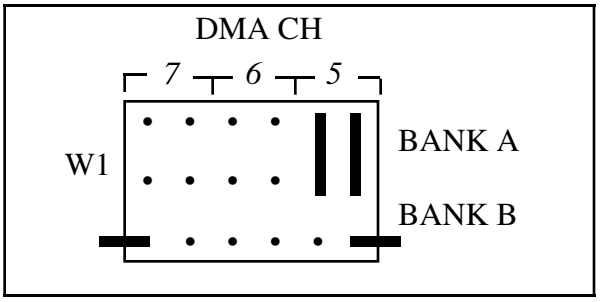


Figure 2-4. DMA Jumper Settings for DMA Channel 5 Only

If you do not want to use DMA for AT-DIO-32F transfers, then place the configuration jumpers in the positions shown in Figure 2-5.

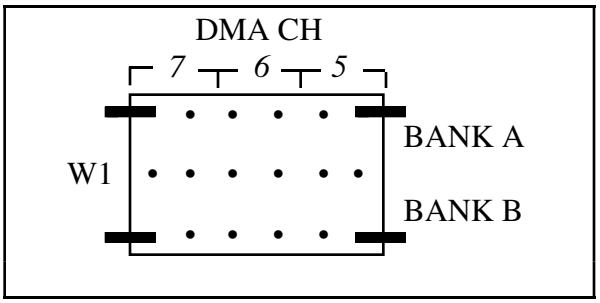


Figure 2-5. DMA Jumper Settings for Disabling DMA Transfers

Interrupt Selection

The AT-DIO-32F board can connect to one or two of any of the 11 interrupt lines of the PC I/O channel. The interrupt lines are selected by jumper W2, which is located above the I/O slot edge connector on the AT-DIO-32F (see Figure 2-1). To use the interrupt capability of the AT-DIO-32F, select one or two interrupt lines and place the jumpers in the appropriate positions to enable the particular interrupt lines. The Group 1 interrupt uses the upper two rows of jumper W2, and the Group 2 interrupt uses the lower two rows of jumper W2.

The AT-DIO-32F can share interrupt lines with other devices by using a tri-state driver to drive its selected interrupt line. The AT-DIO-32F hardware can use the following interrupt lines: IRQ3, IRQ4, IRQ5, IRQ6, IRQ7, IRQ9, IRQ10, IRQ11, IRQ12, IRQ14, and IRQ15.

Note: *Do not* use interrupt line 6 or interrupt line 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 default interrupt lines are IRQ11 for Group 1 and IRQ12 for Group 2. These lines are selected by placing the jumper on the pins in row 11 of W1 and row 12 of W1, respectively. Figure 2-6 shows the default interrupt jumper setting IRQ11 and IRQ12. To change the default setting, remove the jumpers from their current settings and place them on the new pins.

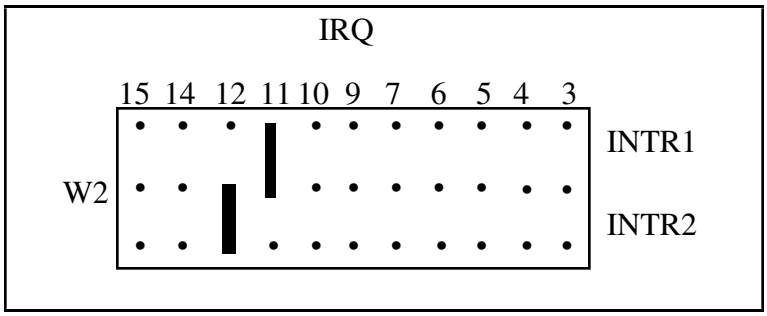


Figure 2-6. Interrupt Jumper Settings IRQ11 and IRQ12 (Factory Settings)

If you do not want to use interrupts, place the jumper on W2 in the positions shown in Figure 2-7. This setting disables the AT-DIO-32F from asserting any interrupt lines on the PC I/O channel.

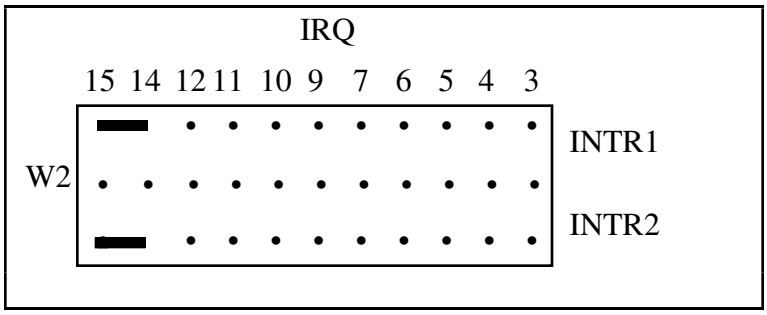


Figure 2-7. Interrupt Jumper Settings for Disabling Interrupts

Figure 2-8 shows the Group 2 interrupts disabled, and interrupt line 5 is selected for Group 1.

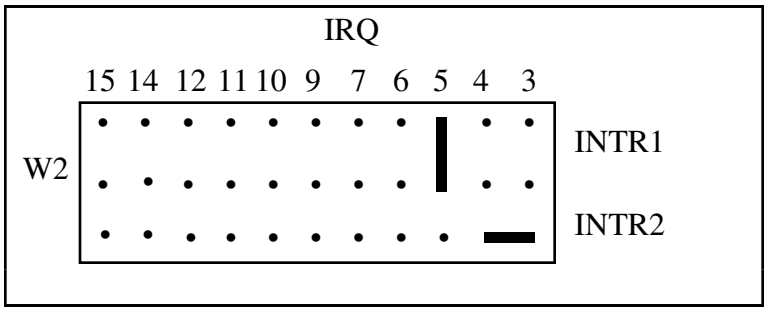


Figure 2-8. Interrupt Jumper Setting IRQ5 Only

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 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 jumper W3 determines whether a board drives the onboard 10-MHz clock onto the RTSI bus, receives the RTSI bus clock, or disconnects from the RTSI bus clock. This clock source, whether a local or RTSI signal, is then used as the onboard counter frequency source.

The jumper selections are shown in Table 2-5.

Table 2-5. Configurations for RTSI Bus Clock Selection

Configuration	W3
Disconnect board from RTSI bus clock; use local oscillator	STANDBY, BRDCLK–OSC (factory setting)
Receive RTSI bus clock signal	STANDBY, BRDCLK–RTSICLK
Drive RTSI bus clock signal with local oscillator	BRDCLK–OSC, BRDCLK–RTSICLK

Figures 2-9, 2-10, and 2-11 show the jumper positions for each of the preceding configurations.

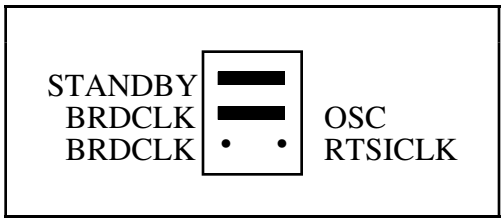


Figure 2-9. Disconnect from RTSI Bus Clock; Use Onboard Oscillator (Factory Settings)

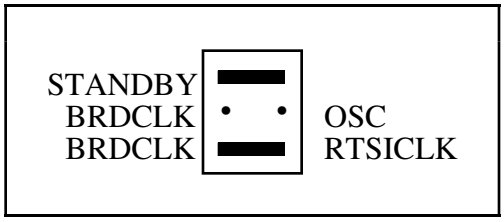


Figure 2-10. Receive RTSI Bus Clock Signal

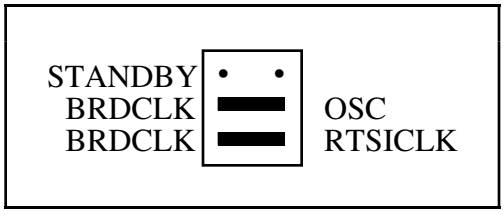


Figure 2-11. Drive RTSI Bus Clock Signal with Onboard Oscillator

Installation

The AT-DIO-32F can be installed in any available 16-bit expansion slot (AT style) in your computer. The AT-DIO-32F *does not* work if installed in an 8-bit expansion slot (PC style). After you have made any necessary changes, verified, and recorded the switch settings and jumper settings (a form is given in Appendix E), you are ready to install the AT-DIO-32F. 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-DIO-32F into a 16-bit slot. It may be a tight fit, but do not force the board into place.
5. If you want to connect multiple AT Series boards, attach a RTSI cable to the RTSI connector at this time.
6. Screw the mounting bracket of the AT-DIO-32F to the back panel rail of the computer.
7. Check the installation.
8. Replace the cover.

The AT-DIO-32F board is installed and ready for operation.

Signal Connections

I/O Connector Pin Description

Figure 2-12 shows the pin assignments for the AT-DIO-32F digital I/O connector.

Warning: ***Connections that exceed any of the maximum ratings of input or output signals on the AT-DIO-32F may result in damage to the AT-DIO-32F 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 any such signal connections.***

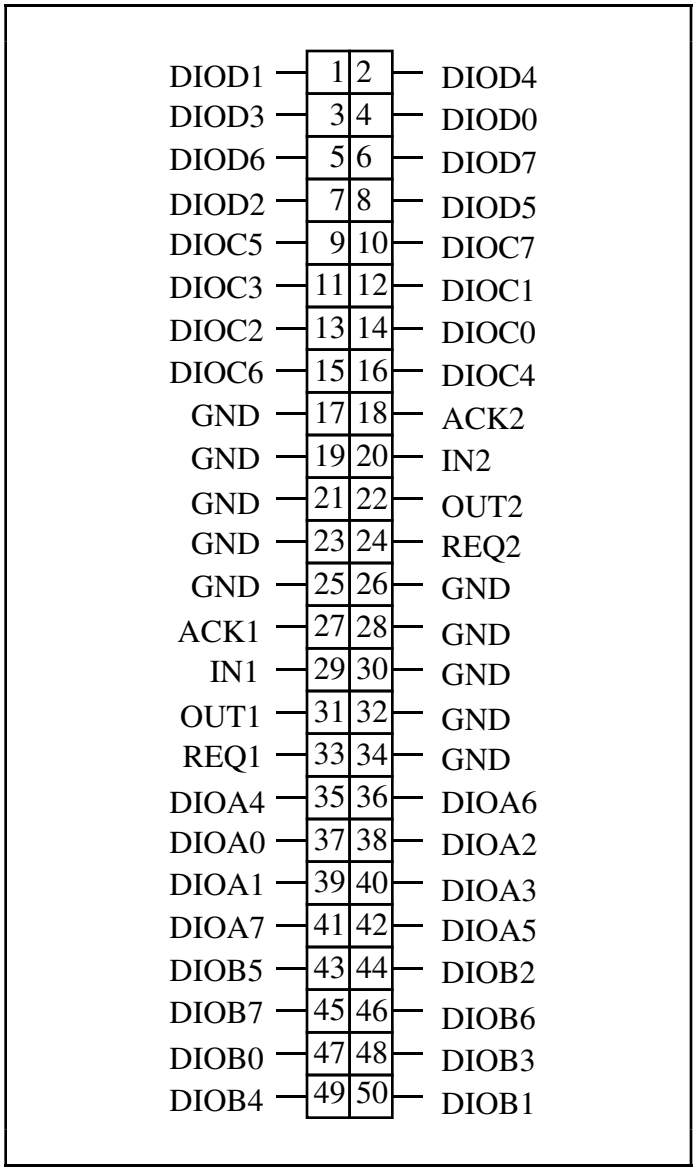


Figure 2-12. Digital I/O Connector Pin Assignments

Signal Connection Descriptions

Pins	Signal Names	Description
43-50	DIOB<0..7>	Bidirectional data lines for Port B. DIOB7 is the MSB; DIOB0 is the LSB.
35-42	DIOA<0..7>	Bidirectional data lines for Port A. DIOA7 is the MSB; DIOA0 is the LSB.
33	REQ1	Input handshaking request line for Group 1. When the AT-DIO-32F is in write mode, the external device activates this signal to indicate that it is ready to receive data. When the AT-DIO-32F is in read mode, the external device activates this signal if data can be read on the data lines. The polarity of this signal is changed by the INVRQ1 bit in the CFG1 Register.
31	OUT1	Extra output signal #1. This additional output signal can be connected to extra control lines, and is controlled by the OUT1 bit in the CFG1 Register.
	IN1	Extra input signal #1. This additional input signal is pulled up to +5 V by an onboard resistor. The status of this signal can be obtained by reading the IN1 bit in the STAT Register. This input signal can be used as an extra input signal line or as an external enable signal of Counter 1 of the board.
27	ACK1	Output handshaking acknowledge signal for Group 1. When the AT-DIO-32F is in write mode, this signal becomes active when data has been written to the data lines. When the AT-DIO-32F is in read mode, this signal becomes active when the available data on the data lines has been read. The polarity of this signal is configured by the INVACK1 bit in the CFG1 Register.
24	REQ2	Input handshaking request line for Group 2. When the AT-DIO-32F is in write mode, the external device should activate this signal to indicate that it is ready to receive data. When the AT-DIO-32F is in read mode, the external device should activate this signal if data is available to be read on the data lines. The polarity of this signal is changed by the INVRQ2 bit in the CFG2 Register.

Pins	Signal Names	Description (continued)
22	OUT2	Extra output signal #2. This additional output signal can be connected to extra control lines and is controlled by the OUT2 bit in the CFG2 Register.
20	IN2	Extra input signal #2. This additional input signal is pulled up to +5 V by an onboard resistor. The status of this signal can be obtained by reading the IN2 bit in the STAT Register. This input signal can be used as an extra input signal line or as an external enable signal of Counter 2 of the board.
18	ACK2	Output handshaking acknowledge signal for Group 2. When the AT-DIO-32F is in write mode, this signal becomes active when data has been written to the data lines. When the AT-DIO-32F is in read mode, this signal becomes active when the available data on the data lines has been read. The polarity of this signal is configured by the INVACK2 bit in the CFG2 Register.
9-16	DIOC<0..7>	Bidirectional data lines for Port C. DIOC7 is the MSB; DIOC0 is the LSB.
1-8	DIOD<0..7>	Bidirectional data lines for Port D. DIOD7 is the MSB; DIOD0 is the LSB.
17, 19, 21, 23, 25, 26 28, 30, 32, 34	GND	These signals are connected to the ground signal of the PC.

I/O Connector Electrical Specifications

I/O Signals Rating

Absolute maximum voltage input rating: -0.5 to Vcc +0.5 V
(Vcc: -0.5 V to 6.0 V)

Input Signal Specifications

	Minimum	Maximum
Input logic high voltage	2 V	5.5 V
Input logic low voltage	0 V	0.8 V
Input current		
at V_{cc} = 5.5 V, V_{in} = 5.5 V	–	10 µA

Output Signal Specifications

	Minimum	Maximum
Output logic high voltage		
at I_{out} = -15 mA	2.4 V	5 V
Output logic low voltage		
at I_{out} = 48 mA	0 V	0.5 V
Output logic high current	–	-30 mA
Output logic low current	–	70 mA

Timing Specifications

This section lists the timing specifications for handshaking with the AT-DIO-32F. The REQ and ACK signals are available on the I/O connector, and in the following diagrams they are non-inverted. The digital I/O ports are divided into two groups: Group 1 and Group 2. The timing specifications for Group 1 and Group 2 handshaking are identical.

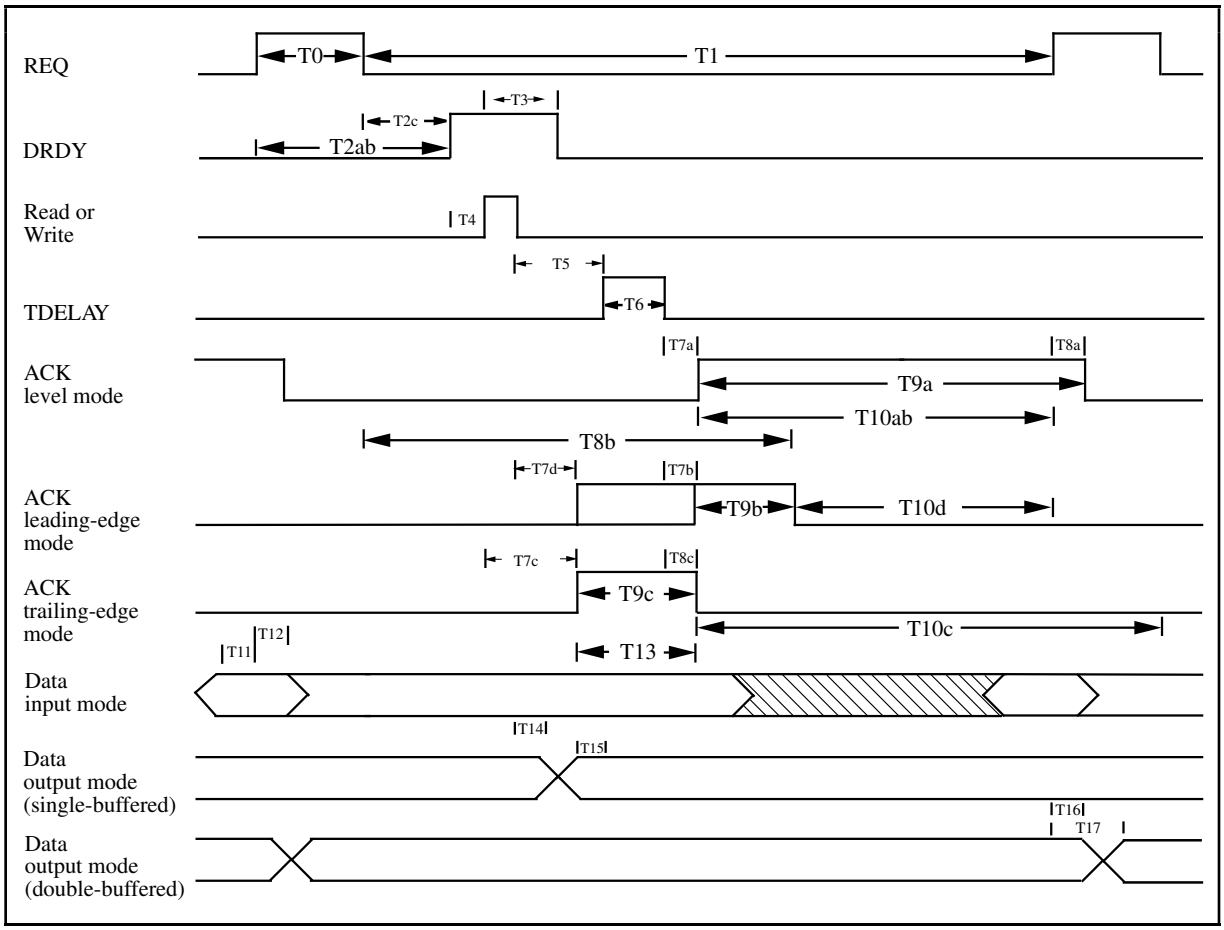
The following signals are used in the timing diagrams later in this chapter.

Name	Type	Description
GO	Internal	Internal GO pulse. This pulse is sent to the handshaking circuitry when the group's WRITE bit is set. This signal initializes the circuitry for a data write transfer. The GO signal is not available on the I/O connector.

(continues)

Name	Type	Description (continued)
REQ	Input	Handshaking request signal. If the AT-DIO-32F is in write mode, this signal is asserted when the external device is ready to receive data. If the AT-DIO-32F is in read mode, this signal is asserted when data is available to be read. This signal is available on the I/O connector.
ACK	Output	Handshaking acknowledge signal. If the AT-DIO-32F is in read mode, this signal is asserted by the AT-DIO-32F when it has read the available data. If the AT-DIO-32F is in write mode, this signal is asserted when the AT-DIO-32F has written the data to the specified port. This signal is available on the I/O connector.
DRDY	Internal	Data transfer ready. In read mode, this signal is high when data is available to be read. In write mode, this signal is high when the external device is ready to receive the data. The status of this signal is available in the STAT register. This signal is not available on the I/O connector.
RD	Internal	Read signal. This signal is the read signal generated from the control lines of the PC. This signal is not available on the I/O connector.
WR	Internal	Write signal. This signal is the write signal generated from the control lines of the PC. This signal is not available on the I/O connector.
TDELAY	Internal	Data transmission delay. A data-settling delay is added to ensure that data has settled during a transfer. The delay for Group 1 is controlled by bits T1S2 through T1S0 in the CFG1 Register. The delay for Group 2 is controlled by bits T2S2 through T2S0 in CFG2. This signal is not available on the I/O connector.
DATA	I/O	Data signals on the I/O connector. In write mode these lines are driven by the AT-DIO-32F, and data is transfered from memory to the external device. In read mode these lines are driven by the external device, and data is transfered from the external device to memory.

AT-DIO-32F Read and Write Timing



Name	Description	Minimum	Maximum
T0a	REQ pulse width in level mode	125	-
T0bc	REQ pulse width in leading- or trailing-edge mode	100	-
T1	REQ low duration	160	-
T2ab	REQ to DRDY in level or leading-edge mode	0	225
T2c	REQ inactive to DRDY in trailing-edge mode	0	100
T3	Start of read or write to DRDY inactive	50	240
T4	DRDY to read or write	0	-
T5	End of read or write to TDELAY		
	(with TDELAY equal to 0)	0	175
	(with TDELAY not equal to 0)	50	360
T6	TDELAY (programmable)	0	700
T7ab	TDELAY to ACK in level mode, or in leading-edge mode without LPULSE	10	100
T7c	Start of read or write to ACK in trailing-edge mode	60	220
T7d	End of read or write to ACK in leading-edge mode with LPULSE set	0	180

T8a	REQ to ACK inactive in level mode	110	320
T8b	REQ inactive to ACK inactive in leading-edge mode (but T9b can prolong ACK)	110	320
T8c	TDELAY to ACK inactive in trailing-edge mode	50	90
T9a	ACK pulse width in level mode (delaying REQ prolongs ACK)	225	-
T9b	ACK pulse width in leading-edge mode without LPULSE (but T8b can prolong ACK)	125	175
T9c	ACK pulse width in trailing-edge mode (increasing TDELAY prolongs ACK)	225	-
T10ab	ACK to next REQ in level mode or in leading-edge mode without LPULSE	35	-
T10c	ACK inactive to next REQ inactive in trailing-edge mode	0	-
T10d	ACK inactive to next REQ in leading-edge mode with LPULSE set	0	-
T11	Input data valid before REQ	0	-
T12	Input data valid after REQ (double-buffered input)	120	-
T13	Input data valid after ACK (single-buffered input)	0	-
T14	Old output data invalid after write (single-buffered output)	0	-
T15	Output data valid before TDELAY (single-buffered output)	5	-
T16	Old output data invalid after REQ (double-buffered output)	0	-
T17	Output data valid after REQ (double-buffered output)	0	100

All timing values are in nanoseconds.

Cabling

The AT-DIO-32F can be interfaced to a wide range of printers, plotters, test instruments, I/O racks and modules, screw terminal panels, and almost any device with a parallel interface. The AT-DIO-32F digital I/O connector is a standard 50-pin header connector. The pin assignments are compatible with the DEC DRV11J parallel interface and most standard 32-channel I/O module mounting racks (such as those manufactured by Opto 22 and Gordos).

The CB-50, a cable termination accessory, is available from National Instruments for use with the AT-DIO-32F board. This kit includes a 50-conductor, flat ribbon cable and a connector block. Signal input and output wires can be attached to screw terminals on the connector block and thereby connected to the AT-DIO-32F I/O connector.

The CB-50 is useful for initially prototyping an application or in situations where AT-DIO-32F interconnections are frequently changed. Once a final field wiring scheme has been developed, however, you may want to develop your own cable. This section contains information and guidelines for the design of such a cable.

The AT-DIO-32F I/O connector is a 50-pin male ribbon cable header. Recommended manufacturers and the appropriate 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-DIO-32F is a 50-position, polarized, ribbon socket connector with strain relief. National Instruments uses a polarized (keyed) connector to prevent inadvertent upside-down connection to the AT-DIO-32F. Recommended manufacturers and the appropriate 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 manufacturers and the appropriate part numbers for the standard ribbon cable (50-conductor, 28 AWG, stranded) that can be used with these connectors are as follows:

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

If you plan to use the AT-DIO-32F for a communications application, you may need shielded cables to meet FCC requirements. The AT-DIO-32F I/O bracket has been designed so that the shield of the I/O cable can be grounded through the computer chassis when a mating connector such as the following is used:

AMP Special Industries	part number 2-746483-2

Many varieties of shielded ribbon cable can work with the preceding mating connector. One type of shielded cable encloses a standard ribbon cable with a shielded jacket. Recommended manufacturers and the appropriate part numbers for this type of cable are as follows:

Belden Electronic Wire and Cable	part number 9L28350
T&B/Ansley Corporation	part number 187-50

Chapter 3
Theory of Operation

This chapter explains the basic operation of the AT-DIO-32F circuitry.

The AT-DIO-32F is a high-speed, 32-bit, parallel, digital I/O interface for the PC. The 32 lines of digital I/O on the AT-DIO-32F are divided into four 8-bit ports (DIOA, DIOB, DIOC, and DIOD). Ports A and B are assigned to handshaking Group 1, and Ports C and D are assigned to handshaking Group 2. Each group can be programmed as either an input or an output group, and each group has its own independent handshaking signals for data transfers.

The key functional components of the hardware are illustrated in the block diagram shown in Figure 3-1.

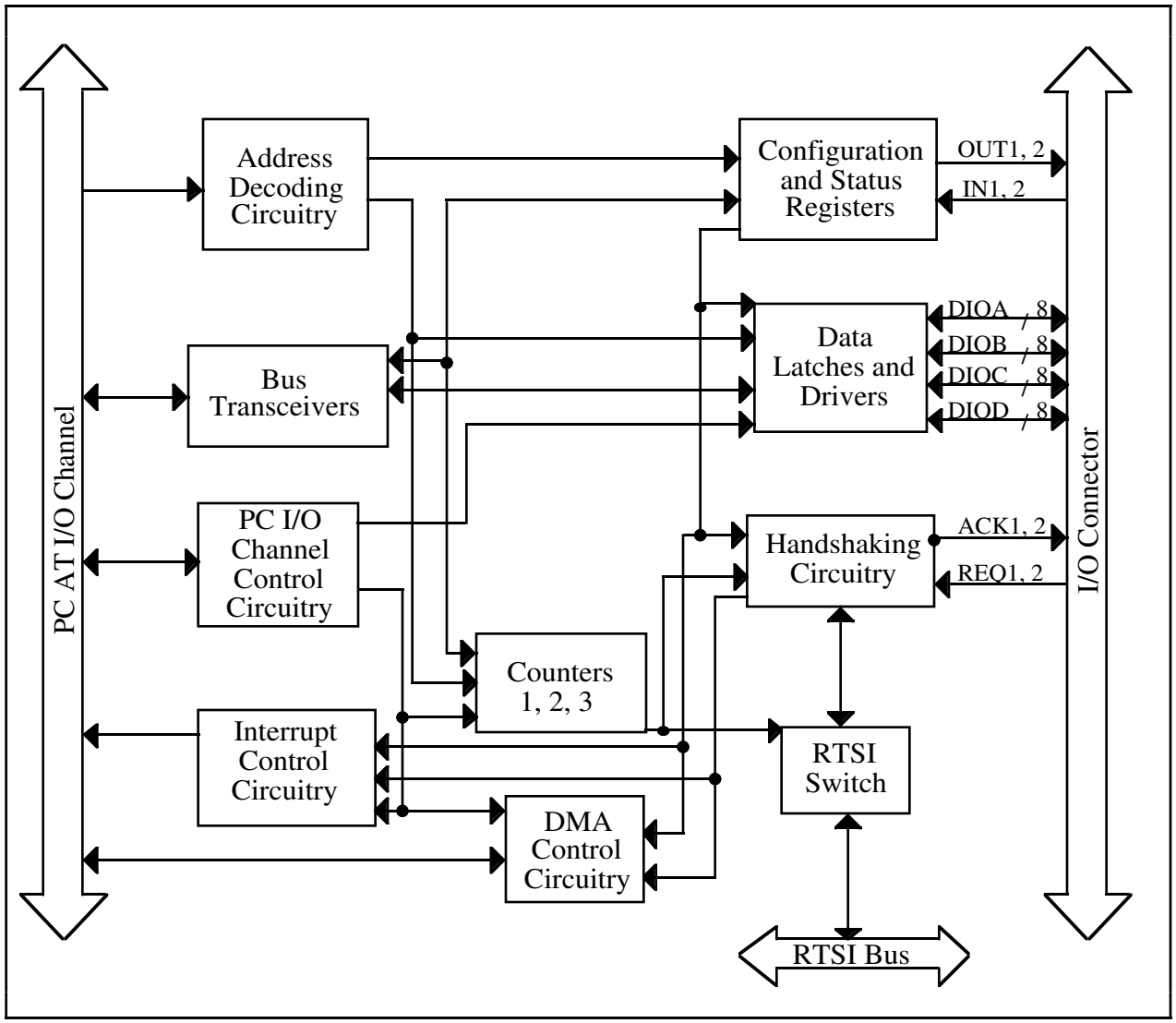


Figure 3-1. AT-DIO-32F Block Diagram

The AT-DIO-32F 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.

Address Decoder

The PC I/O channel has 24 address lines; the AT-DIO-32F uses ten of these lines to decode the board address. Therefore, the board address range is hex 000 to 3FF. Address lines SA5 through SA9 are used to generate the board enable signal. SA0 through SA4 are used to select the onboard registers.

Bus Transceivers

The bus transceivers control the sending and receiving of the data lines to and from the PC I/O channel.

PC I/O Channel Control Circuitry

This circuitry monitors and transmits the PC I/O channel control and support signals. The control signals identify transfers as read or write, configuration or I/O, and 8-bit or 16-bit. A support signal is returned to the PC I/O channel from the AT-DIO-32F to indicate the size of the current data transfer.

Configuration and Status Registers

The AT-DIO-32F has seven configuration registers and a status register. Four 16-bit configuration registers (CFG1, CFG2, CFG3, and CFG4) are used to program all of the digital I/O modes of the
AT-DIO-32F. The other configuration registers are used to configure three onboard counters and the RTSI bus and to clear certain interrupt status bits. The 16-bit status register (STAT) contains DMA, interrupt, and handshaking signal status information. Refer to Chapter 4, *Programming*, for additional information about these registers.

Data Latches and Drivers

The four 8-bit digital I/O ports are divided into two handshaking groups. Ports A and B are assigned to handshaking Group 1. Ports C and D are assigned to handshaking Group 2. Each port can be configured as read or write, and single-buffered or doubled-buffered. When the board is first turned on, each port is configured as a single-buffered read port. Reading a single-buffered input port returns the data currently available for that port at the I/O connector. Data is latched in a read port on the appropriate active edge of the handshaking request line, REQ1 or REQ2, when the port is configured as a double-buffered input port.

If a port is configured as a single-buffered write port, the data written to that port is latched into the port and driven on the corresponding digital I/O lines. Reading the port returns the data that is currently driven by the port. Write ports configured for double-buffering are often used for pattern generation. A double-buffered write port consists of two buffers: the write buffer and the output buffer. Data written to the port is loaded into the write buffer. When a handshaking request, REQ1 or REQ2, is received on the I/O connector for the double-buffered group, the contents of the write buffer are loaded into the output buffer. Data loaded in the output buffer is driven on the digital I/O lines.

Onboard Counters

The AT-DIO-32F includes three onboard counters, useful for pattern generation and periodic data acquisition. Counter 1 can be programmed to generate group 1 handshaking requests on the REQ1 line. Likewise, counter 2 can be programmed to generate group 2 handshaking requests on the REQ2 line. Counter 3 can alter the counting rate of the other two counters and can also be programmed to generate periodic interrupts.

The clock that runs the counters, BRDCLK, connects to a 10 MHz clock source: either an onboard clock, OSC, or to the RTSI clock line, RTSICLK. With BRDCLK connected to both OSC and RTSICLK, the onboard clock source can drive both the counters and the RTSI clock line. See *RTSI Bus Clock Selection* in Chapter 2, *Configuration and Installation*, for information about connecting the BRDCLK signal.

The signal you select for BRDCLK is slowed by a factor of 5, creating a 2 MHz clock signal that operates Counter 3. The output of Counter 3 forms a time base for Counters 1 and 2, but Counters 1 and 2 can also run directly from the BRDCLK clock, if so directed in CFG3 register. Figure 3-2 shows the clock routing scheme.

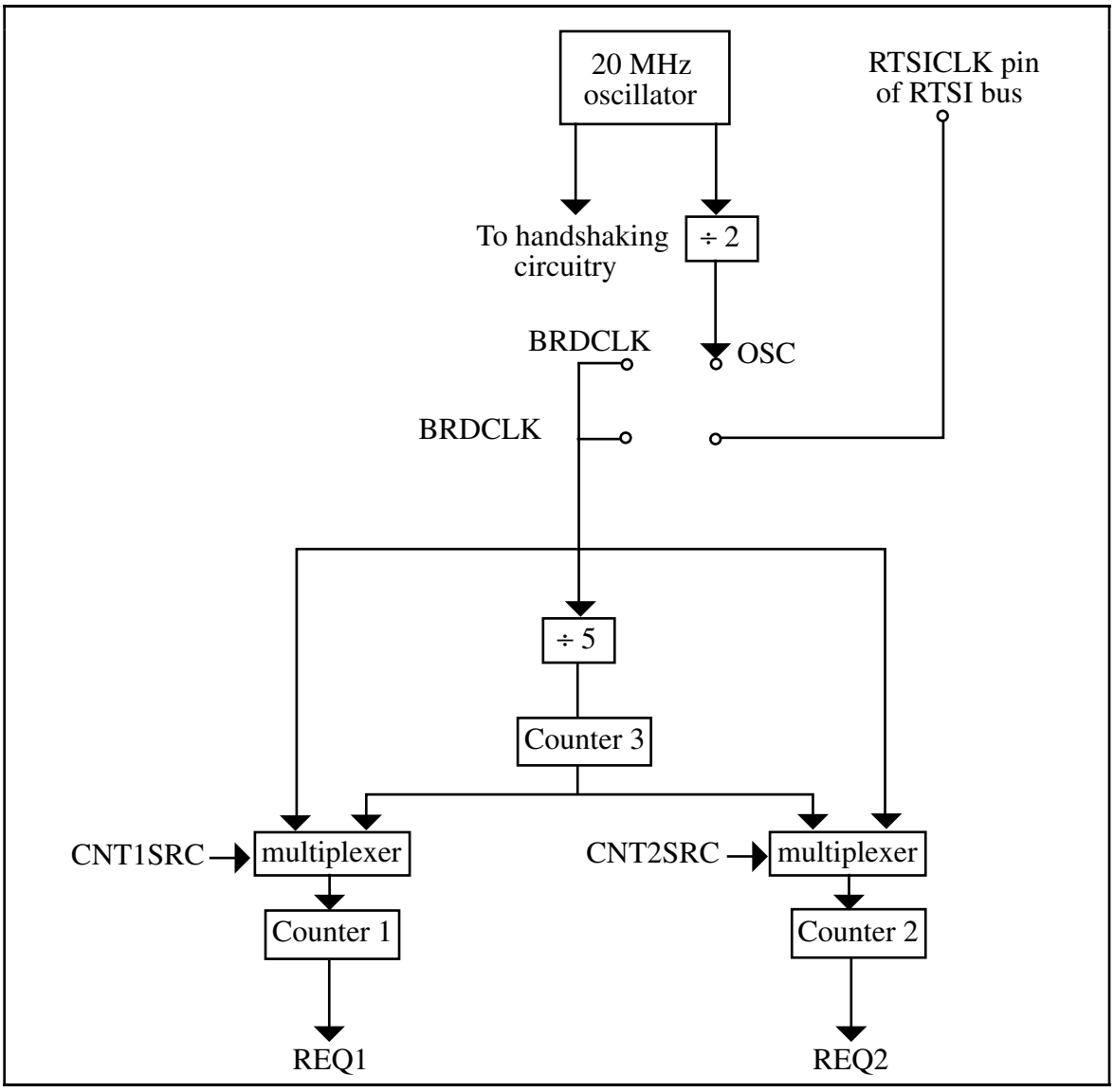


Figure 3-2. AT-DIO-32F Clock Routing Scheme

Digital I/O Connector

All digital I/O is through a standard, 50-pin, male connector. The pin assignments for this connector are compatible with the DEC DRV11-J parallel interface and most 32-channel I/O module racks. Refer to *Signal Connections* in Chapter 2, *Configuration and Installation*, for pin assignments and additional information.

Handshaking Circuitry

The four 8-bit digital I/O ports are divided into two handshaking groups: Group 1 and Group 2. These groups are independent of each other, so two separate transfers can occur simultaneously.

Group 1 handshaking is controlled by REQ1 and ACK1. Group 2 handshaking is controlled by REQ2 and ACK2. The handshaking circuitry controls these handshaking signals and the data flow through the digital I/O data latches and drivers.

Each group can be programmed to operate in one of three handshaking modes: level signals, leading edge signals, or trailing edge signals. Refer to Chapter 2, *Configuration and Installation*, and Chapter 4, *Programming*, for the timing diagrams for each of these modes. The state diagrams that describe each handshaking mode are shown later in this chapter.

After start-up or LRESET, the handshaking circuitry is in the INIT state. During a read transfer, DRDY is set once a REQ is received. DRDY remains set until the data is read from the selected ports. During a write transfer, DRDY is set as soon as the group's WRITE bit is set, indicating that the group is in write mode. DRDY remains set until data is written to the selected ports.

In the Figures 3-3 and 3-4, REQ and ACK are shown as active high logic; that is, INVRQ and INVACK are cleared.

Level Mode

In level mode, once the data is read or written, TDELAY begins. TDELAY is the data-settling delay that is programmed by setting bits in CFG1 or CFG2. After the delay, ACK is sent to the digital I/O connector. The circuitry waits for REQ to go low. When REQ is received again, ACK is cleared; then DRDY is set again and the cycle continues.

Figure 3-3 shows a read transfer in level mode.

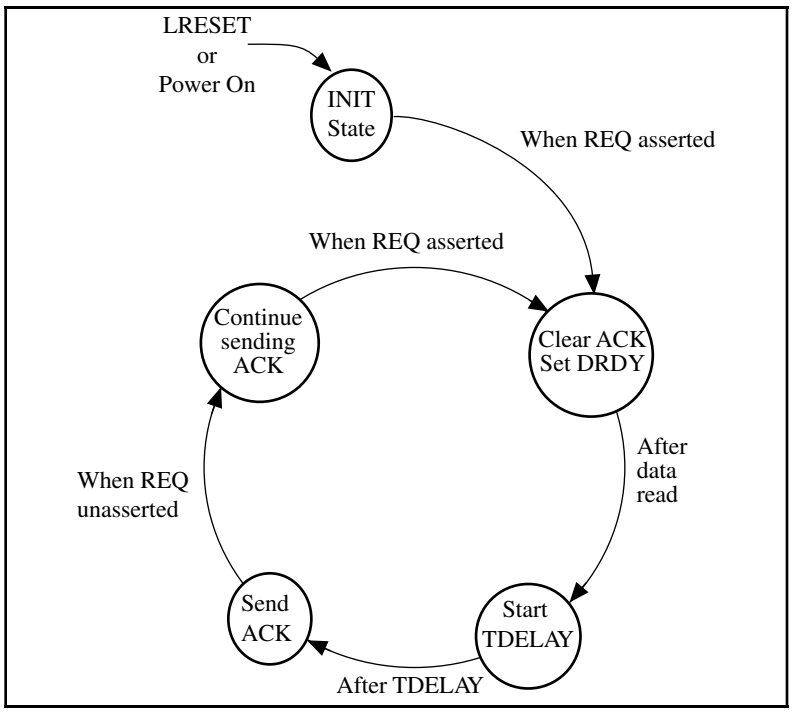


Figure 3-3. Level Mode – Read

Figure 3-4 shows a write transfer in level mode.

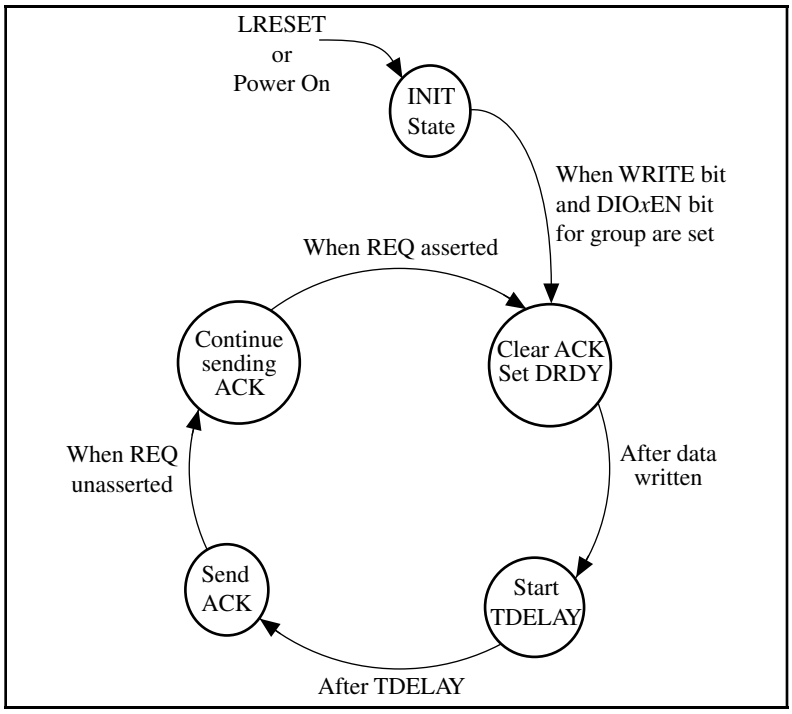


Figure 3-4. Level Mode – Write

Leading Edge Mode

In leading edge mode, the handshaking lines REQ and ACK are viewed as pulses that are active on the leading edge of the pulse. Once the data is read or written, TDELAY begins. After TDELAY, ACK is sent to the digital I/O connector. When another leading edge of REQ is received, DRDY is set and the AT-DIO-32F is ready for another cycle.

Figure 3-5 shows a read transfer in leading edge mode.

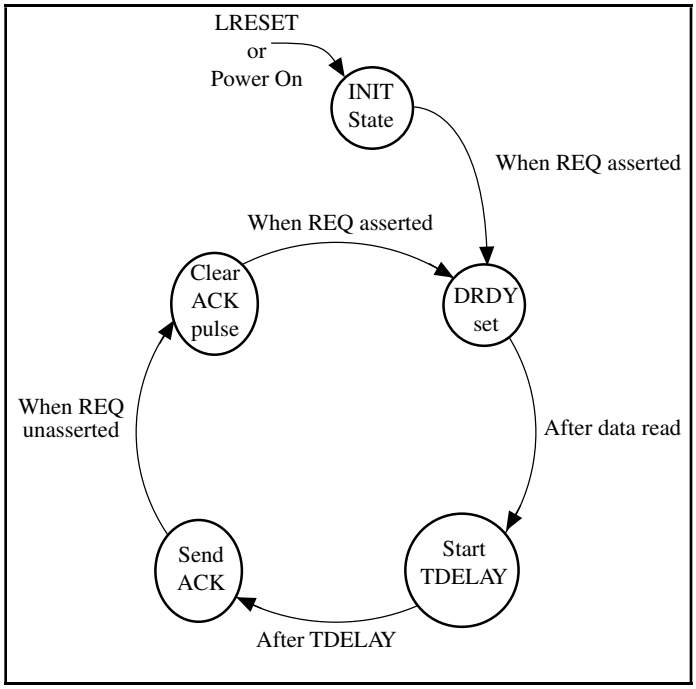


Figure 3-5. Leading Edge Mode – Read

Note: If LPULSE is set, ACK starts when TDELAY starts.

Figure 3-6 shows a write transfer in leading edge mode.

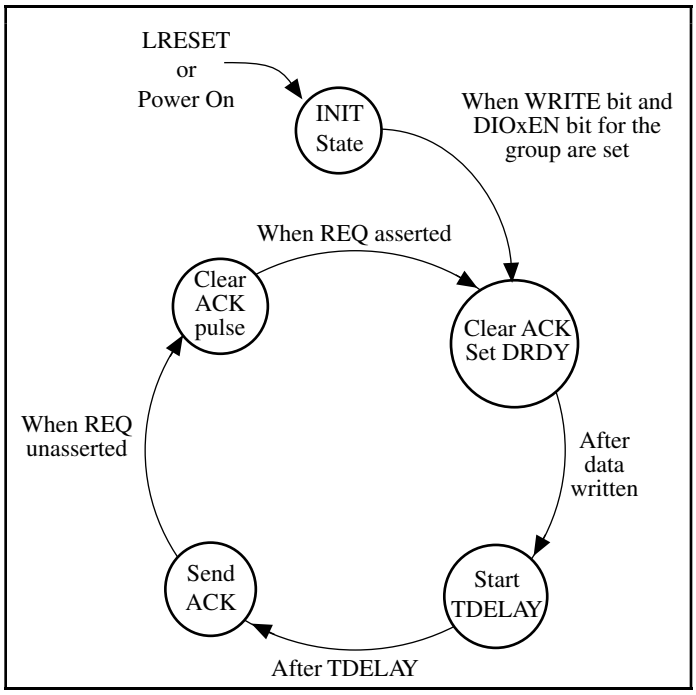


Figure 3-6. Leading Edge Mode – Write

Note: If LPULSE is set, ACK starts when TDELAY starts.

Trailing Edge Mode

In trailing edge mode, REQ and ACK are treated as pulses that are active on the trailing edge of the pulse. Once the data is read or written, ACK is asserted. At this time, TDELAY begins. After TDELAY, ACK is cleared, which means that the pulse width of ACK is equivalent to TDELAY. If TDELAY is programmed to 0, the pulse width of ACK is 100 nsec. When another trailing edge of REQ is received, DRDY is set and the AT-DIO-32F is ready for another cycle.

Figure 3-7 shows a read transfer in trailing edge mode.

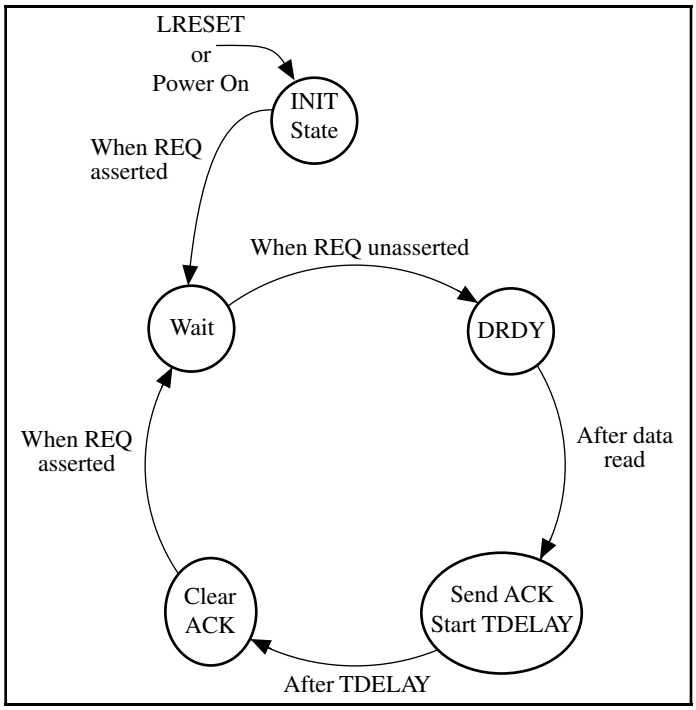


Figure 3-7. Trailing Edge Mode – Read

Figure 3-8 shows a write transfer in trailing edge mode.

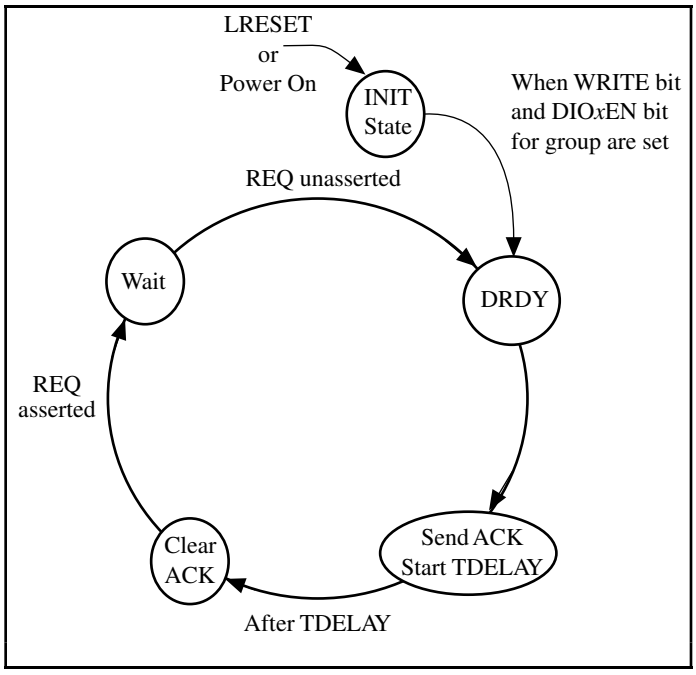


Figure 3-8. Trailing Edge Mode – Write

Interrupt Control Circuitry

The interrupt control circuitry routes any enabled interrupts to the selected interrupt request lines. Eleven interrupt request lines are available for use by the AT-DIO-32F: IRQ3, IRQ4, IRQ5, IRQ6, IRQ7, IRQ9, IRQ10, IRQ11, IRQ12, IRQ14, and IRQ15. Group 1 and Group 2 can each be set to have an interrupt request line. With the interrupt requests, which are tri-state output signals, the AT-DIO-32F board can share the interrupt lines with other devices. Five different interrupts can be generated by the AT-DIO-32F: DRDY1 set, DRDY2 set, Group 1 DMA terminal count received, Group 2 DMA terminal count received, and a rising edge on Counter 3 received. Each one of these interrupts is individually enabled and cleared. See Chapter 4, *Programming*, for additional information about programming with interrupts.

DMA Control Circuitry

Each handshaking group can be assigned a separate DMA channel for 16-bit data transfer, and each group has a DMA enable bit, DMAEN. When DMA is enabled, the AT-DIO-32F sends a DMA request to a port that is ready to receive data during a write transfer, or to a port that is ready to read data during a read transfer. DMA channels 5, 6, and 7 of the PC I/O channel are available for such transfers.

RTSI Bus Interface

The AT-DIO-32F 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 that have RTSI bus connectors can be wired together inside the PC to share these signals.

The RTSI bus RTSICLK line can be used to send a 10-MHz signal across the RTSI bus, or to receive another clock signal from another AT board connected to the RTSI bus. MYCLK is the system clock used by the AT-DIO-32F.

The RTSI switch is a National Instruments custom-integrated circuit that acts as a seven by seven 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 drive any of the seven trigger line signals onto any one or more of the pins A<6..0>. With this capability, any AT Series board sharing the RTSI bus has a completely flexible signal interconnection scheme. The RTSI switch is programmed via its select and data inputs.

On the AT-DIO-32F board, seven signals are connected to pins A<6..0> of the RTSI switch: REQ1, REQ2, ACK1, ACK2, IN1, RWGRP1*, and RWGRP2*. These signals can be controlled over the RTSI bus or externally across the I/O connector. The RTSI bus connections 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-DIO-32F control and status registers. This chapter also includes important information about programming the AT-DIO-32F.

The AT-DIO-32F has four 8-bit ports divided into two handshaking groups, Group 1 and Group 2. Each group can be independently programmed for input or output and a handshaking mode. Counters onboard can be used for pattern generation for one or both handshaking groups. A signal from another AT Series board can be sent across the RTSI bus to implement pattern generation as well. This chapter contains the register descriptions and functional descriptions necessary to program the many modes of the AT-DIO-32F.

Note: If you plan to use a programming software package such as NI-DAQ or LabWindows with your AT-DIO-32F board, you need not read this chapter.

Register Map

The register map for the AT-DIO-32F 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-DIO-32F Register Map

Register Name	Address (Hex)	Type	Size
Configuration and Status Register Group			
CFG1 Register	00	Write-only	16-bit
CFG2 Register	02	Write-only	16-bit
CFG3 Register	04	Write-only	16-bit
CFG4 Register	14	Write-only	16-bit
STAT Register	00	Read-only	16-bit
CNTINTCLR Register	0A	Write-only	16-bit
DMACLR1 Register	0C	Write-only	16-bit
DMACLR2 Register	0E	Write-only	16-bit
Digital I/O Port Register Group			
PORT A Register	06	Read-and-write	8-bit or 16-bit
PORT B Register	07	Read-and-write	8-bit
PORT C Register	08	Read-and-write	8-bit or 16-bit
PORT D Register	09	Read-and-write	8-bit
RTSI Bus Register Group			
RTSISHFT Register	10	Write-only	8-bit
RTSISTRB Register	12	Write-only	8-bit
Counter Register Group			
CNTR1 Register	18	Read-and-write	8-bit
CNTR2 Register	1A	Read-and-write	8-bit
CNTR3 Register	1C	Read-and-write	8-bit
CNTRCMD Register	1E	Write-only	8-bit

Register Sizes

Two different transfer sizes can be used for read and write operations with the PC: byte (8-bit), and word (16-bit). Table 4-1 shows the size of each AT-DIO-32F register. For example, reading the STAT Register requires a 16-bit (word) read operation at the selected address, whereas writing to the RTSISHFT Register requires an 8-bit (byte) write operation at the selected address. Addresses selected as 8-bit locations can also be accessed with a 16-bit operation; the upper byte in this case should be thought of as eight *don't care* bits.

Register Description

Table 4-1 divides the AT-DIO-32F registers into four 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-DIO-32F hardware. The configuration registers are used to program the digital I/O handshaking modes and to enable DMA or interrupt requests. The status registers reflect the state of the digital I/O handshaking, interrupt requests, and DMA requests. The registers in the Digital I/O Port Group access the four 8-bit digital I/O ports. The Counter Register Group selects the counting mode and initial count of the three counters. The RTSI Bus Register Group configures the RTSI bus switch.

Register Description Format

The remainder of this section discusses each of the AT-DIO-32F 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 this square. An asterisk (*) after the bit name indicates that the bit is inverted (negative logic).

In many of the registers, one or more bits are labeled with Xs, 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. When a register is written to, setting or clearing these bit locations has no effect on the AT-DIO-32F hardware.

Configuration and Status Register Group

The seven registers making up the Configuration and Status Register Group can be used for general monitoring and control of the AT-DIO-32F hardware. The four configuration registers (CFG1, CFG2, CFG3, and CFG4) control the digital I/O modes, handshaking modes, interrupt, and DMA operations. The other three configuration registers (CNTINTCLR, DMACLR1, and DMACLR2) clear various interrupt status bits. The status register (STAT) reflects the DMA, interrupt, and handshaking signal status.

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

CFG1 Register

The CFG1 Register contains 16 bits that control the Group 1 I/O mode, handshaking mode, and interrupt and DMA operations.

Address: Base address + 00 (hex)

Type: Write-only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
DMAEN1	INTEN1	T1S2	T1S1	T1S0	DIOBEN	DIOAEN	LRESET1

7	6	5	4	3	2	1	0
X	INVRQ1	DBLBUFA	PULSE1	EDGE1	INVACK1	SETACK1	OUT1

Bit	Name	Description
15	DMAEN1	Group 1 DMA Enable Bit. When DMAEN1 is set, DMA is enabled for Group 1 handshaking. A DMA request is asserted when DRDY1 is set.
14	INTEN1	Group 1 Interrupt Enable Bit. When INTEN1 is set, interrupts are enabled for Group 1 handshaking. An interrupt request is asserted when DRDY1 is set.
13-11	T1S<2..0>	Group 1 Data Transmission Delay (TDELAY1) Bits. These bits select the data-settling delay used by Group 1 handshaking. Long cable lengths or special handshaking specifications may require a data-settling delay to ensure proper data transmission.

T1S2	T1S1	T1S0	TDELAY1 (nsec)
0	0	0	0
0	0	1	100
0	1	0	200
0	1	1	300
1	0	0	400
1	0	1	500
1	1	0	600
1	1	1	700

Bit	Name	Description (continued)
10	DIOBEN	Port B Handshaking Enable Bit. When DIOBEN is set, Port B is enabled for handshaking.
9	DIOAEN	Port A Handshaking Enable Bit. When DIOAEN is set, Port A is enabled for handshaking.
8	LRESET1	Local Reset for Group 1 Bit. Setting and clearing LRESET1 resets the handshaking circuitry for Group 1. Handshaking configuration bits must be reset after an LRESET1.
7	X	Don't Care Bit. This bit is unused on the Revision C and later revisions of the board. On the Revision B board this bit is used as the Group 1 Handshaking Enable Bit.
6	INVRQ1	Group 1 Request Invert Bit. When INVRQ1 is set, the handshaking request REQ1 is considered an active low signal. Group 1 recognizes a request when REQ1 is low. When INVRQ1 is cleared, REQ1 is considered an active high signal. Group 1 recognizes a request when REQ1 is high.
5	DBLBUFA	Port A Double-Buffer Enable Bit. If DBLBUFA is set, Port A is double-buffered. When Port A is configured as an output port, a write operation to the port loads data into the first buffer of the port. When a REQ1 is received, the data is transferred to the second buffer of the port, which dumps the data to the digital I/O connector. Double-buffering is usually used for pattern generation. If DBLBUFA is cleared, Port A is a single buffer; that is, data written to the port is immediately dumped to the digital I/O connector. When Port A is configured as an input port and DBLBUFA is set, an active level or edge of a REQ1 signal latches the data into the input buffer of Port A. A read operation reads the data in the buffer instead of the I/O connector. If DBLBUFA is cleared, the port is transparent; that is, a read operation reads the data on the I/O connector.
4	PULSE1	Group 1 Pulse Mode Bit. When PULSE1 is set, the Group 1 handshaking signals REQ1 and ACK1 are configured as pulse signals. When PULSE1 is cleared, the Group 1 handshaking signals are configured as level signals.
3	EDGE1	Group 1 Leading/Trailing Pulse Mode Bit. When EDGE1 and PULSE1 are both set, the Group 1 handshaking signals are active on the trailing edge of the pulse. When EDGE1 is cleared and PULSE1 is set, the Group 1 handshaking signals are active on the leading edge of the pulse.

Bit	Name	Description (continued)
2	INVACK1	Group 1 Acknowledge Invert Bit. When INVACK1 is set, the handshaking acknowledge signal ACK1 is configured as an active low signal. Group 1 sends ACK1 as a low signal to acknowledge the end of a data transfer. When INVACK1 is cleared, ACK1 is configured as an active high signal. Group 1 sends ACK1 as a high signal to acknowledge the end of a data transfer.
1	SETACK1	Group 1 Acknowledge Control Bit. Setting and clearing SETACK1 controls the ACK1 bit on the digital I/O connector.
0	OUT1	Extra Output Bit 1. Setting and clearing OUT1 controls the OUT1 bit on the digital I/O connector.

CFG2 Register

The CFG2 Register contains 16 bits that control the Group 2 I/O mode and the handshaking mode.

Address: Base address + 02 (hex)

Type: Write-only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
DMAEN2	INTEN2	T2S2	T2S1	T2S0	DIODEN	DIOCEN	LRESET2

7	6	5	4	3	2	1	0
X	INVRQ2	DBLBUFC	PULSE2	EDGE2	INVACK2	SETACK2	OUT2

Bit	Name	Description
15	DMAEN2	Group 2 DMA Enable Bit. When DMAEN2 is set, DMA is enabled for Group 2 handshaking. A DMA request is asserted when DRDY2 is set.
14	INTEN2	Group 2 Interrupt Enable Bit. When INTEN2 is set, interrupts are enabled for Group 2 handshaking. An interrupt request is asserted when DRDY2 is set.
13-11	T2S<2..0>	Group 2 Data Transmission Delay (TDELAY2) Bit. These bits select the data-settling delay used by Group 2 handshaking. Long cable lengths or special handshaking specifications may require a data-settling delay to ensure proper data transmission.

T2S2	T2S1	T2S0	TDELAY2 (nsec)
0	0	0	0
0	0	1	100
0	1	0	200
0	1	1	300
1	0	0	400
1	0	1	500
1	1	0	600
1	1	1	700

Bit	Name	Description (continued)
10	DIODEN	Port D Handshaking Enable Bit. When DIODEN is set, Port D is enabled for handshaking.
9	DIOCEN	Port C Handshaking Enable Bit. When DIOCEN is set, Port C is enabled for handshaking.
8	LRESET2	Local Reset for Group 2 Bit. Setting and then clearing LRESET2 resets the handshaking circuitry for Group 2. Handshaking configuration bits must be reset after an LRESET2.
7	X	Don't Care Bit. This bit is unused on the Revision C board. On the Revision B board this bit is used as the Group 1 Handshaking Enable Bit.
6	INVRQ2	Group 2 Request Invert Bit. When INVRQ2 is set, the handshaking request REQ2 is considered an active low signal. Group 2 recognizes a request when REQ2 is low. When INVRQ2 is cleared, REQ2 is considered an active high signal. Group 2 recognizes a request when REQ1 is high.
5	DBLBUFC	Port C Double-Buffer Enable Bit. If DBLBUFC is set, Port C is double-buffered. When Port C is configured as an output port, a write operation to the port loads data into the first buffer of the port. When a REQ2 is received, the data is transferred to the second buffer of the port, which dumps the data to the digital I/O connector. Double-buffering is usually used for pattern generation. If DBLBUFC is cleared, Port C is a single buffer; that is, data written to the port is immediately dumped to the digital I/O connector. When Port C is configured as an input port, and DBLBUFC is set, an active level or edge of a REQ2 signal latches the data into the input buffer of Port C. A read operation reads the data in the buffer instead of the I/O connector. If DBLBUFC is cleared, the port is transparent; that is, a read operation reads the data on the I/O connector.
4	PULSE2	Group 2 Pulse Mode Bit. When PULSE2 is set, the Group 2 handshaking signals REQ2 and ACK2 are configured as pulse signals. When PULSE2 is cleared, the Group 2 handshaking signals are configured as level signals.
3	EDGE2	Group 2 Leading/Trailing Pulse Mode Bit. When EDGE2 and PULSE2 are both set, the Group 2 handshaking signals are active on the trailing edge of the pulse. When EDGE2 is cleared and PULSE2 is set, the Group 2 handshaking signals are active on the leading edge of the pulse.

Bit	Name	Description (continued)
2	INVACK2	Group 2 Acknowledge Invert Bit. When INVACK2 is set, the handshaking acknowledge signal ACK2 is configured as an active low signal. Group 2 sends ACK2 as a low signal to acknowledge the end of a data transfer. When INVACK2 is cleared, ACK2 is configured as an active high signal. Group 2 sends ACK2 as a high signal to acknowledge the end of a data transfer.
1	SETACK2	Group 2 Acknowledge Control Bit. Setting and clearing SETACK2 controls the ACK2 bit on the digital I/O connector.
0	OUT2	Extra Output Bit 2. Setting and clearing OUT2 controls the OUT2 bit on the digital I/O connector.

CFG3 Register

The CFG3 Register contains 13 bits that control the Counter 1 and Counter 2 operations, clock selection, and I/O port transfer mode.

Address: Base address + 04 (hex)

Type: Write-only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
DBLBUFB	WRITED	WRITEB	TRANS32	WRITEC	WRITEA	CNT2SRC	CNT1SRC

7	6	5	4	3	2	1	0
DBLDMA	CNTINTEN	CNT2HSEN	CNT1HSEN	CNT2EN	CNT1EN	TCINTEN2	TCINTEN1

Bit	Name	Description
15	DBLBUFB	Port B Double-Buffer Enable Bit. If DBLBUFB is set, Port B is double-buffered. When Port B is configured as an output port, a write operation to the port loads data into the first buffer of the port. When a REQ1 is received, the data is transferred to the second buffer of the port, which dumps the data to the digital I/O connector. Double-buffering is usually used for pattern generation. If DBLBUFB is cleared, Port B is a single buffer; that is, data written to the port is immediately dumped to the digital I/O connector. When Port B is configured as an input port and DBLBUFB is set, an active level or edge of a REQ1 signal latches the data into the input buffer of Port B. A read operation reads the data in the buffer instead of the I/O connector. If DBLBUFB is cleared, the port is transparent; that is, a read operation reads the data on the I/O connector.
14	WRITED	Port D Write/Read Bit. When WRITED is set, Port D is configured for a write operation. When WRITED is cleared, Port D is configured for a read operation. If Port D is configured for a write operation, after every LRESET2 this bit must first be cleared and then set again for reintializing handshaking purposes (or clear and set DIODEN of the CFG2 Register).
13	WRITEB	Port B Write/Read Bit. When WRITEB is set, Port B is configured for a write operation. When WRITEB is cleared, Port B is configured for a read operation. If Port B is configured for a write operation, after every LRESET1 this bit must first be cleared and then set again for reintializing handshaking purposes (or clear and set DIOBEN of the CFG1 Register).

Bit	Name	Description (continued)
12	TRANS32	32-Bit Transfer Enable. When TRANS32 is set, the AT-DIO-32F is in 32-bit transfer mode. This mode uses the Group 1 handshaking lines (REQ1 and ACK1). When REQ1 is received, DRDY1 and DRDY2 are both set. When data is read or written for both Group 1 and Group 2, the ACK1 line is asserted. Two read or write operations are required for each 32-bit transfer. If TRANS32 is cleared, the AT-DIO-32F is in regular 16-bit mode.
11	WRITEC	Port C Write/Read Bit. When WRITEC is set, Port C is configured for a write operation. When WRITEC is cleared, Port C is configured for a read operation. If Port C is configured for a write operation, after every LRESET2 this bit must first be cleared and then set again for reinitializing handshaking purposes (or clear and set DIOCEN of the CFG2 Register).
10	WRITEA	Port A Write/Read Bit. When WRITEA is set, Port A is configured for a write operation. When WRITEA is cleared, Port A is configured for a read operation. If Port A is configured for a write operation, after every LRESET1 this bit must first be cleared and then set again for reinitializing handshaking purposes (or clear and set DIOAEN of the CFG1 Register).
9	CNT2SRC	Counter 2 Source Select Bit. The CNT2SRC bit selects the counting source for Counter 2. If CNT2SRC is set, the output of Counter 3 is used as the counting source for Counter 2. If CNT2SRC is cleared, a 10-MHz clock is used as the counting source for Counter 2.
8	CNT1SRC	Counter 1 Source Select Bit. The CNT1SRC bit selects the counting source for Counter 1. If CNT1SRC is set, the output of Counter 3 is used as the counting source for Counter 1. If CNT1SRC is cleared, a 10-MHz clock is used as the counting source for Counter 1.
7	DBLDMA	Double DMA Mode Enable Bit. When DBLDMA is set, the AT-DIO-32F is in double DMA channel mode. In double DMA mode, only the Group 1 handshaking lines are used. DMA transfers use the DMA channel selected for Group 1 until a DMA terminal count is received; then the DMA channel selected for Group 2 is used for the Group 1 DMA transfers until a DMA terminal count for that channel is received. DMA transfers switch between the two channels. This configuration transfers data on one DMA channel and services data on another channel at the same time.

Bit	Name	Description (continued)
6	CNTINTEN	Counter Interrupt Enable Bit. When CNTINTEN is set, interrupts are enabled for Counter 3. An interrupt request is asserted when a rising edge is received on the output of Counter 3. This interrupt request is put on the interrupt line that is set for Group 2.
5	CNT2HSEN	Counter 2 Handshake Enable Bit. If CNT2HSEN is set, the output of Counter 2 generates the handshaking request for Group 2.
4	CNT1HSEN	Counter 1 Handshake Enable Bit. If CNT1HSEN is set, the output of Counter 1 generates the handshaking request for Group 1.
3	CNT2EN	Counter 2 Enable Bit. If CNT2EN is set and the IN2 line is high, Counter 2 is enabled for counting.
2	CNT1EN	Counter 1 Enable Bit. If CNT1EN is set and the IN2 line is high, Counter 1 is enabled for counting.
1	TCINTEN2	Group 2 Terminal Count Interrupt Enable Bit. When TCINTEN2 is set, interrupts are enabled for Group 2 DMA terminal counts. An interrupt request is asserted when a DMA terminal count for Group 2 is received.
0	TCINTEN1	Group 1 Terminal Count Interrupt Enable Bit. When TCINTEN1 is set, interrupts are enabled for Group 1 DMA terminal counts. An interrupt request is asserted when a DMA terminal count for Group 1 is received.

CFG4 Register

Revision C and later versions of the AT-DIO-32F have a CFG4 Register. This register contains four bits that set the leading pulse delay mode, Port D double-buffer mode, and version compatibility.

Address: Base address + 14 (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
0	0	0	0	LPULSE2	LPULSE1	DBLBUFD	REVC

Bit	Name	Description
15-4		Reserved Bits. These bits must be set to zero.
3	LPULSE2	Long Pulse Bit for Group 2. This bit selects the data-settling delay mode of the leading-edge pulse handshaking mode for Group 2. If this bit is set, the delay is added after the leading edge of the ACK pulse; therefore, the pulse width is lengthened. The delay is 0 to 700 nsec, depending on the settings of T2S <2..0>. If this bit is cleared, the delay is added before the leading edge of the pulse; therefore, the pulse width is fixed.
2	LPULSE1	Long Pulse Bit for Group 1. This bit selects the data-settling delay mode of the leading-edge pulse handshaking mode for Group 1. If this bit is set, the delay is added after the leading edge of the ACK pulse; therefore, the pulse width is lengthened. The delay is 0 to 700 nsec, depending on the settings of T1S <2..0>. If this bit is cleared, the delay is added before the leading edge of the pulse; therefore, the pulse width is fixed.

Bit	Name	Description (continued)
1	DBLBUFD	Port D Double-Buffer Enable Bit. If DBLBUFD is set, Port D is double-buffered. When Port D is configured as an output port, a write operation to the port loads data into the first buffer of the port. When a REQ2 is received, the data is transferred to the second buffer of the port, which dumps the data to the digital I/O connector. Double-buffering is usually used for pattern generation. If DBLBUFD is cleared, Port D is a single buffer; that is, data written to the port is immediately dumped to the digital I/O connector. When Port D is configured as an input port and DBLBUFD is set, an active level or edge of a REQ2 signal latches the data into the input buffer of Port D. A read operation reads the data in the buffer instead of the I/O connector. If DBLBUFD is cleared, the port is transparent; that is, a read operation reads the data on the I/O connector.
0	REVC	Version Compatibility Bit. Writing zero to this bit causes a Revision C board to function like a Revision B board. Therefore, the Revision C board is compatible with Revision B board software. Writing one to this bit adds the Revision C feature to the board. This bit is automatically cleared at startup.

STAT Register

The STAT Register contains 13 bits that reflect the handshaking state of Group 1 and Group 2.

Address: Base address + 00 (hex)

Type: Read-only

Word Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
DMACH	CNTINT	X	TRANS32	IN1	DMATC2	DMATC1	DRDY1

7	6	5	4	3	2	1	0
REQ1	ACK1	IN2	0	X	DRDY2	REQ2	ACK2

Bit	Name	Description
15	DMACH	Current DMA Channel Bit. If the DBLDMA bit in the CFG3 Register is set, the DMACH bit indicates which DMA channel is currently in use for the DMA operation for Group 1. If DMACH is cleared, the DMA channel selected for Group 1 is in use. If DMACH is set, the DMA channel selected for Group 2 is in use. The DBLDMA bit selects the double DMA mode, in which DMA transfers for Group 1 switch between the two DMA channels.
14	CNTINT	Counter 3 Interrupt Status Bit. This bit reflects the status of the Counter 3 interrupt. CNTINT is set whenever the CNTINTEN bit in the CFG3 Register is set and a rising edge on Counter 3 output is detected. CNTINT is cleared by writing to the CNTINTCLR Register.
13	X	Don't Care Bit.
12	TRANS32	16-Bit/32-Bit Transfer Indicator Bit. This bit is set when Group 1 and Group 2 are configured to transfer 32-bit data. Otherwise, TRANS32 is cleared. See the *32-Bit Transfers* section at the end of this chapter for more information about 32-bit transfer mode.
11	IN1	Extra Input Line 1 Bit. IN1 is pulled up to +5 V and is connected to input line IN1 on the digital I/O connector. IN1 can be used as an extra input line. IN1 is ANDed with CNT1EN and therefore can also be used as an external control signal to enable/disable Counter 1.

Bit	Name	Description (continued)
10	DMATC2	DMA2 Terminal Count Indicator Bit. This bit reflects the status of DMA for the Group 2 terminal count. If this bit and TCINTEN2 in the CFG3 Register are both set, then a DMA terminal count interrupt has occurred. This bit is cleared by writing to the DMACLR2 Register.
9	DMATC1	DMA1 Terminal Count Indicator Bit. This bit reflects the status of DMA for the Group 1 terminal count. If this bit and TCINTEN1 in the CFG3 Register are both set, then a DMA terminal count interrupt has occurred. This bit is cleared by writing to the DMACLR1 Register.
8	DRDY1	Group 1 Data Transfer Ready. When Group 1 is in read mode and enabled for handshaking, DRDY1 is set when data can be read at the I/O connector. When Group 1 is in write mode, DRDY1 is set when the external device is ready to receive the data.
7	REQ1	Group 1 Handshaking Request Status. REQ1 reflects the status of the Group 1 handshaking request line, as seen at the digital I/O connector.
6	ACK1	Group 1 Handshaking Acknowledge Status Bit. ACK1 reflects the status of the Group 1 handshaking acknowledge signal, as seen at the digital I/O connector.
5	IN2	Extra Input Line 2. IN2 is pulled up to +5 V and is connected to input line IN2 on the digital I/O connector. IN2 can be used as an extra input line. IN2 is ANDed with CNTEN2 and therefore can also be used as an external control signal to enable/disable Counter 2.
4	0	Revision C ID Bit. If this bit is cleared, the board is a Revision C board. If this bit is set, the board is a Revision B board.
3	X	Don't Care Bit.
2	DRDY2	Group 2 Data Transfer Ready. When Group 2 is in read mode and enabled for handshaking, DRDY2 is set when data can be read at the I/O connector. When Group 2 is in write mode, DRDY2 is set when the external device is ready to receive the data.
1	REQ2	Group 2 Handshaking Request Status. REQ2 reflects the status of the Group 2 handshaking request line, as seen at the digital I/O connector.
0	ACK2	Group 2 Handshaking Acknowledge Status Bit. ACK2 reflects the status of the Group 2 handshaking acknowledge signal, as seen at the digital I/O connector.

CNTINTCLR Register

Writing to the CNTINTCLR Register clears the interrupt request asserted when a rising edge on the Counter 3 output is detected.

Address: Base address + 0A (hex)

Type: Write-only

Word Size: 16-bit

Bit Map: Not applicable, no bits used

DMACLR1 Register

Writing to the DMACLR1 Register clears the interrupt request asserted when the DMA terminal count signal of Group 1 is detected.

Address: Base address + 0C (hex)

Type: Write-only

Word Size: 16-bit

Bit Map: Not applicable, no bits used

DMACLR2 Register

Writing to the DMACLR2 Register clears the interrupt request asserted when the DMA terminal count signal of Group 2 is detected.

Address: Base address + 0E (hex)

Type: Write-only

Word Size: 16-bit

Bit Map: Not applicable, no bits used

Digital I/O Port Register Group

The four registers making up the Digital I/O Register Group monitor and control the AT-DIO-32F digital I/O lines. There are four 8-bit ports on the AT-DIO-32F. These ports are grouped so that either 8-bit or 16-bit operations can be performed.

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

Port A Register

The Port A Register contains eight bits that connect to the digital I/O connector and can either be read from or written to.

Address: Base address + 06 (hex)

Type: Read-and-write

Word Size: 8-bit or 16-bit

Bit Map:

7	6	5	4	3	2	1	0
DIOA7	DIOA6	DIOA5	DIOA4	DIOA3	DIOA2	DIOA1	DIOA0

When Port A is configured as a write port, writing data to this register latches, or stores, the data into Port A and sends the data out on Port A on the digital I/O connector. Reading Port A when it is in write mode returns the value that is currently driven on the port. If the DBLBUFA bit in the CFG1 Register is set, the data written to Port A is stored in the buffer. However, this data is not dumped to the I/O connector until the active REQ1 level or pulse is received. During an active REQ1 level or pulse, the data is dumped to the I/O connector.

When Port A is configured as a read port, reading Port A returns the current data at Port A. If the DBLBUFA bit in the CFG1 Register is set, data is latched into Port A on the active level or active edge of REQ1. In level mode, data is latched during the active level of REQ1 until REQ1 is inactive. In pulse mode, data is latched on the active edge of REQ1 until the data is read from the port.

Either 8-bit or 16-bit transfers can be performed on Port A.

16-Bit Write or Read from Port A

Port B								Port A							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

8-Bit Write or Read from Port A

Port A							
7	6	5	4	3	2	1	0

Port B Register

The Port B Register contains eight bits that connect to the digital I/O connector and can either be read from or written to.

Address: Base address + 07 (hex)

Type: Read-and-write

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
DIOB7	DIOB6	DIOB5	DIOB4	DIOB3	DIOB2	DIOB1	DIOB0

When Port B is configured as a write port, writing data to this register latches the data into Port B and sends the data out on Port B. Reading Port B when it is in write mode returns the value that is currently driven on the port. If the DBLBUFB bit in the CFG3 Register is set, the data written to Port A is stored in the buffer. However, this data is not dumped to the I/O connector until the active REQ1 level or pulse is received. During an active REQ1 level or pulse, the data is dumped to the I/O connector.

When Port B is configured as a read port, reading Port B returns the current data at Port B. If the DBLBUFB bit in the CFG3 Register is set, data is latched into Port B on the active level or active edge of REQ1. In level mode, data is latched during the active level of REQ1 until REQ1 is inactive. In pulse mode, data is latched on the active edge of REQ1 until the data is read from the port.

Only 8-bit transfers can be performed on Port B.

8-Bit Write or Read from Port B

Port B							
7	6	5	4	3	2	1	0

Port C Register

The Port C Register contains eight bits that connect to the digital I/O connector and can either be read from or written to.

Address: Base address + 08 (hex)

Type: Read-and-write

Word Size: 8-bit or 16-bit

Bit Map:

7	6	5	4	3	2	1	0
DIOC7	DIOC6	DIOC5	DIOC4	DIOC3	DIOC2	DIOC1	DIOC0

When Port C is configured as a write port, writing data to this register latches the data into Port C and sends the data out on Port C. Reading Port C when it is in write mode returns the value that is currently driven on the port. If the DBLBUFC bit in the CFG2 Register is set, the data written to Port A is stored in the buffer. However, this data is not dumped to the I/O connector until the active REQ2 level or pulse is received. During an active REQ2 level or pulse, the data is dumped to the I/O connector.

When Port C is configured as a read port, reading Port C returns the current data at Port C. If the DBLBUFC bit in the CFG2 Register is set, data is latched into Port C on the active level or active edge of REQ1. In level mode, data is latched during the active level of REQ2 until REQ2 is inactive. In pulse mode, data is latched on the active edge of REQ2 until the data is read from the port.

Either 8-bit or 16-bit transfers can be performed on Port C.

16-Bit Write or Read from Port C

Port D								Port C							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

8-Bit Write or Read from Port C

Port C							
7	6	5	4	3	2	1	0

Port D Register

The Port D Register contains eight bits that connect to the digital I/O connector and can either be read from or written to.

Address: Base address + 09 (hex)

Type: Read-and-write

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
DIOD7	DIOD6	DIOD5	DIOD4	DIOD3	DIOD2	DIOD1	DIOD0

When Port D is configured as a write port, writing data to this register latches the data into Port D and sends the data out on Port D. Reading Port D when it is in write mode returns the value that is currently driven on the port. If the DBLBUFD bit in the CFG4 Register is set, the data written to Port A is stored in the buffer. However, this data is not dumped to the I/O connector until the active REQ2 level or pulse is received. During an active REQ2 level or pulse, the data is dumped to the I/O connector.

When Port D is configured as a read port, reading Port D returns the current data at Port D. If the DBLBUFD bit in the CFG4 Register is set, data is latched into Port D on the active level or active edge of REQ1. In level mode, data is latched during the active level of REQ2 until REQ2 is inactive. In pulse mode, data is latched on the active edge of REQ2 until the data is read from the port.

Only 8-bit transfers can be performed on Port D.

8-Bit Write or Read from Port D

Port D							
7	6	5	4	3	2	1	0

RTSI Bus Register Group

The two registers making up the RTSI Bus Register Group program the AT-DIO-32F RTSI switch for routing of signals on the RTSI bus trigger lines to and from AT-DIO-32F request (REQ) and acknowledge (ACK) signal lines.

Bit descriptions of the two registers making up the RTSI Bus Register Group are given on the following pages.

RTSISHFT Register

The RTSISHFT Register contains one bit, RSI, that is a serial input to the RTSI switch. RSI must be written to 56 times to load the internal 56-bit RTSI control register.

Address: Base address + 10 (hex)

Type: Write-only

Word Size: 8-bit

Bit Map:

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

Bit	Name	Description
7-1	X	Don't Care Bits.
0	RSI	RTSI Switch Serial Input. This bit is the serial input to the RTSI switch. Each time the RSI bit is written to, the value written is shifted into the RTSI switch internal 56-bit control register. The data in the control register routes information for switching signals to and from the RTSI bus trigger lines. The RSI bit must be written to 56 times to shift the 56 bits of routing data into the internal control register. See *Programming the RTSI Switch* later in this chapter for more information.

RTSISTRB Register

Writing to the RTSISTRB Register loads the contents of the RTSI Shift Register into the RTSI Switch Control Register, thereby updating the RTSI switch routing pattern. The RTSISTRB Register is written to after shifting the 56-bit routing pattern into the RTSISHFT Register.

Address: Base address + 12 (hex)

Type: Write-only

Word Size: 8-bit

Bit Map: Not applicable, no bits used

Counter Register Group

The four registers making up the Counter Register Group access the onboard 8254-2 Counter/Timer.

The 8254-2 contains three counters. Counters 1, 2, and 3 can be used for pattern generation, and Counter 3 can also be used to generate timed interrupts.

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

CNTR1 Register (REQ1 Generator)

The CNTR1 Register contains eight bits that are used to load a value into Counter 1 or to read back the value of Counter 1. The CNTR1 Register can be used as an 8-bit register or as a 16-bit register by two successive write/read operations.

Address: Base address + 18 (hex)

Type: Read-and-write

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
CNTR1B7	CNTR1B6	CNTR1B5	CNTR1B4	CNTR1B3	CNTR1B2	CNTR1B1	CNTR1B0

Bit	Name	Description
7-0	CNTR1B<7..0>	Counter 1 Load/Read Bits. Writing a data value to these bits loads the starting value into Counter 1. Reading these bits returns the current count of Counter 1 or latched data for Counter 1. If the Counter Latch command or the Read-Back command is used to latch the count or status of Counter 1, reading these bits returns the latched information. The latched data remains latched until it is read. If multiple Latch commands or Read-Back commands are issued before the latched data is read, only the data from the first Status Latch command and the first Counter Latch command are latched; all commands after the first are ignored. If 16-bit data is latched, the first read from this register returns the least significant byte, and the second read returns the most significant byte. If status and count information are both latched, the first read from this register returns the status byte, and the next one read for 8-bit mode, or two reads for 16-bit mode, return the count bytes, regardless of the order in which the information was latched.

CNTR2 Register (REQ2 Generator)

The CNTR2 Register contains eight bits that are used to load a value into Counter 2 or to read back the value of Counter 2. The CNTR2 Register can be used as an 8-bit register or as a 16-bit register by two successive write/read operations.

Address: Base address + 1A (hex)

Type: Read-and-write

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
CNTR2B7	CNTR2B6	CNTR2B5	CNTR2B4	CNTR2B3	CNTR2B2	CNTR2B1	CNTR2B0

Bit	Name	Description
7-0	CNTR2B<7..0>	Counter 2 Load/Read Bits. Writing a data value to these bits loads the starting value into Counter 2. Reading these bits returns the current count of Counter 2 or latched data for Counter 2. If the Counter Latch command or the Read-Back command is used to latch the count or status of Counter 2, reading these bits returns the latched information. The latched data remains latched until it is read. If multiple Latch commands or Read-Back commands are issued before the latched data is read, only the data from the first Status Latch command and the first Counter Latch command are latched; all commands after the first are ignored. If 16-bit data is latched, the first read from this register returns the least significant byte, and the second read returns the most significant byte. If status and count information are both latched, the first read to this register returns the status byte, and the next one read for 8-bit mode, or two reads for 16-bit mode, returns the count bytes, regardless of the order in which the information was latched.

CNTR3 Register (Timebase Generator)

The CNTR3 Register contains eight bits that are used to load a value into Counter 3 or to read back the value of Counter 3. The CNTR3 Register can be used as an 8-bit register or as a 16-bit register by two successive write/read operations.

Address: Base address + 1C (hex)

Type: Read-and-write

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
CNTR3B7	CNTR3B6	CNTR3B5	CNTR3B4	CNTR3B3	CNTR3B2	CNTR3B1	CNTR3B0

Bit	Name	Description
7-0	CNTR3B<7..0>	Counter 3 Load/Read Bits. Writing a data value to these bits loads the starting value into Counter 3. Reading these bits returns the current count of Counter 3 or latched data for Counter 3. If the Counter Latch command or the Read-Back command is used to latch the count or status of Counter 3, reading these bits returns the latched information. The latched data remains latched until it is read. If multiple Latch commands or Read-Back commands are issued before the latched data is read, only the data from the first Status Latch command and the first Counter Latch command are latched; all commands after the first are ignored. If 16-bit data is latched, the first read from this register returns the least significant byte, and the second read returns the most significant byte. If status and count information are both latched, the first read to this register returns the status byte, and the next one read for 8-bit mode, or two reads for 16-bit mode, return the count bytes, regardless of the order in which the information was latched.

CNTRCMD Register

The CNTRCMD Register contains eight bits that determine the counter selection, counter size, counting format, and operation mode.

Address: Base address + 1E (hex)

Type: Write-only

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
CNTRSEL1	CNTRSEL0	RWSEL1	RWSEL0	MODESEL2	MODESEL1	MODESEL0	BCDSEL

Bit	Name	Description
7-6	CNTRSEL<1..0>	Counter Select Bits. These bits select the counter on which the command operates.

CNTRSEL1	CNTRSEL0	Operation
0	0	Select Counter 1
0	1	Select Counter 2
1	0	Select Counter 3
1	1	Read-Back command

5-4 RWSEL<1..0> Read/Write Select Bits.
These bits select data written to or read from a counter, or these bits send a Counter Latch command.

RWSEL1	RWSEL0	Operation
0	0	Counter Latch command
0	1	Read and write least significant byte only
1	0	Read and write most significant byte only
1	1	Read and write least significant byte then most significant byte

Bit	Name	Description (continued)
5-4	RWSEL<1..0>	The Counter Latch command latches the current count of the register selected by CNTRSEL1 and CNTRSEL0. The next read from the selected counter returns the latched data.
3-1	MODESEL<2..0>	Counter Mode Select Bits. These bits select the counting mode of the selected counter. The following table lists six available modes and the corresponding bit settings. Refer to Appendix D, *Intel Data Sheet*, for additional information.

MODESEL2	MODESEL1	MODESEL0	Mode
0	0	0	Mode 0 – Interrupt on Terminal Count
0	0	1	Mode 1 – Hardware Retriggerable One Shot
0	1	0	Mode 2 – Rate Generator
0	1	1	Mode 3 – Square Wave Mode
1	0	0	Mode 4 – Software Retriggerable Strobe
1	0	1	Mode 5 – Hardware Retriggerable Strobe

0	BCDSEL	Binary Coded Decimal Select Bit. If BCDSEL is set, the selected counter keeps count in BCD. If BCDSEL is cleared, the selected counter keeps count in 16-bit binary.

Read-Back Command

When bits 7 and 6 (CNTRSEL1 and CNTRSEL0) are 1, the CNTRCMD Register can be used to execute the Read-Back command. With the Read-Back command, the current totals of multiple counters can be latched in one command. The Read-Back command also can latch the status of selected counters. The control word format used for the Read-Back command is as follows:

7	6	5	4	3	2	1	0
CNTRSEL1	CNTRSEL0	COUNT*	STATUS*	CNTR3	CNTR2	CNTR1	0

Bit	**Name**	**Description**
7-6	CNTRSEL<1..0>	Counter Select Bits. Both bits *must* be one for the Read-Back command to be used.
5	COUNT*	Read-Back Count Command. If COUNT* is cleared, the current count in each of the selected counters is latched. The next read from the selected counter returns the latched data.
4	STATUS*	Read-Back Status Command. If STATUS* is cleared, the current status in each of the selected counters is latched. The next read from the selected counter returns the latched data.
3-1	CNTR<3..1>	Counter Select Bits for Read-Back Command. These bits select the counters for the Read-Back command; that is, if CNTR3 and CNTR1 are set, the Read-Back command latches data for Counter 3 and Counter 1.
0	0	Zero Bit. This bit must be zero for proper operation of the AT-DIO-32F.

Status Byte. If the STATUS* bit is zero in the Read-Back command, status information for the selected counters is latched. The status byte format is as follows:

7	6	5	4	3	2	1	0
OUT	NULL	RW1	RW0	MODE2	MODE1	MODE0	BCD

Bit	Name	Description
7	OUT	Counter Output. The OUT bit reflects the current status of the counter output.
6	NULL	Last Count Written Status. If NULL is zero, the last count written to the selected counter has been loaded into the counter. If NULL is set, the last count written to the counter has not been loaded.
5-4	RW<1..0>	RWSEL1 and RWSEL0 Status. The RW1 and RW0 bits reflect the status of the RWSEL1 and RWSEL0 bits of the selected counter.
3-1	MODE<2..0>	MODE2, MODE1, and MODE0 Status. The MODE2, MODE1, and MODE0 bits reflect the state of the MODESEL2, MODESEL1, and MODESEL0 bits of the selected counter.
0	BCD	Binary Coded Decimal Select (BCDSEL) Status. The BCD bit reflects the status of the BCDSEL bit of the selected counter.

Refer to Appendix D, *Intel Data Sheet*, for more information on programming the counters.

Programming Considerations

The AT-DIO-32F has four 8-bit digital I/O ports. These ports are organized into two groups: Group 1 contains Ports A and B, and Group 2 contains Ports C and D. The AT-DIO-32F can also operate in two modes: Mode 0 and Mode 1. Mode 0 is basic I/O where each port can be configured as a read or write port. Mode 1 is strobed I/O. Two handshaking lines are used to synchronize the sending and receiving of data for each port. Mode 1 can also act as a pattern generator by using the onboard counters or by using a signal routed across the RTSI bus interface from another AT Series board. The following paragraphs discuss Mode 0, Mode 1, and pattern generation.

Initializing the AT-DIO-32F Board

The AT-DIO-32F hardware must be initialized in order for the AT-DIO-32F circuitry to operate properly. To initialize the AT-DIO-32F hardware, complete the following steps:

1. Write hex 0100 to the CFG1 Register.
2. Write hex 0100 to the CFG2 Register.
3. Write 0000 to the CFG3 Register.
4. Write 0000 to the CFG1 Register.
5. Write 0000 to the CFG2 Register.
6. Write 0001 to the CFG4 Register (only for Revision D and newer versions of the board).
7. Write hex 14 to the CNTRCMD Register.
8. Write hex 54 to the CNTRCMD Register.
9. Write 0 to the DMACLR1 Register.
10. Write 0 to the DMACLR2 Register.
11. Write 0 to the CNTINTCLR Register.

This sequence leaves the AT-DIO-32F circuitry in the following state:

- All digital I/O ports are in input mode.
- All interrupts are cleared and disabled.
- The outputs of Counter 0 and Counter 1 are high.
- The handshaking circuitry are cleared.

Mode 0 Programming

Whenever the AT-DIO-32F is started up, each digital I/O port is configured as a read port. The status of the lines connected to each port can be determined by reading that port. To configure a port as a write port, the port's corresponding WRITE bit must be set. If a write port is read, the result is the data currently driven on that port. Any value written to a write port is then driven on the corresponding digital I/O lines. When the WRITE bit for a port is cleared, the port is once again configured as a read port.

For example, to use Ports A and B as write ports, write hex 2400 to the CFG3 Register. Clearing bit 10 and bit 13 of the CFG3 Register configures Ports A and B as read ports. Similarly, writing hex 4800 to the CFG3 Register configures Ports C and D as write ports, and clearing bit 11 and bit 14 of the CFG3 Register configures Ports C and D as read ports. Each port can be assigned a direction independently.

Mode 1 Programming

Each group of ports has its own set of handshaking lines. Group 1 (Ports A and B) handshaking lines are REQ1 and ACK1. Group 2 (Ports C and D) handshaking lines are REQ2 and ACK2. The individual ports in a group must be enabled for handshaking within each group by setting DIOAEN, DIOBEN, DIOCEN, and DIODEN. For 8-bit transfers, only one port should be enabled for handshaking in a group. For 16-bit transfers, both ports in a group should be enabled.

The direction of handshaking for each group is determined by the corresponding WRITE bit. If data is to be driven by the group, the corresponding WRITE bit must be set. If a write port is read, the result is the data currently driven on that port. If data is to be received by the group, the corresponding WRITE bit must be cleared.

Any ports not enabled for handshaking can be used for Mode 0 operations, that is, for reading additional status data or for driving additional control lines. For example, if Port B is enabled for handshaking and is configured as a read port, Port A can be used to read or write data lines in Mode 0 operations.

For example, to set up Group 1 as a 16-bit handshaking write port with REQ1 and ACK1 as negative logic, follow these steps:

1. Write hex 0644 to the CFG1 Register to set up the handshaking mode.
2. Write hex 2400 to the CFG3 Register to set up the group as a write port (if the port is a read port, skip this step).

Reading the STAT Register returns the status of REQ1, ACK1, and the DRDY1 bit.

The polarity of the handshaking lines for each group is programmable. If the external signal connected to REQ1 or REQ2 is active low, INVRQ1 or INVRQ2, respectively, should be set. If the external signal connected to ACK1 or ACK2 is active low, INVACK1 or INVACK2, respectively, should be set.

The status of REQ1, REQ2, ACK1, and ACK2 as seen at the digital I/O connector can be read from the STAT Register. If handshaking is not enabled, the ACK lines can be used as extra

output lines on the digital I/O connector. These lines can be controlled by setting and clearing the SETACK1 and SETACK2 bits.

Each handshaking group has an LRESET bit. Setting and then clearing this bit resets the handshaking circuitry for that group. To perform another handshaking write operation after an LRESET, the appropriate WRITE bit must first be cleared and then set to logic high to set up the circuitry for another transfer.

For example, to set up Port C as an 8-bit handshaking write port with REQ2 and ACK2 as positive logic, follow these steps:

1. Write hex 0200 to the CFG2 Register to set up the Port C handshaking mode.
2. Write hex 0800 to the CFG3 Register to set up Port C as a write port (if the port is a read port, do not set bit 11 of the CFG3 Register).
3. Write hex 0300 to the CFG2 Register to reset Group 2 handshaking status.
4. Write hex 0200 to the CFG2 Register.
5. Write hex 0000 to the CFG3 Register to clear the WRITEC bit after the LRESET.
6. Write hex 0800 to the CFG3 Register to set the WRITEC bit again after the LRESET.

Reading the STAT Register returns the status of REQ2, ACK2, and the DRDY2 bit.

Handshaking Modes

The AT-DIO-32F can be programmed for one of three types of handshake timing: level mode, leading edge mode, and trailing edge mode. These modes are described in detail in the following pages. For detailed timing information, refer to Chapter 2, *Configuration and Installation*.

In the following timing diagrams, REQ and ACK are shown as positive logic; that is, INVRQ and INVACK are cleared. When the WRITE bit is set, an internal pulse called GO initializes a write transfer. DRDY is the data transfer ready bit as seen in the STAT Register. WR and RD are the write and read pulses from the PC. TDELAY is the programmable data-settling delay, which is set either in the CFG1 Register or the CFG2 Register. This delay is between 0 and 700 nsec.

Level Mode

In level mode, ACK remains active until another REQ is received. A handshaking group is in level mode when its PULSE and EDGE bits are cleared. Figures 4-1 and 4-2 show active high level ACK and REQ signals.

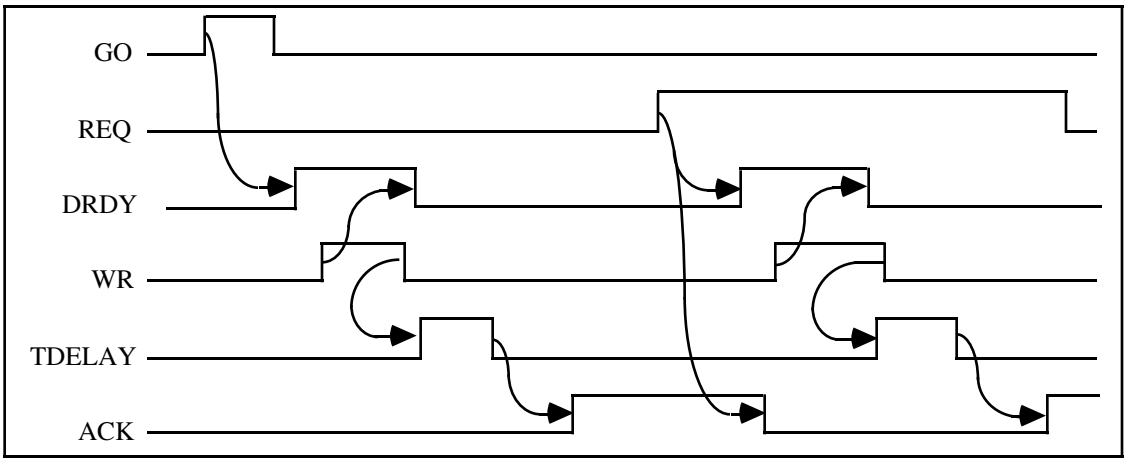


Figure 4-1. Level Mode Write Handshake Timing

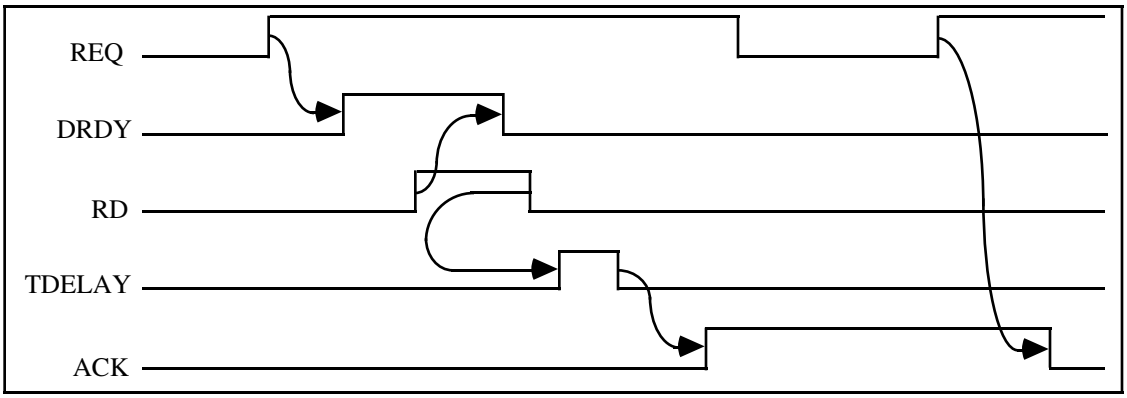


Figure 4-2. Level Mode Read Handshake Timing

Leading Edge Mode

In leading edge mode, REQ and ACK are viewed as pulses that are active on the leading edge of the pulse. A handshaking group is in leading edge mode when its PULSE bit is set and its EDGE bit is cleared. Figures 4-3, 4-4, and 4-5 show the timing diagrams for leading edge mode. For detailed timing information, refer to Chapter 2, *Configuration and Installation*.

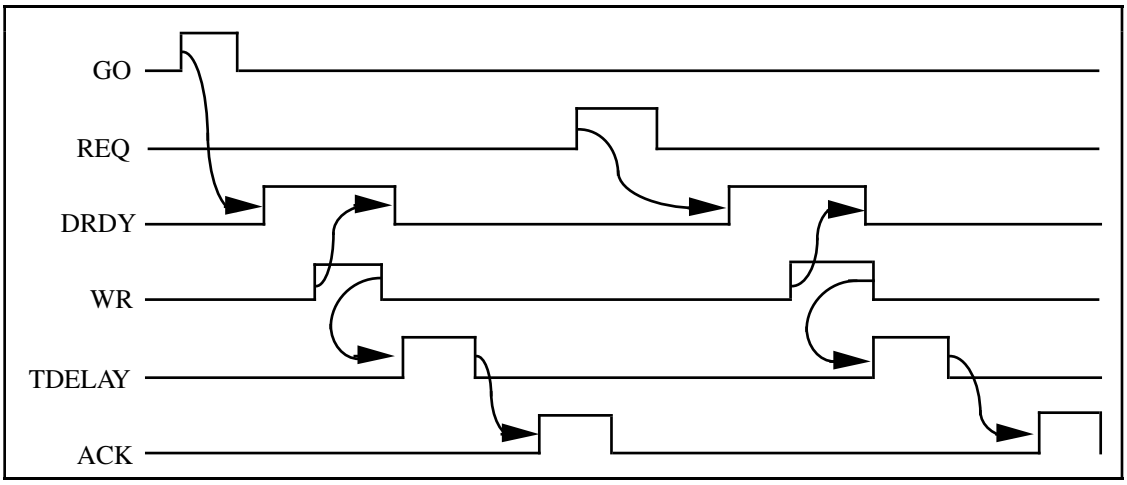


Figure 4-3. Leading Edge Mode Write Handshake Timing (LPULSE*x* cleared)

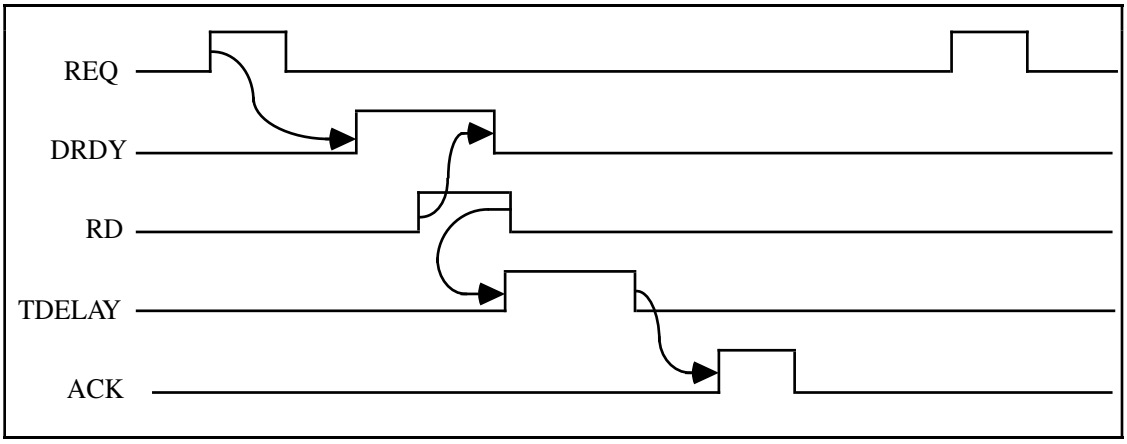


Figure 4-4. Leading Edge Mode Read Handshake Timing (LPULSE*x* cleared)

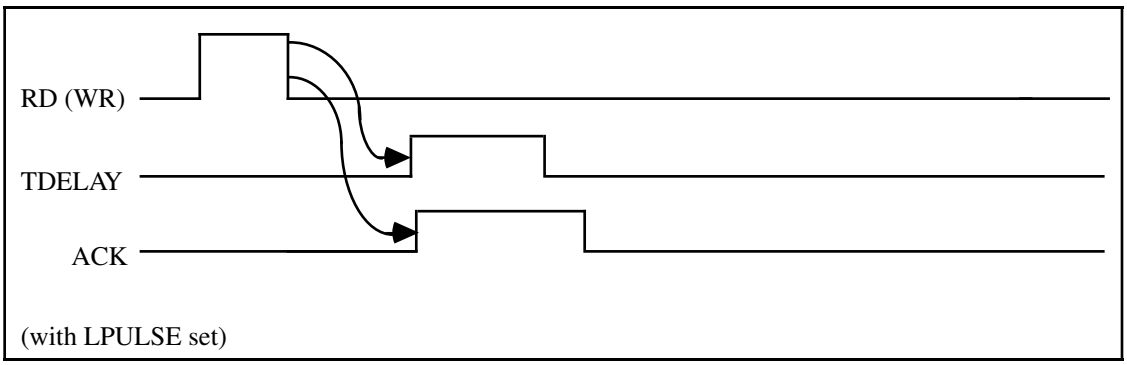


Figure 4-5. Leading Edge Mode Read/Write ACK Pulse Width with LPULSE*x* of CFG4 Set

Trailing Edge Mode

In trailing edge mode, REQ and ACK are treated as pulses that are active on the trailing edge of the pulse. A handshaking group is in trailing edge mode when its PULSE and EDGE bits are set. Figures 4-6 and 4-7 show the timing diagrams for trailing edge mode. For detailed timing information, refer to Chapter 2, *Configuration and Installation*.

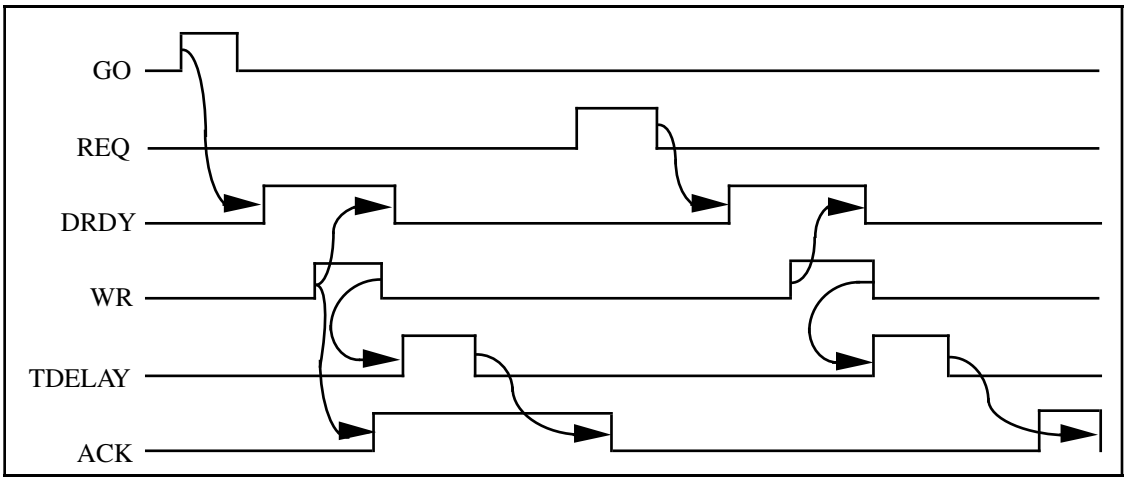


Figure 4-6. Trailing Edge Mode Write Handshake Timing

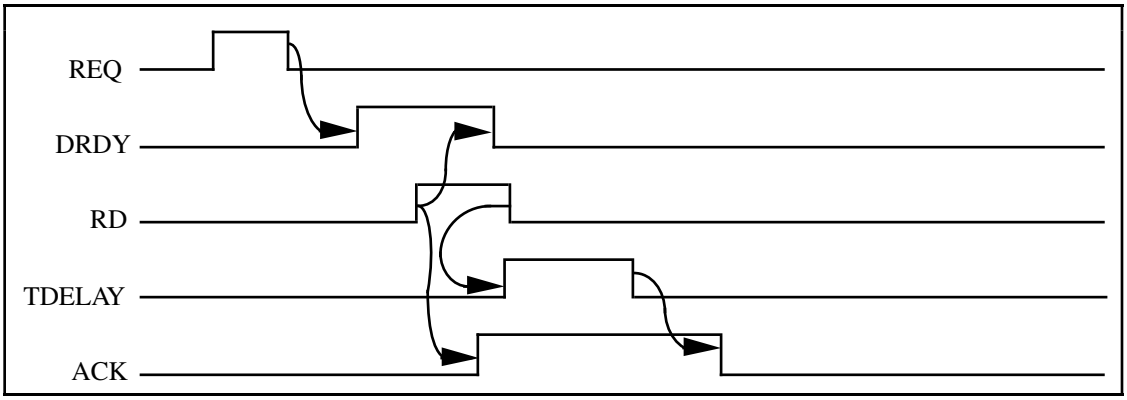


Figure 4-7. Trailing Edge Mode Read Handshake Timing

Data-Settling Delay

Each handshaking group has a set of bits to select a data-settling time (TDELAY) between each transfer. For short cable lengths, this delay is usually zero. For longer cable lengths, noisy environments, or special handshaking specifications, a longer data-settling time may be required. This delay can also be used to lengthen the ACK pulse. Table 4-2 shows these data-settling time settings.

Table 4-2. CFG1 Data-Settling Time Settings

Group 1				Group 2			
T1S2	**T1S1**	**T1S0**	**Delay (nsec)**	**T2S2**	**T2S1**	**T2S0**	**Delay (nsec)**
0	0	0	0	0	0	0	0
0	0	1	100	0	0	1	100
0	1	0	200	0	1	0	200
0	1	1	300	0	1	1	300
1	0	0	400	1	0	0	500
1	1	0	600	1	1	0	600
1	1	1	700	1	1	1	700

Programmed I/O Transfers

To perform programmed I/O handshaking, the DRDY bit in the STAT Register must be polled by software. When the bit is set, the software can then write or read data to or from the digital I/O port. First, configure the handshaking group to be used and enable the ports to be used for handshaking. Be sure that the direction of each group is programmed correctly. In write mode, the external device is ready to receive data when the DRDY bit in the STAT Register is set. The program waits until DRDY is set, then writes the data to the appropriate output port. The circuitry handshakes the data and sets DRDY again when the external device is ready for another transfer. In write mode, set the WRITE bit again after a reset of that group.

To perform Group 1 16-bit write-programmed I/O transfers using positive level handshaking, follow these steps:

1. Write hex 0600 to the CFG1 Register to set up the handshaking mode.
2. Write hex 2400 to the CFG3 Register to set up the group as a write port.
3. Wait until DRDY1 becomes set (poll the STAT Register).
4. Write the 16-bit data to Port A.
5. Wait until DRDY1 becomes set, then write the next data value to Port A.
6. Repeat step 5 until all data has been written to Port A.

In read mode, data is available for reading when the DRDY bit is set. The program waits until the DRDY bit is set, then reads the data from the enabled port. Each time the DRDY bit is set, another data value has been latched and can be read.

To perform read-programmed I/O transfers, follow these steps:

1. Write hex 0600 to the CFG1 Register to set up the handshaking mode.

2. Wait until DRDY1 becomes set (poll the STAT Register).

3. Read the 16-bit data from Port A.

4. Wait until DRDY1 becomes set, then read the next data value from Port A.

5. Repeat step 4 until all data has been read from Port A.

The digital I/O connector has two extra input lines and two extra output lines. The status of the input lines IN1 and IN2 can be read from the STAT Register. The output lines OUT1 and OUT2 are controlled by bits in the CFG1 and CFG2 Registers. These lines are useful for reading and sending status or control information needed in addition to the handshaking lines. These extra lines are independent of the group designations.

Input Data Latch

When an I/O port is configured as an input port (read mode) and the corresponding DBLBUF bit is cleared, reading the port returns the current data on the I/O lines, whether or not handshaking is enabled.

When an I/O port is configured as a double-buffered input port, that is, the corresponding DBLBUF bit is set, and the handshaking is enabled, the REQ signal latches the data into the port.

- Level Mode–In either the active high or active low level mode, data is latched into the port on the leading edge of REQ until the REQ is inactive.

- Pulse Mode–The active edge (leading edge or trailing edge) of REQ latches data into the port, until the data in the input buffer is read.

Interrupt Handling

Five conditions can generate an interrupt: DRDY1 set, DRDY2 set, Group 1 DMA terminal count received, Group 2 DMA terminal count received, and a rising edge on Counter 3 received. Each of these conditions has an interrupt enable bit in a CFG Register and an interrupt status bit in the STAT Register. The interrupt condition is enabled to generate interrupts by setting the appropriate enable bit in the CFG Registers. When the interrupt service routine is entered, the appropriate status bits of the STAT Register reflect which interrupt requests are active. When the interrupt condition has been serviced, writing to an interrupt clear register may be necessary to clear the interrupt status bit. Table 4-3 lists all of the interrupt conditions and status.

Table 4-3. Interrupt Condition and Status

Interrupt Enable Bit in the CFG Register	Status Bit in the STAT Register	Type of Interrupt	To Clear the Status Bit
INTEN1	DRDY1	Data ready of Group 1	Read/write data
INTEN2	DRDY2	Data ready of Group 2	Read/write data
TCINTEN1	DMATC1	DMA terminal count of Group 1	Write to DMACLR1 Register
TCINTEN2	DMATC2	DMA terminal count of Group 2	Write to DMACLR2 Register
CNTINTEN	CNTINT	Counter 3 interrupt	Write to CNTINTCLR Register
Note: Counter 3 interrupts use the interrupt line for Group 2.			

Example: Interrupt Generation on DRDY1

Instead of polling the DRDY1 bit, interrupts can be used to wait for the DRDY1 condition. The interrupt service routine can then write or read the data to or from Group 1. To use interrupts for this task, follow these steps:

1. Set up Group 1 with the desired handshaking mode and direction.
2. Set the INTEN1 bit in the CFG1 Register to enable interrupts on DRDY1.
3. When DRDY1 is set, an interrupt request is asserted and software control enters the interrupt service routine.

   In the interrupt service routine, follow these steps:

   a. Check the DRDY1 bit in the STAT Register to verify that the current interrupt was generated by a DRDY1 condition.

   b. Write or read data to or from Group 1.

   When DRDY1 becomes set again, software control again jumps to the interrupt service routine.

Example: Interrupt Generation on DRDY2

Instead of polling the DRDY2 bit, interrupts can be used to wait for the DRDY2 condition. The interrupt service routine can then write or read the data to or from Group 2. To use interrupts for this task, follow these steps:

1. Set up Group 2 with the desired handshaking mode and direction.
2. Set the INTEN2 bit in the CFG2 Register to enable interrupts on DRDY2.
3. When DRDY2 is set, an interrupt request is asserted and software control enters the interrupt service routine.

   In the interrupt service routine, follow these steps:

   a. Check the DRDY2 bit in the STAT Register to verify that the current interrupt was generated by a DRDY2 condition.

   b. Write or read data to or from Group 2.

   When DRDY2 becomes set again, software control again jumps to the interrupt service routine.

Example: Interrupt Generation on Group 1 DMA Terminal Count

The DMA terminal count signals the end of the current DMA transfer. To use the Group 1 DMA terminal count interrupt service routine, follow these steps:

1. Set up Group 1 with the desired handshaking mode and direction.
2. Set the TCINTEN1 bit in the CFG3 Register to enable interrupts on Group 1 DMA terminal counts.
3. Enable DMA for Group 1 transfers.
4. Program the PC DMA controller.
5. When a Group 1 DMA terminal count is received, an interrupt request is asserted and software control enters the interrupt service routine.

   In the interrupt service routine, follow these steps:

   a. Check the DMATC1 bit in the STAT Register to verify that the current interrupt was generated by a Group 1 DMA terminal count condition.

   b. Perform the desired work.

   c. Write to the DMACLR1 Register to acknowledge that the Group 1 DMA terminal count interrupt condition has been serviced.

   When another Group 1 DMA terminal count is received, software control again jumps to the interrupt service routine.

Example: Interrupt Generation on Group 2 DMA Terminal Count

To use the Group 2 DMA terminal count interrupt service routine, follow these steps:

1. Set up Group 2 with the desired handshaking mode and direction.
2. Set the TCINTEN2 bit in the CFG3 Register to enable interrupts on Group 2 DMA terminal counts.
3. Enable DMA for Group 2 transfers.
4. Program the PC DMA controller.
5. When a Group 2 DMA terminal count is received, an interrupt request is asserted and software control enters the interrupt service routine.

   In the interrupt service routine, follow these steps:

   a. Check the DMATC2 bit in the STAT Register to verify that the current interrupt was generated by a Group 2 DMA terminal count condition.

   b. Perform the desired work.

   c. Write to the DMACLR2 Register to acknowledge that the Group 2 DMA terminal count interrupt condition has been serviced.

   When another Group 2 DMA terminal count is received, software control again jumps to the interrupt service routine.

Example: Interrupt Generation on Counter 3

Counter 3 can be programmed to generate pulses at a specified time interval. A rising edge on Counter 3 generates an interrupt request.

To use an interrupt service routine, follow these steps:

1. Set up rate generation mode for Counter 3.
2. Set up Counter 3 for the desired time interval between data patterns.
3. Set the CNTINTEN bit in the CFG3 Register to enable interrupts on Counter 3.
4. When a rising edge pulse on Counter 3 is received, an interrupt request is asserted and software control enters the interrupt service routine.

   In the interrupt service routine, follow these steps:

   a. Check the CNTINT bit in the STAT Register to verify that the current interrupt was generated by a Counter 3 pulse.

   b. Perform the desired work.

   c. Write to the CNTINTCLR Register to acknowledge that the Counter 3 interrupt condition has been serviced.

   When another rising edge pulse on Counter 3 is received, software control again jumps to the interrupt service routine.

DMA Transfers

DMA increases transfer rates when handshaking data is transferred to or from the PC memory. Each handshaking group can be assigned a separate DMA channel by setting the jumper on W1 (see Chapter 2, *Configuration and Installation*, for details).

There are two DMA modes: single-channel DMA for Groups 1 and 2 and double-channel DMA using Group 1 handshaking. In single DMA mode, enable DMA transfers for each group by setting DMAEN1 and DMAEN2 in the CFG1 and CFG2 Registers for Group 1 and Group 2, respectively. The DMA request is asserted when the DRDY bit for the enabled group is set. The DMA controller sends a terminal count when the value in its Terminal Count Register changes from hex 0000 to hex FFFF. See the *Interrupt Handling* section earlier in this chapter for information about generating an interrupt on the DMA terminal count. Refer to the *IBM Personal Computer AT Technical Reference* manual for additional information about programming the DMA controller.

In double DMA mode (DBLDMA bit set in the CFG3 Register), only Group 1 handshaking lines are used for 16-bit DMA transfer, but both DMA channels should be enabled. DMA transfers use the DMA channel for Group 1 until a DMA terminal count for Group 1 is received, then the DMA transfers switch to the DMA channel for Group 2. When a DMA terminal count is received for Group 2, the DMA transfers switch back to the Group 1 DMA channel. While one DMA channel is acquiring data, the other channel can service the acquired data. The DMACH bit in the STAT Register indicates which group's DMA channel is currently in use. If DMACH is cleared, the DMA channel for Group 1 is currently in use; if DMACH is set, the DMA channel for Group 2 is currently in use. If the DMA controller is programmed for auto-reinitialize mode, the two DMA channels are continuously served in turn.

32-Bit Transfers

The four digital I/O ports are divided into two 16-bit groups: Group 1 and Group 2. Either 8-bit or 16-bit operations can be performed on these groups. When the TRANS32 bit is set in the CFG3 Register, 32-bit operations can also be performed. For 32-bit transfer mode, Groups 1 and 2 must be enabled for handshaking (DIOAEN, DIOBEN, DIOCEN, and DIODEN must be set). The two groups should be programmed for identical configurations, with the same PULSE, EDGE, LPULSE, and TS (TDELAY) values. Requests should not begin on the REQ1 line until both groups are fully configured.

Configured in this way, the AT-DIO-32F can transfer data to or from all 32 of its data lines simultaneously, using only the ACK1 and REQ1 handshaking lines. ACK2 and REQ2 can be ignored. When a REQ1 is received, both DRDY1 and DRDY2 are set. When data has been written or read for both Group 1 and Group 2, the ACK1 line asserts. Two write or read operations are required for each 32-bit transfer: one to Group 1 for the lower word, and one to Group 2 for the upper word of the 32-bit datum. Asserting both LRESET1 and LRESET2 resets both groups and reinitializes the 32-bit transfer.

DMA can be used in conjunction with the 32-bit transfer mode for high-speed 32-bit transfers. The Group 1 data transfer and the Group 2 data transfer both use the DMA channel selected for Group 1. The count written to the DMA Terminal Count Register of the DMA controller should be twice the selected number of 32-bit transfers to be performed. If double DMA mode is used, the DMA channel used for the transfers swaps after each terminal count.

Pattern Generation Using Onboard Counters

There are three onboard counters on the AT-DIO-32F, each designated for a specific purpose. The output of Counter 1 is connected to the Group 1 handshaking request line REQ1. The output of Counter 2 is connected to the Group 2 handshaking request line REQ2. The output of Counter 3 can be used as the counting source for Counter 1 or 2. The counting source for Counter 3 is a 2-MHz square wave. The output of Counter 3 can also be used to generate interrupts.

Counters 1 and 2 have counting enable bits that connect to the gate input of the counter. The output of Counter 3 is always enabled. The CNT1EN bit in the CFG3 Register enables Counter 1 for counting; the CNT2EN bit in the CFG3 Register enables Counter 2 for counting; and both counters can be enabled or disabled by external signals IN1 and IN2, respectively. When IN1 and IN2 are used to control the counters, CNT1EN and CNT2EN must be set so that an active high-level signal on IN1 can enable Counter 1 and an active high-level signal on IN2 can enable Counter 2.

Counters 1 and 2 can be used for pattern generation. Counter 1 implements pattern generation for Group 1, and Counter 2 implements pattern generation for Group 2. When the CNT1HSEN bit in the CFG3 Register is set, the output of Counter 1 controls the REQ1 line. When the CNT2HSEN bit in the CFG3 Register is set, the output of Counter 2 controls the REQ2 line. If a counter is used for pattern generation, the corresponding REQ line on the digital I/O connector should be disconnected or in a high-impedance state.

Table 4-4. Counter 3 Programmable Frequency Output

Data Written to CNTR3 (hex)	Counter 3 Output Frequency
FFFF	30.0 Hz
8000	61.0 Hz
4000	122.0 Hz
2000	244.0 Hz
FFF	488.0 Hz
A00	780.0 Hz
800	976.0 Hz
400	1.95 kHz
FF	7.8 kHz
A0	12.5 kHz
80	15.62 kHz
60	20.83 kHz
40	31.25 kHz
20	62.5 kHz
10	125.0 kHz
8	250.0 kHz
4	500.0 kHz
2	1.0 MHz

The CNT1SRC and CNT2SRC bits in the CFG3 Register select the counting source for Counter 1 and Counter 2, respectively. If the CNT1SRC bit is cleared, the counting source for Counter 1

is a 10-MHz square wave. If the CNT1SRC bit is set, the counting source for Counter 1 is the output of Counter 3. The same applies for Counter 2. The output of Counter 3 is programmed as follows:

1. Set up Counter 3 for square wave generation by writing hex 96 to the CNTRCMD Register for an 8-bit count or hex B6 for a 16-bit count.

2. Write the count to the CNTR3 Register (see Table 4-4). If the count is a 16-bit value, write the least significant byte first, then the most significant.

The following equation is used to calculate the output frequency of Counter 3:

Counter 3 Output Frequency = 2000000 / data written to counter

Counters 1 and 2 can use either a 10-MHz clock or the output of Counter 3 as the counting source. If the counting source is the 10-MHz square wave, then the following equation is used to calculate the output frequency of the counter:

Counter 1, 2 Output Frequency = 10000000 / data written to counter

If the counting source is the output of Counter 3, then the following equation is used for calculating the output frequency of the counter:

Counter 1, 2 Output Frequency = Counter 3 output frequency / data written to counter

Table 4-5. Counters 1 and 2 Programmable Frequency Output (Source = 10 MHz)

Data Written to CNTR1 or CNTR2 (hex)	Counter Output Frequency	Pattern Interval
FFFF	152.0 Hz	6.58 msec
8000	305.0 Hz	3.28 msec
4000	610.0 Hz	1.67 msec
2000	1.22 kHz	820.0 µsec
FFF	2.44 kHz	410.0 µsec
A00	3.9 kHz	260.0 µsec
800	4.88 kHz	200.0 µsec
400	9.76 kHz	100.0 µsec
FF	39.0 kHz	25.6 µsec
A0	62.5 kHz	16.0 µsec
80	78.0 kHz	12.8 µsec
60	104.0 kHz	9.6 µsec
40	156.0 kHz	6.4 µsec
20	313.0 kHz	3.2 µsec
10	625.0 kHz	1.6 µsec
A	1.0 MHz	1.0 µsec
5	2.0 MHz	500.0 nsec
2	5.0 MHz	200.0 nsec

The ports used for pattern generation should be configured for double-buffered outputs. Ports A and B are double-buffered when the DBLBUFA and DBLBUFB bits in the CFG1 Register are set. Ports C and D are double-buffered when the DBLBUFC and DBLBUFD bits in the CFG2 Register are set. Double-buffering means that the port has two buffers: a write buffer and an output buffer. Writing to the port loads the write buffer. Contents of the write buffer are loaded into the output buffer when the corresponding REQ is active. The output buffer is connected to the digital I/O connector.

After a double-buffered port has been configured for handshaking, the first data pattern should be written to the port. This write loads the data pattern into the write buffer of the port. When a REQ is received, the data in the write buffer is transferred to the output buffer of the port, which is connected to the digital I/O connector. The trailing edge of REQ also sets the DRDY bit for the group. A polling routine, interrupt routine, or DMA can be used to detect the DRDY condition and then write the data pattern to the port.

The double-buffered configuration sends the data pattern as soon as a REQ is received. In the normal mode of operation, the data pattern is not dumped to the digital I/O connector until the DRDY bit is detected and the data is written to the port.

The output of the counters is an active low pulse. The handshaking mode for pattern generation must be set to active low REQ, trailing edge mode (INVREQ, PULSE, and EDGE are set). Therefore, the latched data can be driven to the output lines during the active REQ pulse, and new patterns can be written and latched after the active REQ duration.

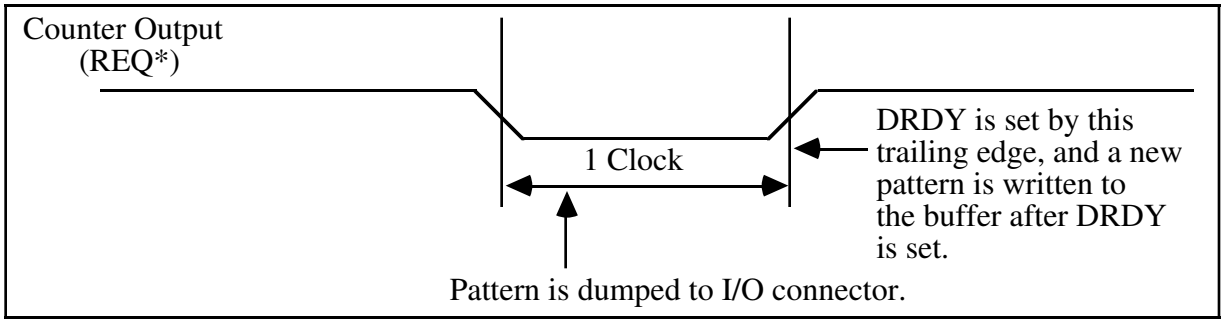


Figure 4-8. Pattern Generation

The programming steps to set up Counter 1 for pattern generation are as follows:

1. Set up Counter 1 for rate generation by writing hex 14 to the CNTRCMD Register for an 8-bit count or by writing hex 34 to the CNTRCMD Register for a 16-bit count.

2. Write the count to the CNTR1 Register (see Table 4-5). If the count is a 16-bit value, write the least significant byte first, then the most significant.

3. Write hex 10 to the CFG3 Register to enable Counter 1 for pattern generation (set the CNT1HSEN bit).

4. Set up Group 1 to select double-buffered trailing pulse mode and to clear handshaking: write hex 0378 to the CFG1 Register for an 8-bit Port A, or write hex 07F8 to the CFG1 Register for a 16-bit Port A.

5. Write hex 0278 (for 8-bit) or 0678 (for 16-bit) to the CFG1 Register to finish clearing handshaking.

6. Write hex A410 to the CFG3 Register to enable Counter 1 for pattern generation and to set Group 1 in write mode (set the WRITEA and WRITEB bits).

7. Write hex A414 to the CFG3 Register to start pattern generation on Counter 1 (set the CNT1EN bit).

The counter is loaded with the initial count in the CNTR1 Register, and when the CNT1EN bit is set, the counter is decremented by 1 on each clock pulse. REQ1 is initially high. When the counter decrements to 1, REQ1 goes low for one clock pulse and then returns to high. The counter is then reloaded from the CNTR1 Register and decrementing continues. The trailing edge of REQ1 causes the DRDY1 bit to go high. The DRDY1 bit can be monitored by a polling loop, by interrupt request generation, or by DMA request generation. With one of these methods, data can then be written out to Port A.

The programming steps to set up Counter 2 for pattern generation are as follows:

1. Set up Counter 2 for rate generation by writing hex 54 to the CNTRCMD Register for an 8-bit count or by writing hex 74 to the CNTRCMD Register for a 16-bit count.

2. Write the count to the CNTR2 Register (see Table 4-5). If the count is a 16-bit value, write the least significant byte first, then the most significant.

3. Write hex 20 to the CFG3 Register to enable Counter 2 for pattern generation (set the CNT2HSEN bit).

4. Set up Group 2 for trailing pulse mode and to clear handshaking: write hex 0378 to the CFG2 Register for an 8-bit Port C, or write hex 0778 to the CFG2 Register for a 16-bit Port C.

5. Write hex 0278 (for 8-bit) or 0678 (for 16-bit) to the CFG2 Register to finish clearing handshaking.

6. Write hex 4820 to the CFG3 Register to enable Counter 2 for pattern generation and to set Group 2 in write mode (set the WRITEC and WRITED bits).

7. Write hex 003 to the CFG4 Register to set double buffer output of Port D.

8 Write hex 4828 to the CFG3 Register to start pattern generation on Counter 2 (set the CNT2EN bit).

The counter is loaded with the initial count in the CNTR2 Register, and when the CNT2EN bit is set, the counter is decremented by 1 on each clock pulse. REQ2 is initially high. When the counter decrements to 1, REQ2 goes low for one clock pulse and then returns to high. The counter is then reloaded from the CNTR2 Register and decrementing continues. The trailing edge of REQ2 causes the DRDY2 bit to go high. The DRDY2 bit can be monitored by a polling loop, by interrupt request generation, or by DMA request generation. With one of these methods, data can then be written out to Port C. To use DMA for pattern generation, you must set the DMAEN bit, then program the DMA controller between steps 5 and 6.

Pattern Generation Using an External Signal

An external signal connected to the REQ line of a group can be used for pattern generation. The programming steps to configure Group 1 for 16-bit pattern generation on an external signal are as follows:

1. Connect the external signal to the REQ1 line.
2. If the external signal is active high pulse, write hex 0638 to the CFG1 Register. If the external signal is active low pulse, write hex 0678 to the CFG1 Register.
3. Write hex A400 to the CFG3 Register.
4. Write the first pattern to Port A. This operation stores the pattern in the write buffer of the port.
5. When REQ1 is received, the first pattern loaded in the write buffer of the port is transferred to the output buffer of the port; therefore, the first pattern is available on the digital I/O connector.
6. The received REQ1 sets the DRDY1 bit. Write the next pattern to Port A.
7. The next REQ1 received dumps the second pattern to the digital I/O connector.

The sequence continues until an LRESET1 is received.

The programming steps are similar for configuring Group 2 for pattern generation with an external signal.

Programming the RTSI Bus Interface

The RTSI switch connects signals on the AT-DIO-32F to the seven RTSI bus trigger lines. The RTSI switch has seven pins labeled A<6..0> connected to AT-DIO-32F signals, and seven pins labeled B<6..0> connected to the seven RTSI bus trigger lines. The signals connected to each pin are given in Table 4-6. RWGRP1* and RWGRP2* are board-generated, active low pulse signals that are 250-nsec to 500-nsec wide. RWGRP1* is the read/write signal for Port A and Port B. RWGRP2* is the read/write signal for Port C and Port D.

Table 4-6. RTSI Switch Signal Connections

RTSI Switch Pin	Signal Name	Signal Direction
Side A:		
A0	RWGRP1*	Output
A1	RWGRP2*	Output
A2	REQ1	Input
A3	REQ2	Input
A4	ACK1	Output
A5	ACK2	Output
A6	IN1	Input

(continues)

Table 4-6. RTSI Switch Signal Connections (Continued)

RTSI Switch Pin	Signal Name	Signal Direction
Side B:		
B0	TRIGGER0	Bidirectional
B1	TRIGGER1	Bidirectional
B2	TRIGGER2	Bidirectional
B3	TRIGGER3	Bidirectional
B4	TRIGGER4	Bidirectional
B5	TRIGGER5	Bidirectional
B6	TRIGGER6	Bidirectional

Programming the RTSI Bus Switch

The RTSI switch can be programmed to connect any of the signals on Side A to any of the signals on Side B 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 Shift Register and then writing to the RTSI 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 Side B pins. The high-order 28 bits select the signal sources for the Side A 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 4-9 shows the bit map of the RTSI switch 56-bit pattern.

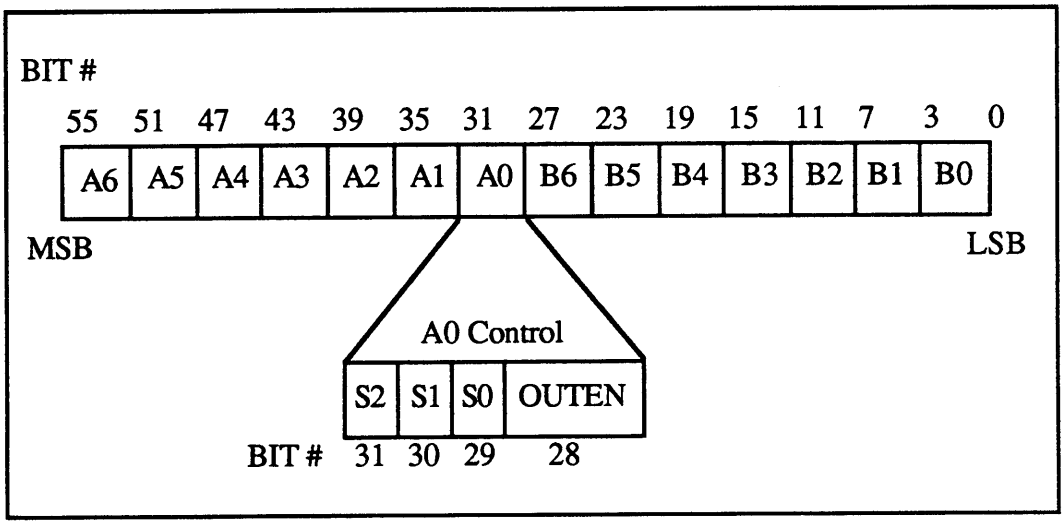


Figure 4-9. RTSI Switch Control Pattern

In Figure 4-9, 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 4-9.

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 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 A3 control field in Figure 4-9 contains the pattern 0111, the signal connected to pin B3 (Trigger Line 3) appears at pin A3. With this arrangement, the REQ2 signal can 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. With this arrangement, Trigger Line 4 can be driven by the AT-DIO-32F ACK2 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, follow 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. Specify 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 Shift Register (8-bit write).
3. Write 0 to the RTSI 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 performed by writing the low-order 8 bits of the 56-bit pattern to the RTSI 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 Shift Register is used. The higher-order bits are ignored.

Initializing the RTSI Bus Switch

Use the following sequence to initialize the RTSI bus switch. This sequence configures all signals to the RTSI bus switch as read signals rather than drive signals. Drive signals may cause undesirable results. All writes are 8-bit write operations.

1. Load the RTSI switch with the program data pattern:

   For $i = 0$ to 55 (decimal), write 0 to the RTSI Shift Register (base address + 10 hex).

2. Write to the RTSI Strobe Register to load the pattern into the RTSI switch:

   Write 0 to the RTSI Strobe Register (base address + 12 hex).

At startup, the RTSI bus switch is automatically initialized.

Appendix A
Specifications

This appendix lists the specifications for the AT-DIO-32F. These specifications are typical at 25° C unless otherwise noted.

Digital I/O

Number of channels ..... 32 I/O
Compatibility ..... TTL
Digital logic levels .....

Level	Min	Max
Input low voltage	0.0 V	0.8 V
Input high voltage	2.0 V	5.5 V
Input high current (V_{in} = 5 V)	—	10 µA
Output low voltage (I_{out} = 48 mA)	—	0.5 V
Output high voltage (I_{out} = -15 mA)	2.4 V	—

Transfer rate* (1 word = 16 bits) Absolute max
  Programmed I/O ..... 450 kwords/s
  DMA ..... 330 kwords/s
Handshaking 2-wire
Power-on state ..... Configured as inputs
Data transfers ..... DMA, interrupts, programmed I/O

* *Transfer rate depends on the computer and software. These tests were made using Assembly language programs running on a 16 MHz IBM PC AT compatible.*

RTSI

Trigger lines 7

Power Requirement

+5 VDC (±10%) ..... 1.05 A typ

Physical

Dimensions ..... 34.0 by 12.7 cm (13.4 by 5.0 in.)
I/O connector ..... 50-pin male

Environment

Operating temperature ..... 0° to 50° C
Storage temperature ..... -40° to 100° C
Relative humidity ..... 5% to 90% noncondensing

Noise Emission

FCC Class A verified, only with shielded ribbon cable

Appendix B
I/O Connector and Register Descriptions

This appendix contains a description of the AT-DIO-32F I/O connector and references to the registers of the AT-DIO-32F.

I/O Connector

Figure B-1 shows the pinout and signal names for the AT-DIO-32F 50-pin I/O connector.

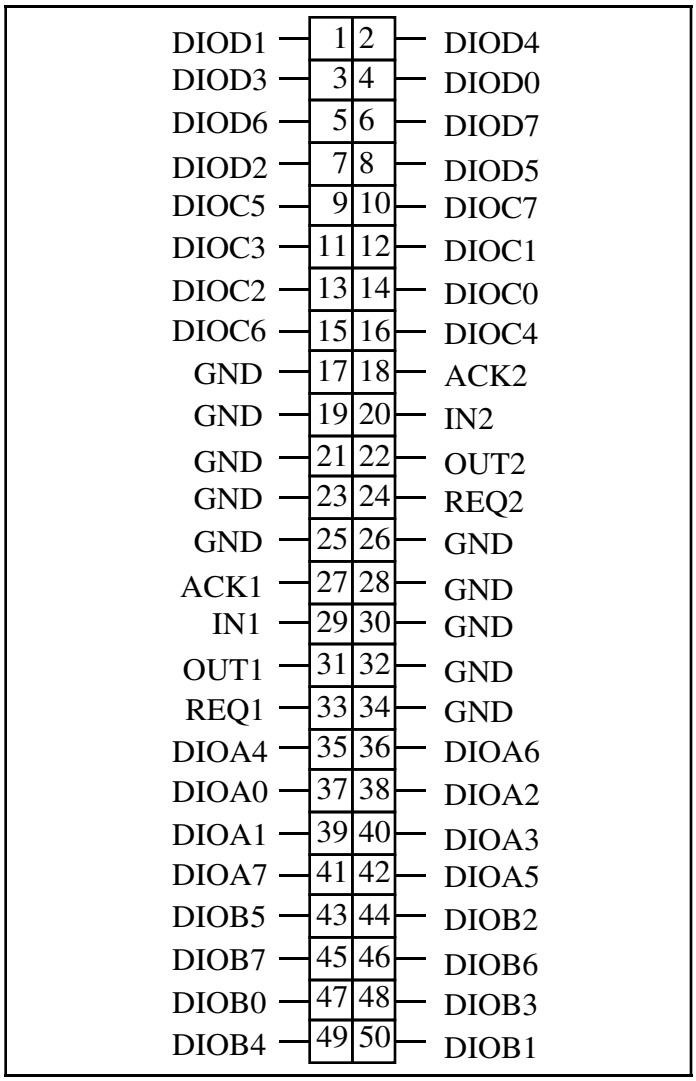


Figure B-1. AT-DIO-32F I/O Connector

Detailed signal specifications are included in Chapter 2, *Configuration and Installation*.

AT-DIO-32F Register Descriptions

A quick reference for the AT-DIO-32F appears on the following pages.

CFG1 Register

Base Address Offset = 00 (hex), 16-bit write only

15	14	13	12	11	10	9	8
DMAEN1	INTEN1	T1S2	T1S1	T1S0	DIOBEN	DIOAEN	LRESET1

7	6	5	4	3	2	1	0
X	INVRQ1	DBLBUFA	PULSE1	EDGE1	INVACK1	SETACK1	OUT1

CFG2 Register

Base Address Offset = 02 (hex), 16-bit write only

15	14	13	12	11	10	9	8
DMAEN2	INTEN2	T2S2	T2S1	T2S0	DIODEN	DIOCEN	LRESET2

7	6	5	4	3	2	1	0
X	INVRQ2	DBLBUFC	PULSE2	EDGE2	INVACK2	SETACK2	OUT2

CFG3 Register

Base Address Offset = 04 (hex), 16-bit write only

15	14	13	12	11	10	9	8
DBLBUFB	WRITED	WRITEB	TRANS32	WRITEC	WRITEA	CNT2SRC	CNT1SRC

7	6	5	4	3	2	1	0
DBLDMA	CNTINTEN	CNT2HSEN	CNT1HSEN	CNT2EN	CNT1EN	TCINTEN2	TCINTEN1

CFG4 Register

Base Address Offset = 14 (hex), 16-bit write only

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

7	6	5	4	3	2	1	0
0	0	0	0	LPULSE2	LPULSE1	DBLBUFD	REVC

STAT Register

Base Address Offset = 00 (hex), 16-bit read only

15	14	13	12	11	10	9	8
DMACH	CNTINT	X	TRANS32	IN1	DMATC2	DMATC1	DRDY1

7	6	5	4	3	2	1	0
REQ1	ACK1	IN2	0	X	DRDY2	REQ2	ACK2

CNTINTCLR Register

Base Address Offset = 0A (hex), 16-bit write only
Bit map not applicable, no bits used

DMACLR1 Register

Base Address Offset = 0C (hex), 16-bit write only
Bit map not applicable, no bits used

DMACLR2 Register

Base Address Offset = 0E (hex), 16-bit write only
Bit map not applicable, no bits used

Port A Register

Base Address Offset = 06 (hex), 8-bit or 16-bit read and write

7	6	5	4	3	2	1	0
DIOA7	DIOA6	DIOA5	DIOA4	DIOA3	DIOA2	DIOA1	DIOA0

Port B Register

Base Address Offset = 07 (hex), 8-bit read and write

7	6	5	4	3	2	1	0
DIOB7	DIOB6	DIOB5	DIOB4	DIOB3	DIOB2	DIOB1	DIOB0

Port C Register

Base Address Offset = 08 (hex), 8-bit or 16-bit read and write

7	6	5	4	3	2	1	0
DIOC7	DIOC6	DIOC5	DIOC4	DIOC3	DIOC2	DIOC1	DIOC0

Port D Register

Base Address Offset = 09 (hex), 8-bit read and write

7	6	5	4	3	2	1	0
DIOD7	DIOD6	DIOD5	DIOD4	DIOD3	DIOD2	DIOD1	DIOD0

RTSISHFT Register

Base Address Offset = 10 (hex), 8-bit write only

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

RTSISTRB Register

Base Address Offset = 12 (hex), 8-bit write only
Bit map not applicable, no bits used

CNTR1 Register (REQ1 Generator)

Base Address Offset = 18 (hex), 8-bit read and write

7	6	5	4	3	2	1	0
CNTR1B7	CNTR1B6	CNTR1B5	CNTR1B4	CNTR1B3	CNTR1B2	CNTR1B1	CNTR1B0

CNTR2 Register (REQ2 Generator)

Base Address Offset = 1A (hex), 8-bit read and write

7	6	5	4	3	2	1	0
CNTR2B7	CNTR2B6	CNTR2B5	CNTR2B4	CNTR2B3	CNTR2B2	CNTR2B1	CNTR2B0

CNTR3 Register (Timebase Generator)

Base Address Offset = 1C (hex), 8-bit read and write

7	6	5	4	3	2	1	0
CNTR3B7	CNTR3B6	CNTR3B5	CNTR3B4	CNTR3B3	CNTR3B2	CNTR3B1	CNTR3B0

CNTRCMD Register

Base Address Offset = 1E (hex), 8-bit write only

7	6	5	4	3	2	1	0
CNTRSEL1	CNTRSEL0	RWSEL1	RWSEL0	MODESEL2	MODESEL1	MODESEL0	BCDSEL

Read-Back Command

7	6	5	4	3	2	1	0
CNTRSEL1	CNTRSEL0	COUNT*	STATUS*	CNTR3	CNTR2	CNTR1	0

Status Byte

7	6	5	4	3	2	1	0
OUT	NULL	RW1	RW0	MODE2	MODE1	MODE0	BCD

Appendix C
Application Notes

This appendix contains the application notes for the AT-DIO-32F board. The versatile AT-DIO-32F can interface the PC to almost any 8-bit, 16-bit, or 32-bit parallel device or I/O module rack. These programs explore the several handshaking modes of the AT-DIO-32F and can easily be modified to fulfill your own specialized communication needs.

All programs were written in Microsoft C. Functions within the programs are designed as independent modules with the necessary parameters passed in the function call. Therefore, the functions can easily be transported to your own program.

Several software packages that support the AT-DIO-32F are also available from National Instruments. These packages simplify programming, thus decreasing software development time. For more information about optional software, see Chapter 1, *Introduction*.

Communicating with a Printer

The National Instruments AT-DIO-32F can interface the PC to any Centronics or Centronics-compatible printer. In the following program, Port A is configured for Group 1 handshaking, and is connected to the printer's data lines. Ports B, C, and D are not used and can be configured in Group 2 to communicate with another device.

Cabling

Build a cable according to the following list to connect the AT-DIO-32F 50-pin connector to the printer 36-pin connector. Any pins not listed are not connected and should be left open. Refer to Chapter 2, *Configuration and Installation*, of this manual for connector and cable specifications.

Centronics		AT-DIO-32F	
/STROBE	1	27	ACK1
Data0	2	37	DIOA0
Data1	3	39	DIOA1
Data2	4	38	DIOA2
Data3	5	40	DIOA3
Data4	6	35	DIOA4
Data5	7	42	DIOA5
Data6	8	36	DIOA6
Data7	9	41	DIOA7
/ACKN	10	33	REQ1
BUSY	11	29	IN1
PE	12	20	IN2
/ATFD	14	31	OUT1
/INIT	31	22	OUT2

Centronics		AT-DIO-32F	
GND	19	17	GND
GND	20	17	GND
GND	21	19	GND
GND	22	19	GND
GND	23	21	GND
GND	24	21	GND
GND	25	23	GND
GND	26	23	GND
GND	27	28	GND
GND	28	28	GND
GND	29	30	GND
GND	30	32	GND
GND	33	34	GND

Sending Files to be Printed

A selected disk file can be printed with the following program, which is written in C. The program begins by prompting the user to enter the base address of the board and the name of the file to be printed. After verifying that the file exists, the program reads characters from the file and writes them to the port A data lines. Handshaking is automatic using the group 1 REQ and ACK lines.

The Centronics specification requires the handshaking acknowledge signal to be a minimum of 500 nsec. This is accomplished by setting the Data Settling Delay bits in the CFG1 Register. A 500-nsec delay is sufficient for the printer to recognize communication.

```
/*   Send a file from the PC to a Centronics printer via the
     National Instruments AT-DIO-32F.
*/

#include "stdio.h"

#define CFG1offset  0x00
#define CFG2offset  0x02
#define CFG3offset  0x04
#define CFG4offset  0x14
#define STAToffset  0x00
#define PORTAoffset 0x06

int base_address,  /* base_address of the AT-DIO-32F board */
    cfg1,                /* address of CFG1 register */
    cfg2,                /* address of CFG2 register */
    cfg3,                /* address of CFG3 register */
    cfg4,                /* address of CFG4 register */
    tstat,               /* address of STAT register */
    porta;               /* address of PORTA register */

FILE *fp;
FILE *fopen();

char filename[80];

main()
{
  int i;

  /* base_address is input from the key board */
  printf("\nEnter the base_address(Hex #): ");
  scanf("%x", &base_address);
  printf("\nbase_address = %x(Hex)",base_address);  /* echo print */

  /* calculate registers address */
  cfg1 = base_address + CFG1offset;
  cfg2 = base_address + CFG2offset;
  cfg3 = base_address + CFG3offset;
  cfg4 = base_address + CFG4offset;
  tstat = base_address + STAToffset;
  porta = base_address + PORTAoffset;
```

```
  /* set up AT-DIO-32F to communicate with printer */
  Setup_DIO();     /* setup the AT-DIO-32F */

  /* get the chars to print and send out */
  printf("\nEnter the name of the file to print: \n");
  scanf("%s",filename);
  /* gets(filename);*/
  fp = fopen(filename,"r");

  if ( Valid_file())   /* test if a valid file opened */
   Read_n_Write();     /* read the file and send to printer */

  fclose(fp);
} /* main */


/*****************     functions    ************************************/

/* read file and print */
Read_n_Write()
{
  char ch;
  while ((ch=fgetc(fp)) != EOF)   /* print chars until EOF is reached */
  {
    Print_char(ch);
  }
  Print_char(0x0D);               /* send linefeed to complete printing */
}

/* check whether given filename is valid */
Valid_file()
{
  if (fp == NULL)
  {
    printf("\nCan't open the file %s.",filename);
    return(0);          /* no file access */
  }
  else
    return(1);          /* file exists, go ahead */
}

/* setup the AT-DIO-32F board */
Setup_DIO()
{
  outpw(cfg1, 0x0000);    /* initialize the board */
  outpw(cfg2, 0x0000);
  outpw(cfg3, 0x0000);
  outpw(cfg4, 0x0001);
  outpw(cfg2, 0x001);     /* OUT2 = *INIT high */
  outpw(cfg1, 0x2B44);    /* reset porta handshaking */
  outpw(cfg1, 0x2A44);    /* porta level, invert ACK1 and REQ1, delay=500ns */
  outpw(cfg3, 0x0400);    /* porta write */
}

/* check printer is ready */
Printer_Rdy()
{
  int err = 0;
```

```
  if (Read_Stat() & 0x20)      /* get IN2(paper error) bit */
  {
    printf("\nPrinter out of paper.");
    printf("\nStat = %x",Read_Stat());
    err = 1;
  }
  while ((Read_Stat() & 0x0800));  /* wait until IN1(BUSY) is not set */
  return(err);
}

/* read the stat register */
Read_Stat()
{
  return(inpw(tstat));
}

/* send a character to the printer by writing to porta */
Print_char(ch)
char ch;
{
  int i = 0, err;

  err = Printer_Rdy();     /* check for paper error or BUSY */
  if (err)
  {
    printf("\nCan't get printer ready.");
    exit(1);
  }
  while(!(Read_Stat() & 0x0100)  && i<20000)    /* wait for DRDY1 bit set */
    i++;
  if (i == 20000)
  {
    printf("\nCan't get DRDY1 set.");
    printf("\nStat = %x",Read_Stat());
    exit(1);
  }

  outp(porta, ch);
  printf("%c",ch);               /* echo character to screen */
}
```

AT-DIO-32F to AT-DIO-32F 16-Bit Communications

This program transmits 16-bit words between two AT-DIO-32F boards in separate PC computers. With the program, the AT-DIO-32F can communicate with National Instruments MC-DIO-32F or NB-DIO-32F. Ports A and B function as the 16-bit read channel with Group 1 handshaking, and Ports C and D comprise the 16-bit write channel utilizing Group 2 handshaking.

Cabling

Use the following cable to connect the two AT-DIO-32F boards. Both connectors are 50-pin, female, ribbon cable connectors. Any pins not listed are not used and should be left open. The signal names PA, PB, PC, and PD refer to Port A, Port B, Port C, and Port D, respectively. Refer to Chapter 2 of this manual for connector and cable specifications.

AT-DIO-32F(1)		AT-DIO-32F(2)	
PA0	37	14	PC0
PA1	39	12	PC1
PA2	38	13	PC2
PA3	40	11	PC3
PA4	35	16	PC4
PA5	42	9	PC5
PA6	36	15	PC6
PA7	41	10	PC7
PB0	47	4	PD0
PB1	50	1	PD0
PB2	44	7	PD2
PB3	48	3	PD3
PB4	49	2	PD4
PB5	43	8	PD5
PB6	46	5	PD6
PB7	45	6	PD7
REQ1	33	18	ACK2
ACK1	27	24	REQ2
GND	17	17	GND
GND	19	19	GND
GND	21	17	GND
GND	23	23	GND

AT-DIO-32F(1)		AT-DIO-32F(2)	
PC0	14	37	PA0
PC1	12	39	PA1
PC2	13	38	PA2
PC3	11	40	PA3
PC4	16	35	PA4
PC5	9	42	PA5
PC6	15	36	PA6
PC7	10	41	PA7
PD0	4	47	PB0
PD1	1	50	PB1
PD2	7	44	PB2
PD3	3	48	PB3
PD4	2	49	PB4
PD5	8	43	PB5
PD6	5	46	PB6
PD7	6	45	PB7
REQ2	24	27	ACK1
ACK2	18	33	REQ1
GND	28	28	GND
GND	30	30	GND
GND	32	32	GND
GND	34	34	GND

Sending and Receiving files with the AT-DIO-32F

The following program is divided into two functional parts, one for sending files from the AT-DIO-32F and the other for receiving files. Both are structured similarly and check for the EOF marker which marks the end of transmission. The handshaking between the AT-DIO-32F boards is fully automatic and requires no software toggling of lines.

For 16-bit communications, the program compresses two 8-bit characters into one 16-bit word using the functions getw() and putw() from the standard unix library. The function getw() is modified to the new function getwd() to recognize and return both 8-bit and 16-bit EOF markers. Both getwd() and putw() are included at the end of the program and either function can be expanded to handle 32-bit words. Notice that Port A and Port B are declared 16-bit type int.

```
/*
      This program allows the AT-DIO-32F to send and receive 16-bit data
      to and from an AT-DIO-32F in another PC.
*/

include <stdio.h>
```

```
/* address offset of the registers of AT-DIO-32F */
#define     CFG1os      0x00
#define     CFG2os      0x02
#define     CFG3os      0x04
#define     CFG4os      0x14
#define     STATos      0x00
#define     PORTAos     0x06
#define     PORTCos     0x08

unsigned int      base_addr;
unsigned int    CFG1,CFG2,CFG3,CFG4,            /* Registers of AT-DIO-32F */
            STAT1,PORTA,PORTC;
FILE        *fp;
char        filename[81];
int         lowEOF;

main()
{

      /*... base_address is input from the keyboard....*/
      printf("\nEnter the base_address of the AT-MIO-32F(Hex):  ");
      scanf("%x", &base_addr);

      CFG1  =     CFG1os + base_addr;
      CFG2  =     CFG2os + base_addr;
      CFG3  =     CFG3os + base_addr;
      CFG4  =     CFG4os + base_addr;
      STAT1 =     STATos + base_addr;
      PORTA =     PORTAos + base_addr;
      PORTC =     PORTCos + base_addr;


      lowEOF = (EOF & 0x00ff);      /* Set so that EOF character is in the low
byte only. */
      setup_dio();
      choose_what();
}/* End main */



/*..... Sets up AT-DIO-32F .....*/
setup_dio()
{
      outpw(CFG1, 0x00);      /* Initialize the board */
      outpw(CFG2, 0x00);
      outpw(CFG3, 0x00);
      outpw(CFG4, 0x01);      /* Set for Rev. C board */

      /* Set up for leading edge. */
      outpw(CFG1, 0x710);     /* Enable DIOAEN and DIOBEN; PULSE1 is high;
                           EDGE1 is low, and LRESET1 */
      outpw(CFG2, 0x710);     /* Enable DIOCEN and DIODEN; PULSE2 is high;

                           EDGE2 is low, and LRESET2. */
      outpw(CFG1, 0x610);     /* Reconfigure after reset. */
      outpw(CFG2, 0x610);
      outpw(CFG3, 0x4800);    /* Enable WRITEC and WRITED to write;
                             WRITEA and WRITEB are configured to read. */
```

```
} /* End of setup_dio */


/*... Ask to send or receive data and then do it ...*/
choose_what()
{
      char  ch;
      choose_msg();
      ch = getch();
      while((ch != 'e') && (ch != 'E')){
            switch(ch){
                  case 'S':
                  case 's': {             /* User wants to send data. */
                        send_main();      /* Send data. */
                        choose_msg();     /* Give choice again. */
                        break;
                        } /* End case 's' */
                  case 'R':
                  case 'r': {       /* User wants to receive data.*/
                        get_main(); /* Get the data. */
                        choose_msg();     /* Give choice again. */
                        break;
                        }/* End case 'r' */
            }/* End switch */
      ch = getch();
      }/* End while */
printf("\nExit requested.");  /* User selected 'e' for exit.      */
}/* End choose_what */


/*... Message for choose_what() ...*/
choose_msg()
{
      printf("\n\nDo you want to send (s) or receive (r) data, or exit (e)?
");
}/* End choose_msg */


/*... Check whether given filename is valid ...*/
valid_file()
{
      if (fp == NULL){
            printf("can't open the file %s.\n", filename);
            return 0;
            }/* End if */
      else
            return 1;   /* file exists, go ahead. */
}/* End valid_file() */


/*************** Routines for sending data *****************/

/*... Get data and send to another board ...*/
send_main()
{
      printf("\nEnter the name of the file to send:\n");
      scanf("%s", filename);
```

```
      fp = fopen(filename, "r");    /* Open file for reading. */
      if(valid_file())        /* Test if valid file. */
            send_file();            /* Read file and send. */
      fclose(fp);
}/* End send_main */


/*... Send a file to the AT-DIO-32F ...*/
send_file()
{
      int word;

      /* Read in words from file and send, do while no EOF in low byte. */
      while(((word=getwd(fp)) & 0x00FF) != lowEOF)
            send_word(word);
      send_word(word);  /* This last word read and sent contains an EOF. */
}/* End send_file */


/*... Send a word to the AT-DIO-32F ...*/
send_word(wd)
int wd;
{
      while(!data_out_rdy());   /* Wait until ok for data to be written. */
      putw(wd, stdout);   /* Echo characters to screen.     */
      outpw(PORTC, wd);   /* Send a 16-bit word to the MC-DIO-32F. */
}


/*... Return non-zero value if the AT-DIO-32F is ready to be written more
data...*/
data_out_rdy()
{
      return (inpw(STAT1) & 0x0004);            /* Returns DRDY2 bit. */
}



/****************** Routines for Receiving Data *********************/

/*... Receive 16-bit data from the other AT-DIO-32F...*/
get_main()
{
      int word;
      printf("\nEnter the name of the file to receive data: \n");
      scanf("%s", filename);
      fp = fopen(filename, "w");          /* Open file for writing. */
      printf("\nWaiting to receive data.....\n\n");

      /* Gets chars until an EOF is received in the low byte. */
      while(((word = in_word()) & 0x0FF) != lowEOF){
            putw(word, stdout);     /* Echo word to screen. */
            putw(word, fp);         /* Put word in file. */
            }/* End while */

      /* Put final word received containing an EOF into the file. */
      putw(word, fp);               /* Put char and EOF in file. */
      putw(word, stdout);           /* Echo word to screen. */
```

```
      printf("\nAn End-of-File has been received.\n");
      fclose(fp);
}/* End get_main */


/*... Get the 16-bit word from the AT-DIO-32F ...*/
in_word()
{
      int wd;
      while(!data_in_rdy()) ;       /* Wait until data is ready. */
      wd = inpw(PORTA);
      return wd;              /* Return 16-bit data.  */
}/* End in_word */


/*... Return non-zero value if data is ready to be read ...*/
data_in_rdy()
{
      return(inpw(STAT1) & 0x0100); /* Return value of DRDY1 bit. */
}/* End data_in_rdy */



/******************** Modified Unix Commands ********************/

/*... getwd() is a modification of the unix.h function getw().
            The returned parameter is the only change in the unix
            function.  getwd() will return an int EOF (0xFFFF) if
            either the first or second byte read was an EOF.
            The function getwd(), however, returns the following:
                  0xFFFF      int EOF if the high byte is an EOF char,
                  0x..FF      a char EOF in the low byte if a valid char
                              was read in the high byte,
                  0x....      a valid char in the high and low byte if no
                              EOF was read in either.
...*/
getwd(who)
FILE *who;
{
      register int      chi, clo;
      if ((chi = fgetc(who)) == EOF)      /* EOF will be word length. */
            return (EOF);           /* Don't read beyond EOF. */
      clo = fgetc(who);       /* Read the high byte.  */
      return((chi & 0xFF) << 8 | (clo & 0xFF)); /* Return word.   */
}/* End getwd */

/*... putwd() is a unix command.
      Note that the functions place characters in the file one at a time
      and checks for an EOF with every character.  This prevents writing
      EOF to a file.
...*/
putw(word, who)
int word;
FILE *who;
{
      if ((word>>8) & 0xFF == EOF)
            return (EOF);
      else
```

```
            fputc(((word>>8) & 0xFF), who);

      if (word & 0xFF == EOF)
            return (EOF);
      else
            fputc((word & 0xFF), who);

      return (word);
}/* End putw */
```

The AT-DIO-32F and I/O Module Racks

The AT-DIO-32F is pin-compatible with several 32-channel I/O module racks produced by several different companies. Any I/O module rack pin-compatible with the DEC DRV11-J parallel interface is pin-compatible with the AT-DIO-32F. The AT-DIO-32F can also be used with I/O module racks that are not pin-compatible if a special cable is built.

The AT-DIO-32F is directly pin-compatible with the following I/O module racks:

OPTO 22	Gordos
PB32DEC	PB-32SM PB-32Q PB32D

The AT-DIO-32F can be used with the following I/O module racks if connected with a cable indicated by the specifications in Table C-1. Any pins not listed are not connected and should be left open.

Table C-1. Cable Specification for Connections to 8-, 16-, or 24-Channel I/O Module Racks

OPTO 22	Gordos	Potter & Brumfield
PB8 PB16A PB16C PB24 PB24Q	PB8 PB16 PB-16SM PB16Q PB24 PB24Q	2IO-8 2IO-16 2IO-24

AT-DIO-32F			I/O Rack	AT-DIO-32F(1)			I/O Rack
PA0	37	47	Data 0	PC0	14	15	Data 16
PA1	39	45	Data 1	PC1	12	13	Data 17
PA2	38	43	Data 2	PC2	13	11	Data 18
PA3	40	41	Data 3	PC3	11	9	Data 19
PA4	35	39	Data 4	PC4	16	7	Data 20
PA5	42	37	Data 5	PC5	9	5	Data 21
PA6	36	35	Data 6	PC6	15	3	Data 22
PA7	41	33	Data 7	PC7	10	1	Data 23
PB0	47	31	Data 8	GND	17	16	GND
PB1	50	29	Data 9	GND	19	18	GND
PB2	44	27	Data 10	GND	21	20	GND
PB3	48	25	Data 11	GND	23	22	GND
PB4	49	23	Data 12	GND	28	28	GND
PB5	43	21	Data 13	GND	30	30	GND
PB6	46	19	Data 14	GND	32	32	GND
PB7	45	17	Data 15	GND	34	34	GND

The majority of these racks interface to parallel data through a 50-pin edge connector. A cable must be made with a 50-pin edge connector on one end and a 50-pin header connector on the other.

The AT-DIO-32F I/O connector is a 50-pin male ribbon cable header. Recommended manufacturer part numbers for this header are:

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

The mating connector for the AT-DIO-32F is a 50-position, polarized, ribbon socket connector with strain relief. National Instruments uses a polarized (keyed) connector to prevent inadvertent upside-down connection to the AT-DIO-32F. Recommended manufacturer part numbers for this mating connector are:

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

Recommended manufacturer part numbers for the 50-pin edge connector that connects a module rack to an edge connector are:

Electronic Products Division/3M	part number 3415-0001
T&B Ansley Corporation	part number 609-5015M

A polarizing key can be plugged into these edge connectors to prevent inadvertent upside-down connection to the I/O module rack. The location of this key varies from rack to rack. Consult the specification for the rack you intend to use for the location of any polarizing key. The recommended manufacturer part numbers for this polarizing key are:

Electronic Products Division/3M	part number 3439-2
T&B Ansley Corporation	part number 609-0005

Use Mode 0 when communicating with I/O module racks. Refer to Chapter 3, *Theory of Operation*, of this manual for programming information.

Appendix D
Intel Data Sheet*

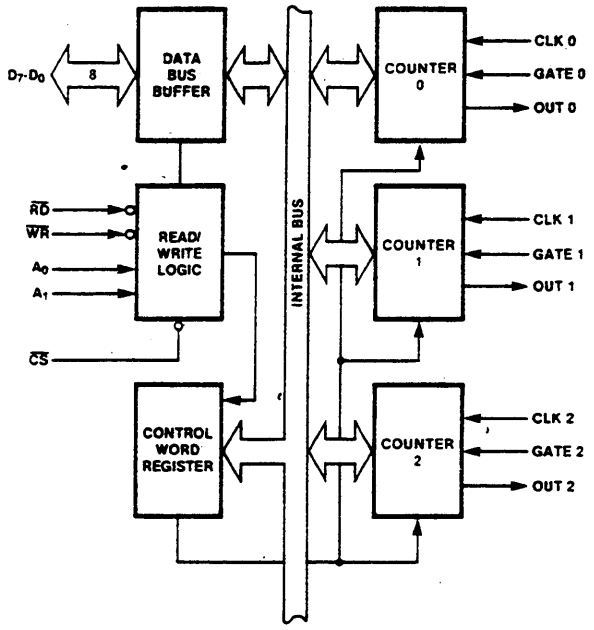
This appendix contains the *8254 Programmable Interval Timer* (Intel Corporation) data sheet. This counter/timer device is used on the AT-DIO-32F board.

8254
PROGRAMMABLE INTERVAL TIMER

- **Compatible with All Intel and Most Other Microprocessors**
- **Handles Inputs from DC to 10 MHz**
  - **— 5 MHz 8254-5**
  - **— 8 MHz 8254**
  - **— 10 MHz 8254-2**
- **Status Read-Back Command**
- **Six Programmable Counter Modes**
- **Three Independent 16-Bit Counters**
- **Binary or BCD Counting**
- **Single +5V Supply**
- **Available in EXPRESS**
  - **— Standard Temperature Range**

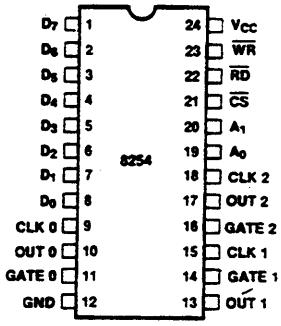
The Intel® 8254 is a counter/timer device designed to solve the common timing control problems in microcomputer system design. It provides three independent 16-bit counters, each capable of handling clock inputs up to 10 MHz. All modes are software programmable. The 8254 is a superset of the 8253.

The 8254 uses HMOS technology and comes in a 24-pin plastic or CERDIP package.



231164-1

Figure 1. 8254 Block Diagram



231164-2

Figure 2. Pin Configuration

August 1987
Order Number: 231164-004

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
D_7–D_0	1–8	I/O	**DATA:** Bi-directional three state data bus lines, connected to system data bus.
CLK 0	9	I	**CLOCK 0:** Clock input of Counter 0.
OUT 0	10	O	**OUTPUT 0:** Output of Counter 0.
GATE 0	11	I	**GATE 0:** Gate input of Counter 0.
GND	12		**GROUND:** Power supply connection.
V_{CC}	24		**POWER:** +5V power supply connection.
$\overline{WR}$	23	I	**WRITE CONTROL:** This input is low during CPU write operations.
$\overline{RD}$	22	I	**READ CONTROL:** This input is low during CPU read operations.
$\overline{CS}$	21	I	**CHIP SELECT:** A low on this input enables the 8254 to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and $\overline{WR}$ are ignored otherwise.
A_1, A_0	20–19	I	**ADDRESS:** Used to select one of the three Counters or the Control Word Register for read or write operations. Normally connected to the system address bus.

A_1	A_0	Selects
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Word Register

CLK 2	18	I	**CLOCK 2:** Clock input of Counter 2.
OUT 2	17	O	**OUT 2:** Output of Counter 2.
GATE 2	16	I	**GATE 2:** Gate input of Counter 2.
CLK 1	15	I	**CLOCK 1:** Clock input of Counter 1.
GATE 1	14	I	**GATE 1:** Gate input of Counter 1.
OUT 1	13	O	**OUT 1:** Output of Counter 1.

FUNCTIONAL DESCRIPTION

General

The 8254 is a programmable interval timer/counter designed for use with Intel microcomputer systems. It is a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 8254 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in software, the programmer configures the 8254 to match his requirements and programs one of the counters for the desired delay. After the desired delay, the 8254 will interrupt the CPU. Software overhead is minimal and variable length delays can easily be accommodated.

Some of the other counter/timer functions common to microcomputers which can be implemented with the 8254 are:

- Real time clock
- Event-counter
- Digital one-shot
- Programmable rate generator
- Square wave generator
- Binary rate multiplier
- Complex waveform generator
- Complex motor controller

Block Diagram

DATA BUS BUFFER

This 3-state, bi-directional, 8-bit buffer is used to interface the 8254 to the system bus (see Figure 3).

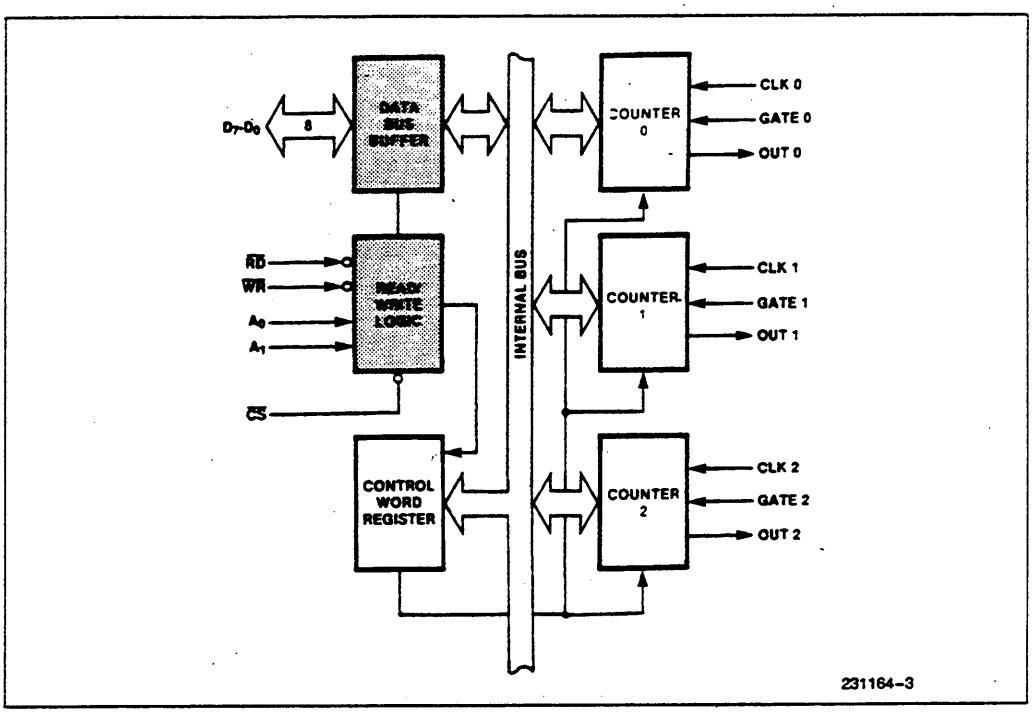


Figure 3. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

READ/WRITE LOGIC

The Read/Write Logic accepts inputs from the system bus and generates control signals for the other functional blocks of the 8254. A_1 and A_0 select one of the three counters or the Control Word Register to be read from/written into. A "low" on the $\overline{RD}$ input tells the 8254 that the CPU is reading one of the counters. A "low" on the $\overline{WR}$ input tells the 8254 that the CPU is writing either a Control Word or an initial count. Both $\overline{RD}$ and $\overline{WR}$ are qualified by $\overline{CS}$; $\overline{RD}$ and $\overline{WR}$ are ignored unless the 8254 has been selected by holding $\overline{CS}$ low.

CONTROL WORD REGISTER

The Control Word Register (see Figure 4) is selected by the Read/Write Logic when $A_1,A_0 = 11$. If the CPU then does a write operation to the 8254, the data is stored in the Control Word Register and is interpreted as a Control Word used to define the operation of the Counters.

The Control Word Register can only be written to; status information is available with the Read-Back Command.

COUNTER 0, COUNTER 1, COUNTER 2

These three functional blocks are identical in operation, so only a single Counter will be described. The internal block diagram of a single counter is shown in Figure 5.

The Counters are fully independent. Each Counter may operate in a different Mode.

The Control Word Register is shown in the figure; it is not part of the Counter itself, but its contents determine how the Counter operates.

The status register, shown in Figure 5, when latched, contains the current contents of the Control Word Register and status of the output and null count flag. (See detailed explanation of the Read-Back command.)

The actual counter is labelled CE (for "Counting Element"). It is a 16-bit presettable synchronous down counter.

OL_M and OL_L are two 8-bit latches. OL stands for "Output Latch"; the subscripts M and L stand for "Most significant byte" and "Least significant byte"

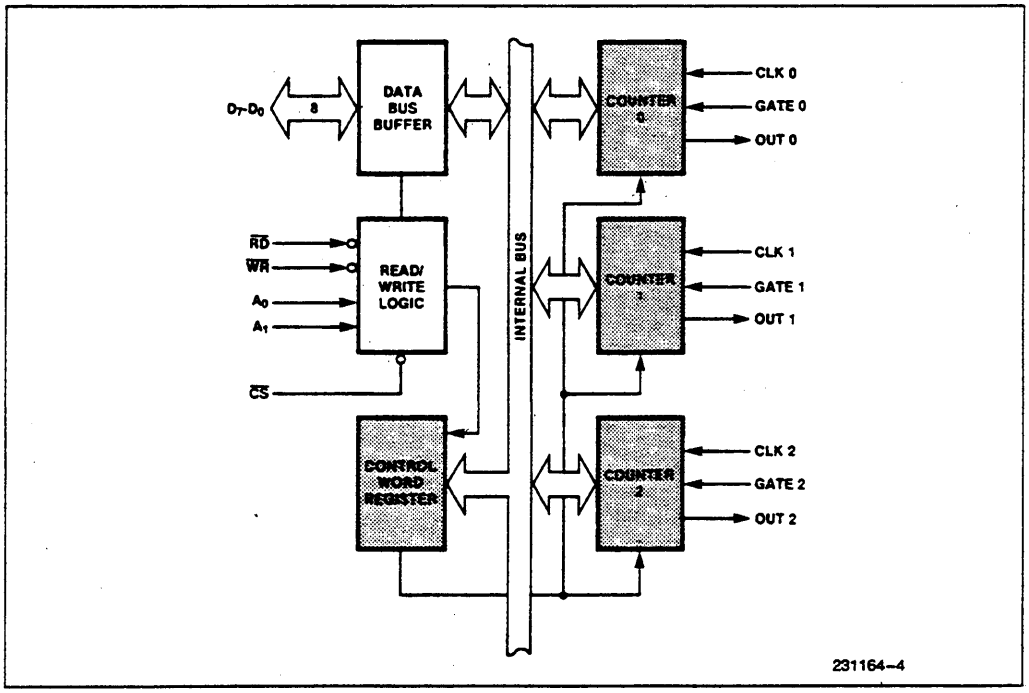


Figure 4. Block Diagram Showing Control Word Register and Counter Functions

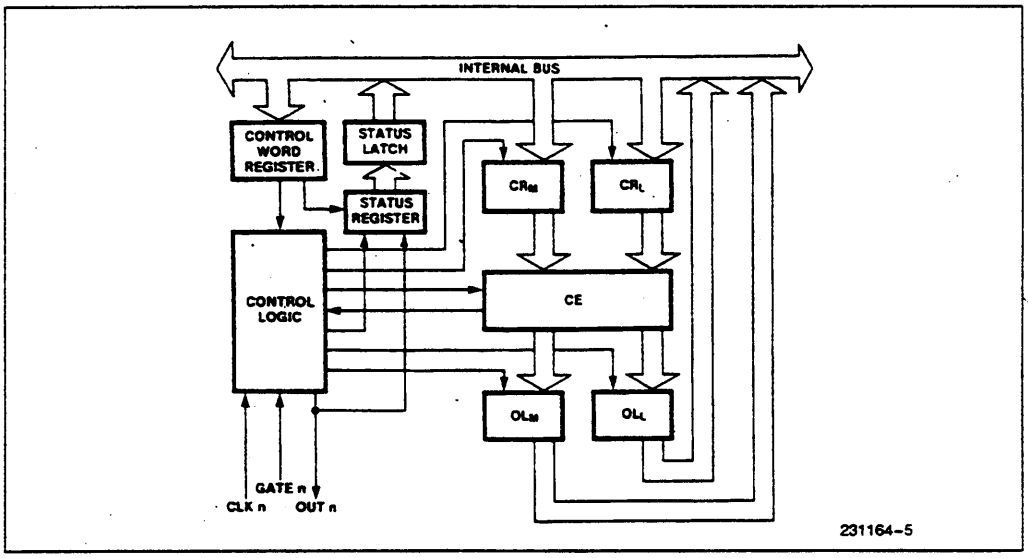


Figure 5. Internal Block Diagram of a Counter

respectively. Both are normally referred to as one unit and called just OL. These latches normally "follow" the CE, but if a suitable Counter Latch Command is sent to the 8254, the latches "latch" the present count until read by the CPU and then return to "following" the CE. One latch at a time is enabled by the counter's Control Logic to drive the internal bus. This is how the 16-bit Counter communicates over the 8-bit internal bus. Note that the CE itself cannot be read; whenever you read the count, it is the OL that is being read.

Similarly, there are two 8-bit registers called CR_M and CR_L (for "Count Register"). Both are normally referred to as one unit and called just CR. When a new count is written to the Counter, the count is stored in the CR and later transferred to the CE. The Control Logic allows one register at a time to be loaded from the internal bus. Both bytes are transferred to the CE simultaneously. CR_M and CR_L are cleared when the Counter is programmed. In this way, if the Counter has been programmed for one byte counts (either most significant byte only or least significant byte only) the other byte will be zero. Note that the CE cannot be written into; whenever a count is written, it is written into the CR.

The Control Logic is also shown in the diagram. CLK n, GATE n, and OUT n are all connected to the outside world through the Control Logic.

8254 SYSTEM INTERFACE

The 8254 is a component of the Intel Microcomputer Systems and interfaces in the same manner as all other peripherals of the family. It is treated by the system's software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A_0,A_1 connect to the A_0, A_1 address bus signals of the CPU. The CS can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an Intel 8205 for larger systems.

OPERATIONAL DESCRIPTION

General

After power-up, the state of the 8254 is undefined. The Mode, count value, and output of all Counters are undefined.

How each Counter operates is determined when it is programmed. Each Counter must be programmed before it can be used. Unused counters need not be programmed.

Programming the 8254

Counters are programmed by writing a Control Word and then an initial count.

The Control Words are written into the Control Word Register, which is selected when A_1,A_0 = 11. The Control Word itself specifies which Counter is being programmed.

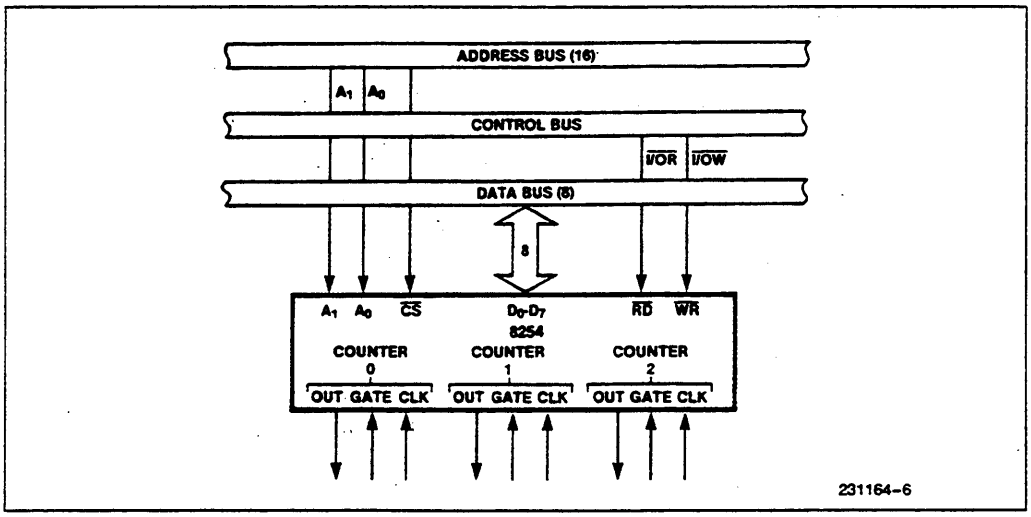


Figure 6. 8254 System Interface

Control Word Format

A_1,A_0 = 11 $\overline{CS}$ = 0 $\overline{RD}$ = 1 $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC—Select Counter

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (see Read Operations)

M—Mode

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

RW—Read/Write

RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only
1	0	Read/Write most significant byte only
1	1	Read/Write least significant byte first, then most significant byte

BCD

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Figure 7. Control Word Format

By contrast, initial counts are written into the Counters, not the Control Word Register. The A_1,A_0 inputs are used to select the Counter to be written into. The format of the initial count is determined by the Control Word used.

Write Operations

The programming procedure for the 8254 is very flexible. Only two conventions need to be remembered:

1) For each Counter, the Control Word must be written before the initial count is written.
2) The initial count must follow the count format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

Since the Control Word Register and the three Counters have separate addresses (selected by the A_1,A_0 inputs), and each Control Word specifies the Counter it applies to (SC0,SC1 bits), no special instruction sequence is required. Any programming sequence that follows the conventions in Figure 7 is acceptable.

A new initial count may be written to a Counter at any time without affecting the Counter's programmed Mode in any way. Counting will be affected as described in the Mode definitions. The new count must follow the programmed count format.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same Counter. Otherwise, the Counter will be loaded with an incorrect count.

	A_1	A_0		A_1	A_0
Control Word—Counter 0	1	1	Control Word—Counter 2	1	1
LSB of count—Counter 0	0	0	Control Word—Counter 1	1	1
MSB of count—Counter 0	0	0	Control Word—Counter 0	1	1
Control Word—Counter 1	1	1	LSB of count—Counter 2	1	0
LSB of count—Counter 1	0	1	MSB of count—Counter 2	1	0
MSB of count—Counter 1	0	1	LSB of count—Counter 1	0	1
Control Word—Counter 2	1	1	MSB of count—Counter 1	0	1
LSB of count—Counter 2	1	0	LSB of count—Counter 0	0	0
MSB of count—Counter 2	1	0	MSB of count—Counter 0	0	0

	A_1	A_0		A_1	A_0
Control Word—Counter 0	1	1	Control Word—Counter 1	1	1
Control Word—Counter 1	1	1	Control Word—Counter 0	1	1
Control Word—Counter 2	1	1	LSB of count—Counter 1	0	1
LSB of count—Counter 2	1	0	Control Word—Counter 2	1	1
LSB of count—Counter 1	0	1	LSB of count—Counter 0	0	0
LSB of count—Counter 0	0	0	MSB of count—Counter 1	0	1
MSB of count—Counter 0	0	0	LSB of count—Counter 2	1	0
MSB of count—Counter 1	0	1	MSB of count—Counter 0	0	0
MSB of count—Counter 2	1	0	MSB of count—Counter 2	1	0

NOTE:
In all four examples, all Counters are programmed to read/write two-byte counts. These are only four of many possible programming sequences.

Figure 8. A Few Possible Programming Sequences

Read Operations

It is often desirable to read the value of a Counter without disturbing the count in progress. This is easily done in the 8254.

There are three possible methods for reading the counters: a simple read operation, the Counter Latch Command, and the Read-Back Command. Each is explained below. The first method is to perform a simple read operation. To read the Counter, which is selected with the A1, A0 inputs, the CLK input of the selected Counter must be inhibited by using either the GATE input or external logic. Otherwise, the count may be in the process of changing when it is read, giving an undefined result.

COUNTER LATCH COMMAND

The second method uses the "Counter Latch Command". Like a Control Word, this command is written to the Control Word Register, which is selected when $A_1,A_0 = 11$. Also like a Control Word, the SC0, SC1 bits select one of the three Counters, but two other bits, D5 and D4, distinguish this command from a Control Word.

$A_1,A_0 = 11$; CS = 0; RD = 1; WR = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	0	0	X	X	X	X

SC1,SC0—specify counter to be latched

SC1	SC0	Counter
0	0	0
0	1	1
1	0	2
1	1	Read-Back Command

D5,D4—00 designates Counter Latch Command

X—don't care

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Figure 9. Counter Latching Command Format

The selected Counter's output latch (OL) latches the count at the time the Counter Latch Command is received. This count is held in the latch until it is read by the CPU (or until the Counter is reprogrammed). The count is then unlatched automatically and the OL returns to "following" the counting element (CE). This allows reading the contents of the Counters "on the fly" without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one Counter. Each latched Counter's OL holds its count until it is read. Counter Latch Commands do not affect the programmed Mode of the Counter in any way.

If a Counter is latched and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read will be the count at the time the first Counter Latch Command was issued.

With either method, the count must be read according to the programmed format; specifically, if the Counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other; read or write or programming operations of other Counters may be inserted between them.

Another feature of the 8254 is that reads and writes of the same Counter may be interleaved; for example, if the Counter is programmed for two byte counts, the following sequence is valid.

1) Read least significant byte.
2) Write new least significant byte.
3) Read most significant byte.
4) Write new most significant byte.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between reading the first and second byte to another routine which also reads from that same Counter. Otherwise, an incorrect count will be read.

READ-BACK COMMAND

The third method uses the Read-Back Command. This command allows the user to check the count value, programmed Mode, and current states of the OUT pin and Null Count flag of the selected counter(s).

The command is written into the Control Word Register and has the format shown in Figure 10. The command applies to the counters selected by setting their corresponding bits D3, D2, D1 = 1.

A0, A1 = 11 $\overline{CS}$ = 0 $\overline{RD}$ = 1 $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	$\overline{COUNT}$	$\overline{STATUS}$	CNT 2	CNT 1	CNT 0	0

D_5: 0 = Latch count of selected counter(s)
D_4: 0 = Latch status of selected counters(s)
D_3: 1 = Select Counter 2
D_2: 1 = Select Counter 1
D_1: 1 = Select Counter 0
D_0: Reserved for future expansion; Must be 0

Figure 10. Read-Back Command Format

The read-back command may be used to latch multiple counter output latches (OL) by setting the $\overline{COUNT}$ bit D5 = 0 and selecting the desired counter(s). This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read (or the counter is reprogrammed). The counter is automatically unlatched when read, but other counters remain latched until they are read. If multiple count read-back commands are issued to the same counter without reading the count, all but the first are ignored; i.e., the count which will be read is the count at the time the first read-back command was issued.

The read-back command may also be used to latch status information of selected counter(s) by setting $\overline{STATUS}$ bit D4 = 0. Status must be latched to be read; status of a counter is accessed by a read from that counter.

The counter status format is shown in Figure 11. Bits D5 through D0 contain the counter's programmed Mode exactly as written in the last Mode Control Word. OUTPUT bit D7 contains the current state of the OUT pin. This allows the user to monitor the counter's output via software, possibly eliminating some hardware from a system.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Output	Null Count	RW1	RW0	M2	M1	M0	BCD

D_7 1 = OUT Pin is 1
0 = OUT Pin is 0

D_6 1 = Null Count
0 = Count available for reading

D_5–D_0 Counter programmed mode (see Figure 7)

Figure 11. Status Byte

NULL COUNT bit D6 indicates when the last count written to the counter register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the Mode of the counter and is described in the Mode Definitions, but until the count is loaded into the counting element (CE), it can't be read from the counter. If the count is latched or read before this time, the count value will not reflect the new count just written. The operation of Null Count is shown in Figure 12.

This Action	Causes
A. Write to the control word register;(1)	Null Count = 1
B. Write to the count register (CR);(2)	Null Count = 1
C. New Count is loaded into CE (CR → CE);	Null Count = 0

NOTE:
1. Only the counter specified by the control word will have its Null Count set to 1. Null count bits of other counters are unaffected.
2. If the counter is programmed for two-byte counts (least significant byte then most significant byte) Null Count goes to 1 when the second byte is written.

Figure 12. Null Count Operation

If multiple status latch operations of the counter(s) are performed without reading the status, all but the first are ignored; i.e., the status that will be read is the status of the counter at the time the first status read-back command was issued.

Both count and status of the selected counter(s) may be latched simultaneously by setting both $\overline{\text{COUNT}}$ and $\overline{\text{STATUS}}$ bits D5,D4 = 0. This is functionally the same as issuing two separate read-back commands at once, and the above discussions apply here also. Specifically, if multiple count and/or status read-back commands are issued to the same counter(s) without any intervening reads, all but the first are ignored. This is illustrated in Figure 13.

If both count and status of a counter are latched, the first read operation of that counter will return latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two type counts) return latched count. Subsequent reads return unlatched count.

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Write into Counter 0
0	1	0	0	1	Write into Counter 1
0	1	0	1	0	Write into Counter 2
0	1	0	1	1	Write Control Word
0	0	1	0	0	Read from Counter 0
0	0	1	0	1	Read from Counter 1
0	0	1	1	0	Read from Counter 2
0	0	1	1	1	No-Operation (3-State)
1	X	X	X	X	No-Operation (3-State)
0	1	1	X	X	No-Operation (3-State)

Figure 14. Read/Write Operations Summary

Command								Description	Result
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
1	1	0	0	0	0	1	0	Read back count and status of Counter 0	Count and status latched for Counter 0
1	1	1	0	0	1	0	0	Read back status of Counter 1	Status latched for Counter 1
1	1	1	0	1	1	0	0	Read back status of Counters 2, 1	Status latched for Counter 2, but not Counter 1
1	1	0	1	1	0	0	0	Read back count of Counter 2	Count latched for Counter 2
1	1	0	0	0	1	0	0	Read back count and status of Counter 1	Count latched for Counter 1, but not status
1	1	1	0	0	0	1	0	Read back status of Counter 1	Command ignored, status already latched for Counter 1

Figure 13. Read-Back Command Example

Mode Definitions

The following are defined for use in describing the operation of the 8254.

CLK Pulse: a rising edge, then a falling edge, in that order, of a Counter's CLK input.

Trigger: a rising edge of a Counter's GATE input.

Counter loading: the transfer of a count from the CR to the CE (refer to the "Functional Description")

MODE 0: INTERRUPT ON TERMINAL COUNT

Mode 0 is typically used for event counting. After the Control Word is written, OUT is initially low, and will remain low until the Counter reaches zero. OUT then goes high and remains high until a new count or a new Mode 0 Control Word is written into the Counter.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After the Control Word and initial count are written to a Counter, the initial count will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not go high until N + 1 CLK pulses after the initial count is written.

If a new count is written to the Counter, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte disables counting. OUT is set low immediately (no clock pulse required)
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the counting sequence to be synchronized by software. Again, OUT does not go high until N+1 CLK pulses after the new count of N is written.

If an initial count is written while GATE = 0, it will still be loaded on the next CLK pulse. When GATE goes high, OUT will go high N CLK pulses later; no CLK pulse is needed to load the Counter as this has already been done.

MODE 1: HARDWARE RETRIGGERABLE ONE-SHOT

OUT will be initially high. OUT will go low on the CLK pulse following a trigger to begin the one-shot pulse, and will remain low until the Counter reaches zero. OUT will then go high and remain high until the CLK pulse after the next trigger.

After writing the Control Word and initial count, the Counter is armed. A trigger results in loading the Counter and setting OUT low on the next CLK pulse, thus starting the one-shot pulse. An initial count of N will result in a one-shot pulse N CLK cycles in duration. The one-shot is retriggerable, hence OUT will remain low for N CLK pulses after any trigger. The one-shot pulse can be repeated without rewriting the same count into the counter. GATE has no effect on OUT.

If a new count is written to the Counter during a one-shot pulse, the current one-shot is not affected unless the counter is retriggered. In that case, the Counter is loaded with the new count and the one-shot pulse continues until the new count expires.

MODE 2: RATE GENERATOR

This Mode functions like a divide-by-N counter. It is typically used to generate a Real Time Clock interrupt. OUT will initially be high. When the initial count has decremented to 1, OUT goes low for one CLK pulse. OUT then goes high again, the Counter reloads the initial count and the process is repeated. Mode 2 is periodic; the same sequence is repeated indefinitely. For an initial count of N, the sequence repeats every N CLK cycles.

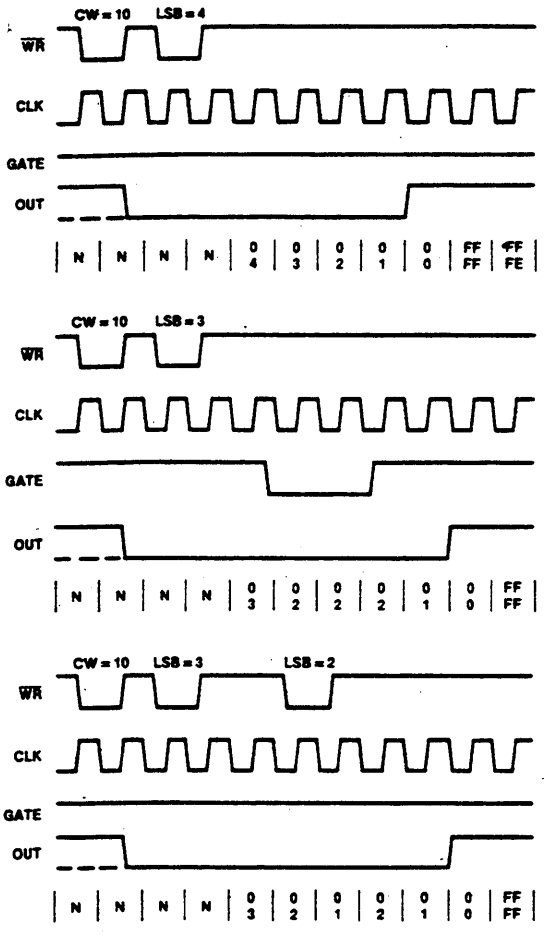
GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low during an output pulse, OUT is set high immediately. A trigger reloads the Counter with the initial count on the next CLK pulse; OUT goes low N CLK pulses after the trigger. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. OUT goes low N CLK Pulses after the initial count is written. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current period, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current counting cycle. In mode 2, a COUNT of 1 is illegal.

MODE 3: SQUARE WAVE MODE

Mode 3 is typically used for Baud rate generation. Mode 3 is similar to Mode 2 except for the duty cycle of OUT. OUT will initially be high. When half the



NOTE:
The following conventions apply to all mode timing diagrams:
1. Counters are programmed for binary (not BCD) counting and for reading/writing least significant byte (LSB) only.
2. The counter is always selected ($\overline{CS}$ always low).
3. CW stands for "Control Word"; CW = 10 means a control word of 10 HEX is written to the counter.
4. LSB stands for "Least Significant Byte" of count.
5. Numbers below diagrams are count values. The lower number is the least significant byte. The upper number is the most significant byte. Since the counter is programmed to read/write LSB only, the most significant byte cannot be read.
   N stands for an undefined count.
   Vertical lines show transitions between count values.

Figure 15. Mode 0

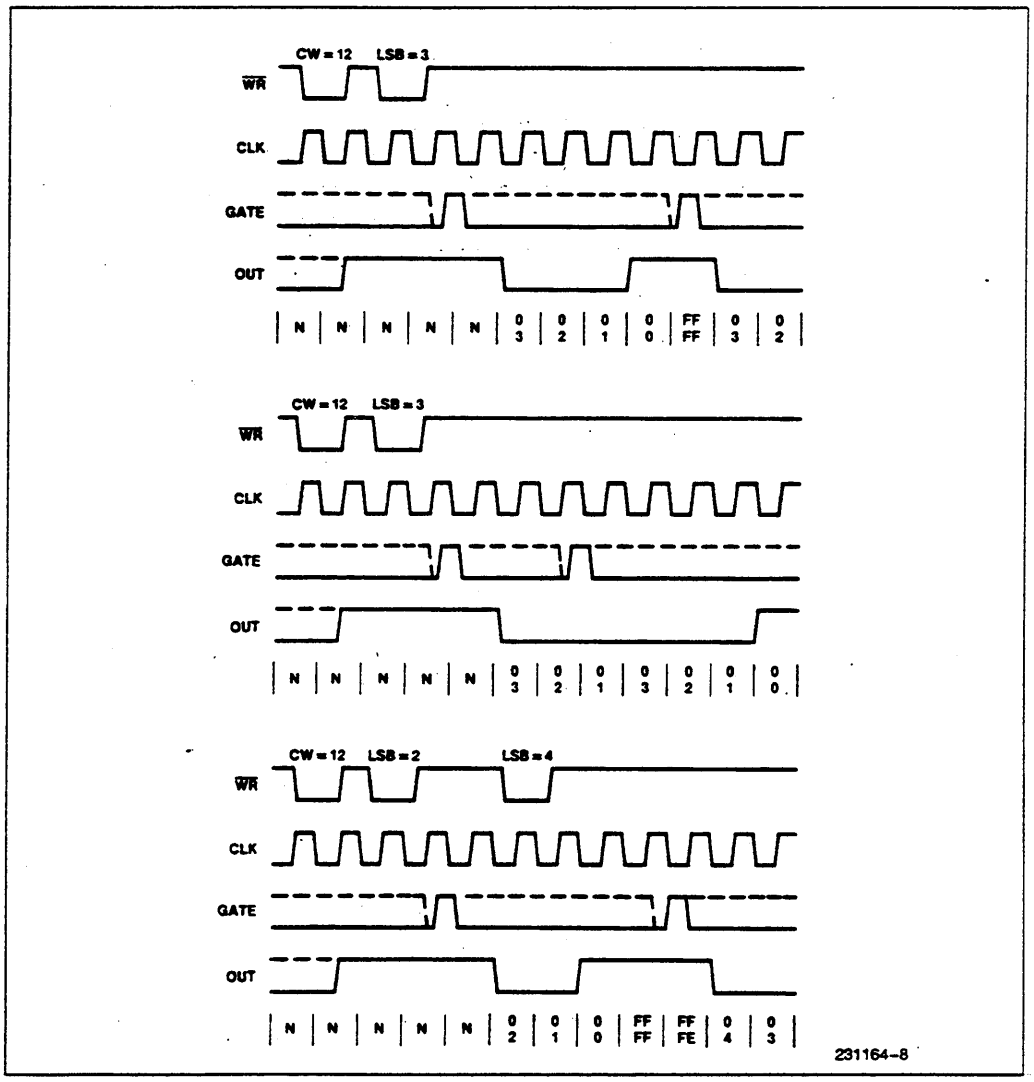


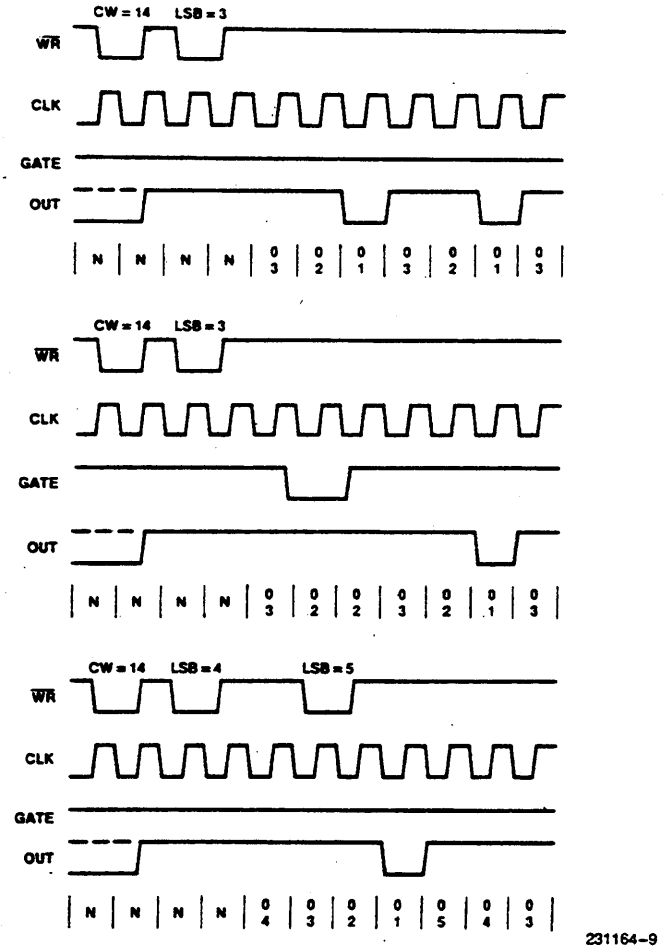
Figure 16. Mode 1

initial count has expired, OUT goes low for the remainder of the count. Mode 3 is periodic; the sequence above is repeated indefinitely. An initial count of N results in a square wave with a period of N CLK cycles.

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low while OUT is low, OUT is set high immediately; no CLK pulse is required. A trigger reloads the Counter with the initial count on the next CLK pulse. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current half-cycle of the square wave, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the



NOTE:
A GATE transition should not occur one clock prior to terminal count.

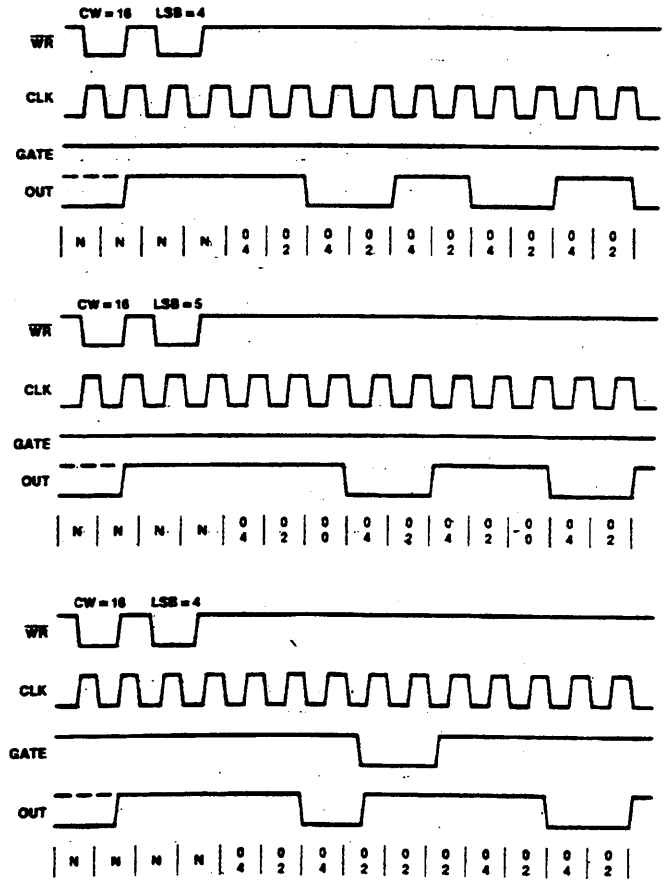
Figure 17. Mode 2

new count. Otherwise, the new count will be loaded at the end of the current half-cycle.

Mode 3 is implemented as follows:

Even counts: OUT is initially high. The initial count is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. When the count expires OUT changes value and the Counter is reloaded with the initial count. The above process is repeated indefinitely.

Odd counts: OUT is initially high. The initial count minus one (an even number) is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. One CLK pulse *after* the count expires, OUT goes low and the Counter is reloaded with the initial count minus one. Succeeding CLK pulses decrement the count by two. When the count expires, OUT goes high again and the Counter is reloaded with the initial count minus one. The above process is repeated indefinitely. So for odd counts, OUT will be high for (N + 1)/2 counts and low for (N − 1)/2 counts.



NOTE:
A GATE transition should not occur one clock prior to terminal count.

Figure 18. Mode 3

MODE 4: SOFTWARE TRIGGERED STROBE

OUT will be initially high. When the initial count expires, OUT will go low for one CLK pulse and then go high again. The counting sequence is "triggered" by writing the initial count.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N + 1 CLK pulses after the initial count is written.

If a new count is written during counting, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte has no effect on counting.
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the sequence to be "retriggered" by software. OUT strobes low N + 1 CLK pulses after the new count of N is written.

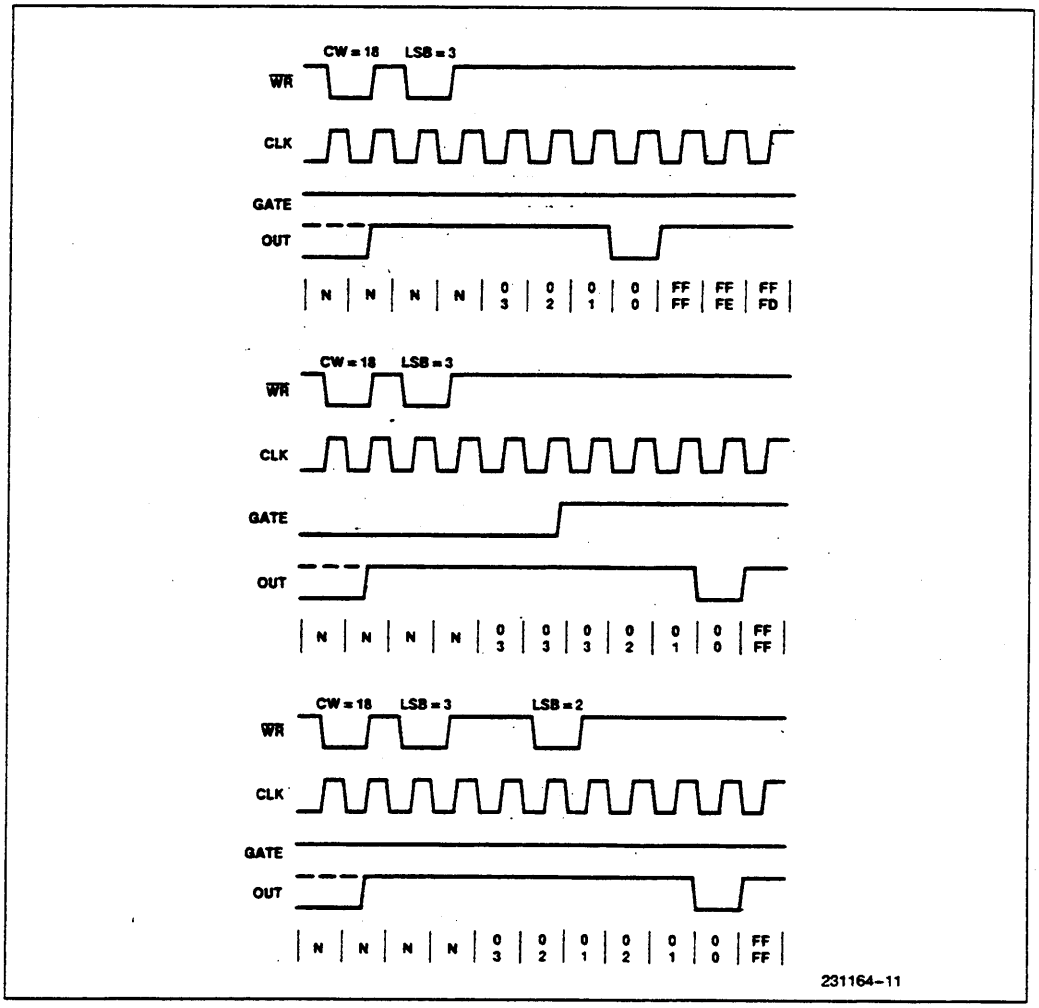


Figure 19. Mode 4

MODE 5: HARDWARE TRIGGERED STROBE (RETRIGGERABLE)

OUT will initially be high. Counting is triggered by a rising edge of GATE. When the initial count has expired, OUT will go low for one CLK pulse and then go high again.

After writing the Control Word and initial count, the counter will not be loaded until the CLK pulse after a trigger. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N + 1 CLK pulses after a trigger.

A trigger results in the Counter being loaded with the initial count on the next CLK pulse. The counting sequence is retriggerable. OUT will not strobe low for N + 1 CLK pulses after any trigger. GATE has no effect on OUT.

If a new count is written during counting, the current counting sequence will not be affected. If a trigger occurs after the new count is written but before the current count expires, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from there.

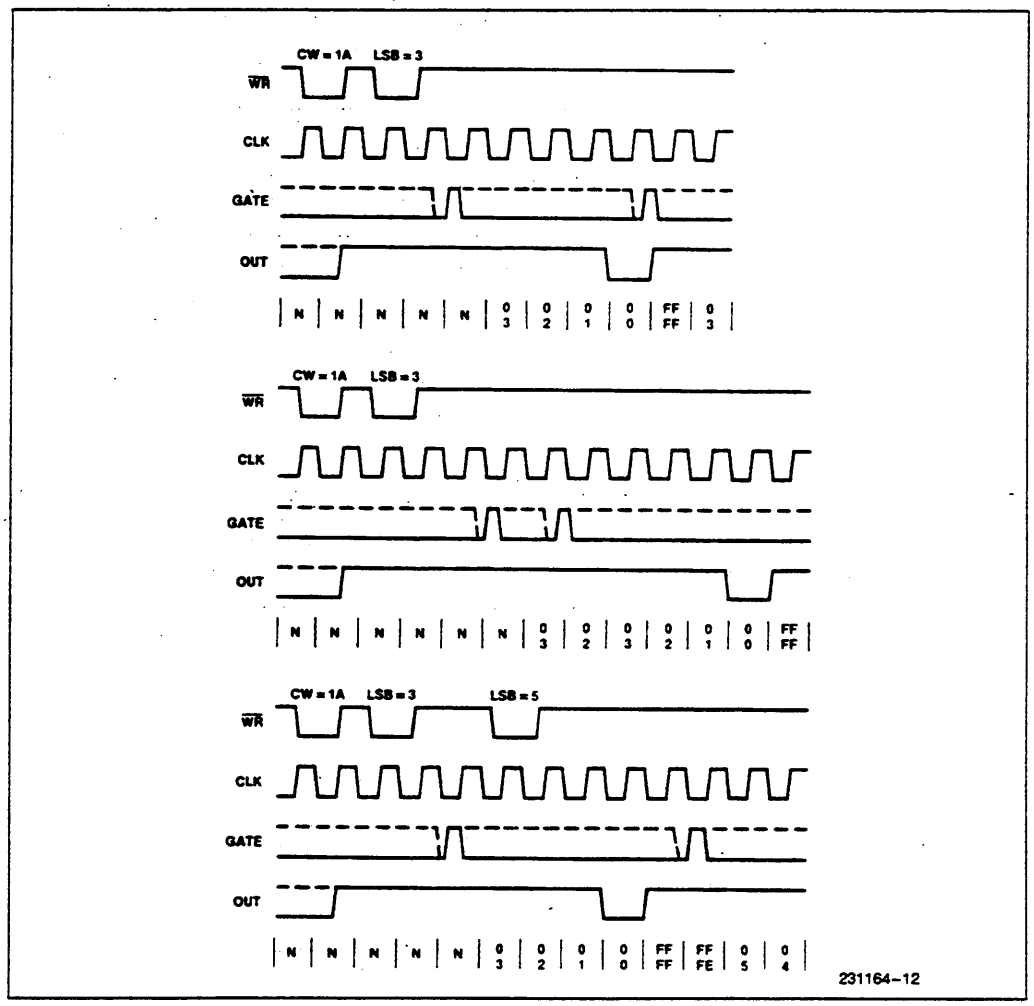


Figure 20. Mode 5

Signal Status Modes	Low Or Going Low	Rising	High
0	Disables Counting	— —	Enables Counting
1	— —	1) Initiates Counting 2) Resets Output after Next Clock	— —
2	1) Disables Counting 2) Sets Output Immediately High	Initiates Counting	Enables Counting
3	1) Disables Counting 2) Sets Output Immediately High	Initiates Counting	Enables Counting
4	Disables Counting	— —	Enables Counting
5	— —	Initiates Counting	— —

Figure 21. Gate Pin Operations Summary

Mode	Min Count	Max Count
0	1	0
1	1	0
2	2	0
3	2	0
4	1	0
5	1	0

NOTE:
0 is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

Figure 22. Minimum and Maximum Initial Counts

Operation Common to All Modes

PROGRAMMING

When a Control Word is written to a Counter, all Control Logic is immediately reset and OUT goes to a known initial state; no CLK pulses are required for this.

GATE

The GATE input is always sampled on the rising edge of CLK. In Modes 0, 2, 3, and 4 the GATE input is level sensitive, and the logic level is sampled on the rising edge of CLK. In Modes 1, 2, 3, and 5 the GATE input is rising-edge sensitive. In these Modes, a rising edge of GATE (trigger) sets an edge-sensitive flip-flop in the Counter. This flip-flop is then sampled on the next rising edge of CLK; the flip-flop is reset immediately after it is sampled. In this way, a trigger will be detected no matter when it occurs—a high logic level does not have to be maintained until the next rising edge of CLK. Note that in Modes 2 and 3, the GATE input is both edge- and level-sensitive. In Modes 2 and 3, if a CLK source other than the system clock is used, GATE should be pulsed immediately following $\overline{WR}$ of a new count value.

COUNTER

New counts are loaded and Counters are decremented on the falling edge of CLK.

The largest possible initial count is 0; this is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

The Counter does not stop when it reaches zero. In Modes 0, 1, 4, and 5 the Counter "wraps around" to the highest count, either FFFF hex for binary counting or 9999 for BCD counting, and continues counting. Modes 2 and 3 are periodic; the Counter reloads itself with the initial count and continues counting from there.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin with Respect to Ground.....−0.5V to +7V
Power Dissipation 1W

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} +0.5V	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0.45V
I_{CC}	V_{CC} Supply Current		170	mA	
C_{IN}	Input Capacitance		10	pF	f_c = 1 MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins returned to V_{SS}(4)

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%, GND = 0V

Bus Parameters(1)

READ CYCLE

Symbol	Parameter	8254-5		8254		8254-2		Unit
		Min	Max	Min	Max	Min	Max	
t_{AR}	Address Stable Before $\overline{RD}$ ↓	45		45		30		ns
t_{SR}	$\overline{CS}$ Stable Before $\overline{RD}$ ↓	0		0		0		ns
t_{RA}	Address Hold Time After $\overline{RD}$ ↑	0		0		0		ns
t_{RR}	$\overline{RD}$ Pulse Width	150		150		95		ns
t_{RD}	Data Delay from $\overline{RD}$ ↓		120		120		85	ns
t_{AD}	Data Delay from Address		220		220		185	ns
t_{DF}	$\overline{RD}$ ↑ to Data Floating	5	90	5	90	5	65	ns
t_{RV}	Command Recovery Time	200		200		165		ns

NOTE:
1. AC timings measured at V_{OH} = 2.0V, V_{OL} = 0.8V.

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%, GND = 0V (Continued)

WRITE CYCLE

Symbol	Parameter	8254-5		8254		8254-2		Unit
		Min	Max	Min	Max	Min	Max	
t_{AW}	Address Stable Before $\overline{WR}$ ↓	0		0		0		ns
t_{SW}	$\overline{CS}$ Stable Before $\overline{WR}$ ↓	0		0		0		ns
$t_{\overline{WA}}$	Address Hold Time After $\overline{WR}$ ↓	0		0		0		ns
t_{WW}	$\overline{WR}$ Pulse Width	150		150		95		ns
t_{DW}	Data Setup Time Before $\overline{WR}$ ↑	120		120		95		ns
t_{WD}	Data Hold Time After $\overline{WR}$ ↑	0		0		0		ns
t_{RV}	Command Recovery Time	200		200		165		ns

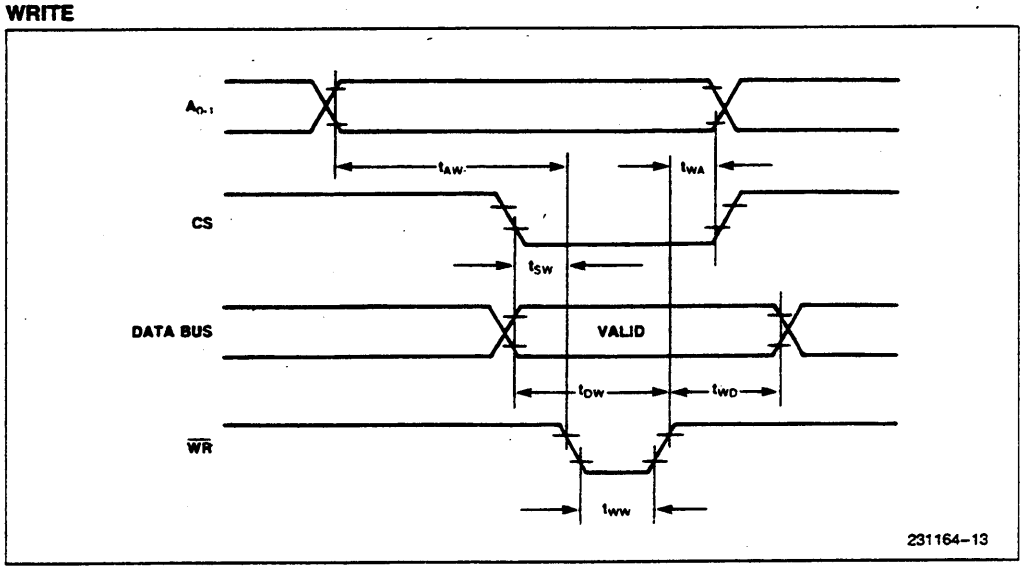
CLOCK AND GATE

Symbol	Parameter	8254-5		8254		8254-2		Unit
		Min	Max	Min	Max	Min	Max	
t_{CLK}	Clock Period	200	DC	125	DC	100	DC	ns
t_{PWH}	High Pulse Width	60(3)		60(3)		30(3)		ns
t_{PWL}	Low Pulse Width	60(3)		60(3)		50(3)		ns
t_R	Clock Rise Time		25		25		25	ns
t_F	Clock Fall Time		25		25		25	ns
t_{GW}	Gate Width High	50		50		50		ns
t_{GL}	Gate Width Low	50		50		50		ns
t_{GS}	Gate Setup Time to CLK ↑	50		50		40		ns
t_{GH}	Gate Setup Time After CLK ↑	50(2)		50(2)		50(2)		ns
t_{OD}	Output Delay from CLK ↓		150		150		100	ns
t_{ODG}	Output Delay from Gate ↓		120		120		100	ns
t_{WC}	CLK Delay for Loading ↓	0	55	0	55	0	55	ns
t_{WG}	Gate Delay for Sampling	−5	50	−5	50	−5	40	ns
t_{WO}	OUT Delay from Mode Write		260		260		240	ns
t_{CL}	CLK Set Up for Count Latch	−40	45	−40	45	−40	40	ns

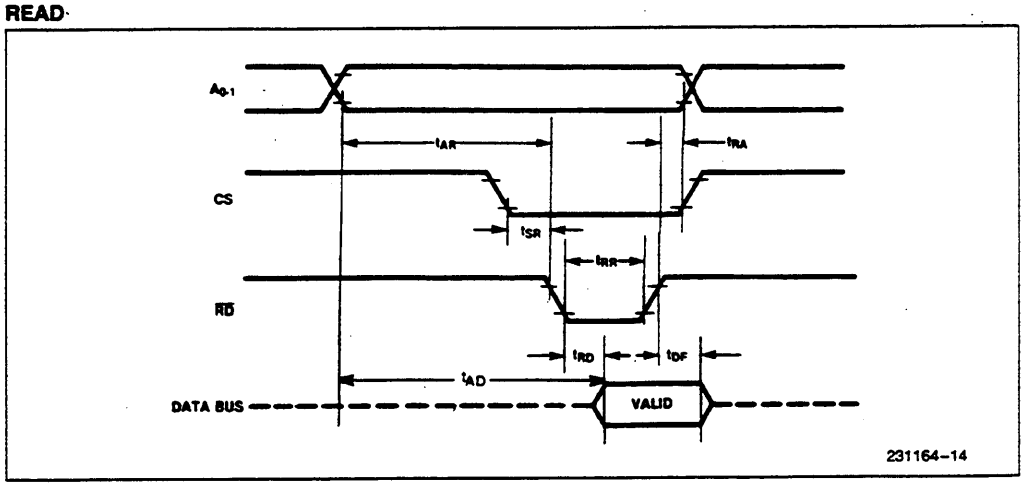
NOTES:
2. In Modes 1 and 5 triggers are sampled on each rising clock edge. A second trigger within 120 ns (70 ns for the 8254-2) of the rising clock edge may not be detected.
3. Low-going glitches that violate t_{PWH}, t_{PWL} may cause errors requiring counter reprogramming.
4. Sampled, not 100% tested. T_A = 25°C.
5. If CLK present at TWC min then Count equals N+2 CLK pulses, TWC max equals Count N+1 CLK pulse. TWC min to TWC max, count will be either N+1 or N+2 CLK pulses.
6. In Modes 1 and 5, if GATE is present when writing a new Count value, at TWG min Counter will not be triggered, at TWG max Counter will be triggered.
7. If CLK present when writing a Counter Latch or ReadBack Command, at TCL min CLK will be reflected in count value latched, at TCL max CLK will not be reflected in the count value latched.

WAVEFORMS

WRITE

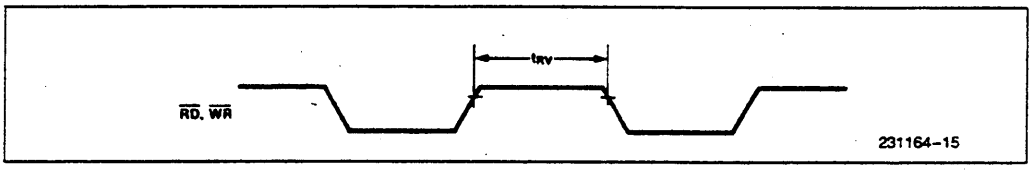


READ

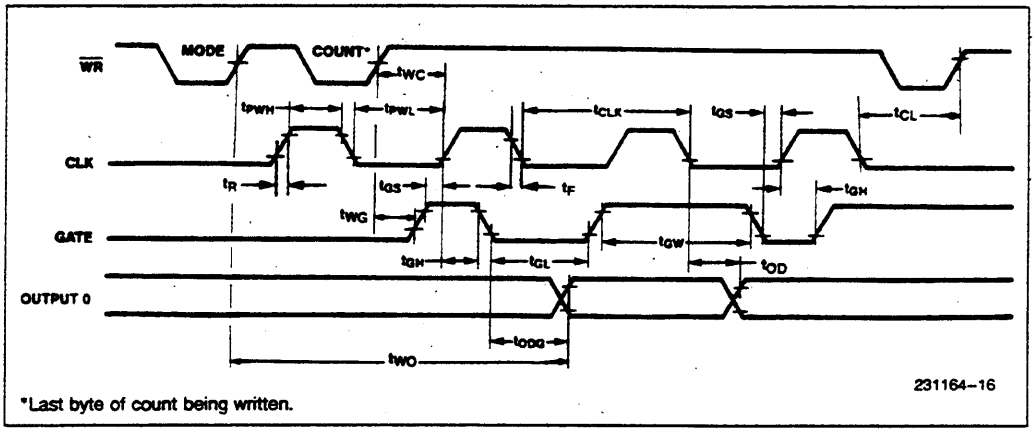


WAVEFORMS (Continued)

RECOVERY

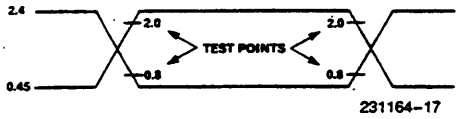


CLOCK AND GATE



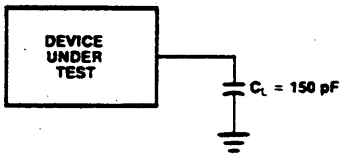
*Last byte of count being written.

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0." Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT



C_L = 150 pF
C_L Includes Jig Capacitance

Appendix E
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
Mexico	95 800 010 0793	95 800 010 0793
Netherlands	03480-33466	03480-30673
Norway	32-848400	32-848600
Singapore	2265886	2265887
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
Taiwan	02 377 1200	02 737 4644
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-DIO-32F 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

- Base I/O Address of AT-DIO-32F (Factory Setting: hex 240) _____
- DMA Channels of AT-DIO-32F (Factory Setting: 5 and 6) _____
- Interrupt Levels of AT-DIO-32F (Factory Setting: 11 and 12) _____
- RTSI Clock Setting (Factory Setting: disconnect board from RTSI clock) _____
- NI-DAQ or LabWindows Version _____
- Handshaking Mode _____

Other Products

- Computer Make and Model _____
- Microprocessor _____
- Clock Frequency _____
- Type of Video Board Installed _____
- DOS Version _____
- Programming Language _____
- Programming Language Version _____
- Other Boards in System _____
- Base I/O Address of Other Boards _____
- DMA Channels of Other Boards _____
- Interrupt Level 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-DIO-32F User Manual**

Edition Date: **April 1995**

Part Number: **320147-01**

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}

°	degrees
%	percent
A	amperes
AC	alternating current
AWG	American Wire Gauge
BCD	binary-coded decimal
C	Celsius
D/A	digital-to-analog
dB	decibels
DC	direct current
DMA	direct memory access
F	Farads
hex	hexadecimal
Hz	hertz
in.	inches
INTR*	Interrupt signal
I/O	input/output
I_{out}	output current
LSB	least significant bit
m	meters
MB	megabytes of memory
MSB	most significant bit
RAM	random-access memory
RTSI	Real-Time System Integration
s	seconds
V	volts
VDC	volts direct current
V_{cc}	positive voltage supply
V_{in}	volts in
W	watts

Index

Numbers

A

B

C

D

E

G

H

I

J

K

L

M

N

O

P

R

S

T

U

W

X