COMPREHENSIVE SERVICES

We offer competitive repair and calibration services, as well as easily accessible documentation and free downloadable resources.

SELL YOUR SURPLUS

We buy new, used, decommissioned, and surplus parts from every NI series. We work out the best solution to suit your individual needs.

Sell For Cash Get Credit Receive a Trade-In Deal

OBSOLETE NI HARDWARE IN STOCK & READY TO SHIP

We stock **New, New Surplus, Refurbished,** and **Reconditioned** NI Hardware.

APEX WAVES

Bridging the gap *between the manufacturer and your legacy test system.*

1-800-915-6216

www.apexwaves.com

sales@apexwaves.com

Request a Quote CLICK HERE *GPIB-PCIII*

DAQ

PC-TIO-10 User Manual

NATIONAL INSTRUMENTS™

April 1999 Edition
Part Number 371349A-01

Worldwide Technical Support and Product Information

www.natinst.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 794 0100

Worldwide Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 284 5011,
Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521, China 0755 3904939, Denmark 45 76 26 00,
Finland 09 725 725 11, France 01 48 14 24 24, Germany 089 741 31 30, Hong Kong 2645 3186,
India 91805275406, Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456,
Mexico (D.F.) 5 280 7625, Mexico (Monterrey) 8 357 7695, Netherlands 0348 433466, Norway 32 27 73 00,
Singapore 2265886, Spain (Madrid) 91 640 0085, Spain (Barcelona) 93 582 0251, Sweden 08 587 895 00,
Switzerland 056 200 51 51, Taiwan 02 2377 1200, United Kingdom 01635 523545

For further support information, see the *Technical Support Resources* appendix. To comment on the documentation, send e-mail to techpubs@natinst.com.

Important Information

Warranty

The PC-TIO-10 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 document 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

CVI™, LabVIEW™, `natinst.com`™, and NI-DAQ™ are trademarks of National Instruments Corporation.

Product and company names mentioned herein are trademarks or trade names of their respective companies.

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

National Instruments products are not designed with components and testing for a level of reliability suitable for use in or in connection with surgical implants or as critical components in any life support systems whose failure to perform can reasonably be expected to cause significant injury to a human. Applications of National Instruments products involving medical or clinical treatment can create a potential for death or bodily injury caused by product failure, or by errors on the part of the user or application designer. Because each end-user system is customized and differs from National Instruments testing platforms and because a user or application designer may use National Instruments products in combination with other products in a manner not evaluated or contemplated by National Instruments, the user or application designer is ultimately responsible for verifying and validating the suitability of National Instruments products whenever National Instruments products are incorporated in a system or application, including, without limitation, the appropriate design, process and safety level of such system or application.

Contents

About This Manual

Chapter 1
Introduction

Chapter 2
Configuration and Installation

Chapter 3
Theory of Operation

Chapter 4
Programming

Appendix A
Specifications

Appendix B
I/O Connector

Appendix C
AMD Am9513A Data Sheet

Appendix D
Motorola MC6821Data Sheet

Appendix E
Switch Settings

Appendix F
Technical Support Resources

Glossary

Index

Figures

Tables

About This Manual

Introduction to the PC-TIO-10

This manual describes the mechanical and electrical aspects of the PC-TIO-10 and contains information concerning its operation and programming. The PC-TIO-10 is a timing and digital I/O interface for the PC. Two Advanced Micro Devices (AMD) Am9513A System Timing Controllers (STCs) are used for the timing interface. With these chips, which feature many different timing and counting modes, the PC-TIO-10 can perform a wide range of pulse measurement and wave generation functions. A Motorola MC6821 Peripheral Interface Adapter (PIA) is used for the digital I/O interface; each of the two 8-bit I/O ports is bit-configurable. In addition, the PC-TIO-10 has two edge-sensitive interrupt inputs with programmable edge selection. Any external transistor-transistor logic (TTL) signal, including any of the counter outputs, can be connected to these interrupt inputs.

This manual describes installation, theory of operation, and basic programming considerations for the PC-TIO-10. The example programs included are written in C and assembly language.

Conventions

The following conventions appear in this manual:

<> Angle brackets that contain numbers separated by an ellipsis represent a range of values associated with a bit or signal name—for example, DBIO<3..0>.

» The **»** symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.

This icon denotes a note, which alerts you to important information.

This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.

	This icon denotes a warning, which advises you of precautions to take to avoid being electrically shocked.
italic	Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. This font also denotes text that is a placeholder for a word or value that you must supply.
`monospace`	Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and code excerpts.
NI-DAQ	NI-DAQ is used throughout this manual to refer to the NI-DAQ software for DOS/Windows/LabWindows unless otherwise noted.

Related Documentation

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

- *Am9513A/Am9513 System Timing Controller* technical manual
- Your computer's technical manual

1

Introduction

This chapter describes the PC-TIO-10, lists the contents of your PC-TIO-10 kit, lists the optional software and equipment for use with the PC-TIO-10, and explains how to unpack the PC-TIO-10 kit.

The PC-TIO-10 is a timing and digital I/O interface for the PC. Two AMD Am9513A STCs are used for the timing interface. With these chips, which feature many different timing and counting modes, the PC-TIO-10 can perform a wide range of pulse measurement and wave generation functions. A Motorola MC6821 PIA is used for the digital I/O interface; each of the two 8-bit I/O ports is bit-configurable. In addition, the PC-TIO-10 has two edge-sensitive interrupt inputs with programmable edge selection. Any external TTL signal, including any of the counter outputs, can be connected to these interrupt inputs.

The timing circuits on the board make the PC-TIO-10 useful for the following operations:

- Wave and pulse generation
- Frequency shift keying (FSK)
- Pulse-width measurement
- Time-of-day counting and alarm generation
- Event counting

The digital I/O lines on the PC-TIO-10 interface the PC to the following:

- BCD-compatible panel meters and test equipment
- Opto-isolated, solid-state relays and I/O module mounting racks

The PC-TIO-10 turns the PC into a timing and digital I/O system controller for applications in laboratory testing, production testing, and industrial process monitoring and control.

What Your Kit Should Contain

The contents of the PC-TIO-10 kit are listed as follows.

- PC-TIO-10 board
- PC-TIO-10 User Manual
- NI-DAQ software for DOS/Windows/LabWindows, with manuals

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

Your PC-TIO-10 is shipped with the NI-DAQ software. 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.

Optional Software

This manual contains complete instructions for directly programming the PC-TIO-10. 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 PC-TIO-10 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*.

There are several options to choose from when programming your National Instruments DAQ hardware. You can use LabVIEW, LabWindows/CVI, or NI-DAQ.

National Instruments Application Software

LabVIEW features interactive graphics, a state-of-the-art user interface, and a powerful graphical programming language. The LabVIEW Data Acquisition VI Library, a series of *virtual instruments* (VIs) for using LabVIEW with National Instruments DAQ hardware, is included with LabVIEW. The LabVIEW Data Acquisition VI Library is functionally equivalent to the NI-DAQ software.

LabWindows/CVI features interactive graphics and a state-of-the-art user interface, and uses the ANSI standard C programming language. The LabWindows/CVI Data Acquisition Library, a series of functions for using

National Instruments DAQ hardware, is included with LabVIEW/CVI. The LabWindows/CVI Data Acquisition VI Library is functionally equivalent to the NI-DAQ software.

Using LabVIEW or LabWindows/CVI software will greatly reduce the development time for your data acquisition and control application.

NI-DAQ Driver Software

The NI-DAQ driver software is included at no charge with all National Instruments DAQ hardware. NI-DAQ has an extensive library of functions that you can call from your application programming environment. These functions include routines for digital I/O, counter/timer operations, RTSI, and acquiring data to extended memory.

NI-DAQ has both high-level DAQ I/O functions for maximum ease of use and low-level DAQ I/O functions for maximum flexibility and performance. Examples for high-level functions are streaming data to disk or acquiring a certain number of data points. An example of a low-level function is writing directly to registers on the DAQ device. NI-DAQ does not sacrifice performance of National Instruments DAQ devices because it lets multiple devices operate at their peak performance, even simultaneously.

NI-DAQ also internally addresses many of the complex issues between the computer and the DAQ hardware such as programming interrupts and DMA controllers. NI-DAQ maintains a consistent software interface so that you can change platforms with minimal modifications to your code. Whether you are using conventional programming languages or NI-DAQ software, your application uses the NI-DAQ driver software.

Optional Equipment

The following is a list of optional equipment available for the PC-TIO-10:

- CB-50 I/O connector block—0.5 m cable
- Standard ribbon cable—0.5 m
- Standard ribbon cable—1.0 m
- Shielded ribbon cable—1.0 m
- Shielded ribbon cable—2.0 m

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

Unpacking

Your PC-TIO-10 board is shipped in an antistatic 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, touch the antistatic package to a metal part of your computer 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.

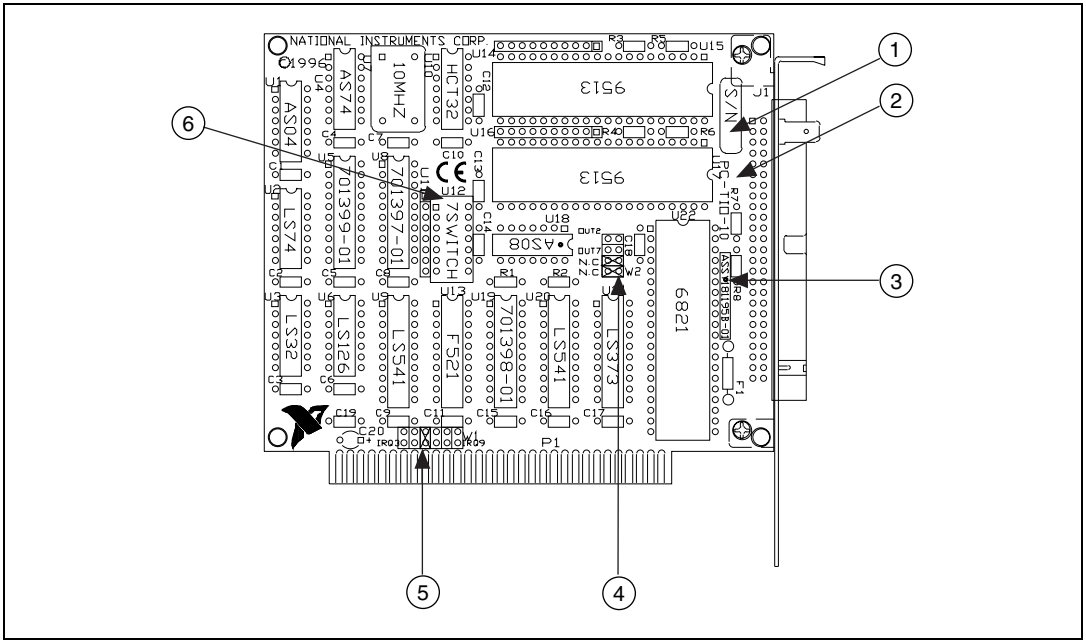
2

Configuration and Installation

This chapter describes the PC-TIO-10 jumper configurations, installation of the PC-TIO-10 board in your computer, signal connections to the PC-TIO-10 board, and cabling instructions.

Board Configuration

The PC-TIO-10 contains one DIP switch and two jumpers to configure the base I/O address and interrupts, respectively. The DIP switch and jumpers are shown in the parts locator diagram in Figure 2-1.



1	Serial Number	3	Assembly Number	5	W1
2	Product Name	4	W2	6	U12

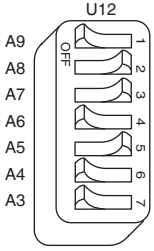
Figure 2-1. PC-TIO-10 Parts Locator Diagram

Address lines A9 through A0 are used to communicate with your PC-TIO-10. Address lines A9 through A3 determine the address of the board within your computer system. This address is called the base I/O address whereas address lines A2 through A0 are used by the PC-TIO-10 to decode accesses to the onboard registers. The positions of the switches at U12, as shown in Figure 2-1, determine the base I/O address for your PC-TIO-10. Each switch at U12 corresponds to one of the address lines A9 through A3.

The switches at U12 are set at the factory to provide a base I/O address of hex 1A0. With this default setting, the PC-TIO-10 uses the I/O address space hex 1A0 through 1A7. Similarly, the PC-TIO-10 has a factory default setting of interrupt level 5 while its local interrupt setting is set to no connect and no connect.

The settings, shown in Table 2-1, are suitable for most systems. However, if your system has other hardware at this base I/O address or interrupt level, you need to change these settings on the PC-TIO-10, as described in the following pages, or on the other hardware.

Table 2-1. PC-TIO-10 Factory-Set Switch and Jumper Settings

Need Head	Need Head	Need Head
Base I/O Address	Hex 1A0 (factory setting)	
Interrupt Level	Interrupt level 5 selected (factory setting)	**W1:** Row 5
Local Interrupt	No Connect and No Connect (factory setting)	**W2:** No Connect No Connect

The different permutations of A9 through A3 yield 128 different possible base I/O addresses in the range from hex 000 though 3F8. Appendix E, *Switch Settings*, lists the switch settings corresponding to these base addressees.

On the U12 DIP switches, 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. Suppose you wish to use Hex 228 as your base I/O address. The corresponding binary pattern is 1000101000 for A9 through A0. A9 through A3 should be set to 1000101 via the switch at U12 as shown in Figure 2-2B. Figure 2-2A shows the default factory setting for the switches at U12 that yields a base I/O address of hex 1A0.

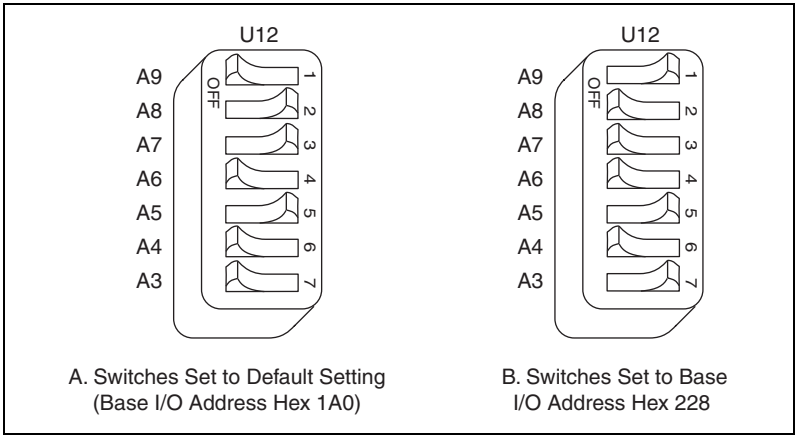


Figure 2-2. Example Base I/O Address Switch Settings

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

Board	DMA Channel	Interrupt Level	Base I/O Address
AT-AO-6/10	Channel 5	Lines 11, 12	1C0 hex
AT-DIO-32F	Channels 5, 6	Lines 11, 12	240 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
GPIB-PCII	Channel 1	Line 7	2B8 hex
GPIB-PCIIA	Channel 1	Line 7	2E1 hex
GPIB-PCIII	Channel 1	Line 7	280 hex

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

Board	DMA Channel	Interrupt Level	Base I/O Address
Lab-PC-1200	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.			

Interrupt Level Selection

There are two sets of jumpers for interrupt selection on the PC-TIO-10 board. W1 is used for selecting the interrupt level, while W2 is used for local selection of two of the counter outputs as interrupt sources. The locations of these jumpers are shown in Figure 2-1.

The PC-TIO-10 board can connect to any one of six interrupt lines of the PC I/O Channel: IRQ3, IRQ4, IRQ5, IRQ6, IRQ7, or IRQ9. You select the interrupt line by setting a jumper on W1. The default interrupt line is IRQ5. To change to another line, remove the jumper from IRQ5 and place it on the pins for another request line. Figure 2-3 shows the default factory setting for IRQ5.

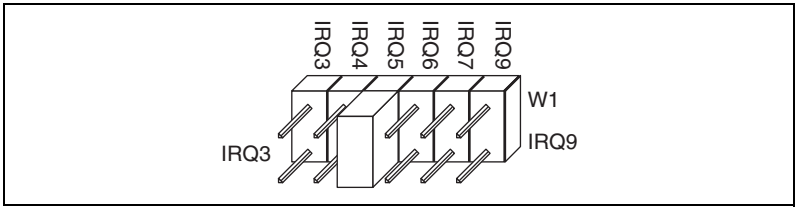


Figure 2-3. Interrupt Jumper Setting for IRQ5 (Factory Setting)

To disable the PC-TIO-10 interrupt request line, change the jumper setting as shown in Figure 2-4.

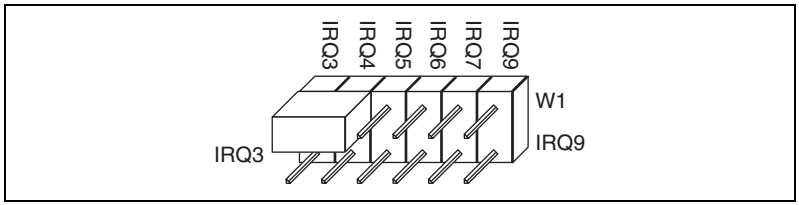


Figure 2-4. Interrupt Jumper Setting for Disabling Interrupts

Local Interrupt Selection

In addition to the jumpers for selecting the interrupt level used by the PC-TIO-10, a set of jumpers, W2, is used to locally connect two of the counter outputs to the interrupt generation circuitry. There are four positions on this set of jumpers: two No Connect positions, labelled N.C., a position for OUT2, and a position for OUT7. The position for OUT2 connects the output of counter 2 to the EXTIRQ1 input, while the position for OUT7 connects the output of counter 7 to the EXTIRQ2 input. The No Connect positions are intended as storage positions for one or both of the jumpers if you do not want to use one or both of the counter outputs for interrupt purposes. The default positions for the jumpers on W2 are shown in Figure 2-5.

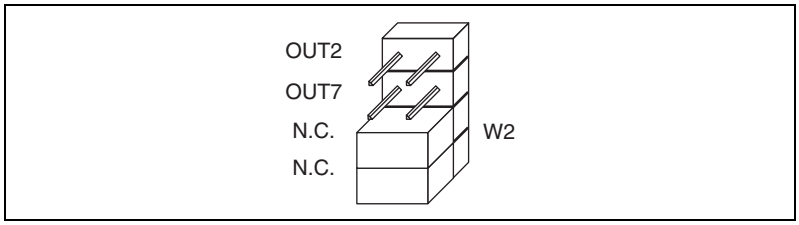


Figure 2-5. Local Interrupt Jumper Setting (Factory Setting)

OUT2 and OUT7 can be jumpered simultaneously. The interrupt for OUT2 is enabled and disabled through access to the port A interrupt-control circuitry of the MC6821 PIA. OUT7 is enabled and disabled through access to the port B interrupt-control circuitry of the MC6821 PIA. One or both of these interrupts can be asserted at any time if they are enabled. If both interrupts are enabled simultaneously, your interrupt handler must check both channels for interrupts before returning control to the foreground task. For more information, see Chapter 4, *Programming*.

Installation

The PC-TIO-10 can be installed in any unused ISA 8-bit, 16-bit, or 32-bit expansion slot in your computer. You are now ready to install the PC-TIO-10.

The following are general installation instructions, but consult the user manual or technical reference manual of your personal computer for specific instructions and warnings. If you want to install this board in an EISA-class computer, you can obtain a configuration file for the board by contacting National Instruments.

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 PC-TIO-10 in an unused ISA 8-bit, 16-bit, or 32-bit slot. It may be a tight fit, but *do not* force the board into place.
5. Screw the mounting bracket of the PC-TIO-10 to the back panel rail of the computer.
6. Check the installation.
7. Replace the cover to the computer.

Note If you have an ISA-class computer and you are using a configurable software package, such as NI-DAQ, you may need to reconfigure your software to reflect any changes in jumper or switch settings. If you have an EISA-class computer, you need to update the computer's resource allocation or configuration table by reconfiguring your computer. See your computer's user manual for information about updating the configuration table.

The PC-TIO-10 board is now installed and ready for operation.

Signal Connections

This section includes specifications and connection instructions for the signals given on the PC-TIO-10 I/O connector.

Caution Connections that exceed any of the maximum ratings of input or output signals on the PC-TIO-10 may result in damage to the PC-TIO-10 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.

I/O Connector Pin Description

Figure 2-6 show the pin assignments for the PC-TIO-10 I/O connector.

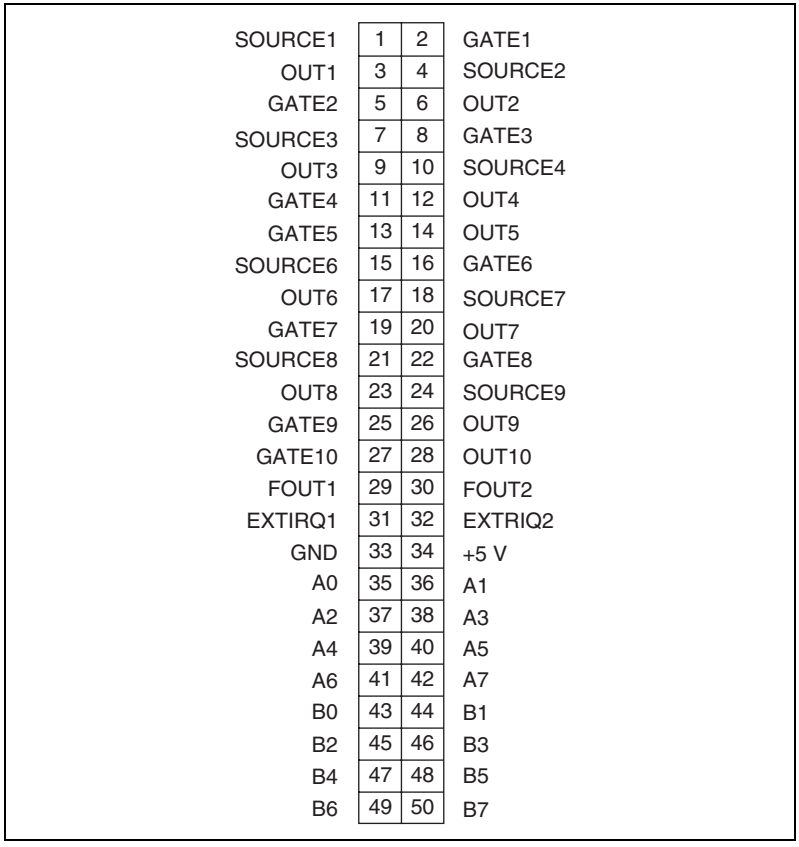


Figure 2-6. PC-TIO-10 I/O Connector Pin Assignments

Signal Connection Descriptions

Pins	Signal Names	Description
1, 4, 7, 1 10, 15, 18, 21, 24	SOURCE<1..4> SOURCE<6..9>	These are the source inputs for Counters through 4 and Counters 6 through 9. The source inputs for Counters 5 and 10 do not appear on the I/O connector because they are internally connected to a 5-MHz clock.
2, 5, 8, 11, 13, 16, 19, 22, 25, 27	GATE<1..10>	These are the gate inputs for Counters 1 through 10.
3, 6, 9, 12, 14, 17, 20, 23, 26, 28	OUT<1..10>	These are the outputs for Counters 1 through 10.
29-30	FOUT<1..2>	These are the frequency outputs of the two Am9513A devices.
31-32	EXTIRQ<1..2>	These are the interrupt inputs for the PC-TIO-10.
33	GND	This pin is connected to the computer's ground signal.
34	+5 V	This pin is connected to the computer’s +5 VDC power supply.
35-42	A<0..7>	These are the eight digital I/O lines on Port A of the MC6821. The MSB is A7.
43-50	B<0..7>	These are the eight digital I/O lines on Port B of the MC6821. The MSB is B7.

Timing Signal Connections

Pins 1 through 30 of the I/O connector are connections for timing I/O signals on the two onboard Am9513A Counter/Timers. The timing signals include the GATE, SOURCE, and OUT signals for the Am9513A Counters 1 through 10, and the FOUT1 and FOUT2 signals generated by the Am9513A STCs. Counters 1 through 10 of the Am9513A Counter/Timers can be used for general-purpose applications, such as pulse and square wave generation, event counting, and pulse-width, time-lapse, and frequency measurements. For these applications, SOURCE and GATE signals can be directly applied to the counters from the I/O connector, and the counters are programmed for various operations.

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

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

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

Figure 2-7 shows connections for a typical event-counting operation where a switch is used to gate the counter on and off.

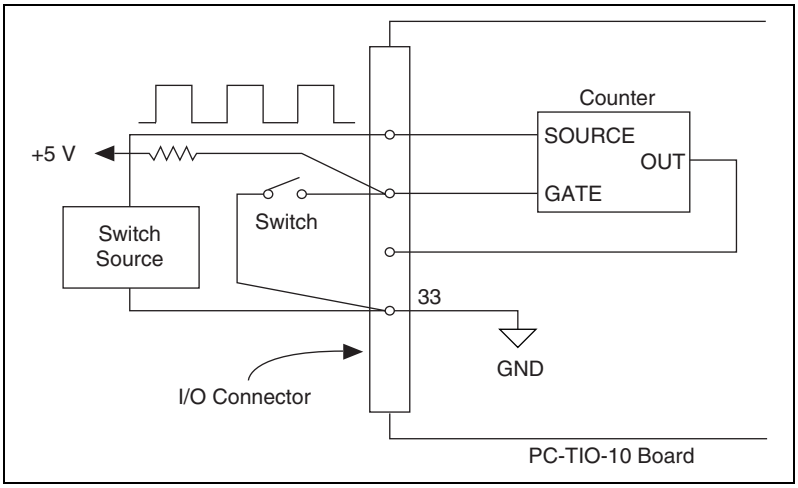


Figure 2-7. Event-Counting Application with External Switch Gating

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

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

To measure frequency, a counter is programmed to be level-gated and the rising or falling edges are counted in a signal applied to a SOURCE input. The gate signal applied to the counter GATE input is of some known duration. In this case, the counter is programmed to count either rising or falling edges at the SOURCE input while the gate is applied. The frequency of the input signal is then the count value divided by the known gate period. Figure 2-8 shows the connections for a frequency measurement application. A second counter can also be used to generate the gate signal in this application.

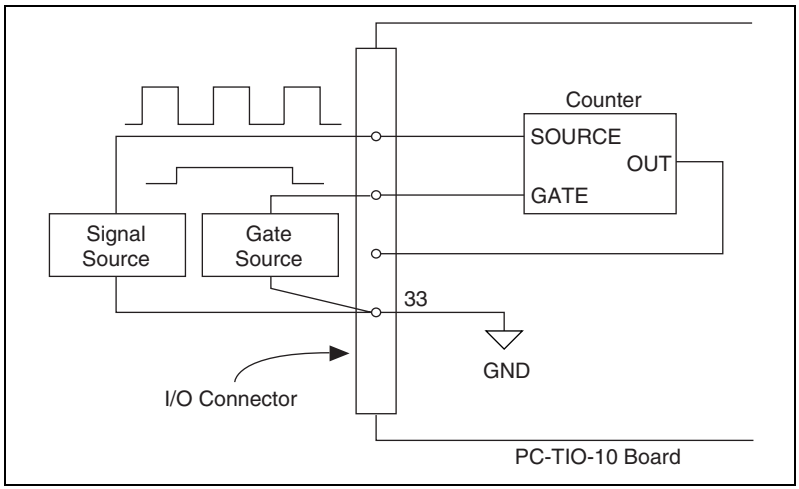


Figure 2-8. Frequency Measurement Application

Two or more counters can be concatenated by connecting the OUT signal from one counter to the SOURCE signal of another counter. The counters can then be treated as one 32-bit or larger counter for most counting applications. It is possible to create up to a 160-bit counter in this manner.

The GATE, SOURCE, OUT, and FOUT signals on the I/O connector are connected directly to the Am9513A input and output pins. The input and output ratings and timing specifications for the Am9513A signals are given as follows.

The following specifications and ratings apply to the Am9513A I/O signals.

Absolute maximum voltage rating –0.5 to +7.0 V with respect to GND

Am9513A Digital Input Specifications (referenced to GND):

	Minimum	Maximum
Input logic high voltage	2.0 V	5.25 V
Input logic low voltage	0.0 V	0.8 V
Input current ($0 < V_{in} < 5.25$ V)	–10 µA	10 µA

Am9513A Digital Output Specifications (referenced to GND):

	Minimum	Maximum
Output logic high voltage, all outputs at I_{out} = –200 µA	2.4 V	5.0 V
Output logic low voltage, all outputs at I_{out} = 3.2 mA	0.0 V	0.4 V

Figure 2-9 shows the timing requirements for the GATE and SOURCE input signals and the timing specifications for the OUT signals of the Am9513A STCs.

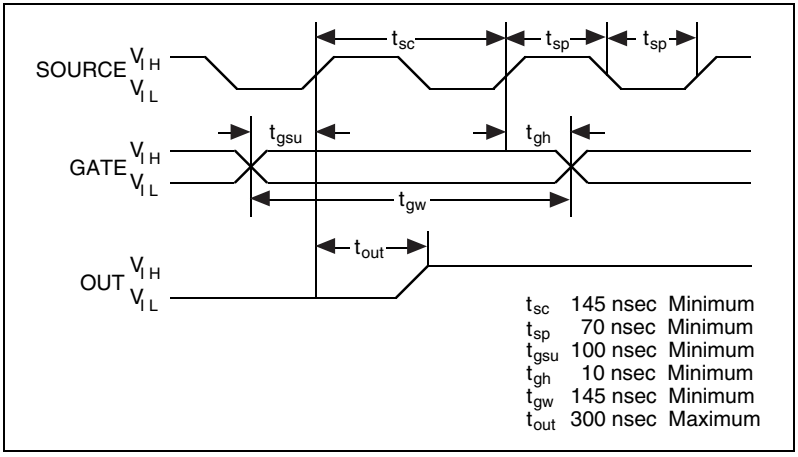


Figure 2-9. Timing Signal Relationships

The GATE and OUT signal transitions in Figure 2-9 are referenced to the rising edge of the SOURCE signal. This timing diagram assumes that the counters are programmed to count rising edges. The same timing diagram, with the source signal inverted and referenced to the falling edge of the source signal, applies to the case in which the counter is programmed to count falling edges.

The signal applied at a SOURCE input can be used as a clock source by any of the Am9513A counter/timers and by the Am9513A frequency division output FOUT. The signal applied to a SOURCE input must not exceed a frequency of 7 MHz for proper operation of the Am9513A. The Am9513A counters can be individually programmed to count rising or falling edges of signals applied at any of the Am9513A SOURCE or GATE input pins.

In addition to the signals applied to the SOURCE and GATE inputs, the Am9513A generates five internal timebase clocks from the clock signal supplied by the PC-TIO-10. The five internal timebase clocks can be used as counting sources, and these clocks have a maximum skew of 75 nsec between them. The SOURCE signal shown in Figure 2-9 represents any of the signals applied at the SOURCE inputs, GATE inputs, or internal timebase clocks. See Appendix C, *AMD Am9513A Data Sheet* for further details.

Specifications for signals at the GATE input are referenced to the signal at the SOURCE input or one of the Am9513A internally generated signals. Figure 2-9 shows the GATE signal referenced to the rising edge of a source signal. The gate must be valid, either high or low, at least 100 nsec before the rising or falling edge of a source signal for the gate to take effect at that source edge as shown by t_{gsu} and t_{gh} in Figure 2-9. Similarly, the gate signal must be held for at least 10 nsec after the rising or falling edge of a source signal for the gate to take effect at that source edge. The gate high or low period must be at least 145 nsec in duration. If an internal timebase clock is used, the gate signal cannot be synchronized with the clock. In this case, gates applied close to a source edge take effect either on that source edge or on the next one. This arrangement creates an uncertainty of one source clock period with respect to unsynchronized gating sources.

Signals generated at the OUT pin are referenced to the signal at the SOURCE input or to one of the Am9513A internally generated clock signals. Figure 2-9 shows the OUT signal referenced to the rising edge of a source signal. Any OUT signal state changes occur within 300 nsec after the source signal's rising or falling edge.

Digital I/O Signal Connections

Pins 31, 32, and 35 through 50 of the I/O connector are digital I/O signal pins.

Pins 35 through 42 are connected to the digital lines A<0..7> for digital I/O Port A. Pins 43 through 50 are connected to the digital lines B<0..7> for digital I/O Port B. Pins 31 and 32 are connected to the external interrupt lines, EXTIRQ1 and EXTIRQ2. Ports A and B can be programmed on a bitwise basis to be inputs or outputs.

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

Absolute maximum voltage rating –0.3 to +7.0 V with respect to GND

Digital Input Specifications (referenced to GND):

	Minimum	Maximum
Input logic high voltage	2.0 V	5.25 V
Input logic low voltage	0.0 V	0.8 V
Input current, Port A ($0 < V_{in} < 0.8$ V)	—	–2.4 mA
Input current, Port A ($2.0 < V_{in} < 5.25$ V)	—	–400 µA
Input current, Port B ($0.4 < V_{in} < 2.4$ V)	—	10 µA
Input current, EXTIRQ1 and EXTIRQ2 ($0 < V_{in} < 5.25$ V)	—	2.5 µA

Digital Output Specifications (referenced to GND):

	Minimum	Maximum
Output logic high voltage at I_{out} = –200 µA	2.4 V	5.0 V
Output logic low voltage at I_{out} = 3.2 mA	0.0 V	0.4 V
Darlington drive current, Port B at V_{EXT} = 1.5 V	–1.0 mA	–10.0 mA

Figure 2-10 depicts signal connections for three typical digital I/O applications.

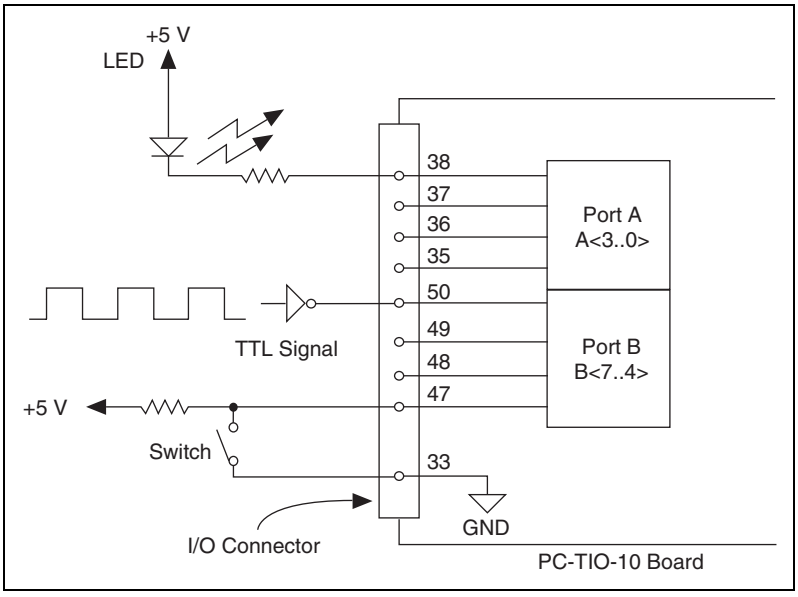


Figure 2-10. Digital I/O Connections

In Figure 2-10, port A is configured for digital output, and port B is configured for digital input. Digital input applications include receiving TTL signals and sensing external device states such as the state of the switch in Figure 2-10. Digital output applications include sending TTL signals and driving external devices such as the LED shown in Figure 2-10.

Power Connections

Pin 34 of the I/O connector is connected to the +5 V supply from the PC power supply. This pin is referenced to GND and can be used to power external digital circuitry. For more information on this output pin, see *Output Signal Specifications* in Appendix A, *Specifications*.

Power Rating	1.0 A at +5 V ± 10%

Caution *Under no circumstances* should this +5-V power pin be connected directly to ground or to any other voltage source on the PC-TIO-10 or any other device. Doing so may damage the PC-TIO-10 and the PC. National Instruments is not liable for damage resulting from such a connection.

Cabling

The PC-TIO-10 digital I/O connector is a standard, 50-pin, header connector, which can be interfaced using 50-pin ribbon cable with appropriate connectors. Signal input and output wires can be attached to screw terminals on the connector block and are therefore connected to the PC-TIO-10 I/O connector.

The CB-50 is useful for initial prototyping of an application or in situations where PC-TIO-10 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 for the design of custom cables.

The PC-TIO-10 I/O connector is a 50-pin, male, ribbon-cable header connector. The manufacturer and the appropriate part number for this connector is as follows:

- 3M (part number 2550-5002UB)

The mating connector for the PC-TIO-10 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 PC-TIO-10. The manufacturer and the appropriate part number for this mating connector is as follows:

- 3M (part number 3425-7650)

The manufacturer part number for the standard ribbon cable (50-conductor, 28 AWG, stranded) that can be used with these connectors is as follows:

- 3M (part number 3365/50)

3

Theory of Operation

This chapter explains the basic operation of the PC-TIO-10 circuitry.

The block diagram in Figure 3-1 illustrates the key functional components of the PC-TIO-10 board.

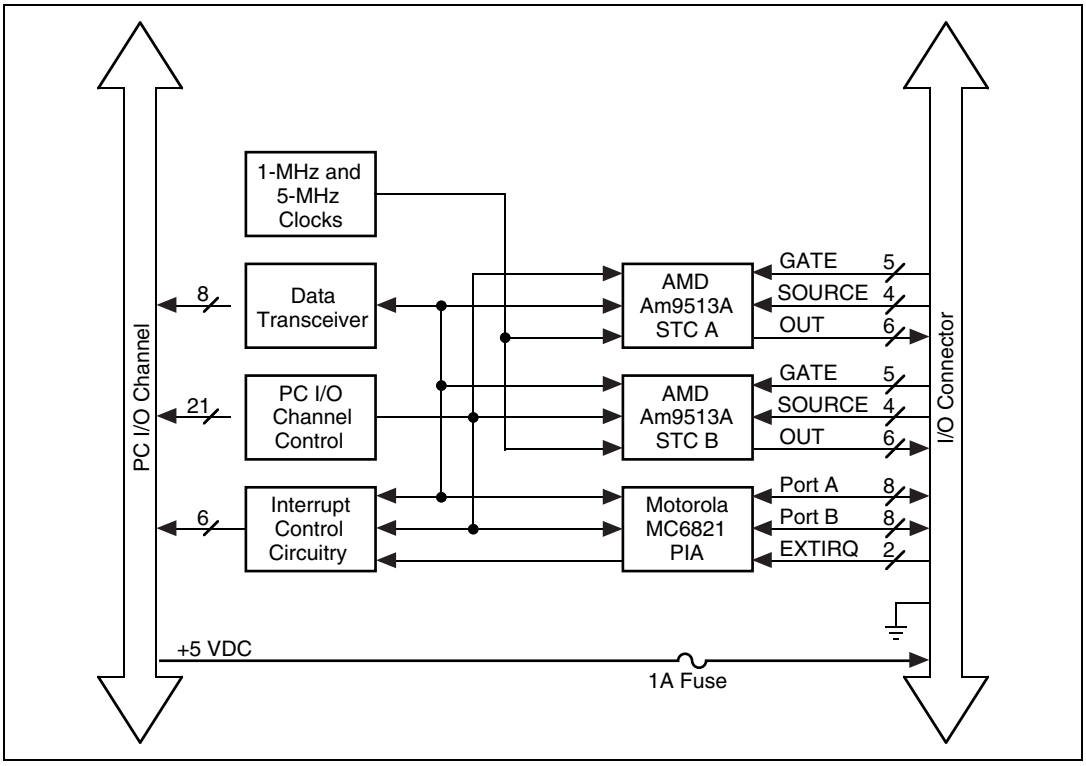


Figure 3-1. PC-TIO-10 Block Diagram

The PC I/O channel consists of an address bus, a data bus, a DMA arbitration bus, interrupt lines, and several control and support signals.

Data Transceivers

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

PC I/O Channel Control Circuitry

The base address used by the board is determined by an onboard switch setting. The address on the PC I/O channel bus is monitored by the address decoder, which is part of the I/O channel control circuitry. If the address on the bus matches the selected I/O base address of the board, the board is enabled and the corresponding register on the PC-TIO-10 is accessed.

In addition, the I/O channel control circuitry monitors and transmits the PC I/O channel control and support signals. The control signals identify transfers as read or write, memory or I/O, and 8-bit, 16-bit, or 32-bit transfers. The PC-TIO-10 uses only 8-bit transfers.

Am9513A System Timing Controller

The Am9513A STCs are the heart of the PC-TIO-10. These chips have five individually-controlled 16-bit counters, each of which can be configured to operate in a number of different modes. Therefore, the PC-TIO-10 can be used for applications such as rate generation, FSK, and pulse parameter measurement. Each of the counters has its own source (SOURCE), gate (GATE), and output (OUT) connections. Each STC has an independently-controlled, frequency-scaler output. The STCs are clocked by an onboard 1-MHz crystal oscillator to give 1-µsec timing resolution. In addition, SOURCE5 and SOURCE10 are clocked at 5 MHz to give 200-nsec resolution on all timing channels. Refer to Chapter 4, *Programming*, or to Appendix C, *AMD Am9513A Data Sheet*, for more detailed information.

MC6821 Peripheral Interface Adapter

The MC6821 PIA features sixteen bits of bit-configurable digital I/O. In addition, this device has two edge-programmable interrupt inputs, with which the PC-TIO-10 can receive external interrupts. Refer to Chapter 4, *Programming*, or to Appendix D, *Motorola MC6821Data Sheet*, for more detailed information.

Interrupt Control Circuitry

The interrupt level used by the PC-TIO-10 is selected by the onboard jumper W1. Interrupts can be generated from two different sources, EXTIRQ1 and EXTIRQ2, each of which has programmable-edge polarity and individual enable, clear, and disable commands. A second set of jumpers, W2, locally connects two of the counter outputs to the interrupt circuitry. With these connections, external wrap-backs are unnecessary if you want to use a counter to generate timed interrupts. Refer to Chapter 4, *Programming*, or to Appendix D, *Motorola MC6821Data Sheet*, for more detailed information on controlling interrupts. Refer to Chapter 2, *Configuration and Installation*, for more information on configuring the jumper settings.

Timing and Digital I/O Connector

All timing and digital I/O is transmitted through a standard, 50-pin, male connector. Pin 34 is connected to +5 V through a protection fuse (F1). This +5 V supply is often required to operate I/O module mounting racks. Pin 33 is connected to ground. See Chapter 2, *Configuration and Installation*, for additional information.

4

Programming

This chapter describes in detail the address and function of each of the PC-TIO-10 control and status registers. This chapter also includes important information about programming the PC-TIO-10.

The PC-TIO-10 is a timing and digital I/O board designed around two Am9513A integrated circuits and one MC6821 integrated circuit. The Am9513A is a general-purpose counter/timer with five 16-bit, individually-controlled counters and a 4-bit frequency-scaler output. The MC6821 is a 16-bit, bit-configurable, digital I/O device with two interrupt inputs that are edge-programmable. This chapter includes programming information for the PC-TIO-10, along with program examples written in C and assembly language.

Note If you plan to use a programming software package such as LabWindows or NI-DAQ with your PC-TIO-10 board, you need not read this chapter.

Introduction

Each of the two Am9513A STC devices is controlled by three different registers–a data register, a command register, and a status register. These registers are defined later in this chapter. Because there are two Am9513A STC devices on the board, they are referenced as STC A and STC B when differentiation is required.

The MC6821 PIA has four different registers that control its operation. The 16 I/O lines are grouped into two 8-bit ports, Port A and Port B, each of which has a control register and a data register associated with it. These registers are defined later in this chapter.

For clarification, both *registers* and *ports* are referenced in the sections that follow. A *register* refers to a given 8-bit or 16-bit register on the actual Am9513A STC or MC6821 PIA, whereas a port refers to the I/O channel register through which the device must be accessed. Therefore, the size shown for a register indicates both the register size and the I/O channel port size. The digital I/O ports associated with the MC6821 PIA are always referenced as Port A and Port B.

Register Map

The following table lists the address map for the PC-TIO-10.

Table 4-1. PC-TIO-10 Address Map

Register	Offset Address (Hex)	Size	Type
Am9513A Register Group			
STC A			
Data Register	00	8-bit	Read-and-write
Read-and-write	01	8-bit	Write-only
Command Register	01	8-bit	Read-only
STC B			
Data Register	02	8-bit	Read-and-write
Command Register	03	8-bit	Write-only
Status Register	03	8-bit	Read-only
MC6821 Register Group			
PIA			
Port A Data Register	04	8-bit	Read-and-write
Port A Control Register	05	8-bit	Read-and-write
Port B Data Register	06	8-bit	Read-and-write
Port B Control Register	07	8-bit	Read-and-write

Register Descriptions

The register descriptions for the devices on the PC-TIO-10, including the Am9513A STCs and the MC6821 PIA, are given on the pages that follow.

Register Descriptions for the Am9513A STCs

Each of the two Am9513A STC devices has three registers–a data register, a command register, and a status register. The bit maps and signal definitions for each of these registers are as follows. Counters 1, 2, 3, 4, and 5 map to Counters 1, 2, 3, 4, and 5 of STC A, respectively; Counters 6, 7, 8, 9, and 10 map to Counters 1, 2, 3, 4, and 5 of STC B, respectively.

Am9513A Data Registers

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

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

All these registers are 16-bit registers that must be accessed through an 8-bit port, least significant byte first. Bit descriptions for each of these registers are included in Appendix C, *AMD Am9513A Data Sheet*.

Address: Base address + 00 (hex) for Am9513A STC A
Base address + 02 (hex) for Am9513A STC B

Type: Read-and-write

Word Size: 16-bit register, 8-bit port

Bit Map:

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8

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

Bit	Name	Description
7-0	D<15..8>	These eight bits are the most significant byte to be loaded into or read from the Am9513A Internal Register currently selected. These eight bits should be accessed after the eight bits of the least significant byte are accessed.
7-0	D<7..0>	These eight bits are the least significant byte to be loaded into or read from the Am9513A Internal Register currently selected. These eight bits should be accessed before the eight bits of the most significant byte are accessed.

Am9513A Command Registers

The Am9513A Command Registers control the overall operation of the Am9513A Counter/Timer and selection of the internal registers that are accessed through the Am9513A Data Registers.

Address: Base address + 01 (hex) for Am9513A STC A
Base address + 03 (hex) for Am9513A STC B

Type: Write-only

Word Size: 8-bit register, 8-bit port

Bit Map:

7	6	5	4	3	2	1	0
C7	C6	C5	C4	C3	C2	C1	C0

Bit	Name	Description
7-0	C<7..0>	These eight bits are loaded into the Am9513A Command Register. See Appendix C, *AMD Am9513A Data Sheet*, for detailed bit descriptions of the Am9513A Command Registers.

Am9513A Status Registers

The Am9513A Status Registers give information about the output pin status of each counter in the Am9513A. In addition, these registers indicate the current setting of the byte pointer, which indicates whether the next byte to be accessed is the most significant byte or the least significant byte.

Address: Base address + 01 (hex) for Am9513A STC A
Base address + 03 (hex) for Am9513A STC B

Type: Read-only

Word Size: 8-bit register, 8-bit port

Bit Map:

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

Bit	Name	Description
7-6	X	Unused bits. They may be returned as 0 or 1.
5-1	OUT<5..1>	Each of these five bits returns the logic state of the associated counter output pin. For example, if the bit OUT4 is set, then the output pin of Counter 4 (or Counter 9) is at a logic-high state.
0	BYTE POINTER	This bit represents the state of the Am9513A Byte Pointer Flip-Flop. If this bit is set, the next byte to be written to or read from the Data Port is the least significant byte; if this bit is clear, the next byte to be written to or read from the Data Port is the most significant byte.

Register Descriptions for the MC6821

The MC6821 PIA has four registers–Port A and Port B both have a Data Register and a Control Register. The bit maps and signal definitions for each of these registers are as follows. For more information on the various registers, refer to Appendix D, *Motorola MC6821Data Sheet*.

MC6821 Data Registers

The MC6821 Data Registers are used to read from or write to the Output Registers (the I/O registers for Ports A and B) and the Data Direction Registers.

Address: Base address + 04 (hex) for Port A
Base address + 06 (hex) for Port B

Type: Read-and-write

Word Size: 8-bit register, 8-bit port

Bit Map:

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

Bit	Name	Description
7-0	D<7..0>	If the Output Register is being accessed (see the description of the Control Registers on the page that follows), writing a value to the Data Register updates all output bits and has no effect on input bits. Reading the Data Register returns the current signal value of all bits, including those configured for output. If the Data Direction Register is being accessed, writing a zero to a bit makes the corresponding I/O line an input, while writing a one to a bit makes the corresponding I/O line an output. Reading the Data Direction Register returns the current configuration.

MC6821 Control Registers

The MC6821 Control Registers control the overall operation of the MC6821 and the selection of the two internal registers that are accessed through each of the MC6821 Data Registers. Some of the bits in the Control Registers are not used because of the design of the PC-TIO-10. These bits should be set as follows.

Address: Base address + 05 (hex) for Port A
Base address + 07 (hex) for Port B

Type: Read-and-write

Word Size: 8-bit register, 8-bit port

Bit Map:

7	6	5	4	3	2	1	0
IRQ	0	0	0	0	DRS	EDGE	INTEN

Bit	Name	Description
7	IRQ	This is a read-only bit that reflects the current status of the interrupt input for the selected Control Register. If this bit is one in the Port A Control Register, an interrupt request is pending on the external interrupt line EXTIRQ1. If this bit is one in the Port B Control Register, an interrupt request is pending on the external interrupt line EXTIRQ2. Always write a zero to this bit.
6-3	Reserved	These bits are not used on the PC-TIO-10. Always write a zero to each of these bits.
2	DRS	This is the Data Register Select bit. Writing a one to this bit selects the Output Register, while writing a zero to this bit selects the Data Direction Register. Reading this bit shows the bit's current state. Refer to the description of the Data Register for more information.
1	EDGE	This is the control bit for selecting the edge that will cause an interrupt. Writing a one to this bit selects rising-edge interrupts, while writing a zero to this bit selects falling-edge interrupts. The Port A Control Register controls external interrupt line EXTIRQ1, while the Port B Control Register controls external interrupt line EXTIRQ2. Reading this bit shows the bit's current state.

0	INTEN	This bit enables and disables the interrupt generation capability of EXTIRQ1 or EXTIRQ2. Writing a one to this bit enables interrupts, while writing a zero to this bit disables interrupts. The Port A Control Register controls EXTIRQ1, while the Port B Control Register controls EXTIRQ2. Reading this bit shows the bit's current state.

Programming Considerations for the Am9513A STCs

Before using the Am9513A STC devices, you must initialize them. To do this, perform the following steps on each of the Am9513A STC devices. All writes are 8-bit write operations. All values are given in hexadecimal.

1. Issue a master reset by writing FF to the Am9513A Command Register.
2. Initialize all five counters. For *ctr* = 1 to 5, follow these steps:
   - Write *ctr* to the Am9513A Command Register (select the Counter Mode Register).
   - Write 00 to the Am9513A Data Register (store the least significant byte of the counter mode value).
   - Write 00 to the Am9513A Data Register (store the most significant byte of the counter mode value).
   - Write *ctr* + 8 to the Am9513A Command Register (select the Counter Load Register).
   - Write 03 to the Am9513A Data Register (store the least significant byte of the counter load value).
   - Write 00 to the Am9513A Data Register (store the most significant byte of the counter load value).
3. Load all counters with their Counter Load Register values by writing 5F to the Am9513A Command Register.

Note When you initialize Am9513A STC B, which contains Counters 6 through 10, *ctr* must range from 1 to 5, *not* from 6 to 10. Also, each Am9513A STC must always be configured to use the 8-bit bus mode in order to function properly.

Programming Example for the Am9513A STCs

The code below lists a sample function that can be used to reset the Am9513A STCs on the PC-TIO-10. In addition, the code lists a sample function that can be used to generate a variable duty-cycle square-wave.

```
/*  miscellaneous definitions   */

#define     cmd_port        0x0001
#define     data_port       0x0000
#define     no_err          0
#define     range_err       -1
#define     stc_a           0x0000
#define     stc_b           0x0002
#define     tio_ba          0x01a0


/*  function prototypes */

void     main(void);
void     reset9513(unsigned int, unsigned int);
int      square_wave(unsigned int, unsigned int, unsigned int,
         unsigned long, unsigned long);

/*  support functions   */

void    reset9513(base_address, chip_offset)
   unsigned int      base_address,
                     chip_offset;
   {  unsigned int       cmd,
                         data;
      int                ctr;

      /*  set up the register addresses   */

      cmd = base_address | chip_offset | cmd_port;
      data = base_address | chip_offset | data_port;
```

```
        /*  reset the 9513  */

        outp(cmd, 0xff);          /*  reset the chip  */
        for (ctr = 1; ctr <=5; ctr++)
            { outp(cmd, ctr);     /*  select Counter Mode Register */
              outp(data, 0x00);   /*  store mode low-byte */
              outp(data, 0x00);   /*  store mode high-byte    */
              outp(cmd, (ctr + 8));/*  select Counter Load Register */
              outp(data, 0x03);   /*  store load low-byte */
              outp(data, 0x00);   /*  store load high-byte    */
            }
        outp(cmd, 0x5f);               /*  load all counters   */
    }


int square_wave(base_address, counter, timebase, high_time, low_time)
    unsigned int    base_address,
                    counter,
                    timebase;
    unsigned long   high_time,
                    low_time;
    {   unsigned int    cmd,
                        data,
                        mode;

        /*  check ranges    */

        if ((counter < 1) || (counter > 10) ||
            (timebase > 15) ||
            (high_time < 1L) || (high_time > 65536L) ||
            (low_time < 1L) || (low_time > 65536L))
            return range_err;

        /*  set up the register addresses   */

        cmd = base_address | ((counter > 5) ? stc_b : stc_a) | cmd_port;
        data = base_address | ((counter > 5) ? stc_b : stc_a) |
        data_port;
```

```
        /*  adjust some parameters and program the counter  */

        if (counter > 5)                /*  5 ctrs per chip */
           counter -= 5;
        mode = 0x0062 | (timebase << 8);/*  counter mode    */
        if (high_time == 65536L)        /*  count of 65,536 */
           high_time = 0L;              /*      goes to 0   */
        if (low_time == 65536L)         /*  count of 65,536 */
           low_time = 0L;               /*      goes to 0   */

        outp(cmd,(0xc0 | (0x01 << (counter - 1))));/* disarm the ctr*/
        outp(cmd, counter);                     /*  select Mode Reg */
        outp(data, mode);                       /*  send mode
                                                    low-byte   */
        outp(data, (mode >> 8));                /*  send mode
                                                    high-byte  */
        outp(cmd, (counter + 0x08));            /*  select Load Reg */
        outp(data, ((unsigned int) high_time));/*  send load
                                                    low-byte   */
        outp(data, ((unsigned int) (high_time >> 8)));/*  send load
                                                    high-byte  */
        outp(cmd, (counter + 0x10));            /*  select Hold Reg */
        outp(data, ((unsigned int) low_time)); /*  send hold
                                                    low-byte   */
        outp(data, ((unsigned int) (low_time >> 8)));/*  send hold
                                                    high-byte  */
        outp(cmd, (0x40 | (0x01 << (counter - 1))));/*  load the ctr */
        outp(cmd, (0xe8 | counter));            /*  set output high */
        outp(cmd, (0x20 | (0x01 << (counter - 1))));/*  arm the ctr */

        return no_err;
    }

/*  the main function   */

void    main()
    {   /*  reset both 9513s    */

        reset9513(tio_ba, stc_a);
        reset9513(tio_ba, stc_b);
```

```
    /*  start a 100 khz, 70% duty cycle, square wave on Counter 8:
        tio_ba selects the board's base address
        8 selects the counter
        0x000b selects timebase F1, or 1 MHz
        7L selects a high time of 7 µsec
        3L selects a low time of 3 µsec
        a total of 10 µsec/cycle gives a 100 kHz wave
        7 clocks high out of 10 clocks gives a 70% duty cycle
    */

    square_wave(tio_ba, 8, 0x000b, 7L, 3L);
}
```

Interrupt Programming Example for the MC6821

The PC-TIO-10 is configured so that EXTIRQ1 on the I/O connector is connected to CA1 on the MC6821, EXTIRQ2 on the I/O connector is connected to CB1 on the MC6821, and CA2 and CB2 of the MC6821 are disabled. The signal names CA1, CA2, CB1, and CB2 refer to the names of pins located on the MC6821. The names are given to clarify how the interrupt circuitry is connected on the MC6821. For more information on these signals, see Appendix D, *Motorola MC6821Data Sheet*. Interrupts are enabled and disabled through the MC6821 Control Register. In addition, the edge that generates the interrupt is programmable through the MC6821 Control Register.

When an interrupt is generated (as indicated when the Control Register is read), the only way the interrupt can be cleared is by reading the Output Register (through the Data Register) of the I/O port that indicated the interrupt. For instance, if IRQ in the Port B Control Register is set, you must set DRS of the Port B Control Register to one, and then you must read the Port B Data Register. The data returned may not be important depending on how you are using interrupts.

The code that follows demonstrates how to set up the MC6821 for interrupt generation.

```
/*  defines for the program */

#define base_address   0x01A0   /*  board located at address 1A0    */
#define porta_offset   0x04     /*  offset for Port A   */
#define portb_offset   0x06     /*  offset for Port B   */
#define data_offset    0x00     /*  offset of Data Register */
#define ctrl_offset    0x01     /*  offset of Control Register  */
#define irq_channel    5        /*  the interrupt channel set on W1 */


/*  a sample structure for the interrupt service routine    */

typedef struct  {   unsigned int    pa_ctrl,
                                    pa_data,
                                    pb_ctrl,
                                    pb_data;
                    int             done;
                }   isr_block_type;


/* prototypes for the assembly language functions   */

void far    install_isr(int, isr_block_type far *);
void far    remove_isr(void);


/*  the main program    */

void    main()
    {   unsigned int    pa_ctrl,
                        pa_data,
                        pb_ctrl,
                        pb_data;
        isr_block_type  isr_block;

        /*  calculate register addresses    */

        pa_ctrl = base_address + porta_offset + ctrl_offset;
        pa_data = base_address + porta_offset + data_offset;
        pb_ctrl = base_address + portb_offset + ctrl_offset;
        pb_data = base_address + portb_offset + data_offset;
```

```
        /*  clear any active interrupts by reading Data Registers   */

         outp(pa_ctrl, 0x04);    /*  select Output Register  */
         inp(pa_data);           /*  clear Port A interrupts */
         outp(pb_ctrl, 0x04);    /*  select Output Register  */
         inp(pb_data);           /*  clear Port B interrupts */


         /*  install the interrupt service routine   */

        isr_block.pa_ctrl = pa_ctrl;    /*  initialize isr_block    */
         isr_block.pa_data = pa_data;
         isr_block.pb_ctrl = pb_ctrl;
         isr_block.pb_data = pb_data;
         isr_block.done = 0;
         install_isr(irq_channel, &isr_block);


         /*  configure Ports A and B for interrupts  */

        outp(pa_ctrl, 0x05);    /*  enable falling-edge interrupts  */
        outp(pb_ctrl, 0x07);    /*  enable rising-edge interrupts   */


         /*  wait for the process to be completed    */

         while (!isr_block.done)
             /* call_foreground_code() */ ;


       /*  disable interrupts and remove the interrupt service routine
*/

         outp(pa_ctrl, 0x04);
         inp(pa_data);
         outp(pb_ctrl, 0x04);
         inp(pb_data);
         remove_isr();
    }
```

Sample code for the functions `install_isr()` and `remove_isr()` is presented as follows. Be sure to pass a 32-bit structure pointer to the

install_isr() function, because the main program's data will probably be stored in a different memory segment than the one where the interrupt functions are located. In addition, if you call the installation function from a language besides C, make sure the parameters are passed in the proper order. C pushes parameters on the stack from right to left, but most other languages, most notably Pascal, push parameters from left to right. Finally, be sure to make the calls to the functions using 32-bit addresses, because all of the code assumes data is offset with respect to a 32-bit return address. The code can be modified to use 16-bit addresses by changing far to near and decrementing all references to the base page register, bp, by two in install_isr() and remove_isr() only. Do not modify isr_handler().

Also, isr_handler() should check, service, and clear both Port A and Port B interrupts before issuing the end-of-interrupt command. If interrupts are still active when the end-of-interrupt command is issued, program operation usually becomes unstable and is likely to lock up the computer.

```
; assemble this file with the following command:
;   masm /MX filename;
;   /MX preserves case sensitivity
;
;
; function prototypes:
;
;   void   install_isr(int level, isr_block_type far * isr_block);
;
;       on input, level indicates the interrupt level that is to
;       be modified
;       on input, isr_block points to the data structure that will be
;       used by the isr_handler function
;
;   void   isr_handler(void);
;
;       the isr_handler() function will never be called from C.....
;
;   void   remove_isr(void);
;

public _install_isr, _isr_handler, _remove_isr

_DATA   segment word public 'DATA'
```

```
; declarations

ackm        equ     00020h
acks        equ     000a0h
eoi         equ     00020h
maskm       equ     00021h
masks       equ     000a1h

int_addr    dd      0
int_mask    dw      0
isrb_addr   dd      0
slave_ack   db      0
vect_num    db      0

_DATA       ends


_TEXT   segment word public 'CODE'
        assume cs:_TEXT, ss:_TEXT, ds:_DATA

; install_isr
;
; bp reg            at [bp+0]
; ret addr ofs      at [bp+2]
; ret addr seg      at [bp+4]
; level             at [bp+6]
; isr_block ofs     at [bp+8]
; isr_block seg     at [bp+10]
;

_install_isr        proc    far
            cli
            push    bp
            mov     bp,sp
            push    ax
            push    bx
            push    cx
            push    dx
            push    ds
            push    es
            mov     ax,seg _DATA
            mov     ds,ax
```

```
; save the pointer for the isr_block structure--used in isr_handler

            mov     ax,[bp+8]                ; get ofs into ax
            mov     word ptr isrb_addr[0],ax ; save address in
                                             ; variable
            mov     ax,[bp+10]               ; get seg into ax
            mov     word ptr isrb_addr[2],ax ; save address in
                                             ; variable


; set interrupt vector--save the current vector before writing out
; new one

            mov     ax,[bp+6]   ; get interrupt level
            cmp     al,7        ; check to see if it belongs to master
            ja      short slave ; or slave interrupt chip
            add     al,008h     ; offset for master vector list
            jmp     short setvec; go set the vector
slave:
            add     al,068h     ; offset for slave vector list
            mov     slave_ack,1 ; flag for slave channel
setvec:
            push    ax          ; save vector number for later
            mov     ah,35h      ; get current vector
            int     21h         ; get previous int_addr in es:bx
            pop     ax           ; restore vector number
            mov     cx,cs        ; prep to compare current/new vectors
            mov     dx,es
            cmp     dx,cx        ; see if vector is already there
            jne     short ii_0
            cmp     bx,offset _isr_handler
            je      short ii_exit; vector already installed--exit
ii_0:
            mov     vect_num,al  ; save vector number for remove_isr
            mov     word ptr int_addr[0],bx   ; save the address
            mov     word ptr int_addr[2],es
            push    ds           ; save the data segment
            mov     ds,cx        ; copy cx (== cs) into ds
            mov     dx,offset _isr_handler    ; ds:dx points to new
handler
            mov     ah,25h
            int     21h          ; install the handler in the system
            pop     ds
```

```
; mask interrupt level in the interrupt controller register and store
;   the original setting of the mask bit for the selected interrupt
level

            mov     cx,[bp+6]    ; get interrupt level
            mov     bx,1         ; generate some masks
            shl     bx,cl
            mov     cx,bx        ; cx has 1 in bit pos of int-level
            not     bx           ; bx has 0 in bit pos of int-level
            in      al,maskm     ; get mask data from master chip
            jmp     $+2          ; delay--wait for data transfer
            and     cl,al        ; determine setting of mask bit
            and     al,bl        ; enable interrupts for selected
level
            out     maskm,al
            jmp     $+2          ; delay--wait for data transfer
            in      al,masks     ; get mask data from slave chip
            jmp     $+2          ; delay--wait for data transfer
            and     ch,al        ; determine setting of mask bit
            and     al,bh        ; enable interrupts for selected
level
            out     masks,al
            mov     int_mask,cx  ; save the previous value of the mask


; restore saved registers

ii_exit:
            pop     es
            pop     ds
            pop     dx
            pop     cx
            pop     bx
            pop     ax
            pop     bp
            sti
            ret
_install_isr        endp
```

```
; remove_isr
;
; bp reg            at [bp+0]
; ret addr ofs      at [bp+2]
; ret addr seg      at [bp+4]
;

_remove_isr proc    far
            cli
            push    ax
            push    bx
            push    cx
            push    dx
            push    ds
            push    es
            mov     ax,seg _DATA
            mov     ds,ax


; see if our vector is installed--if not, do not remove the vector

            cmp     vect_num,0     ; see if vect_num was ever set
            jz      short ri_exit ; our vector never installed--exit
            mov     al,vect_num    ; get vector number
            mov     ah,35h         ; get current vector from DOS
            int     21h          ; get previous int_addr in es:bx
            mov     cx,cs        ; prep to compare old/current vectors
            mov     dx,es
            cmp     dx,cx        ; see if our vector is already there
            jne     short ri_exit; different vector segment--exit
            cmp     bx,offset _isr_handler
            jne     short ri_exit; different vector offset--exit
```

```
; restore old mask and vector values

            mov     cx,int_mask  ; get the old mask value
            in      al,maskm     ; get current master mask
            jmp     $+2          ; delay--wait for data transfer
            or      al,cl        ; OR in old mask value
            out     maskm,al     ; send out new setting
            jmp     $+2          ; delay--wait for data transfer
            in      al,masks     ; get current slave mask
            jmp     $+2          ; delay--wait for data transfer
            or      al,ch        ; OR in old mask value
            out     masks,al     ; send out new setting
            jmp     $+2          ; delay--wait for data transfer
            mov     al,vect_num  ; al holds interrupt level
            mov     ah,25h
            lds     dx,int_addr  ; ds:dx points to new handler
            int     21h          ; install the old vector


; restore saved registers

ri_exit:
            pop     es
            pop     ds
            pop     dx
            pop     cx
            pop     bx
            pop     ax
            sti
            ret
_remove_isr endp



; isr_handler
;

_isr_handler        proc    far
            cli
            push    ax
            push    ds
```

```
; service interrupt

           ; your code here...
           ;   if this was not your interrupt, jump to 'ih_0'
           ;   if this was your interrupt, service it as appropriate;
           ;       the pointer for the data structure 'isr_block' is
           ;      stored at _DATA:isrb_addr; to access the structure,
           ;      use the following steps:
           ;
           ;            mov    ax,seg _DATA
           ;            mov    ds,ax
           ;            lds    si,isrb_addr
           ;
           ;        you need not use ds:si, but be sure to save any
           ;        registers you use...


; acknowledge the interrupt

ih_0:
           mov     ax,seg _DATA
           mov     ds,ax
           mov     al,eoi          ; signify end of interrupt
           cmp     slave_ack,0     ; see if we need to acknowledge
slave
           je      short ih_1      ; jump if not
           out     acks,al         ; send slave acknowledge
           jmp     $+2             ; delay--wait for data transfer
ih_1:
           out     ackm,al         ; send master acknowledge


; restore saved registers

           pop     ds
           pop     ax
           sti
           iret
_isr_handler       endp


_TEXT      ends
           end
```

A

Specifications

This appendix lists the specifications of the PC-TIO-10. These specifications are typical at 25 °C, unless otherwise stated. The operating temperature range is 0° to 70 °C.

I/O Connector Electrical Specifications

I/O Signal Ratings

Absolute maximum voltage rating..... –0.3 to +7.0 V with respect to GND

Input Signal Specifications

Input logic high voltage, all inputs..... 2.0 V min 5.25 V max

Input logic low voltage, all inputs..... 0.0 V min 0.8 V max

Input current, Am9513A
($0 < V_{in} < 5.25$ V)..... –10 µA min 10 µA max

Input current, MC6821, Port A
($0 < V_{in} < 0.8$ V)..... –2.4 mA max

Input current, MC6821, Port A
($2.0 < V_{in} < 5.25$ V)..... –400 µA max

Input current, MC6821, Port B
($0.4 < V_{in} < 2.4$ V)..... 10 µA max

Input current, MC6821, EXTIRQ1
and EXTIRQ2 ($0 < V_{in} < 5.25$ V) ..... 2.5 µA max

Pulse width, Am9513A, source inputs... 70 nsec min

Pulse width, Am9513A, gate inputs..... 145 nsec min

Pulse width, MC6821,
EXTIRQ1 and EXTIRQ2..... 100 nsec min

Output Signal Specifications

Pin 34 at +5 V.....1.0 A max

Note The total current output from pin 34 may be limited by the available current from your computer's power supply. To determine the available current, subtract the maximum power consumption of the board from the maximum current per slot. The difference, if less than 1 A, is the maximum current available to pin 34. If the difference is equal to or greater than 1 A, the maximum current available is restricted by the limitations of the connector, as shown previously.

Output logic high voltage,
all outputs at I_{out} = –200 µA.....2.4 V min 5.0 V max

Output logic low voltage,
all outputs at I_{out} = 3.2 mA .....0.0 V min 0.4 V max

Darlington drive current, MC6821,
Port B at V_{EXT} = 1.5 V.....–1.0 mA min –10.0 mA max

Operating Environment

Temperature.....0° to 70 °C

Relative humidity .....5% to 90% noncondensing

Storage Environment

Temperature.....–55° to 150 °C

Relative humidity .....5% to 90% noncondensing

Physical

Dimensions .....9.9 cm by 12.0 cm
(3.9 in. by 4.75 in.)

I/O connector .....50-pin male ribbon
cable connector

Power Requirement (from PC I/O Channel)

Typical power ..... 0.6 A at 5 VDC (±5%)

Maximum power ..... 1.4 A at 5 VDC (±5%)

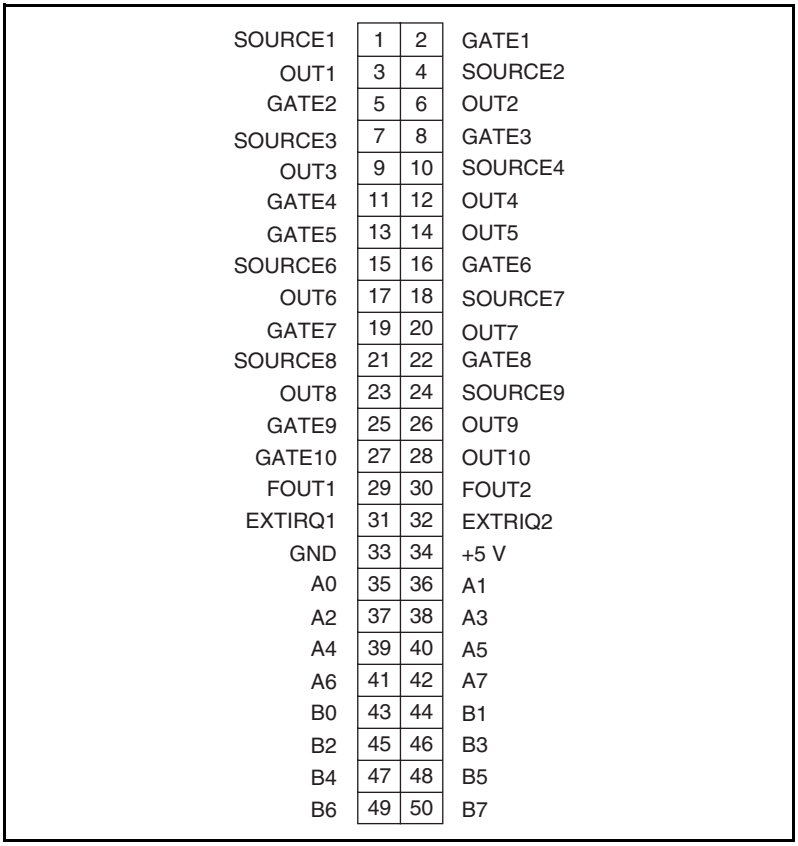
Note These power usage figures do not include the power used by external devices that are connected to the fused supply present on the I/O connector.

B

I/O Connector

This appendix describes the pinout and signal names for the I/O connector on the PC-TIO-10.

Figure B-1 shows the PC-TIO-10 I/O connector.



SOURCE1	1	2	GATE1
OUT1	3	4	SOURCE2
GATE2	5	6	OUT2
SOURCE3	7	8	GATE3
OUT3	9	10	SOURCE4
GATE4	11	12	OUT4
GATE5	13	14	OUT5
SOURCE6	15	16	GATE6
OUT6	17	18	SOURCE7
GATE7	19	20	OUT7
SOURCE8	21	22	GATE8
OUT8	23	24	SOURCE9
GATE9	25	26	OUT9
GATE10	27	28	OUT10
FOUT1	29	30	FOUT2
EXTIRQ1	31	32	EXTRIQ2
GND	33	34	+5 V
A0	35	36	A1
A2	37	38	A3
A4	39	40	A5
A6	41	42	A7
B0	43	44	B1
B2	45	46	B3
B4	47	48	B5
B6	49	50	B7

Figure B-1. PC-TIO-10 I/O Connector

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

C

AMD Am9513A Data Sheet*

This appendix contains the manufacturer data sheet for the AMD Am9513A integrated circuit (Advanced Micro Devices, Inc.). This circuit is used on the PC-TIO-10 board.

Am9513A

System Timing Controller

FINAL

DISTINCTIVE CHARACTERISTICS

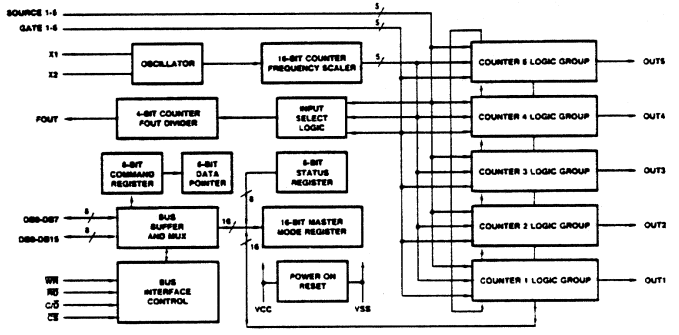
- Five independent 16-bit counters
- High speed counting rates
- Up/down and binary/BCD counting
- Internal oscillator frequency source
- Tapped frequency scaler
- Programmable frequency output
- 8-bit or 16-bit bus interface
- Time-of-day option
- Alarm comparators on counters 1 and 2
- Complex duty cycle outputs
- One-shot or continuous outputs
- Programmable count/gate source selection
- Programmable input and output polarities
- Programmable gating functions
- Retriggering capability
- +5 volt power supply
- Standard 40-pin package
- SMD/DESC qualified

GENERAL DESCRIPTION

The Am9513A System Timing Controller is an LSI circuit designed to service many types of counting, sequencing and timing applications. It provides the capability for programmable frequency synthesis, high resolution programmable duty cycle waveforms, retriggerable digital one-shots, time-of-day clocking, coincidence alarms, complex pulse generation, high resolution baud rate generation, frequency shift keying, stop-watching timing, event count accumulation, waveform analysis, etc. A variety of programmable operating modes and control features allows the Am9513A to be personalized for particular applications as well as dynamically reconfigured under program control.

The STC includes five general-purpose 16-bit counters. A variety of internal frequency sources and external pins may be selected as inputs for individual counters with software selectable active-high or active-low input polarity. Both hardware and software gating of each counter is available. Three-state outputs for each counter provide pulses or levels and can be active-high or active-low. The counters can be programmed to count up or down in either binary or BCD. The host processor may read an accumulated count at any time without disturbing the counting process. Any of the counters may be internally concatenated to form any effective counter length up to 80 bits.

BLOCK DIAGRAM

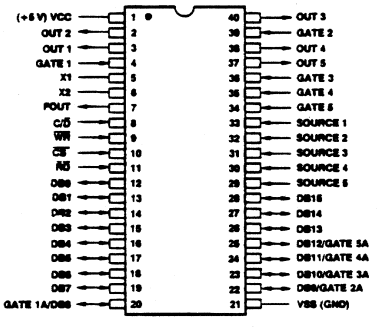


BD003381

Publication # 01731 Rev. F Amendment /0
Issue Date: August 1989

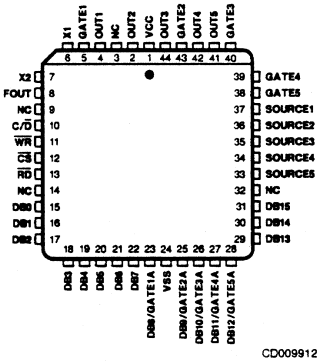
CONNECTION DIAGRAMS
Top View

DIP



CD005212

LCC & PLCC



CD009912

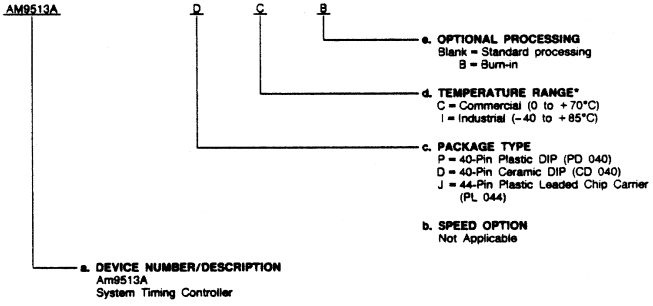
Note: Pin 1 is marked for orientation.

2

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. **Device Number**
b. **Speed Option** (if applicable)
c. **Package Type**
d. **Temperature Range**
e. **Optional Processing**



Valid Combinations	
AM9513A	PC, DC, DCB, DIB, JC

*This device is also available in Military temperature range.

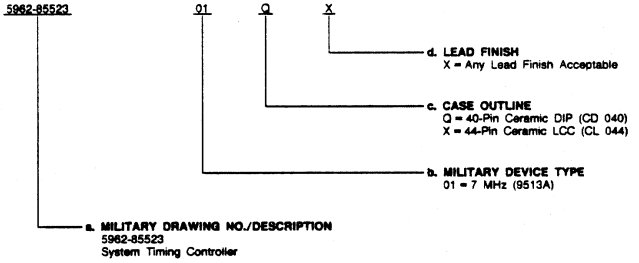
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

ORDERING INFORMATION (continued)

Standard Military Drawing (SMD)/DESC Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. Standard Military Drawing (SMD)/DESC products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for SMD/DESC products is formed by a combination of: **a. Military Drawing Part Number**
b. Device Type
c. Case Outline
d. Lead Finish



Valid Combinations	
5962-8552301	QX, XX

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

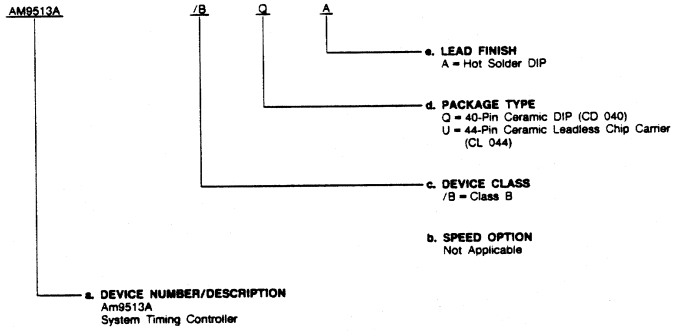
Group A tests consist of Subgroups
1, 2, 3, 7, 8, 9, 10, 11.

ORDERING INFORMATION (continued)

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
AM9513A	/BQA, /BUA

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

Group A tests consist of
Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

2

PIN DESCRIPTION

Pin No.	Name	I/O	Description
1	VCC		+5 V Power Supply.
21	VSS		Ground.
5, 6	X1, X2	O, I	(Crystal). X1 and X2 are the connections for an external crystal used to determine the frequency of the internal oscillator. The crystal should be a parallel-resonant, fundamental-mode type. An RC or LC or other reactive network may be used instead of a crystal. For driving from an external frequency source, X1 should be left open and X2 should be connected to a TTL source and a pull-up resistor.
7	FOUT	O	(Frequency Out). The FOUT output is derived from a 4-bit counter that may be programmed to divide its input by any integer value from 1 through 16 inclusive. The input to the counter is selected from any of 15 sources, including the internal scaled oscillator frequencies. FOUT may be gated on and off under software control and when off will exhibit a low impedance to ground. Control over the various FOUT options resides in the Master Mode register. After power-up, FOUT provides a frequency that is 1/16 that of the oscillator. The input source on power-up is F1.
4, 39, 36 – 34	GATE1 – GATE5	I	(Gate). The Gate inputs may be used to control the operations of individual counters by determining when counting may proceed. The same Gate input may control up to three counters. Gate pins may also be selected as count sources for any of the counters and for the FOUT divider. The active polarity for a selected Gate input is programmed at each counter. Gating function options allow level-sensitive gating or edge-initiated gating. Other gating modes are available including one that allows the Gate input to select between two counter output frequencies. All gating functions may also be disabled. The active Gate input is conditioned by an auxiliary input when the unit is operating with an external 8-bit data bus. See Data Bus description. Schmitt-trigger circuitry on the GATE inputs allows slow transition times to be used.
33 – 29	SRC1 – SRC5	I	(Source). The Source inputs provide external signals that may be counted by any of the counters. Any Source line may be routed to any or all of the counters and the FOUT divider. The active polarity for a selected SRC input is programmed at each counter. Any duty cycle waveform will be accepted as long as the minimum pulse width is at least half the period of the maximum specified counting frequency for the part. Schmitt-trigger circuitry on the SRC inputs allows slow transition times to be used.
3, 2, 40, 38, 37	OUT1 – OUT5	O	(Counter). Each 3-state OUT signal is directly associated with a corresponding individual counter. Depending on the counter configuration, the OUT signal may be a pulse, a square wave, or a complex duty cycle waveform. OUT pulse polarities are individually programmable. The output circuitry detects the counter state that would have been all bits zero in the absence of a reinitialization. That information is used to generate the selected waveform type. An optional output mode for Counters 1 and 2 overrides the normal output mode and provides a true OUT signal when the counter contents match the contents of an Alarm register.
12 – 19, 20, 22 – 28	DB0 – DB7, DB8 – DB15	I/O	(Data Bus). The 16 bidirectional Data Bus lines are used for information exchanges with the host processor. HIGH on a Data Bus line corresponds to one and LOW corresponds to zero. These lines act as inputs when $\overline{WR}$ and $\overline{CS}$ are active and as outputs when $\overline{RD}$ and $\overline{CS}$ are active. When $\overline{CS}$ is inactive, these pins are placed in a high-impedance state. After power-up or reset, the data bus will be configured for 8-bit width and will use only DB0 through DB7. DB0 is the least significant and DB7 is the most significant bit position. The data bus may be reconfigured for 16-bit width by changing a control bit in the Master Mode register. This is accomplished by writing an 8-bit command into the low-order DB lines while holding the DB13 – DB15 lines at a logic high level. Thereafter, all 16 lines can be used, with DB0 as the least significant and DB15 as the most significant bit position. When operating in the 8-bit data bus environment, DB8 – DB15 will never be driven active by the Am9513A. DB8 through DB12 may optionally be used as additional Gate inputs (see Figure 2). If unused, they should be held HIGH. When pulled LOW, a GATENA signal will disable the action of the corresponding counter N gating. DB13 – DB15 should be held HIGH in 8-bit bus mode whenever $\overline{CS}$ and $\overline{WR}$ are simultaneously active.
10	$\overline{CS}$	I	(Chip Select). The active-low Chip Select input enables Read and Write operations on the data bus. When Chip Select is HIGH, the Read and Write inputs are ignored. The first Chip Select signal after power-up is used to clear the power-on reset circuitry. If Chip Select is tied to ground permanently, the power-on reset circuitry may not function. In such a configuration, the software reset command must be issued following power-up to reset the Am9513A.
11	$\overline{RD}$	I	(Read). The active-low Read signal is conditioned by Chip Select and indicates that internal information is to be transferred to the data bus. The source will be determined by the port being addressed and, for Data Port reads, by the contents of the Data Pointer register. $\overline{WR}$ and $\overline{RD}$ should be mutually exclusive.
9	$\overline{WR}$	I	(Write). The active-low Write signal is conditioned by Chip Select and indicates that data bus information is to be transferred to an internal location. The destination will be determined by the port being addressed and, for Data Port writes, by the contents of the Data Pointer register. $\overline{WR}$ and $\overline{RD}$ should be mutually exclusive.
8	C/$\overline{D}$	I	(Control/Data). The Control/Data signal selects source and destination locations for Read and Write operations on the data bus. Control Write operations load the Command register and the Data Pointer. Control Read operations output the Status register. Data Read and Data Write transfers communicate with all other internal registers. Indirect addressing at the data port is controlled internally by the Data Pointer register.

Signal	Abbreviation	Type	Pins
+ 5 Volts	VCC	Power	1
Ground	VSS	Power	1
Crystal	X1, X2	O, I	2
Read	$\overline{RD}$	Input	1
Write	$\overline{WR}$	Input	1
Chip Select	$\overline{CS}$	Input	1
Control/Data	C/$\overline{D}$	Input	1
Source N	SRC	Input	5
Gate N	GATE	Input	5
Data Bus	DB	I/O	16
Frequency Out	FOUT	Output	1
Out N	OUT	Output	5

Figure 1. Interface Signal Summary

Figure 1 summarizes the interface signals and their abbreviations for the STC.

Package Pin	Data Bus Width (MM14)	
	16 Bits	8 Bits
12	DB0	DB0
13	DB1	DB1
14	DB2	DB2
15	DB3	DB3
16	DB4	DB4
17	DB5	DB5
18	DB6	DB6
19	DB7	DB7
20	DB8	GATE 1A
22	DB9	GATE 2A
23	DB10	GATE 3A
24	DB11	GATE 4A
25	DB12	GATE 5A
26	DB13	(VIH)
27	DB14	(VIH)
28	DB15	(VIH)

Figure 2. Data Bus Assignments

Interface Considerations

All of the input and output signals for the Am9513A are specified with logic levels compatible with those of standard TTL circuits. In addition to providing TTL compatible voltage levels, other output conditions are specified to help configure non-standard interface circuitry. The logic level specifications take into account all worst-case combinations of the three variables that affect the logic level thresholds: ambient temperature, supply voltage and processing parameters. A change in any of these toward nominal values will improve the actual operating margins and will increase noise immunity.

Unprotected open gate inputs of high quality MOS transistors exhibit very high resistances on the order of perhaps 10^{14} ohms. It is easy, therefore, in some circumstances, for charge to enter the gate node of such an input faster than it can be discharged and consequently, for the gate voltage to rise high enough to break down the oxides and destroy the transistor. All inputs to the Am9513A include protection networks to help prevent damaging accumulations of static charge. The protection circuitry is designed to slow the transitions of incoming current surges and to provide low-impedance discharge paths for voltages beyond the normal operating levels. Note, however, that input energy levels can nonetheless be too high to be successfully absorbed. Conventional design, storage, and handling precautions should be observed so that the protection networks themselves are not overstressed.

Within the limits of normal operation, the input protection circuitry is inactive and may be modeled as a lumped series RC as shown in Figure 3(a). The functionally active input connection during normal operation is the gate of a MOS transistor. No active sources or drains are connected to the inputs so that neither transient nor steady-state currents are impressed on the driving signals other than the charging or discharging of the input capacitance and the accumulated leakage associated with the protection network and the input circuit.

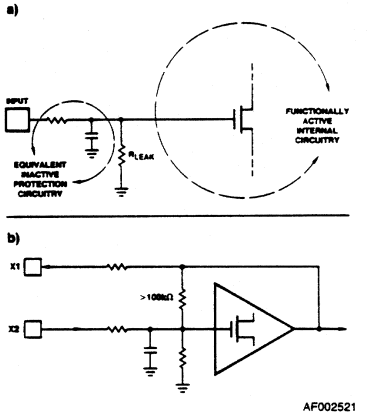


Figure 3. Input Circuitry

The only exception to the purely capacitive input case is the X2 crystal input. As shown in Figure 3(b) an internal resistor connects X1 and X2 in addition to the protection network. The resistor is a modestly high value of more than 100kohms.

Fanout from the driving circuitry into the Am9513A inputs will generally be limited by transition time considerations rather than DC current limitations when the loading is dominated by conventional MOS circuits. In an operating environment, all inputs should be terminated so they do not float and therefore will not accumulate stray static charges. Unused inputs should be tied directly to Ground or VCC, as appropriate. An input in use will have some type of logic output driving it, and termination during operation will not be a problem. Where inputs are driven from logic external to the card containing this chip, however, on-board termination should be provided to protect the chip when the board is unplugged (the input would otherwise float). A pull-up resistor or a simple inverter or gate will suffice.

DETAILED DESCRIPTION

The Am9513A System Timing Controller (STC) is a support device for processor oriented systems that is designed to enhance the available capability with respect to counting and timing operations. It provides the capability for programmable frequency synthesis, high resolution programmable duty cycle waveforms, retriggerable digital timing functions, time-of-day clocking, coincidence alarms, complex pulse generation, high resolution baud rate generation, frequency shift keying, stop-watching timing, event count accumulation, waveform analysis and many more. A variety of programmable operating modes and control features allow the Am9513A to be personalized for particular applications as well as dynamically reconfigured under program control.

The STC includes five general-purpose 16-bit counters. A variety of internal frequency sources and external pins may be selected as inputs for individual counters with software selectable active-high or active-low input polarity. Both hardware and software gating of each counter is available. Three-state outputs for each counter provide either pulses or levels. The counters can be programmed to count up or down in either binary or BCD. The accumulated count may be read without disturbing the counting process. Any of the counters may be internally concatenated to form an effective counter length of up to 80 bits.

The Am9513A block diagrams indicate the interface signals and the basic flow of information. Internal control lines and the internal data bus have been omitted. The control and data registers are all connected to a common internal 16-bit bus. The external bus may be 8- or 16-bits wide; in the 8-bit mode, the internal 16-bit information is multiplexed to the low order data bus pins DB0 through DB7.

An internal oscillator provides a convenient source of frequencies for use as counter inputs. The oscillator's frequency is controlled at the X1 and X2 interface pins by an external reactive network such as a crystal. The oscillator output is divided by the Frequency Scaler to provide several sub-frequencies. One of the scaled frequencies (or one of ten input signals) may be selected as an input to the FOUT divider and then comes out of the chip at the FOUT interface pin.

The STC is addressed by the external system as two locations: a Control port and a Data port. The Control port provides direct access to the Status and Command registers, as well as allowing the user to update the Data Pointer register. The Data port is used to communicate with all other addressable internal locations. The Data Pointer register controls the Data port addressing.

Among the registers accessible through the Data port are the Master Mode register and five Counter Mode registers, one for each counter. The Master Mode register controls the programmable options that are not controlled by the Counter Mode registers.

Each of the five general-purpose counters is 16-bits long and is independently controlled by its Counter Mode register. Through this register, a user can software select one of 16 sources as the counter input, a variety of gating and repetition modes, up or down counting in binary or BCD and active-high or active-low input and output polarities.

Associated with each counter are a Load register and a Hold register, both accessible through the Data port. The Load register is used to automatically reload the counter to any predefined value, thus controlling the effective count period. The Hold register is used to save count values without disturbing the count process, permitting the host processor to read intermediate counts. In addition, the Hold register may be used as a second Load register to generate a number of complex output waveforms.

All five counters have the same basic control logic and control registers. Counters 1 and 2 have additional Alarm registers and comparators associated with them, plus the extra logic necessary for operating in a 24-hour time-of-day mode. For real-time operation, the time-of-day logic will accept 50Hz, 60Hz or 100Hz input frequencies.

Each general counter has a single dedicated output pin. It may be turned off when the output is not of interest or may be configured in a variety of ways to drive interrupt controllers, Darlington buffers, bus drivers, etc. The counter inputs, on the other hand, are specifically not dedicated to any given interface line. Considerable versatility is available for configuring both the input and the gating of individual counters. This not only permits dynamic reassignment of inputs under software control, but also allows multiple counters to use a single input and a single gate pin to control more than one counter. Indeed, a single pin can be the gate for one counter and, at the same time, the count source for another.

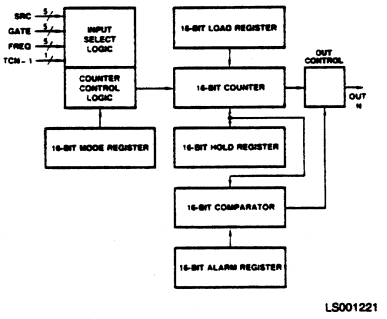


Figure 4. Counter Logic Groups 1 and 2

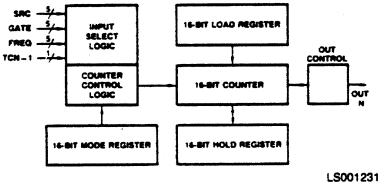


Figure 5. Counter Logic Groups 3, 4 and 5

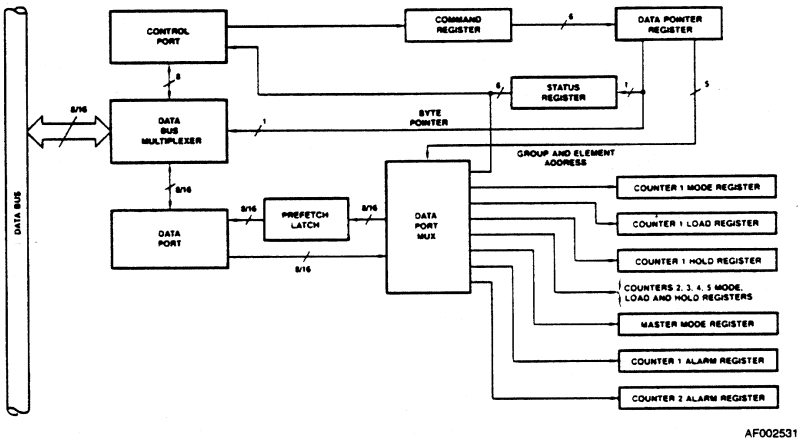


Figure 6. Am9513A Register Access

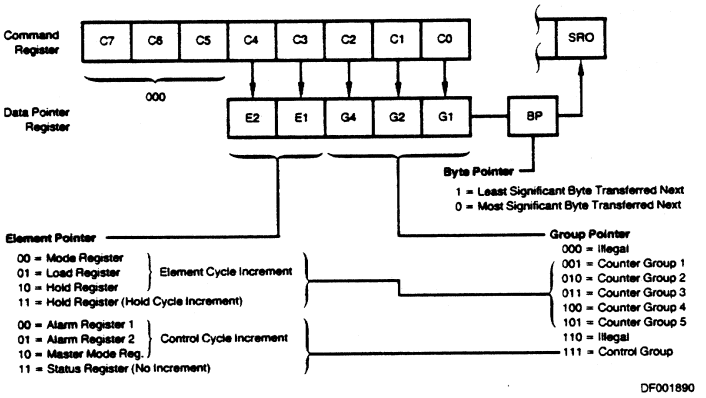


Figure 7. Data Pointer Register

2

	Element Cycle			Hold Cycle
	Mode Register	Load Register	Hold Register	Hold Register
Counter 1	FF01	FF09	FF11	FF19
Counter 2	FF02	FF0A	FF12	FF1A
Counter 3	FF03	FF0B	FF13	FF1B
Counter 4	FF04	FF0C	FF14	FF1C
Counter 5	FF05	FF0D	FF15	FF1D
Master Mode Register = FF17 Alarm 1 Register = FF07 Alarm 2 Register = FF0F Status Register = FF1F				

Notes:
1. All codes are in hex.
2. When used with an 8-bit bus, only the two low order hex digits should be written to the command port; the "FF" prefix should be used only for a 16-bit data bus interface.

Figure 8. Load Data Pointer Commands

Sequencing is enabled by clearing Master Mode bit 14 (MM14) to zero. As shown in Figure 9 several types of sequencing are available depending on the data bus width being used and the initial Data Pointer value entered by command.

When E1 = 0 or E2 = 0 and G4, G2, G1 points to a Counter Group, the Data Pointer will proceed through the Element cycle. The Element field will automatically sequence through the three values 00, 01 and 10 starting with the value entered. When the transition from 10 to 00 occurs, the Group field will also be incremented by one. Note that the Element field in this case does not sequence to a value of 11. The Group field circulates only within the five Counter Group codes.

If E2, E1 = 11 and a Counter Group are selected, then only the Group field is sequenced. This is the Hold cycle. It allows the Hold registers to be sequentially accessed while bypassing the Mode and Load registers. The third type of sequencing is the Control cycle. If G4, G2, G1 = 111 and E2, E1 ≠ 11, the Element Pointer will be incremented through the values 00, 01 and 10, with no change to the Group Pointer.

When G4, G2, G1 = 111 and E2, E1 = 11, no incrementing takes place and only the Status register will be available through the Data port. Note that the Status register can also always be read directly through the Control port.

For all these auto-sequencing modes, if an 8-bit data bus is used, the Byte pointer will toggle after every data transfer to allow the least and most significant bytes to be transferred before the Element or Group fields are incremented.

Prefetch Circuit

To minimize the read access time to internal Am9513A registers, a prefetch circuit is used for all read operations through the Data port. Following each read or write operation through the Data port, the Data Pointer register is updated to point to the next register to be accessed. Immediately following this update, the new register data is transferred to a special prefetch latch at the interface pad logic. When the user performs a subsequent read of the Data port, the data bus drivers are enabled, outputting the prefetched data on the bus. Since the internal data register is accessed prior to the start of the read operation, its access time is transparent to the user. To keep the prefetched data consistent with the Data Pointer, prefetches are also performed after each write to the Data port and after execution of the "Load Data Pointer" command. The following rules should be kept in mind regarding Data port Transfers.

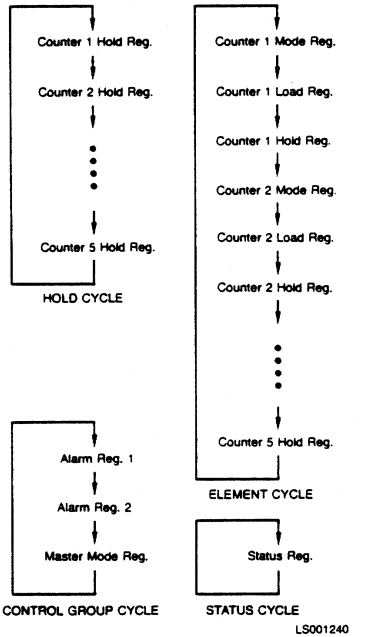


Figure 9. Data Pointer Sequencing

1. The Data Pointer register should always be reloaded before reading from the Data port if a command, other than "Load Data Pointer," was issued to the Am9513A following the last Data port read or write. The Data Pointer does not have to be loaded again if the first Data port transaction after a command entry is a write, since the Data port write will automatically cause a new prefetch to occur.
2. Operating modes N, O, Q, R and X allow the user to save the counter contents in the Hold register by applying an active-going gate edge. If the Data Pointer register had been pointing to the Hold register in question, the prefetched value will not correspond to the new value saved in the Hold Register. To avoid reading an incorrect value, a new "Load Data Pointer" command should be issued before attempting to read the saved data. A Data port write (to another register) will also initiate a prefetch; subsequent reads will access the recently saved Hold register data. Many systems will use the "saving" gate edge to interrupt the host CPU. In systems such as this, the interrupt service routine should issue a "Load Data Pointer" command prior to reading the saved data.

Status Register

The 8-bit read-only Status register indicates the state of the Byte Pointer bit in the Data Pointer register and the state of the

OUT signal for each of the general counters. See Figures 10 and 17. The OUT signals reported are those internal to the chip after the polarity-select logic and just before the three-state interface buffer circuitry. Bits SR6 and SR7 may be 0 or 1.

The Status register OUT bit reflects an active-high or active-low TC output or a TC Toggled output, as programmed in the Output Control Field of the Counter Mode register. That is, it reflects the exact state of the OUT pin. When the low-impedance to Ground Output option (CM2-CM0 = 000) is selected, the Status register will reflect an active-high TC Output. When a high-impedance Output option (CM2-CM0 = 100) is selected, the Status register will reflect an active-low TC output.

For Counters 1 and 2, the OUT pin will reflect the comparator output if the comparators are enabled. The Status register bit and OUT pin are active-high if CM2 = 0 and active-low if CM2 = 1. When the high-impedance option is selected and the comparator is enabled, the status register bit will reflect an active-high comparator output. When the low-impedance to Ground option is selected and the comparator is enabled, the status register bit will reflect an active-low comparator output.

The Status register is normally accessed by reading the Control port (see Figure 6) but may also be read via the Data port as part of the Control Group.

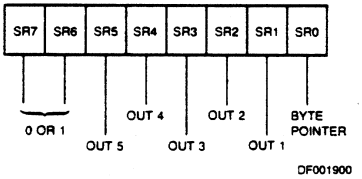


Figure 10. Status Register Bit Assignments

DATA PORT REGISTERS

Counter Logic Groups

As shown in Figures 4 and 5, each of the five Counter Logic Groups consists of a 16-bit general counter with associated control and output logic, a 16-bit Load register, a 16-bit Hold register and a 16-bit Mode register. In addition, Counter Groups 1 and 2 also include 16-bit Comparators and 16-bit Alarm registers. The comparator/alarm functions are controlled by the Master Mode register. The operation of the Counter Mode registers is the same for all five counters. The host CPU has both read and write access to all registers in the Counter Logic Groups through the Data port. The counter itself is never directly accessed.

Load Register

The 16-bit read/write Load register is used to control the effective length of the general counter. Any 16-bit value may be written into the Load register. That value can then be transferred into the counter each time the Terminal Count (TC) occurs. "Terminal Count" is defined as that period of time when the counter contents would have been zero if an external value had not been transferred into the counter. Thus, the terminal count frequency can be the input frequency divided by the value in the Load register. In all operating modes, either the Load or Hold register will be transferred into the counter when TC occurs. In cases where values are being accumulated in the counter, the Load register action can become transparent by filling the Load register with all zeros.

Hold Register

The 16-bit read/write Hold register is dual-purpose. It can be used in the same way as the Load register, thus offering an alternate source for module definition for the counter. The Hold register may also be used to store accumulated counter values for later transfer to the host processor. This allows the count to be sampled while the counting process proceeds without interruption. Transfer of the counter contents into the Hold register is accomplished by the hardware interface in some operating modes or by software commands at any time.

Counter Mode Register

The 16-bit read/write Counter Mode register controls the gating, counting, output and source select functions within each Counter Logic Group. The "Counter Mode Control Options" section of this document describes the detailed control options available. Figure 16 shows the bit assignments for the Counter Mode registers.

Alarm Registers and Comparators

Added functions are available in the Counter Logic Groups for Counters 1 and 2 (see Figure 4). Each contains a 16-bit Alarm register and a 16-bit Comparator. When the value in the counter reaches the value in the Alarm register, the Comparator output will go true. The Master Mode register contains control bits to individually enable/disable the comparators. When enabled, the comparator output appears on the OUT pin of the associated counter in place of the normal counter output. The output will remain true as long as the comparison is true, that is, until the next input causes the count to change. The polarity of the Comparator output will be active-high if the Output Control field of the Counter Mode register is 001 or 010 and active-low if the Output Control field is 101.

MASTER MODE CONTROL OPTIONS

The 16-bit Master Mode (MM) register is used to control those internal activities that are not controlled by the individual Counter Mode registers. This includes frequency control, Time-of-Day operation, comparator controls, data bus width and data pointer sequencing. Figure 11 shows the bit assignments for the Master Mode register. This section describes the use of each control field.

Master Mode register bits MM12, MM13 and MM14 can be individually set and reset using commands issued to the Command register. In addition, they can all be changed by writing directly to the Master Mode register.

After power-on reset or a Master Reset command, the Master Mode register is cleared to an all zero condition. This results in the following configuration:

Time-of-Day disabled
Both Comparators disabled
FOUT Source is frequency F1
FOUT Divider set for divide-by-16
FOUT gated on
Data Bus 8 bits wide
Data Pointer Sequencing enabled
Frequency Scaler divides in binary

2

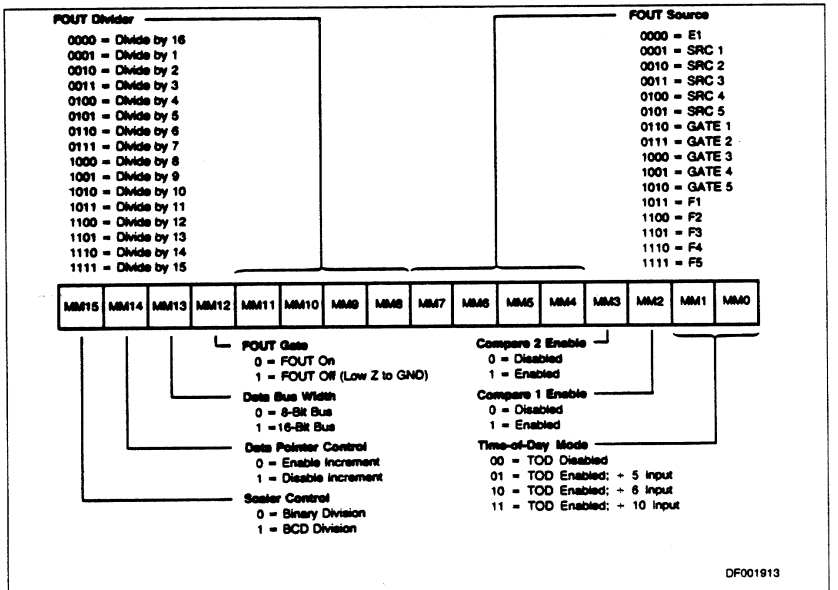


Figure 11. Master Mode Register Bit Assignments

Time-of-Day

Bits MM0 and MM1 of the Master Mode register specify the Time-of-Day (TOD) options. When MM0 = 0 and MM1 = 0, the special logic used to implement TOD is disabled, and Counters 1 and 2 will operate in exactly the same way as Counters 3, 4 and 5. When MM0 = 1 or MM1 = 1, additional counter decoding and control logic is enabled on Counters 1 and 2, which causes their decades to turn over at the counts that generate appropriate 24-hour TOD accumulations. For additional information, see the Time-of-Day chapter in the 9513A System timing controller technical manual.

Comparator Enable

Bits MM2 and MM3 control the Comparators associated with Counters 1 and 2. When a Comparator is enabled, its output is substituted for the normal counter output on the associated OUT1 or OUT2 pin. The comparator output will be active-high if the output control field of the Counter Mode register is 001 or 010 and active-low for a code of 101. Once the compare output is true, it will remain so until the count changes and the comparison therefore goes false.

The two Comparators can always be used individually in any operating mode. One special case occurs when the Time-of-Day option is revoked and both Comparators are enabled. The operation of Comparator 2 will then be conditioned by Comparator 1 so that a full 32-bit compare must be true in order to generate a true signal on OUT2. OUT1 will continue, as usual, to reflect the state of the 16-bit comparison between Alarm 1 and Counter 1.

FOUT Source

Master Mode bits MM4 through MM7 specify the source input for the FOUT divider. Fifteen inputs are available for selection, and they include the five Source pins, the five Gate pins and the five internal frequencies derived from the oscillator. The 16th combination of the four control bits (all zeros) is used to assure that an active frequency is available at the input to the FOUT divider following reset.

FOUT Divider

Bits MM8 through MM11 specify the dividing ratio for the FOUT Divider. The FOUT source (selected by bits MM4 through MM7) is divided by an integer value between 1 and 16, inclusive, and is then passed to the FOUT output buffer. After power-on or reset, the FOUT divider is set to divide-by-16.

FOUT Gate

Master Mode bit MM12 provides a software gating capability for the FOUT signal. When MM12 = 1, FOUT is off and in a low-impedance state to ground. MM12 may be set or cleared in conjunction with the loading of the other bits in the Master Mode register; alternatively, there are commands that allow MM12 to be individually set or cleared directly without changing any other Master Mode bits. After power-up or reset, FOUT is gated on.

When changing the FOUT divider ratio or FOUT source, transient pulses as short as half the period of the FOUT source may appear on the FOUT pin. Turning the FOUT gate on or off can also generate a transient. This should be considered when using FOUT as a system clock source.

Bus Width

Bit MM13 controls the multiplexer at the data bus interface in order to configure the part for an 8-bit or 16-bit external bus. The internal bus is always 16-bits wide. When MM13 = 1, 16-bit data is transferred directly between the internal bus and all 16 of the external bus lines. In this configuration, the Byte Pointer bit in the Data Pointer register remains set at all times. When MM13 = 0, 16-bit internal data is transferred a byte at a time to and from the eight low-order external data bus lines. The Byte Pointer bit toggles with each byte transfer in this mode.

When the Am9513A is set to operate with an 8-bit data bus width, pins DB8 through DB15 are not used for the data bus and are available for other functions. Pins DB13 through DB15 should be tied high. Pins DB8 through DB12 are used as auxiliary gating inputs and are labeled GATE1A through GATE5A respectively. The auxiliary gate pin, GATENA, is logically ANDed with the gate input to Counter N, as shown in Figure 12. The output of the AND gate is then used as the gating signal for Counter N.

Data Pointer Sequencing

Bit MM14 controls the Data Pointer logic to enable or disable the automatic sequencing functions. When MM14 = 1, the contents of the Data Pointer can be changed only directly by entering a command. When MM14 = 0, several types of automatic sequencing of the Data Pointer are available. These are described in the Data Pointer register section of this document.

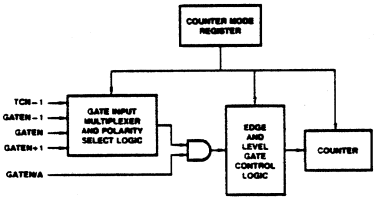
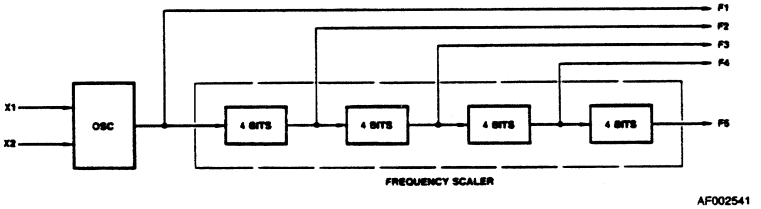


Figure 12. Gating Control

Thus, the host processor, by controlling MM14, may repetitively read/write a single internal location, or may sequentially read/write groups of locations. Bit MM14 can be loaded by writing to the Master Mode register or can be set or cleared by software command.

Scaler Ratios

Master Mode bit MM15 controls the counting configuration of the Frequency Scaler counter. When MM15 = 0, the Scaler divides the oscillator frequency in binary steps so that each subfrequency is 1/16 of the preceding frequency. When MM15 = 1, the Scaler divides in BCD steps so that adjacent frequencies are related by ratios of 10 instead of 16 (see Figure 13).



Frequency	BCD Scaling MM15 = 1	Binary Scaling MM15 = 0
F1	OSC	OSC
F2	F1 ÷ 10	F1 ÷ 16
F3	F1 ÷ 100	F1 ÷ 256
F4	F1 ÷ 1,000	F1 ÷ 4,096
F5	F1 ÷ 10,000	F1 ÷ 65,536

Figure 13. Frequency Scaler Ratios

Counter Mode	A	B	C	D	E	F	G	H	I	J	K	L
Special Gate (CM7)	0	0	0	0	0	0	0	0	0	0	0	0
Reload Source (CM6)	0	0	0	0	0	0	1	1	1	1	1	1
Repetition (CM5)	0	0	0	1	1	1	0	0	0	1	1	1
Gate Control (CM15–CM13)	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE
Count to TC once, then disarm	X	X	X									
Count to TC twice, then disarm							X	X	X			
Count to TC repeatedly without disarming				X	X	X				X	X	X
Gate input does not gate counter input	X			X			X			X		
Count only during active gate level		X			X			X			X	
Start count on active gate edge and stop count on next TC			X			X						
Start count on active gate edge and stop count on second TC									X			X
No hardware retriggering	X	X	X	X	X	X	X	X	X	X	X	X
Reload counter from Load register on TC	X	X	X	X	X	X						
Reload counter on each TC, alternating reload source between Load and Hold registers							X	X	X	X	X	X
Transfer Load register into counter on each TC that gate is LOW, transfer Hold register into counter on each TC that gate is HIGH.												
On active gate edge transfer counter into Hold register and then reload counter from Load register												

Counter Mode	M	N	O	P	Q	R	S	T	U	V	W	X
Special Gate (CM7)	1	1	1	1	1	1	1	1	1	1	1	1
Reload Source (CM6)	0	0	0	0	0	0	1	1	1	1	1	1
Repetition (CM5)	0	0	0	1	1	1	0	0	0	1	1	1
Gate Control (CM15–CM13)	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE
Count to TC once, then disarm		X	X									
Count to TC twice, then disarm							X					
Count to TC repeatedly without disarming					X	X				X		X
Gate input does not gate counter input							X			X		
Count only during active gate level		X			X							
Start count on active gate edge and stop count on next TC			X			X						X
Start count on active gate edge and stop count on second TC												
No hardware retriggering							X			X		X
Reload counter from Load register on TC		X	X		X	X						X
Reload counter on each TC, alternating reload source between Load and Hold registers.												
Transfer Load register into counter on each TC that gate is LOW, transfer Hold register into counter on each TC that gate is HIGH.							X			X		
On active gate edge transfer counter into Hold register and then reload counter from Load register		X	X		X	X						
On active gate edge transfer counter into Hold register, but counting continues												X

Notes: 1. Counter modes M, P, T, U and W are reserved and should not be used.
2. Mode X is available for Am9513A only.

Figure 14. Counter Mode Operating Summary

COUNTER MODE DESCRIPTIONS

Counter Mode register bits CM15–CM13 and CM7–CM5 select the operating mode for each counter (see Figure 14). To simplify references to a particular mode, each mode is assigned a letter from A through X. Representative waveforms for the counter modes are illustrated in Figures 15a through 15v. (Because the letter suffix in the figure number is keyed to the mode, Figures 15m, 15p, 15t, 15u and 15w do not exist.) The figures assume down counting on rising source edges. Those modes which automatically disarm the counter (CM5 = 0) are shown with the $\overline{WR}$ plus entering the required ARM command; for modes which count repetitively (CM5 = 1), the ARM command is omitted. The retriggering modes (N, O, Q and R) are shown with one retrigger operation. Both a TC output waveform and a TC Toggled output waveform are shown for each mode. The symbols L and H are used to represent count values equal to the Load and Hold register contents, respectively. The symbols K and N represent arbitrary count values. For each mode, the required bit pattern in the Counter Mode register is shown; "don't care" bits are marked "X." These figures are designed to clarify the mode descriptions; the Am9513A Electrical Specification should be used as the authoritative reference for timing relationships between signals.

To keep the following mode descriptions concise and to the point, the phrase "source edges" is used to refer to active-going source edges only, not to inactive-going edges. Similarly, the phrase "gate edges" refers only to active-going gate edges. Also, again to avoid verbosity and euphuism, the descriptions of some modes state that a counter is stopped or disarmed "on a TC, inhibiting further counting." As is fully explained in the TC section of this document, for these modes the counter is actually stopped or disarmed following the active-going source edge which drives the counter out of TC. In other words, since a counter in the TC state always counts, irrespective of its gating or arming status, the stopping or disarming of the count sequence is delayed until TC is terminated.

MODE A

Software-Triggered Strobe with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode A, shown in Figure 15a, is one of the simplest operating modes. The counter will be available for counting source edges when it is issued an ARM command. On each TC, the counter will reload from the Load register and automatically disarm itself, inhibiting further counting. Counting will resume when a new ARM command is issued.

MODE B

Software-Triggered Strobe with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode B, shown in Figure 15b, is identical to Mode A except that source edges are counted only when the assigned Gate is active. The counter must be armed before counting can occur. Once armed, the counter will count all source edges which occur while the Gate is active and disregard those edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. On each TC the counter will reload from the Load register and automatically disarm itself, inhibiting further counting unitl a new ARM command is issued.

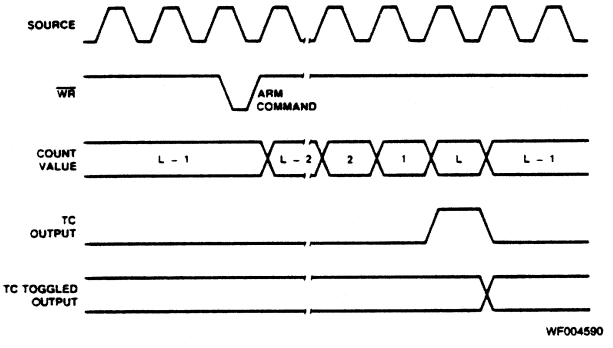


Figure 15a. Mode A Waveforms

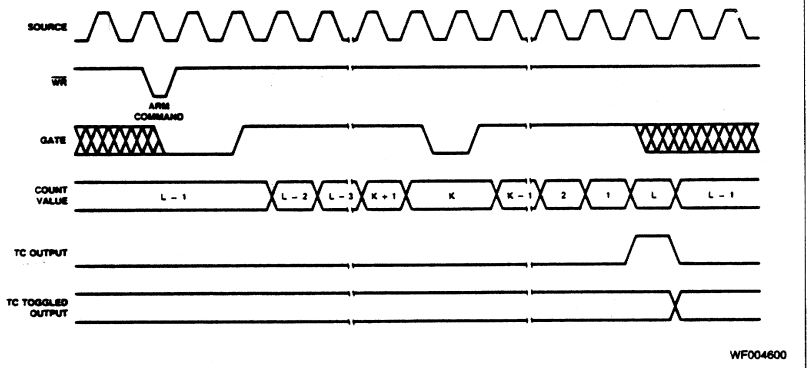


Figure 15b. Mode B Waveforms

MODE C

Hardware-Triggered Strobe

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode C, shown in Figure 15c, is identical to Mode A, except that counting will not begin until a Gate edge is applied to the armed counter. The counter must be armed before application of the triggered Gate edge; Gate edges applied to a disarmed counter are disregarded. The counter will start counting on the first source edge after the triggering Gate edge and will continue counting until TC. At TC, the counter will reload from the Load register and automatically disarm itself. Counting will then remain inhibited until a new ARM command and a new Gate edge are applied in that order. Note that after application of a triggered Gate edge, the Gate input will be disregarded for the remainder of the count cycle. This differs from Mode B, where the Gate can be modulated throughout the count cycle to stop and start the counter.

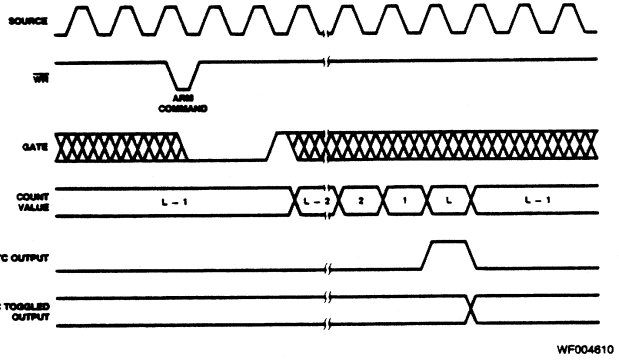


Figure 15c. Mode C Waveforms

MODE D

Rate Generator with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode D, shown in Figure 15d, is typically used in frequency generation applications. In this mode, the Gate input does not affect counter operation. Once armed, the counter will count to TC repetitively. On each TC, the counter will reload itself from the Load register; hence, the Load register value determines the time between TCs. A square wave rate generator may be obtained by specifying the TC Toggled output mode in the Counter Mode register.

MODE E

Rate Generator with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode E, shown in Figure 15e, is identical to Mode D, except the counter will only count those source edges which occur while the Gate input is active. This feature allows the counting process to be enabled and disabled under hardware control. A square wave rate generator may be obtained by specifying the TC Toggled output mode.

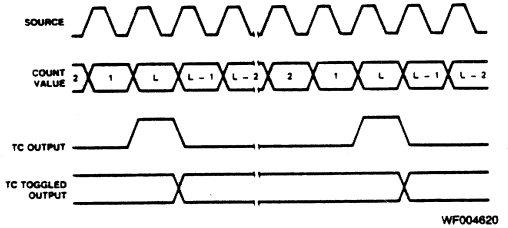


Figure 15d. Mode D Waveforms

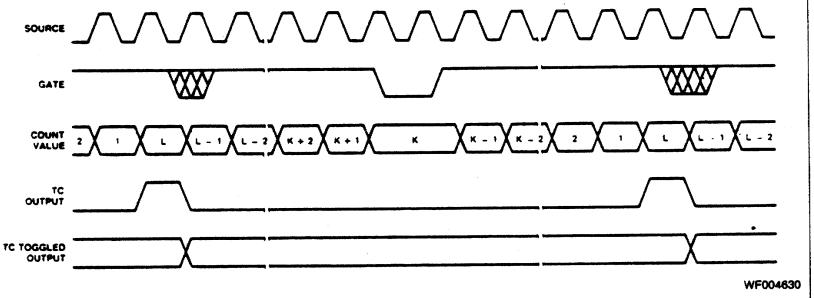


Figure 15e. Mode E Waveforms

MODE F

Non-Retriggerable One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode F, shown in Figure 15f, provides a non-retriggerable one-shot timing function. The counter must be armed before it will function. Application of a Gate edge to the armed counter will enable counting. When the counter reaches TC, it will reload itself from the Load register. The counter will then stop counting, awaiting a new Gate edge. Note that unlike Mode C, a new ARM command is not needed after TC, only a new Gate edge. After application of a triggering Gate edge, the Gate input is disregarded until TC.

MODE G

Software-Triggered Delayed Pulse One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

In Mode G, the Gate does not affect the counter's operation. Once armed, the counter will count to TC twice and then automatically disarm itself. For most applications, the counter will initially be loaded from the Load register either by a LOAD command or by the last TC of an earlier timing cycle. Upon counting to the first TC, the counter will reload itself from the Hold register. Counting will proceed until the second TC, when the counter will reload itself from the Load register and automatically disarm itself, inhibiting further counting. Counting can be resumed by issuing a new ARM command. A software-triggered delayed pulse one-shot may be generated by specifying the TC Toggled output mode in the Counter Mode register. The initial counter contents control the delay from the ARM command until the output pulse starts. The Hold register contents control the pulse duration. Mode G is shown in Figure 15g.

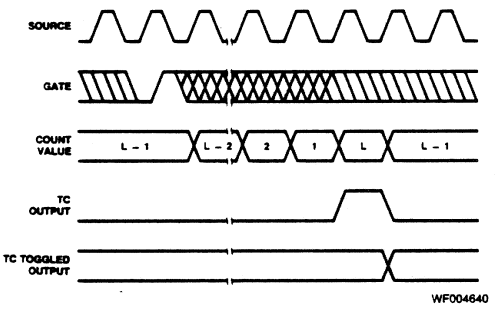


Figure 15f. Mode F Waveforms

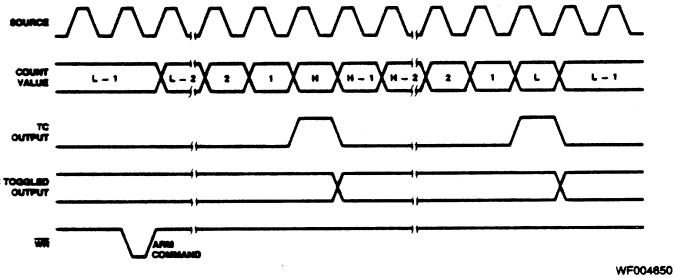


Figure 15g. Mode G Waveforms

MODE H

Software-Triggered Delayed Pulse One-Shot with Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

Mode H, shown in Figure 15h, is identical to Mode G except that the Gate input is used to qualify which source edges are to be counted. The counter must be armed for counting to occur. Once armed, the counter will count all source edges that occur while the Gate is active. This permits the Gate to turn the count process on and off. As with Mode G, the counter will be reloaded from the Hold register on the first TC and reloaded from the Load register and disarmed on the second TC. This mode allows the Gate to control the extension of both the initial output delay time and the pulse width.

MODE I

Hardware-Triggered Delayed Pulse Strobe

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

Mode I, shown in Figure 15i, is identical to Mode G, except that counting will not begin until a Gate edge is applied to an armed counter. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. An armed counter will start counting on the first source edge after the triggering Gate edge. Counting will then proceed in the same manner as in Mode G. After the second TC, the counter will disarm itself. An ARM command and Gate edge must be issued in this order to restart counting. Note that after application of a triggering Gate edge, the Gate input will be disregarded until the second TC. This differs from Mode H, where the Gate can be modulated throughout the count cycle to stop and start the counter.

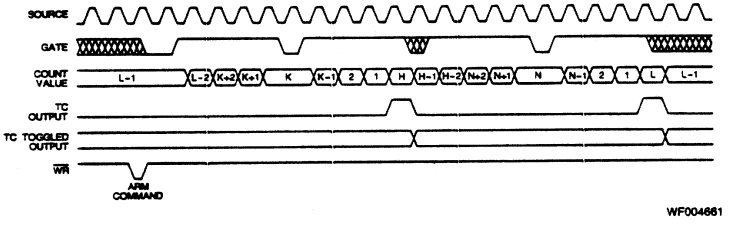


Figure 15h. Mode H Waveforms

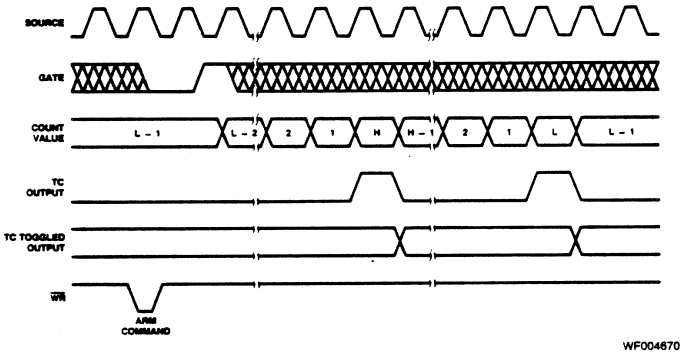


Figure 15i. Mode I Waveforms

MODE J

Variable Duty Cycle Rate Generator with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode J, shown in Figure 15j, will find the greatest usage in frequency generation applications with variable duty cycle requirements. Once armed, the counter will count continuously until it is issued a DISARM command. On the first TC, the counter will be reloaded from the Hold register. Counting will then proceed until the second TC at which time the counter will be reloaded from the Load register. Counting will continue, with the reload source alternating on each TC, until a DISARM command is issued to the counter. (The third TC reloads from the Hold register, the fourth TC reloads from the Load register, etc.) A variable duty cycle output can be generated by specifying the TC Toggled output in the Counter Mode register. The Load and Hold values then directly control the output duty cycle, with high resolution available when relatively high count values are used.

MODE K

Variable Duty Cycle Rate Generator with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode K, shown in Figure 15k, is identical to Mode J, except that source edges are only counted when the Gate is active. The counter must be armed for counting to occur. Once armed, the counter will count all source edges which occur while Gate is active and disregard those source edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. As with Mode J, the reload source used will alternate on each TC, starting with the Hold register on the first TC after any ARM command. When the TC Toggled output is used, this mode allows the Gate to modulate the duty cycle of the output waveform. It can affect both the HIGH and LOW portions of the output waveform.

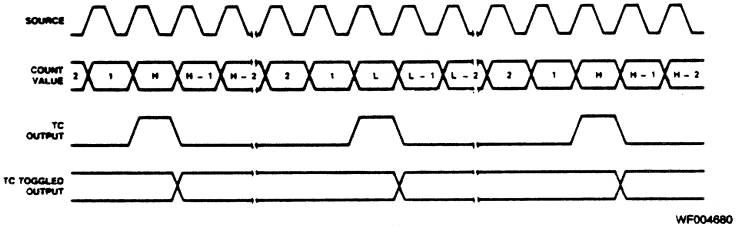


Figure 15j. Mode J Waveforms

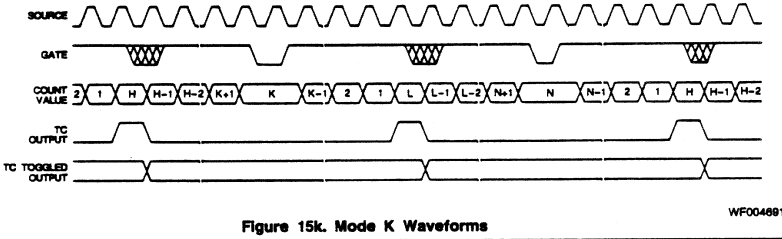


Figure 15k. Mode K Waveforms

2

MODE L

Hardware-Triggered Delayed Pulse One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode L, shown in Figure 15l, is similar to Mode J except that counting will not begin until a Gate edge is applied to an armed counter. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. The counter will start counting source edges after the triggering Gate edge, and counting will proceed until the second TC. Note that after application of a triggering Gate edge, the Gate input will be disregarded for the remainder of the count cycle. This differs from Mode K, where the gate can be modulated throughout the count cycle to stop and start the counter. On the first TC after application of the triggering Gate edge, the counter will be reloaded from the Hold register. On the second TC, the counter will be reloaded from the Load register, and counting will stop until a new gate edge is issued to the counter. Note that unlike Mode K, new Gate edges are required after every second TC to continue counting.

MODE N

Software-Triggered Strobe with Level Gating and Hardware Retriggering

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	0	X	X	X	X	X

Mode N, shown in Figure 15n, provides a software-triggered strobe with level gating that is also hardware retriggerable. The counter must be issued an ARM command before counting can occur. Once armed, the counter will count all source edges which occur while the gate is active and disregard those source edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. After the issuance of the ARM command and the application of an active Gate, the counter will count to TC. Upon reaching TC, the counter will reload from the Load register and automatically disarm itself, inhibiting further counting. Counting will resume upon the issuance of a new ARM command. All active-going Gate edges issued to an armed counter will cause a retrigger operation. Upon application of the Gate edge, the counter contents will be saved in the Hold register. On the first qualified source edge after application of the retriggering gate edge, the contents of the Load register will be transferred into the counter. Counting will resume on the second qualified source edge after the retriggering Gate edge. Qualified source edges are active-going edges which occur while the Gate is active.

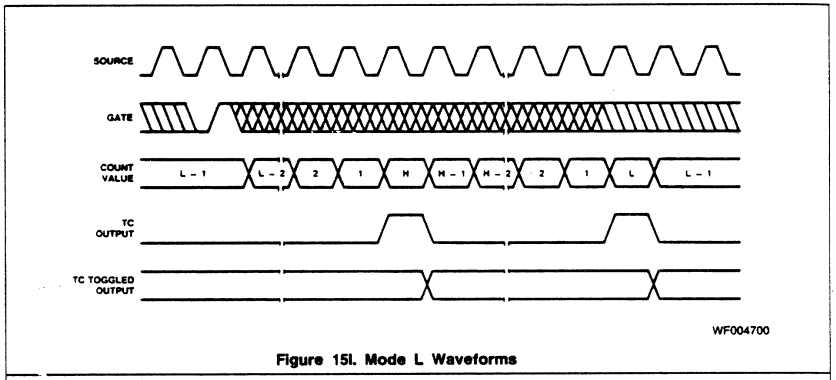


Figure 15l. Mode L Waveforms

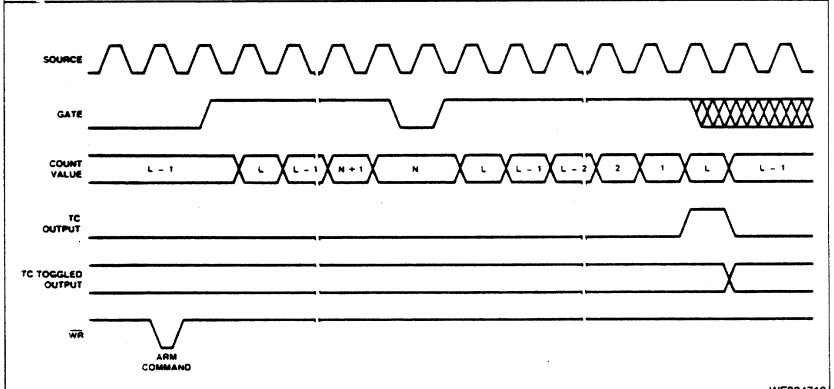


Figure 15n. Mode N Waveforms

MODE O

Software-Triggered Strobe with Edge Gating and Hardware Retriggering

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	0	X	X	X	X	X

Mode O, shown in Figure 15o, is similar Mode N, except that counting will not begin until an active-going Gate edge is applied to an armed counter and the Gate level is not used to modulate counting. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. Irrespective of the Gate level, the counter will count all source edges after the triggering Gate edge until the first TC. On the first TC,the counter will be reloaded from the Load register and disarmed. A new ARM command and a new Gate edge must be applied in that order to initiate a new counting cycle. Unlike Modes C, F, I and L, which disregard the Gate input once counting starts, in Mode O the count process will be retriggered on all active-going Gate edges, including the first Gate edge used to start the counter. On each retriggering Gate edge, the counter contents will be transferred into the Hold register. On the first source edge after the retriggering Gate edge, the Load register contents will be transferred into the counter. Counting will resume on the second-source edge after a retrigger.

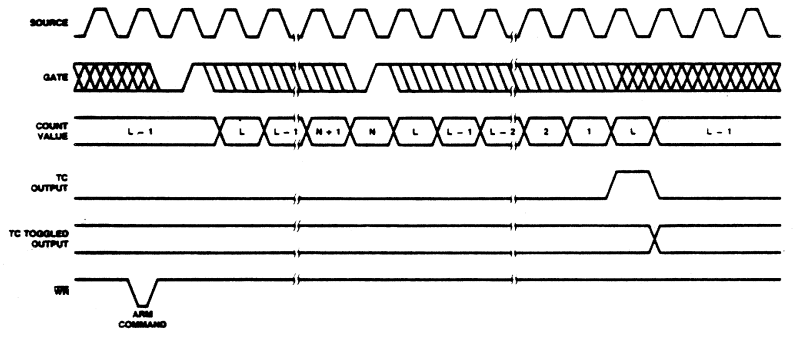


Figure 15o. Mode O Waveforms

MODE Q

Rate Generator with Synchronization (Event Counter with Auto-Read/Reset)

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	1	X	X	X	X	X

Mode Q, shown in Figure 15q, provides a rate generator with synchronization or an event counter with auto-read/reset. The counter must first be issued an ARM command before counting can occur. Once armed, the counter will count all source edges which occur while the Gate is active and disregard those edges which cccur while the Gate is inactive. This permits the Gate to turn the count process on and off. After the issuance of an ARM command and the application of an active Gate, the counter will count to TC repetitively. On each TC,the counter will reload itself from the Load register. The counter may be retriggered at any time by presenting an active-going Gate edge to the Gate input. The retriggering Gate edge will transfer the contents of the counter into the Hold register. The first qualified source edge after the retriggering Gate edge will transfer the contents of the Load register into the Counter. Counting will resume on the second qualified source edge after the retriggering Gate edge. Qualified source edges are active-going edges which occur while the Gate is active.

MODE R

Retriggerable One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	1	X	X	X	X	X

Mode R, shown in Figure 15r, is similar to Mode Q, except that edge gating rather than level gating is used. In other words, rather than use the Gate level to qualify which source edges to count, Gate edges are used to start the counting operation. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. After application of a Gate edge, an armed counter will count all source edges until TC, irrespective of the Gate level. On the first TC, the counter will be reloaded from the Load register and stopped. Subsequent counting will not occur until a new Gate edge is applied. All Gate edges applied to the counter, including the first used to trigger counting, initiate a retrigger operation. Upon application of a Gate edge, the counter contents are saved in the Hold register. On the first source edge after the retriggering Gate edge, the Load register contents will be transferred into the counter. Counting will resume on the second source edge after the retriggering Gate edge.

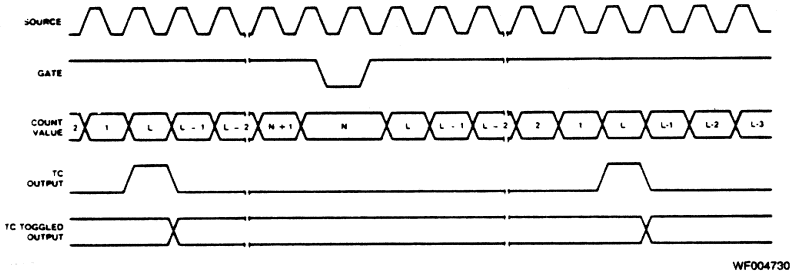


Figure 15q. Mode Q Waveforms

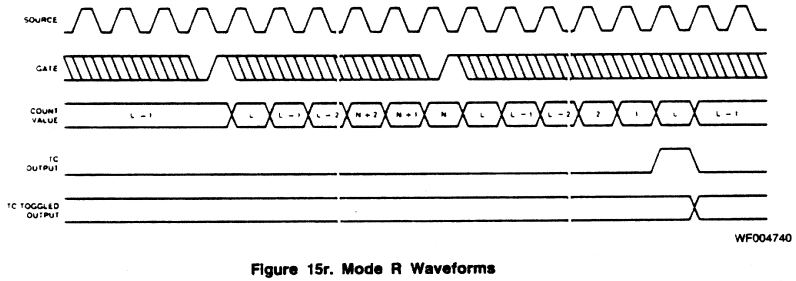


Figure 15r. Mode R Waveforms

MODE S

RELOAD SOURCE

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	0	X	X	X	X	X

In this mode, the reload source for LOAD commands (irrespective of whether the counter is armed or disarmed) and for TC-initiated reloads is determined by the Gate input. The Gate input in Mode S is used only to select the reload source, not to start or modulate counting. When the Gate is Low, the Load register is used; when the Gate is High, the Hold register is used. Note the Low-Load, High-Hold mnemonic convention. Once armed, the counter will count to TC twice and then disarm itself. On each TC, the counter will be reloaded from the reload source selected by the Gate. Following the second TC, an ARM command is required to start a new counting cycle. Mode S is shown in Figure 15s.

MODE V

Frequency-Shift Keying

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	1	X	X	X	X	X

Mode V, shown in Figure 15v, provides frequency-shift keying modulation capability. Gate operation in this mode is identical to that in Mode S. If the Gate is Low, a LOAD command or a TC-induced reload will reload the counter from the Load register. If the Gate is HIGH, LOADs and reloads will occur from the Hold register. The polarity of the Gate only selects the reload source; it does not start or modulate counting. Once armed, the counter will count repetitively to TC. On each TC, the counter will reload itself from the register determined by the polarity of the Gate. Counting will continue in this manner until a DISARM command is issued to the counter. Frequency shift keying may be obtained by specifying a TC Toggled output mode in the Counter Mode register. The switching of frequencies is achieved by modulating the Gate.

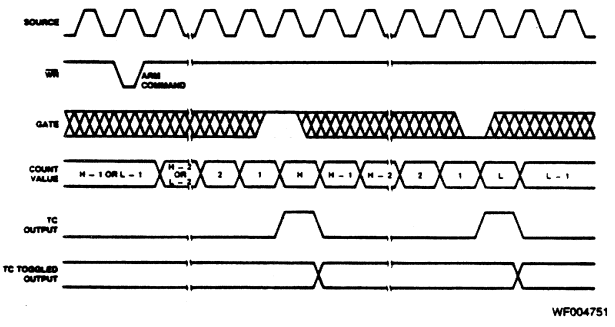


Figure 15s. Mode S Waveforms

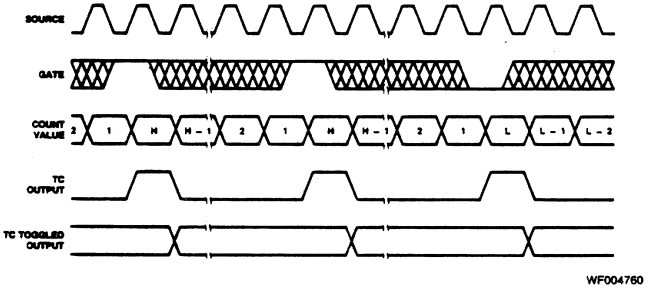


Figure 15v. Mode V Waveforms

2

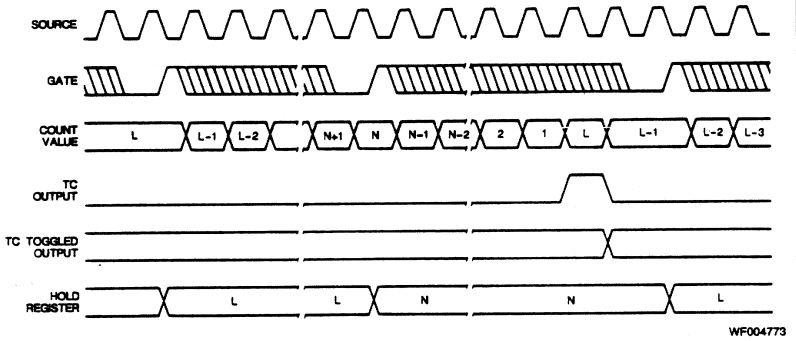


Figure 15x. Mode X Waveforms

MODE X

Hardware Save (available in Am9513A only)

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
Edge			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	1	X	X	X	X	X

Mode X, as shown in Figure 15x, provides a hardware sampling of the counter contents without interrupting the count. A LOAD AND ARM command or a LOAD command followed by an ARM command is required to initialize the counter. Once armed, a Gate edge starts the counting operation; Gate edges applied to a disarmed counter are disregarded. After application of the Triggering Gate edge, the counter will count all qualified source edges until the first TC, irrespective of the gate level. All gate edges applied during the counting sequence will store the current count in the Hold register, but they will not interrupt the counting sequence. On each TC, the counter will be reloaded from the Load register and stopped. Subsequent counting requires a new triggering Gate edge; counting resumes on the first source edge following the triggering Gate edge.

Note: Mode X is only available in the Am9513A devices.

COUNTER MODE CONTROL OPTIONS

Each Counter Logic Group includes a 16-bit Counter Mode (CM) register used to control all of the individual options available with its associated general counter. These options include output configuration, count control, count source and gating control. Figure 16 shows the bit assignments for the Counter Mode registers. This section describes the control options in detail. Note that generally each counter is independently configured and does not depend on information outside its Counter Logic Group. The Counter Mode register should be loaded only when the counter is Disarmed. Attempts to load the Counter Mode register when the counter is armed may result in erratic counter operation.

After power-on reset or a Master Reset command, the Counter Mode registers are initialized to a preset condition. The value entered is 0B00 hex and results in the following control configuration:

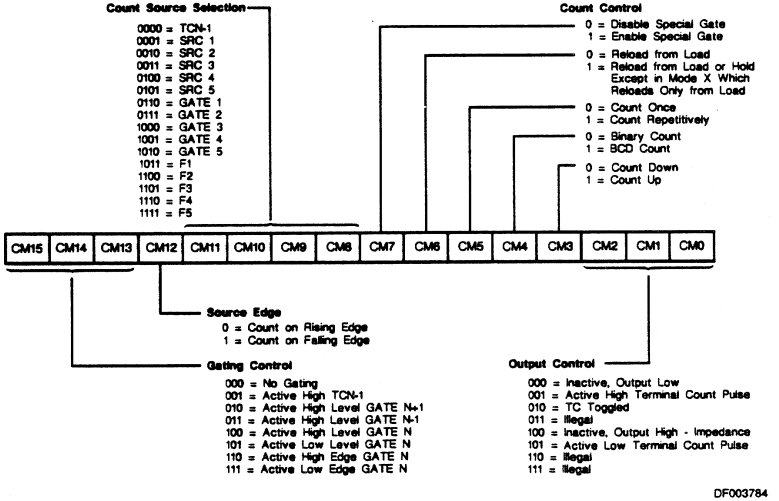
Output low-impedance to ground
Count down
Count binary
Count once
Load register selected
No retriggering
F1 input source selected
Positive-true input polarity
No gating

Output Control

Counter mode bits CM0 through CM2 specify the output control configuration. Figure 17 shows a schematic representation of the output control logic. The OUT pin may be off (a high-impedance state), or it may be inactive with a low-impedance to ground. The three remaining valid combinations represent the active-high, active-low or TC Toggle output waveforms.

One output form available is called Terminal Count (TC) and represents the period in time that the counter reaches an equivalent value of zero. TC will occur on the next count when the counter is at 0001 for down counting, at 9999 (BCD) for BCD up counting or at FFFF (hex) for binary up counting. Figure 18 shows a Terminal Count pulse and an example context that generated it. The TC width is determined by the period of the counting source. Regardless of any gating input or whether the counter is Armed or Disarmed, the terminal count will go active for only one clock cycle. Figure 18 assumes active-high source polarity, counter armed, counter decrementing and an external reload value of K.

The counter will always be loaded from an external location when TC occurs; the user can choose the source location and the value. If a non-zero value is picked, the counter will never really attain a zero state, and TC will indicate the counter state that would have been zero had no parallel transfer occurred.



Note: See Figure 15 for restrictions on Count Control and Gating Control bit combinations.

Figure 16. Counter Mode Register Bit Assignments

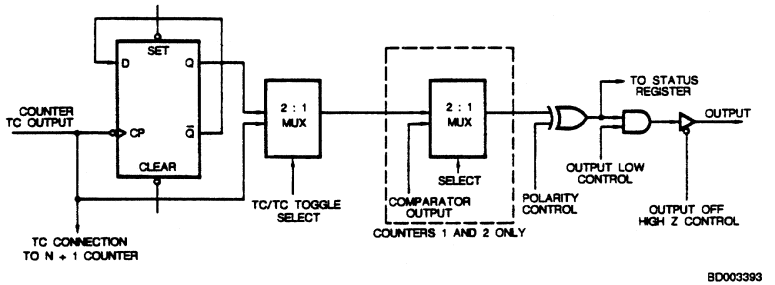


Figure 17. Output Control Logic

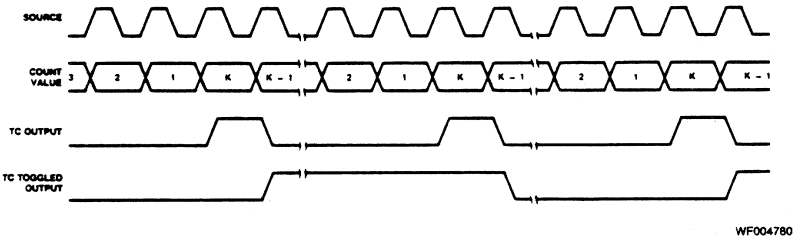


Figure 18. Counter Output Waveforms

The other output form, TC Toggled, uses the trailing edge of TC to toggle a flip-flop to generate an output level instead of a pulse. The toggle output is half the frequency of TC. The TC Toggled output will frequently be used to generate variable duty-cycle square waves in Operating Modes G through K.

In Mode L the TC Toggled output can be used to generate a one-shot function, with the delay to the start of the output pulse and the width of the output pulse separately programmable. With selection of the minimum delay to the start of the pulse, the output will toggle on the second source pulse following application of the triggering Gate edge.

Note that the TC Toggled output form contains no implication about whether the output is active-high or active-low. Unlike the TC output, which generates a transient pulse which can clearly be active-high or active-low, the TC Toggled output waveform only flips the state of the output on each TC. The sole criterion of whether the TC Toggled output is active-high or active-low is the level of the output at the start of the count cycle. This can be controlled by the Set and Clear Output commands. (See Figure 19.)

TC (Terminal Count)

On each Terminal Count (TC), the counter will reload itself from the Load or Hold register. TC is defined as that period of time when the counter contents would have been zero had no reload occurred. Some special conditions apply to counter operation immediately before and during TC.

1. In the clock cycle before TC, an internal signal is generated that commits the counter to go to TC on the next count, and retriggering by a hardware Gate edge (Modes N, O, Q and R) or a software LOAD or LOAD AND ARM command will not extend the time to TC. Note that the "next count" driving the counter to TC can be caused by the application of a count source edge (in level gating modes, the edge must occur while the gate is active, or it will be disregarded), by the application of a LOAD or LOAD AND ARM command (see 2 below) or by the application of a STEP command.
2. If a LOAD or LOAD AND ARM command is executed during the cycle preceding TC, the counter will immediately go to TC. If these commands are issued during TC, the TC state will immediately terminate.
3. When TC is active, the counter will always count the next source edge issued to it, even if it is disarmed or gated off during TC. This means that TC will never be active for longer than one count period and it may, in fact, be shorter if a STEP command or a LOAD or LOAD AND ARM command is applied during TC (see item 2 above). This also means that a counter that is disarmed or stopped on TC is actually disarmed/stopped immediately following TC.

This may cause count sequences different from what a user might expect. Since the counter is always reloaded at the start of TC and since it always counts at the end of TC, the counter contents following TC will differ by one from the reloaded value, irrespective of the operating mode used.

If the reloaded value was 0001 for down counting, 9999 (BCD) for BCD up counting or FFFF (hex) for binary up counting, the count at the end of TC will drive the counter into TC again regardless of whether the counter is gated off or disarmed. As long as these values are reloaded, the TC output will stay active. If a TC Toggled output is selected, it will toggle on each count. Execution of a LOAD, LOAD AND ARM or STEP command with these counter contents will act the same as application of a source pulse, causing TC to remain active and a TC Toggled output to toggle.

Count Control

Counter Mode bits CM3 through CM7 specify the various options available for direct control of the counting process. CM3 and CM4 operate independently of the others and control up/down and BCD/binary counting. They may be combined freely with other control bits to form many types of counting configurations. The other three bits and the Gating Control field interact in complex ways. Bit CM5 controls the repetition of the count process. When CM5 = 1, counting will proceed in the specified mode until the counter is disarmed. When CM5 = 0, the count process will proceed only until one full cycle of operation occurs. This may occur after one or two TC events. The counter is then disarmed automatically. The single or double TC requirement will depend on the state of other control bits. Note that even if the counter is automatically disarmed upon a TC, it always counts the count source edge which generates the trailing TC edge.

When TC occurs, the counter is always reloaded with a value from either the Load register of the Hold register. Bit CM6 specifies the source options for reloading the counter. When CM6 = 0, the contents of the Load register will be transferred into the counter at every occurrence of TC. When CM6 = 1, the counter reload location will be either the Load or Hold Register. The reload location in this case may be controlled externally by using a Gate pin (Modes S and V) or may alternate on each TC (Modes G through L). With alternating sources and with the TC Toggled output selected, the duty cycle of the output waveform is controlled by the relative Load and Hold values and very fine resolution of duty cycles ratios may be achieved.

Bit CM7 controls the special gating functions that allow retriggering and the selection of Load or Hold sources for counter reloading. The use and definition of CM7 will depend

on the status of the Gating Control field and bits CM5 and CM6.

Hardware Retriggering

Whenever hardware retriggering is enabled (Modes N, O, Q, and R), all active-going Gate edges initiate retrigger operations. On application of the Gate edge, the counter contents will be transferred to the Hold register. On the first qualified source edge after application of the retriggering Gate edge, the Load register contents will be transferred into the counter. (Qualified source edges are edges which occur while the counter is gated on and Armed.)

This means that, if level gating is used, the edge occurring on active-going gate transitions will initiate a retrigger. Similarly, when edge gating is enabled, an edge used to start the counter will also initiate a retrigger. The first count source edge applied after the Gate edge will not increment/decrement the counter but retrigger it.

If a LOAD, LOAD AND ARM, or a STEP Command occurs between the retriggering Gate edge and the first qualified source edge, it will be interpreted as a source edge and transfer the Load register contents into the counter. Thereafter, the counter will count all qualified source edges.

When some form of Gating is specified, CM7 controls hardware retriggering. In this case, when CM7 = 0, hardware retriggering does not occur; when CM7 = 1, the counter is retriggered any time an active-going Gate edge occurs. Retriggering causes the counter value to be saved in the Hold register and the Load register contents to be transferred into the counter.

When No Gating is specified, the definition of CM7 changes. In this case, when CM7 = 0, the Gate input has no effect on the counting; when CM7 = 1, the Gate input specifies the source (selecting either the Load or Hold register) used to reload the counter when TC occurs. Figure 14 shows the various available control combinations for these interrelated bits.

Count Source Selection

Counter Mode bits CM8 through CM12 specify the source used as input to the counter and the active edge that is counted. Bit CM12 controls the polarity for all the sources; logic zero counts rising edges and logic one counts falling edges. Bits CM8 through CM11 select 1 of 16 counting sources to route to the counter input. Five of the available inputs are internal frequencies derived from the internal oscillator (see Figure 13 for frequency assignments). Ten of the available inputs are interface pins; five are labeled SRC and five are labeled GATE.

The 16th available input is the TC output from the adjacent lower-numbered counter. (The Counter 5 TC wraps around to the Counter 1 input.) This option allows internal concatenating that permits very long counts to be accumulated. Since all five counters may be concatenated, it is possible to configure a counter that is 80-bits long on one Am9513A chip. When TCN-1 is the source, the count ripples between the connected counters. External connections can also be made, and can use the toggle bit for even longer counts. This is easily accomplished by selecting a TC Toggled output mode and wiring OUTN to one of the SRC inputs.

Gating Control

Counter Mode bits CM15, CM14, CM13 specify the hardware gating options. When "no gating" is selected (000), the counter will proceed unconditionally as long as it is armed. For any other gating mode, the count process is conditioned by the specified gating configuration.

For a code of 100 in this field, counting can proceed only when the pin labeled GATEN associated with Counter N is at a logic high level. When it goes LOW, counting is simply suspended until the Gate goes HIGH again. A code of 101 performs the same function with an opposite active polarity. Codes 010 and 011 offer the same function as 100, but specify alternate input pins as Gating Sources. This allows any of three interface pins to be used as gates for a given counter. On Counter 4, for example, pin 34, pin 35 or pin 36 may be used to perform the gating function. This also allows a single Gate pin to simultaneously control up to three counters. Counters 1 and 5 are considered adjacent when using TCN-1 (001), Gate N + 1 (010) and Gate N – 1 (011) controls.

For codes of 110 or 111 in this field, counting proceeds after the specified active Gate edge until one or two TC events occur. Within this interval, the Gate input is ignored, except for the retriggering option. When repetition is selected, a cycle will be repeated as soon as another Gate edge occurs. With repetition selected, any Gate edge applied after TC goes active will start a new count cycle. Edge gating is useful when implementing a digital single-shot since the gate can serve as a convenient firing trigger.

A 001 code in this field selects the TC (not TOGGLE) output from the adjacent lower-numbered counter as the gate. This is useful for synchronous counting when adjacent counters are concatenated.

COMMAND DESCRIPTIONS

The command set for the Am9513A allows the host processor to customize and manage the operating modes and features for particular applications, to initialize and update both the internal data and control information, and to manipulate operating bits during operation. Commands are entered directly into the 8-bit Command register by writing into the Control port (see Figure 6).

All available commands are described in the following text. Figure 19 summarizes the command codes and includes a brief description of each function. Figure 20 shows all the unused code combinations; unused codes should not be entered into the Command register since undefined activities may occur.

Six of the command types are used for direct software control of the counting process and they each contain a 5-bit S field. In a linear-select fashion, each bit in the S field corresponds to one of five general counters (S1 = Counter 1, S2 = Counter 2, etc.). When an S bit is a one, the specified operation is performed on the counter so designated; when an S bit is a zero, no operation occurs for the corresponding counter. This type of command format has three basic advantages. It saves host software by allowing any combination of counters to be acted on by a single command. It allows simultaneous action on multiple counters where synchronization of commands is important. It allows counter-specific service routines to control individual counters without needing to be aware of the operating context of other counters.

Three of the commands use a 3-bit binary code (N4, N2, N1) to identify the affected counter (a 001 programs counter 1, etc.). Unlike the previously mentioned commands, these commands allow you to program only one counter at a time.

Command Code								Command Description
C7	C6	C5	C4	C3	C2	C1	C0	
0	0	0	E2	E1	G4	G2	G1	Load Data Pointer register with contents of E and G fields. (G ≠ 000, G ≠ 110)
0	0	1	S5	S4	S3	S2	S1	Arm counting for all selected counters
0	1	0	S5	S4	S3	S2	S1	Load contents of specified source into all selected counters
0	1	1	S5	S4	S3	S2	S1	Load and Arm all selected counters*
1	0	0	S5	S4	S3	S2	S1	Disarm and Save all selected counters
1	0	1	S5	S4	S3	S2	S1	Save all selected counters in Hold register
1	1	0	S5	S4	S3	S2	S1	Disarm all selected counters
1	1	1	0	1	N4	N2	N1	Set Toggle out (HIGH) for counter N (001 ≤ N ≤ 101)
1	1	1	0	0	N4	N2	N1	Clear Toggle out (LOW) for counter N (001 ≤ N ≤ 101)
1	1	1	1	0	N4	N2	N1	Step counter N (001 ≤ N ≤ 101)
1	1	1	0	1	0	0	0	Set MM14 (Disable Data Pointer Sequencing)
1	1	1	0	1	1	1	0	Set MM12 (Gate off FOUT)
1	1	1	0	1	1	1	1	Set MM13 (Enter 16-bit bus mode)
1	1	1	0	0	0	0	0	Clear MM14 (Enable Data Pointer Sequencing)
1	1	1	0	0	1	1	0	Clear MM12 (Gate on FOUT)
1	1	1	0	0	1	1	1	Clear MM13 (Enter 8-bit bus mode)
1	1	1	1	1	0	0	0	Enable Prefetch for Write operations (A¬9513'A' only)
1	1	1	1	1	0	0	1	Disable Prefetch for Write operations (Am9513'A' only)
1	1	1	1	1	1	1	1	Master reset

*Not to be used for asynchronous operations.

Figure 19. Am9513A Command Summary

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	0	0	0	0
1	1	1	1	0	1	1	0
1	1	1	1	0	1	1	1
0	0	0	X	X	1	1	0
0	0	0	X	X	0	0	0
*1	1	1	1	1	X	X	X

*Unused except when XXX = 111, 001 or 000.

Figure 20. Am9513A Unused Command Codes

Arm Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	0	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be enabled for counting. A counter must be armed before counting can commerce. Once armed, the counting process may be further enabled or disabled using the hardware gating facilities. This command can only arm or do nothing for a given counter; a zero in the S field does not disarm the counter.

ARM and DISARM commands can be used to gate counter operation on and off under software control. DISARM commands entered while a counter is in the TC state will not take effect until the counter leaves TC. This ensures that the counter never latches up in a TC state. (The counter may leave the TC state because of application of a count source edge, execution of a LOAD or LOAD AND ARM command, or execution of a STEP command.)

In modes which alternate reload sources (Modes G – L), the ARMing operation is used as a reset for the logic which determines which reload source to use on the upcoming TC. Following each ARM or LOAD AND ARM command, a counter in one of these modes will reload from the Hold register on the first TC and alternate reload sources thereafter (reload from the Load register on the second TC, the Hold register on the third, etc.).

Load Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	1	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified in the S field, will be loaded with previously entered values. The source of information for each counter will be either the associated Load register or the associated Hold register, as determined by the operating configuration in the Mode register. The Load/Hold contents are not changed. This command will cause a transfer independent of any current operating configuration for the counter. It will often be used as a software retrigger or as counter initialization prior to active hardware gating.

If a LOAD or LOAD AND ARM command is executed during the cycle preceding TC, the counter will go immediately to TC. This occurs because the LOAD operation is performed by generating a pseudo-count pulse internal to the Am9513A, and the Am9513A is expecting to go into TC on the next count pulse. The reload source used to reload the counter will be the same as that which would have been used if the TC were generated by a source edge rather than by the LOAD operation.

Execution of a LOAD or LOAD AND ARM command while the counter is in TC will cause the TC to end. For Armed counters in all modes except S or V, the LOAD source used will be that to be used for the upcoming TC. (The LOADing operation will not alter the selection of reload source for the upcoming TC.) For Disarmed counters in modes except S or V, the reload sources used will be the LOAD register. For modes S or V, the reload source will be selected by the GATE input, regardless of whether the counter is Armed or Disarmed.

Special considerations apply when modes with alternating reload sources are used (Modes G – L). If a LOAD command drives the counter to TC in these modes, the reload source for the next TC will be from the opposite reload location. In other words, the LOAD-generated TC will cause the reload sources to alternate just as a TC generated by a source edge would. Note that if a second LOAD command is issued during the LOAD-generated TC (or during any other TC, for that matter), the second LOAD command will terminate the TC and cause a reload from the source designated for use with the next TC. The second LOAD will not alter the reload source for the next TC since the second LOAD does not generate a TC; reload sources alternate on TCs only, not on LOAD commands.

Load and Arm Counters*

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	1	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified in the S field, will be first loaded and then armed. This command is equivalent to issuing a LOAD command and then an ARM command.

A LOAD AND ARM command which drives a counter to TC generates the same sequence of operations as execution of a LOAD command and then an ARM command. In modes which disarm on TC (Modes A – C and N – O, and Modes G – I and S if the current TC is the second in the cycle), the ARM part of the LOAD AND ARM command will re-enable counting for another cycle. In modes which alternate reload sources (Modes G – L), the ARMing operation will cause the next TC to reload from the HOLD register, irrespective of which reload source the current TC used.

*This command should not be used during asynchronous operations.

Disarm Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be disabled from counting. A disarmed counter will cease all counting independent of other conditions. The only exception to this is that a counter in the TC state will always count once, in order to leave TC, before DISARMing. This count may be generated by a source edge, by a LOAD or LOAD AND ARM command (the LOAD AND ARM command will negate the DISARM command) or by a STEP command. A disarmed counter may be updated using the LOAD command and may be read using the SAVE command. A count process may be resumed using an ARM command. See the ARM command description for further details.

Save Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	0	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by S field, will have their contents transferred into their associated Hold register. The transfer takes place without interfering with any counting that may be underway. This command will overwrite any previous Hold register contents. The SAVE command is designed to allow an accumulated count to be preserved so that it can be read by the host CPU at some later time.

Disarm and Save Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	0	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be disarmed, and the contents of the counter will be transferred into the associated Hold registers. This command is identical to issuing a DISARM command followed by a SAVE command.

Set TC Toggle Output

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	N4	N2	N1

(001 ≤ N ≤ 101)

Description: The initial output level for TC Toggle mode is set (HIGH) for counter N selected by N4, N2, N1 = 001 (Counter 1) thru 101 (Counter 5) respectively. This command conditions the TC Toggle flip-flop (see Figure 17) but does not appear at the counter output unless TC Toggle mode (CM2, CM1, CM0 = 010) is selected.

Clear TC Toggle Output

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	N4	N2	N1

(001 ≤ N ≤ 101)

Description: The initial output level for TC Toggle mode is Cleared (LOW) for counter N selected by N4, N2, N1 = 001 (Counter 1) thru 101 (Counter 5) respectively. This command conditions the TC Toggle flip-flop (see Figure 17) but does not appear at the counter output unless TC Toggle mode (CM2, CM1, CM0 = 010) is selected.

Step Counter

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	0	N4	N2	N1

(001 ≤ N ≤ 101)

Description: Counter N is incremented or decremented by one, depending on its operating configuration. If the Counter Mode register associated with the selected counter has its CM3 bit cleared to zero, this command will cause the counter to decrement by one. If CM3 is set to a logic high, this command will increment the counter by one. The STEP command will take effect even on a disarmed counter.

Load Data Pointer Register

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	0	0	E2	E1	G4	G2	G1

(G4, G2, G1 ≠ 000, ≠ 110)

Description: Bits in the E and G fields will be transferred into the corresponding Element and Group fields of the Data Pointer register as shown in Figure 7. The Byte Pointer bit in the Data Pointer register is set. Transfers into the Data Pointer only occur for G field values of 001, 010, 011, 100, 101 and 111. Values of 000 and 110 for G should not be used. See the "Setting the Data Pointer Register" section of this document for additional details.

Disable Data Pointer Sequencing

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	0	0	0

Description: This command sets Master Mode bit 14 without affecting other bits in the Master Mode register. MM14 controls the automatic sequencing of the Data Pointer register. Disabling the sequencing allows repetitive host processor access to a given internal location without repetitive updating of the Data Pointer. MM14 may also be controlled by loading a full word into the Master Mode register.

Enable Data Pointer Sequencing

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	0	0	0

Description: This command clears Master Mode bit 14 without affecting other bits in the Master Mode register. MM14 controls the automatic sequencing of the Data Pointer register. Enabling the sequencing allows sequential host processor access to several internal locations without repetitive updating of the Data Pointer. MM14 may also be controlled by loading a full word into the Master Mode register. See the "Data Pointer Register" section of this document for additional information on Data Pointer sequencing

Enable 16-Bit Data Bus

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	1	1	1

Description: This command sets Master Mode bit 13 without affecting other bits in the Master Mode register. MM13 controls the multiplexer in the data bus buffer. When MM13 is set, no multiplexing takes place and all 16 external data bus lines are used to transfer information into and out of the STC. MM13 may also be controlled by loading the full Master Mode register in parallel.

Enable 8-Bit Data Bus

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	1	1	1

Description: This command clears Master Mode bit 13 without affecting other bits in the Master Mode register. MM13 controls the multiplexer in the data bus buffer. When MM13 is cleared, the multiplexer is enabled and 16-bit internal information is transferred eight bits at a time to the eight low-order external data bus lines. MM13 may also be controlled by loading the full Master Mode register in parallel.

Gate Off FOUT

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	1	1	0

Description: This command sets Master Mode bit 12 without affecting other bits in the Master Mode register. MM12 controls the output state of the FOUT signal. When gated off, the FOUT line will exhibit a low-impedance to ground. MM12 may also be controlled by loading the full Master Mode register in parallel.

Gate On FOUT

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	1	1	0

Description: This command clears Master Mode bit 12 without affecting other bits in the Master Mode register. MM12 controls the output status of the FOUT signal. When MM12 is cleared, FOUT will become active and will drive out the selected and divided FOUT signal. MM12 may also be controlled by loading the full Master Mode register in parallel. When FOUT is gated on or off, a transient pulse may be generated on the FOUT signal.

Disable Prefetch for Write Operations

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	0	0	1

Description: This command disables the prefetch circuitry during Write operations (it does not affect Read operations). This reduces the write recovery time and allows the user to use block move instructions for initialization of the Am9513A registers. Once prefetch is disabled for writing, an Enable Prefetch for Write or a Reset command is necessary to re-enable the prefetch circuitry for writing.
Note: This command is only available in Am9513A devices; it is an illegal command in the "non-A" Am9513 device.

Enable Prefetch for Write Operations

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	0	0	0

Description: This command re-enables the prefetch circuitry for Write operations. It is used only to terminate the Disable Prefetch Command.
Note: This command is only available in Am9513A devices; it is an illegal command in the "non-A" Am9513 device.

Master Reset

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	1	1	1

Description: The Master Reset command duplicates the action of the power-on reset circuitry. It disarms all counters, enters 0000 in the Master Mode, Load and Hold registers and enters 0B00 (hex) in the Counter Mode registers.

Following either a power-up or software reset, the LOAD command should be applied to all the counters to clear any that may be in a TC state. The Data Pointer register should also be set to a legal value, since reset does not initialize it. A complete reset operation follows.

1. Using the procedure given in the "Command Initiation" section of this document, enter the FF (hex) command to perform a software reset.
2. Using the "Command Initiation" procedure, enter the LOAD command for all counters, opcode 5F (hex).
3. Using the procedure given in the "Setting the Data Pointer Register" section of this document, set the Data Pointer to a valid code. The legal Data Pointer codes are given in Figure 8.

The Master Mode, Counter Mode, Load and Hold registers can now be initialized to the desired values.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
VCC with Respect to VSS ..... −0.5 V to +7.0 V
All Signal Voltages
with Respect to VSS ..... −0.5 V to +7.0 V
Power Dissipitation (Package Limitation) ..... 1.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

Military (M) Devices
Temperature (T_C) ..... −55 to +125°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

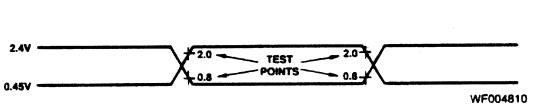
Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified.

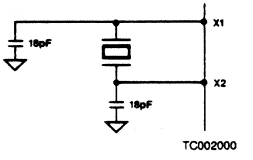
Parameters	Description		Test Conditions	Min	Max		Units
VIL	Input Low Voltage	All Inputs Except X2		VSS − 0.5	0.8		Volts
		X2 Input		VSS − 0.5	0.8		
VIH	Input High Voltage	All Input Except X2		2.2 V	VCC		Volts
		X2 Input		3.8	VCC		
VITH	Input Hysteresis (SRC and GATE Inputs Only)			0.2			Volts
VOL	Output Low Voltage		IOL = 3.2 mA		0.4		Volts
VOH	Output High Voltage		IOH = −200 µA	2.4			Volts
IIX	Input Load Current (Except X2)		VSS ≤ VIN ≤ VCC		±10		µA
IIX	Input Load Current X2		VSS ≤ VIN ≤ VCC		±100		µA
IOZ	Output Leakage Current (Except X1)		VSS + 0.4 ≤ VOUT ≤ VCC High-Impedance State		±25		µA
ICC	VCC Supply Current (Steady State)				255	275	mA
CIN	Input Capacitance		f = 1 MHz, T_A = +25°C. All pins not under test at 0 V.		10*	20*	pF
COUT	Output Capacitance				15*	20*	
CIO	IN/OUT Capacitance				20*	20*	

* Guaranteed by design.

SWITCHING TEST INPUT/OUTPUT WAVEFORMS



WF004810



TC002000

Crystal is fundamental mode parallel resonant 32 pF load capacitance less than 100 Ω ESR C_o less than 100 pF.

The second and fourth letters designate the reference states of the signals named in the first and third letters respectively, using the following abbreviations.

H = HIGH
L = LOW
V = VALID
X = Unknown or Don't care
Z = High-Impedance

2. Any input transition that occurs before this minimum setup requirement will be reflected in the contents read from the status register.
3. Any input transition that occurs before this minimum setup requirement will act on the counter before the execution of the operation initiated by the write and the counter may be off by one count.
4. Any input transition that occurs after this minimum hold time is guaranteed to not influence the contents read from the status register on the current read operation.
5. Any input transition that occurs after this minimum hold time is guaranteed to be seen by the counter as occurring after the action initiated by the write operation and the counter may be off by one count.
6. This parameter applies to cases where the write operation causes a change in the output bit.
7. The enabled count source is one of F1 – F5, TCN-1 SRC1 – SRC5 or GATE1 – GATE 5, as selected in the applicable Counter Mode register. The timing diagram assumes the counter counts on rising source edges. The timing specifications are the same for falling-edge counting.
8. This parameter applies to edge gating (CM15 – CM13 = 110 or 111) and gating when both CM7 = 1 and CM15 – CM13 ≠ 000. This parameter represents the minimum GATE pulse width needed to ensure that the pulse initiates counting or counter reloading.
9. This parameter applies to both edge and level gating (CM15 – CM13 = 001 through 111 and CM7 = 0). This parameter represents the minimum setup or hold times to ensure that the Gate input is seen at the intended level on the active source edge and the counter may be off by one count.
10. This parameter assumes that the GATENA input is unused (16-bit bus mode) or is tied high. In cases where the GATENA input is used, this timing specification must be met by both the GATE and GATENA inputs.
11. Signals F1 – F5 cannot be directly monitored by the user. The phase difference between these signals will manifest itself by causing counters using two different F signals to count at different times on nominally simultaneous transitions in the F signals. F1 = X2.
12. This timing specification assumes that $\overline{CS}$ is active whenever $\overline{RD}$ or $\overline{WR}$ are active. $\overline{CS}$ may be held active indefinitely.
13. This parameter assumes X2 is driven from an external gate with a square wave.
14. This parameter assumes that the write operation is to the command register.
15. This timing specification applies to single-action commands only (e.g., LOAD, ARM, SAVE, etc.). For double-action commands such as LOAD AND ARM and DISARM AND SAVE, TWHEH minimum = 700 ns.
16. In short data write mode, TWHRL and TWHWL minimum = 1000 ns.
17. This parameter applies to the hardware retrigger/save modes N, O, Q, R, and X (CM7 = 1 and CM15 – CM13 < > 000). This parameter ensures that the gating pulse initiates a hardware retrigger/save operation.
18. This parameter applies to hardware load source select modes S and V (CM7 = 1 and CM15 – CM13 = 000). This parameter represents the minimum hold time to ensure that the GATE input selects the correct load source on the active source edge.

SWITCHING CHARACTERISTICS over **COMMERCIAL** operating ranges unless otherwise specified (Note 1)

Parameters	Description		Figure	Am9513A Min	Am9513A Max	Unit
TAVRL	C/$\overline{D}$ Valid to Read Low		21	25		ns
TAVWH	C/$\overline{D}$ Valid to Write High		21	170		ns
TCHCH	X2 High to X2 High (X2 Period) (Note 13)		22	145		ns
TCHCL	X2 High to X2 Low (X2 High Pulse Width) (Note 13)		22	70		ns
TCLCH	X2 Low to X2 High (X2 Low Pulse Width) (Note 13)		22	70		ns
TDVWH	Data In Valid to Write High		21	80		ns
TEHEH	Count Source High to Count Source High (Source Cycle Time) (Note 7)		22	145		ns
TEHEL TELEH	Count Source Pulse Duration (Note 7)		22	70		ns
TEHFV	Count Source High to FOUT Valid (Note 7)		22		500	ns
TEHGV	Count Source High to Gate Valid (Level Gating Hold Time) (Notes 7, 9, 10)		22	10		ns
TEHRL	Count Source High to Read Low (Set-up Time) (Notes 2, 7)		21	190		ns
TEHWH	Count Source High to Write High (Set-up Time) (Notes 3, 7)		21	-100		ns
TEHYV	Count Source High to Out Valid (Note 7)	TC Output	22		300	ns
		Immediate or Delayed Toggle Output	22		300	
		Comparator Output	22		350	
TFN	FN High to FN + 1 Valid (Note 11)		22		75	ns
TGVEH	Gate Valid to Count Source High (Level Gating Set-up Time) (Notes 7, 9, 10)		22	100		ns
TGVGV	Gate Valid to Gate Valid (Gate Pulse Duration) (Notes 8, 10)		22	145		ns
TGVWH	Gate Valid to Write High (Notes 3, 10)		21	-100		ns
TRHAX	Read High to C/$\overline{D}$ Don't Care		21	0		ns
TRHEH	Read High to Count Source High (Notes 4, 7)		21	0		ns
TRHQX	Read High to Data Out Invalid		21	10		ns
TRHQZ	Read High to Data Out at High-Impedance (Data Bus Release Time)		21		85	ns
TRHRL	Read High to Read Low (Read Recovery Time)		21	1000		ns
TRHSH	Read High to $\overline{CS}$ High (Note 12)		21	0		ns
TRHWL	Read High to Write Low (Read Recovery Time)		21	1000		ns
TRLQV	Read Low to Data Out Valid		21		110	ns
TRLQX	Read Low to Data Bus Driven (Data Bus Drive Time)		21	20		ns
TRLRH	Read Low to Read High (Read Pulse Duration) (Note 12)		21	160		ns
TSLRL	$\overline{CS}$ Low to Read Low (Note 12)		21	20		ns
TSLWH	$\overline{CS}$ Low to Write High (Note 12)		21	170		ns
TWHAX	Write High to C/$\overline{D}$ Don't Care		21	20		ns
TWHDX	Write High to Data In Don't Care		21	20		ns
TWHEH	Write High to Count Source High (Notes 5, 7, 14, 15)		21	550		ns
TWHGV	Write High to Gate Valid (Notes 5, 10, 14)		21	475		ns
TWHRL	Write High to Read Low (Write Recovery Time) (Note 16)		21	1500*		ns
TWHSH	Write High to $\overline{CS}$ High (Note 12)		21	20		ns
TWHWL	Write High to Write Low (Write Recovery Time) (Note 16)		21	1500*		ns
TWHYV	Write High to Out Valid (Notes 6, 14)		21		650	ns
TWLWH	Write Low to Write High (Write Pulse Duration) (Note 12)		21	150		ns
TGVEH2	Gate Valid to Count Source High (Special Gate) (Notes 10, 13, 17)		22	200		ns
TEHGV2	Count Source High to Gate Valid (Special Gate) (Notes 10, 13, 18)		22	80		ns

Notes:

1. Abbreviations used for the switching parameter symbols are given as the letter T followed by four or five characters. The first and third characters represent the signal names on which the measurements start and end. Signal abbreviations used are:

   A (Address) = C/$\overline{D}$
   C (Clock) = X2
   D (Data In) = DB0-DB15
   E (Enabled counter source input) = SRC1 - SRC5, GATE1 - GATE5, F1 - F5,TCN-1
   F = FOUT
   G (Counter gate input) = GATE1 - GATE5, TCN-1
   Q (Data Out) = DB0 - DB15
   R (Read) = $\overline{RD}$
   S (Chip Select) = $\overline{CS}$
   W (Write) = $\overline{WR}$
   Y (Output) = OUT1 - OUT5

SWITCHING CHARACTERISTICS over **MILITARY** operating range (for SMD/DESC and APL Products, Group A, Subgroups 9, 10, 11 are tested unless otherwise noted)

Parameter Symbol	Description		Am9513A Min.	Am9513A Max.	Unit
TAVRL	C/$\overline{D}$ Valid to Read Low		25		ns
TAVWH	C/$\overline{D}$ Valid to Write High		170		ns
TCHCH	X2 High to X2 High (X2 Period) (Note 13)		145		ns
TCHCL	X2 High to X2 Low (X2 High Pulse Width) (Note 13)		70		ns
TCLCH	X2 Low to X2 High (X2 Low Pulse Width) (Note 13)		70		ns
TDVWH	Data In Valid to Write High		80		ns
TEHEH	Count Source High to Count Source High (Source Cycle Time) (Note 7)		145		ns
TEHEL TELEH	Count Source Pulse Duration (Note 7)		70		ns
TEHFV	Count Source High to FOUT Valid (Note 7)			500	ns
TEHGV	Count Source High to Gate Valid (Level Gating Hold Time) (Notes 7, 9, 10)		10		ns
TEHRL	Count Source High to Read Low (Set-up Time) (Notes 2, 7)		190		ns
TEHWH	Count Source High to Write High (Set-up Time) (Notes 3, 7)		-100		ns
TEHYV	Count Source High to Out Valid (Note 7)	TC Output		300	ns
		Immediate or Delayed Toggle Output		300	
		Comparator Output		350	
TFN	FN High to FN + 1 Valid (Note 11)			75	ns
TGVEH	Gate Valid to Count Source High (Level Gating Set-up Time) (Notes 7, 9, 10)		100		ns
TGVGV	Gate Valid to Gate Valid (Gate Pulse Duration) (Notes 8, 10)		145		ns
TGVWH	Gate Valid to Write High (Notes 3, 10)		-100		ns
TRHAX	Read High to C/$\overline{D}$ Don't Care		0		ns
TRHEH	Read High to Count Source High (Notes 4, 7)		0		ns
TRHQX	Read High to Data Out Invalid		10		ns
TRHQZ	Read High to Data Out at High-Impedance (Data Bus Release Time)			85	ns
TRHRL	Read High to Read Low (Read Recovery Time)		1000		ns
TRHSH	Read High to $\overline{CS}$ High (Note 12)		0		ns
TRHWL	Read High to Write Low (Read Recovery Time)		1000		ns
TRLQV	Read Low to Data Out Valid			110	ns
TRLQX	Read Low to Data Bus Driven (Data Bus Drive Time)		20		ns
TRLRH	Read Low to Read High (Read Pulse Duration) (Note 12)		160		ns
TSLRL	$\overline{CS}$ Low to Read Low (Note 12)		20		ns
TSLWH	$\overline{CS}$ Low to Write High (Note 12)		170		ns
TWHAX	Write High to C/$\overline{D}$ Don't Care		20		ns
TWHDX	Write High to Data In Don't Care		20		ns
TWHEH	Write High to Count Source High (Notes 5, 7, 14, 15)		550		ns
TWHGV	Write High to Gate Valid (Notes 5, 10, 14)		475		ns
TWHRL	Write High to Read Low (Write Recovery Time) (Note 16)		1500		ns
TWHSH	Write High to $\overline{CS}$ High (Note 12)		20		ns
TWHWL	Write High to Write Low (Write Recovery Time) (Note 16)		1500		ns
TWHYV	Write High to Out Valid (Notes 6, 14)			650	ns
TWLWH	Write Low to Write High (Write Pulse Duration) (Note 12)		150		ns
TGVEH2	Gate Valid to Count Source High (Special Gate) (Notes 10, 13, 17)		200		ns
TEHGV2	Count Source High to Gate Valid (Special Gate) (Notes 10, 13, 18)		80		ns

Notes:

1. Abbreviations used for the switching parameter symbols are given as the letter T followed by four or five characters. The first and third characters represent the signal names on which the measurements start and end. Signal abbreviations used are:

   A (Address) = C/$\overline{D}$
   C (Clock) = X2
   D (Data In) = DB0 - DB15
   E (Enabled counter source input) = SRC1 - SRC5, GATE1 - GATE5, F1 - F5,TCN-1
   F = FOUT
   G (Counter gate input) = GATE1 - GATE5, TCN-1
   Q (Data Out) = DB0 - DB15
   R (Read) = $\overline{RD}$
   S (Chip Select) = $\overline{CS}$
   W (Write) = $\overline{WR}$
   Y (Output) = OUT1 - OUT5

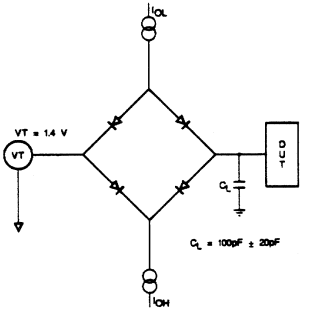
The second and fourth letters designate the reference states of the signals named in the first and third letters respectively, using the following abbreviations.

H = HIGH
L = LOW
V = VALID
X = Unknown or Don't care
Z = High-Impedance

2. Any input transition that occurs before this minimum setup requirement will be reflected in the contents read from the status register.
3. Any input transition that occurs before this minimum setup requirement will act on the counter before the execution of the operation initiated by the write and the counter may be off by one count.
4. Any input transition that occurs after this minimum hold time is guaranteed to not influence the contents read from the status register on the current read operation.
5. Any input transition that occurs after this minimum hold time is guaranteed to be seen by the counter as occurring after the action initiated by the write operation and the counter may be off by one count.
6. This parameter applies to cases where the write operation causes a change in the output bit.
7. The enabled count source is one of F1 – F5, TCN-1 SRC1 – SRC5 or GATE1 – GATE 5, as selected in the applicable Counter Mode register. The timing diagram assumes the counter counts on rising source edges. The timing specifications are the same for falling-edge counting.
8. This parameter applies to edge gating (CM15 – CM13 = 110 or 111) and gating when both CM7 = 1 and CM15 – CM13 ≠ 000. This parameter represents the minimum GATE pulse width needed to ensure that the pulse initiates counting or counter reloading.
9. This parameter applies to both edge and level gating (CM15 – CM13 = 001 through 111 and CM7 = 0). This parameter represents the minimum setup or hold times to ensure that the Gate input is seen at the intended level on the active source edge and the counter may be off by one count.
10. This parameter assumes that the GATENA input is unused (16-bit bus mode) or is tied high. In cases where the GATENA input is used, this timing specification must be met by both the GATE and GATENA inputs.
11. Signals F1 – F5 cannot be directly monitored by the user. The phase difference between these signals will manifest itself by causing counters using two different F signals to count at different times on nominally simultaneous transitions in the F signals. F1 = X2.
12. This timing specification assumes that $\overline{CS}$ is active whenever $\overline{RD}$ or $\overline{WR}$ are active. $\overline{CS}$ may be held active indefinitely.
13. This parameter assumes X2 is driven from an external gate with a square wave.
14. This parameter assumes that the write operation is to the command register.
15. This timing specification applies to single-action commands only (e.g., LOAD, ARM, SAVE, etc.). For double-action commands such as LOAD AND ARM and DISARM AND SAVE, TWHEH minimum = 700 ns.
16. In short data write mode, TWHRL and TWHWL minimum = 1000 ns.
17. This parameter applies to the hardware retrigger/save modes N, O, Q, R, and X (CM7 = 1 and CM15 – CM13 < > 000). This parameter ensures that the gating pulse initiates a hardware retrigger/save operation.
18. This parameter applies to hardware load source select modes S and V (CM7 = 1 and CM15 – CM13 = 000). This parameter represents the minimum hold time to ensure that the GATE input selects the correct load source on the active source edge.

2

SWITCHING TEST CIRCUIT



TC003853

This test circuit is the dynamic load of a Teradyne J941.

SWITCHING WAVEFORMS

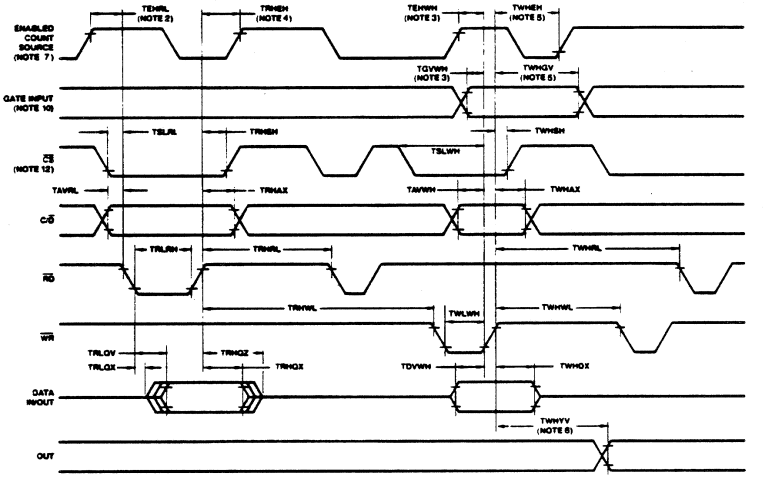


Figure 21. Bus Transfer Switching Waveforms

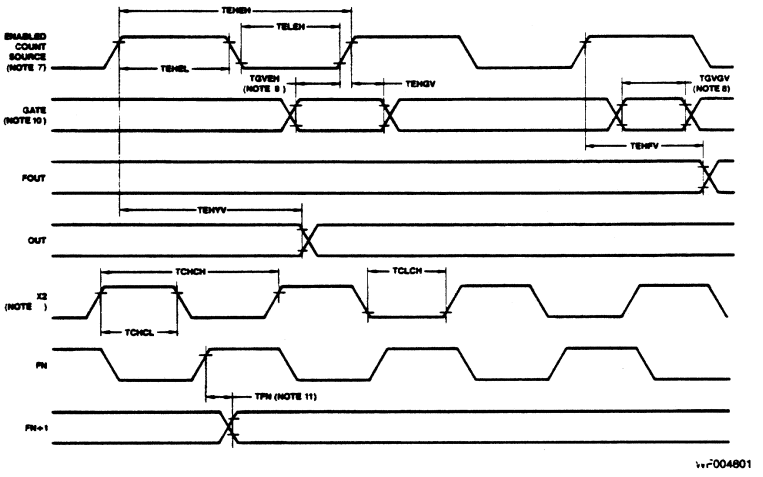


Figure 22. Counter Switching Waveforms

APPENDIX A

Design Hints

1) When a crystal is not being used, X1 and X2 should be connected as shown for TTL input (Figure A1) and no input (Figure A2).

2) Recommended oscillator capacitor values are 18 pF on X1 and X2.

3) Unused inputs should be tied to VCC.

4) The TC output can glitch when the counter is loaded. For this reason this output should not be connected to edge sensitive interrupts. The counter output should be set or cleared after the LOAD command.

5. The two most significant bits of the status register are not specified. They may be zero or one.

6. The mode register should not be modified when the counter is armed.

7. The LOAD and HOLD registers should not be changed during TC.

8. When using the different clocks for different counters be aware that there is a 75 ns skew between F1, F2, F3, F4 and F5.

9. The TC output will remain inactive if programmed to be in the TC TOGGLE mode and the step command is used to increment or decrement the counter. The output will go into TC if programmed to be in the active High or active Low terminal count modes. The only two ways out of TC in this case are:

   —Arming the counter and having an active source connected to it.

   —Issuing another step command.

10) Timing parameters TEHWH and TGVWH are specified as negative. The diagrams in Figure A3 show the relationship between these signals.

11) In mode X the counter will count all qualified source edges until the second (not the first) TC and then stop.

12) A TC can occur when the counters are loaded if the counter was stopped at $FFFF_H$ or 9999_{10} in the count up mode or at count 0001 when counting down. This is because an internal TC is generated which forces TC to be generated on the next count pulse.

13) In modes that alternate the reload source between the load and the hold registers (e.g., mode J), if the counter is disarmed at 0001_H for down counting or 9999_{10} for BCD up counting or $FFFF_H$ for binary up counting and rearmed, the reload source after the first TC will be the load register instead of the hold register. To avoid this, issue a software "dummy" load to the counter immediately after the disarm command.

14) In the down counting mode of the Am9513A, if a 0001 is loaded into the counter and another LOAD COUNTER command is issued, the TC of that counter will go active. If the load register contents are subsequently changed, and the counter armed, the first clock edge will cause the new load register contents to transfer into the counter and the next clock edge will decrement the counter and make it go out of TC.

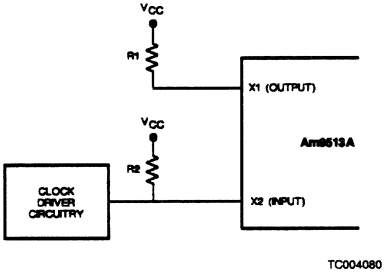
15) Glitches on $\overline{CS}$ just before the $\overline{RD}$ or $\overline{WR}$ pulse may cause the part to behave incorrectly.

16) Timing parameters TGVEH & TEHGV must not be violated; Figure A4 shows a method.

Troubleshooting (Symptom: Solution)

1) Registers not being programmed correctly: Check READ or WRITE recovery time.

2) Setup and hold problems observed in synchronous systems: Try switching from positive edge to negative edge triggering.



R1 = 6.8 kΩ ±10%
R2 is a function of Driver Circuitry to meet
X2 VIH = 3.8 V
X2 VIL = 0.8 V

Figure A1. Crystal Input Configuration

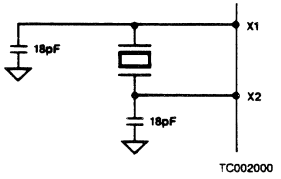


Figure A2. Crystal Input Configuration

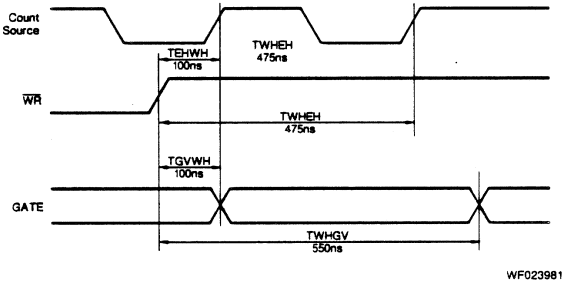


Figure A3. TEHWH/TGVWH Timing Diagram

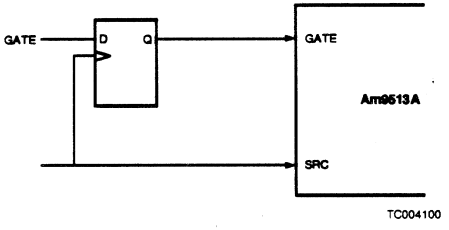


Figure A4. GATE/SRC Configuration Suggestion

D

Motorola MC6821Data Sheet*

This appendix contains the manufacturer data sheet for the Motorola MC6821 integrated circuit (Motorola, Inc.). This circuit is used on the PC-TIO-10 board.

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC6821

Peripheral Interface Adapter (PIA)

The MC6821 Peripheral Interface Adapter provides the universal means of interfacing peripheral equipment to the M6800 Family of microprocessors. This device is capable of interfacing the MPU to peripherals through two 8-bit bidirectional peripheral data buses and four control lines. No external logic is required for interfacing to most peripheral devices.

The functional configuration of the PIA is programmed by the MPU during system initialization. Each of the peripheral data lines can be programmed to act as an input or output, and each of the four control interrupt lines may be programmed for one of several control modes. This allows a high degree of flexibility in the overall operation of the interface.

- 8-Bit Bidirectional Data Bus for Communication with the MPU
- Two Bidirectional 8-Bit Buses for Interface to Peripherals
- Two Programmable Control Registers
- Two Programmable Data Direction Registers
- Four Individually-Controlled Interrupt Input Lines; Two Usable as Peripheral Control Outputs
- Handshake Control Logic for Input and Output Peripheral Operation
- High-Impedance Three-State and Direct Transistor Drive Peripheral Lines
- Program Controlled Interrupt and Interrupt Disable Capability
- CMOS Drive Capability on Side A Peripheral Lines
- Two TTL Drive Capability on All A and B Side Buffers
- TTL-Compatible
- Static Operation

3

This document contains information on a new product. Specifications and information herein are subject to change without notice.

MAXIMUM RATINGS

Characteristics	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6821, MC68A21, MC68B21 MC6821C, MC68A21C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Cerdip	θ_{JA}	 100 60	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{CC}$.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{PORT}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} < P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
BUS CONTROL INPUTS (R/$\overline{W}$, Enable, $\overline{RESET}$, RS0, RS1, CS0, CS1, CS2)					
Input High Voltage	V_{IH}	V_{SS}+2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}−0.3	–	V_{SS}+0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	I_{in}	–	1.0	2.5	μA
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{in}	–	–	7.5	pF
INTERRUPT OUTPUTS ($\overline{IRQA}$, $\overline{IRQB}$)					
Output Low Voltage (I_{Load} = 3.2 mA)	V_{OL}	–	–	V_{SS}+0.4	V
Hi-Z Output Leakage Current	I_{OZ}	–	1.0	10	μA
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{out}	–	–	5.0	pF
DATA BUS (D0-D7)					
Input High Voltage	V_{IH}	V_{SS}+2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}−0.3	–	V_{SS}+0.8	V
Hi-Z Input Leakage Current (V_{in} = 0.4 to 2.4 V)	I_{IZ}	–	2.0	10	μA
Output High Voltage (I_{Load} = −205 μA)	V_{OH}	V_{SS}+2.4	–	–	V
Output Low Voltage (I_{Load} = 1.6 mA)	V_{OL}	–	–	V_{SS}+0.4	V
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{in}	–	–	12.5	pF

MC6821

DC ELECTRICAL CHARACTERISTICS (Continued)

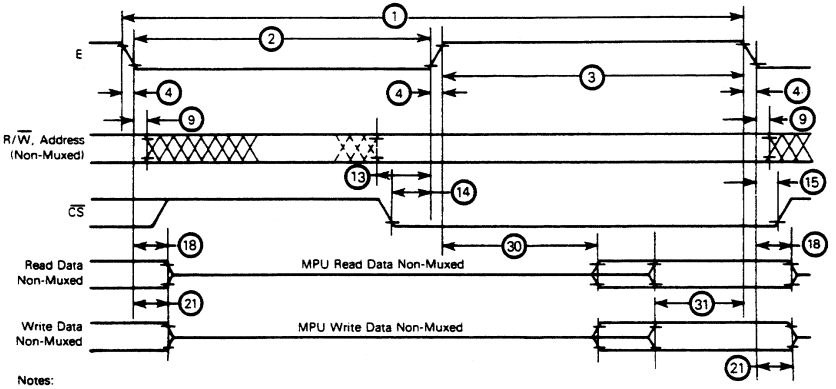
Characteristic		Symbol	Min	Typ	Max	Unit
PERIPHERAL BUS (PA0-PA7, PB0-PB7, CA1, CA2, CB1, CB2)						
Input Leakage Current (V_{in} = 0 to 5.25 V)	R/$\overline{W}$, $\overline{RESET}$, RS0, RS1, CS0, CS1, $\overline{CS2}$, CA1, CB1, Enable	I_{in}	–	1.0	2.5	μA
Hi-Z Input Leakage Current (V_{in} = 0.4 to 2.4 V)	PB0-PB7, CB2	I_{IZ}	–	2.0	10	μA
Input High Current (V_{IH} = 2.4 V)	PA0-PA7, CA2	I_{IH}	– 200	– 400	–	μA
Darlington Drive Current (V_O = 1.5 V)	PB0-PB7, CB2	I_{OH}	– 1.0	–	– 10	mA
Input Low Current (V_{IL} = 0.4 V)	PA0-PA7, CA2	I_{IL}	–	– 1.3	– 2.4	mA
Output High Voltage (I_{Load} = – 200 μA)	PA0-PA7, PB0-PB7, CA2, CB2	V_{OH}	V_{SS} + 2.4	–	–	V
(I_{Load} = – 10 μA)	PA0-PA7, CA2		V_{CC} – 1.0	–	–	
Output Low Voltage (I_{Load} = 3.2 mA)		V_{OL}	–	–	V_{SS} + 0.4	V
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)		C_{in}	–	–	10	pF
POWER REQUIREMENTS						
Internal Power Dissipation (Measured at T_L = 0°C)		P_{INT}	–	–	550	mW

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristic	Symbol	MC6821		MC68A21		MC68B21		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	–	280	–	210	–	ns
3	Pulse Width, E High	PW_{EH}	450	–	280	–	220	–	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance).

FIGURE 1 — BUS TIMING



Notes:
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

PERIPHERAL TIMING CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0 V, $T_A = T_L$ to T_H, unless otherwise specified)

Characteristic	Symbol	MC6821		MC68A21		MC68B21		Unit	Reference Fig. No.
		Min	Max	Min	Max	Min	Max		
Data Setup Time	t_{PDS}	200	—	135	—	100	—	ns	6
Data Hold Time	t_{PDH}	0	—	0	—	0	—	ns	6
Delay Time, Enable Negative Transition to CA2 Negative Transition	t_{CA2}	—	1.0	—	0.670	—	0.500	µs	3, 7, 8
Delay Time, Enable Negative Transition t CA2 Positive Transition	t_{RS1}	—	1.0	—	0.670	—	0.500	µs	3, 7
Rise and Fall Times for CA1 and CA2 Input Signals	t_r, t_f	—	1.0	—	1.0	—	1.0	µs	8
Delay Time from CA1 Active Transition to CA2 Positive Transition	t_{RS2}	—	2.0	—	1.35	—	1.0	µs	3, 8
Delay Time, Enable Negative Transition to Data Valid	t_{PDW}	—	1.0	—	0.670	—	0.5	µs	3, 9, 10
Delay Time, Enable Negative Transition to CMOS Data Valid PA0-PA7, CA2	t_{CMOS}	—	2.0	—	1.35	—	1.0	µs	4, 9
Delay Time, Enable Positive Transition to CB2 Negative Transition	t_{CB2}	—	1.0	—	0.670	—	0.5	µs	3, 11, 12
Delay Time, Data Valid to CB2 Negative Transition	t_{DD}	20	—	20	—	20	—	ns	3, 10
Delay Time, Enable Positive Transition to CB2 Positive Transition	t_{RS1}	—	1.0	—	0.670	—	0.5	µs	3, 11
Control Output Pulse Width, CA2/CB2	PW_{CT}	500	—	375	—	250	—	ns	3, 7, 11
Rise and Fall Time for CB1 and CB2 Input Signals	t_r, t_f	—	1.0	—	1.0	—	1.0	µs	12
Delay Time, CB1 Active Transition to CB2 Positive Transition	t_{RS2}	—	2.0	—	1.35	—	1.0	µs	3, 12
Interrupt Release Time, $\overline{IRQA}$ and $\overline{IRQB}$	t_{IR}	—	1.60	—	1.10	—	0.85	µs	5, 14
Interrupt Response Time	t_{RS3}	—	1.0	—	1.0	—	1.0	µs	5, 13
Interrupt Input Pulse Time	PW_I	500	—	500	—	500	—	ns	13
$\overline{RESET}$ Low Time*	t_{RL}	1.0	—	0.66	—	0.5	—	µs	15

3

FIGURE 2 — BUS TIMING TEST LOADS

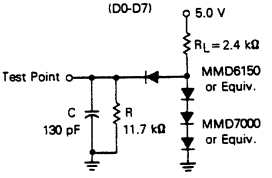


FIGURE 3 — TTL EQUIVALENT TEST LOAD

(PA0—PA7, PB0—PB7, CA2, CB2)

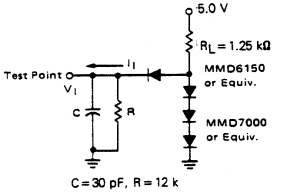


FIGURE 4 — CMOS EQUIVALENT TEST LOAD

(PA0-PA7, CA2)

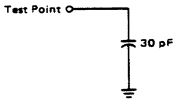


FIGURE 5 — NMOS EQUIVALENT TEST LOAD

($\overline{IRQ}$ Only)

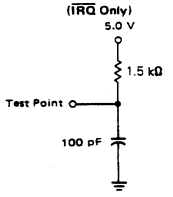


FIGURE 6 — PERIPHERAL DATA SETUP AND HOLD TIMES
(Read Mode)

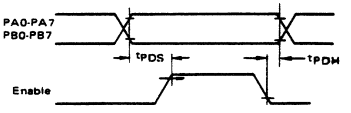
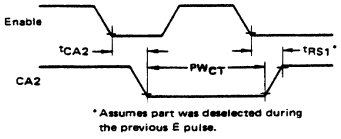


FIGURE 7 — CA2 DELAY TIME
(Read Mode; CRA-5 = CRA3 = 1, CRA-4 = 0)



*Assumes part was deselected during the previous E pulse.

FIGURE 8 — CA2 DELAY TIME
(Read Mode; CRA-5 = 1, CRA-3 = CRA-4 = 0)

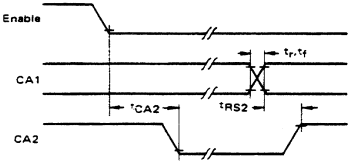


FIGURE 9 — PERIPHERAL CMOS DATA DELAY TIMES
(Write Mode; CRA-5 = CRA-3 = 1, CRA-4 = 0)

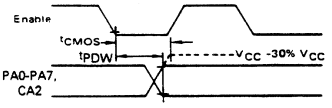
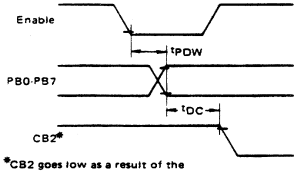
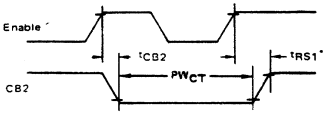


FIGURE 10 — PERIPHERAL DATA AND CB2 DELAY TIMES
(Write Mode; CRB-5 = CRB-3 = 1, CRB-4 = 0)



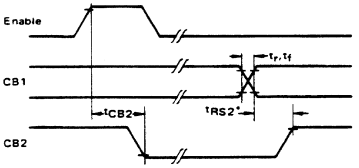
*CB2 goes low as a result of the positive transition of Enable.

FIGURE 11 — CB2 DELAY TIME
(Write Mode; CRB-5 = CRB-3 = 1, CRB-4 = 0)



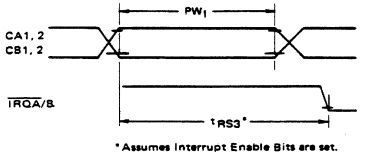
*Assumes part was deselected during the previous E pulse.

FIGURE 12 — CB2 DELAY TIME
(Write Mode; CRB-5 = 1, CRB-3 = CRB-4 = 0)



*Assumes part was deselected during any previous E pulse.

FIGURE 13 — INTERRUPT PULSE WIDTH AND $\overline{IRQ}$ RESPONSE



*Assumes Interrupt Enable Bits are set.

Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

MC6821

FIGURE 14 — $\overline{IRQ}$ RELEASE TIME

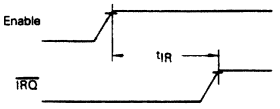
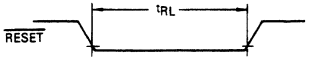


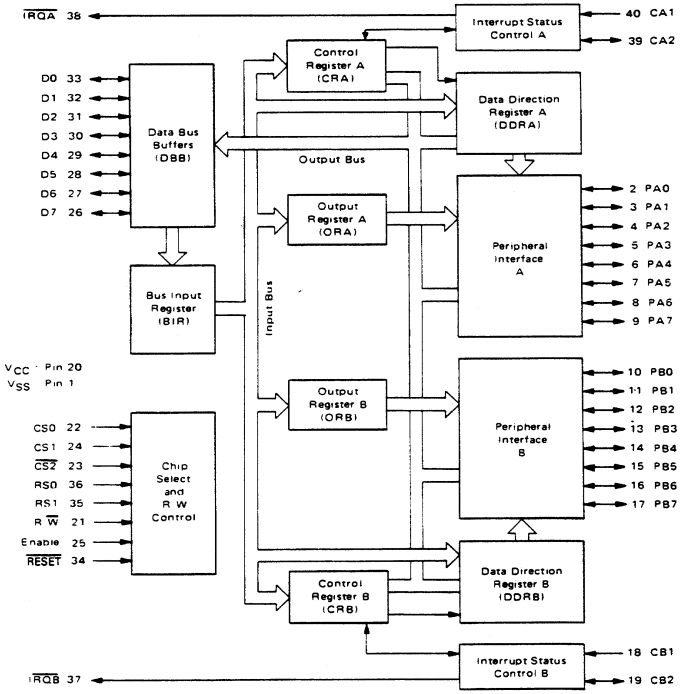
FIGURE 15 — $\overline{RESET}$ LOW TIME



*The $\overline{RESET}$ line must be a V_{IH} for a minimum of 1.0 μs before addressing the PIA.

Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

FIGURE 16 — EXPANDED BLOCK DIAGRAM



MC6821

PIA INTERFACE SIGNALS FOR MPU

The PIA interfaces to the M6800 bus with an 8-bit bidirectional data bus, three chip select lines, two register select lines, two interrupt request lines, a read/write line, an enable line and a reset line. To ensure proper operation with the MC6800, MC6802, or MC6808 microprocessors, VMA should be used as an active part of the address decoding.

Bidirectional Data (D0-D7) — The bidirectional data lines (D0-D7) allow the transfer of data between the MPU and the PIA. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs a PIA read operation. The read/write line is in the read (high) state when the PIA is selected for a read operation.

Enable (E) — The enable pulse, E, is the only timing signal that is supplied to the PIA. Timing of all other signals is referenced to the leading and trailing edges of the E pulse.

3

Read/Write (R/$\overline{W}$) — This signal is generated by the MPU to control the direction of data transfers on the data bus. A low state on the PIA read/write line enables the input buffers and data is transferred from the MPU to the PIA on the E signal if the device has been selected. A high on the read/write line sets up the PIA for a transfer of data to the bus. The PIA output buffers are enabled when the proper address and the enable pulse E are present.

$\overline{RESET}$ — The active low $\overline{RESET}$ line is used to reset all register bits in the PIA to a logical zero (low). This line can be used as a power-on reset and as a master reset during system operation.

Chip Selects (CS0, CS1, and $\overline{CS2}$) — These three input signals are used to select the PIA. CS0 and CS1 must be high and $\overline{CS2}$ must be low for selection of the device. Data transfers are then performed under the control of the enable and read/write signals. The chip select lines must be stable for the duration of the E pulse. The device is deselected when any of the chip selects are in the inactive state.

Register Selects (RS0 and RS1) — The two register select lines are used to select the various registers inside the PIA. These two lines are used in conjunction with internal Control Registers to select a particular register that is to be written or read.

The register and chip select lines should be stable for the duration of the E pulse while in the read or write cycle.

Interrupt Request ($\overline{IRQA}$ and $\overline{IRQB}$) — The active low Interrupt Request lines ($\overline{IRQA}$ and $\overline{IRQB}$) act to interrupt the MPU either directly or through interrupt priority circuitry. These lines are "open drain" (no load device on the chip). This permits all interrupt request lines to be tied together in a wire-OR configuration.

Each Interrupt Request line has two internal interrupt flag bits that can cause the Interrupt Request line to go low. Each flag bit is associated with a particular peripheral interrupt line. Also, four interrupt enable bits are provided in the PIA which may be used to inhibit a particular interrupt from a peripheral device.

Servicing an interrupt by the MPU may be accomplished by a software routine that, on a prioritized basis, sequentially reads and tests the two control registers in each PIA for interrupt flag bits that are set.

The interrupt flags are cleared (zeroed) as a result of an MPU Read Peripheral Data Operation of the corresponding data register. After being cleared, the interrupt flag bit cannot be enabled to be set until the PIA is deselected during an E pulse. The E pulse is used to condition the interrupt control lines (CA1, CA2, CB1, CB2). When these lines are used as interrupt inputs, at least one E pulse must occur from the inactive edge to the active edge of the interrupt input signal to condition the edge sense network. If the interrupt flag has been enabled and the edge sense circuit has been properly conditioned, the interrupt flag will be set on the next active transition of the interrupt input pin.

PIA PERIPHERAL INTERFACE LINES

The PIA provides two 8-bit bidirectional data buses and four interrupt/control lines for interfacing to peripheral devices.

Section A Peripheral Data (PA0-PA7) — Each of the peripheral data lines can be programmed to act as an input or output. This is accomplished by setting a "1" in the corresponding Data Direction Register bit for those lines which are to be outputs. A "0" in a bit of the Data Direction Register causes the corresponding peripheral data line to act as an input. During an MPU Read Peripheral Data Operation, the data on peripheral lines programmed to act as inputs appears directly on the corresponding MPU Data Bus lines. In the input mode, the internal pullup resistor on these lines represents a maximum of 1.5 standard TTL loads.

The data in Output Register A will appear on the data lines that are programmed to be outputs. A logical "1" written into the register will cause a "high" on the corresponding data line while a "0" results in a "low." Data in Output Register A may be read by an MPU "Read Peripheral Data A" operation when the corresponding lines are programmed as outputs. This data will be read properly if the voltage on the peripheral data lines is greater than 2.0 volts for a logic "1" output and less than 0.8 volt for a logic "0" output. Loading the output lines such that the voltage on these lines does not reach full voltage causes the data transferred into the MPU on a Read operation to differ from that contained in the respective bit of Output Register A.

Section B Peripheral Data (PB0-PB7) — The peripheral data lines in the B Section of the PIA can be programmed to act as either inputs or outputs in a similar manner to PA0-PA7. They have three-state capability, allowing them to enter a high-impedance state when the peripheral data line is used as an input. In addition, data on the peripheral data lines

PB0-PB7 will be read properly from those lines programmed as outputs even if the voltages are below 2.0 volts for a "high" or above 0.8 V for a "low". As outputs, these lines are compatible with standard TTL and may also be used as a source of at least 1 milliampere at 1.5 volts to directly drive the base of a transistor switch.

Interrupt Input (CA1 and CB1) – Peripheral input lines CA1 and CB1 are input only lines that set the interrupt flags of the control registers. The active transition for these signals is also programmed by the two control registers.

Peripheral Control (CA2) – The peripheral control line CA2 can be programmed to act as an interrupt input or as a peripheral control output. As an output, this line is compatible with standard TTL; as an input the internal pullup resistor on this line represents 1.5 standard TTL loads. The function of this signal line is programmed with Control Register A.

Peripheral Control (CB2) – Peripheral Control line CB2 may also be programmed to act as an interrupt input or peripheral control output. As an input, this line has high input impedance and is compatible with standard TTL. As an output it is compatible with standard TTL and may also be used as a source of up to 1 milliampere at 1.5 volts to directly drive the base of a transistor switch. This line is programmed by Control Register B.

INTERNAL CONTROLS

INITIALIZATION

A $\overline{\text{RESET}}$ has the effect of zeroing all PIA registers. This will set PA0-PA7, PB0-PB7, CA2 and CB2 as inputs, and all interrupts disabled. The PIA must be configured during the restart program which follows the reset.

There are six locations within the PIA accessible to the MPU data bus: two Peripheral Registers, two Data Direction Registers, and two Control Registers. Selection of these locations is controlled by the RS0 and RS1 inputs together with bit 2 in the Control Register, as shown in Table 1.

Details of possible configurations of the Data Direction and Control Register are as follows:

TABLE 1 – INTERNAL ADDRESSING

		Control Register Bit		
RS1	RS0	CRA-2	CRB-2	Location Selected
0	0	1	X	Peripheral Register A
0	0	0	X	Data Direction Register A
0	1	X	X	Control Register A
1	0	X	1	Peripheral Register B
1	0	X	0	Data Direction Register B
1	1	X	X	Control Register B

X = Don't Care

PORT A-B HARDWARE CHARACTERISTICS

As shown in Figure 17, the MC6821 has a pair of I/O ports whose characteristics differ greatly. The A side is designed to drive CMOS logic to normal 30% to 70% levels, and incorporates an internal pullup device that remains connected even in the input mode. Because of this, the A side requires more drive current in the input mode than Port B. In contrast, the B side uses a normal three-state NMOS buffer which cannot pullup to CMOS levels without external resistors. The B side can drive extra loads such as Darlingtons without problem. When the PIA comes out of reset, the A port represents inputs with pullup resistors, whereas the B side (input mode also) will float high or low, depending upon the load connected to it.

Notice the differences between a Port A and Port B read operation when in the output mode. When reading Port A, the actual pin is read, whereas the B side read comes from an output latch, ahead of the actual pin.

CONTROL REGISTERS (CRA and CRB)

The two Control Registers (CRA and CRB) allow the MPU to control the operation of the four peripheral control lines CA1, CA2, CB1, and CB2. In addition they allow the MPU to enable the interrupt lines and monitor the status of the interrupt flags. Bits 0 through 5 of the two registers may be written or read by the MPU when the proper chip select and register select signals are applied. Bits 6 and 7 of the two registers are read only and are modified by external interrupts occurring on control lines CA1, CA2, CB1, or CB2. The format of the control words is shown in Figure 18.

DATA DIRECTION ACCESS CONTROL BIT (CRA-2 and CRB-2)

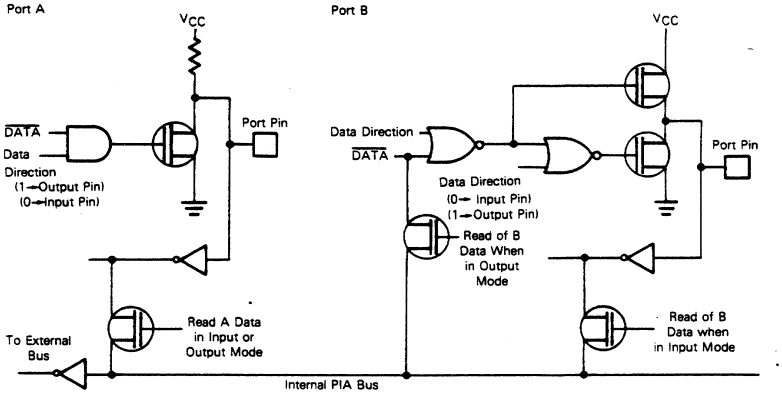
Bit 2, in each Control Register (CRA and CRB), determines selection of either a Peripheral Output Register or the corresponding Data Direction E Register when the proper register select signals are applied to RS0 and RS1. A "1" in bit 2 allows access of the Peripheral Interface Register, while a "0" causes the Data Direction Register to be addressed.

Interrupt Flags (CRA-6, CRA-7, CRB-6, and CRB-7) – The four interrupt flag bits are set by active transitions of signals on the four Interrupt and Peripheral Control lines when those lines are programmed to be inputs. These bits cannot be set directly from the MPU Data Bus and are reset indirectly by a Read Peripheral Data Operation on the appropriate section.

Control of CA2 and CB2 Peripheral Control Lines (CRA-3, CRA-4, CRA-5, CRB-3, CRB-4, and CRB-5) – Bits 3, 4, and 5 of the two control registers are used to control the CA2 and CB2 Peripheral Control lines. These bits determine if the control lines will be an interrupt input or an output control signal. If bit CRA-5 (CRB-5) is low, CA2 (CB2) is an interrupt input line similar to CA1 (CB1). When CRA-5 (CRB-5) is high, CA2 (CB2) becomes an output signal that may be used to control peripheral data transfers. When in the output mode, CA2 and CB2 have slightly different loading characteristics.

Control of CA1 and CB1 Interrupt Input Lines (CRA-0, CRB-0, CRA-1, and CRB-1) — The two lowest-order bits of the control registers are used to control the interrupt input lines CA1 and CB1. Bits CRA-0 and CRB-0 are used to enable the MPU interrupt signals $\overline{IRQA}$ and $\overline{IRQB}$, respectively. Bits CRA-1 and CRB-1 determine the active transition of the interrupt input signals CA1 and CB1.

FIGURE 17 — PORT A AND PORT B EQUIVALENT CIRCUITS



ORDERING INFORMATION

MC68A21CP

- Motorola Integrated Circuit
- M6800 Family
- Blanks = 1.0 MHz
- A = 1.5 MHz
- B = 2.0 MHz
- Device Designation In M6800 Family
- Temperature Range
  - Blank = 0° — + 70°C
  - C = – 40° — + 85°C
- Package
  - P = Plastic
  - S = Cerdip
  - L = Ceramic

BETTER PROGRAM

Better program processing is available on all types listed. Add suffix letters to part number.

Level 1 add "S" Level 2 add "D" Level 3 add "DS"

Level 1 "S" = 10 Temp Cycles — (– 25 to 150°C); Hi Temp testing at T_A max.
Level 2 "D" = 168 Hour Burn-in at 125°C
Level 3 "DS" = Combination of Level 1 and 2.

MC6821

Determine Active CA1 (CB1) Transition for Setting Interrupt Flag IRQA(B)1 — (bit 7)

b1 = 0: IRQA(B)1 set by high-to-low transition on CA1 (CB1)

b1 = 1: IRQA(B)1 set by low-to-high transition on CA1 (CB1).

IRQA(B) 1 Interrupt Flag (bit 7)

Goes high on active transition of CA1 (CB1); Automatically cleared by MPU Read of Output Register A(B). May also be cleared by hardware Reset.

FIGURE 18 — CONTROL WORD FORMAT

CA1 (CB1) Interrupt Request Enable/Disable

b0 = 0: Disables IRQA(B) MPU Interrupt by CA1 (CB1) active transition.[1]

b0 = 1: Enable IRQA(B) MPU Interrupt by CA1 (CB1) active transition.

1. IRQA(B) will occur on next (MPU generated) positive transition of b0 if CA1 (CB1) active transition occurred while interrupt was disabled.

Control Register	b7	b6	b5	b4	b3	b2	b1	b0
	IRQA(B)1 Flag	IRQA(B)2 Flag	CA2 (CB2) Control			DDR Access	CA1 (CB1) Control	

IRQA(B)2 Interrupt Flag (bit 6)

When CA2 (CB2) is an input, IRQA(B) goes high on active transition CA2 (CB2); Automatically cleared by MPU Read of Output Register A(B). May also be cleared by hardware Reset.

CA2 (CB2) Established as Output (b5 = 1): IRQA(B) 2 = 0, not affected by CA2 (CB2) transitions.

Determines Whether Data Direction Register Or Output Register is Addressed

b2 = 0: Data Direction Register selected.

b2 = 1: Output Register selected.

CA2 (CB2) Established as Output by b5 = 1

(Note that operation of CA2 and CB2 output functions are not identical)

b5 b4 b3

1 0 → CA2

b3 = 0: **Read Strobe with CA1 Restore**
CA2 goes low on first high-to-low E transition following an MPU read of Output Register A; returned high by next active CA1 transition, as specified by bit 1.

b3 = 1: **Read Strobe with E Restore**
CA2 goes low on first high-to-low E transition following an MPU read of Output Register A; returned high by next high-to-low E transition during a deselect.

→ CB2

b3 = 0: **Write Strobe with CB1 Restore**
CB2 goes low on first low-to-high E transition following an MPU write into Output Register B; returned high by the next active CB1 transition as specified by bit 1. CRB-b7 must first be cleared by a read of data.

b3 = 1: **Write Strobe with E Restore**
CB2 goes low on first low-to-high E transition following an MPU write into Output Register B; returned high by the next low-to-high E transition following an E pulse which occurred while the part was deselected.

b5 b4 b3

1 1 → **Set/Reset CA2 (CB2)**

CA2 (CB2) goes low as MPU writes b3 = 0 into Control Register.

CA2 (CB2) goes high as MPU writes b3 = 1 into Control Register.

CA2 (CB2) Established as Input by b5 = 0

b5 b4 b3

0 → **CA2 (CB2) Interrupt Request Enable/Disable**

b3 = 0: Disables IRQA(B) MPU Interrupt by CA2 (CB2) active transition.*

b3 = 1: Enables IRQA(B) MPU Interrupt by CA2 (CB2) active transition.

*IRQA(B) will occur on next (MPU generated) positive transition of b3 if CA2 (CB2) active transition occurred while interrupt was disabled.

→ **Determines Active CA2 (CB2) Transition for Setting Interrupt Flag IRQA(B)2 — (Bit b6)**

b4 = 0: IRQA(B)2 set by high-to-low transition on CA2 (CB2).

b4 = 1: IRQA(B)2 set by low-to-high transition on CA2 (CB2).

3

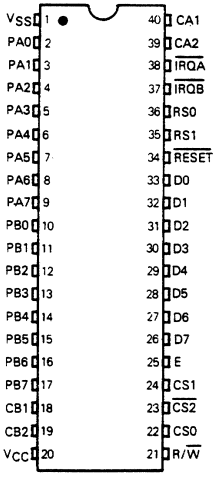
MC6821

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Cerdip	1.0	0°C to 70°C	MC6821S
S Suffix	1.0	−40°C to 85°C	MC6821CS
	1.5	0°C to 70°C	MC68A21S
	1.5	−40°C to 85°C	MC68A21CS
	2.0	0°C to 70°C	MC68B21S
Plastic	1.0	0°C to 70°C	MC6821P
P Suffix	1.0	−40°C to 85°C	MC6821CP
	1.5	0°C to 70°C	MC68A21P
	1.5	−40°C to 85°C	MC68A21CP
	2.0	0°C to 70°C	MC68B21P

PIN ASSIGNMENT

3



E

Switch Settings

Table E-1 lists the possible switch settings, the corresponding base I/O address, and the base I/O address space used for that setting.

Note I/O address space hex 000 through hex 0FF is not listed in Table E-1 since it is reserved for system use.

Table E-1. Switch Settings with Corresponding Base I/O Address and Base I/O Address Space

Switch Setting							Base I/O Address (hex)	Base I/O Address Space Used (hex)
A9	A8	A7	A6	A5	A4	A3		
0	1	0	0	0	0	0	100	100 - 107
0	1	0	0	0	0	1	108	108 - 10F
0	1	0	0	0	1	0	110	110 - 117
0	1	0	0	0	1	1	118	118 - 11F
0	1	0	0	1	0	0	120	120 - 127
0	1	0	0	1	0	1	128	128 - 12F
0	1	0	0	1	1	0	130	130 - 137
0	1	0	0	1	1	1	138	138 - 13F
0	1	0	1	0	0	0	140	140 - 147
0	1	0	1	0	0	1	148	148 - 14F
0	1	0	1	0	1	0	150	150 - 157
0	1	0	1	0	1	1	158	158 - 15F
0	1	0	1	1	0	0	160	160 - 167
0	1	0	1	1	0	1	168	168 - 16F
0	1	0	1	1	1	0	170	170 - 177

Table E-1. Switch Settings with Corresponding Base I/O Address and Base I/O Address Space (Continued)

Switch Setting							Base I/O Address (hex)	Base I/O Address Space Used (hex)
A9	A8	A7	A6	A5	A4	A3		
0	1	0	1	1	1	1	178	178 - 17F
0	1	1	0	0	0	0	180	180 - 187
0	1	1	0	0	0	1	188	188 - 18F
0	1	1	0	0	1	0	190	190 - 197
0	1	1	0	0	1	1	198	198 - 19F
0	1	1	0	1	0	0	1A0	1A0 - 1A7
0	1	1	0	1	0	1	1A8	1A8 - 1AF
0	1	1	0	1	1	0	1B0	1B0 - 1B7
0	1	1	0	1	1	1	1B8	1B8 - 1BF
0	1	1	1	0	0	0	1C0	1C0 - 1C7
0	1	1	1	0	0	1	1C8	1C8 - 1CF
0	1	1	1	0	1	0	1D0	1D0 - 1D7
0	1	1	1	0	1	1	1D8	1D8 - 1DF
0	1	1	1	1	0	0	1E0	1E0 - 1E7
0	1	1	1	1	0	1	1E8	1E8 - 1EF
0	1	1	1	1	1	0	1F0	1F0 - 1F7
0	1	1	1	1	1	1	1F8	1F8 - 1FF
1	0	0	0	0	0	0	200	200 - 207
1	0	0	0	0	0	1	208	208 - 20F
1	0	0	0	0	1	0	210	210 - 217
1	0	0	0	0	1	1	218	218 - 21F
1	0	0	0	1	0	0	220	220 - 227
1	0	0	0	1	0	1	228	228 - 22F
1	0	0	0	1	1	0	230	230 - 237

Table E-1. Switch Settings with Corresponding Base I/O Address and Base I/O Address Space (Continued)

Switch Setting							Base I/O Address (hex)	Base I/O Address Space Used (hex)
A9	A8	A7	A6	A5	A4	A3		
1	0	0	0	1	1	1	238	238 - 23F
1	0	0	1	0	0	0	240	240 - 247
1	0	0	1	0	0	1	248	248 - 24F
1	0	0	1	0	1	0	250	250 - 257
1	0	0	1	0	1	1	258	258 - 25F
1	0	0	1	1	0	0	260	260 - 267
1	0	0	1	1	0	1	268	268 - 26F
1	0	0	1	1	1	0	270	270 - 277
1	0	0	1	1	1	1	278	278 - 27F
1	0	1	0	0	0	0	280	280 - 287
1	0	1	0	0	0	1	288	288 - 28F
1	0	1	0	0	1	0	290	290 - 297
1	0	1	0	0	1	1	298	298 - 29F
1	0	1	0	1	0	0	2A0	2A0 - 2A7
1	0	1	0	1	0	1	2A8	2A8 - 2AF
1	0	1	0	1	1	0	2B0	2B0 - 2B7
1	0	1	0	1	1	1	2B8	2B8 - 2BF
1	0	1	1	0	0	0	2C0	2C0 - 2C7
1	0	1	1	0	0	1	2C8	2C8 - 2CF
1	0	1	1	0	1	0	2D0	2D0 - 2D7
1	0	1	1	0	1	1	2D8	2D8 - 2DF
1	0	1	1	1	0	0	2E0	2E0 - 2E7
1	0	1	1	1	0	1	2E8	2E8 - 2EF
1	0	1	1	1	1	0	2F0	2F0 - 2F7

Table E-1. Switch Settings with Corresponding Base I/O Address and Base I/O Address Space (Continued)

Switch Setting							Base I/O Address (hex)	Base I/O Address Space Used (hex)
A9	A8	A7	A6	A5	A4	A3		
1	0	1	1	1	1	1	2F8	2F8 - 2FF
1	1	0	0	0	0	0	300	300 - 307
1	1	0	0	0	0	1	308	308 - 30F
1	1	0	0	0	1	0	310	310 - 317
1	1	0	0	0	1	1	318	318 - 31F
1	1	0	0	1	0	0	320	320 - 327
1	1	0	0	1	0	1	328	328 - 32F
1	1	0	0	1	1	0	330	330 - 337
1	1	0	0	1	1	1	338	338 - 33F
1	1	0	1	0	0	0	340	340 - 347
1	1	0	1	0	0	1	348	348 - 34F
1	1	0	1	0	1	0	350	350 - 357
1	1	0	1	0	1	1	358	358 - 35F
1	1	0	1	1	0	0	360	360 - 367
1	1	0	1	1	0	1	368	368 - 36F
1	1	0	1	1	1	0	370	370 - 377
1	1	0	1	1	1	1	378	378 - 37F
1	1	1	0	0	0	0	380	380 - 387
1	1	1	0	0	0	1	388	388 - 38F
1	1	1	0	0	1	0	390	390 - 397
1	1	1	0	0	1	1	398	398 - 39F
1	1	1	0	1	0	0	3A0	3A0 - 3A7
1	1	1	0	1	0	1	3A8	3A8 - 3AF
1	1	1	0	1	1	0	3B0	3B0 - 3B7

Table E-1. Switch Settings with Corresponding Base I/O Address and Base I/O Address Space (Continued)

Switch Setting							Base I/O Address (hex)	Base I/O Address Space Used (hex)
A9	A8	A7	A6	A5	A4	A3		
1	1	1	0	1	1	1	3B8	3B8 - 3BF
1	1	1	1	0	0	0	3C0	3C0 - 3C7
1	1	1	1	0	0	1	3C8	3C8 - 3CF
1	1	1	1	0	1	0	3D0	3D0 - 3D7
1	1	1	1	0	1	1	3D8	3D8 - 3DF
1	1	1	1	1	0	0	3E0	3E0 - 3E7
1	1	1	1	1	0	1	3E8	3E8 - 3EF
1	1	1	1	1	1	0	3F0	3F0 - 3F7
1	1	1	1	1	1	1	3F8	3F8 - 3FF

F

Technical Support Resources

This appendix describes the comprehensive resources available to you in the Technical Support section of the National Instruments Web site and provides technical support telephone numbers for you to use if you have trouble connecting to our Web site or if you do not have internet access.

NI Web Support

To provide you with immediate answers and solutions 24 hours a day, 365 days a year, National Instruments maintains extensive online technical support resources. They are available to you at no cost, are updated daily, and can be found in the Technical Support section of our Web site at `www.natinst.com/support`.

Online Problem-Solving and Diagnostic Resources

- **KnowledgeBase**—A searchable database containing thousands of frequently asked questions (FAQs) and their corresponding answers or solutions, including special sections devoted to our newest products. The database is updated daily in response to new customer experiences and feedback.
- **Troubleshooting Wizards**—Step-by-step guides lead you through common problems and answer questions about our entire product line. Wizards include screen shots that illustrate the steps being described and provide detailed information ranging from simple getting started instructions to advanced topics.
- **Product Manuals**—A comprehensive, searchable library of the latest editions of National Instruments hardware and software product manuals.
- **Hardware Reference Database**—A searchable database containing brief hardware descriptions, mechanical drawings, and helpful images of jumper settings and connector pinouts.
- **Application Notes**—A library with more than 100 short papers addressing specific topics such as creating and calling DLLs, developing your own instrument driver software, and porting applications between platforms and operating systems.

Software-Related Resources

- **Instrument Driver Network**—A library with hundreds of instrument drivers for control of standalone instruments via GPIB, VXI, or serial interfaces. You also can submit a request for a particular instrument driver if it does not already appear in the library.
- **Example Programs Database**—A database with numerous, non-shipping example programs for National Instruments programming environments. You can use them to complement the example programs that are already included with National Instruments products.
- **Software Library**—A library with updates and patches to application software, links to the latest versions of driver software for National Instruments hardware products, and utility routines.

Worldwide Support

National Instruments has offices located around the globe. Many branch offices maintain a Web site to provide information on local services. You can access these Web sites from `www.natinst.com/worldwide`.

If you have trouble connecting to our Web site, please contact your local National Instruments office or the source from which you purchased your National Instruments product(s) to obtain support.

For telephone support in the United States, dial 512 795 8248. For telephone support outside the United States, contact your local branch office:

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20,
Brazil 011 284 5011, Canada (Ontario) 905 785 0085,
Canada (Québec) 514 694 8521, China 0755 3904939,
Denmark 45 76 26 00, Finland 09 725 725 11, France 01 48 14 24 24,
Germany 089 741 31 30, Hong Kong 2645 3186, India 91805275406,
Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970,
Korea 02 596 7456, Mexico (D.F.) 5 280 7625,
Mexico (Monterrey) 8 357 7695, Netherlands 0348 433466,
Norway 32 27 73 00, Singapore 2265886, Spain (Madrid) 91 640 0085,
Spain (Barcelona) 93 582 0251, Sweden 08 587 895 00,
Switzerland 056 200 51 51, Taiwan 02 2377 1200,
United Kingdom 01635 523545

Glossary

Prefix	Meaning	Value
n-	nano-	10^{-9}
µ-	micro-	10^{-6}
m-	milli-	10^{-3}
k-	kilo-	10^{3}
M-	mega-	10^{6}

Numbers/Symbols

°	degrees
%	percent

A

A	amperes
A/D	analog-to-digital
ADC	analog-to-digital converter—an electronic device, often an integrated circuit, that converts an analog voltage to a digital number
ADC resolution	the resolution of the ADC, which is measured in bits. An ADC with 16 bits has a higher resolution, and thus a higher degree of accuracy, than a 12-bit ADC.
ALU	arithmetic logic unit—the element(s) in a processing system that perform(s) the mathematical functions such as addition, subtraction, multiplication, division, inversion, AND, OR, NAND, and NOR
AMD	Advanced Micro Devices
amplification	a type of signal conditioning that improves accuracy in the resulting digitized signal and reduces noise

amplitude flatness	a measure of how close to constant the gain of a circuit remains over a range of frequencies
antialiasing filter	a low-pass filter preceding an ADC (usually a brickwall filter) that rejects signal energy above the Nyquist frequency (1/2 the sample rate) of the ADC so that the ADC does not mistake out-of-band signals for in-band signals
anti-imaging filter	a low-pass filter after an DAC (usually a brickwall filter) that rejects signal energy above the Nyquist frequency (1/2 the sample rate) of the DAC in order to suppress out-of-band images of the in-band signal created by the D/A conversion process
attenuate	to decrease the amplitude of a signal
attenuation ratio	the factor by which a signal's amplitude is decreased
autozero	the technique of internally shorting the internal circuit while disconnecting the measurement to compensate for temperature effects
AWG	American Wire Gauge

B

BCD	binary-coded decimal
bipolar	a signal range that includes both positive and negative values (for example, –5 V to +5 V)
break-before-make	a type of switching contact that is completely disengaged from one terminal before it connects with another terminal
breakdown voltage	the voltage high enough to cause breakdown of optical isolation, semiconductors, or dielectric materials. *See also* working voltage.
brickwall filter	a low-pass filter having very flat passband, a very sudden, sharp transition region, and high rejection in the stopband
burden voltage	the voltage drop across the input section of the current mode
burst-mode	a high-speed data transfer in which the address of the data is sent followed by back-to-back data words while a physical signal is asserted

C

C	Celsius
CalDAC	calibration DAC
capacitively coupled	connecting a capacitor in a signal path to remove the DC content of the signal
channel clock	the clock controlling the time interval between individual channel sampling within a scan. Boards with simultaneous sampling do not have this clock.
chromatograph	an instrument used in chemical analysis of gases and liquids
CI	computing index
circuit trigger	a condition for starting or stopping clocks
clip	clipping occurs when an input signal exceeds the input range of the amplifier
CMRR	common-mode rejection ratio—a measure of an instrument's ability to reject interference from a common-mode signal, usually expressed in decibels (dB)
code width	the smallest detectable change in an input voltage of a DAQ device
cold-junction compensation	a method of compensating for inaccuracies in thermocouple circuits
common-mode range	the input range over which a circuit can handle a common-mode signal
common-mode signal	the mathematical average voltage, relative to the computer's ground, of the signals from a differential input
common-mode voltage	any voltage present at the instrumentation amplifier inputs with respect to amplifier ground
Compact PCI	refers to the core specification defined by the PCI Industrial Computer Manufacturer's Group (PICMG)
compensation range	the range of a parameter for which compensating adjustment can be made

conditional retrieval	a method of triggering in which you simulate an analog trigger using software. Also called software triggering.
conversion device	device that transforms a signal from one form to another. For example, analog-to-digital converters (ADCs) for analog input, digital-to-analog converters (DACs) for analog output, digital input or output ports, and counter/timers are conversion devices.
conversion time	the time required, in an analog input or output system, from the moment a channel is interrogated (such as with a read instruction) to the moment that accurate data is available

D

D/A	digital-to-analog
DAC	digital-to-analog converter—an electronic device, often an integrated circuit, that converts a digital number into a corresponding analog voltage or current
daisy-chain	a method of propagating signals along a bus, in which the devices are prioritized on the basis of their position on the bus
dB	decibel—the unit for expressing a logarithmic measure of the ratio of two signal levels: dB=20log10 V1/V2, for signals in volts
DDS	direct digital synthesis
delta-sigma modulating ADC	a high-accuracy circuit that samples at a higher rate and lower resolution than is needed and (by means of feedback loops) pushes the quantization noise above the frequency range of interest. This out-of-band noise is typically removed by digital filters.
derivative control	a control action with an output that is proportional to the rate of change of the error signal. Derivative control anticipates the magnitude difference between the process variable and the setpoint.
DIFF	differential mode
differential input	an analog input consisting of two terminals, both of which are isolated from computer ground, whose difference is measured

differential measurement system	a way you can configure your device to read signals, in which you do not need to connect either input to a fixed reference, such as the earth or a building ground
digital trigger	a TTL level signal having two discrete levels—a high and a low level
DIN	Deutsche Industrie Norme
dithering	the addition of Gaussian noise to an analog input signal
DMA	direct memory access
DNL	differential nonlinearity—a measure in least significant bit of the worst-case deviation of code widths from their ideal value of 1 LSB
DRAM	dynamic RAM
DSP	digital signal processing
dual-access memory	memory that can be sequentially accessed by more than one controller or processor but not simultaneously accessed. Also known as shared memory.
dual-ported memory	memory that can be simultaneously accessed by more than one controller or processor
dynamic range	the ratio of the largest signal level a circuit can handle to the smallest signal level it can handle (usually taken to be the noise level), normally expressed in decibels

E

ECL	emitter-coupled logic
EGA	enhanced graphics adapter
EISA	Extended Industry Standard Architecture
electrostatically coupled	propagating a signal by means of a varying electric field
ETS	equivalent-time sampling
excitation current	the current used to excite a device to produce a voltage to be measured
expansion ROM	an onboard EEPROM that may contain device-specific initialization and system boot functionality

F

false triggering	triggering that occurs at an unintended time
fetch-and-deposit	a data transfer in which the data bytes are transferred from the source to the controller, and then from the controller to the target
filtering	a type of signal conditioning that allows you to filter unwanted signals from the signal you are trying to measure
flash ADC	an ADC whose output code is determined in a single step by a bank of comparators and encoding logic
floating signal sources	signal sources with voltage signals that are not connected to an absolute reference or system ground. Also called nonreferenced signal sources. Some common example of floating signal sources are batteries, transformers, or thermocouples.
flyby	a type of high-performance data transfer in which the data bytes pass directly from the source to the target without being transferred to the controller
FPGA	field programmable gate array
FSK	frequency shift keying

G

gain	the factor by which a signal is amplified, sometimes expressed in decibels
gain accuracy	a measure of deviation of the gain of an amplifier from the ideal gain
glitch	an undesired change in a signal that typically lasts a short period of time
g_{rms}	level of random vibration

H

half-flash ADC	an ADC that determines its output code by digitally combining the results of two sequentially performed, lower-resolution flash conversions
half-power bandwidth	the frequency range over which a circuit maintains a level of at least –3 dB with respect to the maximum level
handshaked digital I/O	a type of digital acquisition/generation where a device or module accepts or transfers data after a digital pulse has been received. Also called latched digital I/O.
hex	hexadecimal
Hz	hertz

I

IDE	integrated development environment
IEEE 488	the shortened notation for ANSI/IEEE Standards 488-1978, 488.1-1987, and 488.2-1987. *See also* GPIB.
IMD	intermodulation distortion—the ratio, in dB, of the total rms signal level of harmonic sum and difference distortion products, to the overall rms signal level. The test signal is two sine waves added together according to the following standards: SMPTE—A 60 Hz sine wave and a 7 kHz sine wave added in a 4:1 amplitude ratio. DIN—A 250 Hz sine wave and an 8 kHz sine wave added in a 4:1 amplitude ratio. CCIF—A 14 kHz sine wave and a 15 kHz sine wave added in a 1:1 amplitude ratio.
in.	inches
Industrial Device Networks	standardized digital communications networks used in industrial automation applications; they often replace vendor-proprietary networks so that devices from different vendors can communicate in control systems
INL	integral nonlinearity—a measure in LSB of the worst-case deviation from the ideal A/D or D/A transfer characteristic of the analog I/O circuitry

input bias current	the current that flows into the inputs of a circuit
input impedance	the measured resistance and capacitance between the input terminals of a circuit
input offset current	the difference in the input bias currents of the two inputs of an instrumentation amplifier
instrument driver	a set of high-level software functions that controls a specific GPIB, VXI, or RS-232 programmable instrument or a specific plug-in DAQ board. Instrument drivers are available in several forms, ranging from a function callable language to a virtual instrument (VI) in LabVIEW.
instrumentation amplifier	a circuit whose output voltage with respect to ground is proportional to the difference between the voltages at its two inputs
integral control	a control action that eliminates the offset inherent in proportional control
integrating ADC	an ADC whose output code represents the average value of the input voltage over a given time interval
interval scanning	scanning method where there is a longer interval between scans than there is between individual channels comprising a scan
I_{out}	output current
ISA	Industry Standard Architecture
isolation	a type of signal conditioning in which you isolate the transducer signals from the computer for safety purposes. This protects you and your computer from large voltage spikes and makes sure the measurements from the DAQ device are not affected by differences in ground potentials.
isolation voltage	the voltage that an isolated circuit can normally withstand, usually specified from input to input and/or from any input to the amplifier output, or to the computer bus
isothermal	constructed to maintain constant temperature across area. Isothermal construction of terminal blocks increases thermocouple measurement accuracy.

K

kS	1,000 samples

L

linearity	the adherence of device response to the equation R = KS, where R = response, S = stimulus, and K = a constant
linearization	a type of signal conditioning in which software linearizes the voltage levels from transducers, so the voltages can be scaled to measure physical phenomena
low frequency corner	in an AC-coupled circuit, the frequency below which signals are attenuated by at least 3 dB
LSB	least significant bit

M

m	meters
MBLT	eight-byte block transfers in which both the Address bus and the Data bus are used to transfer data
MFLOPS	million floating-point operations per second—the unit for expressing the computational power of a processor
MIO	multifunction I/O
MIPS	million instructions per second—the unit for expressing the speed of processor machine code instructions
MITE	MXI Interface to Everything—a custom ASIC designed by National Instruments that implements the PCI bus interface. The MITE supports bus mastering for high-speed data transfers over the PCI bus.
MSB	most significant bit
MTBF	mean time between failure
MTTR	mean time to repair—predicts downtime and how long it takes to fix a product

multiplexed mode	an SCXI operating mode in which analog input channels are multiplexed into one module output so that your cabled DAQ device has access to the module's multiplexed output as well as the outputs on all other multiplexed modules in the chassis through the SCXI bus. Also called serial mode.
mux	multiplexer—a switching device with multiple inputs that sequentially connects each of its inputs to its output, typically at high speeds, in order to measure several signals with a single analog input channel

N

NB	NuBus—a slot-dependent, 32-bit bus type used in Macintosh computers that has 32 interrupts and does not use DMA
NCO	numerically controlled oscillator
NIST	National Institute of Standards and Technology
NI-TIO	*See* TIO

O

onboard RAM	optional RAM usually installed into SIMM slots
optical coupler, optocoupler	a device designed to transfer electrical signals by utilizing light waves to provide coupling with electrical isolation between input and output. Sometimes called optoisolator or photocoupler.
optical isolation	the technique of using an optoelectric transmitter and receiver to transfer data without electrical continuity, to eliminate high-potential differences and transients
output settling time	the amount of time required for the analog output voltage to reach its final value within specified limits
output slew rate	the maximum rate of change of analog output voltage from one level to another

P

parallel mode	a type of SCXI operating mode in which the module sends each of its input channels directly to a separate analog input channel of the device to the module
passband	the range of frequencies which a device can properly propagate or measure
pattern generation	a type of handshaked (latched) digital I/O in which internal counters generate the handshaked signal, which in turn initiates a digital transfer. Because counters output digital pulses at a constant rate, this means you can generate and retrieve patterns at a constant rate because the handshaked signal is produced at a constant rate.
PC Card	a credit-card-sized expansion card that fits in a PCMCIA slot often referred to as a PCMCIA card
PCI	Peripheral Component Interconnect—a high-performance expansion bus architecture originally developed by Intel to replace ISA and EISA. It is achieving widespread acceptance as a standard for PCs and workstations; it offers a theoretical maximum transfer rate of 132 Mbytes/s.
PCMCIA	an expansion bus architecture that has found widespread acceptance as a de facto standard in notebook-size computers. It originated as a specification for add-on memory cards written by the Personal Computer Memory Card International Association.
PFI	programmable function input
PGIA	programmable gain instrumentation amplifier
photoelectric sensor	an electrical device that responds to a change in the intensity of the light falling upon it
PIA	Peripheral Interface Adapter
PID control	a three-term control mechanism combining proportional, integral, and derivative control actions. Also see proportional control, integral control, and derivative control.
posttriggering	the technique used on a DAQ board to acquire a programmed number of samples after trigger conditions are met

potentiometer	an electrical device the resistance of which can be manually adjusted; used for manual adjustment of electrical circuits and as a transducer for linear or rotary position
pretriggering	the technique used on a DAQ board to keep a continuous buffer filled with data, so that when the trigger conditions are met, the sample includes the data leading up to the trigger condition
proportional control	a control action with an output that is to be proportional to the deviation of the controlled variable from a desired setpoint
protocol	the exact sequence of bits, characters, and control codes used to transfer data between computers and peripherals through a communications channel, such as the GPIB bus
proximity sensor	a device that detects the presence of an object without physical contact. Most proximity sensors provide a digital on/off relay or digital output signal.
PXI	PCI eXtensions for Instrumentation—an open specification that builds on the CompactPCI specification by adding instrumentation-specific features

Q

quantization error	the inherent uncertainty in digitizing an analog value due to the finite resolution of the conversion process
quantizer	a device that maps a variable from a continuous distribution to a discrete distribution

R

reglitch	to modify the glitches in a signal in order to make them less disruptive
relative accuracy	a measure in LSB of the accuracy of an ADC. It includes all nonlinearity and quantization errors. It does not include offset and gain errors of the circuitry feeding the ADC.
resolution	the smallest signal increment that can be detected by a measurement system. Resolution can be expressed in bits, in proportions, or in percent of full scale. For example, a system has 12-bit resolution, one part in 4,096 resolution, and 0.0244% of full scale.

resource locking	a technique whereby a device is signaled not to use its local memory while the memory is in use from the bus
rms	root mean square—the square root of the average value of the square of the instantaneous signal amplitude; a measure of signal amplitude
RTD	resistance temperature detector—a metallic probe that measures temperature based upon its coefficient of resistivity
RTI	referred to input
RTSI bus	real-time system integration bus—the National Instruments timing bus that connects DAQ boards directly, by means of connectors on top of the boards, for precise synchronization of functions

S

S	samples
sample counter	the clock that counts the output of the channel clock, in other words, the number of samples taken. On boards with simultaneous sampling, this counter counts the output of the scan clock and hence the number of scans.
SCADA	supervisory control and data acquisition—a common PC function in process control applications, where programmable logic controllers (PLCs) perform control functions but are monitored and supervised by a PC
scan	one or more analog or digital input samples. Typically, the number of input samples in a scan is equal to the number of channels in the input group. For example, one pulse from the scan clock produces one scan which acquires one new sample from every analog input channel in the group.
scan clock	the clock controlling the time interval between scans. On boards with interval scanning support (for example, the AT-MIO-16F-5), this clock gates the channel clock on and off. On boards with simultaneous sampling (for example, the EISA-A2000), this clock clocks the track-and-hold circuitry.
scan rate	the number of scans per second. For example, a scan rate of 10 Hz means sampling each channel 10 times per second.
sec	seconds

self-calibrating	a property of a DAQ board that has an extremely stable onboard reference and calibrates its own A/D and D/A circuits without manual adjustments by the user
settling time	the amount of time required for a voltage to reach its final value within specified limits
Shannon Sampling Theorem	a law of sampling theory stating that if a continuous bandwidth-limited signal contains no frequency components higher than half the frequency at which it is sampled, then the original signal can be recovered without distortion
S/H	sample-and-hold—a circuit that acquires and stores an analog voltage on a capacitor for a short period of time
shared memory	*See* dual-access memory
SIMM	single in-line memory module
SMB	a type of miniature coaxial signal connector
SNR	signal-to-noise ratio—the ratio of the overall rms signal level to the rms noise level, expressed in decibels
software trigger	a programmed event that triggers an event such as data acquisition
SPDT	single-pole double throw—a property of a switch in which one terminal can be connected to one of two other terminals
SS	simultaneous sampling—a property of a system in which each input or output channel is digitized or updated at the same instant
STC	System Timing Controller
strain gauge	a thin conductor, which is attached to a material, that detects stress or vibrations in that material. The conductor's resistance is a function of the applied force.
statically configured device	a device whose logical address cannot be set through software; that is, it is not dynamically configurable
successive-approximation ADC	an ADC that sequentially compares a series of binary-weighted values with an analog input to produce an output digital word in *n* steps, where *n* is the bit resolution of the ADC

switchless device	devices that do not require dip switches or jumpers to configure resources on the devices—also called Plug and Play devices
system noise	a measure of the amount of noise seen by an analog circuit or an ADC when the analog inputs are grounded

T

T/H	track-and-hold—a circuit that tracks an analog voltage and holds the value on command
THD	total harmonic distortion—the ratio of the total rms signal due to harmonic distortion to the overall rms signal, in decibel or a percentage
THD+N	signal-to-THD plus noise—the ratio in decibels of the overall rms signal to the rms signal of harmonic distortion plus noise introduced
thermistor	a semiconductor sensor that exhibits a repeatable change in electrical resistance as a function of temperature. Most thermistors exhibit a negative temperature coefficient.
thermocouple	a temperature sensor created by joining two dissimilar metals. The junction produces a small voltage as a function of the temperature.
TIO	a National Instruments timing and triggering controller ASIC. The TIO includes four general purpose counter/timers used for applications such as event counting, period and frequency measurement, and pulse train generation. It also includes a trigger routing and condition mechanism for connecting RTSI bus and board-specific trigger and timing signals. The TIO also provides advanced digital I/O capabilities for time-stamping multiple I/O lines and controlling digital output lines.
TTL	transistor-transistor logic

U

UART	universal asynchronous receiver/transmitter—an integrated circuit that converts parallel data to serial data (and vice versa), commonly used as a computer bus to serial device interface for serial communication
UI	update interval
unipolar	a signal range that is always positive (for example, 0 to +10 V)

update	the output equivalent of a scan. One or more analog or digital output samples. Typically, the number of output samples in an update is equal to the number of channels in the output group. For example, one pulse from the update clock produces one update which sends one new sample to every analog output channel in the group.
update rate	the number of output updates per second

V

V	volts
VDC	volts direct current
VDMAD	virtual DMA driver
V_{EXT}	external volt
V_{IH}	volts, input high
V_{IL}	volts, input low
V_{in}	volts in
VISA	virtual instrument software architecture—a new driver software architecture developed by National Instruments to unify instrumentation software GPIB, DAQ, and VXI. It has been accepted as a standard for VXI by the VXIplug&play Systems Alliance.
visual basic custom control (VBXs)	a specific form of binary packaged object that can be created by different companies and integrated into applications written using Visual Basic
VPICD	virtual programmable interrupt controller device

W

working voltage	the highest voltage that should be applied to a product in normal use, normally well under the breakdown voltage for safety margin. *See also* breakdown voltage.

Z

zero-overhead looping	the ability of a high-performance processor to repeat instructions without requiring time to branch to the beginning of the instructions
zero-wait-state memory	memory fast enough that the processor does not have to wait during any reads and writes to the memory

Index

E

F

G

I

J

L

M

N

O

P

R

S

T

U

W