DAQ

DAQCard™-DIO-24 User Manual

Low-Cost Digital I/O PC Card

October 1997 Edition
Part Number 320788D-01

Internet Support

`support@natinst.com`
E-mail: `info@natinst.com`
FTP Site: `ftp.natinst.com`
Web Address: `http://www.natinst.com`

Bulletin Board Support

BBS United States: (512) 794-5422
BBS United Kingdom: 01635 551422
BBS France: 01 48 65 15 59

Fax-on-Demand Support

(512) 418-1111

Telephone Support (U.S.)

Tel: (512) 795-8248
Fax: (512) 794-5678

International Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521, 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, Israel 03 5734815, Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456, Mexico 5 520 2635, Netherlands 0348 433466, Norway 32 84 84 00, Singapore 2265886, Spain 91 640 0085, Sweden 08 730 49 70, Switzerland 056 200 51 51, Taiwan 02 377 1200, United Kingdom 01635 523545

National Instruments Corporate Headquarters

6504 Bridge Point Parkway   Austin, TX 78730-5039   Tel: (512) 794-0100

Important Information

Warranty

The DAQCard-DIO-24 is warranted against defects in materials and workmanship for a period of one year from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

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

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

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

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

Copyright

Trademarks

LabVIEW™, NI-DAQ™, RTSI™, BridgeVIEW™, Component Works™, CVI™, DAQArb™, DAQCard™, DAQDesigner™, DAQMeter™, DAQPad™, DAQ-PnP™, DAQ-STC™, DAQWare™, NI-DSP™, NI-PGIA™, SCXI™, and VirtualBench™ are trademarks of National Instruments Corporation.

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

FCC/DOC Radio Frequency Interference Class A Compliance

This equipment generates and uses radio frequency energy and, if not installed and used in strict accordance with the instructions in this manual, may cause interference to radio and television reception. Classification requirements are the same for the Federal Communications Commission (FCC) and the Canadian Department of Communications (DOC). This equipment has been tested and found to comply with the following two regulatory agencies:

Federal Communications Commission

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Notices to User: ***Changes or modifications not expressly approved by National Instruments could void the user's authority to operate the equipment under the FCC Rules.***

This device complies with the FCC rules only if used with shielded interface cables of suitable quality and construction. National Instruments used such cables to test this device and provides them for sale to the user. The use of inferior or nonshielded interface cables could void the user's authority to operate the equipment under the FCC rules.

If necessary, consult National Instruments or an experienced radio/television technician for additional suggestions. The following booklet prepared by the FCC may also be helpful: *Interference to Home Electronic Entertainment Equipment Handbook*. This booklet is available from the U.S. Government Printing Office, Washington, DC 20402.

Canadian Department of Communications

This Class A digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations.

Cet appareil numérique de la classe A respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

Table of Contents

Chapter 4
Signal Connections

Appendix A
Specifications

Appendix B
Register-Level Programming

Appendix C
OKI 82C55A Data Sheet

Appendix D
PC Card Questions and Answers for Windows 3.1

Appendix E
Customer Communication

Glossary

Index

Figures

Tables

About This Manual

This manual describes mechanical and electrical aspects of the DAQCard-DIO-24 and contains information concerning its operation, installation and configuration, basic programming, and hardware operation.

This manual explains how to use the DAQCard-DIO-24 with the NI-DAQ driver software included in your DAQCard-DIO-24 kit.

The DAQCard-DIO-24 is a 24-bit, parallel digital I/O card with an 82C55A programmable peripheral interface (PPI). The DAQCard-DIO-24 is a member of the National Instruments DAQCard Series of PC Card I/O channel expansion cards. These cards are designed for low-cost data acquisition and control for applications in laboratory testing, production testing, and industrial process monitoring and control.

Organization of This Manual

The *DAQCard-DIO-24 User Manual* is organized as follows:

- Chapter 1, *Introduction*, describes the DAQCard-DIO-24, lists what you need to get started, describes software programming choices and optional equipment, and explains how to unpack your DAQCard-DIO-24.
- Chapter 2, *Installation and Configuration*, describes how to install and configure the DAQCard-DIO-24.
- Chapter 3, *Hardware Overview*, contains a functional overview of the DAQCard-DIO-24 and explains the operation of each functional unit making up the card.
- Chapter 4, *Signal Connections*, contains I/O connector signal descriptions, handshake timing diagrams, and cabling instructions.
- Appendix A, *Specifications*, lists the specifications for the DAQCard-DIO-24.

- Appendix B, *Register-Level Programming*, describes in detail the address and function of each of the DAQCard-DIO-24 control and status registers.
- Appendix C, *OKI 82C55A Data Sheet*, contains the manufacturer data sheet for the OKI Semiconductor 82C55A CMOS programmable peripheral interface (PPI). This interface is used on the DAQCard-DIO-24.
- Appendix D, *PC Card Questions and Answers for Windows 3.1*, contains a list of common questions and answers relating to PC Card (PCMCIA) operation.
- Appendix E, *Customer Communication*, contains forms you can use to request help from National Instruments or to comment on our products.
- The *Glossary* contains an alphabetical list and description of terms used in this manual, including abbreviations, acronyms, metric prefixes, mnemonics, and symbols.
- The *Index* alphabetically lists the topics in this manual, including the page where you can find each one.

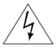
Conventions Used in This Manual

The following conventions are used in this manual:

< > — Angle brackets containing numbers separated by an ellipses represent a range, signal, or port (for example, ACH<0..7> stands for ACH0 through ACH7).

 This icon to the left of bold italicized text denotes a note, which alerts you to important information.

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

 This icon to the left of bold italicized text denotes a warning, which advises you of precautions to take to avoid being electrically shocked.

board — Board refers to National Instruments data acquisition boards.

bold — Bold text denotes the names of error messages, menus, menu items, or parameters.

bold italic	Bold italic text denotes a note, caution, or warning.
card	Card refers to the DAQCard-DIO-24 unless otherwise stated.
device	Device refers to a National Instruments data acquisition board, card, or SCXI module.
italic	Italic text denotes emphasis, a cross reference, or an introduction to a key concept.
`italic monospace`	Italic text in this font denotes that you must supply the appropriate words or values in the place of these items.
Macintosh	Macintosh refers to a Macintosh or Power Book equipped with a Type II PCMCIA +5 V-capable slot.
`monospace`	Text in this font denotes text or characters that are to be literally input 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, variables, filenames, and extensions, and for statements and comments taken from program code.
NI-DAQ	NI-DAQ refers to the NI-DAQ software for PC compatibles or Macintosh unless otherwise stated.
PC	PC refers to an IBM PC/XT, PC AT, Personal System/2, or laptop compatible computer that is equipped with a PCMCIA standard version 2.0 or later bus interface and a Type II +5 V-capable slot.
SCXI	SCXI stands for Signal Conditioning eXtensions for Instrumentation and is a National Instruments product line designed to perform front-end signal conditioning for National Instruments plug-in DAQ devices.

Abbreviations, acronyms, metric prefixes, mnemonics, and symbols are listed in the *Glossary*.

About the National Instruments Documentation Set

The *DAQCard-DIO-24 User Manual* is one piece of the documentation set for your data acquisition (DAQ) system. You could have any of several types of manuals, depending on the hardware and software in your system. Use the different types of manuals you have as follows:

- *Getting Started with SCXI*—If you are using SCXI, this is the first manual you should read. It gives an overview of the SCXI system and contains the most commonly needed information for the modules, chassis, and software.
- Your SCXI hardware user manuals—If you are using SCXI, read these manuals next for detailed information about signal connections and module configuration. They also explain in greater detail how the module works and contain application hints.
- Your DAQ hardware user manuals—These manuals have detailed information about the DAQ hardware that plugs into or is connected to your computer. Use these manuals for hardware installation and configuration instructions, specification information about your DAQ hardware, and application hints.
- Software manuals—Examples of software manuals you may have are the LabVIEW and LabWindows®/CVI manual sets and the NI-DAQ manuals (a 4.6.1 or earlier version of NI-DAQ supports LabWindows for DOS). After you set up your hardware system, use either the application software (LabVIEW or LabWindows/CVI) manuals or the NI-DAQ manuals to help you write your application. If you have a large and complicated system, it is worthwhile to look through the software manuals before you configure your hardware.
- Accessory installation guides or manuals—If you are using accessory products, read the terminal block and cable assembly installation guides or accessory board user manuals. They explain how to physically connect the relevant pieces of the system. Consult these guides when you are making your connections.
- *SCXI Chassis User Manual*—If you are using SCXI, read this manual for maintenance information on the chassis and installation instructions.

Related Documentation

The following documents may be helpful for register-level programming:

- *PC Card Standard; Card Services Specification, Socket Services Specification,* and other volumes. Personal Computer Memory Card International Association (PCMCIA)
- If you are a Macintosh user, the *PC Card Development Kit* (available from Apple through APDA)

Customer Communication

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

Introduction

Chapter 1

This chapter describes the DAQCard-DIO-24, lists what you need to get started, describes software programming choices and optional equipment, and explains how to unpack your DAQCard-DIO-24.

About the DAQCard-DIO-24

Thank you for buying the National Instruments DAQCard-DIO-24. The DAQCard-DIO-24 is a low-cost, low-power, 24-bit, parallel digital I/O card for PCs and Macintoshes equipped with a Type II PCMCIA slot. An OKI 82C55A PPI controls the 24 bits of digital I/O. The 82C55A is flexible and powerful when interfacing with peripheral equipment, can operate in either a unidirectional or bidirectional bus mode, and can generate interrupt request outputs. The 82C55A can be programmed for a wide variety of 8-bit or 16-bit digital I/O applications. With the standard 50-pin I/O connector on available cables, you can easily connect digital signals to the DAQCard-DIO-24. The pin assignments for this connector are compatible with standard 24-channel digital I/O applications.

The DAQCard-DIO-24 is packaged with NI-DAQ software, the National Instruments complete driver with a library of DAQ functions for DOS and Windows or Macintosh applications. Using NI-DAQ, you can quickly and easily start your application without having to program the card at the register level.

The small size and weight of the DAQCard-DIO-24, coupled with its low power consumption, make this card convenient for use in portable computers. This portability makes remote data acquisition practical. The card requires very little power when operating, thus extending the life of your computer batteries.

In addition, the low cost of a system based on the DAQCard-DIO-24 makes it ideal for laboratory work in industrial and academic environments. You can use the 24 TTL-compatible digital I/O lines to switch external devices, such as transistors and solid-state relays, read the status of external digital logic, and generate interrupts.

You can use the DAQCard-DIO-24 in a wide range of digital I/O applications. For example, you can connect the DAQCard-DIO-24 to any of the following: panel meters, instruments and test equipment with BCD readouts and controls, or optically isolated, solid-state relays and I/O module mounting racks.

With the DAQCard-DIO-24, your computer can serve as a versatile, cost-effective digital I/O system controller for laboratory testing, production testing, and industrial process monitoring and control.

For detailed DAQCard-DIO-24 specifications, see Appendix A, *Specifications*.

What You Need to Get Started

To set up and use your DAQCard-DIO-24 card, you will need the following:

- ❑ DAQCard-DIO-24 card
- ❑ *DAQCard-DIO-24 User Manual*
- ❑ One of the following software packages and documentation:
    - ComponentWorks
    - LabVIEW for Macintosh
    - LabVIEW for Windows
    - LabWindows/CVI for Windows
    - NI-DAQ for Macintosh
    - NI-DAQ for PC Compatibles
- ❑ Your computer

Software Programming Choices

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

National Instruments Application Software

ComponentWorks contains tools for data acquisition and instrument control built on NI-DAQ driver software. ComponentWorks provides a higher-level programming interface for building virtual instruments through standard OLE controls and DLLs. With ComponentWorks, you can use all of the configuration tools, resource management utilities, and interactive control utilities included with NI-DAQ.

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

Using ComponentWorks, 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 is not packaged with SCXI or accessory products, except for the SCXI-1200. NI-DAQ has an extensive library of functions that you can call 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 (timed D/A conversion), digital I/O, counter/timer operations, SCXI, RTSI, self-calibration, messaging, and acquiring data to 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 of 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 the performance of National Instruments DAQ devices because it lets multiple devices operate at their peak performance.

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 among its different versions so that you can change platforms with minimal modifications to your code. Whether you are using conventional programming languages, ComponentWorks, LabVIEW, LabWindows/CVI, or other application software, your application uses the NI-DAQ driver software, as illustrated in Figure 1-1.

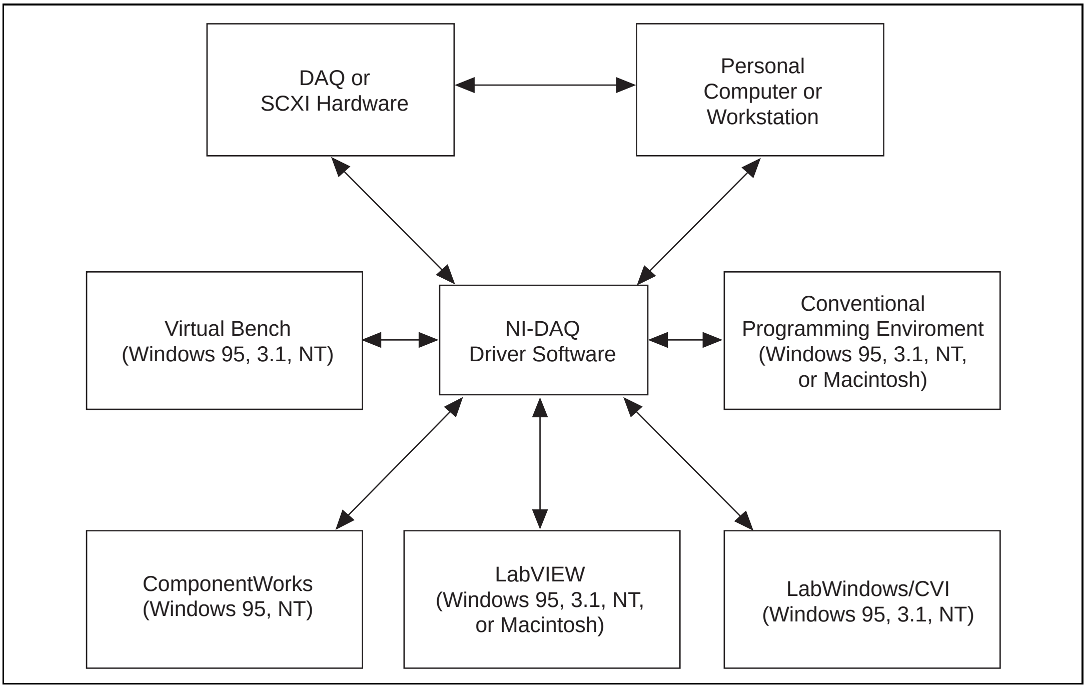


Figure 1-1. The Relationship between the Programming Environment, NI-DAQ, and Your Hardware

Register-Level Programming

The final option for programming any National Instruments DAQ hardware is to write register-level software. Writing register-level programming software can be very time-consuming and inefficient, and is not recommended for most users.

Even if you are an experienced register-level programmer, consider using NI-DAQ, or other National Instruments application software to program your National Instruments DAQ hardware. Using the NI-DAQ, ComponentWorks, LabVIEW, or LabWindows/CVI software is easier than, and as flexible as, register-level programming, and can save weeks of development time.

Optional Equipment

National Instruments offers a variety of products to use with your DAQCard-DIO-24, including cables, connector blocks, and other accessories, as follows:

- Cables and cable assemblies, shielded and ribbon
- Connector blocks, shielded and unshielded 50, 68, and 100-pin screw terminals
- Signal condition eXtension for instrumentation (SCXI) modules and accessories for isolating, amplifying, exciting, and multiplexing signals for relays and analog output. With SCXI you can condition and acquire up to 3,072 channels
- Low channel count signal conditioning modules, boards, and accessories, including conditioning for strain gauges and RTDs, simultaneous sample and hold, and relays

For more specific information about these products, refer to your National Instruments catalog or call the office nearest you.

Custom Cables

The DAQCard-DIO-24 I/O connector is a 25-pin female PC Card I/O connector. The manufacturer part number National Instruments uses for this connector is as follows:

- Elco Corporation (part number 21 5098 0004 00 001)

The mating connector for the DAQCard-DIO-24 is a 25-position male PC Card I/O connector. The recommended manufacturer part number for this mating connector is as follows:

- Elco Corporation (part number 32 5098 0003 00 001)

The cable (26-conductor, 30 AWG, stranded, twisted pair, shielded) that can be used with these connectors is as follows:

- Madison Corporation (part number 10482)

The 50-pin connector on the optional PSH27-50F-D1 cable available for the DAQCard-DIO-24 is a 50-position female polarized ribbon socket connector with strain relief. The manufacturer part numbers National Instruments uses for this header are as follows:

- 3M/Electronic Products Division (part number 3425-H650; strain relief part number 3448-3050)
- AMP Corporation (part number 1-746288-0; strain relief part number 499252-4)

The mating connector for the 50-pin socket is a 50-pin male ribbon cable header. National Instruments uses a polarized (keyed) connector to prevent inadvertent upside-down connection to the DAQCard-DIO-24. Recommended manufacturer part numbers for this mating connector are as follows:

- 3M/Electronic Products Division (part number 3433-6302)
- Berg Corporation (part number 71912-150)

Unpacking

The DAQCard-DIO-24 is shipped in an antistatic vinyl envelope. When you are not using the DAQCard-DIO-24, you should store it in this envelope.

Because the DAQCard-DIO-24 is enclosed in a fully shielded case, no additional electrostatic precautions are necessary.

Caution: ***For your own safety and to protect the DAQCard-DIO-24, NEVER attempt to touch the pins of the connectors.***

Installation and Configuration

Chapter 2

This chapter describes how to install and configure the DAQCard-DIO-24.

Software Installation

Install your software before you install your DAQCard-DIO-24 device. Refer to the appropriate release notes indicated below for specific instructions on the software installation sequence.

If you are using NI-DAQ, refer to your NI-DAQ release notes. Find the installation section for your operating system and follow the instructions given there.

If you are using LabVIEW, LabWindows/CVI, or other National Instruments application software packages, refer to the appropriate release notes. After you have installed your application software, refer to your NI-DAQ release notes and follow the instructions given there for your operating system and application software package.

If you are a register-level programmer, refer to Appendix B, *Register-Level Programming*.

Hardware Installation

You can install your DAQCard-DIO-24 in any available Type II PCMCIA slot in your computer. See Figure 2-1 for the completed installation.

1. Turn off your computer. If your computer supports hot insertion, you may insert or remove the DAQCard-DIO-24 at any time, whether the computer is powered on or off.
2. Remove the PCMCIA slot cover on your computer, if any.
3. Insert the PCMCIA bus connector of the DAQCard-DIO-24 into the PCMCIA slot. The card is keyed so that you can insert it only one way.

4. Attach the I/O cable. Be very careful not to put strain on the I/O cable when inserting it into and removing it from the DAQCard-DIO-24. When plugging and unplugging the cable, always grasp the cable by the connector. *Never* pull directly on the I/O cable to unplug it from the DAQCard-DIO-24.

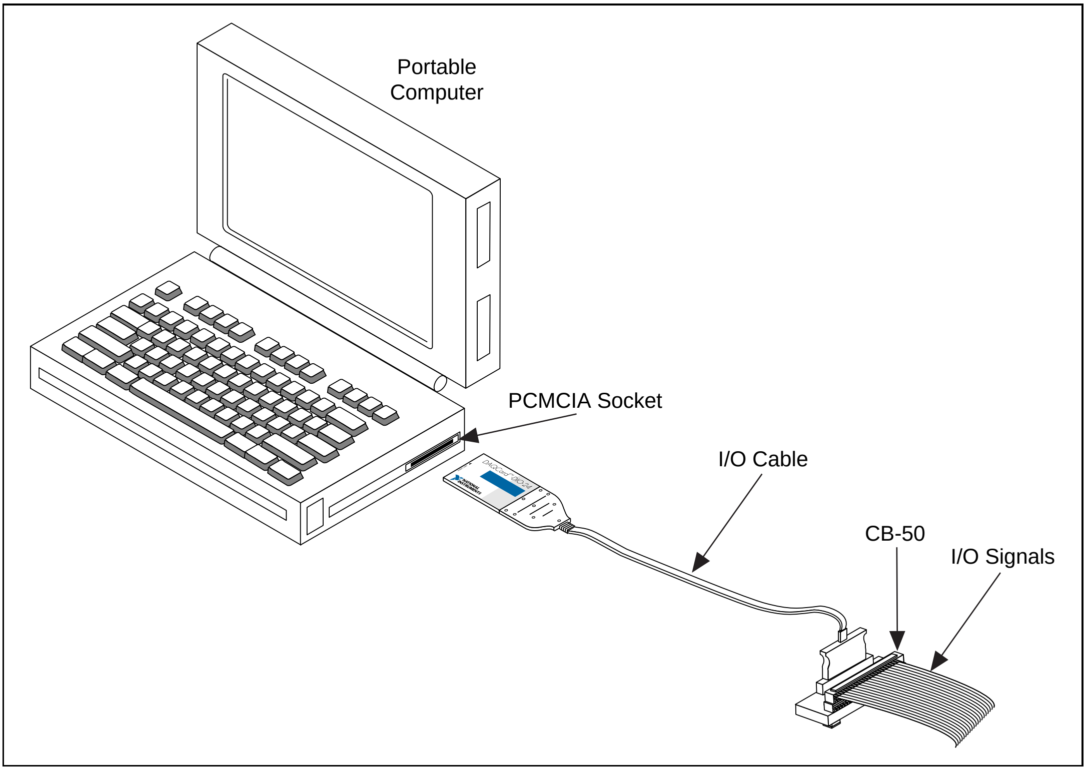


Figure 2-1. A Typical Configuration for the DAQCard-DIO-24

Card Configuration

The DAQCard-DIO-24 is completely software configurable. Refer to your software documentation for configuration information. If you are a register-level programmer, refer to Appendix B, *Register-Level Programming*.

Chapter 3

Hardware Overview

This chapter contains a functional overview of the DAQCard-DIO-24 and explains the operation of each functional unit making up the card.

Functional Overview

The block diagram in Figure 3-1 illustrates the key functional components of the DAQCard-DIO-24.

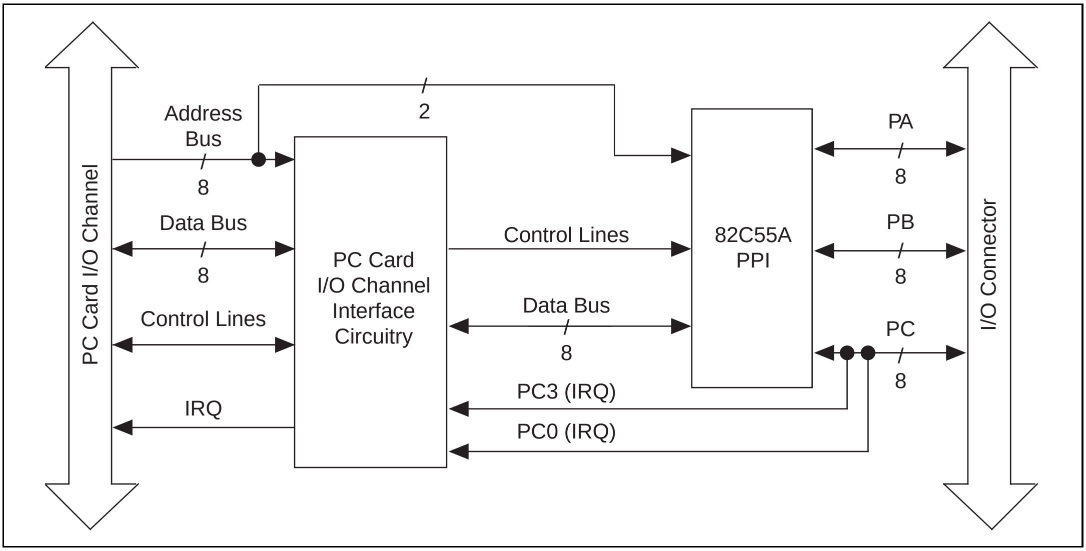


Figure 3-1. DAQCard-DIO-24 Block Diagram.

82C55A Programmable Peripheral Interface

The 82C55A PPI is the heart of the DAQCard-DIO-24. This chip has 24 programmable I/O pins that represent three 8-bit ports—port A, port B, and port C. The chip also has three modes of operation: simple I/O (mode 0), strobed I/O (mode 1), and bidirectional bus I/O (mode 2). In mode 0, you can program port A, port B, and the upper and lower four-bit nibbles of port C as input or output lines. In modes 1 and 2, port A and port B use some or all of the port C lines as handshaking control lines. You can configure port B for mode 0 or 1, and port A for mode 0, 1, or 2.

PC Card I/O Channel Interface Circuitry

The PC Card I/O channel consists of an address bus, a data bus, interrupt lines, and several control and support signals. The PC Card I/O channel interface circuitry consists of an address decoder, data buffers, interrupt controller circuitry, timing interface circuitry, a card information structure, and PC Card control registers. The components making up the DAQCard-DIO-24 PC Card I/O channel interface circuitry are shown in Figure 3-2.

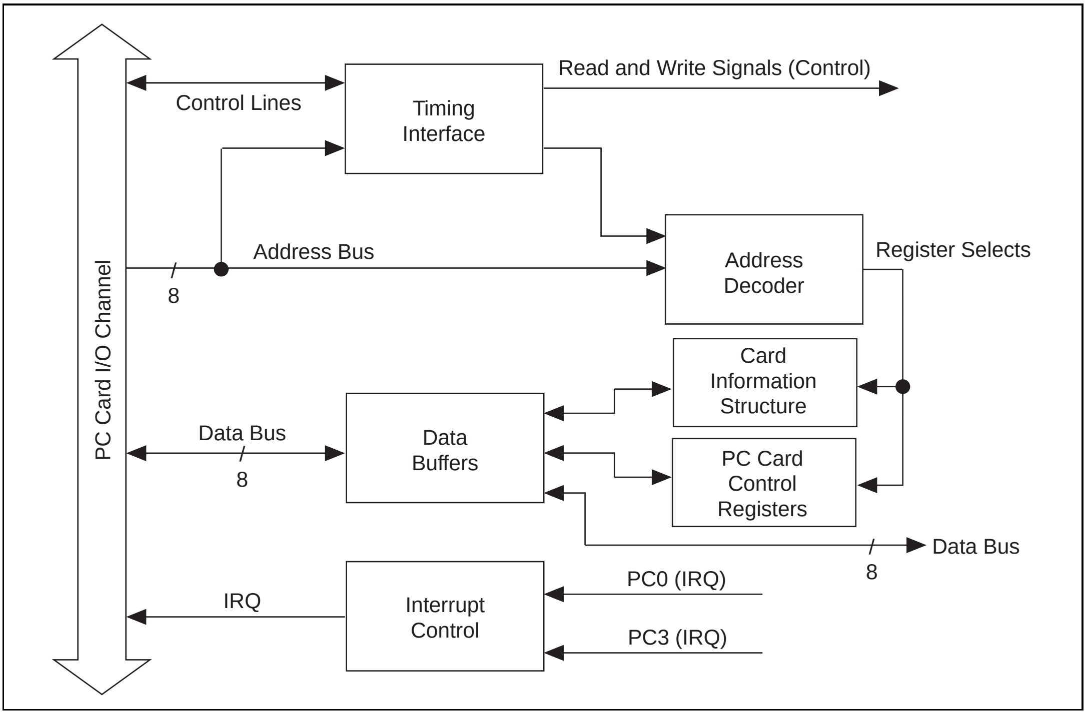


Figure 3-2. PC Card I/O Channel Interface Circuitry Block Diagram

When the card is inserted, the software selects the interrupt level that the DAQCard-DIO-24 uses. Two software-controlled registers determine which events, if any, generate interrupts. You can program the 82C55A to generate interrupt requests by setting PC3 for group A or PC0 for group B. When interrupts are enabled for group A, an active high signal on the PC3 line generates an interrupt request. When interrupts are enabled for group B, an active high signal on the PC0 line generates an interrupt request.

In addition to selecting interrupt levels, the system examines information stored in the DAQCard-DIO-24 card information structure. The software uses this data to configure the card appropriately for the system in which it is used. When the system has assigned the card to a section of memory, it updates the PC Card control registers and initializes the card.

The rest of the circuitry consists of address decoders, data buffers, I/O channel interface timing control circuitry, and interrupt control circuitry. The decode circuitry uses the PCMCIA bus signal CE1*, which the PCMCIA Card and Socket Services software controls, as the board-enable signal and uses lines A0 through A4 plus timing signals to generate the onboard register-select signals and read/write signals. The data buffers control the direction of data transfer on the bidirectional data lines based on whether the transfer is a read or write. The DAQCard-DIO-24 uses only 8-bit transfers. The interrupt control circuitry routes any enabled interrupts to the IREQ* line, which the system motherboard routes to an available interrupt request line.

Digital I/O Connector

All digital I/O is transmitted through a 25-pin PC Card I/O connector. The optional PSH27-50F-D1 cable can connect the DAQCard-DIO-24 to a standard 50-pin male connector. The pin assignments for the 50-pin I/O connector are compatible with standard 24-channel digital I/O applications. All even-numbered pins on this 50-pin connector are attached to the card's ground. Pin 49 is connected to +5 V, which is often required to operate I/O module mounting racks. See Chapter 4, *Signal Connections*, for additional information on pin assignments.

Chapter 4

Signal Connections

This chapter contains I/O connector signal descriptions, handshake timing diagrams, and cabling instructions.

I/O Connector

Figure 4-1 shows the pin assignments for the DAQCard-DIO-24 digital I/O connector.

Caution: ***Connections that exceed any of the maximum ratings of input or output signals on the DAQCard-DIO-24 can damage the card and the computer. Maximum ratings for each signal are given in this chapter under the discussion of that signal. National Instruments is NOT liable for any damages resulting from any such signal connections.***

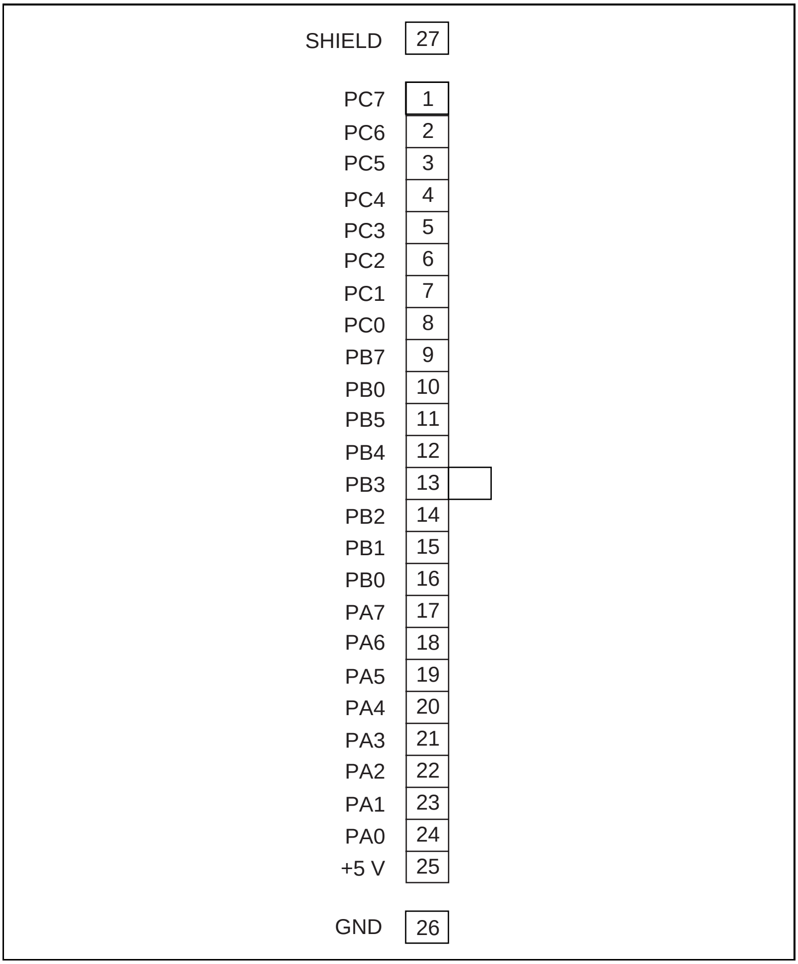


Figure 4-1. Digital I/O Connector Pin Assignments

Signal Connection Descriptions

Table 4-1. Signal Connection Descriptions

Pin	Signal Name	Description
1–8	PC<7..0>	Port C—Bidirectional data lines. PC7 is the MSB, PC0 the LSB.
8–16	PB<7..0>	Port B—Bidirectional data lines. PB7 is the MSB, PB0 the LSB.
17–24	PA<7..0>	Port A—Bidirectional data lines. PA7 is the MSB, PA0 the LSB.
25	+5 V	+5 Volts—This pin provides +5 VDC. The +5 V supply is fused at 1.0 A, which is the maximum current available.
26	SHIELD	Shield—This pin is connected to the card's shield.
27	GND	Ground—This signal is connected to the computer ground signal.

The absolute maximum voltage input rating is –0.5 to +5.5 V with respect to GND.

Port C Pin Assignments

The signals assigned to port C depend on how you configure the 82C55A. In mode 0, or no-handshaking mode, port C is configured as two 4-bit I/O ports. If you configure port B for handshaking, lines PC<0..2> are used as port B handshaking control lines. If you configure port A for handshaking, some or all of lines PC<3..7> are used as port A handshaking control lines, depending on the mode you select (mode 1 input, mode 1 output, or mode 2).

Table 4-2 summarizes the port C signal assignments for modes 0, 1, and 2. Mode 2 is only available for port A. You can configure port A and port B for two different modes. Any port C lines not configured as handshaking control lines are available for general-purpose input or output.

Table 4-2. Port C Signal Assignments

Programming Mode	Port C Data Lines, Upper Nibble				Port C Data Lines, Lower Nibble			
	PC7	**PC6**	**PC5**	**PC4**	**PC3**	**PC2**	**PC1**	**PC0**
Mode 0	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O
Mode 1 Input	I/O	I/O	IBF_A	STB_A*	$INTR_A$	STB_B*	$IBFB_B$	$INTR_B$
Mode 1 Output	OBF_A*	ACK_A*	I/O	I/O	$INTR_A$	ACK_B*	OBF_B*	$INTR_B$
Mode 2	OBF_A*	ACK_A*	IBF_A	STB_A*	$INTR_A$	I/O	I/O	I/O

* Indicates that the signal is active low.
Subscripts A and B denote port A or port B handshaking signals.

Power-up Pin State Considerations and Defaults

At startup, the ports on the DAQCard-DIO-24 default to mode 0 input, and all digital lines are pulled up to logic high. In addition, any digital line that is allowed to float will also be pulled up to logic high. To keep a digital line at logic low, instead of logic high, at startup or when it is being allowed to float, connect a 4.7 kΩ resistor from the digital line to ground in parallel with the external device. For example, if you have the DAQCard-DIO-24 connected to a CB-50 I/O connector block and want to pull down PC7 to logic low, you can connect a 4.7 kΩ resistor from pin 1 to any even-numbered ground pin on the 50-pin connector.

Cable Connector

The optional PSH27-50F-D1 cable assembly available for the DAQCard-DIO-24 is designed to connect the card to such National Instruments products as the CB-50 and the SC-2051.

At one end of this cable assembly is a 27-pin connector that plugs into the DAQCard-DIO-24. At the other end of this cable assembly is a standard 50-pin, female .100 in. centerline cable connector with a center polarization key. Figure 4-2 shows the pinout for this 50-pin female cable connector.

GND	2	1	PC7
GND	4	3	PC6
GND	6	5	PC5
GND	8	7	PC4
GND	10	9	PC3
GND	12	11	PC2
GND	14	13	PC1
GND	16	15	PC0
GND	18	17	PB7
GND	20	19	PB0
GND	22	21	PB5
GND	24	23	PB4
GND	26	25	PB3
GND	28	27	PB2
GND	30	29	PB1
GND	32	31	PB0
GND	34	33	PA7
GND	36	35	PA6
GND	38	37	PA5
GND	40	39	PA4
GND	42	41	PA3
GND	44	43	PA2
GND	46	45	PA1
GND	48	47	PA0
GND	50	49	+5 V

Figure 4-2. Cable Connector Pin Assignments

☞ **Note:** ***Use the key at pin 25 to orient the connector. Ignore any*** *pin 1* ***marking on the connector***

Table 4-3. Cable Connector Pin Descriptions

Pin	Signal Name	Description
1, 3, 5, 7, 9, 11, 13, 15	PC<7..0>	Port C—Bidirectional data lines. PC7 is the MSB; PC0 is the LSB.
2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50	GND	Ground—These signals are connected to the computer ground signal.
17, 19, 21, 23, 25, 27, 29, 31	PB<7..0>	Port B—Bidirectional data lines. PB7 is the MSB; PB0 is the LSB.
33, 35, 37, 39, 41, 43, 45, 47	PA<7..0>	Port A—Bidirectional data lines. PA7 is the MSB; PA0 is the LSB.
49	+5 V	+5 Volts—This pin provides +5 VDC. The +5 V supply is fused by a thermal resettable fuse rated at 1.0 A, which is the maximum current available. Note that the thermal resettable fuse will begin to open after exceeding 1.0 A and will return to normal operating conditions when cooled.

Digital I/O Signal Connections

Pins 1 through 24 of the I/O connector are digital I/O signal pins. Pin 27 is a digital ground pin.

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

- Absolute maximum voltage rating:

  –0.5 to +5.5 V with respect to GND

- Digital input specifications (referenced to GND):

– Input logic high voltage	2.2 V minimum	5.3 V maximum
– Input logic low voltage	–0.3 V minimum	0.8 V maximum
– Maximum input current ($0 < V_{in} < 5$ V)	–1 µA minimum	1 µA maximum

- Digital output specifications (referenced to GND):

– Output logic high voltage at	3.7 V minimum $I_{out} = –2.5$ mA	—
– Output logic low voltage	—	0.4 V maximum at $I_{out} = 2.5$ mA

Figure 4-3 shows an example of connections to the digital input and output ports. In this figure, port A of the 82C55A 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 switch shown in Figure 4-3. Digital output applications include sending TTL signals and driving external devices such as the LED shown in Figure 4-3.

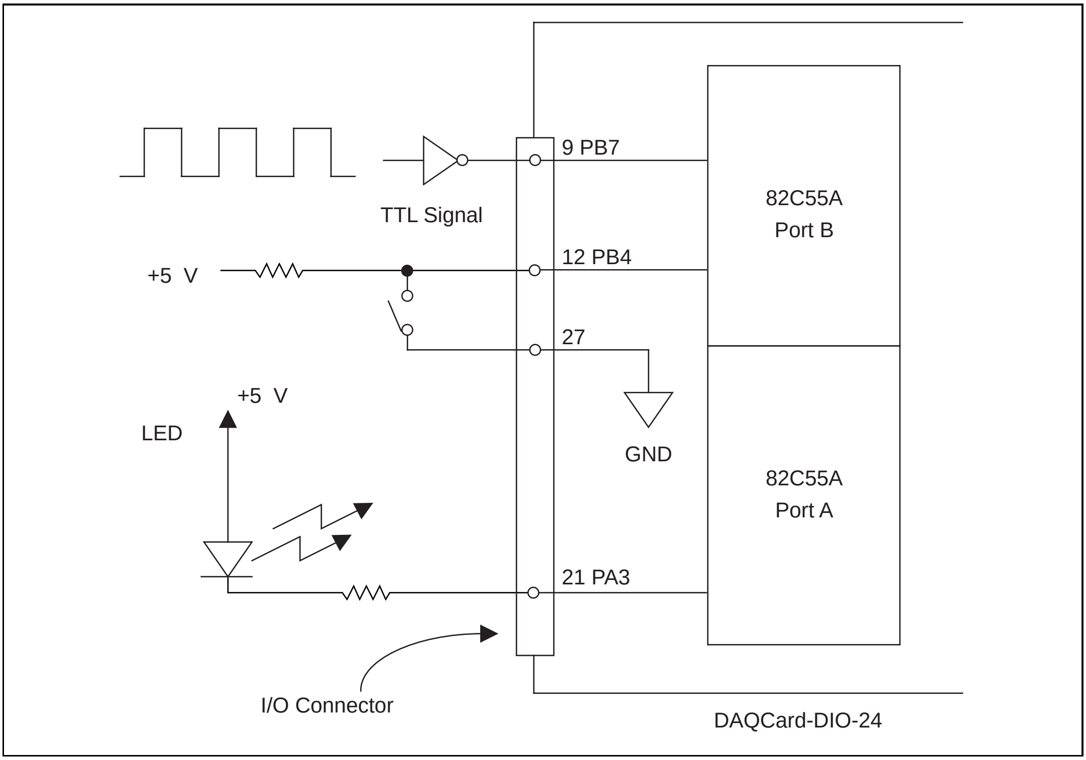


Figure 4-3. Digital I/O Signal Connections

Caution: ***Do not exceed any of the maximum ratings of input or output signal connections on the DAQCard-DIO-24. Doing so can damage the DAQCard-DIO-24 and the computer. Exceeding the maximum connections includes connecting any power signals to ground and vice versa. National Instruments is not liable for any damages resulting from any such signal connections.***

Power Connections

Pin 25 of the I/O connector provides +5 V from the PC Card I/O channel power supply. This pin is referenced to GND and can be used to power external digital circuitry that draws up to 1.0 A. The +5 V power supply is fused by a thermal resettable fuse rated at 1.0 A. Note that the thermal resettable fuse will begin to open after exceeding 1.0 A and will return to normal operating conditions when cooled. The actual current available from this signal may be less than 200 mA, depending on the computer. Notice also that any current drawn from this line adds to the power requirements from the computer.

Timing Specifications

This section lists the timing specifications for handshaking with the DAQCard-DIO-24. The handshaking lines STB* and IBF synchronize input data transfers. The handshaking lines OBF* and ACK* synchronize output data transfers.

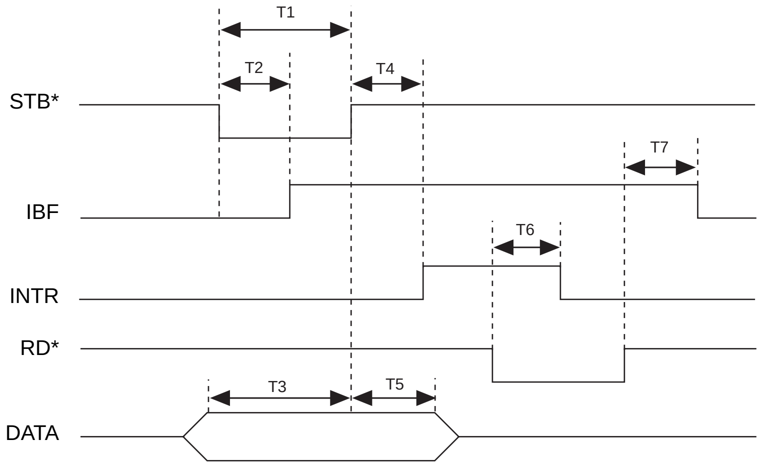
The following signals are used in the timing diagrams on the subsequent pages.

Table 4-4. Signal Descriptions

Signal	Direction	Description
STB*	Input	Strobe Input—A low signal on this handshaking line loads data into the input latch.
IBF	Output	Input Buffer Full—A high signal on this handshaking line indicates that data has been loaded into the input latch. This is an input acknowledge signal; a low signal indicates that the latch is ready for another input.
ACK*	Input	Acknowledge Input—A low signal on this handshaking line indicates that the data written to the selected port has been accepted. This signal is a response from the external device that it has received the data from the DAQCard-DIO-24.
OBF*	Output	Output Buffer Full—A low signal on this handshaking line indicates that data has been written to the selected port
INTR	Output	Interrupt Request—This signal becomes high when the DAQCard-DIO-24 requests service during a data transfer. The appropriate interrupt enable bits must be set to generate this signal.
RD*	Internal	Read Signal—When this signal is low, data is transferred from the DAQCard-DIO-24 to the CPU. This signal is generated from the control lines of the computer.
WR*	Internal	Write Signal—When this signal is low, data or control words are transferred from the CPU to the DAQCard-DIO-24. This signal is generated from the control lines of the computer.
DATA	Bidirectional	Data Lines at the Selected Port—This signal indicates when the data on the data lines at a selected port is or should be available.

Mode 1 Input Timing

Figure 4-4 illustrates the timing specifications for an input transfer in mode 1.

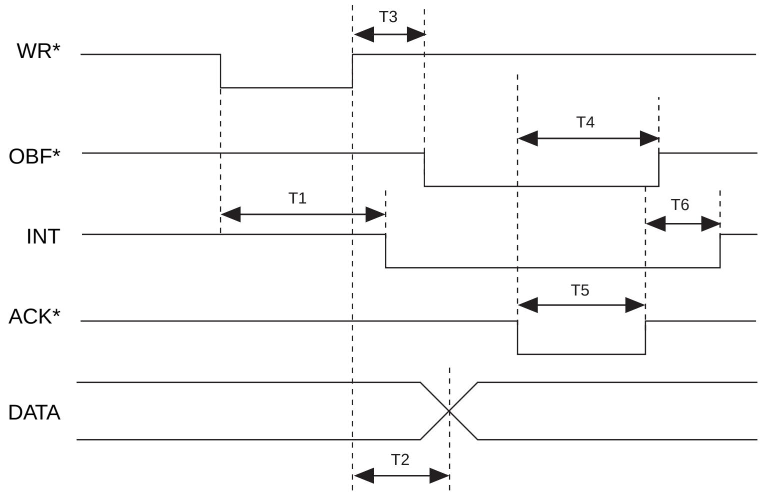


Name	Description	Minimum	Maximum
T1	STB* pulse width	100	—
T2	STB* = 0 to IBF = 1	—	150
T3	Data before STB* = 1	20	—
T4	STB* = 1 to INTR = 1	—	150
T5	Data after STB* = 1	50	—
T6	RD* = 0 to INTR = 0	—	200
T7	RD* = 1 to IBF = 0	—	150
All timing values are in nanoseconds.			

Figure 4-4. Timing Specifications for Mode 1 Input Transfer

Mode 1 Output Timing

Figure 4-5 illustrates the timing specifications for an output transfer in mode 1.

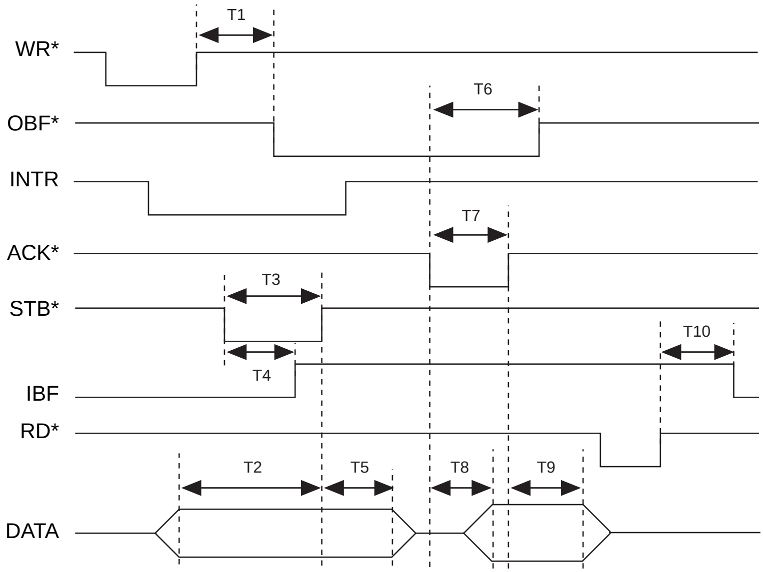


Name	Description	Minimum	Maximum
T1	WR* = 0 to INTR = 0	—	250
T2	WR* = 1 to output	—	200
T3	WR* = 1 to OBF* = 0	—	150
T4	ACK* = 0 to OBF* = 1	—	150
T5	ACK* pulse width	100	—
T6	ACK* = 1 to INTR = 1	—	150
All timing values are in nanoseconds.			

Figure 4-5. Timing Specifications for Mode 1 Output Transfer

Mode 2 Bidirectional Timing

Figure 4-6 illustrates the timing specifications for bidirectional transfers in mode 2.



Name	Description	Minimum	Maximum
T1	WR* = 0 to INTR = 0	—	150
T2	Data before STB*= 1	20	—
T3	STB* pulse width	100	—
T4	STB* = 0 to IBF = 1	—	150
T5	Data after STB* = 1	50	—
T6	ACK* = 0 to OBF = 1	—	150
T7	ACK* pulse width	100	—
T8	ACK* = 0 to output	—	150
T9	ACK* = 1 to output float	20	250
T10	RD* = 1 to IBF = 0	—	150

All timing values are in nanoseconds.

Figure 4-6. Timing Specifications for Mode 2 Bidirectional Transfer

Cabling

You can connect the DAQCard-DIO-24 to a wide range of printers, plotters, test instruments, I/O racks and modules, screw terminal panels, and almost any device with a parallel interface. The optional cable available for the DAQCard-DIO-24 has a standard 50-pin header connector. The pin assignments are compatible with standard 24-channel I/O module mounting racks (such as the SSR-24 backplane).

The CB-50 cable termination accessory is available from National Instruments for use with the DAQCard-DIO-24. The optional PSH27-50F-D1 cable available for your DAQCard-DIO-24 connects directly to the CB-50 connector block. Signal input and output wires can be attached to screw terminals on the connector block and are thereby connected to the DAQCard-DIO-24 I/O connector.

The CB-50 or CB-50LP is useful for initial prototyping of an application or in situations where DAQCard-DIO-24 interconnections are frequently changed. When a final field wiring scheme has been developed, however, you may want to develop your own cable. For information and guidelines for the design of custom cables, see the *Custom Cables* section in Chapter 1, *Introduction*.

If you plan to use the DAQCard-DIO-24 for a communications application, you may need shielded cables to meet FCC requirements. The PSH27-50F-D1 cable available for the DAQCard-DIO-24 is fully shielded. The shield is connected only at the 25-pin end.

Appendix A

Specifications

This appendix lists the specifications for the DAQCard-DIO-24. These specifications are typical at 25° C, unless otherwise stated. The operating temperature range is 0° to 70° C.

I/O Connector

I/O Signals Rating

Absolute max voltage input rating ...... 0.5 to +5.5 V with respect to GND

Input Signals

Pins 1–24

Level	Min	Max
Input logic high voltage	2.2 V	5.3 V
Input logic low voltage	–0.3 V	0.8 V
Input current (0 < V_{in} < +5 V)	–1.0 µA	1.0 µA

Output Signals

Pin 25 (at +5 V) ....1.0 A max

Pins 1–24

Level	Min	Max
Output high voltage (I_{out} = –2.5 mA)	3.7 V	5.0 V
Output low voltage (I_{out} = 2.5 mA)	0.0 V	0.4 V

Power Requirement (from PC Card I/O Channel)

Typ.. 60 mA at 5 VDC (±5%)

Max... 100 mA at 5 VDC (±5%)

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

Physical

PCMCIA type .... Type II

I/O connector .... 25-pin female PC Card I/O connector
50-pin female ribbon-cable connector on optional PSH27-50F-D1 cable

Environment

Operating temperature........................ 0° to 70° C

Storage temperature .... –55° to 150° C

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

Transfer Rates

Transfer rates are a function of the speed with which your program reads data from or writes data to the card and therefore vary with your system, software, and application. The following primary factors control DAQCard-DIO-24 transfer rates:

- Computer system performance
- Programming environment (register-level programming or NI-DAQ)
- Programming language and code efficiency
- Execution mode (foreground or background, with background execution typically using interrupts)
- Other operations in progress
- Application

For example, you can obtain higher transfer rates in a handshaking or data-transfer application, requiring an average rate, than in a pattern generation, data acquisition, or waveform generation application, requiring a constant sustainable rate.

Table A-1 shows maximum rates on two computer systems using register-level programming, with an efficient assembly language or C program, running in the foreground, with no other operations in progress. The numbers shown are average, rather than constant sustainable, rates.

The code used to make the measurements follows the table. The assembly language code was assembled as inline assembly C code using Microsoft Optimizing C Compiler, version 8.00. The C code was compiled using Microsoft Optimizing C Compiler, version 8.00.

A 486 DX machine used was an NEC UltraLite VERSA laptop.
The 486 DX2 machine used was a desktop PC using the CARDport ISA PCMCIA adapter.

Table A-1. DAQCard-DIO-24 Maximum Average Transfer Rates

Bus	CPU	CPU Speed	Assembly	C
AT (ISA16)	486 DX	33 MHz	254 kbytes/s	176 kbytes/s
AT (ISA16)	486 DX2	50 MHz	294 kbytes/s	213 kbytes/s

Assembly language code:

```
mov    cx, 64                    ; Count out 64 transfers
mov    dx, 0180h                 ; The port to access
                       loop:
lodsb                            ; Assume ds:si points to buffer of data
out    dx, al                    ; Send the data
add    dx, 0014h                 ; Add offset to base address for Ireg1
in     al, dx                    ; Dummy read from Ireg1
sub    dx, 0014h                 ; Restore base address
                                 ; The previous four lines are not
                                 ; necessary for measuring transfer rates
dec    cx                        ; Decrement the loop counter
jnz    short loop                ; See if we need to loop
```

C code:

```
address = 0x0180;               /* The port address */
ireg1address = address + 0x0014;
for (i = 0; i < 64; i++) {      /* Loop 64 times */
      outp(address, *data++);   /* Send data */
      inp(ireg1address);
}
```

Appendix B

Register-Level Programming

This appendix describes in detail the address and function of each of the DAQCard-DIO-24 control and status registers.

Register Map and Descriptions

The DAQCard-DIO-24 is a parallel digital I/O card that contains the 82C55A integrated circuit. The 82C55A is a general-purpose peripheral interface containing 24 programmable I/O pins. These pins represent three 8-bit I/O ports—port A, port B, and port C. These ports can be programmed as two handshaking ports or as two 8-bit ports and two 4-bit ports that perform simple I/O.

The three 8-bit ports of the 82C55A are divided into two groups of 12 signals—group A and group B. One 8-bit configuration (or control) word determines the mode of operation for each group. Group A control bits configure PA<0..7> and the upper 4 bits (nibble) of PC, PC<4..7>. Group B control bits configure PB<0..7> and the lower nibble of PC, PC<0..3>. These configuration bits are defined later in this appendix.

In addition to the registers on the 82C55A interface, the DAQCard-DIO-24 provides registers that select which onboard signals are capable of generating interrupts. The 82C55A interface uses two interrupt signals, INTRA and INTRB. Individual enable bits select which of these two signals can generate interrupts. Also, a master enable signal determines whether the card can actually send an interrupt request to the host computer. For more information about the configuration bits for these registers, refer to the *Interrupt Control Registers* section later in this appendix.

Register Map

Table B-1 lists the address map for the DAQCard-DIO-24.

Table B-1. DAQCard-DIO-24 Address Map

Register	Offset Address (Hex)	Size	Type
82C55A Register Group			
PORTA	00	8-bit	Read-and-write
PORTB	01	8-bit	Read-and-write
PORTC	02	8-bit	Read-and-write
CNFG	03	8-bit	Write-only
Interrupt Control Register Group			
Register 1	14	8-bit	Write-only
Register 2	15	8-bit	Write-only

To determine the actual address of these registers, add the offset shown in Table B-1 to the card base address. For information about how to determine the base address, see the *PC Card Initialization* section later in this appendix.

Register Descriptions

The following pages describe the registers on the DAQCard-DIO-24, including the 82C55A registers and each of the interrupt control registers.

The 82C55A CNFG Register

Figure B-1 shows the two control-word formats used to completely program the 82C55A. The control-word flag determines which control-word format is being programmed. When the control-word flag is 1, bits 0 through 6 determine the I/O characteristics of the 82C55A ports and the modes in which they operate (that is, mode 0, mode 1, or mode 2). When the control-word flag is 0, bits 0 through 3 set or reset a bit of port C. An X indicates that a bit can be either 0 or 1.

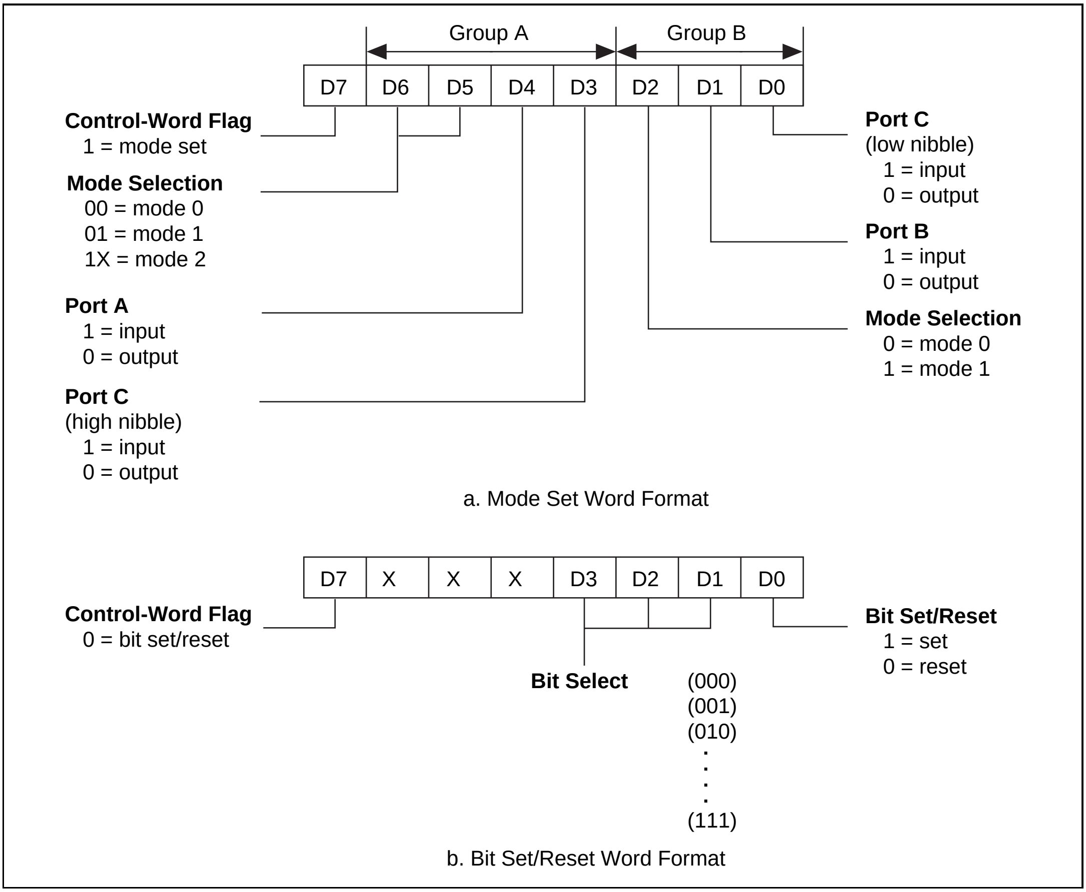


Figure B-1. Control-Word Formats

Single Bit Set/Reset Feature

Table B-2 shows the control words for setting or resetting each bit in port C. Notice that bit 7 of the control word is cleared when programming the set/reset option for the bits of port C. In this table *n* represents the bit to be set or reset.

Table B-2. Port C Set/Reset Control Words

Number	Bit Set Control Word	Bit Reset Control Word	Bit Set or Reset in Port C
0	0XXX0001	0XXX0000	XXXXXXXn
1	0XXX0011	0XXX0010	XXXXXXnX
2	0XXX0101	0XXX0100	XXXXXnXX
3	0XXX0111	0XXX0110	XXXXnXXX
4	0XXX1001	0XXX1000	XXXnXXXX
5	0XXX1011	0XXX1010	XXnXXXXX
6	0XXX1101	0XXX1100	XnXXXXXX
7	0XXX1111	0XXX1110	nXXXXXXX

Interrupt Control Registers

The DAQCard-DIO-24 has two interrupt control registers. One of these registers has individual enable bits for the two interrupt lines from the 82C55A. The other register has a master interrupt enable bit. The bit maps and signal definitions follow.

Interrupt Control Register 1

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	IRQ1	IRQ0

Bit	Name	Description
7–2	0	Reserved—These bits must be set to zero for compatibility with future versions of this card. However, older software that sets these bits to 1 will still work with the DAQCard-DIO-24.
1	IRQ1	Interrupt Selection 1—If this bit and the INTEN bit in Interrupt Control

		Register 2 are both set, the card passes INTRB signals from the 82C55A on to the host computer. If this bit is cleared, the card does not pass the INTRB signals on to the host computer, regardless of the INTEN setting. Note that the 82C55A does not generate INTRB signals unless the INTEB bit on the 82C55A is set.
0	IRQ0	Interrupt Selection 0—If this bit and the INTEN bit in Interrupt Control Register 2 are both set, the card passes INTRA signals from the 82C55A on to the host computer. If this bit is cleared, the card does not pass INTRA signals on to the host computer, regardless of the INTEN setting. Note that the 82C55A does not generate INTRA signals unless the INTEA bit (mode 1) or INTE1 or INTE2 bit (mode 2) on the 82C55A is set.

Interrupt Control Register 2

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	INTEN	0	0

Bit	Name	Description
7–3, 1, 0	0	Reserved—These bits must be set to zero for compatibility with future versions of this card. However, older software that sets these bits to 1 will still work with the DAQCard-DIO-24.
2	INTEN	Interrupt Enable—If this bit is set, the DAQCard-DIO-24 can interrupt the host computer. If this bit is cleared, the DAQCard-DIO-24 interrupt line is put into high-impedance mode, so other devices can use the interrupt line.

Programming

This section presents programming information for the DAQCard-DIO-24 and includes program examples written in C.

PC Card Initialization

Before you can access the digital I/O circuitry on the DAQCard-DIO-24, the card must be activated using Card Services. The I/O PC Cards are kept inactive until a program has requested that Card Services activate the card by assigning an interrupt level and an address space for the card I/O registers. The DAQCard-DIO-24 requires a 32-byte I/O address window and one interrupt level.

If you are using a PC, there are at least two different ways to activate the card:

- If you are using the DAQCard-DIO-24 with National Instruments software such as NI-DAQ or LabVIEW, the NI-DAQ device driver requests the card activation. For more information about this procedure, see the *Device Configuration* section in your *NI-DAQ User Manual for PC Compatibles*.
- If this option is not feasible for your application, you can develop your own program to activate the card. However, this is fairly complicated and requires significantly more programming. If you develop your own program, consult the *PC Card Standard*, *Socket Services Specifications*, which explains how to activate a card using system-level calls. Request an I/O window, an interrupt level, and a configuration. In the configuration, the configuration index should be set to 01 hex for normal operation.

If you are using a Macintosh, there are at least two different ways to activate the card:

- If you are using the DAQCard-DIO-24 with National Instruments software such as NI-DAQ or LabVIEW, the NI-DAQ device driver requests the card activation. For more information about this topic, see the *Device Configuration* section in the *NI-DAQ Software Reference Manual for Macintosh*.
- You can develop your own program to activate the card. However, this is fairly complicated, and it requires significantly more programming. If you develop your own program, consult the *PC Card Standard*, *Socket Services Specifications*, which explains how to activate a card using system-level calls, and the *PC Card*

Development Kit (available from Apple through APDA), which explains how to interface with the Apple PC Card Manager software that is part of your PC Card expansion interface. Request a configuration and an I/O window. In the configuration, the configuration index should be set to 01 hex for normal operation.

After you activate the card, you are ready to configure the DAQCard-DIO-24 for digital I/O. The following section explains how to set the registers for different operations.

Programming Considerations for the 82C55A

82C55A Modes of Operation

The three basic modes of operation for the 82C55A are as follows:

- Mode 0—Basic I/O
- Mode 1—Strobed I/O
- Mode 2—Bidirectional bus

The three ports on the 82C55A are arranged into two groups—group A and group B. Group A contains one 8-bit data port (port A) and the upper nibble of port C. Group B contains one 8-bit data port (port B) and the lower nibble of port C.

☞ **Note:** ***When group A is in mode 1 or 2, it uses one port C bit (PC3) of group B as an interrupt line.***

The 82C55A also has a single bit set/reset feature for port C. The 8-bit control word also programs this function. For additional information, refer to Appendix C, *OKI 82C55A Data Sheet*.

Mode 0

Mode 0 can be used for simple input and output operations for each of the ports. No handshaking is required; data is simply written to or read from a selected port.

Mode 0 has the following features:

- Each group contains one 8-bit data port (port A or port B) and one 4-bit data port (upper or lower nibble of port C).
- Any port can be input or output.
- Lines of ports configured for output hold their last state, but lines of ports configured for input are pulled up when they are allowed to float.

Mode 1

Mode 1 transfers data that is synchronized by handshaking signals. Ports A and B use the eight lines of port C to generate or receive the handshake signals.

Mode 1 has the following features:

- Each group contains one 8-bit data port (port A or port B) and one 4-bit control/data port (upper or lower nibble of port C). If group A is configured in mode 1, it uses the most significant bit, PC3, of the group B port C lines.
- The 8-bit data ports can be either input or output; both inputs and outputs are latched.
- The 4-bit ports are used for control and status of the 8-bit data ports.
- Interrupt generation, enable, and disable functions are available.

Mode 2

Mode 2 can be used for communication over an 8-bit bidirectional bus. Handshaking signals are used in a manner similar to mode 1. Interrupt generation, enable, and disable functions are also available. Other features of this mode include the following:

- These features are available for group A only.
- One 8-bit bidirectional bus port (port A) and a 5-bit control and status port (port C) are used.
- Group A uses the most significant bit, PC3, of the group B port C lines.
- Inputs and outputs are latched.

Single Bit Set/Reset Feature

Any of the eight bits of port C can be set or reset with one control word. This feature generates status and control for port A and port B when operating in mode 1 or mode 2.

Mode 0 Basic I/O

Mode 0 can be used for simple I/O functions for each of the three ports with no handshaking. Each port can be assigned as an input or an output port. The upper and lower nibbles of port C can be configured for different directions. The 16 possible I/O configurations are shown in Table B-3. Notice that bit 7 of the control word is set when programming the mode of operation for each port
.

Table B-3. Mode 0 I/O Configurations

	Control Word Bit	Group A		Group B	
Number	**76543210**	**Port A**	**Port C**	**Port B**[1]	**Port C**[2]
0	10000000	Output	Output	Output	Output
1	10000001	Output	Output	Output	Input
2	10000010	Output	Output	Input	Output
3	10000011	Output	Output	Input	Input
4	10001000	Output	Input	Output	Output
5	10001001	Output	Input	Output	Input
6	10001010	Output	Input	Input	Output
7	10001011	Output	Input	Input	Input
8	10010000	Input	Output	Output	Output
9	10010001	Input	Output	Output	Input
10	10010010	Input	Output	Input	Output

[1] Upper nibble of port C
[2] Lower nibble of port C

Table B-3. Mode 0 I/O Configurations (Continued)

Number	Control Word Bit 76543210	Group A Port A	Group A Port C	Group B Port B[1]	Group B Port C[2]
11	10010011	Input	Output	Input	Input
12	10011000	Input	Input	Output	Output
13	10011001	Input	Input	Output	Input
14	10011010	Input	Input	Input	Output
15	10011011	Input	Input	Input	Input

[1] Upper nibble of port C
[2] Lower nibble of port C

Mode 0 Programming Example

The following example shows how to configure the 82C55A for various combinations of mode 0 input and output. This code is strictly an example and is not intended to be used without modification in a practical situation. The base address used may not correspond to the base address of the card in your system. For information on card activation, base addressing, and interrupt levels, see *PC Card Initialization* earlier in this appendix.

```
Main() {

/* Set PC to 1 if you are using a PC, or set MAC to 1 if you are using a
Mac.*/

#define      PC              0
#define      MAC             0

/* If MAC = 1, set base address and define rd and wrt. */
#if          MAC
#define      BASE_ADDRESS    0xa0000000L
#define      rd(a)     ((unsigned char) *((unsigned char *) (a)))
#define      wrt(a,d)  (*((unsigned char *) (a)) = ((unsigned char) (d)))
```

```
/* Else if PC = 1, set base address and define rd and wrt. */
#elif       PC
#define     BASE_ADDRESS    0x0220
#define     rd(a)    ((unsigned char) inp(((unsigned int) (a))))
#define     wrt(a,d)  (outp(((unsigned int) (a)), ((unsigned char) (d))))
#end


#define PORTAoffset    0x00  /* Offset for port A */
#define PORTBoffset    0x01  /* Offset for port B */
#define PORTCoffset    0x02  /* Offset for port C */
#define CNFGoffset     0x03  /* Offset for CNFG */

unsigned long porta, portb, portc, cnfg
char valread;          /* Variable to store data read from a port*/

/* Calculate register addresses. */
porta = BASE_ADDRESS + PORTAoffset;
portb = BASE_ADDRESS + PORTBoffset;
portc = BASE_ADDRESS + PORTCoffset;
cnfg = BASE_ADDRESS + CNFGoffset;

/* EXAMPLE 1*/
wrt(cnfg,0x80);        /* Ports A, B, and C are outputs. */
wrt(porta,0x12);       /* Write data to port A. */
wrt(portb,0x34);       /* Write data to port B. */
wrt(portc,0x56);       /* Write data to port C. */

/* EXAMPLE 2*/
wrt(cnfg,0x90);        /* Port A is input; ports B and C */
                       /* are outputs. */
wrt(portb,0x22);       /* Write data to port B. */
wrt(portc,0x55);       /* Write data to port C. */
valread = rd(porta);   /* Read data from port A. */

/* EXAMPLE 3 */
wrt(cnfg,0x82);        /* Ports A and C are outputs; port B */
                       /* is an input. */

/* EXAMPLE 4 */
wrt(cnfg,0x89);        /* Ports A and B are outputs; port C */
                       /* is an input. */

}
```

Mode 1 Strobed Input

In mode 1, each group contains one 8-bit port and one 4-bit control/data port. The 8-bit port can be either an input or an output port. The 4-bit port is used for control and status information for the 8-bit port. The transfer of data is synchronized by handshaking signals in the 4-bit port.

The control word written to the CNFG Register to configure port A for input in mode 1 is shown as follows. Bits PC6 and PC7 of port C can be used as extra input or output lines.

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

Port C bits PC6 and PC7
1 = input
0 = output

The control word written to the CNFG Register to configure port B for input in mode 1 is shown as follows.

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

During a mode 1 data read transfer, the status of the handshaking lines and interrupt signals can be obtained by reading port C. The port C status-word bit definitions for an input transfer are shown as follows.

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
I/O	I/O	IBFA	INTEA	INTRA	INTEB	IBFB	INTRB

Bit	**Name**	**Description**
7–6	I/O	Input/Output—Extra I/O status lines when port A is in mode 1 input.
5	IBFA	Input Buffer Full for Port A—High indicates that data has been loaded into the input latch for port A.

4	INTEA	Interrupt Enable Bit for Port A—Enables interrupts from the 82C55A for port A. Controlled by bit set/reset of PC4.
3	INTRA	Interrupt Request Status for Port A—When INTEA is high and IBFA is high, this bit is high, indicating that an interrupt request is asserted.
2	INTEB	Interrupt Enable Bit for Port B—Enables interrupts from the 82C55A for port B. Controlled by bit set/reset of PC2.
1	IBFB	Input Buffer Full for Port B—High indicates that data has been loaded into the input latch for port B.
0	INTRB	Interrupt Request Status for Port B—When INTEB is high and IBFB is high, this bit is high, indicating that an interrupt request is asserted.

At the digital I/O connector, port C has the following pin assignments when in mode 1 input. Notice in Figure B-2 that the status of STBA* and STBB* is not included in the port C status word.

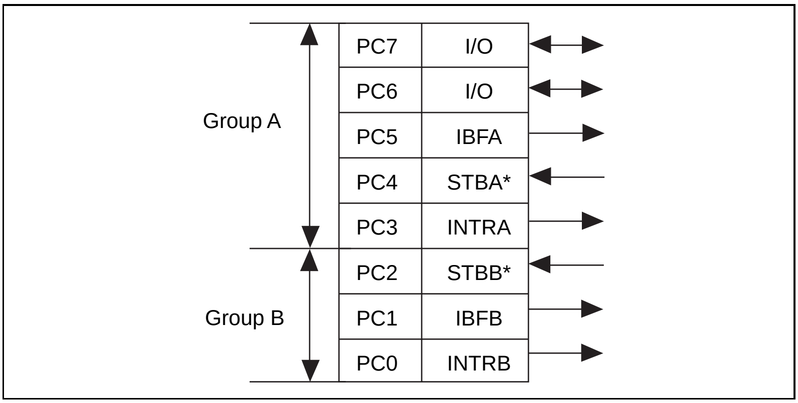


Figure B-2. Port C Pin Assignments in Mode 1 Input

Mode 1 Input Programming Example

The following example shows how to configure the 82C55A for various combinations of mode 1 input. This code is strictly an example and is not intended to be used without modification in a practical situation. The base address used may not correspond to the base address of the card in your system. For information on card activation, base addressing, and interrupt levels, see *PC Card Initialization* earlier in this appendix.

```
 Main() {

/* Set PC to 1 if you are using a PC, or set MAC to 1 if you are using a
Mac.*/

#define      PC              0
#define      MAC             0

/* If MAC = 1, set base address and define rd and wrt. */
#if          MAC
#define      BASE_ADDRESS    0xa0000000L
#define      rd(a)     ((unsigned char) *((unsigned char *) (a)))
#define      wrt(a,d)  (*((unsigned char *) (a)) = ((unsigned char) (d)))

/* Else if PC = 1, set base address and define rd and wrt. */
#elif        PC
#define      BASE_ADDRESS    0x0220
#define      rd(a)     ((unsigned char) inp(((unsigned int) (a))))
#define      wrt(a,d)  (outp(((unsigned int) (a)), ((unsigned char) (d))))

#end


#define PORTAoffset    0x00   /* Offset for port A */
#define PORTBoffset    0x01   /* Offset for port B */
#define PORTCoffset    0x02   /* Offset for port C */
#define CNFGoffset     0x03   /* Offset for CNFG */
#define IREG1offset    0x14   /* Offset for Interrupt Reg. 1*/

unsigned long porta, portb, portc, cnfg, ireg1;
char valread;/* Variable to store data read from a port */

/* Calculate register addresses. */
porta = BASE_ADDRESS + PORTAoffset;
portb = BASE_ADDRESS + PORTBoffset;
portc = BASE_ADDRESS + PORTCoffset;
cnfg = BASE_ADDRESS + CNFGoffset;
ireg1 = BASE_ADDRESS + IREG1offset;
```

```
/* EXAMPLE 1–Port A input */
wrt(cnfg,0xB0);             /* Port A is an input in mode 1. */
while (!(rd(portc) & 0x20))  /* Wait until IBFA is set, indicating that */
      rd(ireg1);            /* data has been loaded in port A. */
valread = rd(porta);        /* Read the data from port A. */

/* EXAMPLE 2–Port B input */
wrt(cnfg,0x86);             /* Port B is an input in mode 1. */
while (!(rd(portc) & 0x02))  /* Wait until IBFB is set, indicating that */
      rd(ireg1);            /* data has been loaded in port B. */

valread = rd(portb);

}
```

Mode 1 Strobed Output

The control word written to the CNFG Register to configure port A for output in mode 1 is shown as follows. Bits PC4 and PC5 of port C can be used as extra input or output lines when port A uses mode 1 output.

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

Port C bits PC4 and PC5
1 = input
0 = output

The control word written to the CNFG Register to configure port B for output in mode 1 is shown as follows.

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

During a mode 1 data write transfer, the status of the handshaking lines and interrupt signals can be obtained by reading port C. Notice that the bit definitions are different for a write and a read transfer.

The following are the port C status-word bit definitions for output (port A and port B).

7	6	5	4	3	2	1	0
OBFA*	INTEA	I/O	I/O	INTRA	INTEB	OBFB*	INTRB

Bit	Name	Description
7	OBFA*	Output Buffer Full for Port A—Low indicates that the CPU has written data to port A.
6	INTEA	Interrupt Enable Bit for Port A—If this bit is high, interrupts are enabled from the 82C55A for port A. Controlled by bit set/reset of PC6.
5–4	I/O	Input/Output—Extra I/O status lines when port A is in mode 1 output.
3	INTRA	Interrupt Request Status for Port A—When INTEA is high and OBFA* is high, this bit is high, indicating an asserted interrupt request.
2	INTEB	Interrupt Enable Bit for Port B—If this bit is high, interrupts are enabled from the 82C55A for port B. Controlled by bit set/reset of PC2.
1	OBFB*	Output Buffer Full for Port B—Low indicates that the CPU has written data to port B.
0	INTRB	Interrupt Request Status for Port B—When INTEB is high and OBFB* is high, this bit is high, indicating an asserted interrupt request.

At the digital I/O connector, port C has the following pin assignments when in mode 1 output. Notice in Figure B-3 that the status of ACKA* and ACKB* is not included when port C is read.

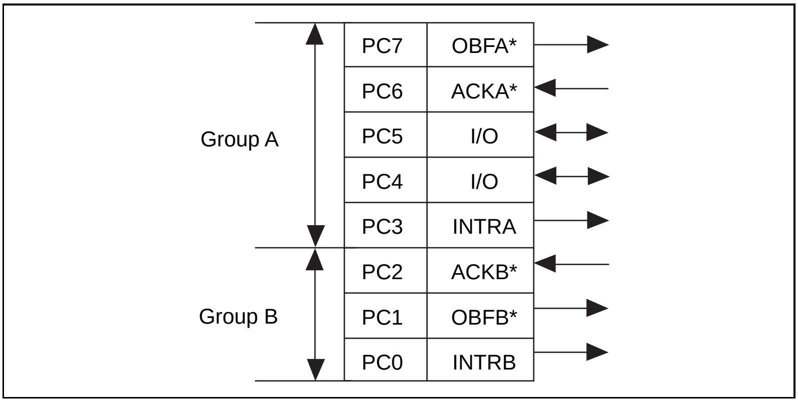


Figure B-3. Port C Pin Assignments in Mode 1 Output

Mode 1 Output Programming Example

The following example shows how to configure the 82C55A for various combinations of mode 1 output. This code is strictly an example and is not intended to be used without modification in a practical situation. The base address used may not correspond to the base address of the card in your system. For information on card activation, base addressing, and interrupt levels, see *PC Card Initialization* earlier in this appendix.

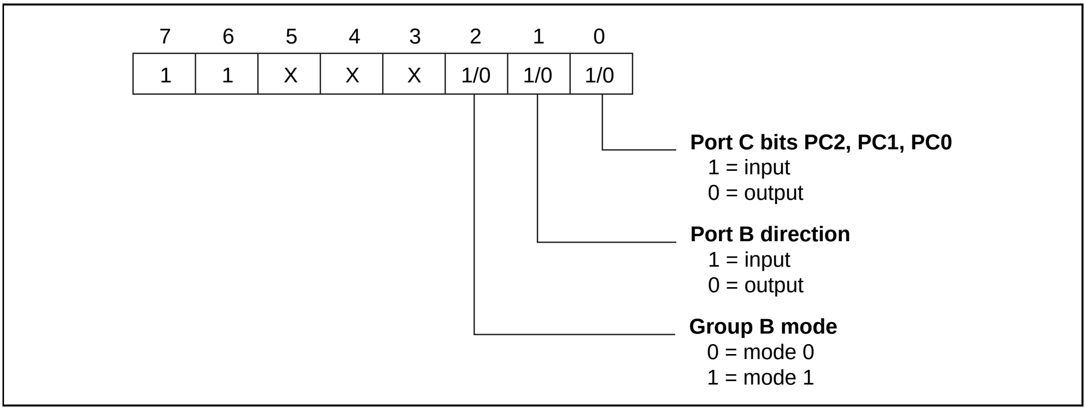
```
 Main() {
/* Set PC to 1 if you are using a PC, or set MAC to 1 if you are using a
Mac.*/
#define      PC              0
#define      MAC             0

/* If MAC = 1, set base address and define rd and wrt. */
#if          MAC
#define      BASE_ADDRESS    0xa0000000L
#define      rd(a)    ((unsigned char) *((unsigned char *) (a)))
#define      wrt(a,d)  (*((unsigned char *) (a)) = ((unsigned char) (d)))

/* Else if PC = 1, set base address and define rd and wrt. */
#elif        PC
#define      BASE_ADDRESS    0x0220
#define      rd(a)    ((unsigned char) inp(((unsigned int) (a))))
#define      wrt(a,d)  (outp(((unsigned int) (a)), ((unsigned char) (d))))

#end
```

```
#define PORTAoffset    0x00   /* Offset for port A */
#define PORTBoffset    0x01   /* Offset for port B */
#define PORTCoffset    0x02   /* Offset for port C */
#define CNFGoffset     0x03   /* Offset for CNFG */
#define IREG1offset    0x14   /* Offset for Interrupt Reg. 1*/

unsigned long porta, portb, portc, cnfg, ireg1;
char valread;                 /* Variable to store data read from a port */

/* Calculate register addresses. */
porta = BASE_ADDRESS + PORTAoffset;
portb = BASE_ADDRESS + PORTBoffset;
portc = BASE_ADDRESS + PORTCoffset;
cnfg = BASE_ADDRESS + CNFGoffset;
ireg1 = BASE_ADDRESS + IREG1offset;

/* EXAMPLE 1–Port A output */
wrt(cnfg,0xA0);               /* Port A is an output in mode 1.*/
while (!(rd(portc) & 0x80))   /* Wait until OBFA* is set, indicating */
      rd(ireg1);              /* that the data last written to port A*/
                              /* has been read. */
wrt(porta,0x12);              /* Write data to port A. */

/* EXAMPLE 2–Port B output */
wrt(cnfg,0x84);               /* Port B is an output in mode 1.*/
while (!(rd(portc) & 0x02))   /* Wait until OBFB* is set, indicating */
      rd(ireg1);              /* that the data last written to port B */
                              /* has been read. */
wrt(portb,0x34);              /* Write the data to port B. */

}
```

Mode 2 Bidirectional Bus

Mode 2 has an 8-bit bus that can transfer both input and output without changing the configuration. The data transfers are synchronized with handshaking lines in port C. This mode uses only port A; however, port B can be used in either mode 0 or mode 1 while port A is configured for mode 2.

The control word written to the CNFG Register to configure port A as a bidirectional data bus in mode 2 is shown as follows. If port B is configured for mode 0, then PC<2..0> of port C can be used as extra input or output lines.



During a mode 2 data transfer, the status of the handshaking lines and interrupt signals can be obtained by reading port C. The port C status-word bit definitions for a mode 2 transfer are shown as follows.

The following are port C status-word bit definitions for bidirectional bus data path (port A only).

7	6	5	4	3	2	1	0
OBFA*	INTE1	IBFA	INTE2	INTRA	I/O or port B handshaking		

Bit	**Name**	**Description**
7	OBFA*	Output Buffer Full for Port A—Low indicates that the CPU has written data to port A.
6	INTE1	Interrupt Enable Bit for Output—If this bit is set, interrupts are enabled from the 82C55A for OBFA*. Controlled by bit set/reset of PC6.
5	IBFA	Input Buffer Full for Port A—High indicates that data has been loaded into the input latch for port A.

4	INTE2	Interrupt Enable Bit for Input—If this bit is set, interrupts are enabled from the 82C55A for IBFA. Controlled by bit set/reset of PC4.
3	INTRA	Interrupt Request Status—If INTE2 is high and IBFA is high, this bit is high, indicating that an interrupt request is asserted for input transfers. If INTE1 is high and OBFA* is high, this bit is high, indicating that an interrupt request is asserted for output transfers.
2–0	I/O or port B	Input/Output—Extra I/O status lines available if port B is not configured for handshaking mode 1.

Figure B-4 shows the pin assignments for port C when in mode 2 at the digital I/O connector.

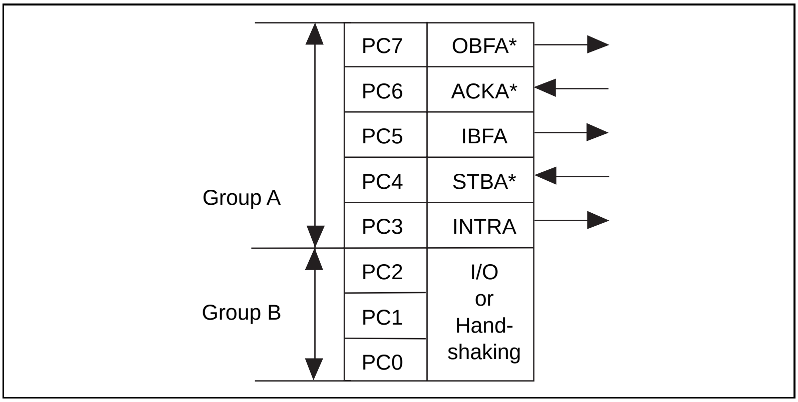


Figure B-4. Port C Pin Assignments in Mode 2

Mode 2 Programming Example

The following example shows how to configure the 82C55A for various combinations of mode 2 input and output and how to use the handshaking signals to control data flow. This code is strictly an example and is not intended to be used without modification in a practical situation. The base address used may not correspond to the base address of the card in your system. For information on card activation, base addressing, and interrupt levels, see *PC Card Initialization* earlier in this appendix.

```
 Main() {

/* Set PC to 1 if you are using a PC, or set MAC to 1 if you are using a
Mac.*/

#define      PC               0
#define      MAC              0

/* If MAC = 1, set base address and define rd and wrt. */
#if          MAC
#define      BASE_ADDRESS    0xa0000000L
#define      rd(a)     ((unsigned char) *((unsigned char *) (a)))
#define      wrt(a,d)  (*((unsigned char *) (a)) = ((unsigned char) (d)))

/* Else if PC = 1, set base address and define rd and wrt. */
#elif        PC
#define      BASE_ADDRESS    0x0220
#define      rd(a)     ((unsigned char) inp(((unsigned int) (a))))
#define      wrt(a,d)  (outp(((unsigned int) (a)), ((unsigned char) (d))))
#end

#define PORTAoffset    0x00   /* Offset for port A */
#define PORTBoffset    0x01   /* Offset for port B */
#define PORTCoffset    0x02   /* Offset for port C */
#define CNFGoffset     0x03   /* Offset for CNFG */
#define IREG1offset    0x14   /* Offset for Interrupt Reg. 1*/

unsigned long porta, portb, portc, cnfg, ireg1;
char valread;                 /* Variable to store data read from a port */

/* Calculate register addresses. */
porta = BASE_ADDRESS + PORTAoffset;
portb = BASE_ADDRESS + PORTBoffset;
portc = BASE_ADDRESS + PORTCoffset;
cnfg = BASE_ADDRESS + CNFGoffset;
ireg1 = BASE_ADDRESS + IREG1offset;
```

```
/* EXAMPLE 1*/
wrt(cnfg,0xC0);              /* Port A is in mode 2. */
while (!(rd(portc) & 0x80))  /* Wait until OBFA* is set, indicating */
      rd(ireg1);             /* that the data last written to port A has */
                             /* been read. */
wrt(porta,0x67);             /* Write the data to port A. */
while (!(rd(portc) & 0x20))  /* Wait until IBFA is set, indicating */
      rd(ireg1);             /* that data is available in port A to */
                             /* be read. */

valread = rd(porta);         /* Read data from port A. */

}
```

Interrupt Programming Examples

The following examples show the process required to enable interrupts for several different operating modes. The interrupt handling routines and interrupt installation routines are not included. Also, if you generate interrupts with the PC3 or PC0 lines of the 82C55A, you must maintain the active high level until the interrupt service routine is entered. Otherwise, the host computer considers the interrupt a spurious interrupt and routes the request to the channel responsible for handling spurious interrupts. To prevent this problem, try using some other I/O bit to send feedback to the device generating the interrupt. In this way, the interrupting device can be signaled that the interrupt service routine has been entered. For further information on using PC3 and PC0 for interrupts, see the *Interrupt Handling* section later in this appendix. The base address used may not correspond to the base address of the card in your system. For information on card activation, base addresses, and interrupt levels, see *PC Card Initialization* earlier in this appendix.

```
 Main() {

/* Set PC to 1 if you are using a PC, or set MAC to 1 if you are using a
Mac.*/

#define      PC             0
#define      MAC            0

/* If MAC = 1, set base address and define rd and wrt. */
#if          MAC
#define      BASE_ADDRESS    0xa0000000L
#define      rd(a)     ((unsigned char) *((unsigned char *) (a)))
#define      wrt(a,d)  (*((unsigned char *) (a)) = ((unsigned char) (d)))
```

```
/* Else if PC = 1, set base address and define rd and wrt. */
#elif        PC
#define      BASE_ADDRESS    0x0220
#define      rd(a)     ((unsigned char) inp(((unsigned int) (a))))
#define      wrt(a,d)  (outp(((unsigned int) (a)), ((unsigned char) (d))))
#end
#define PORTAoffset    0x00   /* Offset for port A */
#define PORTBoffset    0x01   /* Offset for port B */
#define PORTCoffset    0x02   /* Offset for port C */
#define CNFGoffset     0x03   /* Offset for CNFG */
#define IREG1offset    0x14   /* Offset for Interrupt Reg. 1*/
#define IREG2offset    0x15   /* Offset for Interrupt Reg. 2*/
unsigned long porta, portb, portc, cnfg, ireg1, ireg2;
char valread;                 /* Variable to store data read from a port */
/* Calculate register addresses. */
porta = BASE_ADDRESS + PORTAoffset;
portb = BASE_ADDRESS + PORTBoffset;
portc = BASE_ADDRESS + PORTCoffset;
cnfg = BASE_ADDRESS + CNFGoffset;
ireg1 = BASE_ADDRESS + IREG1offset;
ireg2 = BASE_ADDRESS + IREG2offset;
/* EXAMPLE 1–Set up interrupts for mode 1 input for port A. Enable the
appropriate interrupt bits. */
wrt(cnfg,0xB0);               /* Port A is an input in mode 1. */
wrt(ireg1,0x01);              /* Set IRQ0 to enable port A interrupts. */
wrt(cnfg,0x09);               /* Set PC4 to enable interrupts from */
                              /* 82C55A. */
wrt(ireg2,0x04);              /* Set INTEN bit. */
/* EXAMPLE 2–Set up interrupts for mode 1 input for port B. Enable the
appropriate interrupt bits. */
wrt(cnfg,0x86);               /* Port B is an input in mode 1. */
wrt(ireg1,0x02);              /* Set IRQ1 to enable port B interrupts.*/
wrt(cnfg,0x05);               /* Set PC2 to enable interrupts from */
                              /* 82C55A. */
wrt(ireg2,0x04);              /* Set INTEN bit. */
```

```
/* EXAMPLE 3–Set up interrupts for mode 1 output for port A. Enable the
appropriate interrupt bits. */
wrt(cnfg,0xA0);              /* Port A is an output in mode 1. */
wrt(ireg1,0x01);             /* Set IRQ0 to enable port A interrupts.*/
wrt(cnfg,0x0D);              /* Set PC6 to enable interrupts from */
                             /* 82C55A. */
wrt(ireg2,0x04);             /* Set INTEN bit. */

/* EXAMPLE 4–Set up interrupts for mode 1 output for port B. Enable the
appropriate interrupt bits. */
wrt(cnfg,0x84);              /* Port B is an output in mode 1. */
wrt(ireg1,0x02);             /* Set IRQ1 to enable port B interrupts. */
wrt(cnfg,0x05);              /* Set PC2 to enable interrupts from */
                             /* 82C55A. */
wrt(ireg2,0x04);             /* Set INTEN bit. */

/* EXAMPLE 5–Set up interrupts for mode 2 output transfers. Enable the
appropriate interrupt bits. */
wrt(cnfg,0xC0);              /* Mode 2 output */
wrt(ireg1,0x01);             /* Set IRQ0 to enable port A interrupts. */
wrt(cnfg,0x0D);              /* Set PC6 to enable interrupts from */
                             /* 82C55A. */
wrt(ireg2,0x04);             /* Set INTEN bit. */

/* EXAMPLE 6–Set up interrupts for mode 2 input transfers. Enable the
appropriate interrupt bits. */
wrt(cnfg,0xD0);              /* Mode 2 input */
wrt(ireg1,0x01);             /* Set IRQ0 to enable port A interrupts. */
wrt(cnfg,0x09);              /* Set PC4 to enable interrupts from */
                             /* 82C55A. */
wrt(ireg2,0x04);             /* Set INTEN bit. */

}
```

Interrupt Handling

To enable interrupts from the DAQCard-DIO-24, set the INTEN bit of Interrupt Register 2. First, clear this bit to disable unwanted interrupts. After all sources of interrupts have been disabled or placed in an inactive state, set INTEN.

To interrupt the host computer using the 82C55A, program the 82C55A for the I/O mode desired. In mode 1, set either the INTEA or the INTEB bit to enable interrupts from port A or port B, respectively. In mode 2, set either INTE1 or INTE2 for interrupts on output or input transfers, respectively. The INTE1 and INTE2 interrupt outputs cascade into a single interrupt output for port A. After enabling interrupts from the 82C55A, set the appropriate enable bit or bits in Interrupt Register 1; for example, if you selected both mode 2 interrupts from the 82C55A, you would set IRQ0 in order to interrupt the host computer.

External signals can be used to interrupt the DAQCard-DIO-24 when port A or port B is in mode 0 and the low nibble of port C is configured for input. If port A is in mode 0, use PC3 to generate an interrupt; if port B is in mode 0, use PC0 to generate an interrupt. When you have configured the 82C55A, you must set the corresponding interrupt enable bit in Interrupt Register 1. If you are using PC3, set IRQ0; if you are using PC0, set IRQ1. When the external signal becomes logic high, an interrupt request occurs.

Although the host computer's interrupt-monitoring circuitry is triggered by the positive-going edge of the interrupt signal, the signal must remain high until the interrupt routine is entered and interrupts are masked out. Make sure your external interrupt signal meets these qualifications. To disable the external interrupt, clear the appropriate IRQx bit or clear the INTEN bit.

OKI 82C55A Data Sheet*

Appendix C

This appendix contains the manufacturer data sheet for the OKI Semiconductor 82C55A CMOS programmable peripheral interface (PPI). This interface is used on the DAQCard-DIO-24.

OKI Semiconductor Data Book *Microprocessor*, Seventh Edition, March 1993.

OKI semiconductor

MSM82C55A-2RS/GS/VJS

CMOS PROGRAMMABLE PERIPHERAL INTERFACE

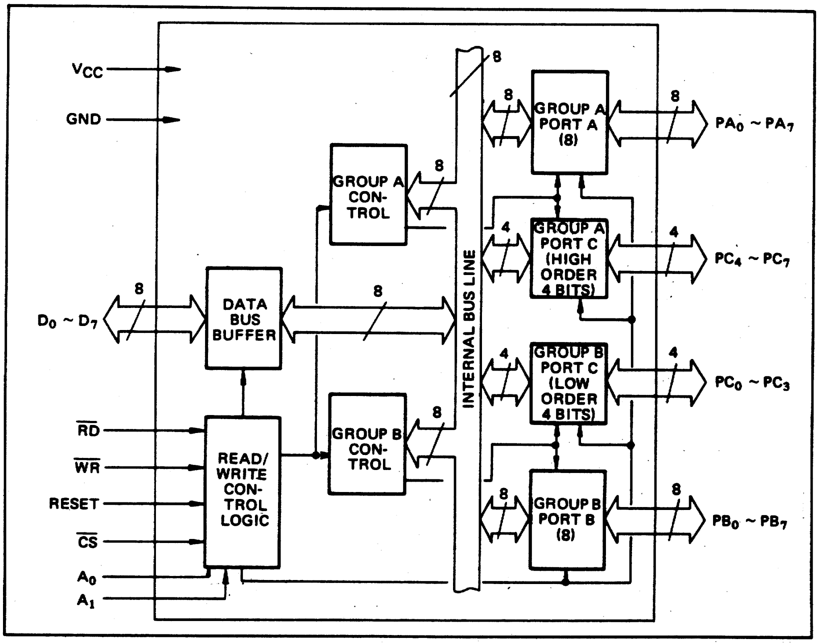
GENERAL DESCRIPTION

The MSM82C55A is a programmable universal I/O interface device which operates as high speed and on low power consumption due to 3 μ silicon gate CMOS technology. It is the best fit as an I/O port in a system which employs the 8-bit parallel processing MSM80C85A CPU. This device has 24-bit I/O pins equivalent to three 8-bit I/O ports and all inputs/outputs are TTL interface compatible.

FEATURES

- High speed and low power consumption due to 3 μ silicon gate CMOS technology
- 3 V to 6 V single power supply
- Full static operation
- Programmable 24-bit I/O ports
- Bidirectional bus operation (Port A)
- Bit set/reset function (Port C)
- TTL compatible
- Compatible with 8255A-5
- 40 pin Plastic DIP (DIP40-P-600)
- 44 pin PLCC (QFJ44-P-S650)
- 44 pin-V Plastic QFP (QFP44-P-910-VK)
- 44 pin-VI Plastic QFP (QFP44-P-910-VIK)

CIRCUIT CONFIGURATION

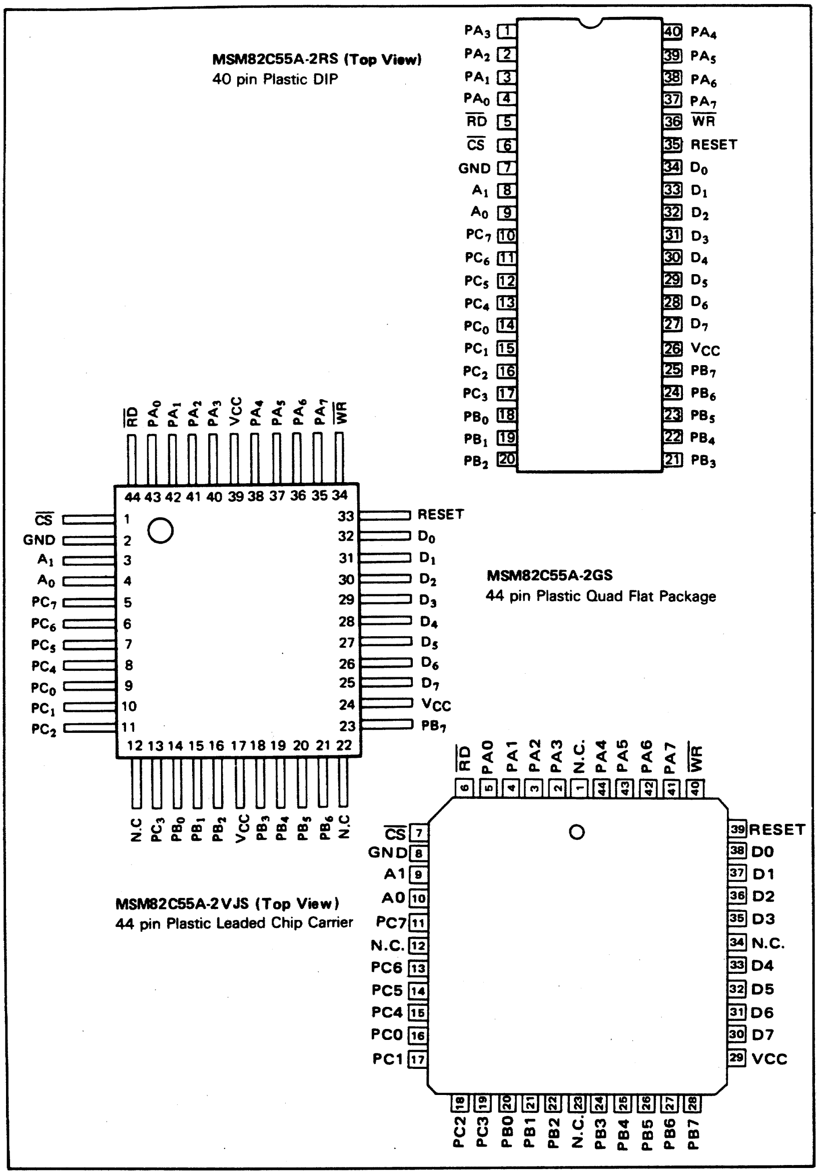


5

329

■ I/O·MSM82C55A-2RS/GS/VJS

PIN CONFIGURATION



5

330

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits			Unit
			MSM82C55A-2RS	MSM82C55A-2GS	MSM82C55A-2VJS	
Ssupply Voltage	V_{CC}	Ta = 25°C with respect to GND	−0.5 to +7			V
Input Voltage	V_{IN}		−0.5 to V_{CC} + 0.5			V
Output Voltage	V_{OUT}		−0.5 to V_{CC} + 0.5			V
Storage Temperature	T_{stg}	–	−55 to +150			°C
Power Dissipation	P_D	Ta = 25°C	1.0	0.7	1.0	W

OPERATING RANGE

Parameter	Symbol	Limits	Unit
Supply Voltage	V_{CC}	3 to 6	V
Operating Temperature	T_{OP}	−40 to 85	°C

RECOMMENDED OPERATING RANGE

Parameter	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	V_{CC}	4.5	5	5.5	V
Operating Temperature	T_{OP}	−40	+25	+85	°C
"L" Input Voltage	V_{IL}	−0.3		+0.8	V
"H" Input Voltage	V_{IH}	2.2		V_{CC} +0.3	V

DC CHARACTERISTICS

Parameter	Symbol	Conditions		MSM82C55A-2			Unit
				Min.	Typ.	Max.	
"L" Output Voltage	V_{OL}	I_{OL} = 2.5 mA	V_{CC} = 4.5V to 5.5V Ta = −40°C to +85°C (C_L = 0pF)			0.4	V
"H" Output Voltage	V_{OH}	I_{OH} = −40 μA		4.2			V
		I_{OH} = −2.5 mA		3.7			V
Input Leak Current	I_{LI}	$0 \leq V_{IN} \leq V_{CC}$		-1		1	μA
Output Leak Current	I_{LO}	$0 \leq V_{OUT} \leq V_{CC}$		-10		10	μA
Supply Current (standby)	I_{CCS}	$\overline{CS} \geq V_{CC}$ −0.2V $V_{IH} \geq V_{CC}$ −0.2V $V_{IL} \leq$ 0.2V			0.1	10	μA
Average Supply Current (active)	I_{CC}	I/O wire cycle 82C55A-2 ... 8MHzCPU timing				8	mA

5

331

■ I/O·MSM82C55A-2RS/GS/VJS ■

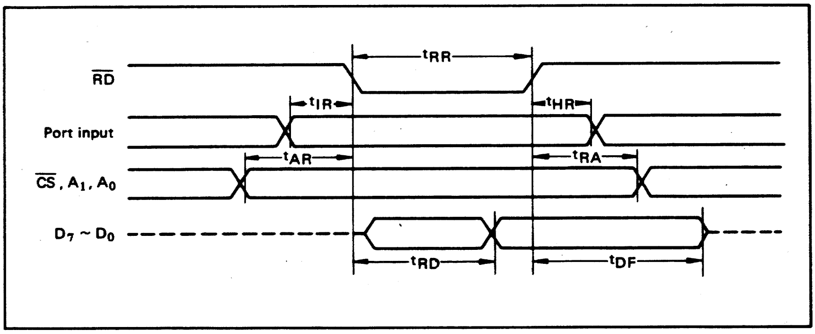
AC CHARACTERISTICS

(V_{CC} = 4.5 to 5.5V, Ta = −40 to +80°C)

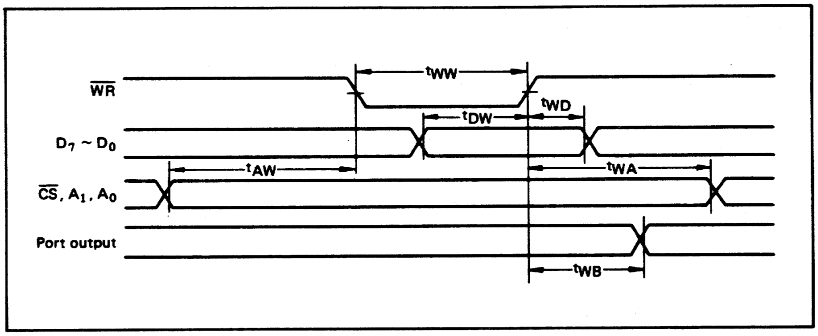
Parameter	Symbol	MSM82C55A-2		Unit	Remarks
		Min.	Max.		
Setup Time of address to the falling edge of $\overline{RD}$	t_{AR}	20		ns	
Hold Time of address to the rising edge of $\overline{RD}$	t_{RA}	0		ns	
$\overline{RD}$ Pulse Width	t_{RR}	100		ns	
Delay Time from the falling edge of $\overline{RD}$ to the output of defined data	t_{RD}		120	ns	
Delay Time from the rising edge of $\overline{RD}$ to the floating of data bus	t_{DF}	10	75	ns	
Time from the rising edge of $\overline{RD}$ or $\overline{WR}$ to the next falling edge of $\overline{RD}$ or $\overline{WR}$	t_{RV}	200		ns	
Setup Time of address before the falling edge of $\overline{WR}$	t_{AW}	0		ns	
Hold Time of address after the rising edge or $\overline{WR}$	t_{WA}	20		ns	
$\overline{WR}$ Pulse Width	t_{WW}	150		ns	
Setup Time of bus data before the rising edge of $\overline{WR}$	t_{DW}	50		ns	
Holt Time of bus data after the rising edge of $\overline{WR}$	t_{WD}	30		ns	
Delay Time from the rising edge of $\overline{WR}$ to the output of defined data	t_{WB}		200	ns	
Setup Time of port data before the falling edge of $\overline{RD}$	t_{IR}	20		ns	
Hold Time of port data after the rising edge of $\overline{RD}$	t_{HR}	10		ns	
$\overline{ACK}$ Pulse Width	t_{AK}	100		ns	
$\overline{STB}$ Pulse Width	t_{ST}	100		ns	Load 150 pF
Setup Time of port data before the rising edge of $\overline{STB}$	t_{PS}	20		ns	
Hold Time of port data after the rising edge of $\overline{STB}$	t_{PH}	50		ns	
Delay Time from the falling edge of $\overline{ACK}$ to the output of defined data	t_{AD}		150	ns	
Delay Time from the rising edge of $\overline{ACK}$ to the floating of port (Port A in mode 2)	t_{KD}	20	250	ns	
Delay Time from the rising edge of WR to the falling edge of $\overline{OBF}$	t_{WOB}		150	ns	
Delay Time from the falling edge of ACK to the rising edge of $\overline{OBF}$	t_{AOB}		150	ns	
Delay Time from the falling edge of STB to the rising edge of IBF	t_{SIB}		150	ns	
Delay Time from the rising edge of $\overline{RD}$ to the falling edge of IBF	t_{RIB}		150	ns	
Delay Time from the falling edge of $\overline{RD}$ to the falling edge of INTR	t_{RIT}		200	ns	
Delay Time from the rising edge of $\overline{STB}$ to the rising edge of INTR	t_{SIT}		150	ns	
Delay Time from the rising edge of $\overline{ACK}$ to the rising edge of INTR	t_{AIT}		150	ns	
Delay Time from the falling edge of $\overline{WR}$ to the falling edge of INTR	t_{WIT}		250	ns	

Note: Timing is measured at V_L = 0.8 V and V_H = 2.2 V for both input and outputs.

Basic Input Operation (Mode 0)

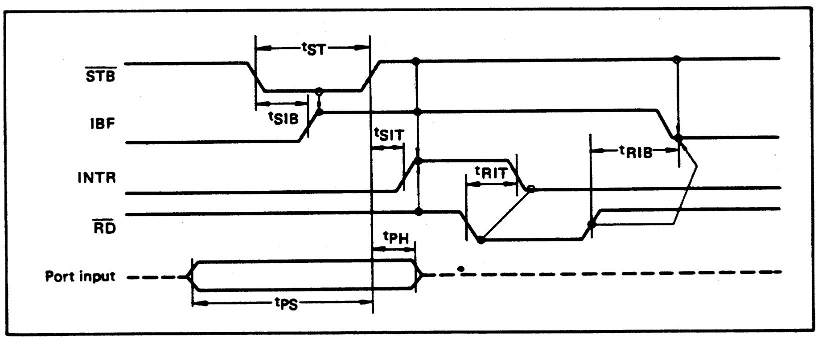


Basic Output Operation (Mode 0)



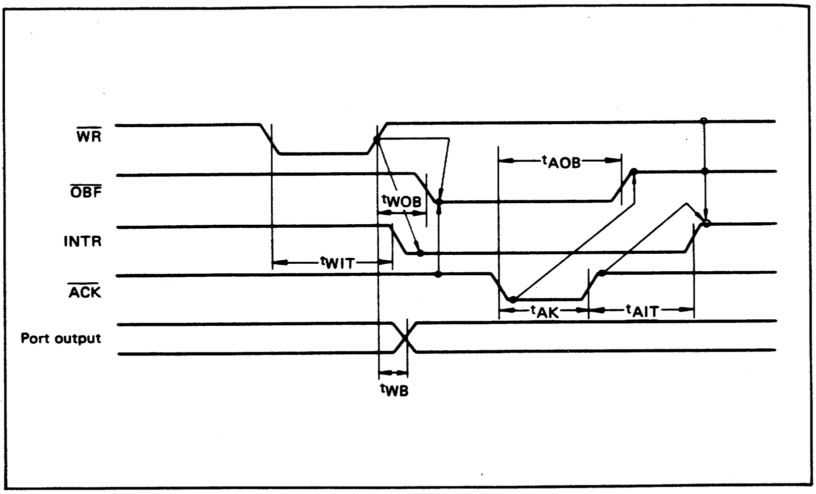
5

Strobe Input Operation (Mode 1)

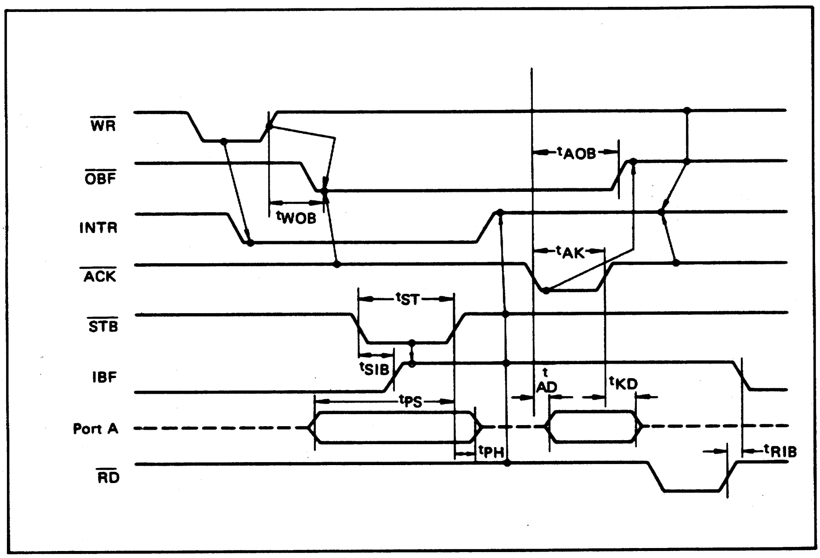


■ I/O·MSM82C55A-2RS/GS/VJS ■

Strobe Output Operation (Mode 1)



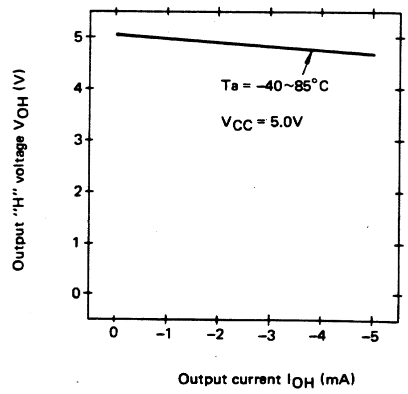
Bidirectional Bus Operation (Mode 2)



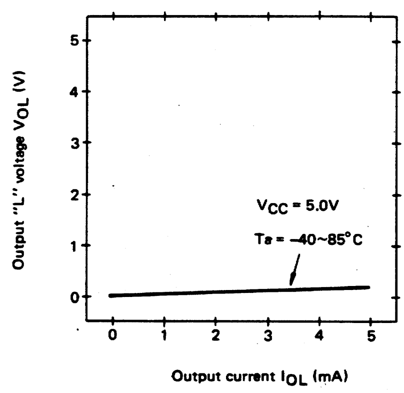
5

OUTPUT CHARACTERISTICS (REFERENCE VALUE)

1 Output "H" Voltage (V_{OH}) vs. Output Current (I_{OH})



2 Output "L" Voltage (V_{OL}) vs. Output Current (I_{OL})



Note: The direction of flowing into the device is taken as positive for the output current.

■ I/O·MSM82C55A-2RS/GS/VJS

FUNCTIONAL DESCRIPTION OF PIN

Pin No.	Item	Input/Output	Function
D7 ~ D0	Bidirectional data bus	Input and output	These are three-state 8-bit bidirectional buses used to write and read data upon receipt of the $\overline{WR}$ and $\overline{RD}$ signals from CPU and also used when control words and bit set/reset data are transferred from CPU to MSM82C55A.
RESET	Reset input	Input	This signal is used to reset the control register and all internal registers when it is in high level. At this time, ports are all made into the input mode (high impedance status). all port latches are cleared to O. and all ports groups are set to mode O.
$\overline{CS}$	Chip select input	Input	When the $\overline{CS}$ is in low level, data transmission is enabled with CPU. When it is in high level, the data bus is made into the high impedance status where no write nor read operation is performed. Internal registers hold their previous status, however.
$\overline{RD}$	Read input	Input	When RD is in low level, data is transferred from MSM82C55A to CPU.
$\overline{WR}$	Write input	Input	When WR is in low level, data or control words are transferred from CPU to MSM82C55A.
A0, A1	Port select input (address)	Input	By combination of A0 and A1, either one is selected from among port A, port B, port C, and control register. These pins are usually connected to low order 2 bits of the address bus.
PA7 ~ PA0	Port A	Input and output	These are universal 8-bit I/O ports. The direction of inputs/outputs can be determined by writing a control word. Especially, port A can be used as a bidirectional port when it is set to mode 2.
PB7 ~ PB0	Port B	Input and output	These are universal 8-bit I/O ports. The direction of inputs/outputs can be determined by writing a control word.
PC7 ~ PC0	Port C	Input and output	These are universal 8-bit I/O ports. The direction of inputs/outputs can be determined by writing a control word as 2 ports with 4 bits each. When port A or port B is used in mode 1 or mode 2 (port A only), they become control pins. Especially when port C is used as an output port, each bit can be set/reset independently.
V_{CC}			+5 V power supply.
GND			GND

5

BASIC FUNCTIONAL DESCRIPTION

Group A and Group B

When setting a mode to a port having 24 bits, set it by dividing it into two groups of 12 bits each.

Group A: Port A (8 bits) and high order 4 bits of port C (PC7 ~ PC4)

Group B: Port B (8 bits) and low order 4 bits of port C (PC3 ~ PC0)

Mode 0, 1, 2

There are 3 types of modes to be set by grouping as follows:

Mode 0: Basic input operation/output operation (Available for both groups A and B)

Mode 1: Strobe input operation/output operation (Available for both groups A and B)

Mode 2: Bidirectional bus operation (Available for group A only)

When used in mode 1 or mode 2, however, port C has bits to be defined as ports for control signal for operation ports (port A for group A and port B for group B) of their respective groups.

Port A, B, C

The internal structure of 3 ports is as follows:

Port A: One 8-bit data output latch/buffer and one 8-bit data input latch

Port B: One 8-bit data input/output latch/buffer and one 8-bit data input buffer

Port C: One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input)

Single bit set/reset function for port C

When port C is defined as an output port, it is possible to set (to turn to high level) or reset (to turn to low level) any one of 8 bits individually without affecting other bits.

336

OPERATIONAL DESCRIPTION

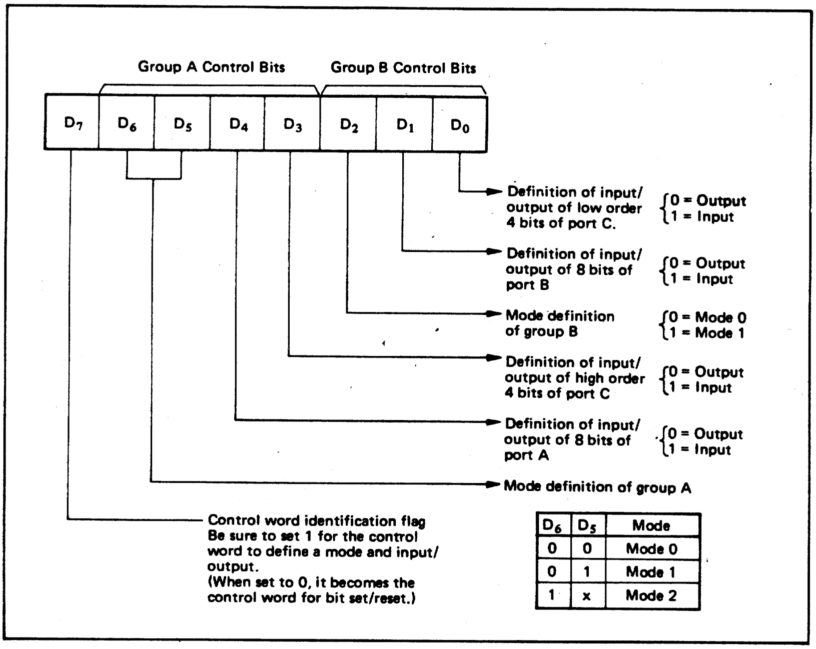
Control Logic

Operations by addresses and control signals, e.g., read and write, etc. are as shown in the table below:

Operation	A1	A0	$\overline{CS}$	$\overline{WR}$	$\overline{RD}$	Operation
Input	0	0	0	1	0	Port A → Data Bus
	0	1	0	1	0	Port B → Data Bus
	1	0	0	1	0	Port C → Data Bus
Output	0	0	0	0	1	Data Bus → Port A
	0	1	0	0	1	Data Bus → Port B
	1	0	0	0	1	Data Bus → Port C
Control	1	1	0	0	1	Data Bus → Control Register
Others	1	1	0	1	0	Illegal Condition
	x	x	1	x	x	Data bus is in the high impedance status.

Setting of Control Word

The control register is composed of 7-bit latch circuit and 1-bit flag as shown below.



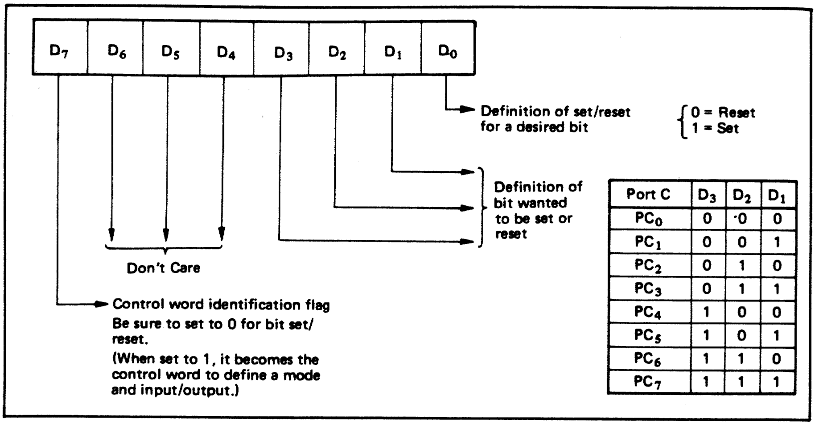
Precaution for mode selection

The output registers for ports A and C are cleared to ϕ each time data is written in the command register and the mode is changed, but the port B state is undefined.

Bit Set/Reset Function

When port C is defined as output port, it is possible to set (set output to 1) or reset (set output to 0) any one of 8 bits without affecting other bits as shown next page.

■ I/O·MSM82C55A-2RS/GS/VJS ■



Port C	D_3	D_2	D_1
PC_0	0	0	0
PC_1	0	0	1
PC_2	0	1	0
PC_3	0	1	1
PC_4	1	0	0
PC_5	1	0	1
PC_6	1	1	0
PC_7	1	1	1

Interrupt Control Function

When the MSM82C55A is used in mode 1 or mode 2, the interrupt signal for the CPU is provided. The interrupt request signal is output from port C. When the internal flip-flop INTE is set beforehand at this time, the desired interrupt request signal is output. When it is reset beforehand, however, the interrupt request signal is not output. The set/reset of the internal flip-flop is made by the bit set/reset operation for port C virtually.

Bit set → INTE is set → Interrupt allowed
Bit reset → INTE is reset → Interrupt inhibited

Operational Description by Mode

1. **Mode 0 (Basic input/output operation)**
   Mode 0 makes the MSM82C55A operate as a basic input port or output port. No control signals such as interrupt request, etc. are required in this mode. All 24 bits can be used as two-8-bit ports and two 4-bit ports. Sixteen combinations are then possible for inputs/outputs. The inputs are not latched, but the outputs are.

5

Type	Control Word								Group A		Group B	
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Port A	High Order 4 Bits of Port C	Port B	Low Order 4 Bits of Port C
1	1	0	0	0	0	0	0	0	Output	Output	Output	Output
2	1	0	0	0	0	0	0	1	Output	Output	Output	Input
3	1	0	0	0	0	0	1	0	Output	Output	Input	Output
4	1	0	0	0	0	0	1	1	Output	Output	Input	Input
5	1	0	0	0	1	0	0	0	Output	Input	Output	Output
6	1	0	0	0	1	0	0	1	Output	Input	Output	Input
7	1	0	0	0	1	0	1	0	Output	Input	Input	Output
8	1	0	0	0	1	0	1	1	Output	Input	Input	Input
9	1	0	0	1	0	0	0	0	Input	Output	Output	Output
10	1	0	0	1	0	0	0	1	Input	Output	Output	Input
11	1	0	0	1	0	0	1	0	Input	Output	Input	Output
12	1	0	0	1	0	0	1	1	Input	Output	Input	Input
13	1	0	0	1	1	0	0	0	Input	Input	Output	Output
14	1	0	0	1	1	0	0	1	Input	Input	Output	Input
15	1	0	0	1	1	0	1	0	Input	Input	Input	Output
16	1	0	0	1	1	0	1	1	Input	Input	Input	Input

Note: When used in mode 0 for both groups A and B

2. Mode 1 (Strobe input/output operation)

In mode 1, the strobe, interrupt and other control signals are used when input/output operations are made from a specified port. This mode is available for both groups A and B. In group A at this time, port A is used as the data line and port C as the control signal.

Following is a descrption of the input operation in mode 1.

$\overline{STB}$ (Strobe input)

- When this signal is low level, the data output from terminal to port is fetched into the internal latch of the port. This can be made independent from the CPU, and the data is not output to the data bus until the RD signal arrives from the CPU.

IBF (Input buffer full flag output)

- This is the response signal for the $\overline{STB}$. This signal when turned to high level indicates that data is fetched into the input latch. This signal turns to high level at the falling edge of $\overline{STB}$ and to low level at the rising edge of $\overline{RD}$.

INTR (Interrupt request output)

- This is the interrupt request signal for the CPU of the data fetched into the input latch. It is indicated by high level only when the internal INTE flip-flop is set. This signal turns to high level at the rising edge of the $\overline{STB}$ (IBF = 1 at this time) and low level at the falling edge of the $\overline{RD}$ when the INTE is set.

$INTE_A$ of group A is set when the bit for PC_4 is set, while $INTE_B$ of group B is set when the bit for PC_2 is set.

Following is a description of the output operation of mode 1.

$\overline{OBF}$ (Output buffer full flag output)

- This signal when turned to low level indicates that data is written to the specified port upon receipt of the $\overline{WR}$ signal from the CPU. This signal turns to low level at the rising edge of the WR and high level at the falling edge of the ACK.

$\overline{ACK}$ (Acknowledge input)

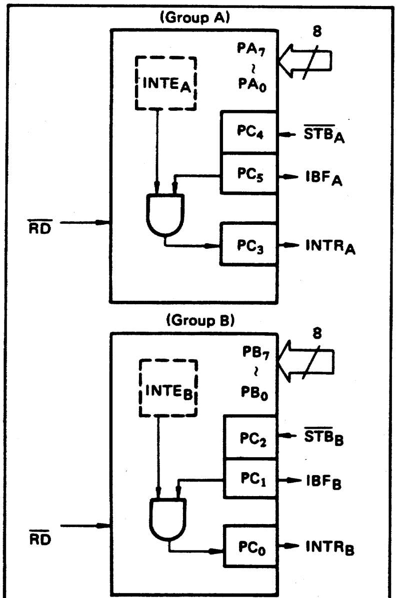
- This signal when turned to low level indicates that the terminal has received data.

INTR (Interrupt request output)

- This is the signal used to interrupt the CPU when a terminal receives data from the CPU via the MSM82C55A-5. It indicates the occurrence of the interrupt in high level only when the internal INTE flip-flop is set. This signal turns to high level at the rising edge of the $\overline{ACK}$ (OBF = 1 at this time) and low level at the falling edge of $\overline{WR}$ when the $INTE_B$ is set.

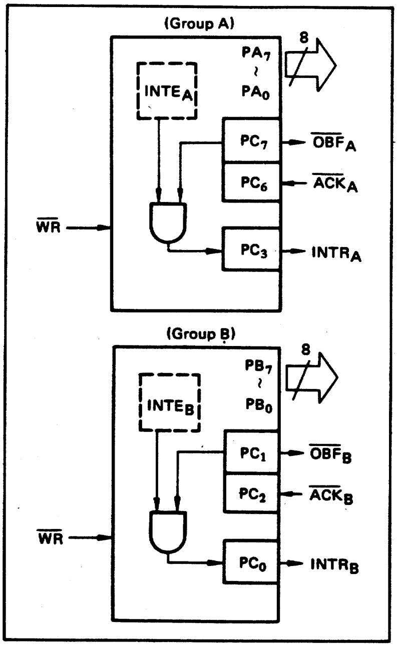
$INTE_A$ of group A is set when the bit for PC_6 is set, while $INTE_B$ of group B is set when the bit for PC_2 is set.

Mode 1 Input



Note: Although belonging to group B, PC_3 operates as the control signal of group A functionally.

Mode 1 output



5

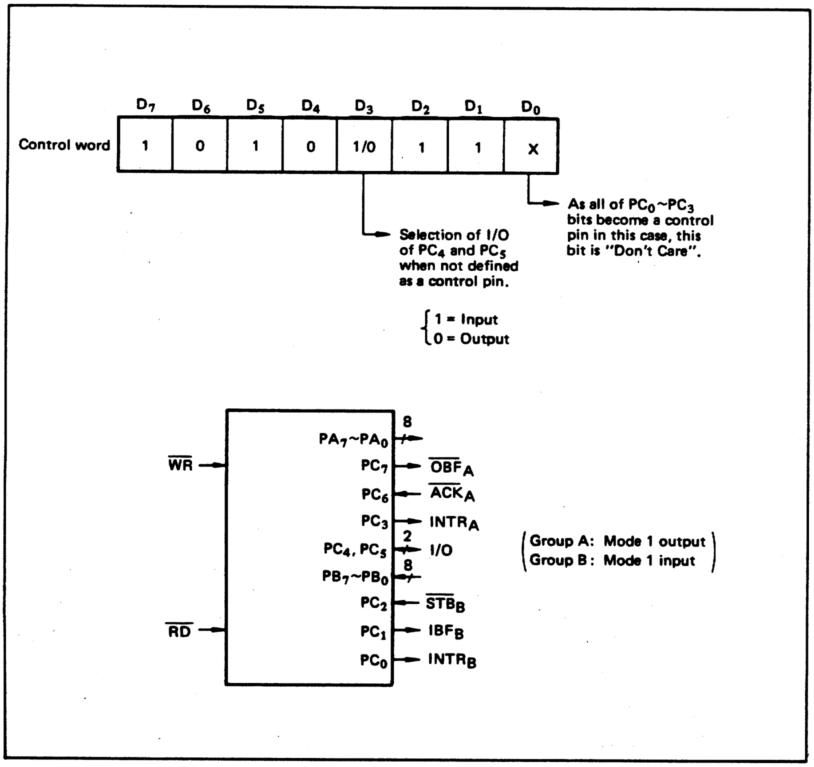
■ I/O·MSM82C55A-2RS/GS/VJS ■

Port C Function Allocation in Mode 1

Port C \ Combination of Input/Output	Group A: Input Group B: Input	Group A: Input Group B: Output	Group A: Output Group B: Input	Group A: Output Group B: Output
PC_0	$INTR_B$	$INTR_B$	$INTR_B$	$INTR_B$
PC_1	IBF_B	$\overline{OBF}_B$	IBF_B	$\overline{OBF}_B$
PC_2	$\overline{STB}_B$	$\overline{ACK}_B$	$\overline{STB}_B$	$\overline{ACK}_B$
PC_3	$INTR_A$	$INTR_A$	$INTR_A$	$INTR_A$
PC_4	$\overline{STB}_A$	$\overline{STB}_A$	I/O	I/O
PC_5	IBF_A	IBF_A	I/O	I/O
PC_6	I/O	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$
PC_7	I/O	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$

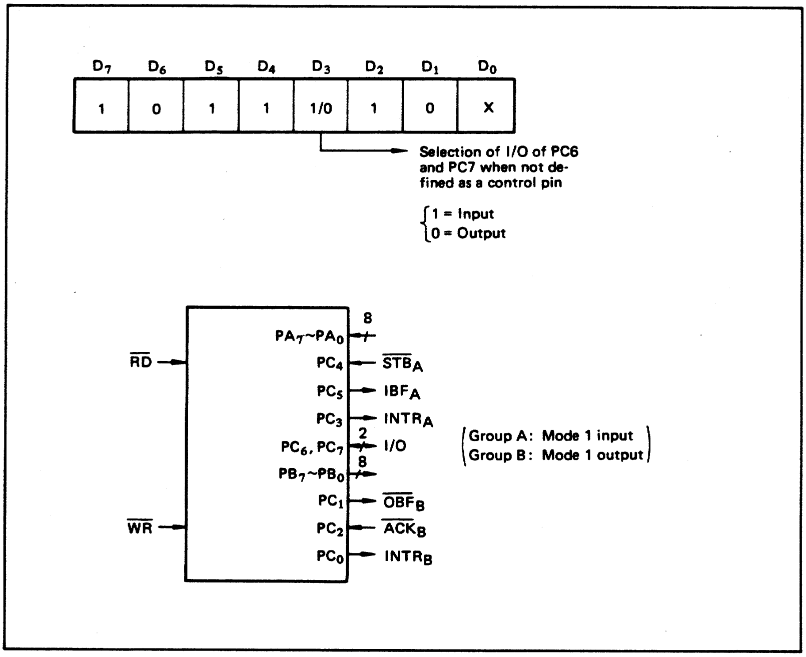
Note: I/O is a bit not used as the control signal, but it is available as a port of mode 0.

Examples of the relation between the control words and pins when used in mode 1 is shown below:
(a) When group A is mode 1 output and group B is mode 1 input.



5

(b) When group A is mode 1 input and group B is mode 1 output.



3. Mode 2 (Strobe bidirectional bus I/O operation)

In mode 2, it is possible to transfer data in 2 directions through a single 8-bit port. This operation is akin to a combination between input and output operations. Port C waits for the control signal in this case, too. Mode 2 is available only for group A, however.

Next, a description is made on mode 2.

$\overline{\text{OBF}}$ (Output buffer full flag output)

- This signal when turned to low level indicates that data has been written to the internal output latch upon receipt of the $\overline{\text{WR}}$ signal from the CPU. At this time, port A is still in the high impedance status and the data is not yet output to the outside. This signal turns to low level at the rising edge of the $\overline{\text{WR}}$ and high level at the falling edge of the $\overline{\text{ACK}}$.

$\overline{\text{ACK}}$ (Acknowledge input)

- When a low level signal is input to this pin, the high impedance status of port A is cleared, the buffer is enabled, and the data written to the internal output latch is output to port A. When the input returns to high level, port A is made into the high impedance status.

$\overline{\text{STB}}$ (Strobe input)

- When this signal turns to low level, the data output to the port from the pin is fetched into the internal input latch. The data is output to the data bus upon receipt of the RD signal from the CPU, but it remains in the high impedance status until then.

IBF (Input buffer full flag output)

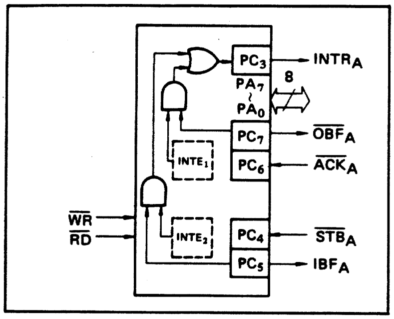
- This signal when turned to high level indicates that data from the pin has been fetched into the input latch. This signal turns to high level at the falling edge of the $\overline{\text{STB}}$ and low level at the rising edge of the $\overline{\text{RD}}$.

INTR (Interrupt request output)

- This signal is used to interrupt the CPU and its operation in the same as in mode 1. There are two INTE flip-flops internally available for input and output to select either interrupt of input or output operation. The INTE1 is used to control the interrupt request for output operation and it can be reset by the bit set for PC6. INTE2 is used to control the interrupt request for the input operation and it can be set by the bit set for PC4.

■ I/O·MSM82C55A-2RS/GS/VJS ■

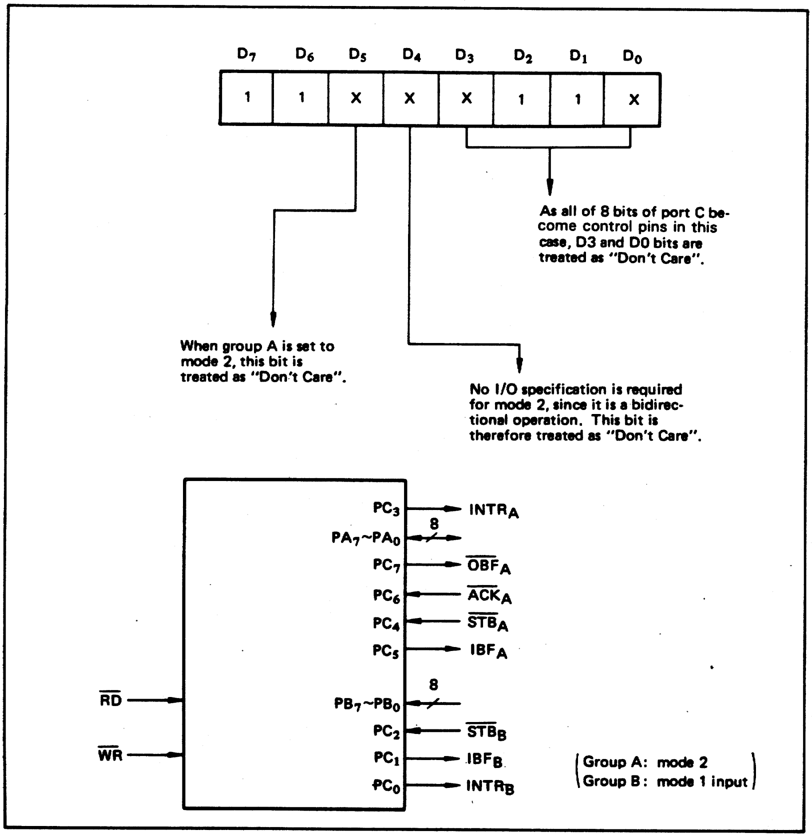
Mode 2 I/O Operation



Port C Function Allocation in Mode 2

Port C	Function
PC_0	Confirmed to the group B mode
PC_1	
PC_2	
PC_3	$INTR_A$
PC_4	$\overline{STB}_A$
PC_5	IBF_A
PC_6	$\overline{ACK}_A$
PC_7	$\overline{OBF}_A$

Following is an example of the relation between the control word and the pin when used in mode 2. When input in mode 2 for group A and in mode 1 for group B.



5

342

4. When Group A is Different in Mode from Group B
Group A and group B can be used by setting them in different modes each other at the same time. When either group is set to mode1 or mode 2, it is possible to set the one not defined as a control pin in port C to both input and output as a port which operates in mode 0 at the 3rd and 0th bits of the control word.

(Mode combinations that define no control bit at port C)

	Group A	Group B	Port C							
			PC_7	PC_6	PC_5	PC_4	PC_3	PC_2	PC_1	PC_0
1	Mode 1 input	Mode 0	I/O	I/O	IBF_A	$\overline{STB}_A$	$INTR_A$	I/O	I/O	I/O
2	Mode 0 output	Mode 0	$\overline{OBF}_A$	$\overline{ACK}_A$	I/O	I/O	$INTR_A$	I/O	I/O	I/O
3	Mode 0	Mode 1 input	I/O	I/O	I/O	I/O	I/O	$\overline{STB}_B$	IBF_B	$INTR_B$
4	Mode 0	Mode 1 output	I/O	I/O	I/O	I/O	I/O	$\overline{ACK}_B$	$\overline{OBF}_B$	$INTR_B$
5	Mode 1 input	Mode 1 input	I/O	I/O	IBF_A	$\overline{STB}_A$	$INTR_A$	$\overline{STB}_B$	IBF_B	$INTR_B$
6	Mode 1 input	Mode 1 output	I/O	I/O	IBF_A	$\overline{STB}_A$	$INTR_A$	$\overline{ACK}_B$	$\overline{OBF}_B$	$INTR_B$
7	Mode 1 output	Mode 1 input	$\overline{OBF}_A$	$\overline{ACK}_A$	I/O	I/O	$INTR_A$	$\overline{STB}_B$	IBF_B	$INTR_B$
8	Mode 1 output	Mode 1 output	$\overline{OBF}_A$	$\overline{ACK}_A$	I/O	I/O	$INTR_A$	$\overline{ACK}_B$	$\overline{OBF}_B$	$INTR_B$
9	Mode 2	Mode 0	$\overline{OBF}_A$	$\overline{ACK}_A$	IBF_A	$\overline{STB}_A$	$INTR_A$	I/O	I/O	I/O

Controlled at the 3rd bit (D3) of the control word (PC_7–PC_4)

Controlled at the 0th bit (D0) of the control word (PC_3–PC_0)

When the I/O bit is set to input in this case, it is possible to access data by the normal port C read operation.
When set to output, PC7 ~ PC4 bits can be accessed by the bit set/reset function only. Meanwhile, 3 bits from PC2 to PC0 can be accessed by normal write operation.

The bit set/reset function can be used for all of PC3 ~ PC0 bits. Note that the status of port C varies according to the combination of modes like this.

■ I/O·MSM82C55A-2RS/GS/VJS ■

5. Port C Status Read

When port C is used for the control signal, that is, in either mode 1 or mode 2, each control signal and bus status signal can be read out by reading the content of port C.

The status read out is as follows:

	Group A	Group B	Status read on the data bus							
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	Mode 1 input	Mode 0	I/O	I/O	IBF_A	$INTE_A$	$INTR_A$	I/O	I/O	I/O
2	Mode 1 output	Mode 0	$\overline{OBF}_A$	$INTE_A$	I/O	I/O	$INTR_A$	I/O	I/O	I/O
3	Mode 0	Mode 1 input	I/O	I/O	I/O	I/O	I/O	$INTE_B$	IBF_B	$INTR_B$
4	Mode 0	Mode 1 output	I/O	I/O	I/O	I/O	I/O	$INTE_B$	$\overline{OBF}_B$	$INTR_B$
5	Mode 1 input	Mode 1 input	I/O	I/O	IBF_A	$INTE_A$	$INTR_A$	$INTE_B$	IBF_B	$INTR_B$
6	Mode 1 input	Mode 1 output	I/O	I/O	IBF_A	$INTE_A$	$INTR_A$	$INTE_B$	$\overline{OBF}_B$	$INTR_B$
7	Mode 1 output	Mode 1 input	$\overline{OBF}_A$	$INTE_A$	I/O	I/O	$INTR_A$	$INTE_B$	IBF_B	$INTR_B$
8	Mode 1 output	Mode 1 output	$\overline{OBF}_A$	$INTE_A$	I/O	I/O	$INTR_A$	$INTE_B$	$\overline{OBF}_B$	$INTR_B$
9	Mode 2	Mode 0	$\overline{OBF}_A$	$INTE_1$	IBF_A	$INTE_2$	$INTR_A$	I/O	I/O	I/O
10	Mode 2	Mode 1 input	$\overline{OBF}_A$	$INTE_1$	IBF_A	$INTE_2$	$INTR_A$	$INTE_B$	IBF_B	$INTR_B$
11	Mode 2	Mode 1 output	$\overline{OBF}_A$	$INTE_1$	IBF_A	$INTE_2$	$INTR_A$	$INTE_B$	$\overline{OBF}_B$	$INTR_B$

5

6. Reset of MSM82C55A

Be sure to keep the RESET signal at power ON in the high level at least for 50 µs. Subsequently, it becomes the input mode at a high level pulse above 500 ns.

Note:

MSM82C55A-5

After a write command is executed to the command register, the internal latch is cleared in PORTA PORTC. For instance, 00H is output at the beginning of a write command when the output port is assigned. However, if PORTB is not cleared at this time, PORTB is unstable. In other words, PORTB only outputs ineffective data (unstable value according to the device) during the period from after a write command is executed till the first data is written to PORTB.

MSM82C55A-2

After a write command is executed to the command register, the internal latch is cleared in All Ports(PORTA,PORTB,PORTC). 00H is ontput at the beginning of a write command when the output port is assigned.

PC Card Questions and Answers for Windows 3.1

Appendix D

This appendix contains a list of common questions and answers relating to PC Card (PCMCIA) operation. The questions are grouped according to the type of information requested. You may find this information useful if you are having difficulty with the PC Card system software configuration and you are using Windows 3.1.

Configuration

Do I need to use my PC Card configuration utility to configure the National Instruments PC Cards?

No. We recommend that you do not configure our PC Cards using PC Card Control or an equivalent PC Card configuration utility. Use the configuration utilities included with the NI-DAQ driver software to properly configure your DAQCard. The appropriate utility is the NI-DAQ Configuration Utility (WDAQCONF) for Windows 3.1 users.

What should I do if my computer does not have Card and Socket Services version 2.0 or later?

Contact the manufacturer of your computer or of your PC Card adapter and request the latest Card and Socket PC Card driver. Our NI-DAQ software will work with any Card and Socket Service driver that is compliant to version 2.0 or higher.

Operation

My PC Card works when inserted before power-on time, but it does not work when hot inserted. What is wrong?

You may have an interrupt conflict. If you have a utility such as `MSD.EXE`, run it to determine the allocated interrupts, then refer to question 5 in the Resources section. `MSD.EXE` is usually shipped with Microsoft Windows.

My computer locks up when I use a PC Card. What should I do?

This usually happens because Card Services allocated an unusable interrupt level to the PC Card. For example, on some computers, interrupt level 11 is not routed to PC Cards. If Card Services is not aware of this, it may assign interrupt 11 to a PC Card even though the interrupt is not usable. When a call uses the interrupt, the interrupt never occurs, and the computer locks up waiting for a response. For information about how to locate an interrupt that is free to be used, refer to the *Resources* section.

Resources

How do I determine if I have a memory conflict?

If no PC Cards are working at all, it is probably because a memory window is not usable. Card Services uses a 4 kB memory window for its own internal use. If the memory cannot be used, then Card Services cannot read the Card Information Structure (CIS) from the DAQCard EPROM, which means it cannot identify boards.

There are two different methods you can use when Card Services has a problem reading the CIS. First, you can determine which memory window Card Services is using, and then exclude that window from use by Card Services and/or the memory manager. Second, you can attempt to determine all of the memory that Card Services can possibly use and then exclude all but that memory from use by Card Services.

How do I determine all of the memory that Card Services can use?

One way to find out which memory addresses Card Services can use is to run a utility such as `MSD.EXE` that scans the system and tells you how the system memory is being used. For example, if you run such a memory utility and it tells you that physical addresses C0000 to C9FFF are being used for ROM access, then you know that C8000-D3FFF is an invalid range for Card Services and should be changed to CA000-D3FFF.

How can I find usable I/O addresses?

Identify usable I/O addresses by trial and error. Of the three resources used—memory, I/O, interrupts—I/O conflicts will be low. You can use the NI-DAQ configuration utility in Windows to diagnose I/O space conflicts. When you have configured the NI-DAQ configuration utility for a particular I/O space, save the configuration. If there is a conflict, the configuration utility will report an error describing the conflict.

How do I find usable interrupt levels?

Some utilities, such as `MSD.EXE`, will scan the system and display information about what is using hardware interrupts. If you have such a utility, you can run it to determine what interrupts Card Services can use. Card Services needs an interrupt for itself as well as one interrupt for each PCMCIA socket in the system. For example, in a system with two PCMCIA sockets, at least three interrupts should be allocated for use by Card Services.

Keep in mind that utilities such as `MSD.EXE` will sometimes report that an interrupt is in use when it really is not. For example, if the computer has one serial port, COM1, and one parallel port, LPT1, you know that IRQs 4 and 7 are probably in use. In general, IRQ5 is used for LPT2, but if the computer does not have two parallel ports, IRQ5 should be usable. IRQ3 is used for COM2, but if the computer has only has one serial port, IRQ3 should be usable.

I run a memory utility, and it appears there is no memory available for Card Services. What should I do?

You should remove your memory manager by commenting it out of the `CONFIG.SYS` file. Next, you can rerun the memory utility. Memory managers often consume an enormous amount of memory, and you will need to determine what memory is really usable by Card Services. When you have determined what memory is available for Card Services, reinstall your memory manager and make the necessary changes to provide Card Services with the memory needed. We suggest that you use the minimum amount of memory for Card Services, namely 4 to 12 kB, which frees more memory for the memory manager.

Resource Conflicts

How do I resolve conflicts between my memory manager and Card Services?

Card Services can usually use memory space that is not being used for real RAM on the system. Even when this is the case, you should still exclude the memory addresses used by Card Services from use by any memory manager that may be installed.

Customer Communication

Appendix E

For your convenience, this appendix contains forms to help you gather the information necessary to help us solve your technical problems and a form you can use to comment on the product documentation. When you contact us, we need the information on the Technical Support Form and the configuration form, if your manual contains one, about your system configuration to answer your questions as quickly as possible.

National Instruments has technical assistance through electronic, fax, and telephone systems to quickly provide the information you need. Our electronic services include a bulletin board service, an FTP site, a Fax-on-Demand system, and e-mail support. If you have a hardware or software problem, first try the electronic support systems. If the information available on these systems does not answer your questions, we offer fax and telephone support through our technical support centers, which are staffed by applications engineers.

Electronic Services

Bulletin Board Support

National Instruments has BBS and FTP sites dedicated for 24-hour support with a collection of files and documents to answer most common customer questions. From these sites, you can also download the latest instrument drivers, updates, and example programs. For recorded instructions on how to use the bulletin board and FTP services and for BBS automated information, call (512) 795-6990. You can access these services at:

United States: (512) 794-5422
Up to 14,400 baud, 8 data bits, 1 stop bit, no parity

United Kingdom: 01635 551422
Up to 9,600 baud, 8 data bits, 1 stop bit, no parity

France: 01 48 65 15 59
Up to 9,600 baud, 8 data bits, 1 stop bit, no parity

FTP Support

To access our FTP site, log on to our Internet host, `ftp.natinst.com`, as `anonymous` and use your Internet address, such as `joesmith@anywhere.com`, as your password. The support files and documents are located in the `/support` directories.

Fax-on-Demand Support

Fax-on-Demand is a 24-hour information retrieval system containing a library of documents on a wide range of technical information. You can access Fax-on-Demand from a touch-tone telephone at (512) 418-1111.

E-Mail Support (currently U.S. only)

You can submit technical support questions to the applications engineering team through e-mail at the Internet address listed below. Remember to include your name, address, and phone number so we can contact you with solutions and suggestions.

`support@natinst.com`

Telephone and Fax Support

National Instruments has branch offices all over the world. Use the list below to find the technical support number for your country. If there is no National Instruments office in your country, contact the source from which you purchased your software to obtain support.

	Telephone	Fax
Australia	03 9879 5166	03 9879 6277
Austria	0662 45 79 90 0	0662 45 79 90 19
Belgium	02 757 00 20	02 757 03 11
Canada (Ontario)	905 785 0085	905 785 0086
Canada (Quebec)	514 694 8521	514 694 4399
Denmark	45 76 26 00	45 76 26 02
Finland	09 725 725 11	09 725 725 55
France	01 48 14 24 24	01 48 14 24 14
Germany	089 741 31 30	089 714 60 35
Hong Kong	2645 3186	2686 8505
Israel	03 5734815	03 5734816
Italy	02 413091	02 41309215
Japan	03 5472 2970	03 5472 2977
Korea	02 596 7456	02 596 7455
Mexico	5 520 2635	5 520 3282
Netherlands	0348 433466	0348 430673
Norway	32 84 84 00	32 84 86 00
Singapore	2265886	2265887
Spain	91 640 0085	91 640 0533
Sweden	08 730 49 70	08 730 43 70
Switzerland	056 200 51 51	056 200 51 55
Taiwan	02 377 1200	02 737 4644
United Kingdom	01635 523545	01635 523154
United States	512 795 8248	512 794 5678

Technical Support Form

Photocopy this form and update it each time you make changes to your software or hardware, and use the completed copy of this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

If you are using any National Instruments hardware or software products related to this problem, include the configuration forms from their user manuals. Include additional pages if necessary.

Name ____

Company ____

Address ____

____

Fax (___)____________________ Phone (___) ___________________________________

Computer brand ________________ Model ________________ Processor____________________

Operating system (include version number) ______________________________________

Clock speed ______MHz RAM _____MB Display adapter ___________________________

Mouse ___yes ___no Other adapters installed _____________________________________

Hard disk capacity _____MB Brand ____

Instruments used ____

____

National Instruments hardware product model __________ Revision _____________________

Configuration ____

National Instruments software product ____________________________ Version ____________

Configuration ____

The problem is: ____

____

____

____

____

List any error messages: ____

____

____

The following steps reproduce the problem:_____________________________________

____

____

____

____

____

DAQCard-DIO-24 Hardware and Software Configuration Form

Record the settings and revisions of your hardware and software on the line to the right of each item. Complete a new copy of this form each time you revise your software or hardware configuration, and use this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

National Instruments Products

DAQ hardware ______________________________

Interrupt level of hardware ______________________________

DMA channels of hardware ______________________________

Base I/O address of hardware ______________________________

Programming choice ______________________________

Software and version ______________________________

Other boards in system ______________________________

Base I/O address of other boards ______________________________

DMA channels of other boards ______________________________

Interrupt level of other boards ______________________________

Other Products

Computer make and model ______________________________

Microprocessor ______________________________

Clock frequency or speed ______________________________

Type of video board installed ______________________________

Operating system version ______________________________

Operating system mode ______________________________

Programming language ______________________________

Programming language version ______________________________

Other boards in system ______________________________

Base I/O address of other boards ______________________________

DMA channels of other boards ______________________________

Interrupt level of other boards ______________________________

Documentation Comment Form

National Instruments encourages you to comment on the documentation supplied with our products. This information helps us provide quality products to meet your needs.

Title: *DAQCard™-DIO-24 User Manual*

Edition Date: October 1997

Part Number: 320788D-01

Please comment on the completeness, clarity, and organization of the manual.

____

____

____

____

____

____

____

If you find errors in the manual, please record the page numbers and describe the errors.

____

____

____

____

____

____

____

Thank you for your help.

Name ____

Title ____

Company ____

Address ____

____

Phone (___)____________________________ Fax (___) ___________________________

Mail to: Technical Publications
National Instruments Corporation
6504 Bridge Point Parkway
Austin, TX 78730-5039

Fax to: Technical Publications
National Instruments Corporation
(512) 794-5678

Glossary

Prefix	Meaning	Value
p-	pico-	10^{-12}
n-	nano-	10^{-9}
µ-	micro-	10^{-6}
m-	milli-	10^{-3}
k-	kilo-	10^{3}
M-	mega-	10^{6}
G-	giga-	10^{9}

Numbers/Symbols

°	degree
>	greater than
<	less than
≤	less than or equal to
–	negative of, or minus
Ω	ohm
/	per
%	percent
±	plus or minus
+	positive of, or plus

A

A	amperes
AC	alternating current
ACK	acknowledge input signal
ACKA	acknowledge input signal for port A
ACKB	acknowledge input signal for port B
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
address	character code that identifies a specific location (or series of locations) in memory
AI	analog input
AO	analog output

B

b	bit—one binary digit, either 0 or 1
B	byte—eight related bits of data, an eight-bit binary number. Also used to denote the amount of memory required to store one byte of data
base address	a memory address that serves as the starting address for programmable registers. All other addresses are located by adding to the base address
BCD	binary-coded decimal
bidirectional	the port can be configured for input or output
bidirectional bus	the port can input and output data without being reconfigured
binary	a number system with a base of 2

BIOS	basic input/output system—BIOS functions are the fundamental level of any PC or compatible computer. BIOS functions embody the basic operations needed for successful use of the computer's hardware resources
buffer	temporary storage for acquired or generated data (software)
bus	the group of conductors that interconnect individual circuitry in a computer. Typically, a bus is the expansion vehicle to which I/O or other devices are connected

C

C	Celsius
channel	pin or wire lead to which you apply or from which you read the analog or digital signal. Analog signals can be single-ended or differential. For digital signals, you group channels to form ports. Ports usually consist of either four or eight digital channels
CIS	card information structure
clock	hardware component that controls timing for reading from or writing to groups
CMOS	complementary metal-oxide semiconductor
counter/timer	a circuit that counts external pulses or clock pulses (timing)
coupling	the manner in which a signal is connected from one location to another
CPU	central processing unit
crosstalk	an unwanted signal on one channel due to an input on a different channel
current drive capability	the amount of current a digital or analog output channel is capable of sourcing or sinking while still operating within voltage range specifications
current sinking	the ability of a DAQ board to dissipate current for analog or digital output signals

current sourcing	the ability of a DAQ board to supply current for analog or digital output signals

D

DAQ	data acquisition—(1) collecting and measuring electrical signals from sensors, transducers, and test probes or fixtures and inputting them to a computer for processing; (2) collecting and measuring the same kinds of electrical signals with A/D and/or DIO boards plugged into a computer, and possibly generating control signals with D/A and/or DIO boards in the same computer
DATA	data lines
DC	direct current
device	a plug-in data acquisition board, card, or pad that can contain multiple channels and conversion devices. Plug-in boards, PCMCIA cards, and devices such as the DAQPad-1200, which connects to your computer parallel port, are all examples of DAQ devices. SCXI modules are distinct from devices, with the exception of the SCXI-1200, which is a hybrid
DGND	digital ground
differential input	an analog input consisting of two terminals, both of which are isolated from computer ground, whose difference is measured
DIG	digital
digital port	*See* port
DIO	digital input/output
drivers	software that controls a specific hardware device such as a DAQ board or a GPIB interface board

E

EEPROM	electrically erasable programmable read-only memory—ROM that can be erased with an electrical signal and reprogrammed
EISA	extended industry standard architecture

EPROM	erasable programmable read-only memory—ROM that can be erased (usually by ultraviolet light exposure) and reprogrammed

F

FIFO	first-in first-out memory buffer—the first data stored is the first data sent to the acceptor. FIFOs are often used on DAQ devices to temporarily store incoming or outgoing data until that data can be retrieved or output. For example, an analog input FIFO stores the results of A/D conversions until the data can be retrieved into system memory, a process that requires the servicing of interrupts and often the programming of the DMA controller. This process can take several milliseconds in some cases. During this time, data accumulates in the FIFO for future retrieval. With a larger FIFO, longer latencies can be tolerated. In the case of analog output, a FIFO permits faster update rates, because the waveform data can be stored on the FIFO ahead of time. This again reduces the effect of latencies associated with getting the data from system memory to the DAQ device
filtering	a type of signal conditioning that allows you to filter unwanted signals from the signal you are trying to measure

G

GND	ground signal
GPCTR	general-purpose counter

H

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
hardware	the physical components of a computer system, such as the circuit boards, plug-in boards, chassis, enclosures, peripherals, cables, and so on
hex	hexadecimal
Hz	hertz—the number of scans read or updates written per second

I

IBF	input buffer full signal
IBFA	input buffer full for port A signal
IBFB	input buffer full for port B signal
IC	integrated circuit
immediate digital I/O	a type of digital acquisition/generation where LabVIEW updates the digital lines or port states immediately or returns the digital value of an input line; also called nonlatched digital I/O
in.	inches
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
INTE1	interrupt enable bit for output
INTE2	interrupt enable bit for input
INTEA	interrupt enable bit for port A
INTEB	interrupt enable bit for port B
INTEN	interrupt enable bit/signal
INTR	interrupt request signal
INTRA	interrupt request status bit for port A
INTRB	interrupt request status but for port B
interrupt	a computer signal indicating that the CPU should suspend its current task to service a designated activity
interrupt level	the relative priority at which a device can interrupt
I/O	input/output—the transfer of data to/from a computer system involving communications channels, operator interface devices, and/or data acquisition and control interfaces

IRQ	interrupt request
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

K

k	kilo—the standard metric prefix for 1,000, or 10^3, used with units of measure such as volts, hertz, and meters
K	kilo—the prefix for 1,024, or 2^{10}, used with B in quantifying data or computer memory
kbytes/s	a unit for data transfer that means 1,000 or 10^3 bytes/s
kS	1,000 samples
Kword	1,024 words of memory

L

LabVIEW	laboratory virtual instrument engineering workbench
latched 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 handshaked digital I/O
LSB	least significant bit

M

m	meters
M	(1) Mega, the standard metric prefix for 1 million or 10^6, when used with units of measure such as volts and hertz; (2) mega, the prefix for 1,048,576, or 2^{20}, when used with B to quantify data or computer memory

MB	megabytes of memory
Mbytes/s	a unit for data transfer that means 1 million or 10^6 bytes/s
MIO	multifunction I/O
MSB	most significant bit

N

nibble	one-half of a byte; namely, four adjacent bits; in this manual, upper and lower nibbles
NI-DAQ	National Instruments driver software for DAQ hardware
NO	normally open
noise	an undesirable electrical signal—Noise comes from external sources such as the AC power line, motors, generators, transformers, fluorescent lights, soldering irons, CRT displays, computers, electrical storms, welders, radio transmitters, and internal sources such as semiconductors, resistors, and capacitors. Noise corrupts signals you are trying to send or receive
nonlatched digital I/O	a type of digital acquisition/generation where software updates the digital lines or port states immediately or returns the digital value of an input line. Also called immediate digital I/O or non-handshaking

O

OBF	output buffer full signal
OBFA	output buffer full for port A signal
OBFB	output buffer full for port B signal
onboard channels	channels provided by the plug-in data acquisition board
operating system	base-level software that controls a computer, runs programs, interacts with users, and communicates with installed hardware or peripheral devices

OUT	output pin—a counter output pin where the counter can generate various TTL pulse waveforms

P

PA	port A
pattern generation	a type of handshaked (latched) digital I/O in which 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
PB	port B
PC	port C, or personal computer
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 work-stations; 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
Plug and Play devices	devices that do not require dip switches or jumpers to configure resources on the devices—also called switchless devices
Plug and Play ISA	a specification prepared by Microsoft, Intel, and other PC-related companies that will result in PCs with plug-in boards that can be fully configured in software, without jumpers or switches on the boards
port	(1) a communications connection on a computer or a remote controller (2) a digital port, consisting of four or eight lines of digital input and/or output
PPI	programmable peripheral interface

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
pts	points

R

RAM	random-access memory
RD	read signal
ribbon cable	a flat cable in which the conductors are side by side
ROM	read-only memory
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	seconds
S	samples
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
SCXI	Signal Conditioning eXtensions for Instrumentation—the National Instruments product line for conditioning low-level signals within an external chassis near sensors so only high-level signals are sent to DAQ boards in the noisy PC environment
SHIELD	shield signal
signal conditioning	the manipulation of signals to prepare them for digitizing
S/s	samples per second—used to express the rate at which a DAQ board samples an analog signal

SSR	solid state relay
STB	strobe input signal
STBA	strobe input signal for port A
STBB	stobe input signal for port B
switchless device	devices that do not require dip switches or jumpers to configure resources on the devices—also called Plug and Play devices

T

TC	terminal count—the highest value of a counter
transfer rate	the rate, measured in bytes/s, at which data is moved from source to destination after software initialization and set up operations; the maximum rate at which the hardware can operate
TTL	transistor-transistor logic

V

V	volts
VDC	volts direct current
VI	virtual instrument—(1) a combination of hardware and/or software elements, typically used with a PC, that has the functionality of a classic stand-alone instrument (2) a LabVIEW software module (VI), which consists of a front panel user interface and a block diagram program
V_{IH}	volts, input high
V_{IL}	volts, input low
V_{in}	volts in
V_{OH}	volts, output high
V_{OL}	volts, output low

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
WR	write signal

X

XMS	extended memory specification

Index

Numbers

A

B

C

D

E

F

N

O

P

Q

R

S

T

U

W