

PC-14B

User's Manual

Copyright 1991, 1992
All Rights Reserved

All right reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form by any means, electronic, mechanical, by photocopying, recording, or otherwise, in whole or in part, without prior written permission.

Third edition.

April 1992

Information furnished in this manual is believed to be accurate and reliable; however no responsibility is assumed for its use, nor for any infringements of patents or other rights of third parties which may result from its use.

IBM, IBM PC/XT/AT and IBM PS/2 are trademarks of International Business Machine Corporation.

BASIC is a trademark of Dartmouth College.

Microsoft is a trademark of Microsoft Corporation.

LabWindows is a trademark of National Instruments.

Table of Contents

Glossary	7
Introduction	9
Overview	9
Features	10
PC-14B Package Contents	11
Installation	13
Configuration	13
Installation	17
Interconnections	19
Connections to the Computer Backplane	19
User Connectors	19
Architecture	23
Functional Sub-Systems	24
Support Sub-Systems	24
I/O Connectors	25
Register Structure	27
Programming Guide	39
Programming the 8255 PIA	40
Programming the 8254 Timer/Counter	49
Using the External Interrupt Input	58
Driver Software	61
Quick Function Reference	62

Hardware Specifications	63
Appendix A	65
Base Address Switch Settings	65
Appendix B	69
Timing Specifications for 8255 Modes 1 and 2	69
Appendix C	71
User Serviceable Parts	71

Glossary

Acknowledge	A handshake line which is used by the receiving device to indicate that it has read the transmitted data.
Clock Pulse	A rising edge, then a falling edge, in that order applied to an 8254 timer/counter's clock input.
Counter Data Register	The eight-bit register of an 8254 timer/counter that corresponds to one of the two bytes in the counter's output latch for read operations and count register for write operations. The three Counter Data Registers (TC2, TC1, TC0) appear in the PC-14B's register map.
Counter Loading	The transfer of a count from an 8254 timer/counter's Count Register to its Counting Element.
DIP	An abbreviation for Dual In-line Package.
Handshake Lines	Dedicated signals which allow two different devices to exchange data under asynchronous hardware control. The 8255s on the PC-14B can provide handshaking lines.
LSB	Context sensitive abbreviation for either Least Significant Byte or Least Significant Bit
MSB	Context sensitive abbreviation for either Most Significant Byte or Most Significant Bit
PIA	Peripheral Interface Adapter. The 8255s on the PC-14B board are PIAs. Also referred to as PPIs (Programmable Peripheral Interfaces).
Programming a Counter	Writing a Control Word to the 8254 timer/counter's Control Register. The Control Word itself specifies which counter is being programmed.
Programming an 8255 PIA	Writing a Control Word to the 8255 PIA's Control Register. All three ports are configured simultaneously.
Strobe	A handshaking line which is used to signal to a receiving device that there is data to be read.
Trigger	A rising edge at an 8254 timer/counter's gate input.

1

Introduction

This chapter describes the PC-14B and lists the contents of the PC-14B package.

Overview

The PC-14B is a general purpose digital input/output board. It is built around two industry standard 8255 programmable peripheral interface (PPI) adapter devices and a three channel 8254 Timer/Counter chip.

The 8255s and 8254 featured on the PC-14B are high speed versions of the standard 8255AC-2 and 8254-2.

The 8255s control 48 lines of digital I/O. They are very flexible and powerful when interfacing to peripheral equipment and can be programmed for almost any 8-bit or 16-bit I/O application. The 8255 can operate in either unidirectional or bidirectional mode, with or without hardware handshaking. Interrupt requests to the host computer can be generated automatically.

8255 PIAs

The 8254 provides three high speed timer/counters. Each timer can be programmed to operate in one of five different modes, with sixteen-bit resolution. All timer signals and controls are available on the auxiliary connector of the board. The PC-14B has an on-board 2MHz reference oscillator. Each timer can generate interrupt requests to the host computer.

**8254 Counter
Timer**

The PC-14B also features a dedicated external interrupt input. This input can be used to cause the host computer to perform tasks on demand from external equipment or hardware.

**External
Interrupt**

PC-14 can be used for versatile interfacing to:

- Other PCs, for example another PC with a PC-14 installed, or a PC-36 or PC-192.
- Any other computer with an 8-bit or 16-bit parallel interface.
- Centronics compatible printers and plotters.
- Relays (electro-mechanical and solid state) and opto-isolators.
- Switches of various types.
- Relay boards (PC-37 and PC-38) and industrial I/O mounting racks.

- Panel meters, instruments and test equipment which have digital readouts and controls.
- Sensors, transducers, instruments and test equipment with frequency (pulse) outputs and controls.

These are a few examples of where the PC-14 can serve as a digital I/O system controller in laboratories, industry, production or part of a dedicated instrument for process monitoring and control.

The board plugs into any fully bussed slot of an IBM PC/XT/AT or PS/2 model 25 or 30 or any compatible ISA or EISA machine, including 8088, 286, 386 or 486 based systems. It occupies sixteen consecutive I/O addresses. The base I/O address can be set in the range 0h to FF0h with a DIP switch on the board.

The PC-14B features wait state generation circuitry. This ensures that all host computer's I/O bus cycles are long enough for the PC-14B. Zero, 1, 2, 4 or 8 additional wait states may be selected with a jumper block on the board.

The power supplies: +12V, +5V -12V, -5V and ground are available on the auxiliary connector of the board.

Features

- Forty eight lines of programmable digital inputs or outputs.
- Three modes of parallel data transfer.
 - Simple inputs and outputs.
 - Strobed inputs and outputs with handshaking.
 - Bidirectional input/output with handshaking.
- Three independent Timer/Counters.
- Timer Clock input programmable from four different sources.
- On board 2MHz reference oscillator.
- Programmable to allow PIAs and timers to generate interrupt requests.
- External interrupt input.
- Interrupt level of each source of interrupts independently selectable from IRQ 2 to IRQ 7.
- Jumper selectable wait state generator allows trouble free operation in all high speed systems.
- All external connections (including all power supplies) available directly at the rear of the host computer.
- Wide base address selection range; many boards may be paralleled.

- TTL and CMOS compatible
- Driver software supplied, complete with C and BASIC source code.
- Includes LabWindows drivers and C library modules.

PC-14B Package Contents

The PC-14B package should contain:

- PC-14B interface board and sufficient jumper caps for all possible operational configurations.
- The PC-14B User's Manual.
- An extension cable with mating connectors for the auxiliary connector on the board.
- A mating connector for the digital I/O connector (50 way) and a mating connector (DB25) for the bracketed auxiliary connector.
- The Data Acquisition Driver Software Manual.
- Two 5 $\frac{1}{4}$ " 360 Kb diskettes containing PC-14B utility and demonstration software and the Data Acquisition Driver Library.

If the package is missing any of the components or damaged in any way, then contact your distributor.

2

Installation

Configuration

There are three aspects of the PC-14B that can be configured. These are:

- The base address. This is the address where the computer will find the board. It is set to an address between 0h and FF0h with the eight switches on the DIP switch block. Note that this address is in the computer's I/O address space.
- Wait state generation. If the host computer generates abnormally fast I/O bus cycles then it may be necessary to slow these down when the computer accesses the PC-14B. The board can do this without affecting any other I/O bus cycles. The number of additional wait states inserted in a PC-14B bus cycle is controlled by a jumper on the board.
- The interrupt level of each sub-system of the board that generates interrupts. There are eight possible sources of PC-14B interrupts, and a corresponding jumper selection block for each.

Setting the base address

The PC-14B uses a block of sixteen I/O addresses. The base address setting controls the address where this block begins. This must be set so that the PC-14B does not use any addresses that are used by another device or card. If more than one PC-14B is to be installed in the computer then each card must have a different base address setting.

The base address may be assigned to any location from 0h to FF0h, on sixteen byte boundaries. The factory default base address is 300h. Refer to Table 1 below for a guide on addresses used by standard I/O devices and Table 6 in the appendices for a list of the base address switch settings.

The base address is set by adjusting the eight switches in the DIP switch block on the board. Switch number 1 is used to compare address line A11, number 2 for A10 and so on up to switch 8 which is used to compare address line A4. In general, if a switch is set to the off position, then its corresponding address weighting contributes to the base address. See Figure 1 for the DIP switch weightings.

For example switches 3 (corresponding to address line A9), 4 (corresponding to A8) and 8 (corresponding to A4) off yield a base address of $200h + 100h + 10h = 310h$.

Address (hex)	Standard Device
000-1FF	Internal system board
200-20F	Games port
210-217	Expansion unit
220-24F	Reserved
250-25F	Not assigned
258-25F	Intel "Above Board"
260-277	Not assigned
278-26F	Reserved
280-2EF	Not assigned
2F0-2F7	LPT2
2F8-2FF	COM2
300-31F	Prototype board
320-32F	Hard disk
330-377	Not assigned
378-37F	LPT1
380-38F	SDLC communications
390-39F	Not assigned
3A0-3AF	Binary communications
3B0-3BF	Monochrome display adapter
3C0-3CF	Reserved
3D0-3DF	Colour graphics adapter
3E0-3E7	Reserved
3E3-3EF	Not assigned
3F0-3F7	Floppy disk
3F8-3FF	COM1
400-FFF	Not used; refer to text

Table 1: Standard I/O Addresses

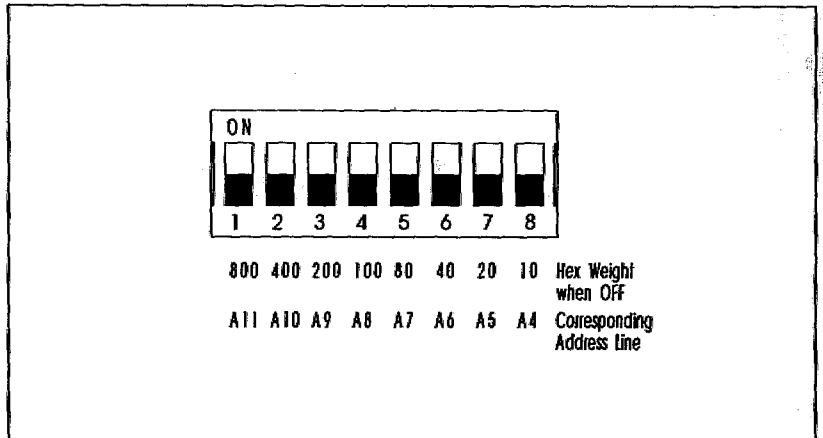


Figure 1: DIP Switch Address Weights [in hex].

Note that addresses from 400h to 7Fh cannot normally be used because these addresses are not normally decoded by other I/O devices and cards in the 0h to 3Fh range.

The PC-14B however (and most other members of the PC-XX family) can use these additional addresses if (and only if) there is no board or device at address 400h (or multiples of 400h) less than the address of the PC-14B or the board at the address 400h less also decodes the extra addresses.

A PC-14B may be installed at address 300h and another PC-14B at address 700h (400h locations apart). However it would not be advisable to install a PC-14B at address 778h because the printer port LPT1 uses a base address of 378h and does not normally decode the extra addresses.

Most other members of the PC-XX family of boards decode the extra addresses and may safely be installed 400h locations apart.

If your computer has boards not listed in Table 1 installed (such as LAN adapters, back-up boards or other engineering boards) then consult the manuals for these boards for information on the address ranges used. In most cases a base address of either 300h or 310h is a good choice. Address 310h is also the factory default address.

Generating additional wait states

The number of additional wait states inserted in I/O bus cycles addressing a PC-14B is controlled by the setting of the wait state jumper JP1. This is labelled on the board with the corresponding number of wait states. Refer to Figure 2 below for the wait state jumper positions.

The factory default setting is for zero wait states.

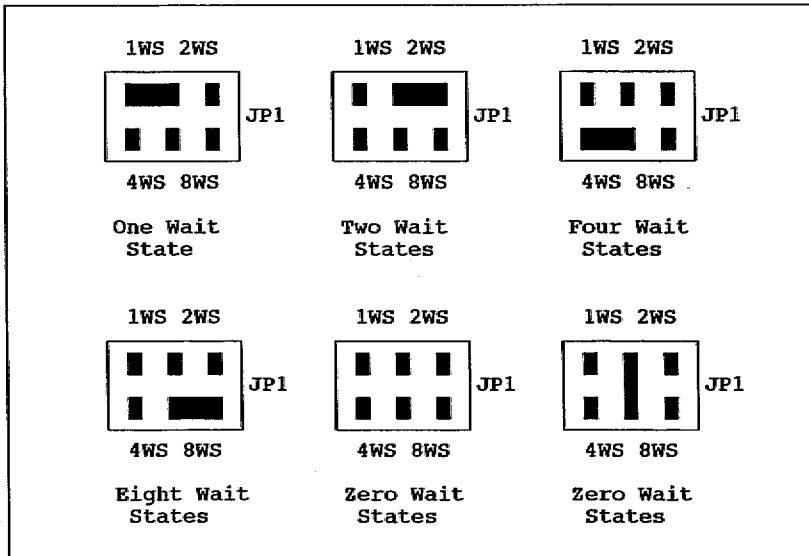


Figure 2: Wait State Jumper Settings

Only a very small number of computers (less than 5%) will require additional wait states. If the PC-14B card in your computer seems to be returning incorrect readings then increase the number of wait states until correct results are obtained. If the card does not produce correct results with the maximum number of additional wait states inserted then either the host computer or the card is faulty and should be serviced.

Setting the interrupt levels

Eight subsystems on the PC-14B that can generate interrupts to the host computer. They are:

- External interrupt input
- Timer 2 output
- Timer 1 output
- Timer 0 output
- INTR_A of the first 8255 (PIA0)
- INTR_B of the first 8255 (PIA0)
- INTR_A of the second 8255 (PIA1)
- INTR_B of the second 8255 (PIA1)

Full details on the function and use of the interrupts are given later in this manual.

Each of the eight sources of interrupts may be connected to any one of the six available interrupts on the PC's I/O channel. They are IRQ2, IRQ3, IRQ4, IRQ5, IRQ6 and IRQ7. Note that on ISA systems (for example the PC AT) IRQ2 is used by the system board itself and any interrupt requests on IRQ2 are transparently routed by the system board to IRQ9. In other words, IRQ2 on the I/O channel simply becomes IRQ9 for ISA systems. See to Table 2 below.

Interrupt Level	Standard Device	Available to PC-14
IRQ0	System Timer	No
IRQ1	Keyboard	No
IRQ2/IRQ9	Display Adaptor (VGA)	Yes
IRQ3	Serial Port 2	Yes
IRQ4	Serial Port 1	Yes
IRQ5	Parallel Port 2	Yes
IRQ6	Floppy Disk Controller	Yes
IRQ7	Parallel Port 1	Yes

Table 2: Standard Interrupt Allocations

When the PC-14B is shipped, all interrupt sources are set to their default level, as shown below in Figure 3. To change the interrupt level for an interrupt source, remove the jumper cap from its original position and place it over the pins for another interrupt request line. Each sub-system that can generate interrupts may only have one jumper cap installed at a time. To disable an interrupt source completely, do not install the jumper cap at all, or place it over only one pin of the jumper block.

The PC-14B can share interrupt lines with other devices and itself because it is possible to enable and disable the PC-14B's interrupt line drivers from software.

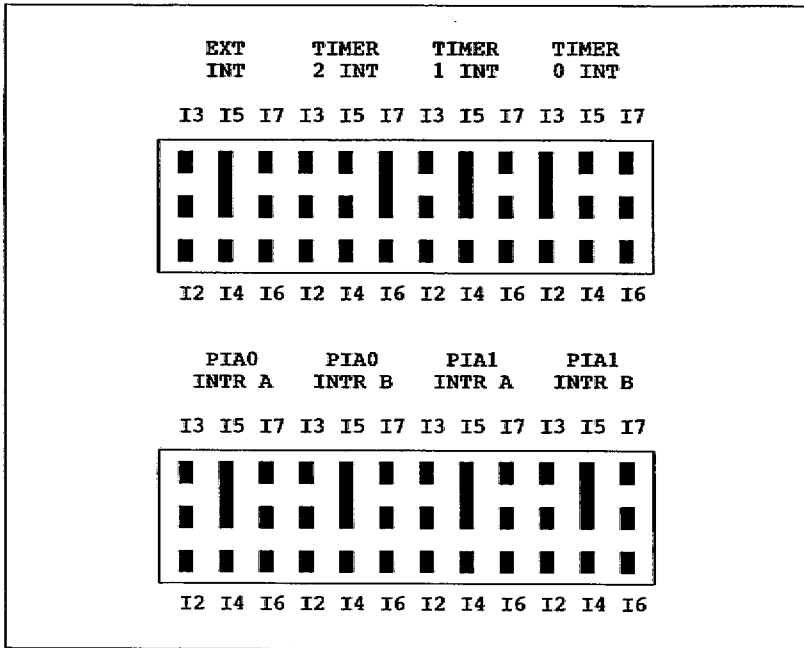


Figure 3 Interrupt Level Jumper Settings

Installation

Installing the PC-14 card is straightforward. You will need a screwdriver to match the screws on the computer's cover and expansion slot bracket.

- a) Switch off the computer and all attached devices.
- b) Unplug the power cord from the computer and all attached devices. Failure to do this may result in hazardous conditions, as there may be dangerous voltage levels present on externally connected cables.
- c) Remove the top cover of the PC or the access port to the I/O channel. If you are not sure how to do this, consult the manual supplied with the system unit.
- d) Choose any unused 8-bit or 16-bit expansion adapter slot and remove the screw from the top of the blank bracket corresponding to the chosen slot. Remove the bracket.
- e) Plug the auxiliary cable into the auxiliary connector in the board. Align the gold plated edge connector of the PC-14B with the edge socket on the computer system board and align the board bracket with the rear adapter slot on the PC case. Firmly press the board down into the edge socket. Ensure that the board's edge connector is seated in the socket and has not slipped sideways past the socket.

- f) Tighten the mounting bracket of the PC-14B to the back panel rail of the computer.
- g) Remove the bracket from another unused slot in the back panel rail of the computer and replace it with the mounting bracket of the auxiliary cable. Tighten it down. On some computer cases there are mounting holes for the DB25 connector on the auxiliary cable and the connector may be mounted there instead. This avoids having to use an expansion slot. Remove the mounting hole blank from the computer case, remove the bracket from the DB25 connector on the auxiliary cable and tighten the DB25 to the computer case with the screws that originally held the bracket to the DB25 connector.
- h) Replace the computer's cover. Plug in all cords and cables. Switch on the power. The PC-14B is now installed and ready for operation.

Interconnections

The PC-14B plugs into any of the computer's expansion slots, along the board's gold plated edge connector. The board communicates to the external world via a connector mounted in its bracket and an auxiliary DB25 connector. This chapter describes these three connectors and the necessary functions of these external lines.

Connections to the Computer Backplane

The PC-14B board may be plugged into any slot of the computer backplane (with the exception of the J8 slot of IBM XT's). All communication to and from the host computer is via this connector. The PC-14 board is situated in part of the host computer system's I/O space.

If the PC-14B sometimes behaves erratically then there is the possibility that the gold plated contacts on the edge connector may have become dirty or abraded, especially if the card is installed and removed many times from various different computers. This condition can be corrected simply by cleaning the contacts with an ordinary pencil eraser.

User Connectors

The PC-14B interfaces to the external world via a 50-pin male ribbon cable header mounted in the board's bracket, and a 25 way D-type male connector mounted in a blank bracket.

The 50-pin connector carries the following signals:

- The Port A and Port B digital input/output lines of both 8255s.
- The Port C digital input/output/handshake lines of both 8255s.
- Power supplies of +5V and digital ground.

The 25-pin D-type connector carries the following signals:

- Clock inputs, Gates and Clock outputs of the three timers
- External interrupt input
- Power supplies of +5V, -5V, +12V, -12V and digital ground.

Figure 4 shows these connections, together with their pin assignments. Note that the pin connections refer to the pin numbers of the connectors when looking into the connectors from the rear of the computer. Pin number 1 is marked with an arrow on the box surrounding the pins of the 50-way header connector. In the case of the D-type connector, the pin numbers are embossed onto the connector itself.

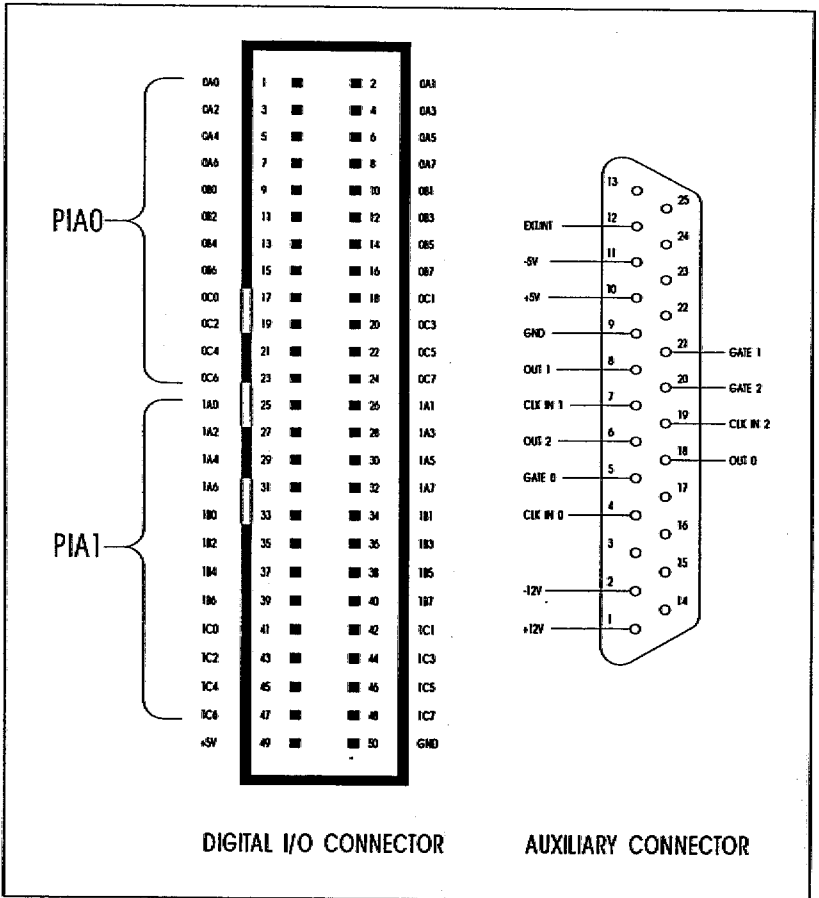


Figure 4: PC-14B connectors, as seen from the rear of the PC.

Some of the lines of Port C take on different meanings when an 8255 is programmed to operate in strobed input or output mode or bidirectional bus mode. They are summarised in the following table and described below that. See Chapter 6: *Programming Guide* for full details on the use of these signals and Appendix B for timing specifications.

Port C Line	Simple I/O: Mode 0	Strobed Input: Mode 1	Strobed Output: Mode 1	Bidirectional Bus: Mode 2
C7	I/O	I/O	/OBF _A	/OBF _A
C6	I/O	I/O	/ACK _A	/ACK _A
C5	I/O	IBF _A	I/O	IBF _A
C4	I/O	/STB _A	I/O	/STB _A
C3	I/O	INTR _A	INTR _A	INTR _A
C2	I/O	/STB _B	/ACK _B	I/O
C1	I/O	IBF _B	/OBF _B	I/O
C0	I/O	INTR _B	INTR _B	I/O

Table 3: Port C Line Usage

The symbol '/' indicates that a signal is active low.

When an 8255 is used in one of the handshaking modes, the /STB and IBF lines are used to synchronise input data transfers. The /OBF and /ACK lines are used to synchronise output transfers. The signals in the table above have the following functions:

Name	Type	Description
/STB	External Input	Strobe Input: A low on this handshaking line loads data from the peripheral bus into the input latch.
IBF	External Output	Input Buffer Full: A high on this line indicates that the external data has been loaded into the input latch. This is an input acknowledge signal.
/ACK	External Input	Acknowledge: The external device asserts this line low to indicate that the data written to the port by the program has been accepted.
/OBF	External Output	Output Buffer Full: The 8255 asserts this line low to indicate to the external device that the program has written data to the selected port. This line can be used to strobe the data into the external device.

Name	Type	Description
INTR	Internal Output	Interrupt Request: This signal becomes active (high) when the 8255 is requesting service from the host computer. For input operations, it indicates that there is data in the corresponding port to be read by the program. For output operations it indicates that the external device has accepted the data and thus the program can write another byte to the 8255. An interrupt line must be selected with the interrupt jumpers and the appropriate interrupt enable bits must be set, both in the 8255 and in the Interrupt Enable Register, to allow this signal to reach the host computer.
/RD	Internal Input	Read Signal: This signal is generated by the control lines of the host computer. It is activated when the program executes an Input Instruction from any PC-14B register.
/WR	Internal Input	Write Signal: This signal is generated by the control lines of the host computer. It is activated when the program executes an Output Instruction to any PC-14B register.

Power supplies

The +12V, -12V, +5V and -5V power supplies are available, together with digital ground, from the auxiliary connector. These can be used to power external circuitry. As a rough guide, no more than about 250mA should be drawn from these power supplies or the PC-14 board, the computer or the power supply may be permanently damaged. Check the rating of the host computer's power supply for details on maximum current loading, especially on the -5V and -12V lines.

Architecture

This chapter describes the architecture of the PC-14B board. There are three functional sub-systems on the board: the digital I/O ports, the timer/counters and the external interrupt. In addition to these, there are some support sub-subsystems and I/O connectors. This chapter provides a brief overview of all these sub-systems, and explains their function and operation. Each part is described fully in appropriate sections later in this manual.

The block diagram below illustrates the architecture of the PC-14B.

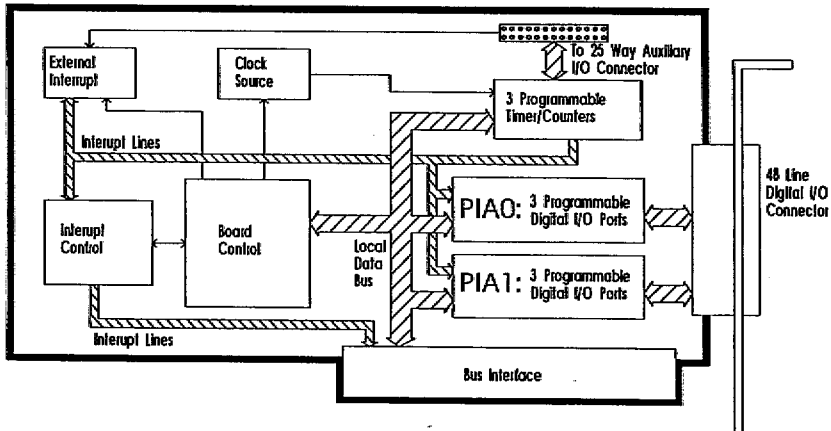


Figure 5: PC-14B Block Diagram

Functional Sub-Systems

Digital I/O Ports

Two 8255 PIA devices on the PC-14B board each provide three 8-bit digital I/O ports. They are named port A, port B and port C. Three modes of data transfer are available: simple polled I/O, strobed I/O and bidirectional bus I/O. The latter two modes use external handshaking lines to synchronise data transfers.

Interrupt controlled data transfers are supported.

Timer/Counters

An 8254 Timer/Counter device integrated into the PC-14B provides three independent timer/counters. A counter counts clock pulses from a programmable source. Counting/triggering is performed using a counter's gate input. The counter's output pin indicates the status/result of counting.

The computer program may latch and read the current count at any time.

There are six different timing/counting modes available.

The timer outputs may generate interrupt requests to the host computer.

External Interrupt

The external interrupt allows an external system, via a signal line, to interrupt the host computer on demand.

The interrupt input is latched, and remains active until serviced.

Support Sub-Systems

Bus Interface

The bus interface provides for the base address of the PC-14B to be set, using an on-board DIP switch, within wide range of PC I/O addresses.

It also provides bus transceivers and buffers which maintain proper bus flow discipline between the PC-14B and the host computer.

In the event of the host computer's I/O bus cycles being abnormally short, this sub-system is, under control of a selection jumper, able to insert additional wait states into any I/O cycle that is directed at the PC-14B board.

Interrupt Control

There are eight sources of interrupts on the PC-14B. This sub-system controls routing and enabling of the interrupt sources.

A source of interrupts from a device on the PC-14B is routed, via selection jumpers, to an interrupt request line on the PC I/O channel.

All interrupts from the board can be enabled or disabled from software by writing appropriate bytes to corresponding PC-14B registers.

Board Control

This is an internal sub-system that is responsible for the general housekeeping of the board.

Clock Source

Each of the three timer/counters may receive clock pulses from one of four sources. This sub-system is responsible for routing a source of clock pulses to each timer and, depending on the exact clock source, for either generating or formatting the clock source. It also provides a software controlled gate for each timer. All aspects of the clock sources are independently programmable, on the fly, from within software.

I/O Connectors

Digital I/O Connector

All 48 digital I/O lines, together with the digital power supplies, are transmitted through a standard 50-way male IDC header connector. This allows quick cable assembly and mass termination.

Auxiliary I/O Connector

All other signals are available from this connector. They are: the timer/counter lines, the external interrupt line and all power supplies.

As there is not sufficient space on the PC-14B's bracket for these connections, they are brought out to the rear of the computer via a short intermediate cable connected to a standard 25-way male D-type connector mounted in a blank bracket.

Register Structure

At the lowest level, the PC-14B is programmed using I/O port input and output instructions. This chapter contains information on all of the PC-14B's registers. Although programming the board is not difficult, it is time consuming and requires detailed knowledge of both the PC-14 and the host PC's operating system. This manual provides the former. As an alternative, a set of software library functions is provided. These cover all of the board's features and should suit most applications. The Driver Library is described in a companion manual to this. In case routines need to be custom written for a specific purpose, it is often quicker to base them on the driver system, by loading and modifying routines from the driver library source code. This chapter and Chapter 6: *Programming Guide* provide full information on how this is done.

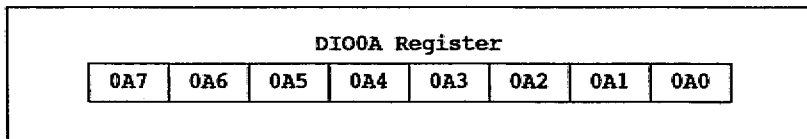
The PC-14B occupies 15 consecutive addresses in the computer's I/O space. The layout of these registers is shown in Table 4: *PC-14B Register Structure*. This table also gives the page number of this manual on which the register is described. The offset of each register is given as an offset address from the base address of the board. The board's base address is set with the DIP switch as detailed in Chapter 2: *Installation*.

Offset	Write	Read	Page No
0	PIA 0: Port A (DIO0A)	PIA 0: Port A (DIO0A)	28
1	PIA 0: Port B (DIO0B)	PIA 0: Port B (DIO0B)	28
2	PIA 0: Port C (DIO0C)	PIA 0: Port C (DIO0C)	28
3	PIA 0: Control Register (DIO0CTRL)	---	29
4	PIA 1: Port A (DIO1A)	PIA 1: Port A (DIO1A)	30
5	PIA 1: Port B (DIO1B)	PIA 1: Port B (DIO1B)	30
6	PIA 1: Port C (DIO1C)	PIA 1: Port C (DIO1C)	31
7	PIA 1: Control Register (DIO1CTRL)	---	31
8	Timer/Counter 0 (TC0)	Timer/Counter 0 (TC0)	33
9	Timer/Counter 1 (TC1)	Timer/Counter 1 (TC1)	33
10	Timer/Counter 2 (TC2)	Timer/Counter 2 (TC2)	33
11	Timer/Counter Control Register (TCCTRL)	---	34
12	Clock Source Register (CSR)	---	36
13	Counter/Interrupt Enable Register (CER)	---	37
14	External Interrupt Reset (IRQRST)	---	38

Table 4: PC-14B Register Structure.

DIO0A - Port A of the first PIA (offset 0, read/write)

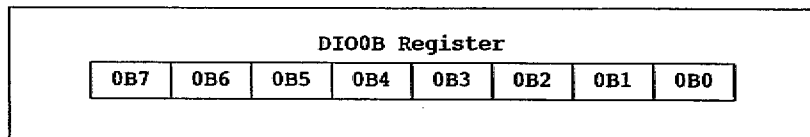
This register is port A of the first PIA. It can be operated in one of three modes as described in Chapter 6: *Programming Guide*. The operating mode is set by writing to the DIOCTRL register, described below.



The bits 0A7 (MSB) down to 0A0 (LSB) reflect the status of the port's I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs or bidirectional.

DIO0B - Port B of the first PIA (offset 1, read/write)

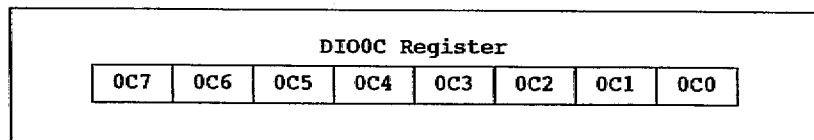
This register is port B of the first PIA. It can be operated in one of three modes as described in Chapter 6: *Programming Guide*. The operating mode is set by writing to the DIOCTRL register, described below.



The bits 0B7 (MSB) down to 0B0 (LSB) reflect the status of the port's I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs or bidirectional.

DIO0C - Port C of the first PIA (offset 2, read/write)

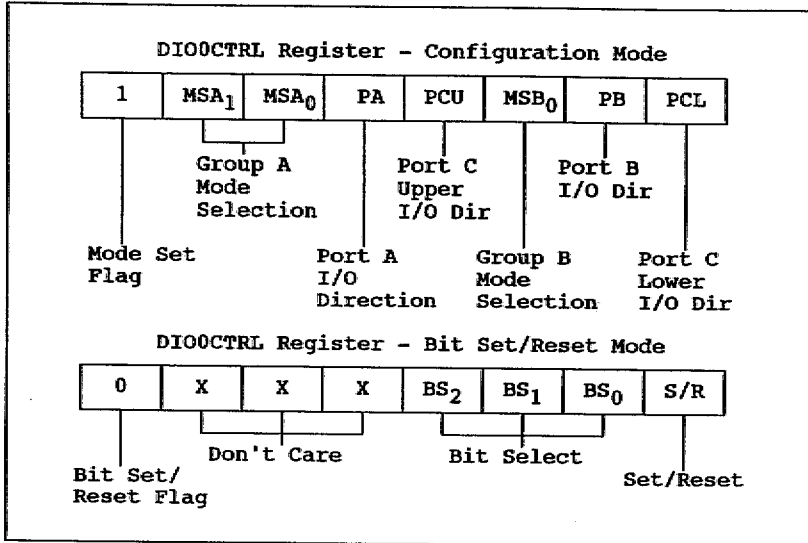
This register is port C of the first PIA. It can be operated in one of three modes as described in Chapter 6: *Programming Guide*. The operating mode is set by writing to the DIOCTRL register, described below.



The bits 0C7 (MSB) down to 0C0 (LSB) reflect the status of the port's I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs, handshakes or interrupt request lines.

DIOCTRL - Control register of the first PIA (offset 3, write only)

This register is used to control the operating modes of the first PIA or to set and reset individual bits in port C of the PIA. The layout of this register is shown below. Note that the function and bit names of the register depend on the setting of bit 7.



Bit 7: Function select: If this bit is set when the register is written to then the register is in configuration mode. If the bit is 0 then the register is in bit set/reset mode.

The functions of the remaining bits are described below, depending on the setting of bit 7.

Configuration Mode

Bits 6-5: Group A mode select: These two bits set the mode of the group A ports. These are port A and the upper four lines of port C. The bit combinations are as follows:

MSA ₁	MSA ₀	Group A I/O Mode
0	0	Mode 0, simple I/O
0	1	Mode 1, strobed I/O
1	X	Mode 2, bidirectional bus

Bit 4: Port A I/O Direction: If this bit is set, then port A functions as an input. If it is 0, then port A is configured as an output.

Bit 3: Port C Upper I/O Direction: If this bit is set, then the upper four lines of port C function as inputs. If the bit is 0, then the lines become outputs.

Bit 2: Group B mode select: This bit sets the mode of the group B ports. These are port B and the lower four lines of port C. The bit combinations are as follows:

MSB ₀	Group B I/O Mode
0	Mode 0, simple I/O
1	Mode 1, strobed I/O

Note that group B can only be used for simple or strobed I/O.

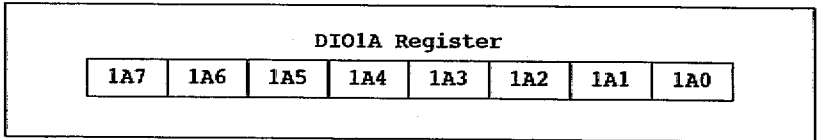
- Bit 1: Port B I/O Direction: If this bit is set, then port B functions as an input. If it is 0, then port B is configured as an output.
- Bit 0: Port C Lower I/O Direction: If this bit is set, then the lower four lines of port C function as inputs. If the bit is 0, then the lines become outputs.

Bit Set/Reset Mode

- Bits 6-5: These bits have no effect in this function.
- Bits 3-1: Bit Select: these bits select the bit in port C which is to be modified. A code of 000 selects port C line 0 to set or reset, 001 selects line 1 and so on up to 111 which selects line 7.
- Bit 0: Set/Reset: This bit specifies the state into which the selected port C line will be placed. Writing a 1 will make the line go high and a 0 makes it go low. This operation has no effect on the other lines of port C.

DIO1A - Port A of the second PIA (offset 4, read/write)

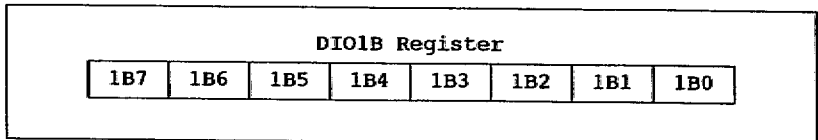
This register is port A of the second PIA. It can be operated in one of three modes as described in Chapter 6: *Programming Guide*. The operating mode is set by writing to the DIO1CTRL register, described below.



The bits 1A7 (MSB) down to 1A0 (LSB) reflect the status of the port's I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs or bidirectional.

DIO1B - Port B of the second PIA (offset 5, read/write)

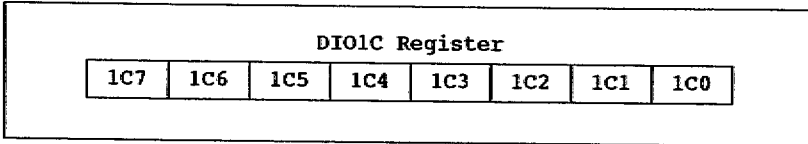
This register is port B of the second PIA. It can be operated in one of three modes as described in Chapter 6: *Programming Guide*. The operating mode is set by writing to the DIO1CTRL register, described below.



The bits 1B7 (MSB) down to 1B0 (LSB) reflect the status of the port's I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs or bidirectional.

DIO1C - Port C of the second PIA (offset 6, read/write)

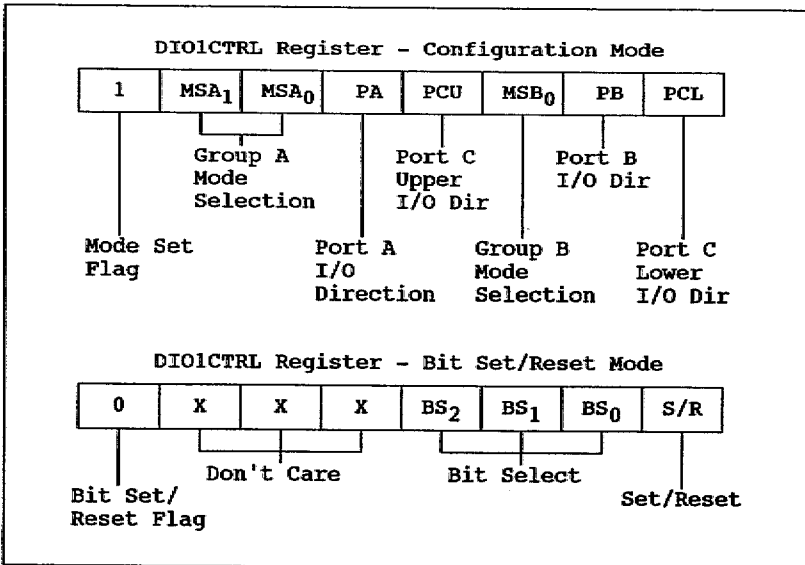
This register is port C of the second PIA. It can be operated in one of three modes as described in Chapter 6: *Programming Guide*. The operating mode is set by writing to the DIO1CTRL register, described below.



The bits 1C7 (MSB) down to 1C0 (LSB) reflect the status of the port's I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs, handshakes or interrupt request lines.

DIO1CTRL - Control register of the second PIA (offset 7, write only)

This register is used to control the operating modes of the second PIA or to set and reset individual bits in port C of the PIA. The layout of this register is shown below. Note that the function and bit names of the register depend on the setting of bit 7.



Bit 7 : Function select: If this bit is set when the register is written to then the register is in configuration mode. If the bit is 0 then the register is in bit set/reset mode.

The functions of the remaining bits are described below, depending on the setting of bit 7.

Configuration Mode

Bits 6-5 : Group A mode select: These two bits set the mode of the group A ports. These are port A and the upper four lines of port C. The bit combinations are as follows:

MSA ₁	MSA ₀	Group A I/O Mode
0	0	Mode 0, simple I/O
0	1	Mode 1, strobed I/O
1	X	Mode 2, bidirectional bus

Bit 4 : Port A I/O Direction: If this bit is set, then port A functions as an input. If it is 0, then port A is configured as an output.

Bit 3 : Port C Upper I/O Direction: If this bit is set, then the upper four lines of port C function as inputs. If the bit is 0, then the lines become outputs.

Bit 2 : Group B mode select: This bit sets the mode of the group B ports. These are port B and the lower four lines of port C. The bit combinations are as follows:

MSB ₀	Group B I/O Mode
0	Mode 0, simple I/O
1	Mode 1, strobed I/O

Note that group B can only be used for simple or strobed I/O.

Bit 1 : Port B I/O Direction: If this bit is set, then port B functions as an input. If it is 0, then port B is configured as an output.

Bit 0 : Port C Lower I/O Direction: If this bit is set, then the lower four lines of port C function as inputs. If the bit is 0, then the lines become outputs.

Bit Set/Reset Mode

Bits 6-5 : These bits have no effect in this function.

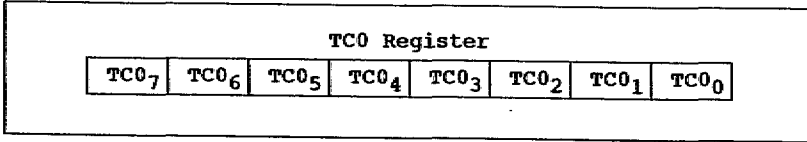
Bits 3-1 : Bit Select: these bits select the bit in port C which is to be modified. A code of 000 selects port C line 0 to set or reset, 001 selects line 1 and so on up to 111 which selects line 7.

Bit 0 : Set/Reset: This bit specifies the state into which the selected port C line will be placed. Writing a 1 will make the line go high and a 0 makes it go low. This operation has no effect on the other lines of port C.

TC0 - Timer/Counter 0 (offset 8, read/write)

This register is the data register of the first timer/counter.

Before reading or writing from this register, a Control Word for this counter must be written to the timer/counter Control Register. Then reads and writes to this register must follow the format specified in the control word.

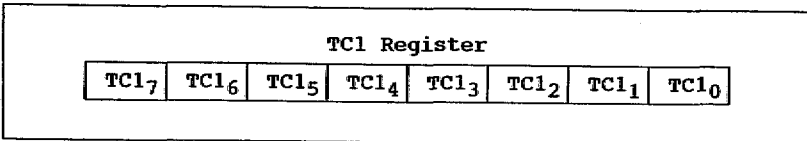


The bits TC0₇ (MSB) down to TC0₀ (LSB) reflect the high byte or the low byte of the data read or written to this counter.

TC1 - Timer/Counter 1 (offset 9, read/write)

This register is the data register of the second timer/counter.

Before reading or writing from this register, a Control Word for this counter must be written to the timer/counter Control Register. Then reads and writes to this register must follow the format specified in the control word.

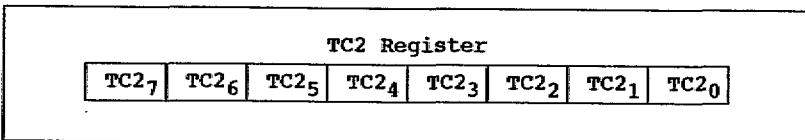


The bits TC1₇ (MSB) down to TC1₀ (LSB) reflect the high byte or the low byte of the data read or written to this counter.

TC2 - Timer/Counter 2 (offset 10, read/write)

This register is the data register of the third timer/counter.

Before reading or writing from this register, a Control Word for this counter must be written to the timer/counter Control Register. Then reads and writes to this register must follow the format specified in the control word.

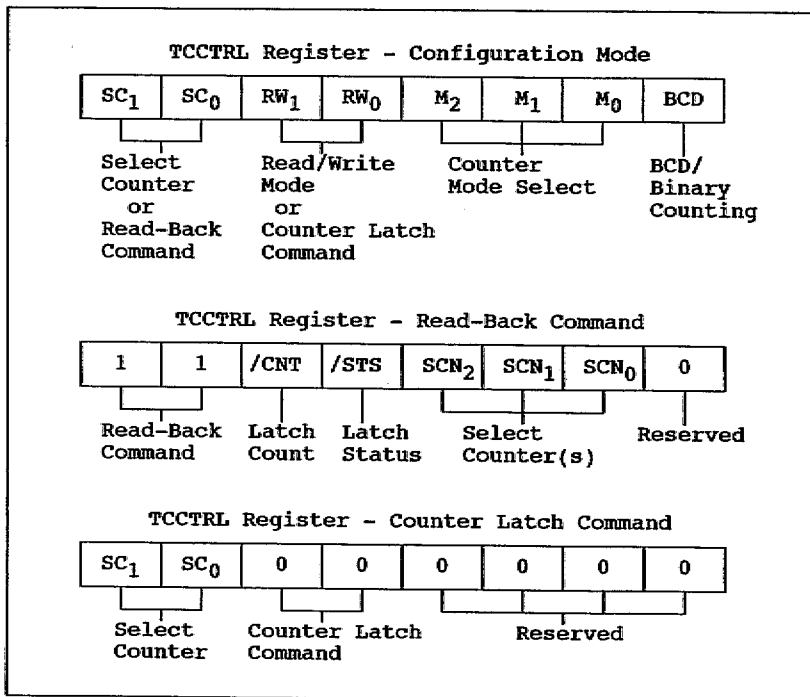


The bits TC2₇ (MSB) down to TC2₀ (LSB) reflect the high byte or the low byte of the data read or written to this counter.

TCCTRL - Timer/Counter Control Register (offset 11, write only)

The Timer/Counter Control Register is used to program, for each counter, the counting mode, the number of bytes to read/write and whether the counter counts in BCD or binary format. In addition, this register can be used to perform Read-Back Commands and Counter Latch Commands. Note that the function and bit names of this register differ according to whether Configuration Mode, Counter Latch Command or Read-Back Command is selected with bits 7-6 (SC_1 and SC_0) and bits 5-4 (RW_1 and RW_0).

See page 49 in Chapter 6 for details on the Read-Back and Counter Latch Commands and how the different counter modes function.



The functions of the remaining bits are described below, depending on the setting of bits 7 and 6 and bits 5 and 4.

Configuration Mode

Bits 7-6: Select Counter (SC): These two bits select the timer/counter to which the rest of the TCCTRL register bits will apply. The SC bits are defined as follows:

SC ₁	SC ₀	Operation Performed
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command

Bits 5-4: Read/Write Mode (RW): These two bits select the read/write mode of the selected timer/counter. The RW bits are defined as follows:

RW ₁	RW ₀	Operation Performed
0	0	Counter Latch Command
0	1	Read/Write Least Significant Byte only
1	0	Read/Write Most Significant Byte only
1	1	Read/Write Least Significant Byte first, then Most Significant Byte

Bits 3-1: Counter Mode Select (M): These three bits select the operating mode of the timer/counter selected with bits 7-6. The Mode bits are defined as follows:

M ₂	M ₁	M ₀	Operating Mode
0	0	0	Mode 0
0	0	1	Mode 1
0	1	0	Mode 2
0	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

Bit 0: Counting Mode Select (BCD): This bit determines whether the selected counter is to count in BCD or binary format. A 0 specifies 16-bit binary counting and a 1 specifies 4 decade Binary Coded Decimal counting.

Read-Back Command

When the Read-Back Command is specified, the bit definitions of the TCCLR Register are:

- Bits 7-6:** These bits must both be set to 1 to invoke the Read-Back Command.
- Bit 5:** Count (/CNT): Setting this bit to 0 causes the timer/counter chip to latch the count(s) of the counters selected with bits 3-1 of this register (see below).
- Bit 4:** Status (/STS): Setting this bit to 0 causes the timer/counter chip to latch certain status information from the counters selected with bits 3-1 of this register (see below). The format of the status byte is discussed on page 52 in Chapter 6.
- Bits 3-1:** Select Counter (SCN): Setting one or more of these bits causes the corresponding counter to latch its count and/or status information, when the Read-Back Command is issued. Setting SCN₂ latches counter 2 information, SCN₁: counter 1 and SCN₀: counter 0.
- Bit 0:** This bit performs no function and should be set to 0.

Counter Latch Command

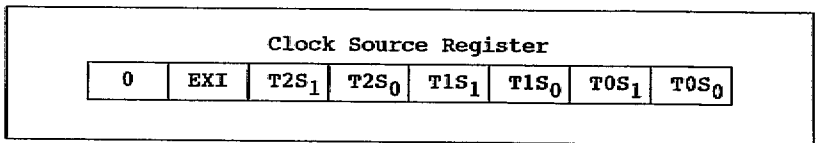
The bit definitions of the TCCLR Register, when the Counter Latch Command is specified, are as follows:

- Bits 7-6 : Select Counter (SC): These two bits select the timer/counter whose count is to be latched.
- Bits 5-4 : These two bits must both be set to 0 to specify the Counter Latch Command.
- Bits 3-0 : These four bits perform no function and must all be set to 0.

This above is a summary of the 8254 timer/counter registers. See page 49 in Chapter 6 for definitions on the Read-Back and Counter Latch Commands and more information on programming the 8254, especially in relation to the additional PC-14B hardware.

CSR - Clock Source Register (offset 12, write only)

The Clock source register is used to select the source of the Clock inputs to each of the timer/counters. It is also used to enable interrupts from the external interrupt. On power up or reset, all the bits in this register are reset to 0.



- Bit 7 : Reserved: This bit is reserved for future use or manufacturing tests. Write a 0 to this bit for future compatibility.
- Bit 6 : Enable External Interrupt (EXI): Writing a 1 to this bit enables interrupts from the external interrupt input onto the PC bus. Writing a 0 disables external interrupts. The interrupt level of this interrupt is selected by jumper. Refer to Chapter 2: *Installation* for information on setting the interrupt level.
- Bits 5-4 : Timer 2 Clock Source (T2S): These two bits specify the source of the Clock input for timer/counter number 2. The Clock source is defined as follows:

T2S ₁	T2S ₀	Timer 2 Clock Source
0	0	PC I/O clock divided by 64
0	1	PC I/O clock divided by 4
1	0	On-board 2MHz reference oscillator
1	1	Timer 2 External clock input

On power up or reset, the Timer 2 Clock source defaults to the PC I/O clock divided by 64.

- Bits 3-2 : Timer 1 Clock Source (T1S): These two bits specify the source of the Clock input for timer/counter number 1. The Clock source is defined as follows:

T1S ₁	T1S ₀	Timer 1 Clock Source
0	0	PC I/O clock divided by 64
0	1	PC I/O clock divided by 4
1	0	On-board 2MHz reference oscillator
1	1	Timer 1 External clock input

On power up or reset, the timer 1 clock source defaults to the PC I/O clock divided by 64.

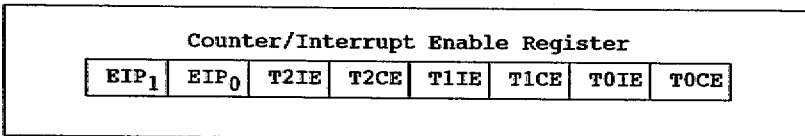
Bits 1-0: **Timer 0 Clock Source (T0S):** These two bits specify the source of the clock input for timer/counter number 0. The clock source is defined as follows:

T0S ₁	T0S ₀	Timer 0 Clock Source
0	0	PC I/O clock divided by 64
0	1	PC I/O clock divided by 4
1	0	On-board 2MHz reference oscillator
1	1	Timer 0 External clock input

On power up or reset, the timer 0 clock source defaults to the PC I/O clock divided by 64.

CER - Counter/Interrupt Enable Register (offset 13, write only)

This register is used to provide an independent software gate for each timer/counter, and to enable the output of the timers to generate interrupts to the host computer. In addition, it is used to enable interrupts from the 8255s onto the PC bus. On power up or reset, all the bits in this register are reset to 0.



Bit 7: **Enable Interrupts from PIA1 (EIP1):** Setting this bit enables the two interrupt outputs from the second 8255 onto the PC bus. With the PIA in mode 1 or mode 2, this electrically connects the INTR_A and INTR_B lines (the port C3 and port C0 lines respectively) to the selected interrupt lines of the host computer. Writing a 0 to this bit disables the two interrupts from the PC I/O channel. Refer to Chapter 2: *Installation* for information on setting the individual interrupt levels.

Note that the port C3 and port C0 lines are enabled onto the selected PC I/O bus IRQ line by this bit regardless of the mode setting of the 8255. What this means is that if the port C3 and port C0 lines are not being used for digital I/O they can be used as additional unlatched external interrupt inputs.

**Additional
External
Interrupts**

Bit 6: **Enable Interrupts from PIA0 (EIP0):** Setting this bit enables the two interrupt outputs from the first 8255 onto the PC bus. With the PIA in mode 1 or mode 2 this electrically connects the INTR_A and INTR_B lines (the port C3 and port C0 lines respectively) to the selected interrupt lines of the host computer. Writing a 0 to this

**Additional
External
Interrupts**

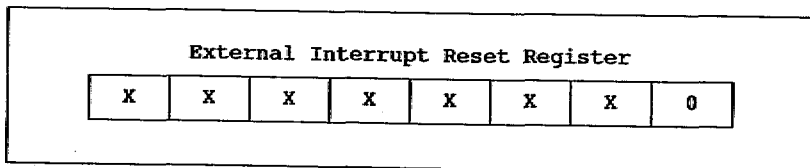
bit disables the two interrupts from the PC I/O channel. Refer to Chapter 2: *Installation* for information on setting the individual interrupt levels.

Note that the port C3 and port C0 lines are enabled onto the selected PC I/O bus IRQ line by this bit regardless of the mode setting of the 8255. What this means is that if the port C3 and port C0 lines are not being used for digital I/O they can be used as additional unlatched external interrupt inputs.

- Bit 5:** Timer 2 Interrupt Enable (T2IE): Setting this bit enables the output of timer 2 onto a jumper selected IRQ line of the PC bus. Writing a 0 to this bit disables the timer output from generating interrupts.
- Bit 4:** Timer 2 Clock input Enable (T2CE): With this bit clear, clock pulses (from one of the four clock sources as set with T2S₁ and T2S₀ (bits 5 and 4) in the Clock Source Register) are allowed to reach the clock input of timer 2. Writing a 1 to this bit disables the input of clock pulses to timer 2 but has no effect on anything else. This bit forms a software gate on timer 2.
- Bit 3:** Timer 1 Interrupt Enable (T1IE): Setting this bit enables the output of timer 1 onto a jumper selected IRQ line of the PC bus. Writing a 0 to this bit disables the timer output from generating interrupts.
- Bit 2:** Timer 1 Clock input Enable (T1CE): With this bit clear, clock pulses (from one of the four clock sources as set with T1S₁ and T1S₀ (bits 3 and 2) in the Clock Source Register) are allowed to reach the clock input of timer 1. Writing a 1 to this bit disables the input of clock pulses to timer 1 but has no effect on anything else. This bit forms a software gate on timer 1.
- Bit 1:** Timer 0 Interrupt Enable (T0IE): Setting this bit enables the output of timer 0 onto a jumper selected IRQ line of the PC bus. Writing a 0 to this bit disables the timer output from generating interrupts.
- Bit 0:** Timer 0 Clock input Enable (T0CE): With this bit clear, clock pulses (from one of the four clock sources as set with T0S₁ and T0S₀ (bits 1 and 0) in the Clock Source Register) are allowed to reach the clock input of timer 0. Writing a 1 to this bit disables the input of clock pulses to timer 0 but has no effect on anything else. This bit forms a software gate on timer 0.

IRQRST - External Interrupt Reset (offset 14, write only)

This register is used to clear the latched external interrupt.



- Bits 7-1:** These bits have no effect in this function, and zeros should be written to them for future compatibility.
- Bit 0:** Writing a 0 to this bit clears the external interrupt latch thus resetting the external interrupt. Writing a 1 to this bit is not defined.

6

Programming Guide

This chapter describes programming the PC-14B board at the lowest level. While this is fairly straightforward, it requires detailed knowledge of the system hardware. As an alternative, the driver software supplied with the board can be used. Then the user need not read this section and can refer instead to the *Data Acquisition Driver Software Manual*, which describes the driver software which accompanies the PC-14B. Advantages of using the supplied driver are:

- Detailed knowledge is not required of the board, the host computer or the interface software.
- The Driver Library is callable from most high level libraries.
- The Driver Library takes into account multiple boards (both the PC-14B and other supported boards) in the same computer.

Users and developers who need to incorporate special routines (especially interrupt handlers) into their application will need to read this chapter.

Once the PC-14 has been installed in the computer and the external connections made, the board is in an operational state. The PC-14B occupies fifteen consecutive I/O addresses, starting at the board's base address as set by the base address DIP switch. Programming the PC-14B is done solely with port input and output instructions. Reading and writing to its addresses allows the board to be configured and data transferred to and from the board. Refer to Chapter 5 for the I/O register map of the PC-14B.

Reading or writing to the ports typically takes the form of the one following instructions:

Language	Port Read	Port Write
'C	<code>data = inp(addr);</code>	<code>outp(addr, data);</code>
Pascal	<code>data := port[addr];</code>	<code>port[addr] := data;</code>
Assembly Language	<code>mov al, data mov dx, addr in al, dx</code>	<code>mov al, data mov dx, addr out dx, al</code>

Where: `addr` is the address in the address map of the PC-14 and
`data` is the byte read or written.

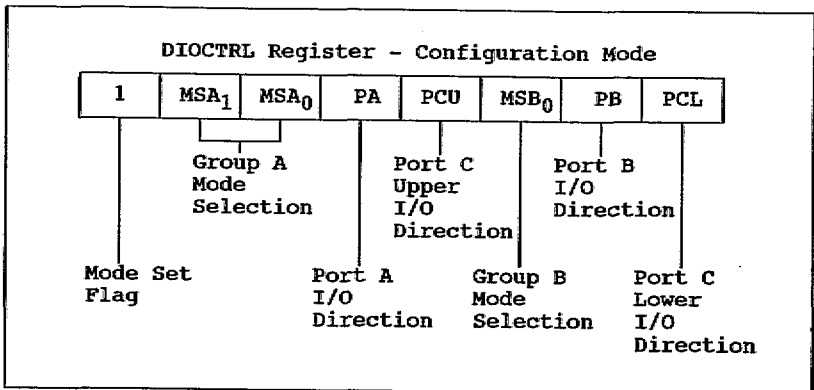
Programming the 8255 PIA

One of the major sub-systems of the PC-14 is based around two industry standard 8255 Peripheral Interface Adaptor (PIA) integrated circuits. This chapter provides detailed programming information on the 8255 for the PC-14B. The two 8255 devices and their support circuitry on the PC-14 are identical in all respects and totally independent from one another. Thus the information provided may be used to program either of the 8255s.

Introduction

The 8255 is a general purpose peripheral interface having 24 programmable I/O lines. The 24 lines are divided into three 8-bit ports (named port A, port B and port C). The ports may be programmed either as two groups of 12 lines (group A and group B) or as the three individual 8-bit ports. The ports/groups may be operated in three modes: simple I/O; strobed I/O and bidirectional I/O. The 8255 also has a single bit set/reset feature for port C.

Before any I/O can be done, the mode and direction of the ports/groups of the 8255 must be set. This is done simply by writing one byte of configuration information to the control register of the 8255. The 8255 then operates in the specified mode until it is reset or new configuration information is written to the control register. The format of this register is repeated below for reference.



Bit 7 of the DIOCTRL register must be set when programming the configuration of the 8255.

The mode is set for each of the two groups of the 8255 and the direction is set for the individual ports. Note that the direction of port C is independently programmable in two 4-line nibbles.

Mode 0: Simple I/O

This mode is used for simple input and output operations for each of the ports. No handshaking is required and no interrupts are generated. Data is simply read from or written to a selected port.

The following characterise Mode 0:

- Two 8-bit ports (ports A and B) and two 4-bit ports (upper and lower nibble of port C).
- Any port can be configured for input or output.
- Outputs are latched, inputs are not latched.
- Data transfer by polled I/O.

Mode 0 Programming

To use the 8255 in mode 0:

- Write a single byte to the control register to set the 8255 into mode 0 with the three ports configured for the desired data direction.
- Then read or write from the I/O port corresponding to an 8255 port (port A, B or C) as many times as necessary to obtain or transfer the required amount of data.

Mode 0 Programming Examples

```
#define BaseAddr    0x300    /* PC-14 base address at 300 hex */
#define DIO0A      0        /* Offset of port A of the first 8255 */
#define DIO0B      1        /* Offset of port B */
#define DIO0C      2        /* Offset of port C */
#define DIO0CTRL   3        /* Offset of the control register */

void main( void )
{
  unsigned int port0a, port0b, port0c, port0ctrl, data, outval = 0x1B;

  /* Calculate register addresses */
  port0a = BaseAddr + DIO0A;
  port0b = BaseAddr + DIO0B;
  port0c = BaseAddr + DIO0C;
  port0ctrl = BaseAddr + DIO0CTRL;

  /* 0x90 = 1001 0000, port A to input, ports B and C for output */
  outp( port0ctrl, 0x90 );

  /* Read data from port A and write data to port B and C */
  data = inp( port0a );
  outp( port0b, 0x3F );
  outp( port0c, outval );
}
```

Mode 1: Strobed I/O

In this mode data transfers are controlled by handshaking signals and hardware interrupts. Some of the port C lines are used for these control signals. Hence they take on different functions and names. Refer to page 21 in Chapter 3 for a description of these signal's functions.

The following characterise Mode 1:

- Two groups, group A and B. Each group consists of an 8-bit data port and three control lines.
- Certain port C lines take on special functions.
- The data ports can be either input or output ports.
- Both inputs and outputs are latched.
- One 2-bit simple I/O port.
- Data transfer by interrupts or polled I/O.

With both groups configured in mode 1, a single 8255 can read or write data 16-bits wide.

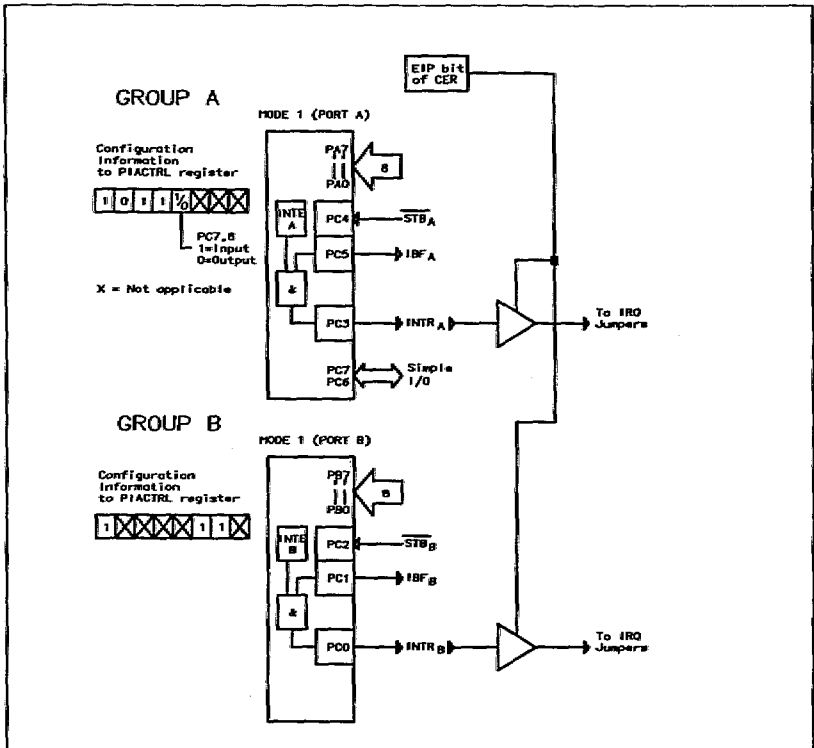


Figure 6: Mode 1 Input of an 8255.

Mode 1 Programming

To use the 8255 in mode 1 input with interrupts:

- Write a byte to the control register to configure the 8255 for mode 1 and the appropriate group for data input.
- With the bit set/reset operation, write a 1 to the Interrupt Enable flip-flop (INTE) of the desired port of the appropriate 8255.
- Write a byte to the Counter/Interrupt Enable Register (CER) to enable interrupts from the appropriate 8255 to the host computer.
- The external device pulses the Strobe (/STB) input line on the digital I/O connector low. The trailing edge of this loads data into the input port.
- The Input Buffer Full (IBF) output line goes high on the digital I/O connector to indicate that the data has been loaded into the input latch.
- When the external device pulls the /STB line high, the Interrupt Request line (INTR) goes high. This indicates to the host computer that there is data to be read from the 8255.
- The computer reads the data using an interrupt service routine (ISR) and by doing so, automatically resets the INTR signal.
- The external device can now pulse the /STB low again to make the PC-14B read another byte of data and hence repeat the cycle.

Whenever a group of the 8255 is in mode 1 input, the status of the handshaking lines and interrupt signals can be obtained by reading port C. The byte read contains the following information:

Port C Mode 1 Input Status Information							
C7	C6	IBF _A	INTE _A	INTR _A	INTE _B	IBF _B	INTR _B

The 8255 may alternatively be used in mode 1 and the data read by polled (program) transfer. This is done as follows:

- Write a byte to the control register to configure the 8255 for mode 1 and the appropriate group for data input.
- The program continually monitors the IBF line by reading port C.
- The external device pulls the /STB input line on the digital I/O connector low and this loads data into the input port.
- The IBF output line goes high on the digital I/O connector to indicate that the data has been loaded into the input latch.
- This also causes the corresponding IBF bit in port C to be set and the program can thus read the data.
- The external device can now pull the /STB low again to make the program read another byte of data and hence repeat the cycle.

The program could also enable the INTR line with the INTE flip-flop and then monitor INTR instead of the IBF line. In this case, interrupts to the host computer are not enabled in the Counter/Interrupt Enable Register.

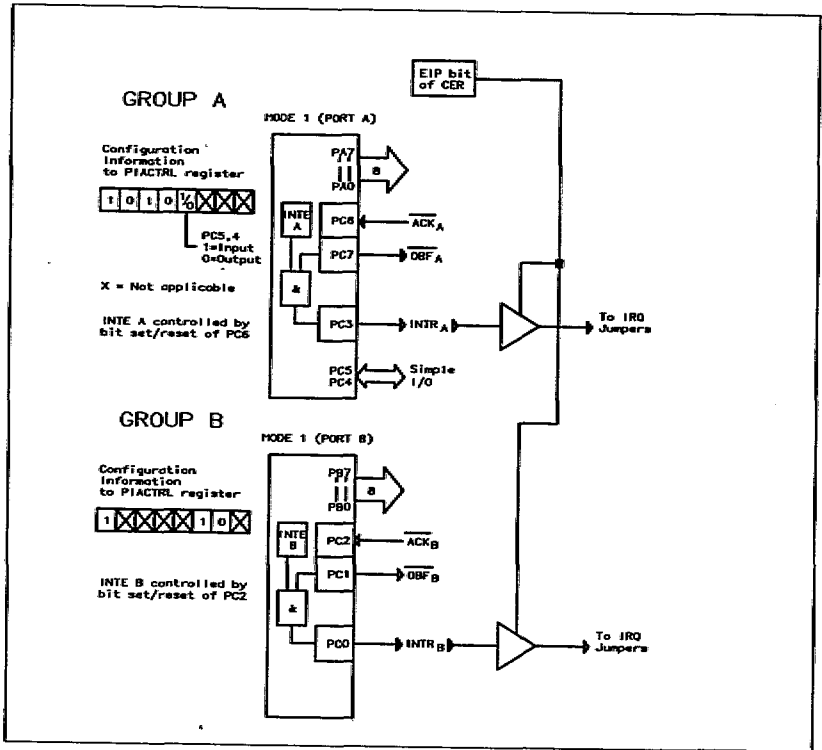


Figure 7: Mode 1 Output of an 8255.

To use the 8255 in mode 1 output with interrupts:

- Write a byte to the control register to configure the 8255 for mode 1 and the appropriate group for data output.
- With the bit set/reset operation, write a 1 to the Interrupt Enable flip-flop (INTE) of the desired port of the appropriate 8255.
- The 8255 Interrupt Request output (INTR) then goes high.
- Write a byte to the Counter/Interrupt Enable Register (CER) to enable interrupts from the appropriate 8255 to the host computer.
- The host computer detects the INTR line is active. From an Interrupt Service Routine (ISR) it writes a byte to the output port. This automatically resets the INTR line.
- The Output Buffer Full (/OBF) line to the digital I/O connector goes low to indicate that there is data to be read by the external device from the 8255.
- The external device pulses the Acknowledge (/ACK) input low and then high again to indicate that it has read the data.
- This makes the INTR line go high again and the cycle may be repeated until all the required data has been written.

Whenever a group of the 8255 is in mode 1 output, the status of the handshaking lines and interrupt signals can be obtained by reading port C. The byte read contains the following information:

Port C Mode 1 Output Status Information							
/OBF _A	INTE _A	C5	C4	INTR _A	INTE _B	/OBF _B	INTR _B

The 8255 may alternatively be used in mode 1 and the data written by polled (program) transfer. This is done as follows:

- Write a byte to the control register to configure the 8255 for mode 1 and the appropriate group for data output.
- The program continually monitors the /OBF line by reading port C, waiting for it to go high. A high indicates that the last data written to the port has been read by the external device.
- Then the program can write new data to the port.
- The /OBF line to the digital I/O connector goes low to indicate that there is data to be read by the external device from the 8255.
- The external device pulls the /ACK input low to read the data.
- This makes the /OBF line go high again and the cycle may be repeated until all the required data has been written.

The program could also enable the INTR line with the INTE flip-flop and then monitor INTR instead of the /OBF line. In this case, interrupts to the host computer are not enabled in the Counter/Interrupt Enable Register.

See below for programming the host computer for interrupt handling and Appendix B for Mode 1 timing specifications.

Mode 1 Programming Examples

```
#define BaseAddr    0x300    /* PC-14 base address at 300 hex */
#define DIO0A      0        /* Offset of port A of the first 8255 */
#define DIO0B      1        /* Offset of port B */
#define DIO0C      2        /* Offset of port C */
#define DIO0CTRL   3        /* Offset of the control register */

void main( void )
{
    unsigned int port0a, port0b, port0c, port0ctrl, data, outval = 0x1B;

    /* Calculate register addresses */
    port0a = BaseAddr + DIO0A;
    port0b = BaseAddr + DIO0B;
    port0c = BaseAddr + DIO0C;
    port0ctrl = BaseAddr + DIO0CTRL;

    /* 0xB0 = 1011 0000, port A input in mode 1, */
    outp( port0ctrl, 0xB0 ); /* ports B and C for output in mode 0 */

    while( !( inp(port0c) & 0x20 ) ); /* Wait for IBFA to become set */
    data = inp( port0a ); /* Read the data from port A */
}
```

```

/* 0xA0 = 1010 0000, port A output in mode 1, */
outp( port0ctrl, 0xA0 ); /* ports B and C for output in mode 0 */

while( !( inp(port0c) & 0x80 ) ); /* Wait for /OBFA to become set */
outp( port0a, outval ); /* Write the data to port A */
}

```

Mode 2: Strobed Bidirectional Bus I/O

This mode provides a means for communicating with an external device using an 8-bit bus for both transmitting and receiving data. Both input and output handshaking signals similar to mode 1 are provided to maintain proper bus flow discipline. Hardware interrupts signal the host computer that the port needs attention.

The following characterise Mode 2:

- Only group A operates in mode 2.
- One 8-bit bidirectional port, functions as both input and output.
- Five of the port C lines take on special functions.
- Both inputs and outputs are latched.
- One 3-bit simple I/O port.
- Data transfer by interrupts or polled I/O.

With both 8255s of the PC-14B configured in mode 2, the board provides the function of a full 16-bit wide bidirectional data bus.

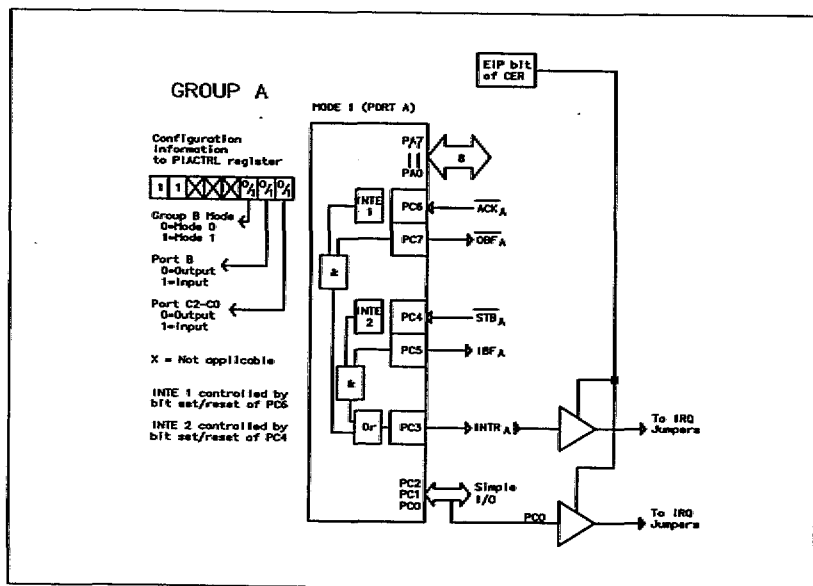


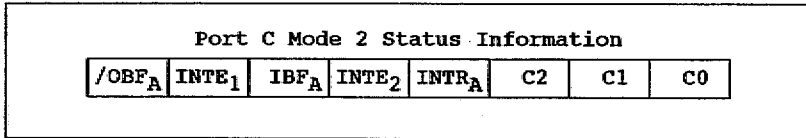
Figure 8: Mode 2 of an 8255.

Mode 2 Programming

To use the 8255 in mode 2 with hardware interrupt transfer:

- Write a byte to the control register to configure the 8255 for mode 2 operation.
- Select an interrupt level with the interrupt jumpers and enable 8255 interrupt(s) in the PC-14B Counter/Interrupt Enable Register (CER).
- With the bit set/reset operation, write a 1 to the Interrupt Enable number 1 flip-flop (INTE₁) to enable output transfer interrupts. Write a 1 to the Interrupt Enable number 2 flip-flop (INTE₂) to enable input transfer interrupts. Both input and output interrupts may be enabled at the same time. For a diagram, see Figure 8 above.
- With both interrupt flip-flops enabled, the interrupt request line to the host computer is activated if the external device has strobed data into the 8255 input latch or if the external device has read the PC-14B output data from the output latch. See Figure 11 in Appendix B for full timing requirements.
- The host computer detects the INTR line is active. The Interrupt Service Routine (ISR) which services this interrupt determines whether it was an input or output interrupt by checking bit 5 (IBF_A) of the mode 2 status information from port C. (See below.) If the IBF line is high (bit 5 is set) then it is an input interrupt, otherwise it is an output interrupt.
- In either case, the ISR simply reads the data from or writes data to the 8255 and then issues an end of interrupt command to the interrupt controller.
- This generates the appropriate handshake signals from the 8255 to the digital I/O connector.
- The cycle continues when the next interrupt is generated.

Whenever the 8255 is in mode 2, the status of the handshaking lines and interrupt signals can be obtained by reading port C. The byte read contains the following information:



The 8255 may alternatively be used in mode 2 with the data read and written by polled (program) transfer. This is done as follows:

- Write a byte to the control register to configure the 8255 for mode 2.
- The program continually monitors both the IBF_A and /OBF_A lines by reading port C.
- If the port C bit corresponding to IBF_A is set, this indicates that the external device has written data into the input latch. The program must therefore read the data from the port A.
- If the port C bit corresponding to /OBF_A is set, this indicates that the external device has read the data written by the program to port A. It must therefore write more data to port A.
- This process can be repeated until the required amount of data has been read and written.

Interrupts to the host computer must not be enabled in the Counter/Interrupt Enable Register when data is transferred by polled I/O.

Mode 2 Programming Examples

```

#define BaseAddr      0x300      /* PC-14 base address at 300 hex */
#define DIO0A         0          /* Offset of port A of the first 8255 */
#define DIO0B         1          /* Offset of port B */
#define DIO0C         2          /* Offset of port C */
#define DIO0CTRL      3          /* Offset of the control register */

void main( void )
{
  unsigned int port0a, port0b, port0c, port0ctrl;
  unsigned int sts, data, outval = 0x1B;

  /* Calculate register addresses */
  port0a = BaseAddr + DIO0A;
  port0b = BaseAddr + DIO0B;
  port0c = BaseAddr + DIO0C;
  port0ctrl = BaseAddr + DIO0CTRL;

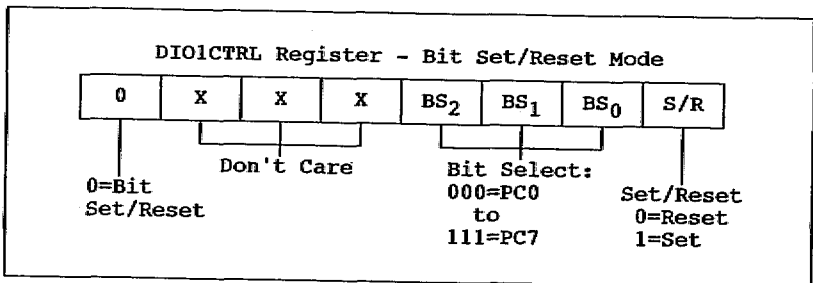
  /* 0xC0 = 1100 0000, port A to input in mode 2,
  outp( port0ctrl, 0xC0 ); /* ports B and C for output in mode 0 */

  while( !( sts = (inp(port0c) & 0xA0) ) ); /* Wait for IBFA or */
  /* /OBF to become set */
  if( sts & 0x80 ) /* /OBF was set */
    outp( port0a, outval ); /* Write some data to port A */
  else /* IBF was set */
    data = inp( port0a ); /* Read the data from port A */
}

```

Single Bit Set/Reset

Any of the eight bits of port C can be set or reset using a single output instruction to the DIOCTRL register. When port C is being used as status/control for port A or B, any of these bits can be set or reset just as if they were data output ports. The format of the byte to write to the DIOCTRL register to set or reset a port C bit is repeated below.



Mixed Mode Programming

An 8255 is not constrained to operate in one mode only. For example, Port A may operate in mode 2 and port B may then operate in either mode 1 or mode 0. For any combination, some or all of the port C lines are used for control or status. The remaining port C lines may be used in mode 0 either as inputs or outputs.

A read operation of port C returns all the port C lines except the /ACK and /STB lines. In their place will appear the status of the Interrupt Enable flip-flops (INTE_x). This is illustrated in figures 6 to 8 above and the status information bytes which follow the figures.

A write operation to port C will only affect lines programmed as mode 0 outputs. To write to any port C output programmed as a mode 1 output or to change an interrupt enable flip-flop, the Bit Set/Reset operation must be used.

Using the Bit Set/Reset command, any port C line programmed as an output (including INTR, IBF and /OBF) can be written or an interrupt enable flag set or reset. Lines programmed as inputs (including /ACK and /STB) are not affected by this command. Writing to these lines will affect the interrupt enable flags.

Bit Set/Reset

Interrupt Handling

This section presents briefly the necessary actions to take to set up interrupt data transfer from an 8255. Refer to the relevant sections in this manual for full details on the points below.

- The PC-14B board must be jumpered, using the PIA INTR jumper blocks, to route the 8255 INTR signal to an appropriate PC interrupt line.
- The corresponding interrupt must be enabled in the PC system board interrupt controller.
- The 8255 must be used in a mode which generates interrupt signals (mode 1 or mode 2).
- The relevant interrupt enable flags (INTE) in the 8255 must be set.
- Appropriate bits must be written to the Counter/Interrupt Register to enable 8255 interrupts to reach the selected PC interrupt line.
- A software routine (ISR) must be written to service the interrupts.
- The address of the ISR must be stored in the host PC's interrupt vector table.

Programming the 8254 Timer/Counter

The other major sub-system of the PC-14 is based around an 8254 timer/counter integrated circuit. This chapter provides detailed programming information on the 8254 for the PC-14B.

Introduction

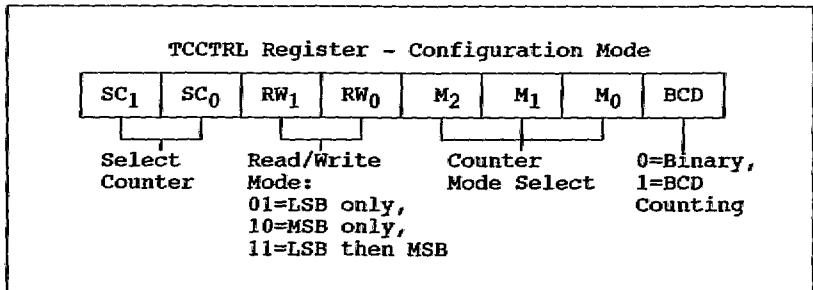
The 8254 is a general purpose three channel timer/counter device. Each timer/counter is totally independent, and each may be programmed in different modes and data formats. Since all three timers are identical, the information provided here applies equally to all the timers.

The timer/counter chip itself consists of a control word register, some logic circuitry and the three counters. Additional hardware on the PC-14B provides a programmable source of clock pulses for each timer, a software controlled counting gate and software controlled interrupt enable/disable.

Each counter consists of a two byte wide Count Register, a 16-bit Counting Element and a two byte wide Output Latch.

Count Register	The Count Register stores the initial 16-bit count written to a counter. It consists of two byte wide registers, which are written to separately. When a counter is programmed with a control word, the Count Register is cleared. Both Count Register bytes are transferred (loaded) to the Counting Element simultaneously.
Counting Element	The Counting Element is simply a 16-bit presettable synchronous down counter. It cannot be read or written to directly. It is automatically loaded on specified conditions from data in the Count Register. The count value is always read from the Output Latch.
Output Latch	The Output Latch normally follows the Counting Element. It consists of two byte wide registers, which are read from separately. If a suitable Counter Latch Command (see below) is sent to the counter, the current count value is latched in the Output Latch until it is read from the counter's data register (TC2, TC1 or TCO). Thereafter, the Output Latch continues to follow the Counting Element.
Software Gate	The Counter/Interrupt Enable Register contains three timer Count Enable bits (T2CE, T1CE, TOCE), one for each counter, that control whether the corresponding counter receives clock pulses from its Clock Source. Setting a Timer Count Enable bit <i>inhibits</i> counting and clearing the bit <i>enables</i> counting. See page 37 in Chapter 5 for more details on this register.
Timer Interrupts	The Counter/Interrupt Enable Register mentioned above also contains three Timer Interrupt Enable bits (T2IE, T1IE, TOIE). These bits control whether the corresponding counter's output is enabled onto a jumper selected IRQ line of the PC's I/O Channel. Setting a Timer Interrupt Enable bit allows the timer output to generate interrupts and clearing the bit inhibits interrupt generation.
Programming a Counter	On power up or reset, the state of the 8254 is undefined. Before any timer/counter operations can be done, each timer to be used must be programmed with a Control Word, which is written to the Control Register. This sets the individual counter's: <ul style="list-style-type: none"> - Operating mode (Mode 0 to 5), - Counting format (BCD or binary) and - Read/Write format (LSB only, MSB only or LSB then MSB).

The programmed counter then operates in the specified format until it is reset or new configuration information is written to the Control Register. The format of this register when used for configuring a counter is repeated below for reference.



See page 34 in Chapter 5 for the full bit definitions of this register.

Using the Control Word, each counter may be programmed to transfer data in one of three ways:

- Read/Write least significant byte only,
- Read/Write most significant byte only or
- Read/Write least significant byte first, then most significant byte.

A new initial count may be written to a counter without affecting the counter's programming in any way. Counting will be affected as described in the Mode Definitions below.

Writing a one byte initial count simply consists of outputting the byte to the counter's data register. Writing a two byte count consists of writing the first byte (the least significant byte) to the counter's data register and then, at any time later, writing the second or most significant byte to the same data register.

Each counter has three physical electrical connections, all of which are available on the auxiliary connector. They are:

- a clock pulse input,
- a gate input and
- a timer output.

The clock pulse input is the physical connection, where clock pulses are applied to a counter. A clock pulse is defined as a rising edge, then a falling edge, in that order, at a counter's clock input. New counts are loaded, and the Counting Element is decremented, on the falling edge of a clock pulse.

The source of clock pulses for each timer is programmable from software via the Clock Source Register (CSR). Clock pulses may come from: a 2MHz oscillator on the PC-14B board; the host computer's I/O bus clock divided by either 4 or 64 or an external source. See Chapter 5 for information on programming the clock source..

Depending on the counter's mode, the gate input provides for: enable/disable counting; count initiating (trigger) or setting/resetting the timer output. The three counter's gates are passively held high by the PC-14 board and may be left unconnected if desired. The functioning of the gate input is described in the individual mode definitions which follow this, and summarised in Table 5 on page 57 below the mode definitions.

Read Operations

It is often necessary to read the value or status of a counter without disturbing the count in progress. This is easily done using one of three methods: a simple read operation; a Counter Latch Command or a Read-Back Command. The results of the read operation are read from the counter's data register, in the programmed format (LSB, MSB or LSB then MSB) of that counter. This will be termed **reading a counter**, and may take *one* physical CPU read instruction (LSB or MSB) or *two* physical CPU read instructions (LSB then MSB).

The simple read operation consists of reading the contents of the desired counter's data register. The clock input of the counter must be disabled for this to be successful, otherwise the count may be in the process of changing when it is read, returning a completely erroneous count value. The clock input may be disabled with the gate input, the software gate (described above) or with external logic.

Data Transfer
FormatHardware
ConnectionsClock Pulse
Input

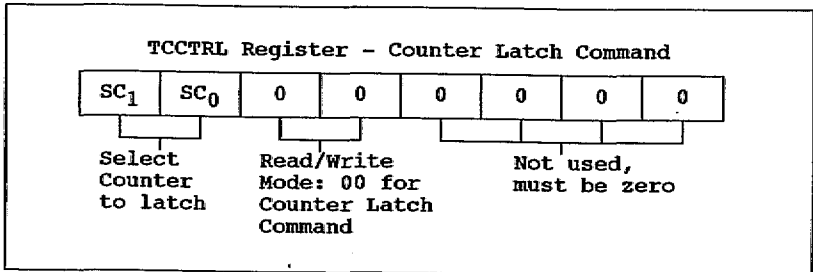
Gate Input

Simple Read
Operation

Counter Latch
Command

A Counter Latch Command is invoked by writing a special type of Control Word to the timer/counter's Control Register, TCCTRL.

The format of the Control Word for a Counter Latch Command is diagrammed below.



Bits SC₁ and SC₀ *select* the counter whose count is to be latched. 00 selects counter 0, 01 selects counter 1 and 10 selects counter 2

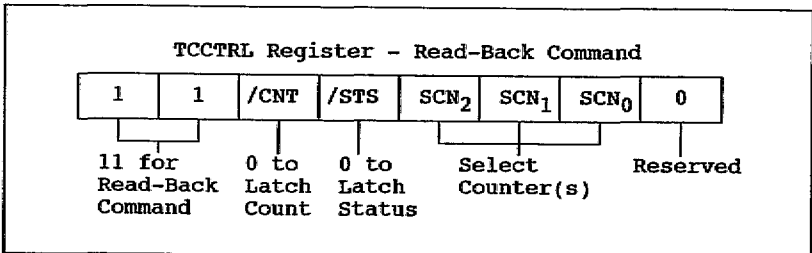
The selected counter's Output Latch latches the count at the time the Counter Latch Command is received. The count is held in the latch until it is read (or the counter reprogrammed). The count is then automatically unlatched and the Output Latch returns to following the Counting Element. Multiple Counter Latch Commands may be used to latch more than one counter. Each latched counter holds its count until read. Counter Latch Commands do not affect the programming of the counter in any way.

If a counter is latched, any subsequent Counter Latch Commands to the same counter, before the count has been read, will be ignored. When read, the count value returned will be the count at the time the first Counter Latch Command was issued.

Read-Back
Command

A Read-Back Command is issued by writing a special type of Control Word to the timer/counter's Control Register, TCCTRL. Read-Back Commands may be used to latch one or more counter's *current count value* and/or *status* information.

The format of the Control Word for a Read-Back Command is diagrammed below.



Setting any or all of bits 3 to 1 (SCN₂ to SCN₀) *selects* the counter(s) to which this command will apply. Setting the Count (/CNT) bit to 0 causes the current count of the selected counter(s) to be latched, setting the Status (/STS) bit to 0 causes the status byte of the selected counter(s) to be latched and setting both bits to 0 causes both the status and current count to be latched (see below).

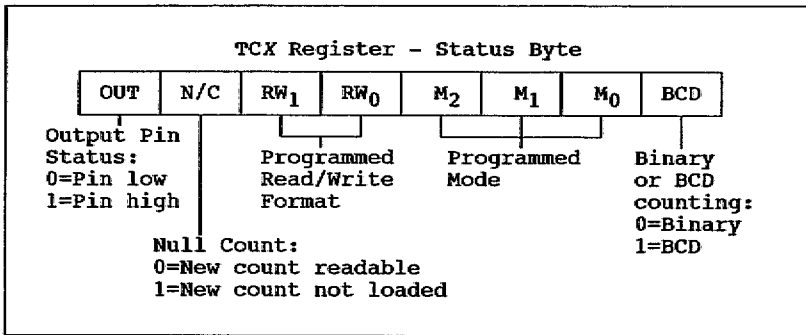
The Read-Back Command is used to latch the current count of multiple counters in their respective Output Latches. This is done by setting the Count Bit to 0 in the Control Word for this command. This single command is functionally equivalent to multiple Counter Latch Commands. Each counter's latched count is held until read (or the counter reprogrammed). That counter is automatically unlatched when read, but the other counters remain latched until they are read. If multiple Read-Back Commands are issued to the same counter without reading the count, all but the first are ignored. The count returned is the current count at the time the first Read-Back Command was issued.

Multiple Counter Latch

The Read-Back Command may also be used to latch certain status information of the selected counter(s), in their respective Output Latches. This is done by setting the Status Bit to 0 in the Control Word for this command.

Counter Status Information

The counter's status byte, when read, provides the information shown in the diagram below.



Bits M₂ to M₀ return a binary number corresponding to the counter's programmed mode.

The Null Count bit indicates if the last count written to the Count Register has been loaded into the Counting Element. If the count has not yet been loaded then it cannot be read and the Null Count bit will be set to 1. Reading the count when the Null Count bit is set will return the current count from the previous initial count written to the Count Register. The exact time the new count is loaded depends on the mode of the counter (see below) but the Null Count bit clear indicates the new count has been loaded.

Bit 7 (OUT) reflects the state of the counter's output pin. This provides software the power to monitor this pin.

Setting both the Status (/STS) bit and the Count (/CNT) bit to 0 causes both the status byte and the current count of the selected counter(s) to be latched simultaneously. This is functionally the same as issuing two separate Read-Back commands at once. If both the status and current count are latched, the first read operation of the counter's Data Register will return that counter's status byte, regardless of which was latched first. The next one or two reads (depending on whether the counter has been programmed for one or two byte counts) return the latched count. Subsequent reads return unlatched counts.

Latching both Status and Current Count

Counter Mode Definitions

The following six sections describe in detail the different counting modes of the 8254 timer/counter. In the descriptions which follow, the output refers to the state of the output pin of the device, and gate refers to the state of the counter's gate input pin (hardware gate). Both signals from all three counters are available on the auxiliary connector. The software gate described in the above sections does not provide all the functions of the hardware gate input, it simply provides software control over whether clock pulses reach or are inhibited from reaching the counter's clock input.

The word *Trigger* is used to mean a rising edge at a counter's gate input, *Counter Loading* the transfer of the initial count from the counter's Count Register to its Counting Element and *Clock Pulse* a rising edge and then a falling edge at a counter's Clock input.

Mode 0: Interrupt on Terminal Count

After the mode byte is written to the Control Register, the output is low. Once an initial count has been written, the output remains low until the counter has counted down to zero. The output then goes high and remains high until a new count is written or the counter reprogrammed.

The gate input inhibits counting when low, and enables counting when high.

After the Control Word and initial count have been written, the counter is loaded on the next clock pulse. This clock pulse does not decrement the count, so for an initial count of N , the output goes high $N+1$ clock pulses after the initial count was written.

If a new count is written to the counter, it will be loaded on the next clock pulse and counting will continue from the new count. If a two-byte count is written, the first byte disables counting and sets the output low. After the second byte is written, the full count is loaded on the next clock pulse. This allows the counting sequence to be synchronised by software. Again the output goes high after $N+1$ clock pulses.

If the initial count is written when the gate is low, it will still be loaded on the next clock pulse. When the gate goes high, the output will go high N clock pulses later.

Using the internal oscillator or bus clock, this mode can be used to generate a positive edge on the external output after a programmable time, or if the board is jumpered for interrupts, to generate an interrupt after a programmable time.

This mode can also be used to count events or frequency.

Mode 1: Hardware Retriggerable One-Shot

After the counter is programmed the output will be high. Writing an initial count arms the counter and a subsequent trigger loads the counter. The output goes low on the next clock pulse and remains low until the counter reaches zero. The output then goes high and remains high until the next clock pulse after the next trigger.

An initial count of N results in a one-shot pulse N clock cycles long. The one-shot is retriggerable, hence the output will remain low for N clock pulses after any trigger. The one-shot pulse can be repeated without rewriting the initial count to the counter. The gate input has no effect on the output.

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

Mode 2: Rate Generator

After the counter is programmed the output will be high. An initial count of N is loaded on the next clock pulse and when it has decremented down to 1, the output goes low for *one* clock pulse. The output then goes high, the counter automatically reloads the initial count and the process is repeated indefinitely. The sequence is repeated every N clock pulses.

The gate input enables counting when high, and inhibits counting when low. If the gate goes low during an output pulse, the output is set high immediately. A trigger reloads the initial count on the next clock pulse and the output goes low for one clock pulse after N clock pulses. Thus the gate input can be used to synchronise the counter.

Writing a new count does not affect the current counting sequence. If a trigger is subsequently received before the end of the current period, the counter will be reloaded on the next clock pulse and counting will continue from the new count. Otherwise the new count will be loaded at the end of the current cycle.

In this mode an initial count of 1 is invalid.

Mode 2 functions like a divide by N counter. It can also be used to generate an output frequency or a periodic interrupt.

Mode 3: Square Wave Generator

Mode 3 is similar to Mode 2 except for the duty cycle of the output. After the counter is programmed the output will be high. An initial count of N is loaded on the next clock pulse. When *half* of the initial count has expired the output goes low for the remainder of the count. The output then goes high, the counter automatically reloads the initial count and the process is repeated indefinitely. This results in a square wave with a period of N clock cycles.

The gate input enables counting when high, and inhibits counting when low. If the gate goes low when the output is low, the output is set high immediately. A trigger reloads the counter with the initial count on the next clock pulse. Thus the gate input can be used to synchronise the counter.

Writing a new count does not affect the current counting sequence. If a trigger is subsequently received before the end of the current half-cycle of the square wave, the counter will be reloaded on the next clock pulse and counting will continue from the new count. Otherwise the new count will be loaded at the end of the current half-cycle.

Mode 3 functions slightly differently for even and odd initial count values.

For even counts: the output is initially high. On the next clock pulse the initial count is loaded. On subsequent clock pulses it is decremented by *two*. When the count expires, the output toggles and the counter is reloaded with the initial count. This process is repeated indefinitely.

For odd counts: the output is initially high. On the next clock pulse the initial count *minus one* (an even number) is loaded. On subsequent clock pulses it is decremented by *two*. One clock pulse *after* the count expires, the output goes low and the counter is reloaded with the initial count minus one. Subsequent clock pulses continue to decrement the count by two. When the count

expires, the output goes high again and the counter is reloaded with the initial count minus one. This process is repeated indefinitely. So for odd counts, the output is high for $(N+1)/2$ counts and low for $(N-1)/2$ counts or high for one count longer than it is low.

Mode 3 is typically used to generate an output frequency.

Mode 4: Software Triggered Strobe.

After the mode byte is written to the Control Register, the output is high. Once an initial count has been written, the output remains high until the counter has counted down to zero. The output then goes low for one clock pulse and then goes high again. The counting sequence is triggered by writing an initial count.

The gate input inhibits counting when low, and enables counting when high. It has no effect on the output.

After the Control Word and initial count have been written, the counter is loaded on the next clock pulse. This clock pulse does not decrement the count, so for an initial count of N , the output strobes low $N+1$ clock pulses after the initial count was written.

If a new count is written while counting, it will be loaded on the next clock pulse and counting will continue from the new count. If a two-byte count is written, the first byte written has no effect on counting. After the second byte is written, the full count is loaded on the next clock pulse. This allows the counting sequence to be retrIGGERED by software. Again the output strobes low after $N+1$ clock pulses.

Using the internal oscillator or bus clock, this mode can be used to generate a negative pulse on the external output after a programmable time. With an external clock source, it generates a pulse after a programmable number of events.

Mode 5: Hardware Triggered Strobe

This mode is similar to mode 4 except that the counting is triggered by a rising edge on the counter's gate input.

After the Control Word and initial count have been written, the output is high. The counter is loaded on the next clock pulse after a trigger is received. This clock pulse does not decrement the count. The output remains high until the counter has counted down to zero. The output goes low for one clock pulse and then goes high again. So for an initial count of N , the output strobes low $N+1$ clock pulses after the initial count was written.

The counting sequence is retrIGGERABLE: a trigger causes the counter to be loaded with the initial count on the next clock pulse. The output will not strobe low until $N+1$ clock pulses after any trigger.

The gate input has no effect on the output.

If a new count is written while counting, it will have no effect on the current count sequence. If a trigger is received after the new count is written but before the current count expires, the counter will be reloaded on the next clock pulse and counting will continue from there.

Using the internal oscillator or bus clock, this mode can be used to generate a negative pulse on the external output after a programmable time from an external trigger.

Operating Mode	Low or Going Low	Rising	High
0	Disables Counting	---	Enables Counting
1	---	Initiates Counting & Resets Output after Next Clock	---
2	Disables Counting & Sets Output Immediately High	Initiates Counting	Enables Counting
3	Disables Counting & Sets Output Immediately High	Initiates Counting	Enables Counting
4	Disables Counting	---	Enables Counting
5	---	Initiates Counting	---

Table 5: Summary of the Functions of the Gate Inputs.

Each timer/counter's gate input pin is passively pulled high by the PC-14 board. Therefore, it is acceptable to leave these inputs unconnected if they are not used. The corresponding counter will then behave according to the last column in the table above.

Unused Gate Inputs

Interrupt handling

Each timer/counter may be configured to generate interrupt requests (IRQs) to the host computer. It is the same in principle as using any of the other sub-systems (8255 PIAs or the External Interrupt) to generate PC interrupts:

- Select the desired interrupt line to use for the desired timer/counter(s) with the jumper blocks on the PC-14B board.

The following procedures are all done from software:

- Program the counter and perform any other setup that has to be done.
- Save the state (enabled or disabled) of the selected interrupt level on the PC system board and then disable it.
- Save the old interrupt vector and install a new one, that will point to the interrupt service routine that will service the timer interrupts.
- Enable the interrupt level.
- Enable the timer interrupt onto the PC I/O channel by writing to the Counter/Interrupt Enable Register (CER).

Now the software can continue with other tasks. The interrupt service routine must:

- Do whatever processing it needs to do.
- Either chain another interrupt routine if the interrupt line is being shared or write an End of Interrupt Command to the PC system board interrupt controller.

To clean up the interrupts after using them:

- Disable the interrupt level.
- Disable timer interrupts with the CER.
- Restore the interrupt vector to what it was before.
- Restore the interrupt level to its previous state.

Note that some timer/counter modes are more suitable for generating interrupts (eg Mode 0) than others.

Using the External Interrupt Input

The external interrupt input provides a single, latched dedicated line to a selected interrupt request (IRQ) line on the PC's I/O Channel. It allows external hardware devices asynchronous requests to the host computer for attention. The input signal must be TTL compatible. Operation is as follows:

- The program sets the interrupt vector, in the host computer's interrupt vector table, corresponding to the desired IRQ to point to the interrupt routine (ISR) that will service the interrupt.
- It then enables the corresponding interrupt level in the interrupt controller on the PC system board.
- Next it enables interrupts from the external interrupt latch on the PC-14B by setting the external interrupt enable bit (EXI, bit-6) in the Clock Source Register.
- Now the system is ready to process external interrupts.
- The signal from the external device is initially low. When it goes high, it is latched by the PC-14B and activates the corresponding IRQ line on the PC bus.
- This causes the host computer to execute the interrupt service routine for the external interrupt. The routine must contain two statements to allow further interrupts to be received:
 - It must send an end of interrupt (EOI) command to the interrupt controller on the computer system board and
 - It must write a 0 to bit 0 of the External Interrupt Reset Register to clear the external interrupt input latch.

Note that it is possible to use up to four of the digital input lines for further external interrupt inputs. The lines 0C0 and 0C3 from the digital connector to the first PIA and the lines 1C0 and 1C3 from the digital connector to the second PIA are all internally connected, via buffers and selection jumpers, to IRQ lines on the host computer's I/O channel. They are normally used to

generate interrupts when the 8255s are operated in mode 1 and mode 2. However, they can be used as additional dedicated external interrupts if the 8255(s) are put into a state in which these lines are not driven. Any mode where these port C lines are configured as inputs will satisfy this condition.

These interrupts, when used in this way, are not latched. Therefore the external device must hold the interrupt line high until it is serviced, and then deactivate it.

Microsoft 'C' Program Example for External Interrupts.

```
#define BaseAddr    0x300    /* PC-14 base address at 300 hex */
#define CSR        12      /* Offset of the clock source register */
#define IRQRST     14      /* Offset of the interrupt reset register */

/* Port addresses of the interrupt controller on the system board: */
#define ICR0       0x20    /* Control register */
#define IMR0       0x21    /* Mask register */
/* Offset into the vector table of the hardware interrupts */
#define IC0OS      0x8
/* Non specific end of interrupt command to interrupt controller */
#define eoi        0x20

/* Function prototypes for the interrupt routines */
void (interrupt far *oldvect)( void );
void interrupt far ExtInt( void );

void main( void )
{
  unsigned int csr, irqrst;
  int irqlvl = 5;      /* Hardware interrupt level */
  /* Interrupt number corresponding to the interrupt level */
  int IntNum = irqlvl+IC0OS;
  int imrval;

  /* Calculate register addresses */
  csr      = BaseAddr + CSR;
  irqrst   = BaseAddr + IRQRST;

  /* Disable the interrupt for the moment */
  outp( IMR0, inp( IMR0 ) | 1 << irqlvl );

  /* Save the address of the old routine pointed to by this interrupt */
  oldvect = _dos_getvect( IntNum );
  /* Put the address of the ExtInt routine in the vector table. */
  _dos_setvect( IntNum, ExtInt );

  /* Reset interrupt latch, just in case something set the interrupt. */
  outp( IRQRST, 0 );
  /* Enable extint and set clock source to CLK/64. */
  outp( CSR, 0x40 );

  /* Enable the interrupt now. */
  outp( IMR0, inp( IMR0 ) | 0 << irqlvl );

  /* The user application is inserted here. Any external interrupts */
  /* will get processed in the ExtInt() interrupt service routine. */
}
```

```
/* Now to finish */

/* Disable the interrupt */
outp( IMR0, inp( IMR0 ) | 1 << irqlvl );

/* Put back the old interrupt vector. */
_dos_setvect( IntNum, oldvect );

/* Disable extint and set clock source to CLK/64. */
outp( CSR, 0x00 );
}

/* The actual Interrupt Service Routine */

void interrupt far ExtInt( void )
{
    /* Do whatever processing is necessary, and then ... */
    outp( IRQRST, 0 );      /* Reset interrupt latch */
    outp( ICRO, eoi );     /* Output end of interrupt to controller */
}

```

Driver Software

Full driver software is supplied as part of the PC-14B package. It is called the Data Acquisition Driver Software Library, and is provided in two equivalent forms:

- A ready to use, precompiled, driver software library, with routines callable from most high level languages.
- Complete 'C source code for all driver routines.

The Driver Library allows programmers to control the PC-14 board via high level function calls, so allowing the user to write custom software applications without having to understand the low level operation of the board.

Also included with the driver package is complete source code, in C for the entire driver package. This allows advanced users to modify existing code, rather than having to start writing low level code from scratch.

The Data Acquisition Driver Library supports not only the PC-14B board, but a number of other boards too. This chapter presents a quick reference to the functions in the Driver Library that apply to the PC-14B board. Full details on using the library functions, the source code and a full function reference are provided in the companion manual to this, *The Data Acquisition Driver Software Manual*.

A Note about Using the Driver Library

If you are using the Driver Library to operate the board, unless you know exactly what you are doing, **always** control the board with calls to the Driver Library and do **not** bypass the Driver Library and read or write directly to or from the board. This is because the Driver Library keeps an internal software copy of the state of the board. If you communicate to the board directly, this will obviously not update the software copy of the board's state. Hence, subsequent calls to the Driver Library may not produce the desired results, or the board may behave erratically or unpredictably.

In summary, do not allow the board's state to get out of sync with the software copy of its state.

Quick Function Reference

The Data Acquisition Driver, when used with the PC-14 board has fourteen appropriate function calls. These are the following:

Function Name	Brief Description
<code>Init_brd</code>	Initialises one of up to eight boards. This function initialises the hardware of the board if necessary and informs the driver system of the board's presence. It must be called before any other function calls are made to the board.
<code>DIO_cfg_line</code>	Configures lines or groups of lines in a digital I/O port. Allows the smallest unit in a port to be configured.
<code>DIOprt_cfg</code>	Configures the operating mode of a port. This function is a subset of the <code>DIO_cfg_line</code> function.
<code>DIO_in_port</code>	Returns digital data from the specified port.
<code>DIO_out_port</code>	Writes digital data to the specified port.
<code>DIO_in_line</code>	Returns the state of a single line of the specified port.
<code>DIO_out_line</code>	Sets the state of a single line of the specified port.
<code>DIOprt_status</code>	Returns a status byte indicating the status of an individual I/O port.
<code>CT_cfg</code>	Configures the operating mode, the counting method and the read/write format of the specified timer/counter.
<code>CT_clock_source</code>	Selects the source of clock pulses for the specified counter.
<code>CT_read</code>	Latches and returns the current count value of a counter.
<code>CT_write</code>	Writes an initial counting value to a counter.
<code>CT_status</code>	Returns a status byte indicating a counter's programming, the state of the its output pin and whether the last initial count has been loaded.
<code>CT_soft_gate</code>	Gates - inhibits or allows clock pulses to reach - the selected counter.

Hardware Specifications

External Interface

Digital I/O subsystem

Number of I/O lines:	48 in six 8-bit digital ports
Input voltage:	Logic 0: -0.5 - 0.8V Logic 1: 2V - 5V
Output voltage:	Logic 0: 0.45V max Logic 1: 2.4V min
Output current:	Low state: 1.7mA max High state -200uA min
Maximum Darlington drive current:	-4.0mA; from port types B and C only
Maximum power:	1W total maximum power from all three ports per PIA
Power supply current at I/O connector:	300mA from +5V typ, others vary up to 300mA depending on host computer power supply rating
Digital I/O connector:	50 way male IDC.

Auxiliary subsystems

Timer/Counters:	Three timer/counter input lines, gate inputs and counter outputs.
Input Clock Pulse Frequency:	8MHz max.
Input voltage:	Logic 0: -0.5 - 0.8V Logic 1: 2V - 5V
Output voltage:	Logic 0: 0.4V max Logic 1: 3.0V min
External Interrupt Pulse:	37ns min before rising edge 30ns min after rising edge
Other lines:	All computer power supplies available on auxiliary I/O connector

Host computer interface

Bus type:	XT, AT, ISA, EISA
I/O address selection:	From 0h to FF0h on 16 byte boundaries, DIP switch selectable.
I/O wait states:	0, 1, 2, 4 or 8 jumper selectable. Only affects cycles directed at the card.
Number of registers:	Eleven 8-bit
Word size:	8-bits
Word transfer rate:	10 MHz max ¹
Power requirements:	+5V Max 750 mA
	+12V 0.0 mA
	-12V 0.0 mA
	-5V 0.0 mA

Environmental

Absolute maximum input voltage:	7V
Absolute minimum input voltage:	-0.5V
Operating temperature:	0°C to 70°C.
Storage temperature:	-55°C to 150°C.
Relative humidity:	5% to 90% non-condensing
Board size:	17.4 x 10.1 cm excluding the bracket and edge connector

¹ This figure is the maximum transfer rate of the PC-14B hardware. In practise, the limiting factors are the host computer and software speed. On an 8MHz AT, an assembly program continuously writing a constant to an output port achieves a transfer rate of around 500kHz.

Appendix A

Base Address Switch Settings

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	Base Address
0h	ON	ON	ON	ON	ON	ON	ON	ON	0h
10h	ON	ON	ON	ON	ON	ON	ON	OFF	10h
20h	ON	ON	ON	ON	ON	ON	OFF	ON	20h
30h	ON	ON	ON	ON	ON	ON	OFF	OFF	30h
40h	ON	ON	ON	ON	ON	ON	OFF	ON	40h
50h	ON	ON	ON	ON	ON	ON	OFF	OFF	50h
60h	ON	ON	ON	ON	ON	ON	OFF	ON	60h
70h	ON	ON	ON	ON	ON	OFF	OFF	OFF	70h
80h	ON	ON	ON	ON	OFF	ON	ON	ON	80h
90h	ON	ON	ON	ON	OFF	ON	ON	OFF	90h
A0h	ON	ON	ON	ON	OFF	ON	OFF	ON	A0h
B0h	ON	ON	ON	ON	OFF	ON	OFF	OFF	B0h
C0h	ON	ON	ON	ON	OFF	OFF	ON	ON	C0h
D0h	ON	ON	ON	ON	OFF	OFF	ON	OFF	D0h
E0h	ON	ON	ON	ON	OFF	OFF	OFF	OFF	E0h
F0h	ON	ON	ON	ON	OFF	OFF	OFF	OFF	F0h
100h	ON	ON	ON	OFF	ON	ON	ON	ON	100h
110h	ON	ON	ON	OFF	ON	ON	OFF	ON	110h
120h	ON	ON	ON	OFF	ON	ON	OFF	ON	120h
130h	ON	ON	ON	OFF	ON	ON	OFF	OFF	130h
140h	ON	ON	ON	OFF	ON	OFF	ON	ON	140h
150h	ON	ON	ON	OFF	ON	ON	ON	OFF	150h
160h	ON	ON	ON	OFF	ON	OFF	OFF	ON	160h
170h	ON	ON	ON	OFF	ON	OFF	OFF	OFF	170h
180h	ON	ON	ON	OFF	OFF	ON	ON	ON	180h
190h	ON	ON	ON	OFF	OFF	ON	ON	OFF	190h
1A0h	ON	ON	ON	OFF	OFF	ON	OFF	ON	1A0h
1B0h	ON	ON	ON	OFF	OFF	ON	OFF	OFF	1B0h
1C0h	ON	ON	ON	OFF	OFF	OFF	ON	ON	1C0h
1D0h	ON	ON	ON	OFF	OFF	OFF	ON	OFF	1D0h
1E0h	ON	ON	ON	OFF	OFF	OFF	OFF	ON	1E0h
1F0h	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	1F0h
200h	ON	ON	OFF	ON	ON	ON	ON	ON	200h
210h	ON	ON	OFF	ON	ON	ON	ON	OFF	210h
220h	ON	ON	OFF	ON	ON	ON	OFF	ON	220h
230h	ON	ON	OFF	ON	ON	ON	OFF	OFF	230h
240h	ON	ON	OFF	ON	ON	OFF	ON	ON	240h
250h	ON	ON	OFF	ON	ON	OFF	ON	OFF	250h
260h	ON	ON	OFF	ON	ON	OFF	OFF	ON	260h
270h	ON	ON	OFF	ON	ON	OFF	OFF	OFF	270h
280h	ON	ON	OFF	ON	OFF	ON	ON	ON	280h
290h	ON	ON	OFF	ON	OFF	ON	ON	OFF	290h
2A0h	ON	ON	OFF	ON	OFF	ON	OFF	ON	2A0h
2B0h	ON	ON	OFF	ON	OFF	ON	OFF	ON	2B0h
2C0h	ON	ON	OFF	ON	OFF	OFF	ON	ON	2C0h
2D0h	ON	ON	OFF	ON	OFF	OFF	ON	OFF	2D0h
2E0h	ON	ON	OFF	ON	OFF	OFF	OFF	ON	2E0h
2F0h	ON	ON	OFF	ON	OFF	OFF	OFF	OFF	2F0h
300h	ON	ON	OFF	OFF	ON	ON	ON	ON	300h
310h	ON	ON	OFF	OFF	ON	ON	ON	OFF	310h
320h	ON	ON	OFF	OFF	ON	ON	OFF	ON	320h
330h	ON	ON	OFF	OFF	ON	ON	OFF	OFF	330h
340h	ON	ON	OFF	OFF	ON	OFF	ON	ON	340h
350h	ON	ON	OFF	OFF	ON	OFF	ON	OFF	350h
360h	ON	ON	OFF	OFF	ON	OFF	OFF	ON	360h
370h	ON	ON	OFF	OFF	ON	OFF	OFF	OFF	370h
380h	ON	ON	OFF	OFF	OFF	ON	ON	ON	380h
390h	ON	ON	OFF	OFF	OFF	ON	ON	OFF	390h
3A0h	ON	ON	OFF	OFF	OFF	ON	OFF	ON	3A0h
3B0h	ON	ON	OFF	OFF	OFF	ON	OFF	OFF	3B0h
3C0h	ON	ON	OFF	OFF	OFF	OFF	ON	ON	3C0h
3D0h	ON	ON	OFF	OFF	OFF	OFF	ON	OFF	3D0h
3E0h	ON	ON	OFF	OFF	OFF	OFF	ON	ON	3E0h
3F0h	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	3F0h

Table 6: Base address switch settings.

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	Base Address
400h	ON	OFF	ON	ON	ON	ON	ON	ON	400h
410h	ON	OFF	ON	ON	ON	ON	ON	OFF	410h
420h	ON	OFF	ON	ON	ON	ON	ON	ON	420h
430h	ON	OFF	ON	ON	ON	ON	OFF	OFF	430h
440h	ON	OFF	ON	ON	ON	ON	OFF	OFF	440h
450h	ON	OFF	ON	ON	ON	ON	OFF	ON	450h
460h	ON	OFF	ON	ON	ON	ON	OFF	OFF	460h
470h	ON	OFF	ON	ON	ON	ON	OFF	OFF	470h
480h	ON	OFF	ON	ON	OFF	ON	ON	ON	480h
490h	ON	OFF	ON	ON	OFF	ON	ON	ON	490h
4A0h	ON	OFF	ON	ON	OFF	ON	OFF	ON	4A0h
4B0h	ON	OFF	ON	ON	OFF	ON	OFF	OFF	4B0h
4C0h	ON	OFF	ON	ON	OFF	OFF	ON	ON	4C0h
4D0h	ON	OFF	ON	ON	OFF	OFF	ON	OFF	4D0h
4E0h	ON	OFF	ON	ON	OFF	OFF	OFF	ON	4E0h
4F0h	ON	OFF	ON	ON	OFF	OFF	OFF	OFF	4F0h
500h	ON	OFF	ON	OFF	ON	ON	ON	ON	500h
510h	ON	OFF	ON	OFF	ON	ON	ON	OFF	510h
520h	ON	OFF	ON	OFF	ON	ON	ON	OFF	520h
530h	ON	OFF	ON	OFF	ON	ON	ON	OFF	530h
540h	ON	OFF	ON	OFF	ON	OFF	ON	ON	540h
550h	ON	OFF	ON	OFF	ON	OFF	ON	OFF	550h
560h	ON	OFF	ON	OFF	ON	OFF	OFF	ON	560h
570h	ON	OFF	ON	OFF	ON	OFF	OFF	OFF	570h
580h	ON	OFF	ON	OFF	OFF	ON	ON	ON	580h
590h	ON	OFF	ON	OFF	OFF	ON	ON	OFF	590h
5A0h	ON	OFF	ON	OFF	OFF	ON	OFF	ON	5A0h
5B0h	ON	OFF	ON	OFF	OFF	ON	OFF	OFF	5B0h
5C0h	ON	OFF	ON	OFF	OFF	OFF	ON	ON	5C0h
5D0h	ON	OFF	ON	OFF	OFF	ON	ON	OFF	5D0h
5E0h	ON	OFF	ON	OFF	OFF	OFF	OFF	ON	5E0h
5F0h	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	5F0h
600h	ON	OFF	OFF	ON	ON	ON	ON	ON	600h
610h	ON	OFF	OFF	ON	ON	ON	ON	OFF	610h
620h	ON	OFF	OFF	ON	ON	ON	OFF	ON	620h
630h	ON	OFF	OFF	ON	ON	ON	OFF	OFF	630h
640h	ON	OFF	OFF	ON	ON	OFF	ON	ON	640h
650h	ON	OFF	OFF	ON	ON	OFF	ON	OFF	650h
660h	ON	OFF	OFF	ON	ON	OFF	OFF	ON	660h
670h	ON	OFF	OFF	ON	ON	OFF	OFF	OFF	670h
680h	ON	OFF	OFF	ON	OFF	ON	OFF	ON	680h
690h	ON	OFF	OFF	ON	OFF	ON	ON	OFF	690h
6A0h	ON	OFF	OFF	ON	OFF	ON	OFF	ON	6A0h
6B0h	ON	OFF	OFF	ON	OFF	ON	OFF	OFF	6B0h
6C0h	ON	OFF	OFF	ON	OFF	OFF	ON	ON	6C0h
6D0h	ON	OFF	OFF	ON	OFF	OFF	ON	OFF	6D0h
6E0h	ON	OFF	OFF	ON	OFF	OFF	OFF	ON	6E0h
6F0h	ON	OFF	OFF	ON	OFF	OFF	OFF	OFF	6F0h
700h	ON	OFF	OFF	OFF	ON	ON	ON	ON	700h
710h	ON	OFF	OFF	OFF	ON	ON	ON	OFF	710h
720h	ON	OFF	OFF	OFF	ON	ON	OFF	ON	720h
730h	ON	OFF	OFF	OFF	ON	ON	OFF	OFF	730h
740h	ON	OFF	OFF	OFF	ON	OFF	ON	ON	740h
750h	ON	OFF	OFF	OFF	ON	OFF	ON	OFF	750h
760h	ON	OFF	OFF	OFF	ON	OFF	OFF	ON	760h
770h	ON	OFF	OFF	OFF	ON	OFF	OFF	OFF	770h
780h	ON	OFF	OFF	OFF	OFF	ON	ON	ON	780h
790h	ON	OFF	OFF	OFF	OFF	ON	ON	OFF	790h
7A0h	ON	OFF	OFF	OFF	OFF	ON	OFF	ON	7A0h
7B0h	ON	OFF	OFF	OFF	OFF	ON	OFF	OFF	7B0h
7C0h	ON	OFF	OFF	OFF	OFF	OFF	ON	ON	7C0h
7D0h	ON	OFF	OFF	OFF	OFF	OFF	ON	OFF	7D0h
7E0h	ON	OFF	OFF	OFF	OFF	OFF	OFF	ON	7E0h
7F0h	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	7F0h

Table 6: Base address switch settings (continued).

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	Base Address
800h	OFF	ON	ON	ON	ON	ON	ON	ON	800h
810h	OFF	ON	ON	ON	ON	ON	ON	OFF	810h
820h	OFF	ON	ON	ON	ON	ON	OFF	ON	820h
830h	OFF	ON	ON	ON	ON	ON	OFF	OFF	830h
840h	OFF	ON	ON	ON	ON	OFF	ON	ON	840h
850h	OFF	ON	ON	ON	ON	ON	OFF	ON	850h
860h	OFF	ON	ON	ON	ON	OFF	OFF	ON	860h
870h	OFF	ON	ON	ON	ON	OFF	OFF	OFF	870h
880h	OFF	ON	ON	ON	OFF	ON	ON	ON	880h
890h	OFF	ON	ON	ON	OFF	ON	ON	OFF	890h
8A0h	OFF	ON	ON	ON	OFF	ON	OFF	ON	8A0h
8B0h	OFF	ON	ON	ON	OFF	ON	OFF	OFF	8B0h
8C0h	OFF	ON	ON	ON	OFF	OFF	ON	ON	8C0h
8D0h	OFF	ON	ON	ON	OFF	OFF	ON	OFF	8D0h
8E0h	OFF	ON	ON	ON	OFF	OFF	OFF	ON	8E0h
8F0h	OFF	ON	ON	ON	OFF	OFF	OFF	OFF	8F0h
900h	OFF	ON	ON	OFF	ON	ON	ON	ON	900h
910h	OFF	ON	ON	OFF	ON	ON	ON	OFF	910h
920h	OFF	ON	ON	OFF	ON	ON	OFF	ON	920h
930h	OFF	ON	ON	OFF	ON	ON	OFF	OFF	930h
940h	OFF	ON	ON	OFF	ON	OFF	ON	ON	940h
950h	OFF	ON	ON	OFF	ON	OFF	ON	OFF	950h
960h	OFF	ON	ON	OFF	ON	OFF	ON	ON	960h
970h	OFF	ON	ON	OFF	ON	OFF	OFF	OFF	970h
980h	OFF	ON	ON	OFF	OFF	ON	OFF	ON	980h
990h	OFF	ON	ON	OFF	OFF	ON	ON	OFF	990h
9A0h	OFF	ON	ON	OFF	OFF	ON	OFF	ON	9A0h
9B0h	OFF	ON	ON	OFF	OFF	ON	OFF	OFF	9B0h
9C0h	OFF	ON	ON	OFF	OFF	OFF	ON	ON	9C0h
9D0h	OFF	ON	ON	OFF	OFF	OFF	ON	OFF	9D0h
9E0h	OFF	ON	ON	OFF	OFF	OFF	OFF	ON	9E0h
9F0h	OFF	ON	ON	OFF	OFF	OFF	OFF	OFF	9F0h
A00h	OFF	ON	OFF	ON	ON	ON	ON	ON	A00h
A10h	OFF	ON	OFF	ON	ON	ON	ON	OFF	A10h
A20h	OFF	ON	OFF	ON	ON	ON	OFF	ON	A20h
A30h	OFF	ON	OFF	ON	ON	ON	OFF	OFF	A30h
A40h	OFF	ON	OFF	ON	ON	OFF	ON	ON	A40h
A50h	OFF	ON	OFF	ON	ON	OFF	ON	OFF	A50h
A60h	OFF	ON	OFF	ON	ON	OFF	OFF	ON	A60h
A70h	OFF	ON	OFF	ON	ON	OFF	OFF	OFF	A70h
A80h	OFF	ON	OFF	ON	OFF	ON	ON	ON	A80h
A90h	OFF	ON	OFF	ON	OFF	ON	ON	OFF	A90h
AA0h	OFF	ON	OFF	ON	OFF	ON	OFF	ON	AA0h
AB0h	OFF	ON	OFF	ON	OFF	ON	OFF	OFF	AB0h
AC0h	OFF	ON	OFF	ON	OFF	OFF	ON	ON	AC0h
AD0h	OFF	ON	OFF	ON	OFF	OFF	ON	OFF	AD0h
AE0h	OFF	ON	OFF	ON	OFF	OFF	OFF	ON	AE0h
AF0h	OFF	ON	OFF	ON	OFF	OFF	OFF	OFF	AF0h
B00h	OFF	ON	OFF	OFF	ON	ON	ON	ON	B00h
B10h	OFF	ON	OFF	OFF	ON	ON	ON	OFF	B10h
B20h	OFF	ON	OFF	OFF	ON	ON	OFF	ON	B20h
B30h	OFF	ON	OFF	OFF	ON	ON	OFF	OFF	B30h
B40h	OFF	ON	OFF	OFF	ON	OFF	ON	ON	B40h
B50h	OFF	ON	OFF	OFF	ON	OFF	ON	OFF	B50h
B60h	OFF	ON	OFF	OFF	ON	OFF	OFF	ON	B60h
B70h	OFF	ON	OFF	OFF	ON	OFF	OFF	OFF	B70h
B80h	OFF	ON	OFF	OFF	OFF	ON	ON	ON	B80h
B90h	OFF	ON	OFF	OFF	OFF	ON	ON	OFF	B90h
BA0h	OFF	ON	OFF	OFF	OFF	ON	OFF	ON	BA0h
BB0h	OFF	ON	OFF	OFF	OFF	ON	OFF	OFF	BB0h
BC0h	OFF	ON	OFF	OFF	OFF	OFF	ON	ON	BC0h
BD0h	OFF	ON	OFF	OFF	OFF	OFF	ON	OFF	BD0h
BE0h	OFF	ON	OFF	OFF	OFF	OFF	OFF	ON	BE0h
BF0h	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	BF0h

Table 6: Base address switch settings (continued).

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	Base Address
C00h	OFF	OFF	ON	ON	ON	ON	ON	ON	C00h
C10h	OFF	OFF	ON	ON	ON	ON	ON	ON	C10h
C20h	OFF	OFF	ON	ON	ON	ON	OFF	ON	C20h
C30h	OFF	OFF	ON	ON	ON	ON	OFF	OFF	C30h
C40h	OFF	OFF	ON	ON	ON	OFF	ON	ON	C40h
C50h	OFF	OFF	ON	ON	ON	OFF	ON	OFF	C50h
C60h	OFF	OFF	ON	ON	ON	OFF	OFF	ON	C60h
C70h	OFF	OFF	ON	ON	ON	OFF	OFF	OFF	C70h
C80h	OFF	OFF	ON	ON	OFF	ON	ON	ON	C80h
C90h	OFF	OFF	ON	ON	OFF	ON	ON	OFF	C90h
CA0h	OFF	OFF	ON	ON	OFF	ON	OFF	ON	CA0h
CB0h	OFF	OFF	ON	ON	OFF	ON	OFF	OFF	CB0h
CC0h	OFF	OFF	ON	ON	OFF	OFF	ON	ON	CC0h
CD0h	OFF	OFF	ON	ON	OFF	OFF	ON	OFF	CD0h
CE0h	OFF	OFF	ON	ON	OFF	OFF	OFF	ON	CE0h
CF0h	OFF	OFF	ON	ON	OFF	OFF	OFF	OFF	CF0h
D00h	OFF	OFF	ON	OFF	ON	ON	ON	ON	D00h
D10h	OFF	OFF	ON	OFF	ON	ON	ON	OFF	D10h
D20h	OFF	OFF	ON	OFF	ON	ON	OFF	ON	D20h
D30h	OFF	OFF	ON	OFF	ON	ON	OFF	OFF	D30h
D40h	OFF	OFF	ON	OFF	ON	OFF	ON	ON	D40h
D50h	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	D50h
D60h	OFF	OFF	ON	OFF	ON	OFF	OFF	ON	D60h
D70h	OFF	OFF	ON	OFF	ON	OFF	OFF	OFF	D70h
D80h	OFF	OFF	ON	OFF	OFF	ON	ON	ON	D80h
D90h	OFF	OFF	ON	OFF	OFF	ON	ON	OFF	D90h
DA0h	OFF	OFF	ON	OFF	OFF	ON	OFF	ON	DA0h
DB0h	OFF	OFF	ON	OFF	OFF	ON	OFF	OFF	DB0h
DC0h	OFF	OFF	ON	OFF	OFF	OFF	ON	ON	DC0h
DD0h	OFF	OFF	ON	OFF	OFF	OFF	ON	OFF	DD0h
DE0h	OFF	OFF	ON	OFF	OFF	OFF	ON	ON	DE0h
DF0h	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	DF0h
E00h	OFF	OFF	OFF	ON	ON	ON	ON	ON	E00h
E10h	OFF	OFF	OFF	ON	ON	ON	ON	OFF	E10h
E20h	OFF	OFF	OFF	ON	ON	ON	OFF	ON	E20h
E30h	OFF	OFF	OFF	ON	ON	ON	OFF	OFF	E30h
E40h	OFF	OFF	OFF	ON	ON	OFF	ON	ON	E40h
E50h	OFF	OFF	OFF	ON	ON	OFF	ON	OFF	E50h
E60h	OFF	OFF	OFF	ON	ON	OFF	OFF	ON	E60h
E70h	OFF	OFF	OFF	ON	ON	OFF	OFF	OFF	E70h
E80h	OFF	OFF	OFF	ON	OFF	ON	ON	ON	E80h
E90h	OFF	OFF	OFF	ON	OFF	ON	ON	OFF	E90h
EA0h	OFF	OFF	OFF	ON	OFF	ON	OFF	ON	EA0h
EB0h	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF	EB0h
EC0h	OFF	OFF	OFF	ON	OFF	OFF	ON	ON	EC0h
ED0h	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF	ED0h
EE0h	OFF	OFF	OFF	ON	OFF	OFF	OFF	ON	EE0h
EF0h	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	EF0h
F00h	OFF	OFF	OFF	OFF	ON	ON	ON	ON	F00h
F10h	OFF	OFF	OFF	OFF	ON	ON	ON	OFF	F10h
F20h	OFF	OFF	OFF	OFF	ON	ON	OFF	ON	F20h
F30h	OFF	OFF	OFF	OFF	ON	ON	OFF	OFF	F30h
F40h	OFF	OFF	OFF	OFF	ON	OFF	ON	ON	F40h
F50h	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	F50h
F60h	OFF	OFF	OFF	OFF	ON	OFF	OFF	ON	F60h
F70h	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	F70h
F80h	OFF	OFF	OFF	OFF	OFF	ON	ON	ON	F80h
F90h	OFF	OFF	OFF	OFF	OFF	ON	ON	OFF	F90h
FA0h	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	FA0h
FB0h	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	FB0h
FC0h	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	FC0h
FD0h	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	FD0h
FE0h	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	FE0h
FF0h	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	FF0h

Table 6: Base address switch settings (continued).

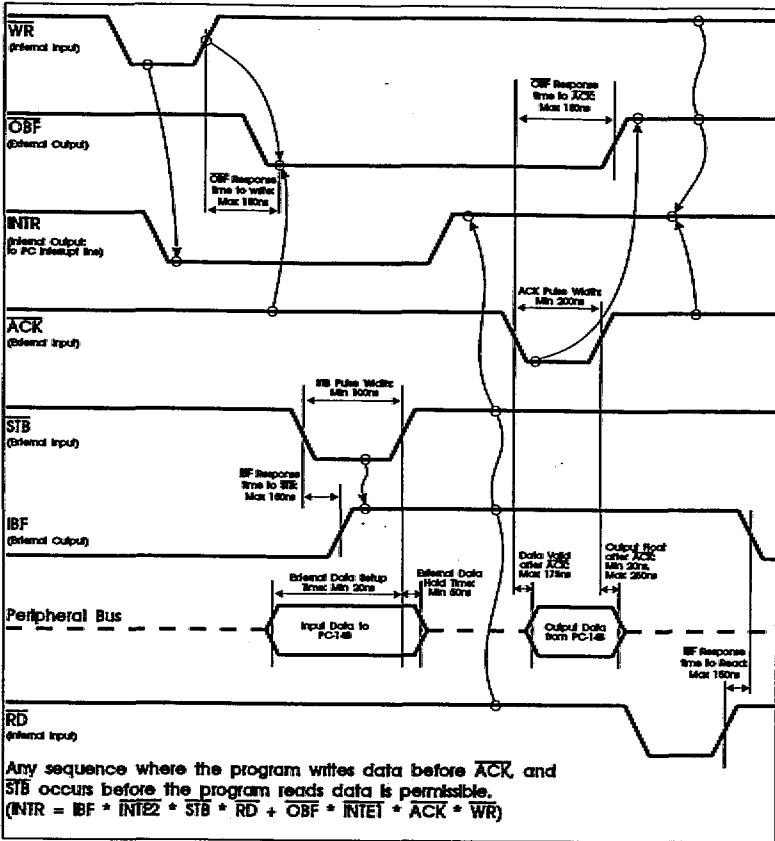


Figure 11: Bidirectional Bus (Mode 2).

Appendix C

User Serviceable Parts

The 8255's are mounted in sockets. If any external shorts or transients damage the PC-14 then replacing the 8255 PIA chip(s) may well correct the problem. The devices may be replaced with any 8255 equivalent, but to ensure the high speed specifications of the PC-14B it is recommended that the replacement is an NEC71055-10 or better.

Nothing more than a screwdriver is required to remove a device from its socket. When refitting the replacement device, make sure it is inserted the correct way around, with the notch in the device lined up with the notch in the socket and the indicator on the board legend.