# PC104-72A

## Enhanced PC104 72 Channel
## Digital I/O Board

# User's Manual

**For all PC104 Compatible Computers**

First Edition

November 1998

27 November 1998 Printing

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.

# TABLE OF CONTENTS

# Introduction

The PC104-72A is an enhanced PC104 72 Channel Digital I/O Board for PC104 Compatible Computers. It provides 72 Channels of Digital I/O using 8255 Programmable Peripheral Interface (PPI) ICs.

The PC104-72A also supports Mode 0, Mode 1 and Mode 2 method of data transfers.

The PC104-72A fits into any one of the PC104 compatible expansion slot.

**Typical applications**

Industrial control
Process Control
Laboratory Automation
Energy management
Product testing

**Key Features**

- ⦿ 72 Channels of Digital I/O channel
- ⦿ Background readback of PC0 and PC3 via Interrupt Mode
- ⦿ IRQ Mode via shared interrupts (PC3 [PPI1], PC0 [PPI2] and PC3 [PPI2]) or individual interrupt lines (PC0 [PPI0], PC3 [PPI0] and PC0 [PPI1])
- ⦿ Ideal Card for multitasking environments
- ⦿ Status of PC0/PC3 is only read when there is a state change in IRQ Mode
- ⦿ Host CPU time kept to a minimum
- ⦿ Interrupt selection IRQ2 thru IRQ7 can be automatically changed via software
- ⦿ Built-in Wait state generation for fast Bus clock speeds
- ⦿ Frequency response up to 10MHz
- ⦿ Address selectable
- ⦿ ID Status readback register available in order to ensure that the PC104-72A is operating correctly
- ⦿ TTL compatible address, data and control signals
- ⦿ Occupies 32 Byte locations in the I/O memory
- ⦿ Fuse protected +5V power available at the connectors to drive User circuits
- ⦿ CE Compliant PC104 Board
- ⦿ Driver libraries for DOS, Windows V3.11, Win ‘95 and Windows NT are supplied
- ⦿ Drivers for TestPoint, DasyLab, LabView, HP VEE are supplied
- ⦿ Autodetection and comprehensive Diagnostics Software provided to ensure proper operation of the PC104-72A
- ⦿ Demonstration examples supplied
- ⦿ Latest Software Update Drivers can be downloaded from our WebSite

## PC104-72A Package

**The PC104-72A package consists of:**

PC104-72A Interface Card
PC104-72A User's Manual
EDR Developers Toolkit User Manual + EDR Software on a CD

If any of the items is missing, contact your dealer immediately specifying which components are missing.

# Chapter 1: Installation

There are two aspects of the PC104-72A that must be configured using Jumpers. All other configurations are done via software:

## 1.1) Setting the base address

This address determines where the board is accessed. This can be set by a 8-way DIP switch found on the PC104-72A Board. The address range is from 0 to 1fffh.

The PC104-72A occupy a block of 32 consecutive Byte I/O addresses. The base address setting controls where the block starts. This base address must be unique to the PC104-72A only and no other card must occupy this address. If multiple PC104-72A boards are installed in one stackable PC104 System then each board must have a different base address.

The base address can be assigned to any location from 0 to 1fffh in 32 byte boundaries. Table 1 shows the I/O addresses occupied by standard interface cards. Refer to the Base Address Setting Table in Appendix A for a list of the various base address settings that the PC104-72A can occupy.

The base address setting can be set by adjusting the 8-way Dip Switch on the PC104-72A. Each line on the DIP switch compares an address line in I/O space. Switch Number 1 compares Address line A12; switch number 2: Address A11 while Switch Number 8 compares address line A5. Factory default setting is 700h.

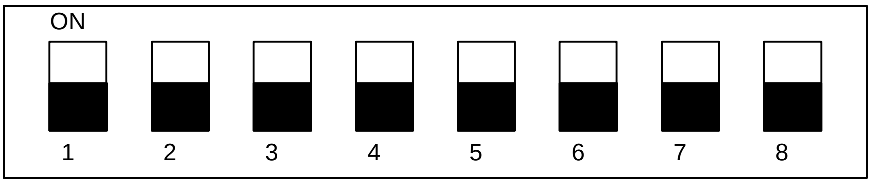
If one of the switches is OFF, then it contributes to the base address. An example is as follows:

| | | |
|---|---|---|
| Base Address | = | Switch 5(A8) + Switch 4 (A9) + Switch 3 (A10) |
| | = | 100 + 200 + 300 |
| | = | 700h |

## Table 1: Addresses for standard I/O devices

| Address | Standard device |
|---|---|
| 000-1FF | Internal system board |
| 200-20F | Games port |
| 210-217 | Expansion unit |
| 220-24F | Reserved |
| 250-257 | 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 comms |
| 3B0-3BF | Mono Display Adaptor |
| 3C0-3CF | Reserved |
| 3D0-3DF | CGA |
| 3E0-3E7 | Reserved |
| 3E3-3EF | Not assigned |
| 3F0-3F7 | Floppy disk |
| 3F8-3FF | COM1 |
| 400-FFF | Not used see below |

*Table 1: Standard I/O Addresses*



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1000 | 800 | 400 | 200 | 100 | 80 | 40 | 20 | Hex Weight when OFF |
| A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | Corresponding Address line |

*Figure 1: Dip Switch Address Weight (in Hex)*

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

The PC104-72A (and most other members of the PC-XX family) can use these address, if and only if the board at address 400h less than the address of the PC104-72A also decodes the extra address.

For example, a PC104-72A can be installed at address 300h and another one at address 700h (400h locations apart). However, it would not be advisable to install a PC104-72A at address 7F8h. This is because communications port COM1 is installed at 3F8h and does not normally decode these extra addresses.

If your computer has boards not listed in Table 1 (such as LAN adaptors, back-up boards or other engineering boards), you should consult the User's Manual for these boards for information on the address ranges used.

In most cases, base address of 300h is a good choice. Address 300h is also the factory default base address setting.

### 1.2) Wait State Generation on the PC104-72A (JP2)

Some computers have very high I/O bus cycles. In this case it is neccessary to slow down these cycles when the computer accesses the PC104-72A Board. Additional wait states can be set by means of a jumper JP2 on the PC104-72A Board.

Additional wait states can be inserted in the I/O bus cycle by changing the jumper setting on JP2 on the PC104-72A. This jumper is marked 'Wait State Jumper' on the PC104-72A board. Refer to figure 1.2 for the Wait State Jumper settings. Note that the factory default setting is zero wait states.

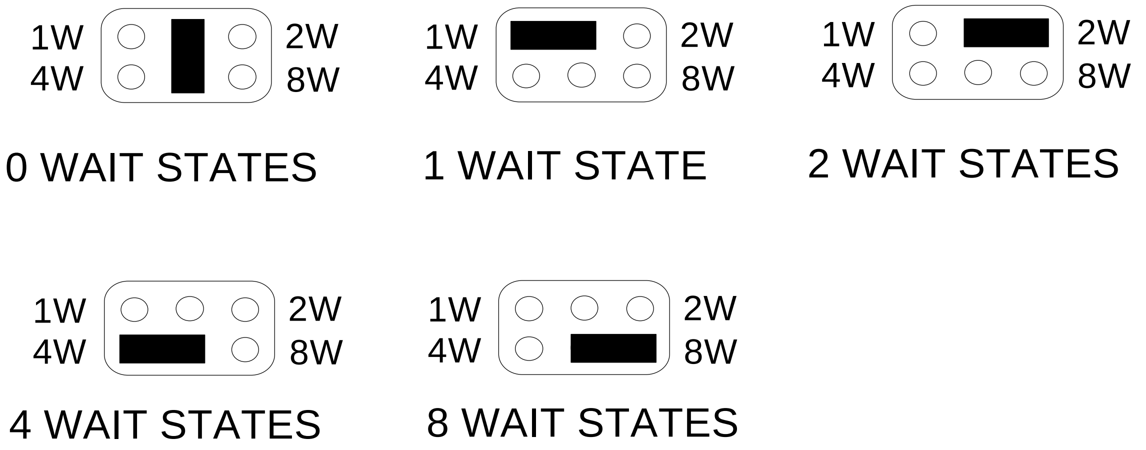


*Figure 1.2: Wait State Jumper config (JP2)*

In most cases, only a very small number of computers require additional wait states. If the PC104-72A seems to be giving incorrect results then try increasing the wait states until correct results are obtained. If the board still does not produce correct results even after the maximum number of wait states has been inserted then the PC104-72A or the host computer are defective and should be serviced.

In a standard PC104 PC, the interrupt levels are allocated as follows:

| Level Level | Allocation |
|---|---|
| IRQ0 | System timer |
| IRQ1 | Keyboard |
| IRQ2/IRQ9 | Display Adaptor |
| IRQ3 | COM1 (if installed) |
| IRQ4 | COM2 (if installed) |
| IRQ5 | LPT2 (if installed) |
| IRQ6 | Floppy disk controller |
| IRQ7 | LPT1 (if installed) |
| IRQ10 | Not used |
| IRQ11 | Not used |
| IRQ12 | Used by PS/2 Mouse (if installed) |
| IRQ13 | Coprocessor |
| IRQ14 | Primary IDE Harddisk (if installed) |
| IRQ15 | Secondary IDE Harddisk (if installed) |

*Table 1.3b: Standard Interrupt Settings*

On PC104 Compatible ISA Computers, IRQ2 is used by the system and any interrupt requests on IRQ2 is transparently rerouted to IRQ9.

The default IRQ setting on the PC104-72A is none. Note that unless the interrupt line is specifically enabled by software, the interrupt output from the PC104-72A is tri-stated (ie: not connected). It is also tri-stated upon power-up.

### 1.3) Connecting the PC104-72A in a stackable PC104 System

Requirements: PC104 Compatible Computer
PC104 Compatible Slot (8 or 16 Bit)

**Procedure:**

a) Switch off the computer and all attached devices
b) Unplug power cord from the PC104 Computer Power Supply and all attached devices.

☞ **Warning**

Failure to disconnect all power cables can result in harzardous conditions, as there may be dangerous voltage levels present in externally connected cables.

c) Firmly press the PC104-72A down into the PC104 Socket.

d) Insert the four Standoffs between the stackable PC104 Adaptor and the PC104-72A.

e) Fasten the PC104-72A using the screws provided (if and only if the PC104-72A is the last adaptor in the PC104 Stackable System).

f) The PC104-72A is now installed.

# CHAPTER 2: Interconnections

## 2.0) Introduction

The PC104-72A 72 Channel Digital I/O board plugs into any PC104 compatible slot at J1/J2 and J3/J4. The board communicates to the User Circuit via IDE connectors mounted on the PC board. This chapter describes these connectors.

## 2.1) Connections to the PC104 Bus

The PC104-72A board may be plugged into any 8 or 16 bit PC104 Compatible Slot. All data transfers to and from the host computer are channelled via these connectors (J1/J2/J3/J4).

## 2.2) PC104-72A PPI0/PPI1 IDC50 Male User Connector 1 (JP3)

The PC104-72A PPI0 and PPI1 interfaces to the external world via an IDC50 Male Connector.

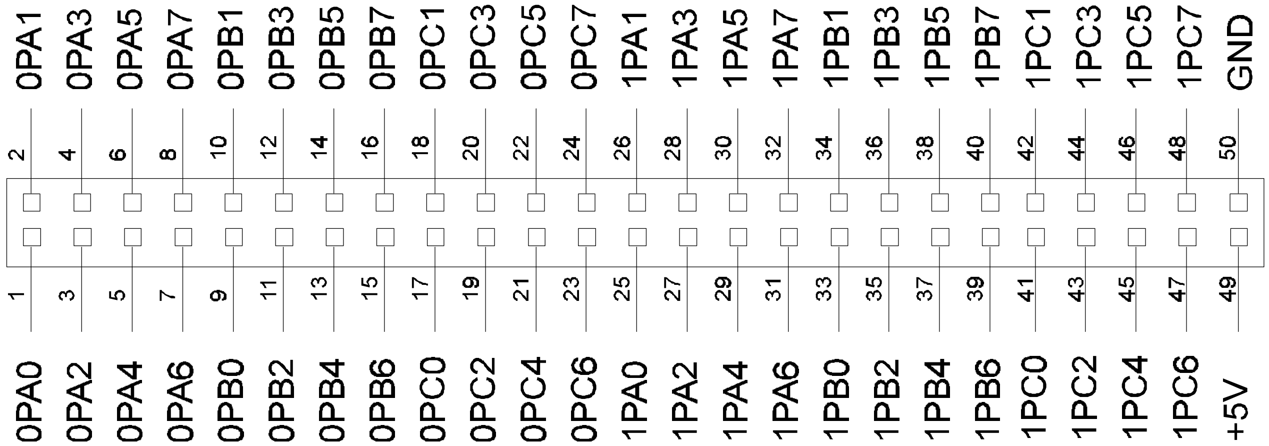


Figure 2.2: PC104-72A Interface Connector for PPI0 and PPI1

Figure 4 graphically shows the connector together with their pin assignments. Note that the pin connections refer to the pin numbers of the connector when looking into the connector from the rear of the computer. Also note that the pin numbers are also found on the PCB Silkscreen.

The +5V power and Ground lines are available on the connector. Precautionary measures should be taken when using these supplies.

**Warning**

The maximum permissible current drawn on the I/O connector (P1) for the +5V supply is 200mA. Exceeding this can cause irreparable damage to the PC104-72A and your computer.

## 2.4) PC104-72A PPI2 IDC40 Male Header User Connector 1 (JP4)

The PC104-72A PPI2 interfaces to the external world via an IDC40 Male Connector..

A logical Low is from 0 to 0.8V and a Logical High is from 2.0V to +5.1V. Although these are defined as TTL Levels it is recommended that a logical Low voltage is from 0V to 0.3V and a logical high is from 4.5V to 5.1V.

The digital Inputs and Outputs can be read and written as single 8 Bits or 1 byte.

Connector Header (JP4) details are shown below:

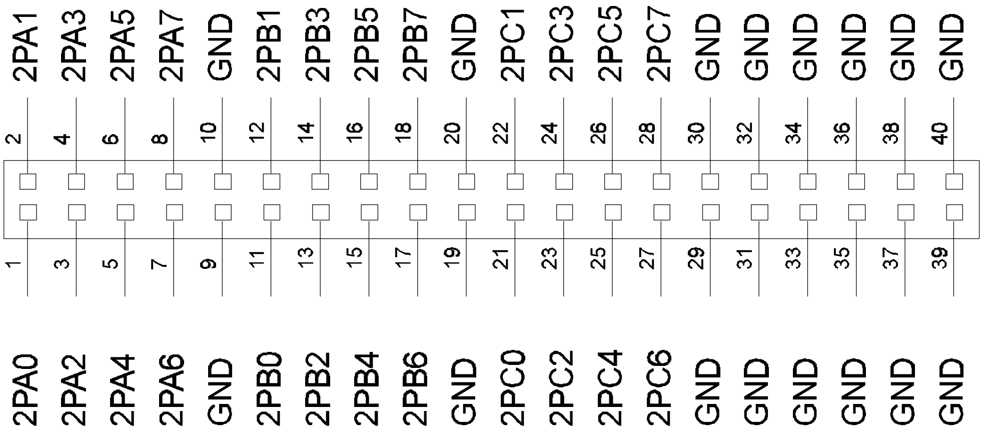


Figure 2.4a: PPI2 Digital I/O Header (JP4)

All unused digital inputs should be grounded via a 1k Resistor. The digital Outputs are capable of driving 2mA in a Low State.

**Warning**

DO NOT exceed 5.1V or fall below 0V on the I/O ports of the PC104-72A. Permanent damage will result if these thresholds are exceeded.

## 2.5) Power supply connections

The +5V power and digital ground are available on the IDC50 (JP3) and the IDC40 (JP4) Connector. These are equipped with Polyswitches that will open circuit if the max permissible current of 200mA is exceeded. Connections to the power lines will be restored if the current falls below the max permissible current draw.

### ✋ Warning

The maximum permissible current draw on the +5V lines on the I/O connectors is 200mA.

# CHAPTER 3: Register Structure

## 3.0) Introduction

At the lowest level, the PC104-72A can be programmed using I/O input and output instructions. This chapter contains the information on all the PC104-72A registers. Although programming the board is not difficult, it is time consuming and requires detailed knowledge of the PC104-72A as well as the operation of the host PC and its operating system. In order to simplify the process, a set of driver libraries is provided. The use of these libraries allows access to all the board's functions and is described in Chapter 5: Programming guide.

The PC104-72A occupies 32 consecutive byte addresses in the computer's I/O space. The layout of these registers is shown in Table 4: PC104-72A Register Structure. The offset of the registers is given as offset addresses from the base address of the board. This base address is set with the DIP Switch as detailed in Chapter 2: Installation.

| Offset | Read | Write |
|---|---|---|
| 0 | PPI 0: Port A (DIOA0) | PPI 0: Port A (DIOA0) |
| 1 | PPI 0: Port B (DIOB0) | PPI 0: Port B (DIOB0) |
| 2 | PPI 0: Port C (DIOC0) | PPI 0: Port C (DIOC0) |
| 3 | PPI 0: Control Register (DIO0CTL) | |
| 4 | PPI 1: Port A (DIO0A1) | PPI 1: Port A (DIOA1) |
| 5 | PPI 1: Port B (DIO0B1) | PPI 1: Port B (DIOB1) |
| 6 | PPI 1: Port C (DIO0C1) | PPI 1: Port C (DIOC1) |
| 7 | PPI 1: Control Register (DIO1CTL) | |
| 8 | PPI 2: Port A (DIOA2) | PPI 2: Port A (DIOA2) |
| 9 | PPI 2 Port B (DIOB2) | PPI 2: Port B (DIOB2) |
| 10 | PPI 2: Port C (DIOC2) | PPI 2: Port C (DIOC2) |
| 11 | PPI 2: Control Register (DIO2CTL) | |
| 12 | IRQ Set Status Register (ISETXRD) | |
| 13 | IRQ Gate Control Register (IGATE) | IRQ Gate Control Register (IGATE) |
| 14 | Local Mux IRQ Register 0 (IMUXP0) | Local Mux IRQ Register 0 (IMUXP0) |
| 15 | Local Mux IRQ Register 1 (IMUXP1) | Local Mux IRQ Register 1 (IMUXP1) |
| 16 | IRQ Reset Register (IRQRES0) | IRQ Reset Status Register (IRQRES0) |
| 17 | Reserved | Reserved |

*Table 4: PC104-72A Register Structure*

### 3.1) DIOA0 - Port A Register of first PPI (offset 0, read/write)

This register is port A of the first PPI. It can be operated in one of three modes which is set by writing to the DIO0CTRL register (Offset 4). The DIOA0 Register is described below.

**DIOA0 Register (read/write)**

| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
|---|---|---|---|---|---|---|---|
| 0A7 | 0A6 | 0A5 | 0A4 | 0A3 | 0A2 | 0A1 | 0A0 |

Bit 0 thru 7: The bits 0A7 (MSB) down to 0A0 (LSB) reflect the status of Port A I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs or bidirectional.

### 3.2) DIOB0 - Port B Register of first PPI (offset 1, read/write)

This register is port B of the first PPI. It can be operated in one of three modes which is set by writing to the DIO0CTRL register (offset 4). DIOB0 Register is described below.

**DIOB0 Register (read/write)**

| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
|---|---|---|---|---|---|---|---|
| 0B7 | 0B6 | 0B5 | 0B4 | 0B3 | 0B2 | 0B1 | 0B0 |

Bit 0 thru 7: The bits 0B7 (MSB) down to 0B0 (LSB) reflect the status of Port B I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs or bidirectional.

### 3.3) DIOC0 - Port C Register of first PPI (offset 2, read/write)

This register is port C of the first PPI. It can be operated in one of three modes which is set by writing to the DIO0CTRL register (offset 4). The DIOC0 Register is described below.

**DIOC0 Register (read/write)**

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

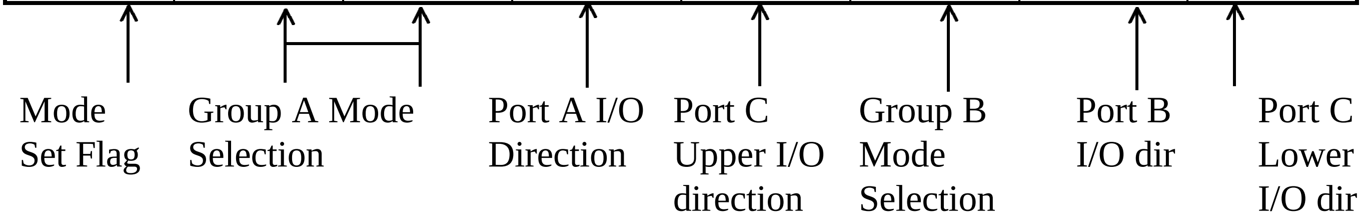
Bit 0 thru 7: The bits 0C7 (MSB) down to 0C0 (LSB) reflect the status of Port C I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs or bidirectional.

### 3.4) DIO0CTRL - Control register of the first PPI (offset 3, write only)

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

**DIO0CTL Register (write only) - Configuration Mode**

| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
|---|---|---|---|---|---|---|---|
| 1 | MAS$_1$ | MSA$_0$ | PA | PCU | MSB$_0$ | PB | PCL |



Mode Set Flag

Group A Mode Selection

Port A I/O Direction

Port C Upper I/O direction

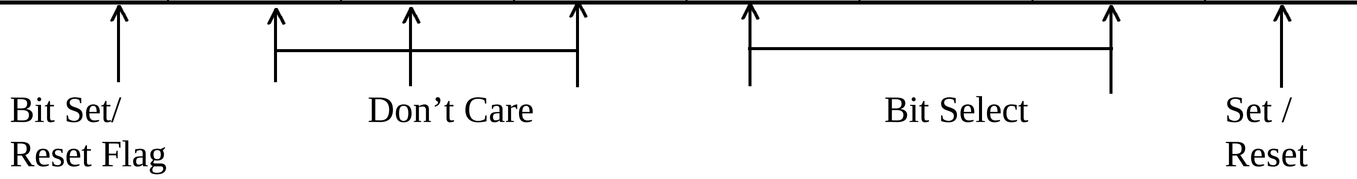
Group B Mode Selection

Port B I/O dir

Port C Lower I/O dir

**DIO0CTL Register (write only) - Bit Set/Reset Mode**

| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
|---|---|---|---|---|---|---|---|
| 0 | x | x | x | BS2 | BS1 | BS0 | S/R |



Bit Set/ Reset Flag

Don’t Care

Bit Select

Set / Reset

Bit 0 thru 7: Function Select: Setting this bit to 1 set the register in Configuration Mode. Writing a 0 to this bit configures this register in Set/Reset Mode.

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

### 3.4a) 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 combination 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 |

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 functions as inputs. If the bit is 0, then the lines become outputs.

☞ **Note**

*Group B can only be used for simple or strobed I/O*

### 3.4b) Bit Set/Reset Mode

Bits 6-5: These bits have not 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.

### 3.5) DIOA1 - Port A Register of 2nd PPI (offset 4, read/write)

This register is port A of the 2nd PPI. It can be operated in one of three modes which is set by writing to the DIO1CTRL register (Offset 7). The DIOA1 Register is described below.

**DIOA1 Register (read/write)**

| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
|---|---|---|---|---|---|---|---|
| 0A7 | 0A6 | 0A5 | 0A4 | 0A3 | 0A2 | 0A1 | 0A0 |

Bit 0 thru 7: The bits 0A7 (MSB) down to 0A0 (LSB) reflect the status of Port A I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs or bidirectional.

### 3.6) DIOB1 - Port B Register of 2nd PPI (offset 5, read/write)

This register is port B of the 2nd PPI. It can be operated in one of three modes which is set by writing to the DIO1CTRL register (offset 7). DIOB1 Register is described below.

**DIOB1 Register (read/write)**

| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
|---|---|---|---|---|---|---|---|
| 0B7 | 0B6 | 0B5 | 0B4 | 0B3 | 0B2 | 0B1 | 0B0 |

Bit 0 thru 7: The bits 0B7 (MSB) down to 0B0 (LSB) reflect the status of Port B I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs or bidirectional.

### 3.7) DIOC1 - Port C Register of 2nd PPI (offset 6, read/write)

This register is port C of the 2nd PPI. It can be operated in one of three modes which is set by writing to the DIO2CTRL register (offset 7). The DIOC1 Register is described below.

**DIOC1 Register (read/write)**

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

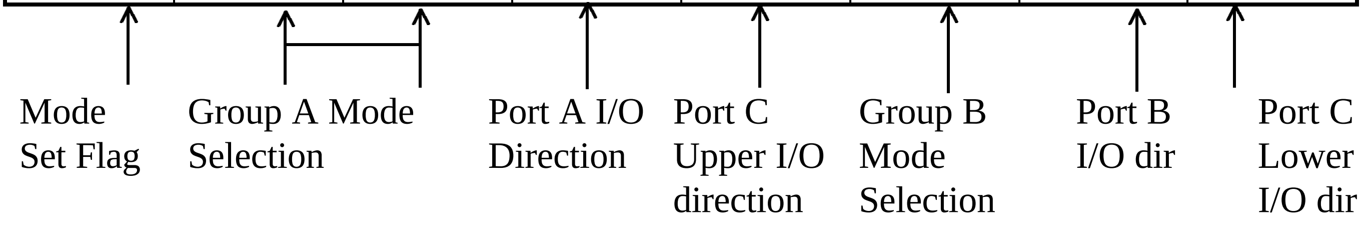
Bit 0 thru 7: The bits 0C7 (MSB) down to 0C0 (LSB) reflect the status of Port C I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs or bidirectional.

### 3.8) DIO1CTRL - Control register of the 2nd PPI (offset 7, write only)

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

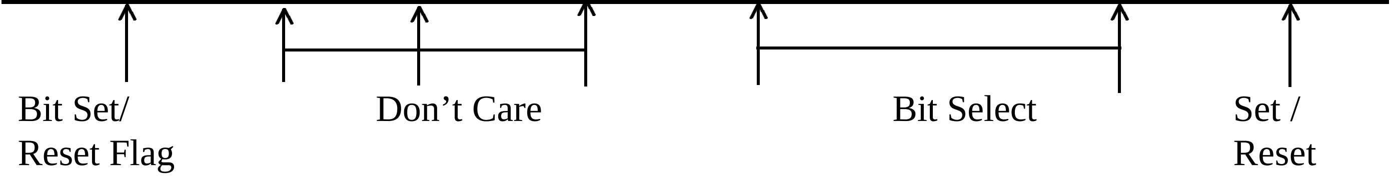
**DIO1CTL Register (write only) - Configuration Mode**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| 1 | MAS$_1$ | MSA$_0$ | PA | PCU | MSB$_0$ | PB | PCL |



**DIO1CTL Register (write only) - Bit Set/Reset Mode**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | X | x | x | BS2 | BS1 | BS0 | S/R |



Bit 0 thru 7: Function Select: Setting this bit to 1 set the register in Configuration Mode. Writing a 0 to this bit configures this register in Set/Reset Mode.

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

### 3.8a) 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 combination 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 |

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 functions as inputs. If the bit is 0, then the lines become outputs.

☞ **Note**

*Group B can only be used for simple or strobed I/O*

### 3.8b) Bit Set/Reset Mode

Bits 6-5: These bits have not 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.

### 3.9) DIOA2 - Port A Register of 3rd PPI (offset 8, read/write)

This register is port A of the 3rd PPI. It can be operated in one of three modes which is set by writing to the DIO1CTRL register (Offset 11). The DIOA2 Register is described below.

**DIOA2 Register (read/write)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| 0A7 | 0A6 | 0A5 | 0A4 | 0A3 | 0A2 | 0A1 | 0A0 |

Bit 0 thru 7: The bits 0A7 (MSB) down to 0A0 (LSB) reflect the status of Port A I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs or bidirectional.

### 3.10) DIOB2 - Port B Register of 3rd PPI (offset 9, read/write)

This register is port B of the 3rd PPI. It can be operated in one of three modes which is set by writing to the DIO2CTRL register (offset 11). DIOB2 Register is described below.

**DIOB2 Register (read/write)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| 0B7 | 0B6 | 0B5 | 0B4 | 0B3 | 0B2 | 0B1 | 0B0 |

Bit 0 thru 7: The bits 0B7 (MSB) down to 0B0 (LSB) reflect the status of Port B I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs or bidirectional.

### 3.11) DIOC2 - Port C Register of 3rd PPI (offset 10, read/write)

This register is port C of the 3rd PPI. It can be operated in one of three modes which is set by writing to the DIO2CTRL register (offset 11). The DIOC2 Register is described below.

**DIOC2 Register (read/write)**

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

Bit 0 thru 7: The bits 0C7 (MSB) down to 0C0 (LSB) reflect the status of Port C I/O lines. Depending on the programmed I/O mode of the port, the lines may be inputs, outputs or bidirectional.

### 3.12) DIO2CTRL - Control register of the 3rd PPI (offset 11, write only)

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

**DIO2CTL Register (write only) - Configuration Mode**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| 1 | $MAS_1$ | $MSA_0$ | PA | PCU | $MSB_0$ | PB | PCL |

Mode Set Flag

Group A Mode Selection

Port A I/O Direction

Port C Upper I/O direction

Group B Mode Selection

Port B I/O dir

Port C Lower I/O dir

**DIO2CTL Register (write only) - Bit Set/Reset Mode**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | X | x | x | BS2 | BS1 | BS0 | S/R |

Bit Set/ Reset Flag

Don’t Care

Bit Select

Set / Reset

Bit 0 thru 7: Function Select: Setting this bit to 1 set the register in Configuration Mode. Writing a 0 to this bit configures this register in Set/Reset Mode.

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

### 3.12a) 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 combination 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 |

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 functions as inputs. If the bit is 0, then the lines become outputs.

☞ **Note**

*Group B can only be used for simple or strobed I/O*

### 3.12b) Bit Set/Reset Mode

Bits 6-5: These bits have not 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.

### 3.13) ISETX – IRQ Set Status Register (offset 12, read / write)

This register is used to determine whether a positive level shift did occur on Port Lines PC0 and PC3 on the 1st, 2nd PPI and 3rd PPIs.

**ISETX Register (read mode)**

| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
|---|---|---|---|---|---|---|---|
| x | PAY6 | PAY5 | PAY4 | PAY3 | PAY2 | PAY1 | PAY0 |
| x | GLOBAL | PC3 | PC0 | PC3 | PC0 | PC3 | PC0 |
| x | Any PPI | PPI2 | PPI2 | PPI1 | PPI1 | PPI0 | PPI0 |

Bits 0-5: PAY0 to PAY5 - These bits determine whether a positive edge has occurred on PC0 and PC3 on the 1st , 2nd and 3rd PPIs. If a high is detected on any of the bits then a positive transition has occurred on that line.

If one of the bits is set, it can result in an interrupt occuring if the IGATE0 to IGATE3 Bits (ie: Port C0 and C3 on PPI0 and PPI1) in the IGATE Register (offset 13) are enabled. You can clear the PAY0 to PAY5 registers by writing a 0 and then a 1 to the RESP0 to RESP5 bit in the IRQRES0 Register (Offset 16) respectively.

Bit 6: PAY6 - This bit determine whether a positive edge has occurred on any of PC0 or PC3 on the 1st , 2nd and 3rd PPIs. If a high is detected on any of the PAY0 to PAY5 bits then a positive transition has occurred on that line resulting in PAY6 flip-flop being set. PAY6 flip-flip is also mapped to an interrupt. Thus if the IGATE3 in the IGATE register is enabled then an interrupt will occur if any of the above port bits result in a positive edge (high).

✋ **Warning**

*Do not exceed the min/max voltage specification fed into the digital input lines. The absolute minimum voltage is –0.1V and the absolute maximum voltage is 5.2V. Exceeding the above specifications will damage the PC104-72A.*

## 3.14) IGATE – IRQ Gate Control Register (offset 13, read/write)

This register is used to enable the Gate controlling the Muxes that map Port C0 and C3 of PPI0, PPI1 and PPI2 to a specific IRQ line. The MUX Control Registers (3 lines per Input) is found in the IMUX0 (offset 14) and IMUX1 (offset 15) Control Registers. For example: In the IMUX0 Control Register, bits 0 to 2 set the IRQ Line of the PC0 of PPI0. If the IGATE0 bit is 1 then a rising edge on the PC0 of PPI0 will be mapped to an IRQ Line set via Bits 0 to 2 in the IMUX0 Register. If IGATE0 bit is 0 then the IRQ Lines are effectively disconnected from the IRQ Bus. This means that a rising edge on PC0 of PPI0 will NOT be mapped to an IRQ Line set via Bits 0 to 2 in the IMUX0 Register.

Note that IGATE3 is used to enable the shared interrupt line. If a positive edge occurred on any of the PC0 or PC3 lines on any PPI then the PAY6 flip-flop in ISETXRD Register [offset 12] will be set and an IRQ generated (assuming the IGATE3 in the IGATE register [offset 13] is enabled). Interrupt sharing is a useful feature if limited IRQ Lines are available. RESP6 in the IRQRES0 Register [offset 16] resets the global shared interrupt flip-flop. You can also determine which Port C0 and C3 lines became active by reading the ISETXRD Register (offset 12)

**IGATE Register (write mode)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| X | x | x | x | IGATE3 | IGATE2 | IGATE1 | IGATE0 |

**IGATE Register (read mode)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | IGATE3 | IGATE2 | IGATE1 | IGATE0 |

Bit0: IGATE0 – This bit is used to enable/disable Interrupt support for PC0 of PPI0. Setting this bit to 1 enables one to select any one of 4 IRQ on the ISA Bus. The actual IRQ selection is determined by the IMUXP0 Register (offset 14). For example, bits 0 to 2 sets the IRQ selection for PC0 (Bit Port C0) of PPI0, with IGATE0 enabling/disable interrupt support. Table 3.14a determines the IRQ selection. Setting the IGATE0 bit to 0 disables Interrupts support for PC0 (Bit Port C0) of PPI0.

Reading this bit determines the status of the IGATE0 bit.

| IMUXP0 | IMUXP0 | IMUXP0 | IGATE0 | IRQ |
|---|---|---|---|---|
| **Bit 2** | **Bit 1** | **Bit 0** | **IGATE0** | **IRQ** |
| 0 | 0 | 0 | 1 | 2 |
| 0 | 0 | 1 | 1 | 7 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 1 | 4 |
| 1 | 0 | 0 | 1 | 3 |

***Table 3.14a: Mux Table for setting IRQs***

Bit1: IGATE1 – This bit is used to enable/disable Interrupt support for PC3 (Port C Bit 3) of PPI0. Setting this bit to 1 enables one to select any one of 4 IRQ on the ISA Bus. The actual IRQ selection is determined by the IMUXP0 Register (offset 14). For example, bits 3 (low) to 5 (high) sets the IRQ selection for PC3 (Port C Bit 3) of PPI0, with IGATE1 enabling/disable interrupt support. Table 3.14a determines the IRQ selection. Setting the IGATE1 bit to 0 disables Interrupts support for PC3 (Port C Bit 3) of PPI0.

Reading this bit determines the status of the IGATE1 bit.

Bit2: IGATE2 – This bit is used to enable/disable Interrupt support for PC0 (Port C Bit 0) of PPI1. Setting this bit to 1 enables one to select any one of 4 IRQ on the ISA Bus. The actual IRQ selection is determined by the IMUXP1 Register (offset 15). For example, bit 0 (A0) is the low order bits in the decoder and Bit2 in the IMUXP1 Register is the high bit in the decoder. These bits set the IRQ selection for PC0 (Port C Bit 0) of PPI1, with IGATE2 enabling/disable interrupt support. Setting the IGATE2 bit to 0 disables Interrupts support for PC0 (Port C Bit 3) of PPI1.

Reading this bit determines the status of the IGATE2 bit.

| IMUXP1 | IMUXP1 | IMUXP1 | IGATE3 | IRQ |
|---|---|---|---|---|
| **Bit 2** | **Bit 2** | **Bit 0** | **Bit 3** | **IRQ** |
| 0 | 0 | 0 | 1 | 2 |
| 0 | 0 | 1 | 1 | 7 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 1 | 4 |
| 1 | 0 | 0 | 1 | 3 |

***Table 3.14b: Mux Table for setting IRQs on shared IRQ Mode***

Bit3: IGATE3 – This bit is used to enable/disable shared Interrupt support for PC0 and PC3 of any of the PPIs. These lines share one interrupt line. In other words, a low-high transition on any of these bits will enable an IRQ to occur. Setting this bit to 1 enables one to select any of 4 IRQ on the PC104 Bus. The actual IRQ selection is determined by the IMUXP1 Register (offset 15). For example, bits 3 (low) to 5 (high) set the IRQ selection for any of port bits C0 of C3 of PPI0, PPI1 and PPI2 with IGATE3 enabling/disable global interrupt support. Table 3.14b determines the IRQ selection. Setting the IGATE3 bit to 0 disables Interrupt support.

Reading this bit determines the status of the IGATE3 bit.

## 3.15) IMUXP0 – Local Mux Interrupt Control Register (offset 14, read/write)

This register is used to set individual interrupts for Port C0 and C3 of PPI0, PPI1 and PPI2 to a specific IRQ line. Effectively MUXP0 is used as address lines to map Port C0 and C3 of PPI0, Port C0 of PPI1 and Port C3 of PPI1 or Port C0 of PPI2 or Port C3 of PPI2 to a specific IRQ. If any one of the Port Lines was enabled, the pulse is set up and vectored to the Decoder to a specific IRQ. This register configures this line to any of 4 IRQs lines.

**IMUXP0 Register (write mode)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | MUXP1A2 | MUXP1A1 | MUXP1A0 | MUXP0A2 | MUXP0A1 | MUXP0A0 |

**IMUXP0 Register (read mode)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| x | x | MUXP1A2 | MUXP1A1 | MUXP1A0 | MUXP0A2 | MUXP0A1 | MUXP0A0 |

Bits 0-2: MUXP0A0 to MUXP0A2 - These bits are used to set up an IRQ line for the PC0 (Port C bit 0) of PPI0. Bits 0 to 2 sets the IRQ selection for PC0, with IGATE0 enabling/disable interrupt support. IGATE0 must be enabled in order to select an interrupt. Table 3.15a determines the appropriate writes to Bits 0 to 2 to select 1 of 4 IRQs.

Reading these bits determines the status of MUXP0A0 to MUXP0A2.

| MUXP0A2 | MUXP0A1 | MUXP0A0 | IGATE0 | IRQ |
|---|---|---|---|---|
| **Bit 2** | **Bit 1** | **Bit 0** | | |
| **A2** | **A1** | **A0** | **Gx** | |
| 0 | 0 | 0 | 1 | 2 |
| 0 | 0 | 1 | 1 | 7 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 1 | 4 |
| 1 | 0 | 0 | 1 | 3 |

***Table 3.15a: Mux Table for setting IRQs for PC0 of PPI0***

Bits 3-5: MUXP1A0 to MUXP1A2 - These bits are used to set up an IRQ line for the PC3 (Port C bit 3) of PPI0. Bits 3 to 5 sets the IRQ selection for PC3, with IGATE1 enabling/disable interrupt support. IGATE1 must be enabled in order to select an interrupt. Table 3.16a gives an example of the appropriate writes from Bit 3, Bit 4 and Bit 5 in order to select 1 of 4 IRQs. Bit 3 is Address Decode A0 (IMUXP0 Register), Bit 4 (IMUXP0 Register) is Address Decode A1, Bit 5 (IMUXP0 Register) is Address Decode A2 and Gx is IGATE2

Reading these bits determines the status of MUXP1A0 to MUXP1A2 bits.

Bits 6-7: These bits are undefined. Write a 0 to these bits in order to maintain future compatibility.

Reading these bits is undefined.

### 3.16) IMUXP1 – Local Mux Interrupt Control Register 2 (offset 15, read/write)

This register is used to set individual interrupts for Port C0 and C3 of PP0, PPI1 and PPI2 to a specific IRQ line. Effectively MUXP0 is used as address lines to map Port C0 and C3 of PPI0, Port C0 of PPI1. Port C0 and C3 of all PPIs is also mapped to one specific IRQ in shared IRQ Mode. If any one of the Port Lines was enabled, the pulse is set up and vectored to the Decoder to a specific IRQ. This register configures this line to any of 4 IRQs lines.

**IMUXP1 Register (write mode)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| x | x | MUXP3A2 | MUXP3A1 | MUXP3A0 | MUXP2A2 | MUXP2A1 | MUXP2A0 |

**IMUXP1 Register (read mode)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| x | x | MUXP3A2 | MUXP3A1 | MUXP3A0 | MUXP2A2 | MUXP2A1 | MUXP2A0 |

Bits 0-2: MUXP2A0 to MUXP2A2 - These bits are used to set up an IRQ line for the PC0 (Port C bit 0) of PPI1. Bits 0 to 2 sets the IRQ selection for PC0, with IGATE2 enabling/disable interrupt support. IGATE2 must be enabled in order to select an interrupt. Table 3.16a determines the appropriate writes to Bits 0 to 2 to select 1 of 4 IRQs.

Reading these bits determines the status of MUXP2A0 to MUXP2A2.

Bits 3-5: MUXP3A0 to MUXP3A2 - These bits are used to set up an IRQ line for shared IRQ Mode. In other words, an IRQ will result if any of the PC0 or PC3 lines on PPI0, PPI1 or PPI2 results in a positive edge (assuming IGATE3 is enabled). Bits 3 to 5 set the IRQ selection for PC0 or PC3 of any PPI, with IGATE3 enabling/disable interrupt support. IGATE3 must be enabled in order to select an interrupt. Table 3.16a gives an example of the appropriate writes from Bit 3, Bit 4 and Bit 5 in order to select 1 of 4 IRQs. Bit 3 is Address Decode A0 (IMUXP1 Register), Bit 4 (IMUXP1 Register) is Address Decode A1, Bit 5 (IMUXP1 Register) is Address Decode A2 and Gx is IGATE3

Reading these bits determines the status of MUXP3A0 to MUXP3A2 bits.

| Address A2 | Address A1 | Address A0 | Gate Cntrl | IRQ |
|---|---|---|---|---|
| MUXP0A2 (Bit2) | MUXP0A1 (Bit1) | MUXP0A0 (Bit0) | IGATE0 (Bit0) | |
| MUXP1A2 (Bit5) | MUXP1A1 (Bit4) | MUXP1A0 (Bit3) | IGATE1 (Bit1) | |
| MUXP2A2 (Bit2) | MUXP2A1 (Bit1) | MUXP2A0 (Bit0) | IGATE2 (Bit2) | |
| MUXP3A2 (Bit5) | MUXP3A1 (Bit4) | MUXP3A0 (Bit3) | IGATE3 (Bit3) | |
| | | | | |
| **A2** | **A1** | **A0** | **IGATE** | **IRQ** |
| 0 | 0 | 0 | 1 | 2 |
| 0 | 0 | 1 | 1 | 7 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 1 | 4 |
| 1 | 0 | 0 | 1 | 3 |

***Table 3.16a: Mux Table for setting IRQs for Port C0 and C3***

## 3.17) IRQRES0 – Interrupt Set/Reset Register (offset 16, read / write)

This register is used reset the flip-flops when a positive level shift did occurred on Port Lines PC0 and PC3 on the 1st, 2nd PPI and 3rd PPIs.

**IRQRES0 Register (read mode)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| x | RESP6 | RESP5 | RESP4 | RESP3 | RESP2 | RESP1 | RESP0 |
| x | PAY6 | PAY5 | PAY4 | PAY3 | PAY2 | PAY1 | PAY0 |
| x | C0 & C3 | PC3 | PC0 | PC3 | PC0 | PC3 | PC0 |
| x | All PPIs | PPI2 | PPI2 | PPI1 | PPI1 | PPI0 | PPI0 |

**IRQRES0 Register (write mode)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| x | RESP6 | RESP5 | RESP4 | RESP3 | RESP2 | RESP1 | RESP0 |

Bits 0-5: RESP0- RESP6 – These bits are used to reset the flip-flops if a positive edge has occurred on PC0 and PC3 on the 1st, 2nd and 3rd PPIs. If a high is detected on any of the bits then a positive transition has occurred on that line. By writing a 0 and then a 1 on any of bits 0 to 6 in this register, you can reset each individual flip-flop.

Read the ISETX Register (Offset 12) in order to determine if any of PC0 and PC3 of the PPIs has been set or if the Global Flip-Flop (PAY6) was set.

If one of the bits is set, it can result in an interrupt occuring if the IGATE0 to IGATE3 Bits (ie: Port C0 and C3 on PPI0 and PPI1) in the IGATE Register (offset 13) are enabled. You can clear the PAY0 to PAY6 bits in the ISETX Register (offset 12) by writing a 0 and then a 1 to the RESP0 to RESP6 bit respectively.

Note that RESP6 resets the global shared interrupt flip-flop. If a positive edge occurred on any of the PC0 or PC3 lines on any PPI then the PAY6 flip-flop in ISETXRD Register [offset 12] will be set and an IRQ generated (assuming the IGATE3 in the IGATE register [offset 13] is enabled). Interrupt sharing is a useful feature if limited IRQ Lines are available. You can also determine which Port C0 and C3 lines became active by reading the ISETXRD Register (offset 12).

## ✋ Warning

*Do not exceed the min/max voltage specification fed into the digital input lines. The absolute minimum voltage is –0.1V and the absolute maximum voltage is 5.2V. Exceeding the above specifications will damage the PC104-72A.*

# Chapter 4: Programming Guide

### 4.0) Introduction

This chapter describes programming the PC104-72A at its lowest level. In order to accomplish this, detailed knowledge of chapter 4 and the system hardware is required.

As an alternative to low level programming, driver software is provided with the PC104-72A. This is described in Chapter 5.

**The advantages of using the driver software are:**

a) Detailed knowledge of the PC104-72A is not required.
b) The Driver Libraries supports multiple boards. In other words, you can cascade boards in the same computer.
c) The Driver Library is callable from most high level languages.

Programmers who need to incorporate special routines into their application will need to read this chapter. Examples are application programs written in Clarion, Clipper, etc.

Once the PC104-72A has been installed into the computer and external connections are made, the board is in an operational state. The PC104-72A occupies 32 consecutive Word I/O addresses starting from the board's base address. The base address is set by a DIP Switch on the PC104-72A. Programming the PC104-72A are done by using input/output instructions in assembly or any other programming language. Reading and writing to these addresses allows data to be moved to and from the PC104-72A.

Reading and writing to these ports are 8 bits wide. Reading and Writing typically takes on the form of one of the following instructions:

| Language | Port Read | Port Write | |
|---|---|---|---|
| **'C'** | value = inp(addre); | outp( addre, value); | (8 bit Read/Write) |
| **‘Pascal’** | value := port[addre]; | port[addre] := value; | (8 bit Read/Write) |

```
'Assembly'  mov al, value       mov al, value     (8 bit Read/Write)
            mov dx, addre       mov dx, addre
            in al, dx           out dx, al
```

**where:** addre is the I/O location of the PC104-72A registers
value is the byte read or written to the register

## 4.1) Initialising the PC104-72A

The PC104-72A board should be initialised before accessing any data from the Board. The following initialization routine should be done.

a) Write a zero to the IGATE Register (offset 13)
b) Write 255 to the IRQRES0 Register (offset 16)

Note that the above registers default to zero upon powerup. However, it is important that the above should be initiated to ensure proper operation.

## 4.2) Programming the 8255 PPI

The 8255 PPI is a general purpose peripheral interface having 24 programmable I/O lines. The 24 lines are divided into three separate 8-bit ports (ie: 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 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 PPI 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 PPI must be set. This is done simply by writing on byte of configuration information to the control register (8255 PPI Base + 3). The 8255 PPI then operates in a specific mode until it is reset or new configuration information is written to the control register. The format of the register is shown below:

**DIO0CTL Register (write only) - Configuration Mode**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| 1 | MAS$_1$ | MSA$_0$ | PA | PCU | MSB$_0$ | PB | PCL |
| Mode Set Flag | Group A Mode Selection | | Port A I/O Direction | Port C Upper I/O direction | Group B Mode Selection | Port B I/O dir | Port C Lower I/O dir |

Bit 7 of the DIO0CTL Register must be set when programming the configuration of the 8255 PPIs. The mode is set for each of two groups of the 8255 PPI and the direction is set for individual ports. Note that the direction of Port C is independently programmable in two 4 line nibbles.

### 4.2.1) Mode 0: Simple I/O

This mode is used for simple input and output operatons 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 (port A and Port B) and two 4-bit nibbles (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:**

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

### 4.2.2) 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 a different functions and names. Refer to Chapter 3: Register Structure for an in depth analysis of the bit 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 20 bit simple I/O Port
Data transfer by interrupts or polled I/O.

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

GROUP A

Configuration Info
to PIACNTRL Register

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 1/0 | X | X | X |

PC7,6
1 = Input
0 = Output
X = Not applicable

GROUP B

Configuration Info
to PIACNTRL Register

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | X | X | X | X | 1 | 1 | X |

X = Not Applicable

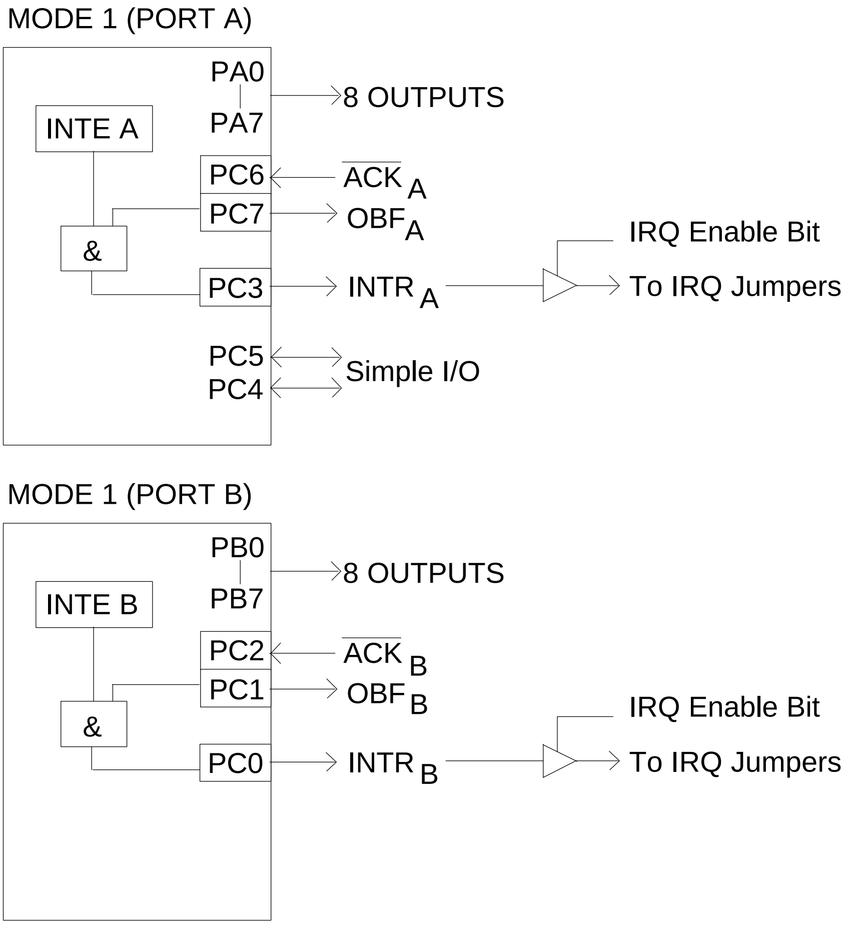


*Figure 4.2b: Mode 1 Input of an 8255*

## Mode 1 Programming

To use the 8255 PPI in mode 1 input with interrupts:

a) Write a byte to control register to configure the 8255 PPI for mode 1 and the appropriate group for data input.

b) With the bit set/reset operation, write a 1 to the Interrupt Enable Flip-Flop (INTE) of the desired port of the appropriate 8255 PPI.

c) Write a byte to the IEN0 register to enable interrupts from the appropriate 8255 PPI to the host computer.

d) The external device pulses the Strobe (/STB) input line on the digital I/O Connector low. The trailing edge of this loads the data into the input port.

e) 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.

f) When the external device pulls the /STB line high, the Interrupt Request line (INTR) goes high. This indicates tothe host computer that there is data to be read from the 8255 PPI.

g) The computer reads the data using an interrupt service routine (ISR) and by doing so, automatically resets the INTR signal.

h) The external device can now pulse the /STB low again to make the PC 36C read another byte of data and hence repeat the cycle.

Whenever a group of the 8255 PPI 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**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| C7 | C6 | $\mathrm{IBF}_A$ | $\mathrm{INTE}_A$ | $\mathrm{INTR}_A$ | $\mathrm{INTE}_B$ | $\mathrm{IBF}_B$ | $\mathrm{INTR}_B$ |

The 8255 PPI may alternatively be used in mode 1 and the data read by polled transfer.

The method is as follows:

a) Write a byte to control register to configure the 8255 PPI for mode 1 and the appropriate group for data input.

b) The program continually monitors the IBF line by reading Port C.

c) The external device pulses the Strobe (/STB) input line on the digital I/O Connector low. The trailing edge of this loads the data into the input port.

d) 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.

e) This also causes the corresponding IBF bit in port C to be set and the program can thus read the data.

f) The external device can now pulse the /STB low again to make the PC 36C 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 IEN0 register.

GROUP A

Configuration Info
to PIACNTRL Register

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1/<br>0 | X | X | X |

PC5, 4
1 = Input
0 = Output
X = Not applicable

GROUP B

Configuration Info
to PIACNTRL Register

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | X | X | X | X | 1 | 0 | X |

INTE B controlled by bit
set/reset of PC2

= Not Applicable

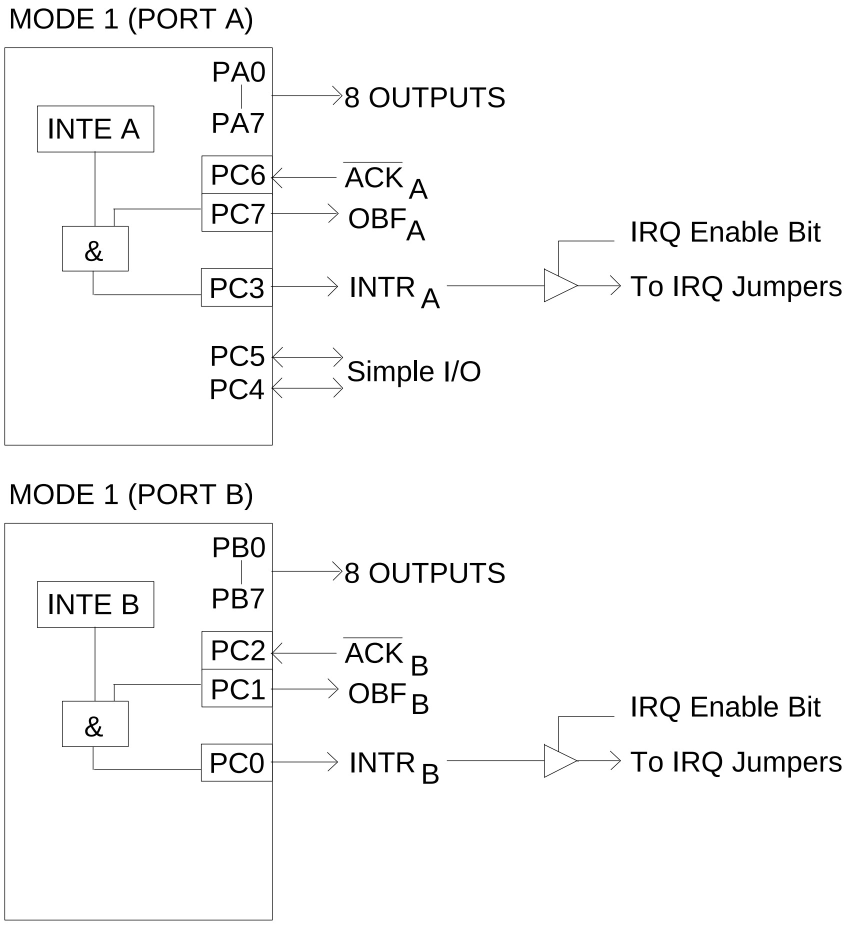


*Figure 4.3b: Mode 1 Output of an 8255*

To use the 8255 PPI in mode 1 output with interrupts:

a) Write a byte to control register to configure the 8255 PPI for mode 1 and the appropriate group for data output.

b) With the bit set/reset operation, write a 1 to the Interrupt Enable Flip-Flop (INTE) of the desired port of the appropriate 8255 PPI.

c) The 8255 PPI Interrupt Request Output (INTR) then goes high.

d) Write a byte to the IEN0 register to enable interrupts from the appropriate 8255 PPI to the host computer.

e) 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.

f) 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 PPI.

g) The external device pulses the Acknowledge (/ACK) input low and then high again to indicate that it has read the data.

h) 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 PPI 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**

| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
|---|---|---|---|---|---|---|---|
| /OBF$_A$ | INTE$_A$ | C5 | C4 | INTR$_A$ | INTE$_B$ | /OBF$_B$ | INTR$_B$ |

***The 8255 PPI may alternatively be used in mode 1 and the data written by polled transfer.***

This is done as follows:

a) Write a byte to control register to configure the 8255 PPI for mode 1 and the appropriate group for data output.

b) 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.

c) Then the program can write new data to the port.

d) 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 PPI.

e) The external device pulls the /ACK input low to read the data.

f) 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 IEN (Interrupt Enable Register).

### 4.2.3) 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 similiar to mode 1 are provided to maintain proper bus flow descipline. 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 available on each PPI
Data transfer by interrupts or polled I/O.

With the 8255 PPI of the PC 36C configured in mode 2, the board provides the function of a full 8 bit wide bidirectional data bus.

**GROUP A**

Configuration Info
to PIACNTRL Register

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | X | X | X | 0/<br>1 | 1/<br>0 | 1/<br>0 |

Group B Mode
0 = Mode 0
1 = Mode 1

Port B
0 = Output
1 = Input

Port C2-C0
0 = Output
1 = Input

X = Not Applicable

INTE 1 controlled by
bit set/reset of PC6

INTE 2 controlled by
bit set/reset of PC4

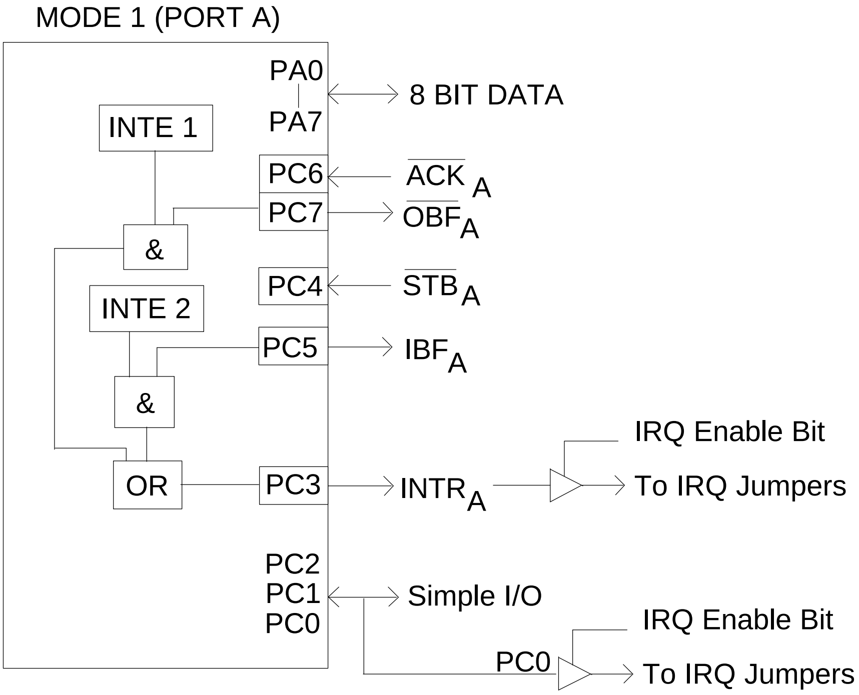


*Figure 4.2.3: Mode 2 of an 8255*

### 4.4.4) Mode 2 Programming

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

a) Write a byte to the control register to configure the 8255 PPI for mode 2 operation.

b) Select an interrupt level with the interrupt jumpers and enable 8255 interrupt(s) in the PC 36C Interrupt Enable Register IEN.

c) With the bit set/reset operation, write a 1 to the Interrupt Enable number 1 flip-flop ($INTE_1$) to enable output transfer interrupts. Write a 1 to the Interrupt Enable number 2 flip-flop ($INTE_2$) to enable input transfer interrupts. Both input and output interrupts may be enabled at the same time.

d) With both interrupt flip-flop enabled, the interrupt request line to the host computer is activated if the external device has a strobed data into the 8255 input latch or if the external device has read the PC 36C output data from the output latch.

e) The host computer detects the INTR line is active. The Interrupt Service Routine (ISR) which services this interrupt can determine whether it was an input or output interrupt by checking Port C bit 5 ($IBF_A$) of the mode 2 status information. If the IBF line is high (bit 5 is set) then it is an input interrupt, otherwise it is an output interrupt.

f) In either case, the ISR simply reads the data from or writes to the 8255 PPI and then issues an end of interrupt command (EOI) to the interrupt controller.

g) This generates the appropriate handshake signals from the 8255 PPI to the digital I/O connector.

h) The cycle continues when the next interrupt is generated.

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

**Table 4.4.4a) Port C Mode 2 Status Information**

| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
|---|---|---|---|---|---|---|---|
| $/OBF_A$ | $INTE_1$ | $IBF_A$ | $INTE_2$ | $INTR_A$ | C2 | C1 | C0 |

The 8255 PPI may alternatively be used in mode 2 with the data read and written by polled I/O transfer. This is done as follows:

a) Write a byte to the control register to configure the 8255 PPI for mode 2 operation.

b) The program continually monitors both the IBFA and /OBFA line by reading port C.

c) If port C bit corresponding to IBFA is set, this indicates that the external device has written data into the input latch. The program must therefore read the data from port A.

d) If the port C bit corresponding to /OBFA 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.

e) This process can be repeated until the required amount of data has been read and written.

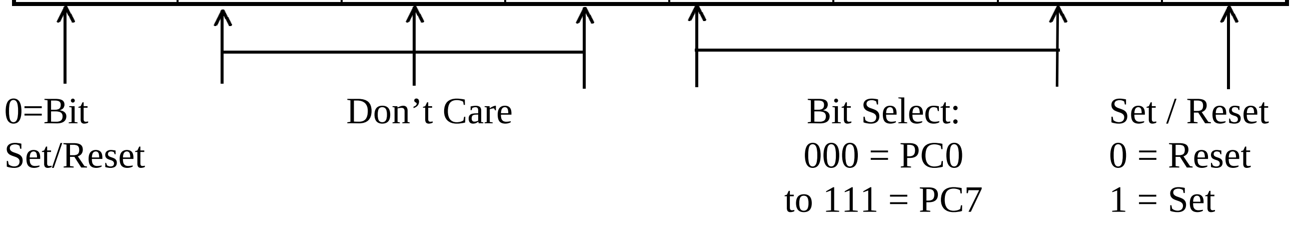
Interrupt to the host computer must not be enabled in the Interrupt Enable register IEN0 when data is transferred by polled I/O.

### 4.2.5) 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 outputs. The format of hte byte to write to the DIOCTRL register to set or reset a port C bit is repeated below.

**DIO0CTL Register (write only) - Bit Set/Reset Mode**

| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
|---|---|---|---|---|---|---|---|
| 0 | x | x | x | BS$_2$ | BS$_1$ | BS$_0$ | S/R |



0=Bit
Set/Reset

Don’t Care

Bit Select:
000 = PC0
to 111 = PC7

Set / Reset
0 = Reset
1 = Set

### 4.2.6) Mixed Mode Programming

An 8255 PPI 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 nay 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 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.

### 4.3) Configuring Port C0 and C3 on each PPI using Interrupt Mode

Port C0 and C3 on the 1st, 2nd PPI and 3rd PPIs can be configured to an Interrupt Line. However, since only 4 IRQ Lines are available, only 3 can be vectored to an individual IRQ. Any of the PPIs Port C3 and Port C0 can be vectored to one common IRQ line using PAY6 [Offset 12], RESP6 [Offset 16] and the IGATE3 [Offset 13] Bits. To configure the IRQ lines proceed as follows:

a) Set the appropriate bit for each of the Port C0 and C3 of each PPI (max 4 at any one time) in the IGATE Register.
b) Select the appropriate IRQ by writing to IMUXP0, IMUXP1 Registers.
c) Setup an ISR for each IRQ
d) When a positive edge is seen on one of the Port C0 and C3 on the 1st, 2nd PPI and 3rd PPIs, an interrupt will be generated.
e) An ISR should take the necessary action.
f) Clear the Interrupt Enable flip-flops after all the necessary action is taken. You can clear each individual flip-flips by writing to the IRQRES0 Register (Offset 16).
g) Issue an End of Interrupt (EOI) command to the Interrupt controller. Note that an EOI command must be issued to both Interrupt controllers.
h) Repeat the above cycle from d) to g).

✋ **Warning**

*Do not enable the Bits in the IGATE Register and configure both IRQ MUXes in the IMUXP0 and the IMUXP1Registers to the same Interrupt. This will lead to two open collector lines driving each other and can yield inexplicable results.*

### 4.4) Configuring Port C0/C3 for Global Sharing Interrupt Mode

Port C0 and Port C3 can be vectored to one common IRQ line using Global IRQ Mode. This means that PAY6 [Offset 12] will be set if any Low-High transition occurred on any of the PPI Port C0 or C3 lines. The IGATE3 in the IGATE Register [Offset 13] must be enabled in order to generate an IRQ.

To configure the IRQ lines for Global IRQ Sharing proceed as follows:

a) Enable IGATE3 in the IGATE Register.
b) Select the appropriate IRQ by writing to IMUXP3A0 to IMUXP3A2 bits in the IMUXP1 Register.
c) Setup an ISR for each IRQ
d) When a positive edge is seen on any of the Port C0 and C3 the 1st, 2nd PPI and 3rd PPIs, an interrupt will be generated.
e) An ISR should take the necessary action.

f) Clear the Interrupt Enable flip-flop RESP6 in the IRQRES0 Register after all the necessary action is taken. You can clear each individual flip-flips by writing to the IRQRES0 Register (Offset 16).

g) Issue an End of Interrupt (EOI) command to the Interrupt controller. Note that an EOI command must be issued to both Interrupt controllers.

h) Repeat the above cycle from d) to g).

## 4.5) Reading the PPI Port Lines

There are 16 TTL compatible Digital Inputs and 16 TTL compatible Digital Outputs on the PC104-72A. These can be read and written too using the DIO Register (offset 20).

A Pascal example of how to read and write to the digital ports is given below:

```
program p72a;



{ Copyright 1998: Shafique Allie – E-mail: shafique@eagle.co.za

  Date:       12/3/98

  Description:

  This Program tests the PC104-72A Digital I/O board by using a simple
  LoopBack Connector. Note that PPI0 and PPI1 are on the IDC50 connector
  and PPI2 is on an IDC40 Connector. Connect PPI0 Port A to PPI1 Port A
  to PPI2 Port A. Next connect PPI0 Port B to PPI1 Port B to PPI2 Port B.
  Next connect PPI0 Port C to PPI1 Port C to PPI2 Port C.

  For Example:

   PPI0 Port  Pin to PPI1 Port Pin to PPI2 Port  Pin
   ---------  ---    --------- ---    ---------  ---
   0PA0       1      1PA0      25     2PA0       1
   0PA1       2      1PA1      26     2PA1       2
   0PA2       3      1PA2      27     2PA2       3
   0PA3       4      1PA3      28     2PA3       4
   0PA4       5      1PA4      29     2PA4       5
   0PA5       6      1PA5      30     2PA5       6
   0PA6       7      1PA6      31     2PA6       7
   0PA7       8      1PA7      32     2PA7       8

   0PB0       9      1PB0      33     2PB0       11
   0PB1       10     1PB1      34     2PB1       12
   0PB2       11     1PB2      35     2PB2       13
   0PB3       12     1PB3      36     2PB3       14
   0PB4       13     1PB4      37     2PB4       15
   0PB5       14     1PB5      38     2PB5       16
   0PB6       15     1PB6      39     2PB6       17
   0PB7       16     1PB7      40     2PB7       18

   0PC0       17     1PC0      41     2PC0       21
   0PC1       18     1PC1      42     2PC1       22
   0PC2       19     1PC2      43     2PC2       23
   0PC3       20     1PC3      44     2PC3       24
   0PC4       21     1PC4      45     2PC4       25
   0PC5       22     1PC5      46     2PC5       26
   0PC6       23     1PC6      47     2PC6       27
   0PC7       24     1PC7      48     2PC7       28

PPI0 is configured as an Output while PPI1 and PPI2 are configured as
Inputs. The Loopback essentially write a value from 1 to 255 to Ports
A, B, C of PPI0 and Reads it back on Ports A, B, C on PPI1 and PPI2
respectively. If hte values are verified correct then a Pass message
is displayed otherwise a fail is displayed. This test is repeated until
the key is pressed.

}

uses crt;

CONST ba = $300;
      count = 1;

var    Pattern, i, j, x : integer;
       Ch : Char;
       dummy, dum1, k, l, m, pa : integer;
       t_count, y , z : longint;
       id, id1 : integer;
       test14     : longint;
       error14, error : longint;


Procedure writebinary(pat:integer);

   Var Mask, I : integer;

   Begin
      Mask := $80;
      For i := 1 to 8 do
         begin
            if  (mask and pat) = 0 then write('0') else write('1');
            Mask := Mask shr 1;
         end;
   End;

Procedure delay_1;   { Software Delay }

begin
  for y := 1 to 100 do
   begin
     for z := 1 to 100 do
      begin
      end
   end;
end;  {software delay}


Procedure testdisplay;

begin
  gotoxy(7,5);
  write('1: External Digital I/O Test: ');
end;

begin                  { main program }
  t_count := 1;        { set counter to 1 }
  pa := 0;          { Set Error Count to 0 }
  clrscr;
```

```
repeat
  id := 0;
  id1 := 1;

  x := 37;
  gotoxy(7,1);
  writeln('---------------------------------------');
  gotoxy(10,2);
  Writeln('PC104-72 I/O Loopback Test Program');
  gotoxy(7,3);
  writeln('---------------------------------------');

  testdisplay;

  gotoxy(9,12);
  write('PC104-72 Digital I/O Test: ');
  gotoxy(49,12);
  write('Test Count: ', t_count);

  test14 := 0;

 port[ba+3] := $80;    { 1st PPI Output }
 port[ba+7] := $9b;    { 2nd PPI Input }
 port[ba+11] := $9b;   { 3rd PPI Input }

  id1 := 0;

 for i := 1 to count do              { external digital I/O test }
   begin
     id:= 0;
     k := 0;
     error := 0;
     error14 := 0;
     test14 := 0;

   for l := 0 to 2 do
     begin
       for k := 0 to 255 do     { 8 bit on each PPI Port }
        begin
         port[ba+l] := k;       { Set PPI0 Port Bits }
         dummy := port[ba+l+4]; { Read PPI1 Port Bits }
         dum1 := port[ba+l+8];  { Read PPI2 Port Bits }
         gotoxy(11,7);
         write('PPI Port No:   ', l:4);
         gotoxy(35,7);
         writebinary(l);
         gotoxy(11,8);
         write('PPI0 (Output): ', k:4);
         gotoxy(35,8);
         writebinary(k);
         gotoxy(11,9);
         write('PPI1 (Input):  ', dummy:4);
         gotoxy(35,9);
         writebinary(dummy);
         gotoxy(11,10);
         write('PPI2 (Input):  ', dum1:4 );
         gotoxy(35,10);
         writebinary(dum1);

         delay_1;  { Delay before next sequence }

         if (dummy <> k) or (dum1 <> k) then  { Check for any errors }
```

```
            begin
              error := error + 1;    { If so increment the error count }
              error14 := error14 + 1;
            end;
        end;
      end;
        if error14 <> 0 then
          begin
            test14 := test14 + 1;
            id1 := 0;
            pa := 1;
            if test14 = 1 then error14 := dummy;
          end;
     end;

      if (pa = 0) and (id1 = 0) then  { Check if no errors occurred }

      begin
          gotoxy(x,5);
          write('Pass');
        end
      else begin          { Otherwise an error has occured }
             gotoxy(x,5);
             write('Fail ');
             id := 1;
           end;

 id:= 0;

 if pa = 1 then   { Check if overall error has occured }
   begin
     gotoxy(36,12);
     write('Fails ');
   end;

 if pa = 0 then   { Check if no errors has occured }
   begin
     gotoxy(36,12);
     write('Pass ');
   end;

   t_count := t_count + 1; { Increment Counter }


 until (keypressed) or (pa = 1) or (t_count > 5) ;

 if id = 1 then
   begin
     gotoxy(35,15);
     write('Fails ');
   end;

   gotoxy(20,17);
   write('Press Any Key to continue..');
   ch := readkey;

end.
```

# Chapter 5: Driver Software

Full driver software is supplied with the PC104-72A package. Full details are explained in the EDR Software developers kit User Manual. A summary of the drivers is explained below.

For the latest update Driver software and technical information, see our WebSite:
http://www.eagle.co.za

Both DOS and Windows Languages are supported: They are:

## DOS Languages:

Borland C/C++ Version 3.1 or 4.0
Microsoft C/C++ Version 6.0 or 7.0
Borland Pascal / Turbo Pascal Version 6.0 or 7.0
Microsoft QuickBasic Version 4.5

### Windows Languages:

Delphi V1.00 / V2.00 / V3.00
Borland C/C++ 3.1 or 4.0
Microsoft C/C++ 6.0 or 7.0
Borland Pascal / Turbo Pascal Version 6.0 or 7.0
Visual Basic V1.00 thru V5.00

### 5.1) Board Handles

All EDR functions used above require a board handle as the first parameter. The board handle defines which board is affected by the function call. Using this method has several advantages, For example, there is no need for a 'select board' function; working with parallel boards is much easier; different applications using the EDR at the same time will not conflict with each other.

Board handles are integers obtained by calling EDR_AllocBoardHandle (see 7.2 of EDR Developers Toolkit). Once allocated a board handle must be initialised to the PC104-72A before it can be accessed. This is achieved by calling EDR_InitBoard or EDR_InitBoardType (see 7.5 of EDR manual) with the base address or EDR_loadConfiguration (see section 7.8 of EDR manual).

EDR_InitBoard will attempt to detect the PC104-72A at the base address specified.

## 5.2) Interrupt functions

Port C0 and C3 lines on the 1st, 2nd PPI and 3rd PPIs are connected to a latch and fed into the IRQ line when Interrupt Mode is used. An ISR can be installed using the EDR driver functions to service these lines when pulses toggles Port C0 and C3 on the 1st, 2nd PPI and 3rd PPIs. Note that all the lines are latched. If any of Port C0 and C3 on the 1st, 2nd PPI and 3rd PPIs lines are vectored to the PC’s Interrupt System, the latches should be cleared in the ISR.

Note that Port Line C3 of PPI1 and Port Line C0 and C3 of PPI2 share an Interrupt Line.

These functions are callable from virtually any programming language. See the EDR Software Developers Kit User Manual.

## 5.3) Quick Function Reference

The PC104-72A 72 Channel Enhanced PC104 Digital I/O Board which utilises the following functions calls contained in the EDR driver developers toolkit. They are:

| Function Name | Description |
|---|---|
| EDR_DIOConfigurePort | Configures a digital port for a particular mode and direction. EDR only directly supports mode 0 (EDR_DIO_SIMPLE) with the driver functions. Other modes can be be used but you will need to write your own support code. |
| EDR_DIOLineInput | Gets the status of a single line (bit) in a digital input port. If the port is an output port then the last value written to the line (bit) is returned. |
| EDR_DIOLineOutput | Changes a single line (bit) in a digital output port. |
| EDR_DIOPortOutput | Writes a byte of data to the PC104-72A I/O port |
| EDR_DIOPortInput | Reads a byte of data from the PC104-72A I/O Port. If the port is an output port then the last value written to the port is returned. |
| EDR_HasDOReadback | Indicates if a particular board type's digital output ports can be read back or not. If the board supports readback from DO ports then doing EDR_DIOPortInput or EDR_DIOLineInput on an output port will read the last value written to the port from a board register. If the readback is not supported the EDR returns its software copy of the last value written using EDR_DIOPortOutput or EDR_DIOLineOutput. |
| EDR_EnableInterrupt | Enables or disables the specified interrupt on the PC104-72A. Note that this just programs the board registers (ie: writing to IEN0 and IEN1 Registers) so that it will/will not generate the interrupt. If the interrupt is masked then interrupts from the board will be blocked.<br>Disabling interrupts or enabling interrupts when they are already enabled may generate extra interrupts. Make sure that your ISR are prepared to handle these rogue interrupts. |
| EDR_InstallISR | Installs an ISR for the specified hardware interrupt request |

| **Function Name** | **Description** |
|---|---|
| EDR_UninstallISR | Removes an interrupt service routine that was installed with EDR_InstallISR |
| EDR_InstallBoardISR | Installs an ISR for a particular type of interrupt installed on the PC104-72A. |
| EDR_UninstallBoardISR | Removes an interrupt service routine that was installed with EDR_InstallBoardISR |
| EDR_MaskIRQ | Masks or unmasks a particular IRQ level |
| EDR_MaskBoardIRQ | Masks or unmasks a particular board interrupt |
| EDR_ResetInterrupt | Resets an interrupt latch on the PC104-72A and sends an EOI command to one of the PCs interrupt controllers on completion of the interrupt |
| EDR_AllocBoardHandle | Allocates a new board handle to the PC104-72A. If no board handles are available then a 0 is returned. This is particularly useful is multiple PC104-72A are present in the same.computer. |
| EDR_FreeBoardHandle | Releases a board handle allocated to the PC104-72A making it available to any other PC card |
| EDR_InitBoardType | Initialises a board and allocates a board handle to it |
| EDR_ConfigDialog | Diplays a dialog box that allows the user to manually configure the driver for the Board in the computer |
| EDR_SaveConfiguration | Function writes configuration information to a file for later loading with EDR_LoadConfiguration |
| EDR_LoadConfiguration | Loads details of the Cards configuration from the file created by EDR_SaveConfiguration |
| EDR_RestoreDefaults | Restores factory default configuration for a baord attached to a handle |
| EDR_IsBaseAddressInUse | Checks if any board initialised with EDR is using the specified I/O address |
| EDR_DetectBoard | Tries to determine the type of board present at a specified I/O address. |
| EDR_SetBoardType | Changes the board type attached to a board handle |
| EDR_SetIRQLevel | Set the IRQ level EDR will use for the interrupt ID specified. |
| EDR_NumDIOPorts | Gets the number of DIO ports on the board. |
| EDR_ValidDIOPortConfig | Checks the configuration of a digital port. |
| EDR_ValidInterruptID | Checks to see if the board supports the ID specified |
| EDR_ValidIRQLevel | Checks an IRQ level |

# Chapter 6: Testing the PC104-72A

Before attempting to interface the PC104-72A with your application, it is essential that you test the board first. This is done using the following procedure:

## 6.1) Testing the PC104-72A Board

Install the PC104-72A using the procedure described in the Chapter 2: Installation. Proceed as follows:

a) Make a test loom by connecting PPI0 Port A to PPI1 Port A to PPI2 Port A.
b) Next connect PPI0 Port B to PPI1 Port B to PPI2 Port B.
c) Next connect PPI0 Port C to PPI1 Port C to PPI2 Port C.

The actual Pinout table is as follows:

| PPI0 Port | Pin to | PPI1 Port | Pin to | PPI2 Port | Pin |
|---|---|---|---|---|---|
| 0PA0 | 1 | 1PA0 | 25 | 2PA0 | 1 |
| 0PA1 | 2 | 1PA1 | 26 | 2PA1 | 2 |
| 0PA2 | 3 | 1PA2 | 27 | 2PA2 | 3 |
| 0PA3 | 4 | 1PA3 | 28 | 2PA3 | 4 |
| 0PA4 | 5 | 1PA4 | 29 | 2PA4 | 5 |
| 0PA5 | 6 | 1PA5 | 30 | 2PA5 | 6 |
| 0PA6 | 7 | 1PA6 | 31 | 2PA6 | 7 |
| 0PA7 | 8 | 1PA7 | 32 | 2PA7 | 8 |
| 0PB0 | 9 | 1PB0 | 33 | 2PB0 | 11 |
| 0PB1 | 10 | 1PB1 | 34 | 2PB1 | 12 |
| 0PB2 | 11 | 1PB2 | 35 | 2PB2 | 13 |
| 0PB3 | 12 | 1PB3 | 36 | 2PB3 | 14 |
| 0PB4 | 13 | 1PB4 | 37 | 2PB4 | 15 |
| 0PB5 | 14 | 1PB5 | 38 | 2PB5 | 16 |
| 0PB6 | 15 | 1PB6 | 39 | 2PB6 | 17 |
| 0PB7 | 16 | 1PB7 | 40 | 2PB7 | 18 |
| 0PC0 | 17 | 1PC0 | 41 | 2PC0 | 21 |
| 0PC1 | 18 | 1PC1 | 42 | 2PC1 | 22 |
| 0PC2 | 19 | 1PC2 | 43 | 2PC2 | 23 |
| 0PC3 | 20 | 1PC3 | 44 | 2PC3 | 24 |
| 0PC4 | 21 | 1PC4 | 45 | 2PC4 | 25 |
| 0PC5 | 22 | 1PC5 | 46 | 2PC5 | 26 |
| 0PC6 | 23 | 1PC6 | 47 | 2PC6 | 27 |
| 0PC7 | 24 | 1PC7 | 48 | 2PC7 | 28 |

Note that PPI0 and PPI1 are located on the IDC50 Male Header and PPI2 is located on the IDC40 Male Header.

b) Switch the Computer on

c) On the DOS prompt, go to the C:\EDR\TPAS\DEMOS\ sub-directory

d) Run the 72TEST.EXE Test program.

If an error message 'Board not found' appear on your screen then the PC104-72A was not installed at that address. Try a different base address as specified in Apendix A (eg: 700h) and re-run the test S/W. If the problem persists then try increasing the wait states on the PC104-72A Board and re-run test software. The board should work.

If the PC104-72A is found, a message will appear on the screen: 'PC104-72A Board found'. The program comprehensively tests the register structure and each port line on each PPI.

## 6.2) Connecting Normally Open devices to the Digital Input Lines

When connecting switches, etc, to the Digital Inputs of the PC104-72A (assuming the PPIs are configured as inputs), it is important to ensure that the inputs are set to a defined state at all times. Figure 6.2 gives an example.

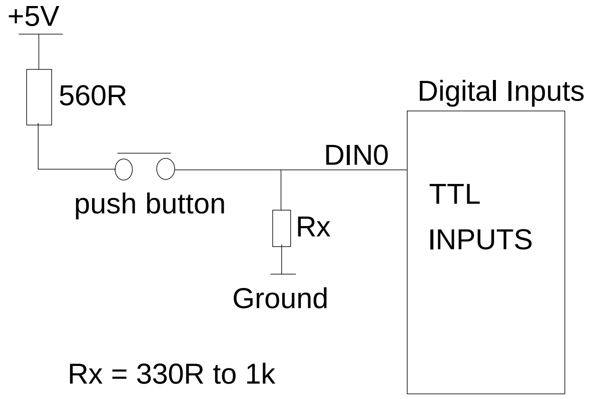


Figure 6.2: Connecting Normally Open devices to the Digital Inputs

Figure 6.2 gives an example of how to connect a push button to one of the Digital Input lines on Connector J6. We assume that when the Push button is closed, the Port line will go high (logic 1 = +5V). When the push button is not closed, the port line should be low. However, many Users do not connect the ‘pull-down’ resistor to the port line. If this is not done, the port line will be in a floating state and can hence yield either a high or a low.

Thus you must ensure that the port line is always connected which is done via a resistor network.

# Chapter 7: Troubleshooting

***Problem:*** 'Board not found' message appears on the screen when running test software.

***Solution:*** Another I/O card might be using the same base address location as the PC104-72A. Try a different base address other than the manufacturer’s default base address (eg: 700h) and re-run test software 62CTEST.EXE found in sub directory C:\EDR\EXAMPLES\TPAS\. If the problem persists then try increasing the wait states on the PC104-72A. If the PC104-72A still does not work even after the maximum number of wait states was chosen then try a different computer with a different motherboard. If the test still fails then the PC104-72A is faulty. Return the board to your distributor for repairs.

***Problem:*** Data written to Port A, B or C (configured as Mode 0 outputs) does not set the output ports correctly.

***Solution:*** Check the ports using the test procedure described in Chapter 6 using the program called 72CTEST.EXE found in EDR\TPAS\DEMOS subdirectory. Monitor the port lines of each 8255 PPI and check if the problem persists. If so, then the PPI might be faulty, replace the PPI in question with another NEC D8255AC-2 or an Intel I8255-2. You could also send the PC104-72A back to your distributor for repairs.

***Problem:*** I connected a push button to one of the input port lines. When it is closed it is set to +5V (yield a logical 1) and the program reads a logical 1. However, when it is open, random numbers are read by the program on the digital input lines.

***Solution:*** When the push button is not connected, the port line is in a floating (not connected). You must connect a ‘pulldown’ resistor (experiment from 330R to 1k) to the port line in order to ensure that it is in a defined state. Also note that if a port is configured as an input then all unused lines MUST be grounded. See Section 6.2 for more details.

***Problem:*** Interrupts does not occur when any of Port C0 and C3 on the 1st, 2nd PPI and 3rd PPIs receives a pulse from 0V to above the threshold of 2.0V. In other words, a square pulse Posituve edged 0 to 5V does not yield an interrupt.

***Solution:*** It is possible that the PPI Port C0 or C3 Bit was not configured in Interrupt Mode. First read back the IGATE Register [Offset 13] and check if the appropriate Interrupt Enable Bit was set. For example, if an IRQ does not occur on Port Bit C0, check if bit 0 on the IGATE Register was enabled. If not, set it to 1. Next, configure the IRQ to any of 4 using the IMUXP0 Register (Bits 0 to 2). Re-trigger Port C Bit 3. An IRQ should occur.

Don't forget to reset the Interrupt Flip Flops after an IRQ has occurred on Port C0 and C3 on the 1st, 2nd PPI and 3rd PPIs. A Reset should be done by writing a 0 and then a 1 to the IRQRES0 Register (Offset 16) to the appropriate Port C0 and C3 on the 1st, 2nd PPI and 3rd PPIs Interrupt Enable flip-flop.

✋**Warning**

Any parts replaced on the PC104-72A must be done by a qualified or trained technician. If you (the user) is not a trained technician then rather return the board to your distributor for repairs explaining in detail what the problem is.

If you cannot solve the problem then simply call your distributor for immediate help.

# Chapter 8: Repair Service

The PC104-72A is guaranteed for a period of 1 year. If the board is faulty within this period, we will gladly repair it free of charge provided that the maximum specification was not exceeded. If any burn't tracks are seen on the PC104-72A Board, warranty will be void. A repair charge will be levied in the user requires the board to be repaired.

Before sending the board to your distributor for repairs, ensure that you go through Chapter 7: TroubleShooting Hints thoroughly. If, after you have gone through this Chapter, the board still does not work, return it for repairs stating in detail what the problem is.

Our repair service centre will be available to repair our products even after the 1 year warranty. A small service fee will be levied which usually covers the cost of the components that are faulty.

# Specifications

## Computer Host Interface

| | |
|---|---|
| *Base Address:* | Switchable from 0 to 17FFh on 32 Word boundaries |
| *Bus Type:* | PC104 Rev 2.3 |
| *I/O Wait States:* | 0, 1, 2, 4, or 8 Jumper selectable. |
| *Number of Registers:* | Thirty Two 8-bit Registers |
| *Word size:* | 8 bits |
| *Word transfer Rate:* | 10Mhz max (depends on computer and program) |
| *Interrupts:* | Software selectable from IRQ2, 3, 4, 5, 7 |

## Digital Inputs/Outputs

| | | |
|---|---|---|
| *Number of lines:* | 48 I/O lines in nine 8-bit digital ports | |
| *Compatibilty:* | TTL | |
| *Input Voltage:* | Logic Low:<br>Logic High: | 0.5V to 0.8V<br>2V to 5V |
| *Output Voltage:* | Logic 0:<br>Logic 1: | 0.45V max<br>2.4V min |
| *Output Current:* | Low State:<br>High State: | 1.7mA max<br>-200µA min |
| *Max Darlington drive current* | -4.0mA from ports types B and C only | |
| *Maximum Power per 8255* | 1W max from all three ports per PPI | |

## Absolute Maximum TTL Inputs

| | |
|---|---|
| *Absolute Max Input voltage:* | 6.5V* |
| *Absolute Min Input voltage:* | -0.5V* |

### ☞ Warning

Stresses greater than listed above may cause permanent damage to the PC104-72A. This stress rating only applies to the PC104-72A at these or any other condition above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect device reliability. We do not recommend exposure to absolute maximum ratings.

## I/O Connector

|  |  |
|---|---|
| *PPI0 and PPI1 Input Connector:* | 50-way IDC Male Header |
| *PPI2 Digital I/O Connector:* | 40 way IDC Male header |
| *Max User +5V current:* | 200mA (at I/O connector) |

## Enviromental

|  |  |
|---|---|
| *Operating temperature:* | 0°C to 55°C |
| *Storage temperature:* | -55°C to 150°C |
| *Relative humidity:* | 5% to 90% non-condensing |
| *PC Board size:* | 181mm x 115mm (including edge connector) |

## Power Supply Requirements

|  |  |
|---|---|
| *+5V Supply:* : | 350mA typ (excludes +5V power for User Interfacing) |

# Appendix A (Base Address Settings)

## Base Address Switch Settings

| Base Address | SW1 | SW2 | SW3 | SW4 | SW5 | SW6 | SW7 | SW8 | Base Address |
|---|---|---|---|---|---|---|---|---|---|
| 00 | ON | ON | ON | ON | ON | ON | ON | ON | 00 |
| 20 | ON | ON | ON | ON | ON | ON | ON | OFF | 20 |
| 40 | ON | ON | ON | ON | ON | ON | OFF | ON | 40 |
| 60 | ON | ON | ON | ON | ON | ON | OFF | OFF | 60 |
| 80 | ON | ON | ON | ON | ON | OFF | ON | ON | 80 |
| A0 | ON | ON | ON | ON | ON | OFF | ON | OFF | A0 |
| C0 | ON | ON | ON | ON | ON | OFF | OFF | ON | C0 |
| E0 | ON | ON | ON | ON | ON | OFF | OFF | OFF | E0 |
| 100 | ON | ON | ON | ON | OFF | ON | ON | ON | 100 |
| 120 | ON | ON | ON | ON | OFF | ON | ON | OFF | 120 |
| 140 | ON | ON | ON | ON | OFF | ON | OFF | ON | 140 |
| 160 | ON | ON | ON | ON | OFF | ON | OFF | OFF | 160 |
| 180 | ON | ON | ON | ON | OFF | OFF | ON | ON | 180 |
| 1A0 | ON | ON | ON | ON | OFF | OFF | ON | OFF | 1A0 |
| 1C0 | ON | ON | ON | ON | OFF | OFF | OFF | ON | 1C0 |
| 1E0 | ON | ON | ON | ON | OFF | OFF | OFF | OFF | 1E0 |
| 200 | ON | ON | ON | OFF | ON | ON | ON | ON | 200 |
| 220 | ON | ON | ON | OFF | ON | ON | ON | OFF | 220 |
| 240 | ON | ON | ON | OFF | ON | ON | OFF | ON | 240 |
| 260 | ON | ON | ON | OFF | ON | ON | OFF | OFF | 260 |
| 280 | ON | ON | ON | OFF | ON | OFF | ON | ON | 280 |
| 2A0 | ON | ON | ON | OFF | ON | OFF | ON | OFF | 2A0 |
| 2C0 | ON | ON | ON | OFF | ON | OFF | OFF | ON | 2C0 |
| 2E0 | ON | ON | ON | OFF | ON | OFF | OFF | OFF | 2E0 |
| 300 | ON | ON | ON | OFF | OFF | ON | ON | ON | 300 |
| 320 | ON | ON | ON | OFF | OFF | ON | ON | OFF | 320 |
| 340 | ON | ON | ON | OFF | OFF | ON | OFF | ON | 340 |
| 360 | ON | ON | ON | OFF | OFF | ON | OFF | OFF | 360 |
| 380 | ON | ON | ON | OFF | OFF | OFF | ON | ON | 380 |
| 3A0 | ON | ON | ON | OFF | OFF | OFF | ON | OFF | 3A0 |
| 3C0 | ON | ON | ON | OFF | OFF | OFF | OFF | ON | 3C0 |
| 3E0 | ON | ON | ON | OFF | OFF | OFF | OFF | OFF | 3E0 |

| Base Address | SW1 | SW2 | SW3 | SW4 | SW5 | SW6 | SW7 | SW8 | Base Address |
|---|---|---|---|---|---|---|---|---|---|
| 400 | ON | ON | OFF | ON | ON | ON | ON | ON | 400 |
| 420 | ON | ON | OFF | ON | ON | ON | ON | OFF | 420 |
| 440 | ON | ON | OFF | ON | ON | ON | OFF | ON | 440 |
| 460 | ON | ON | OFF | ON | ON | ON | OFF | OFF | 460 |
| 480 | ON | ON | OFF | ON | ON | OFF | ON | ON | 480 |
| 4A0 | ON | ON | OFF | ON | ON | OFF | ON | OFF | 4A0 |
| 4C0 | ON | ON | OFF | ON | ON | OFF | OFF | ON | 4C0 |
| 4E0 | ON | ON | OFF | ON | ON | OFF | OFF | OFF | 4E0 |
| 500 | ON | ON | OFF | ON | OFF | ON | ON | ON | 500 |
| 520 | ON | ON | OFF | ON | OFF | ON | ON | OFF | 520 |
| 540 | ON | ON | OFF | ON | OFF | ON | OFF | ON | 540 |
| 560 | ON | ON | OFF | ON | OFF | ON | OFF | OFF | 560 |
| 580 | ON | ON | OFF | ON | OFF | OFF | ON | ON | 580 |
| 5A0 | ON | ON | OFF | ON | OFF | OFF | ON | OFF | 5A0 |
| 5C0 | ON | ON | OFF | ON | OFF | OFF | OFF | ON | 5C0 |
| 5E0 | ON | ON | OFF | ON | OFF | OFF | OFF | OFF | 5E0 |
| 600 | ON | ON | OFF | OFF | ON | ON | ON | ON | 600 |
| 620 | ON | ON | OFF | OFF | ON | ON | ON | OFF | 620 |
| 640 | ON | ON | OFF | OFF | ON | ON | OFF | ON | 640 |
| 660 | ON | ON | OFF | OFF | ON | ON | OFF | OFF | 660 |
| 680 | ON | ON | OFF | OFF | ON | OFF | ON | ON | 680 |
| 6A0 | ON | ON | OFF | OFF | ON | OFF | ON | OFF | 6A0 |
| 6C0 | ON | ON | OFF | OFF | ON | OFF | OFF | ON | 6C0 |
| 6E0 | ON | ON | OFF | OFF | ON | OFF | OFF | OFF | 6E0 |
| 700 | ON | ON | OFF | OFF | OFF | ON | ON | ON | 700 |
| 720 | ON | ON | OFF | OFF | OFF | ON | ON | OFF | 720 |
| 740 | ON | ON | OFF | OFF | OFF | ON | OFF | ON | 740 |
| 760 | ON | ON | OFF | OFF | OFF | ON | OFF | OFF | 760 |
| 780 | ON | ON | OFF | OFF | OFF | OFF | ON | ON | 780 |
| 7A0 | ON | ON | OFF | OFF | OFF | OFF | ON | OFF | 7A0 |
| 7C0 | ON | ON | OFF | OFF | OFF | OFF | OFF | ON | 7C0 |
| 7E0 | ON | ON | OFF | OFF | OFF | OFF | OFF | OFF | 7E0 |

| Base Address | SW1 | SW2 | SW3 | SW4 | SW5 | SW6 | SW7 | SW8 | Base Address |
|---|---|---|---|---|---|---|---|---|---|
| 800 | ON | OFF | ON | ON | ON | ON | ON | ON | 800 |
| 820 | ON | OFF | ON | ON | ON | ON | ON | OFF | 820 |
| 840 | ON | OFF | ON | ON | ON | ON | OFF | ON | 840 |
| 860 | ON | OFF | ON | ON | ON | ON | OFF | OFF | 860 |
| 880 | ON | OFF | ON | ON | ON | OFF | ON | ON | 880 |
| 8A0 | ON | OFF | ON | ON | ON | OFF | ON | OFF | 8A0 |
| 8C0 | ON | OFF | ON | ON | ON | OFF | OFF | ON | 8C0 |
| 8E0 | ON | OFF | ON | ON | ON | OFF | OFF | OFF | 8E0 |
| 900 | ON | OFF | ON | ON | OFF | ON | ON | ON | 900 |
| 920 | ON | OFF | ON | ON | OFF | ON | ON | OFF | 920 |
| 940 | ON | OFF | ON | ON | OFF | ON | OFF | ON | 940 |
| 960 | ON | OFF | ON | ON | OFF | ON | OFF | OFF | 960 |
| 980 | ON | OFF | ON | ON | OFF | OFF | ON | ON | 980 |
| 9A0 | ON | OFF | ON | ON | OFF | OFF | ON | OFF | 9A0 |
| 9C0 | ON | OFF | ON | ON | OFF | OFF | OFF | ON | 9C0 |
| 9E0 | ON | OFF | ON | ON | OFF | OFF | OFF | OFF | 9E0 |
| A00 | ON | OFF | ON | OFF | ON | ON | ON | ON | A00 |
| A20 | ON | OFF | ON | OFF | ON | ON | ON | OFF | A20 |
| A40 | ON | OFF | ON | OFF | ON | ON | OFF | ON | A40 |
| A60 | ON | OFF | ON | OFF | ON | ON | OFF | OFF | A60 |
| A80 | ON | OFF | ON | OFF | ON | OFF | ON | ON | A80 |
| AA0 | ON | OFF | ON | OFF | ON | OFF | ON | OFF | AA0 |
| AC0 | ON | OFF | ON | OFF | ON | OFF | OFF | ON | AC0 |
| AE0 | ON | OFF | ON | OFF | ON | OFF | OFF | OFF | AE0 |
| B00 | ON | OFF | ON | OFF | OFF | ON | ON | ON | B00 |
| B20 | ON | OFF | ON | OFF | OFF | ON | ON | OFF | B20 |
| B40 | ON | OFF | ON | OFF | OFF | ON | OFF | ON | B40 |
| B60 | ON | OFF | ON | OFF | OFF | ON | OFF | OFF | B60 |
| B80 | ON | OFF | ON | OFF | OFF | OFF | ON | ON | B80 |
| BA0 | ON | OFF | ON | OFF | OFF | OFF | ON | OFF | BA0 |
| BC0 | ON | OFF | ON | OFF | OFF | OFF | OFF | ON | BC0 |
| BE0 | ON | OFF | ON | OFF | OFF | OFF | OFF | OFF | BE0 |

| Base Address | SW1 | SW2 | SW3 | SW4 | SW5 | SW6 | SW7 | SW8 | Base Address |
|---|---|---|---|---|---|---|---|---|---|
| C00 | ON | OFF | OFF | ON | ON | ON | ON | ON | C00 |
| C20 | ON | OFF | OFF | ON | ON | ON | ON | OFF | C20 |
| C40 | ON | OFF | OFF | ON | ON | ON | OFF | ON | C40 |
| C60 | ON | OFF | OFF | ON | ON | ON | OFF | OFF | C60 |
| C80 | ON | OFF | OFF | ON | ON | OFF | ON | ON | C80 |
| CA0 | ON | OFF | OFF | ON | ON | OFF | ON | OFF | CA0 |
| CC0 | ON | OFF | OFF | ON | ON | OFF | OFF | ON | CC0 |
| CE0 | ON | OFF | OFF | ON | ON | OFF | OFF | OFF | CE0 |
| D00 | ON | OFF | OFF | ON | OFF | ON | ON | ON | D00 |
| D20 | ON | OFF | OFF | ON | OFF | ON | ON | OFF | D20 |
| D40 | ON | OFF | OFF | ON | OFF | ON | OFF | ON | D40 |
| D60 | ON | OFF | OFF | ON | OFF | ON | OFF | OFF | D60 |
| D80 | ON | OFF | OFF | ON | OFF | OFF | ON | ON | D80 |
| DA0 | ON | OFF | OFF | ON | OFF | OFF | ON | OFF | DA0 |
| DC0 | ON | OFF | OFF | ON | OFF | OFF | OFF | ON | DC0 |
| DE0 | ON | OFF | OFF | ON | OFF | OFF | OFF | OFF | DE0 |
| E00 | ON | OFF | OFF | OFF | ON | ON | ON | ON | E00 |
| E20 | ON | OFF | OFF | OFF | ON | ON | ON | OFF | E20 |
| E40 | ON | OFF | OFF | OFF | ON | ON | OFF | ON | E40 |
| E60 | ON | OFF | OFF | OFF | ON | ON | OFF | OFF | E60 |
| E80 | ON | OFF | OFF | OFF | ON | OFF | ON | ON | E80 |
| EA0 | ON | OFF | OFF | OFF | ON | OFF | ON | OFF | EA0 |
| EC0 | ON | OFF | OFF | OFF | ON | OFF | OFF | ON | EC0 |
| EE0 | ON | OFF | OFF | OFF | ON | OFF | OFF | OFF | EE0 |
| F00 | ON | OFF | OFF | OFF | OFF | ON | ON | ON | F00 |
| F20 | ON | OFF | OFF | OFF | OFF | ON | ON | OFF | F20 |
| F40 | ON | OFF | OFF | OFF | OFF | ON | OFF | ON | F40 |
| F60 | ON | OFF | OFF | OFF | OFF | ON | OFF | OFF | F60 |
| F80 | ON | OFF | OFF | OFF | OFF | OFF | ON | ON | F80 |
| FA0 | ON | OFF | OFF | OFF | OFF | OFF | ON | OFF | FA0 |
| FC0 | ON | OFF | OFF | OFF | OFF | OFF | OFF | ON | FC0 |
| FE0 | ON | OFF | OFF | OFF | OFF | OFF | OFF | OFF | FE0 |

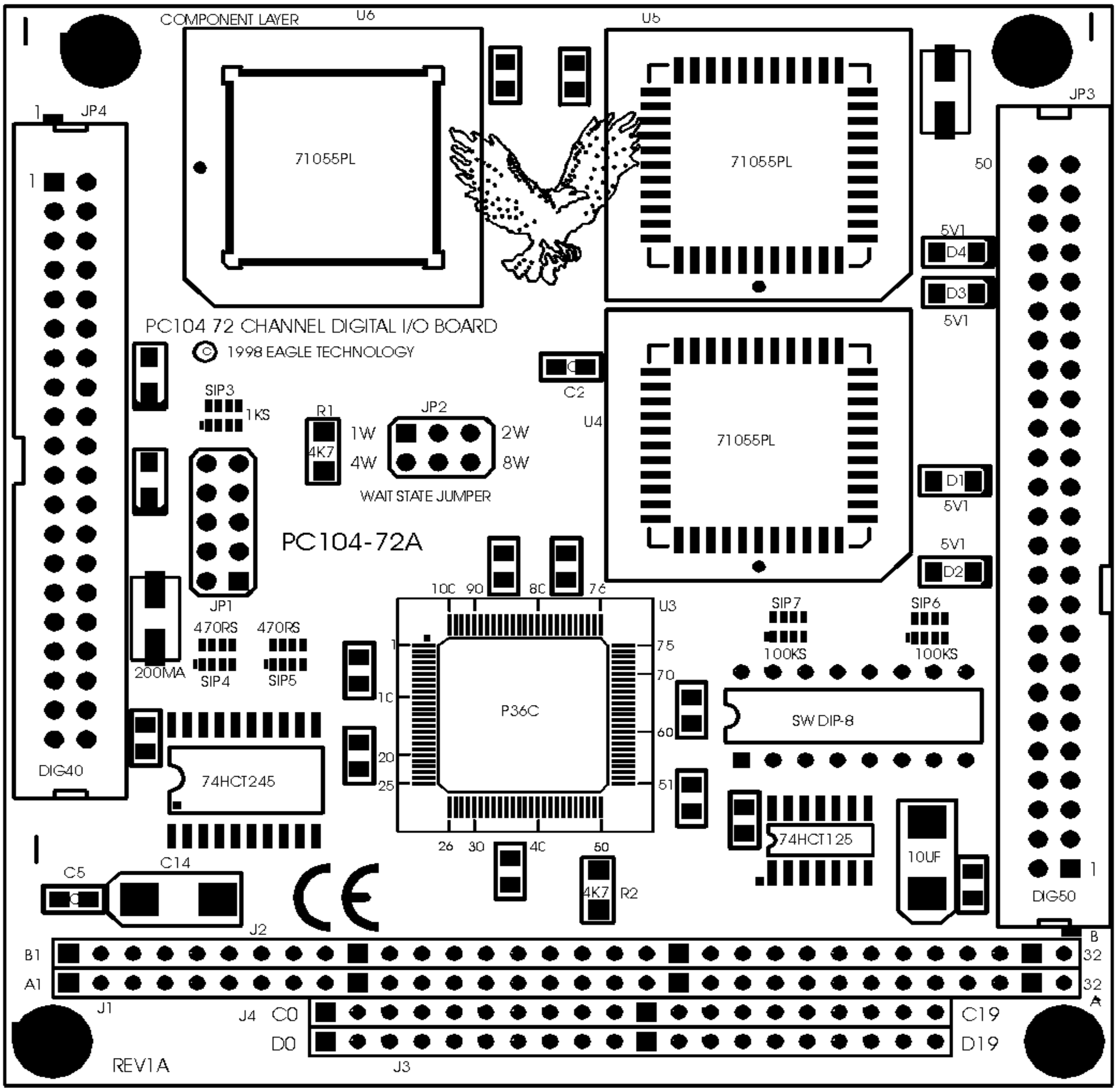
| Base Address | SW1 | SW2 | SW3 | SW4 | SW5 | SW6 | SW7 | SW8 | Base Address |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | OFF | ON | ON | ON | ON | ON | ON | ON | 1000 |
| 1020 | OFF | ON | ON | ON | ON | ON | ON | OFF | 1020 |
| 1040 | OFF | ON | ON | ON | ON | ON | OFF | ON | 1040 |
| 1060 | OFF | ON | ON | ON | ON | ON | OFF | OFF | 1060 |
| 1080 | OFF | ON | ON | ON | ON | OFF | ON | ON | 1080 |
| 10A0 | OFF | ON | ON | ON | ON | OFF | ON | OFF | 10A0 |
| 10C0 | OFF | ON | ON | ON | ON | OFF | OFF | ON | 10C0 |
| 10E0 | OFF | ON | ON | ON | ON | OFF | OFF | OFF | 10E0 |
| 1100 | OFF | ON | ON | ON | OFF | ON | ON | ON | 1100 |
| 1120 | OFF | ON | ON | ON | OFF | ON | ON | OFF | 1120 |
| 1140 | OFF | ON | ON | ON | OFF | ON | OFF | ON | 1140 |
| 1160 | OFF | ON | ON | ON | OFF | ON | OFF | OFF | 1160 |
| 1180 | OFF | ON | ON | ON | OFF | OFF | ON | ON | 1180 |
| 11A0 | OFF | ON | ON | ON | OFF | OFF | ON | OFF | 11A0 |
| 11C0 | OFF | ON | ON | ON | OFF | OFF | OFF | ON | 11C0 |
| 11E0 | OFF | ON | ON | ON | OFF | OFF | OFF | OFF | 11E0 |
| 1200 | OFF | ON | ON | OFF | ON | ON | ON | ON | 1200 |
| 1220 | OFF | ON | ON | OFF | ON | ON | ON | OFF | 1220 |
| 1240 | OFF | ON | ON | OFF | ON | ON | OFF | ON | 1240 |
| 1260 | OFF | ON | ON | OFF | ON | ON | OFF | OFF | 1260 |
| 1280 | OFF | ON | ON | OFF | ON | OFF | ON | ON | 1280 |
| 12A0 | OFF | ON | ON | OFF | ON | OFF | ON | OFF | 12A0 |
| 12C0 | OFF | ON | ON | OFF | ON | OFF | OFF | ON | 12C0 |
| 12E0 | OFF | ON | ON | OFF | ON | OFF | OFF | OFF | 12E0 |
| 1200 | OFF | ON | ON | OFF | OFF | ON | ON | ON | 1200 |
| 1320 | OFF | ON | ON | OFF | OFF | ON | ON | OFF | 1320 |
| 1340 | OFF | ON | ON | OFF | OFF | ON | OFF | ON | 1340 |
| 1360 | OFF | ON | ON | OFF | OFF | ON | OFF | OFF | 1360 |
| 1380 | OFF | ON | ON | OFF | OFF | OFF | ON | ON | 1380 |
| 13A0 | OFF | ON | ON | OFF | OFF | OFF | ON | OFF | 13A0 |
| 13C0 | OFF | ON | ON | OFF | OFF | OFF | OFF | ON | 13C0 |
| 13E0 | OFF | ON | ON | OFF | OFF | OFF | OFF | OFF | 13E0 |

| Base Address | SW1 | SW2 | SW3 | SW4 | SW5 | SW6 | SW7 | SW8 | Base Address |
|---|---|---|---|---|---|---|---|---|---|
| 1400 | OFF | ON | OFF | ON | ON | ON | ON | ON | 1400 |
| 1420 | OFF | ON | OFF | ON | ON | ON | ON | OFF | 1420 |
| 1440 | OFF | ON | OFF | ON | ON | ON | OFF | ON | 1440 |
| 1460 | OFF | ON | OFF | ON | ON | ON | OFF | OFF | 1460 |
| 1480 | OFF | ON | OFF | ON | ON | OFF | ON | ON | 1480 |
| 14A0 | OFF | ON | OFF | ON | ON | OFF | ON | OFF | 14A0 |
| 14C0 | OFF | ON | OFF | ON | ON | OFF | OFF | ON | 14C0 |
| 14E0 | OFF | ON | OFF | ON | ON | OFF | OFF | OFF | 14E0 |
| 1500 | OFF | ON | OFF | ON | OFF | ON | ON | ON | 1500 |
| 1520 | OFF | ON | OFF | ON | OFF | ON | ON | OFF | 1520 |
| 1540 | OFF | ON | OFF | ON | OFF | ON | OFF | ON | 1540 |
| 1560 | OFF | ON | OFF | ON | OFF | ON | OFF | OFF | 1560 |
| 1580 | OFF | ON | OFF | ON | OFF | OFF | ON | ON | 1580 |
| 15A0 | OFF | ON | OFF | ON | OFF | OFF | ON | OFF | 15A0 |
| 15C0 | OFF | ON | OFF | ON | OFF | OFF | OFF | ON | 15C0 |
| 15E0 | OFF | ON | OFF | ON | OFF | OFF | OFF | OFF | 15E0 |
| 1600 | OFF | ON | OFF | OFF | ON | ON | ON | ON | 1600 |
| 1620 | OFF | ON | OFF | OFF | ON | ON | ON | OFF | 1620 |
| 1640 | OFF | ON | OFF | OFF | ON | ON | OFF | ON | 1640 |
| 1660 | OFF | ON | OFF | OFF | ON | ON | OFF | OFF | 1660 |
| 1680 | OFF | ON | OFF | OFF | ON | OFF | ON | ON | 1680 |
| 16A0 | OFF | ON | OFF | OFF | ON | OFF | ON | OFF | 16A0 |
| 16C0 | OFF | ON | OFF | OFF | ON | OFF | OFF | ON | 16C0 |
| 16E0 | OFF | ON | OFF | OFF | ON | OFF | OFF | OFF | 16E0 |
| 1700 | OFF | ON | OFF | OFF | OFF | ON | ON | ON | 1700 |
| 1720 | OFF | ON | OFF | OFF | OFF | ON | ON | OFF | 1720 |
| 1740 | OFF | ON | OFF | OFF | OFF | ON | OFF | ON | 1740 |
| 1760 | OFF | ON | OFF | OFF | OFF | ON | OFF | OFF | 1760 |
| 1780 | OFF | ON | OFF | OFF | OFF | OFF | ON | ON | 1780 |
| 17A0 | OFF | ON | OFF | OFF | OFF | OFF | ON | OFF | 17A0 |
| 17C0 | OFF | ON | OFF | OFF | OFF | OFF | OFF | ON | 17C0 |
| 17E0 | OFF | ON | OFF | OFF | OFF | OFF | OFF | OFF | 17E0 |

| Base Address | SW1 | SW2 | SW3 | SW4 | SW5 | SW6 | SW7 | SW8 | Base Address |
|---|---|---|---|---|---|---|---|---|---|
| 1800 | OFF | OFF | ON | ON | ON | ON | ON | ON | 1800 |
| 1820 | OFF | OFF | ON | ON | ON | ON | ON | OFF | 1820 |
| 1840 | OFF | OFF | ON | ON | ON | ON | OFF | ON | 1840 |
| 1860 | OFF | OFF | ON | ON | ON | ON | OFF | OFF | 1860 |
| 1880 | OFF | OFF | ON | ON | ON | OFF | ON | ON | 1880 |
| 18A0 | OFF | OFF | ON | ON | ON | OFF | ON | OFF | 18A0 |
| 18C0 | OFF | OFF | ON | ON | ON | OFF | OFF | ON | 18C0 |
| 18E0 | OFF | OFF | ON | ON | ON | OFF | OFF | OFF | 18E0 |
| 1900 | OFF | OFF | ON | ON | OFF | ON | ON | ON | 1900 |
| 1920 | OFF | OFF | ON | ON | OFF | ON | ON | OFF | 1920 |
| 1940 | OFF | OFF | ON | ON | OFF | ON | OFF | ON | 1940 |
| 1960 | OFF | OFF | ON | ON | OFF | ON | OFF | OFF | 1960 |
| 1980 | OFF | OFF | ON | ON | OFF | OFF | ON | ON | 1980 |
| 19A0 | OFF | OFF | ON | ON | OFF | OFF | ON | OFF | 19A0 |
| 19C0 | OFF | OFF | ON | ON | OFF | OFF | OFF | ON | 19C0 |
| 19E0 | OFF | OFF | ON | ON | OFF | OFF | OFF | OFF | 19E0 |
| 1A00 | OFF | OFF | ON | OFF | ON | ON | ON | ON | 1A00 |
| 1A20 | OFF | OFF | ON | OFF | ON | ON | ON | OFF | 1A20 |
| 1A40 | OFF | OFF | ON | OFF | ON | ON | OFF | ON | 1A40 |
| 1A60 | OFF | OFF | ON | OFF | ON | ON | OFF | OFF | 1A60 |
| 1A80 | OFF | OFF | ON | OFF | ON | OFF | ON | ON | 1A80 |
| 1AA0 | OFF | OFF | ON | OFF | ON | OFF | ON | OFF | 1AA0 |
| 1AC0 | OFF | OFF | ON | OFF | ON | OFF | OFF | ON | 1AC0 |
| 1AE0 | OFF | OFF | ON | OFF | ON | OFF | OFF | OFF | 1AE0 |
| 1B00 | OFF | OFF | ON | OFF | OFF | ON | ON | ON | 1B00 |
| 1B20 | OFF | OFF | ON | OFF | OFF | ON | ON | OFF | 1B20 |
| 1B40 | OFF | OFF | ON | OFF | OFF | ON | OFF | ON | 1B40 |
| 1B60 | OFF | OFF | ON | OFF | OFF | ON | OFF | OFF | 1B60 |
| 1B80 | OFF | OFF | ON | OFF | OFF | OFF | ON | ON | 1B80 |
| 1BA0 | OFF | OFF | ON | OFF | OFF | OFF | ON | OFF | 1BA0 |
| 1BC0 | OFF | OFF | ON | OFF | OFF | OFF | OFF | ON | 1BC0 |
| 1BE0 | OFF | OFF | ON | OFF | OFF | OFF | OFF | OFF | 1BE0 |

| Base Address | SW1 | SW2 | SW3 | SW4 | SW5 | SW6 | SW7 | SW8 | Base Address |
|---|---|---|---|---|---|---|---|---|---|
| 1C00 | OFF | OFF | OFF | ON | ON | ON | ON | ON | 1C00 |
| 1C20 | OFF | OFF | OFF | ON | ON | ON | ON | OFF | 1C20 |
| 1C40 | OFF | OFF | OFF | ON | ON | ON | OFF | ON | 1C40 |
| 1C60 | OFF | OFF | OFF | ON | ON | ON | OFF | OFF | 1C60 |
| 1C80 | OFF | OFF | OFF | ON | ON | OFF | ON | ON | 1C80 |
| 1CA0 | OFF | OFF | OFF | ON | ON | OFF | ON | OFF | 1CA0 |
| 1CC0 | OFF | OFF | OFF | ON | ON | OFF | OFF | ON | 1CC0 |
| 1CE0 | OFF | OFF | OFF | ON | ON | OFF | OFF | OFF | 1CE0 |
| 1D00 | OFF | OFF | OFF | ON | OFF | ON | ON | ON | 1D00 |
| 1D20 | OFF | OFF | OFF | ON | OFF | ON | ON | OFF | 1D20 |
| 1D40 | OFF | OFF | OFF | ON | OFF | ON | OFF | ON | 1D40 |
| 1D60 | OFF | OFF | OFF | ON | OFF | ON | OFF | OFF | 1D60 |
| 1D80 | OFF | OFF | OFF | ON | OFF | OFF | ON | ON | 1D80 |
| 1DA0 | OFF | OFF | OFF | ON | OFF | OFF | ON | OFF | 1DA0 |
| 1DC0 | OFF | OFF | OFF | ON | OFF | OFF | OFF | ON | 1DC0 |
| 1DE0 | OFF | OFF | OFF | ON | OFF | OFF | OFF | OFF | 1DE0 |
| 1E00 | OFF | OFF | OFF | OFF | ON | ON | ON | ON | 1E00 |
| 1E20 | OFF | OFF | OFF | OFF | ON | ON | ON | OFF | 1E20 |
| 1E40 | OFF | OFF | OFF | OFF | ON | ON | OFF | ON | 1E40 |
| 1E60 | OFF | OFF | OFF | OFF | ON | ON | OFF | OFF | 1E60 |
| 1E80 | OFF | OFF | OFF | OFF | ON | OFF | ON | ON | 1E80 |
| 1EA0 | OFF | OFF | OFF | OFF | ON | OFF | ON | OFF | 1EA0 |
| 1EC0 | OFF | OFF | OFF | OFF | ON | OFF | OFF | ON | 1EC0 |
| 1EE0 | OFF | OFF | OFF | OFF | ON | OFF | OFF | OFF | 1EE0 |
| 1F00 | OFF | OFF | OFF | OFF | OFF | ON | ON | ON | 1F00 |
| 1F20 | OFF | OFF | OFF | OFF | OFF | ON | ON | OFF | 1F20 |
| 1F40 | OFF | OFF | OFF | OFF | OFF | ON | OFF | ON | 1F40 |
| 1F60 | OFF | OFF | OFF | OFF | OFF | ON | OFF | OFF | 1F60 |
| 1F80 | OFF | OFF | OFF | OFF | OFF | OFF | ON | ON | 1F80 |
| 1FA0 | OFF | OFF | OFF | OFF | OFF | OFF | ON | OFF | 1FA0 |
| 1FC0 | OFF | OFF | OFF | OFF | OFF | OFF | OFF | ON | 1FC0 |
| 1FE0 | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF | 1FE0 |

# Appendix B (PC104-72A Template)



PC104-72A Template