ZILOG Z80 PIO USER=S MANUAL

TABLE OF CONTENTS

Chapter 1. Introduction **Page**

1.0 Introduction.. 3
1.1 Features.. 3

Chapter 2. Architecture

2.0 Overview 4

Chapter 3. Pin Description

3.0 Pin Description................................... 6

Chapter 4. Programming the PIO

4.0 Reset.. 9
4.1 Loading the interrupt Vector................. 9
4.2 Selecting an Operating Mode............... 10
4.3 Setting the Interrupt Control Word........ 11

Chapter 5. Timing

5.0 Output Mode (Mode 0).......................... 12
5.1 Input Mode (Mode 1)............................. 13
5.2 Bidirectional Mode (Mode 2).................. 13
5.3 Control Mode (Mode 3)......................... 14

Chapter 6. Interrupt Servicing

6.0 Interrupt servicing................................ 16

Chapter 7. Applications

7.0 Extending the Interrupt Daisy Chain....... 18
7.1 I/O Device Interface.............................. 18
7.2 Control interface................................... 19

Chapter 8. Programming Summary

8.0 Load Interrupt Vector............................ 22
8.1 Set Mode.. 22
8.2 Set Interrupt Control............................. 22

List of figures

Page
Figure 2 -1 PIO Block Diagram............................... 5
Figure 2 -2 Port I/O Block Diagram......................... 5
Figure 3 -1 PIO Pin Functions................................… 8
Figure 3 -2 PIO 44 - Pin PLCC Pin Arrangements.… 8
Figure 5 -1 Mode 0 (Output) Timing.......................… 12
Figure 5 -2 Mode 1 (Input) Timing..........................… 13
Figure 5 -3 Port A, Mode 2 (Bidirectional) Timing.…. 14
Figure 5 -4 Control Mode (Mode 3) Timing.............… 15
Figure 6 -1 Interrupt Acknowledge Timing...............… 16
Figure 6 -2 Return From Interrupt Cycle.................… 17
Figure 6 -3 Daisy Chain Interrupt Service...............… 17
Figure 7 -1 Method of extending the Interrupt
Priority Daisy Chain..............................… 18
Figure 7 -2 Example of I/O Interface.......................… 19
Figure 7 -3 Control Mode Application......................… 21

CHAPTER 1 INTRODUCTION

1.0 INTRODUCTION

The Z8O Parallel I/O (PIO) Circuit is a programmable, two port device which provides a TTL compatible interface between peripheral devices and the Z80-GPU. The CPU can configure the Z8O-PIO to interface with a wide range of peripheral devices with no other external logic required, Typical peripheral devices that are fully compatible with the Z80-PIO include most keyboards, paper tape readers and punches, printers, PROM programmers, etc. The Z8O-PIO is packaged in a 40-pin DIP, or a 44-pin PLCC, or a 44-pin OFP. NMOS and CMOS versions are also available. Major features of the Z80-PIO include.

One of the unique features of the Z80-PIO that separates it from other interface controllers is that all data transfer between the peripheral device and the CPU is accomplished under total interrupt control. The interrupt logic of the PIO permits full usage of the efficient interrupt capabilities of the Z80-CPU during I/0 transfers. All logic necessary to implement a fully nested interrupt structure is included in the PIO so that additional circuits are not required. Another unique feature of the PIO is that it can be programmed to interrupt the CPU on the occurrence of specified status conditions in the peripheral device. For example, the PIO can be programmed to interrupt if any specified peripheral alarm conditions should occur. This interrupt capability reduces the amount of time that the processor must spend in polling peripheral status.

1.1 FEATURES

- Two Independent 8-Bit Bidirectional Peripheral interface Ports with >Handshake= Data Transfer Control.
- Interrupt Driven ‘Handshake’ for Fast Response.
- Any One of Four Distinct Modes of Operation May be Selected for a Port including.
  - Byte Output
  - Byte Input
  - Byte Bidirectional Bus (Available on Port A Only)
  - Bit Control Mode.

    All with Interrupt Controlled Handshake
- Daisy Chain Priority Interrupt Logic included to Provide for Automatic interrupt Vectoring Without External Logic.
- Eight Outputs are Capable of Driving Darlington Transistors
- All Inputs and Outputs Fully TTL Compatible
- Single 5V Supply and Single Phase Clock are Required.

CHAPTER 2 PIO ARCHITECTURE

2.0 OVERVIEW

A block diagram of the Z80-PIO is shown in Figure 2-1.The internal structure of the Z8O-PIO consists of a Z80-CPU bus interface, internal control logic, Port A I/O logic, Port B I/O logic and interrupt control logic. The CPU bus interface logic allows the PIO to interface directly to the Z8O-CPU with no other external Logic. However, address decoders and/or line buffers may be required for large systems. The internal control logic synchronizes the CPU data bus to the peripheral device interfaces (Port A and Port B). The two I/O ports (A and B) are virtually identical and are used to interface directly to peripheral devices.

The Port I/O logic is composed of 6 registers with „hand-shake“ control logic as shown in Figure 2-2. The registers include an 8-bit data input register, an 8-bit data output register, a 2-bit mode control register, an 8-bit mask register an 8-bit input/output select register and a 2-bit mask control register.

The 2-bit mode control register is loaded by the CPU to select the desired operating mode(byte output, byte input, byte bidirectional bus, or bit control mode). All data transfer between the peripheral device and the CPU is achieved through the data input and data output registers. Data may be written into the output register by the CPU or read back to the CPU from the input register at any time. The handshake lines associated with each port are used to control the data transfer between the PIO and the peripheral device.

The 8-bit mask register and the 8-bit input/output select register are used only in the bit control mode. In this mode, any of the eight peripheral data or control bus pins can be programmed to be an input or an output as specified by the select register. The mask register is used in this mode in conjunction with a special interrupt feature. This feature allows an interrupt to be generated when any or all of the unmasked pins reach a specified state (either High or Low). The 2-bit mask control register specifies the active state desired (High or Low) and if the interrupt should be generated when all unmasked pins are active (AND condition) or when any unmasked pin is active (OR condition). This feature reduces the requirement for CPU status checking of the peripheral by allowing an interrupt to be automatically generated on specific peripheral status conditions. For example, in a system with three alarm conditions an interrupt may be generated if any one occurs or if all three occur.

The interrupt control logic section handles all CPU interrupt protocol for nested priority interrupt structures. The priority of any device is determined by its physical location in a daisy chain configuration. Two lines are provided in each PIO to form this daisy chain. The device closest to the CPU has the highest priority. Within a PIO port A interrupts have higher priority then those of Port B. In the byte input, byte output or bidirectional modes, an interrupt can be generated whenever a new byte transfer is requested by the peripheral. In the bit Control mode an interrupt can be generated when the peripheral status matches a programmed value. The PIO provides for complete control of nested interrupts. That is, lower priority devices may not interrupt higher priority devices that have not had their interrupt service routine completed by the CPU. Higher priority devices may interrupt the servicing of lower priority devices.

When an interrupt is accepted by the CPU in Mode 2, the interrupting device must provide an 8-bit interrupt vector for the CPU. This vector is used to form a pointer to a location in the computer memory where the address of the interrupt service routine is located. The 8-bit vector from the interrupting device forms the least significant eight bits of the indirect pointer while the I Register in the CPU provides the most significant eight bits of the pointer. Each port (A and B) has an independent interrupt vector. The least significant bit of the vector is automatically set to a 0 within the PIO since the pointer must point to two adjacent memory Isolations for a complete 16-bit address.

The PIO decodes the RETI (Return from interrupt) instruction directly from the CPU data bus so that each PIO in the system knows at all times whether it is being serviced by the CPU interrupt service routine without any other communication with the CPU.

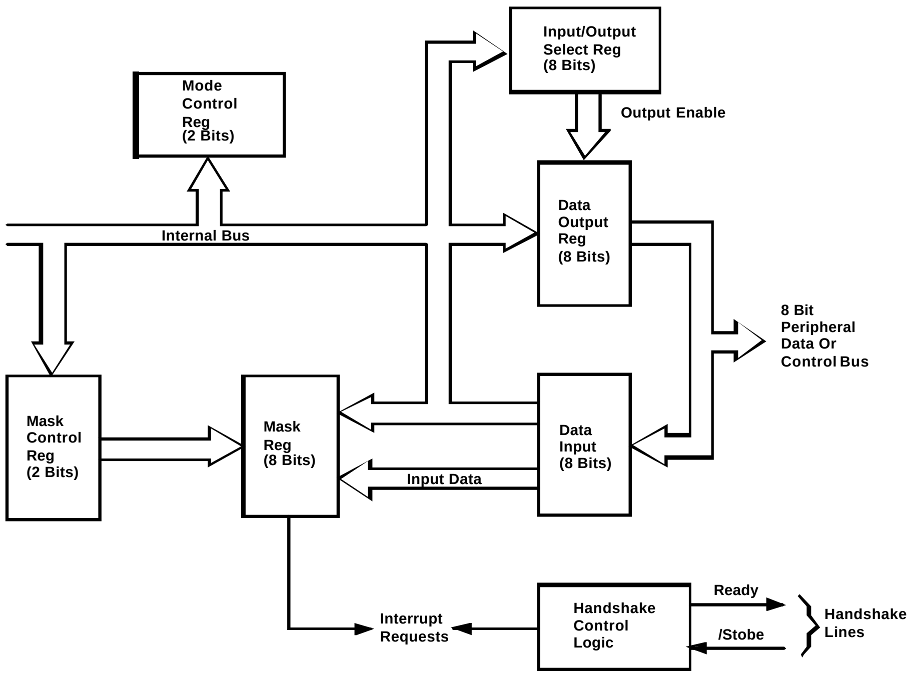

Figure 2-2. Port I/O Block Diagram

CHAPTER 3 PIN DESCRIPTION

3.0 PIN DESCRIPTION

A diagram of the Z8O-PIO pin configuration is shown in Figure 3-1. This section describes the function of each pin.

D7 - DO *Z80-CPU Data Bus* (bidirectional, tri-state) This bus is used to transfer all data and commands between the Z80-CPU and the Z80-PIO. DO is the least significant bit of the bus.

B/A Sel *Port B or A Select* (input, active High). This pin defines which port will be accessed during a data transfer between The Z80-CPU and the Z8O-PIO. A Low level on this pin selects Port A while a High level selects Port B. Often, Address bit A0 from the CPU will be used for this selection function.

C/D Sel *Control or data Select* (input, active High). This pin defines the type of data transfer to be performed between the CPU and the PlO. A High level on this pin during a CPU write to the PIO causes the Z80 data bus to be interpreted as a command for the port selected by the B/A Select line. A Low level on this pin means that the Z80 data bus is being used to transfer data between the CPU and the PlO Often Address bit Al from the CPU will be used for this function.

/CE *Chip Enable* (input, active Low). A Low level on this pin enables the PIO accept command or data inputs from the CPU during a write cycle or to transmit data to the CPU during a read cycle. This signal is generally a decode of four I/O port numbers that encompass Ports A and B, data, and control.

Φ *System Clock* (input). The Z80-PIO uses the standard Z80 system clock to synchronize certain signals internally. This is a single phase clock.

/M1 *Machine Cycle One Signal from CPU* (input, active Low). This signal from the CPU is used as a sync pulse to control several internal PIO operations. When /M1 is active and the /RD signal is active, the Z80-CPU is fetching an instruction from memory. Conversely, when /M1 is active and /IORQ is active, the CPU is acknowledging an interrupt. In addition, the /Ml signal has two other functions within the Z80-PIO.

1. /M1 synchronizes the PIO interrupt logic.
2. When /M1 occurs with out an active /RD or /IORQ signal, the PIO logic enters a reset state.

/IORQ *Input/Output Request from Z80-CPU* (input, active Low). The /IORQ signal is used in conjunction with the B/A Select, C/D Select, ICE, and /RD signals to transfer commands and data between the Z80-CPU and the Z80-PlO. When /CE, /RD, and /IORQ are active, the port addressed by B/A will transfer data to the CPU (a read operation). Conversely, when /CE and /IORQ are active but /RD is not active, then the port addressed by B/A will be written into from the CPU with either data or control information as specified by the C/D Select signal. Also, if /IORQ and /M1 are active simultaneously, the CPU is acknowledging an interrupt and the interrupting port will automatically place its interrupt vector on the CPU data bus if it is the highest priority device requesting an interrupt.

/RD *Reed Cycle Status from the Z80-CPU* (input, active Low). It /RD is active a MEMORY READ or I/O READ operation is in progress. The /RD signal is used with A/B Select, C/D Select, /CE and /IORQ signals to transfer data from the Z80-PIO to the Z80-CPU.

IEl *Interrupt Enable In* (lnput, active High). This signal is used to form a priority interrupt daisy chain when more than one interrupt driven device is being used. A High level on this pin indicates that no other device of higher priority are being serviced by a CPU interrupt service routine.

IEO Interrupt Enable Out (Output. active High). The IEO signal is the other signal required to form a daisy chain priority scheme. It is High only if IEI is High and the CPU is not servicing an interrupt from this PlO. Thus, this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.

/INT *Interrupt Request* (output, open-drain, active Low). When /INT is active, the Z80-PIO is requesting an interrupt from the Z80-CPU.

A7-AO *Port A Bus* (bidirectional, tri-state). This 8-bit bus is used to transfer data and/or status or control information between Port A of the Z80-PIO and a peripheral device. A0 is the least significant bit of the Port A data bus.

/ASTDB *Port A Strobe Pulse from Peripheral Device* (input, active Low) The meaning of this signal depends on the mode of operation selected for Port A as follows:

1. Output mode: The positive edge of this strode is issued by the peripheral to acknowledge the receipt of data made available by the PIO.

2. Input mode: The strobe is issued by the peripheral to load data from the peripheral into the Port A input register. Data is loaded into the PIO when this signal is active.

3. Bidirectional mode: When this signal is active data from the Part A output register is gated onto Port A bidirectional data bus. The positive edge of the strobe acknowledges the receipt of the data.

4. Control mode: The strobe is inhibited internally.

ARDY *Register A Ready* (output, active High). The meaning of this signal depends on the mode of operation selected for Port A as follows:

1. Output mode: This signal goes active to indicate that the Port A output register has been loaded and the peripheral data bus is stable and ready for transfer to the peripheral device.

2. Input mode: This signal is active when the Port input register is empty and is ready to accept data from the peripheral device.

3. Bidirectional mode: This signal is active when data is available in the Port A output register for transfer to the peripheral device. In this mode, data is slot placed on the Port A data bus unless ASTB is active.

4. Control mode: This signal is disabled and forced to a Low state.

B7- B0 Port *B Bus* (bidirectional, tri-state). This 8-bit bus is used to transfer data and/or status or control information between Port B of the PIO and a peripheral device. The Port B data bus is capable of supplying 1.5 m @ 1.5V to drive Darlington transistors. B0 is the least significant bit of the bus.

/BSTB *Port B Strobe Pulse from Peripheral device* (input. active Low). The meaning of this signal is similar to that at /ASTB with the following exception:

In the Port A bidirectional mode, this signal strobes data from the peripheral device into the Port A input register.

BRDY Register *B* Ready (output, active High). The meaning of this signal is similar to that of A Ready with the following exception:

In the Port A bidirectional mode this signal is High when the Port A input register is empty and ready to accept data from the peripheral device.

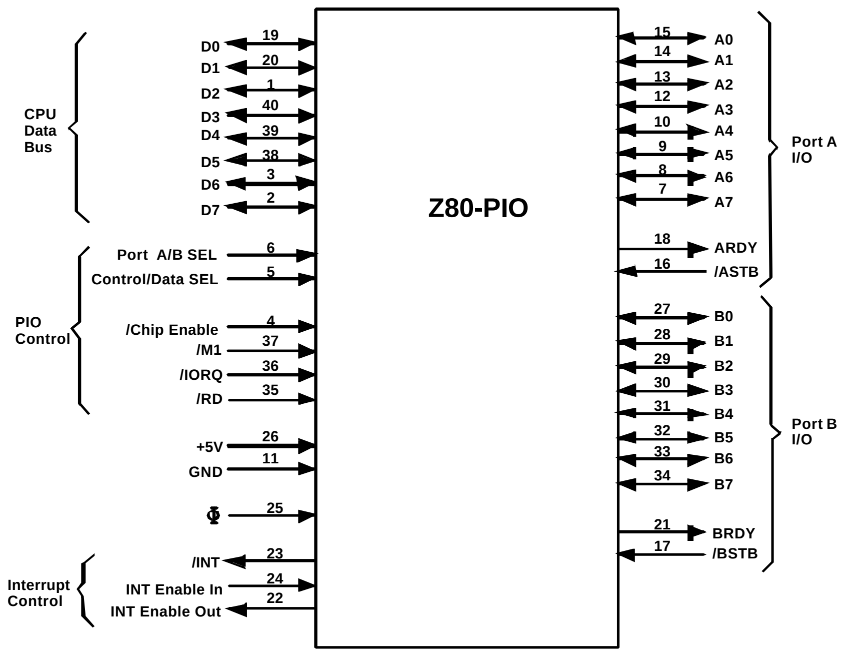

Figure 3-1. PIO Pin Functions

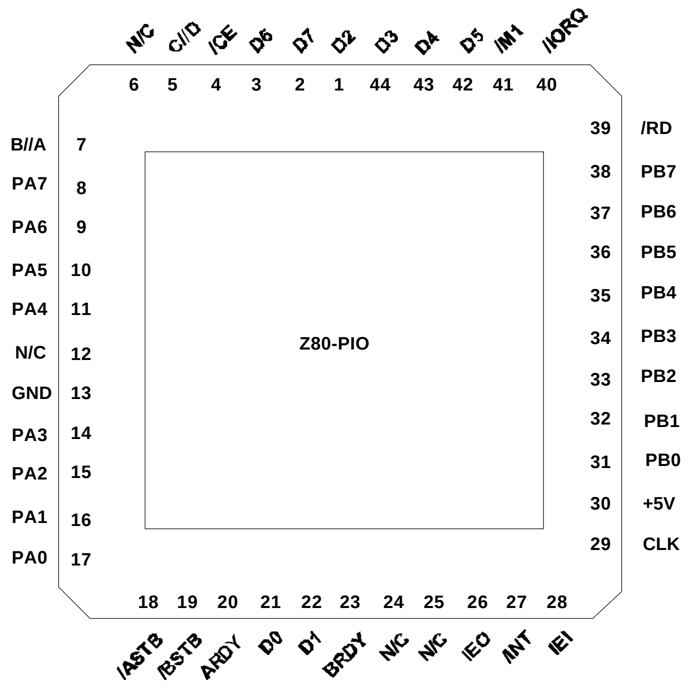

Figure 3-2. PIO 44-Pin PLCC Pin Assignments

CHAPTER 4 PROGRAMMING THE PIO

4.0 RESET

The Z8O-PIO automatically enters a reset state when power is applied. The reset state performs the following functions:

1 Both port mask registers are reset to inhibit All port data bits.

2. Port data bus lines are set to a high-impedance state and the Ready „handshake“ signals are inactive (Low) Mode 1 is automatically selected.

3. The vector address registers are not reset.

4. Both port interrupt enable flip-flops are reset.

5. Both port output registers are reset.

In addition to the automatic power-on reset, the PIO can be reset by applying an /M1 signal without the presence of a /RD or /IORQ signal. If no /RD or /IORQ is detected during /M1, the PIO will enter the reset state immediately after the /M1 signal goes inactive. The purpose of this reset is to allow a single external gate to generate a reset without a power down sequence. This approach was required due to the 40-pin packaging limitation.

Once the PIO has entered the internal reset state, it is held there until the PIO receives a control word from the CPU.

4.1 LOADING THE INTERRUPT VECTOR

The PIO has been designed to operate with the Z80-CPU using the Mode 2 interrupt response. This mode requires that an interrupt vector be supplied by the interrupting device. This vector is used by the CPU to form the address for the interrupt service routine of that port. This vector is placed on the Z80 data bus during an interrupt acknowledge cycle by the highest priority device requesting service at that time. (Refer to the Z80-CPU User’s Manual Section for details on how an interrupt is serviced by the CPU). The desired interrupt vector is loaded into the PIO by writing a control word to the desired port of the PIO with the following format:

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	V2	V1	0

0 in D0 Signifies This Control Word
is an Interrupt Vector

D0 is used in this case as a flag bit which when low, causes V7 through V1 to be loaded into the vector register. At interrupt acknowledge time, the vector of the interrupting port will appear on the Z80 data bus exactly as shown in the format above.

4.2 SELECTING AN OPERATING MODE

Port A of the PIO may be operated in any of four distinct modes: Mode 0 (output mode), Mode 1 (input mode), Mode 2 (bidirectional mode), and Mode 3 (control mode) Note that the mode numbers have been selected for mnemonic significance; ie., 0 = Out, 1 = In, 2 Bidirectional. Port B can operate in any of these modes except Mode 2.

The mode of operation most be established by writing a control word to the PIO in the following format:

D7	D6	D5	D4	D3	D2	D1	D0
M1	M0	X	X	1	1	1	1
Mode Word		X=Unused Bit		Signifies Modes Word To Be Set			

Bits D7 and D0 from the binary code for the desired mode according to the following table:

D7	D6	Mode
0	0	0 (output)
0	1	1 (input)
1	0	2 (bidirectional)
1	1	3 (control)

Bits D5 and D4 are ignored. Bits D3-D0 must be set to 1111 to indicate „Set Mode“.

Selecting Mode 0 enables any data written to the port output register by the CPU to be enabled onto the port data bus. The contents of the output register may be changed at any time by the CPU simply by writing a new data word to the port. Also the current contents of the output register may be read back to the Z80-CPU at any time through the execution of an input instruction.

With Mode 0 active, a data write from the CPU causes the Ready handshake line of that port to go High to notify the peripheral that data is available. This signal remains High until a strobe is received from the peripheral. The raising edge of the strobe generates an interrupt (if it has been enabled) and causes the Ready line to go inactive. This very simple handshake is similar to that used in many peripheral devices.

Selecting Mode 1 puts the port into the input Mode. To start handshake operation, the CPU merely performs an input read operation from the port. This activates the Ready line to the peripheral to signify that data should be loaded Into the empty input register. The peripheral device then strobes data into the port input register using the strode line. Again, the rising edge of the strobe causes an interrupt request (it has been enabled) and deactivates the Ready signal. Data may be strobed into the input register regardless of the state of the Ready signal if care is taken to prevent a data overrun condition.

Mode 2 is a bidirectional data transfer mode which uses all four handshake lines. Therefore, only Port A may be used for Mode 2 operation. Mode 2 operation uses the Port A handshake signals for output control and the Port B handshake signals for input control. Thus, both APDY and BRDY may be active simultaneously. The only operational difference between Mode 0 and the output portion of Mode 2 is that data from the Port A output register is allowed on to the port data bus only when /ASTB is active in order to achieve a bidirectional capability.

Mode 3 operation is intended for status and control applications and does not utilize the handshake signals. When Mode 3 is selected, the next control word sent to the PIO must define which of the port data bus lines are to be inputs and which are outputs. The format of the control word Is shown below:

D7	D6	D5	D4	D3	D2	D1	D0
I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0

If any Bit is set to a one, then the corresponding data bus line will be used as an input. Conversely, if the bit is reset, the line will be used as an output.

During Mode 3 operation, the strode signal is ignored and the Ready line is held Low. Data may be written to a port or read from a port by the Z80-CPU at any time during Mode 3 operation. When reading a port, the data returned to the CPU will be composed of input data from port data bus lines assigned as inputs plus port output register data from those lines assigned as outputs
.

4.3 SETTING THE INTERRUPT CONTROL WORD

The interrupt control word for each port has the following format

D7	D6	D5	D4	D3	D2	D1	D0
Enable Int.	AND/ OR	High/ Low	Mask Follows	0	1	1	1
	Used in Mode 3 Only			Signifies Interrupt Control Word			

If bit D7 = 1, the interrupt enable flip-flop of the port is set and the port may generate an interrupt. If bit D7 = 0, the enable flag is reset and interrupts may not be generated If an interrupt is pending when the enable flag is set, it will then be enabled onto the CPU interrupt request line, bits D6, D5, and D4 are used only with Mode 3 operation. However, setting bit D4 of the interrupt control word during any mode of operation will cause any pending interrupt to be reset. These three bits are used to allow for interrupt operation in Mode 3 when any group of the I/0 lines go to certain defined states. Bit D6 (AND/OR) defines the logical operation to be performed in port monitoring. If bit D6 = 1 an AND function is specified and if D6=0, an OR function is specified. For example if the AND function is specified, all bits must go to a specified state before an interrupt will be generated while the OR function will generate an interrupt if any specified bit goes to the active state.

Bit D5 defines the active polarity of the port data bus line to be monitored. If bit D5 = 1, the port data lines are monitored for a high state while if D6 = 0, they will be monitored for a low state.

If bit D4 = 1, the next control word sent to the PIO must define a mask as follows:

D7	D6	D5	D4	D3	D2	D1	D0
MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0

Only those port lines whose mask bit is zero will be monitored for generating an interrupt is high, the forced state of Ready will prevent input register data from changing while the CPU is reading the PIO. Ready will go High again after the trailing edge of the /IORQ as previously described.

CHAPTER 5 TIMING

5.0 OUTPUT MODE (MODE 0)

Figure 5-1 illustrates the timing associated with Mode 0 operation. An output cycle is always started by the execution of an output instruction by the CPU. A /WR* pulse is generated by the PIO during a CPU I/O write operation and is used to latch the data from the CPU data bus into the addressed ports (A or B) output register.

The rising edge of the /WR* pulse then raises the Ready flag after the next falling edge of Φ to indicate that data is available for the peripheral device. In most systems, the rising edge of the Ready signal can be used as a latching signal in the peripheral device if desired.

The Ready signal will remain active until: (1) a positive edge is received from the strobe line indicating that the peripheral has taken the data, or (2) if already active, Ready will be forced low one and one half Φ cycles after the leading edge of /IORQ if the ports output register is written into.

Ready will return high on the first falling edge of Φ after the trailing edge of /IORQ. This guarantees that Ready is low when port data is changing.

The Ready signal will not go inactive until a falling edge occurs on the clock (Φ) line. The purpose of delaying the negative transition of the Ready signal until after a negative clock transition is that it allows for a very simple generation scheme for the strobe pulse. By merely connecting the Ready line to the strobe line, a strobe with a duration of one clock period will be generated with no other logic required.

The positive edge of the strode pulse automatically generates an /INT request if the interrupt enable flip-flop has been set and this device is the highest priority device requesting an interrupt.

If the PIO is not in a reset state, the output register may be loaded before Mode 0 is selected. This allows the port output lines to become active in a user defined state.

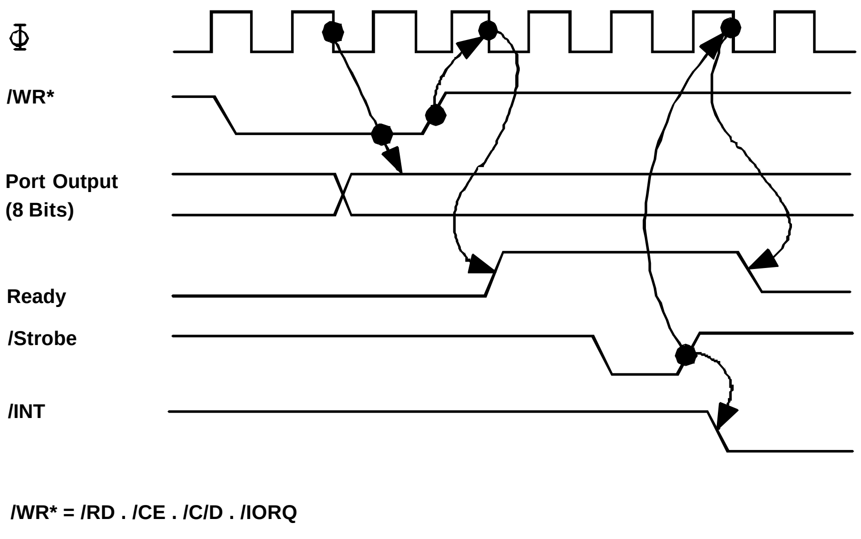

Figure 5-1. Mode 0 (Output) Timing

5.1 INPUT MODE (MODE 1)

Figure 5-2 illustrates the timing of an input cycle. The peripheral initiates this cycle using the strobe line after the CPU has performed a data read. A low level on this line loads data into the port input register and the rising edge of the strobe line activates the interrupt request line (/INT) if interrupt enable is set and this is the highest priority requesting device.

The next falling edge of the clock line (Φ) will then reset the Ready line to an inactive state signifying that the input register is full and further loading must be inhibited until the CPU reads the data. The CPU will in the course of its interrupt service routine read the data from the interrupting port. When this occurs, the positive edge from the CPU /RD signal will raise the Ready line with the next low going transition of Φ indicating that new data can be loaded info the PIO. If already active, Ready will be forced low one and one-half Φ periods following the leading edge of /IORQ during a read of a PIO port. If the user strobes data into the PIO only when Ready.

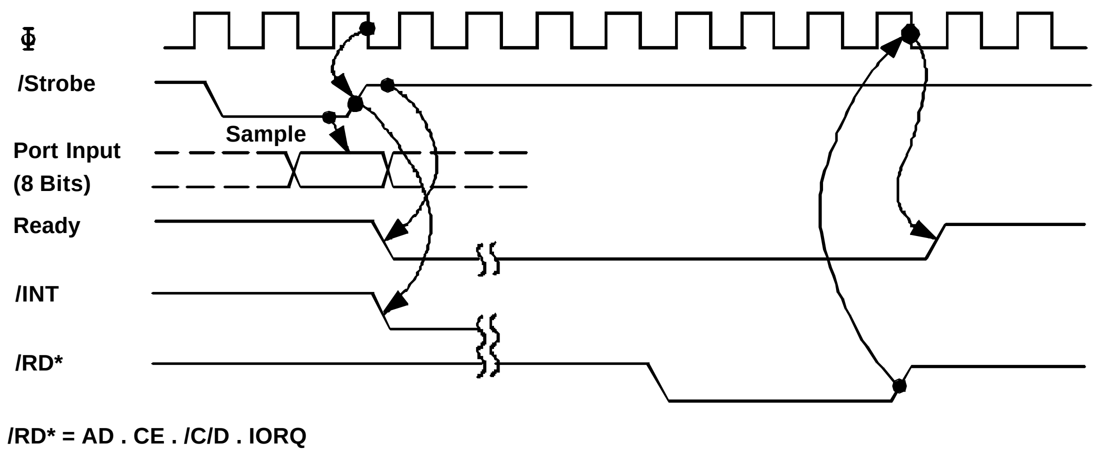

Figure 5-2. Mode 1 (Input) Timing

5.2 BIDIRECTIONAL MODE (MODE 2)

This mode is merely a combination of Mode 0 and Mode 1 using all four handshake lines. Since it requires all four lines, it is available only on Port A. When this mode is used on Port A, Port B must be set to the Bit Control Mode, The same interrupt vector will be returned for a Mode 3 interrupt on Port B and an input transfer interrupt during Mode 2 operation of Port A. Ambiguity is avoided if Port B is operated in a polled mode and the Port B mask register is set to inhibit all bits.

Figure 5-3 illustrates the timing for this mode. It is almost identical to that previously described for Mode 0 and Mode 1 with the Port A handshake lines used for output control and the Port B lines used for input control. The difference between the two modes is that, in Mode 2, data is allowed out onto the bus only when the A strobe is Low. The rising edge of this strobe can be used to latch the data into the peripheral since the data will remain stable until after this edge. The input portion at Mode 2 operates identically to Mode 1. Note that both Port A and Port B must have their interrupts enabled to achieve an interrupt driven bidirectional transfer.

The peripheral must not gate data onto a port data bus while /ASTB is active. Bus connection is avoided if the peripheral uses /BSTB to gate input data onto the bus. The PIO uses the /BSTB low level to latch this data. The PlO has been designed with a zero hold time requirement for the data when latching in this mode so that this simple gating structure can be used by the peripheral. That is, the data can be disabled from the bus immediately after the strobe rising edge.

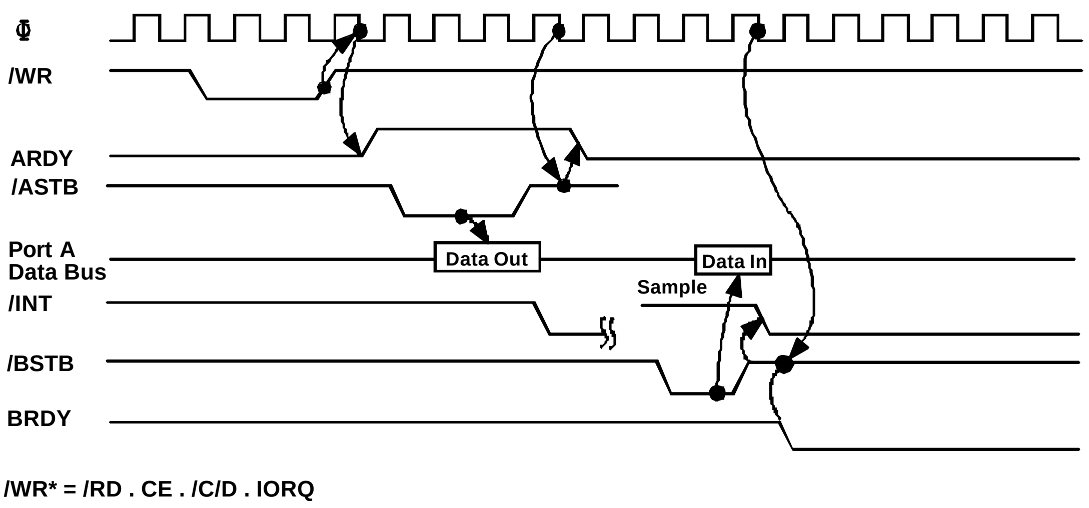

Figure 5-3. Port A, Mode 2 (Bidirectional) Timing

5.3 CONTROL MODE (MODE 3)

The control mode does not utilise the handshake signals and a normal port write or port read can be executed at any time. When writing, the data will be latched into output registers with the same timing as Mode 0. ARDY will be forced low whenever Port A is operated in Mode 3. BRDY will be held low whenever Port B is operated in Mode 3 unless Port A is in Mode 2. In the latter case, the state of BRDY will not be affected.

When reading the PlO, the data returned to the CPU will be composed of output register data from those port data lines assigned as outputs and input register data from those port data lines assigned as Inputs. The input register will contain data which was present immediately prior to the falling edge of /RD. See Figure 5-4.

An interrupt will be generated if interrupts from the port are enabled and the data on the port data lines satisfies the logical equation defined by the 8-bit mask and 2-bit mask control registers. Another interrupt will not be generated until a change occurs in the status of the logical equation. A Mode 3 interrupt will be generated only if the result of a Mode 3 logical operation changes from false to true. For example, assume that the Mode 3 logical equation is an OR function. An unmasked port data line becomes active and an interrupt is requested. If a second unmasked port data line becomes active concurrently with the first, a new interrupt will not be requested since a change in the result of the Mode 3 logical operation has not occurred.

If the result of a logical operation becomes true immediately prior to or during /M1 an interrupt will be requested after the trailing edge of /M1.

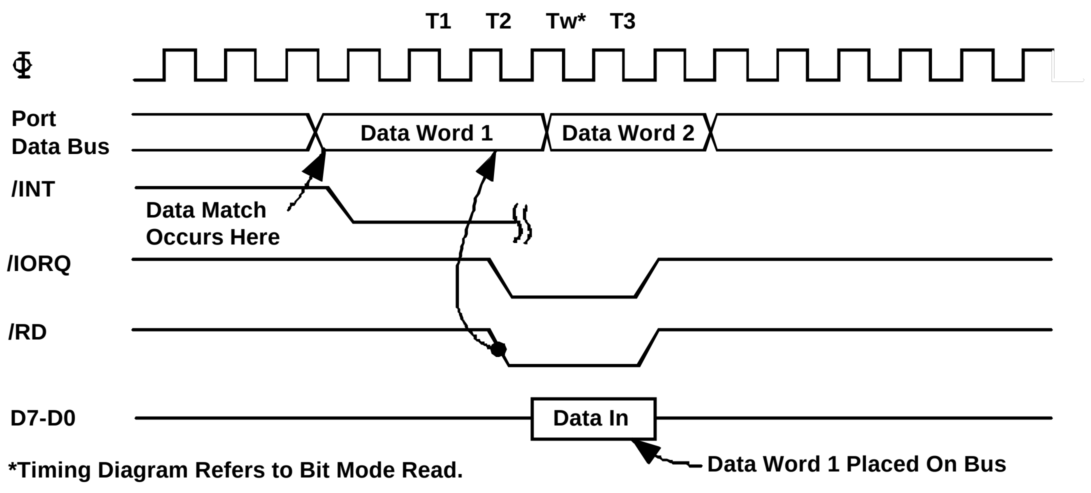

Figure 5-4. Control Mode (Mode 3) Timing

CHAPTER 6 INTERRUPT SERVICING

6.0 INTERRUPT SERVICING

Some time after an interrupt is requested by thin PIO, the CPU will send out an interrupt acknowledge (/M1 and /IORQ). During this time the interrupt logic of the PIO will determine the highest priority port which is requesting an interrupt. (This is simply the device with its Interrupt Enable input high and its Interrupt Enable Output low). To insure that the daisy chain enable lines stabilize, devices are inhibited from changing their interrupt request status when /M1 is active. The highest priority device places the contents of its interrupt vector register onto the Z80 data bus during interrupt acknowledge.

Figure 8-1 illustrates the timing associated with interrupt requests. During /M1 time, no new interrupt requests can be generated. This gives time for the Int Enable signals to ripple through up to four PIO circuits. The PIO, With IEI high and IEO Low during /INTA, will place the 8-bit interrupt vector of the appropriate port on the data bus at this time.

If an interrupt requested by the PIO is acknowledged, the requesting port is under service. IEO of this port will remain low until a return from interrupt instruction (RETI) is executed while IEI of the part is high. If an interrupt request is not acknowledged, IEO will be forced high for one /M1 cycle after the PIO decodes the opcode >ED'. This action guarantees that the 2-byte RETI instruction is decoded by the proper PIO port
(Figure 6-2).

Figure 6-3 illustrates a typical rested interrupt sequence that could occur with four ports connected in the daisy chain. In this sequence Port 2A requests and is granted an interrupt. While this port is being serviced, a higher priority port (1B) requests and is granted an interrupt. The service routine for the higher priority port is completed and a RETI instruction is executed to indicate to the port that its routine is complete. At this time the service routine of the lower priority port is completed.

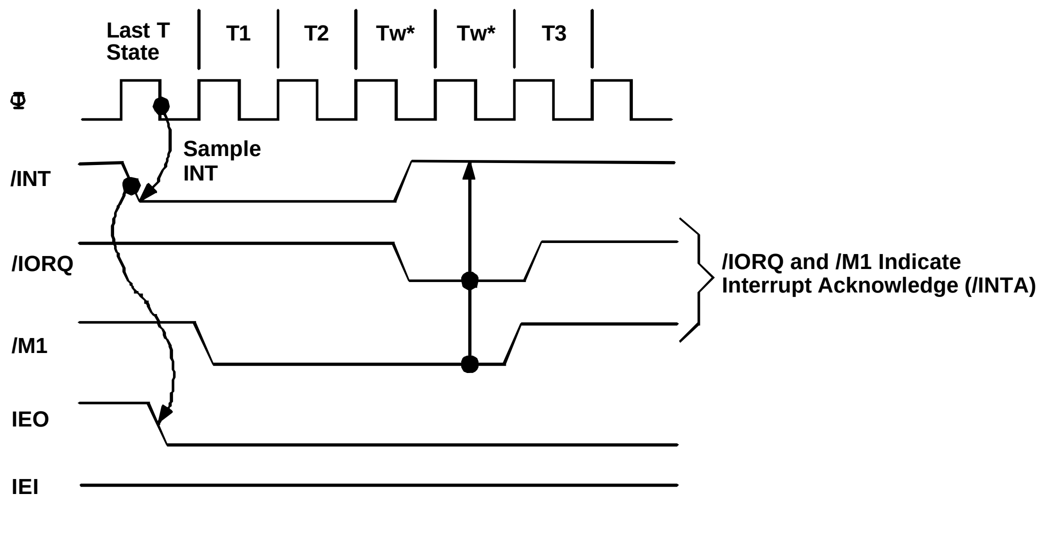

Figure 6-1. Interrupt Acknowledge Timing

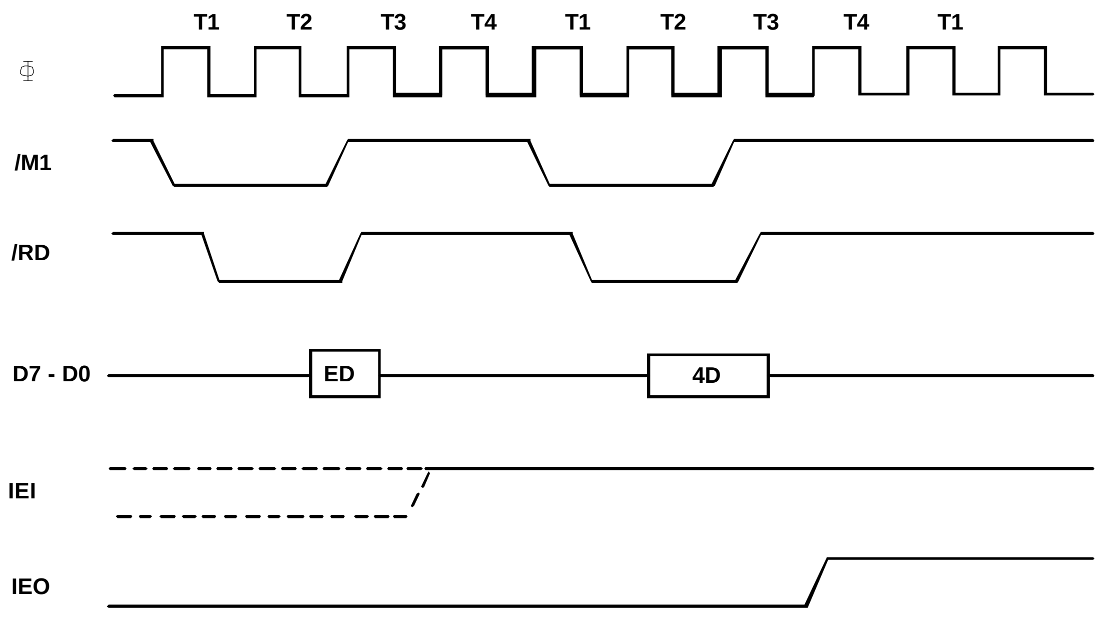

Figure 6-2. Return From Interrupt Cycle

CHAPTER 7 APPLICATIONS

7.0 EXTENDING THE INTERRUPT DAISY CHAIN

Without any external logic, a maximum of four Z80-PIO devices may be daisy chained into a priority interrupt structure. This limitation is required so that the interrupt enable status (IEO) ripples through the entire chain between the beginning of /M1 and the beginning of /IORQ during an interrupt acknowledge cycle. Since the interrupt enable status cannot change during /M1, the vector address returned to the CPU is assured to be from the highest priority device which requested an interrupt.

If more than four PIO devices must be accommodated, a „look-ahead“ structure may be used as shown in Figure 7-1. With this technique, more than thirty PIO’s may be chained together using standard TTL logic.

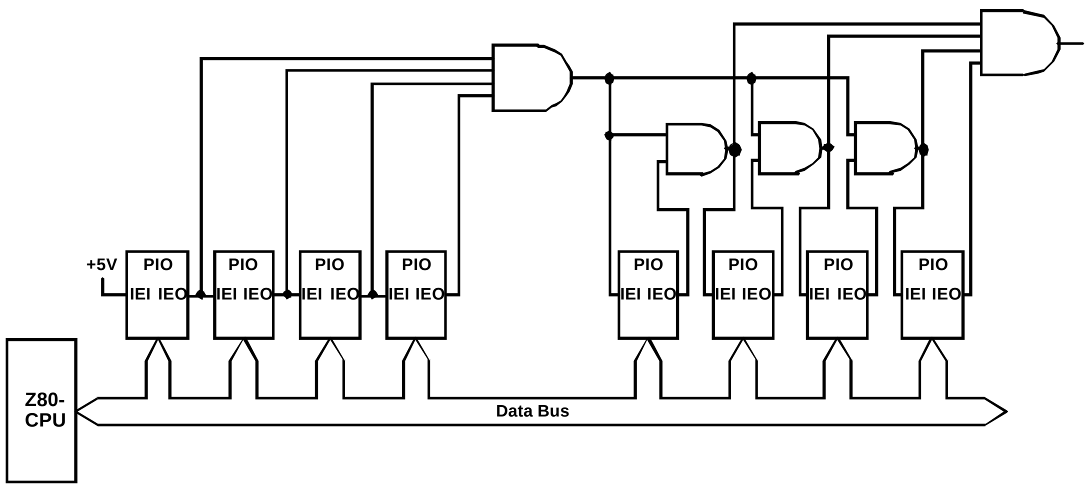

Figure 7-1. A Method of Extending the Interrupt Priority Daisy Chain

7.1 I/O DEVICE INTERFACE

In this example the Z80-PIO is connected to an I/O terminal device which communicates over an 8-bit parallel bidirectional data bus as illustrated in Figure 7-2. Mode 2 operation (bidirectional) is selected by sending the following control word to Port A:

D7	D6	D5	D4	D3	D2	D1	D0
1	0	X	X	1	1	1	1
				Mode Control			

Next, the proper interrupt vector is loaded (refer to CPU Manual for details on the operation of the interrupt).

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	V2	V1	0

Interrupts are then enabled by the rising edge of the first /M1 after the interrupt mode word is set unless that /M1 defines an interrupt acknowledge cycle. If a mask follows the interrupt mode word, interrupts are enabled by the rising edge of the first /M1 following the setting of the mask.

Data can now be transferred between the peripheral and the CPU. The timing for this transfer is as described is Section 5.0.

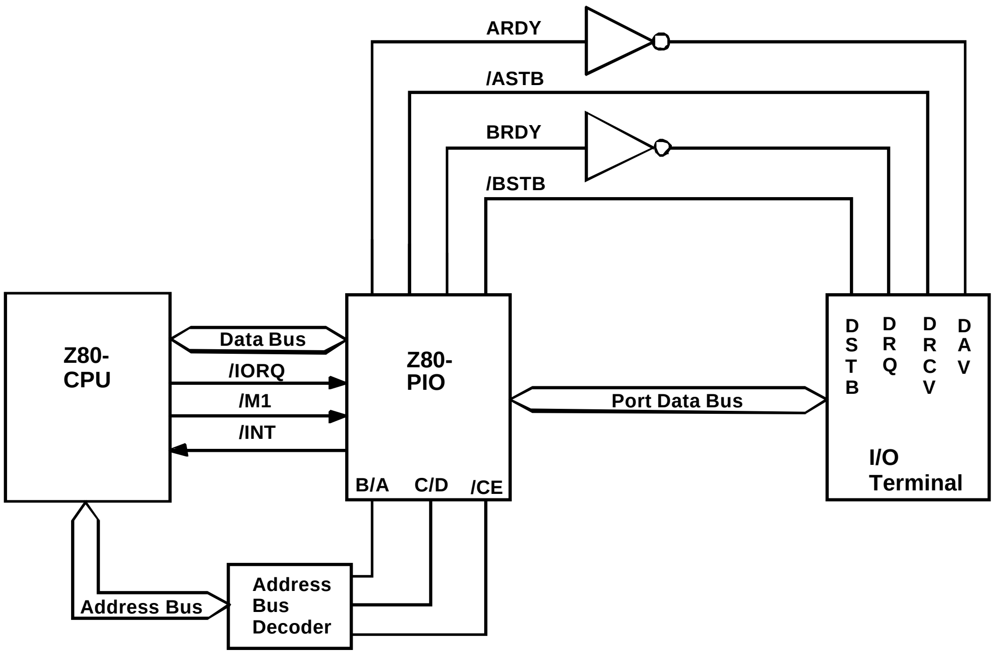

Figure 7-2. Example of I/O Interface

7.2 CONTROL INTERFACE

A typical control mode application is illustrated in Figure 7-3. Suppose an industrial process is to be monitored. The occurrence of any abnormal operating condition is to be reported to a Z80-CPU based control system. The process control and status word has the following format:

D7	D6	D5	D4	D3	D2	D1	D0
Special Test	Turn on Power	Power Failure Alarm	Halt Process ing	Temp Alarm	Turn Htrs On	Pressure on System	Pressure Alarm

The PIO may be used as follows. First Port A is set for Mode 3 operation by writing the following control word to Port A.

D7	D6	D5	D4	D3	D2	D1	D0
1	1	X	X	1	1	1	1

Whenever Mode 3 is selected, the next control word sent to the port must be an I/O select word. In this example we wish to select port data lines A5, A3. and A0 as inputs and so the following control word is written:

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	0	1	0	0	1

Next the desired interrupt vector must be Loaded (refer to the CPU manual for details):

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	V2	V1	0

An interrupt control word is next sent to the port:

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	0	0	1	1	1
Enable Interrupts	OR Logic	Active High	Mask Follows	Interrupt Control			

The mask word following the interrupt mode word is:

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	1	0	1	1	0

Now, if a sensor puts a high level on line A5, A3 or A0, an interrupt request will be generated. The mask word may select any combination of inputs or outputs to cause an interrupt. For example, it the mask word above had been:

D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	0	1	1	0

then an interrupt request would also occur if bit A7 (Special Test) of the output register was set.

Assume that the following port assignments are to be used.

E0H = Port A Data
E1H = Port B Data
E2H = Port A Control
E3H = Port B Control

All port numbers are in hexadecimal notation. This particular assignment of port numbers is convenient since A0 of the address bus can be used as the Port A/B Select and A1 of the address bus can be used as the Control/Data Select. The Chip Enable would be the decode of CPU address bits A7 through A2 (111000). Note that if only a few peripheral devices are being used, a Chip Enable decode may not be required since a higher order address bit could be used directly.

CHAPTER 8 PROGRAMMING SUMMARY

8.0 LOAD INTERRUPT VECTOR

V7	V6	V5	V4	V3	V2	V1	0

8.1 SET MODE

M1	M0	X	X	1	1	1	1

M1	M0	Mode
0	0	Output
0	1	Input
1	0	Bidirectional
1	1	Bit Control

When selecting Mode 3, the next word must set the =I/O Register:

8.2 SET INTERRUPT CONTROL

Enable Int	AND/ OR	High/ Low	Mask Follows	0	1	1	1
	Used in Mode 3 Only						

If the "mask follows" bit is high, the next control word written to the port must be the mask:

MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0

MB = 0, Monitor bit
MB = 1, Mask bit from being monitored

Also, the interrupt enable flip-flop of a port may be set or reset without modifying the rest of the interrupt control word by using the following command:

Enable Int.	X	X	X	0	0	1	1