JOHN BELL ENGINEERING, INC.

Z80 MICROCOMPUTER
PN 80-Z80

JOHN BELL ENGINEERING'S Z80 SINGLE BOARD MICROCOMPUTER IS DESIGNED FOR USE AS A CONTROL COMPUTER AND INCORPORATES THE Z80 PIO AND Z80 MICROPROCESSOR. THE Z80 CPU IS SOFTWARE COMPATIBLE WITH THE 8085 AND 8080.

THE Z80 PIO HAS 2 INPUT/OUTPUT PORTS WHICH ARE SOFTWARE CONTROLLED FOR INPUT OR OUTPUT. DOCUMENTATION ON THE Z80 PIO IS INCLUDED.

TWO 2114S PROVIDE 1K OF RANDOM ACCESS MEMORY AND ONE 2716 PROVIDES 2K OF EPROM MEMORY. A 74LS10 TAKES CARE OF ADDRESSING AND A 74LS04 IS USED TO GENERATE AN RC CONTROLLED 2MHZ CLOCK. IF YOU NEED A CRYSTAL CONTROLLED CLOCK, YOU CAN SIMPLY REPLACE THE CAPACITOR.

ALL 16 DATA LINES, TWO READY LINES, TWO STROBE LINES, RESET LINE AND WAIT LINE COME TO THE EDGE CONNNECTOR ON THE CIRCUIT BOARD.

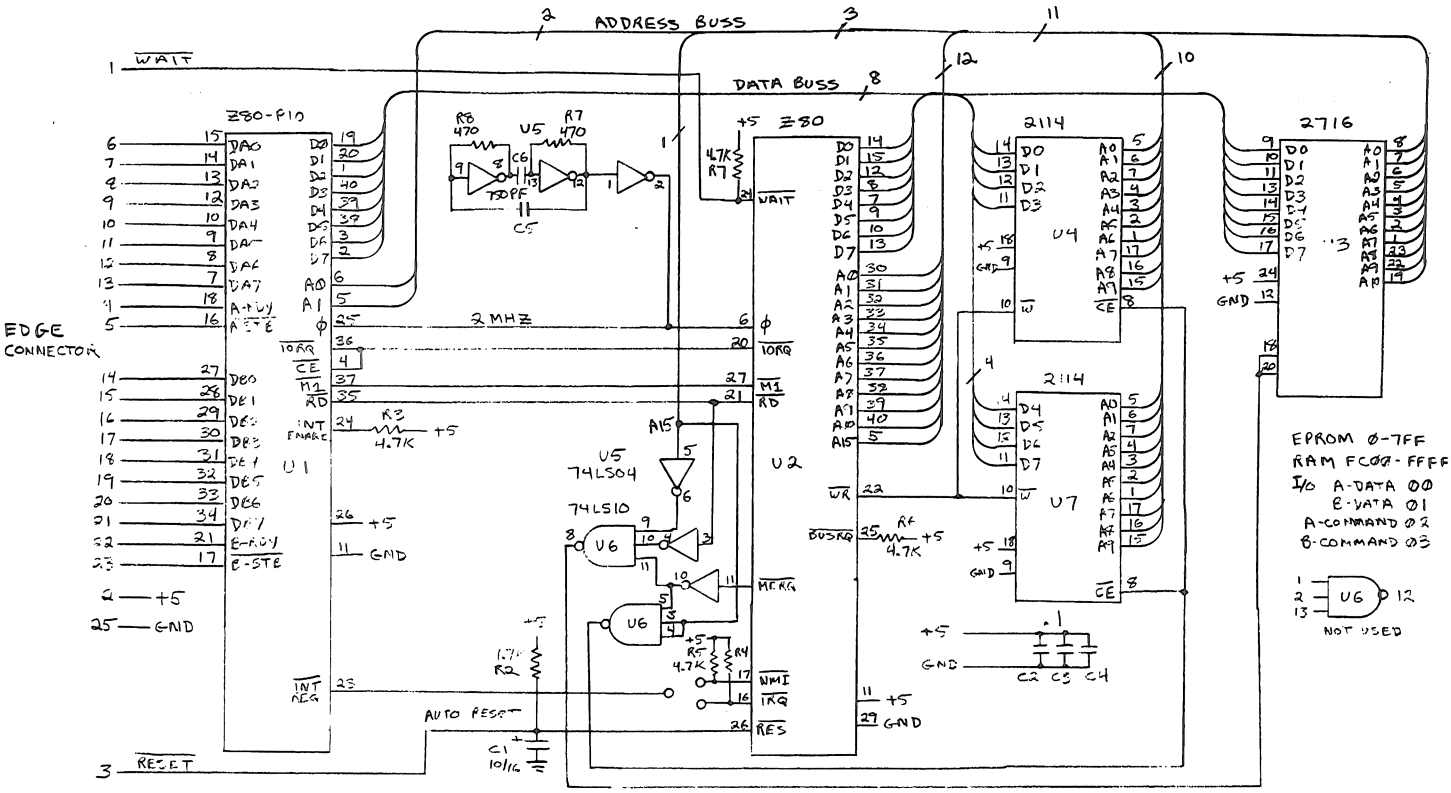
RESISTOR R2 AND CAPACITOR C1 FORM A TIMING CIRCUIT THAT CAUSES AN AUTO RESET FUNCTION. JUMPER HOLES FOR INTERRUPT ARE ALSO PROVIDED.

A SIMPLE TEST PROGRAM IS PROVIDED THAT WILL OUTPUT SQUARE WAVES ON PORT A. A SPEAKER IN SERIES WITH A 1K RESISTOR CAN BE CONNECTED BETWEEN PIN 2 (5V POWER SUPPLY) AND PIN 6 (THE 0 DATA BIT OF PORT 1). THE SPEAKER SHOULD PRODUCE A CONSTANT TONE. DO NOT ATTEMPT TO CONNECT THE SPEAKER DIRECTLY TO THE COMPUTER WITHOUT THE RESISTOR IN SERIES AS THIS WILL PERMANENTLY DAMAGE THE PIO CHIP. THE OTHER LINES OF PORT A WILL PRODUCE FREQUENCIES EACH BEING AN OCTIVE LOWER THAN THE FIRST FREQUENCY AT DATA BIT 0. AN LED CAN BE CONNECTED IN SERIES WITH A 470Ω RESISTOR BETWEEN PIN 2 AND PIN 13 OR 12 OF THE EDGE CONNECTOR. THE FREQUENCY AT THESE PINS IS SO LOW THAT YOU CAN ACTUALLY SEE THE LED FLASH.

PLEASE NOTE THE MODIFICATION FOR USING A 2532 INSTEAD OF THE 2716 TO GIVE 4K OF EPROM MEMORY. FUTURE CIRCUIT BOARDS WILL NOT NEED MODIFICATION. THE MODIFICATION IS ON PAGE 3 OF THE DOCUMENTATION.

*PARTS OF THIS DOCUMENTATION HAVE BEEN REPRODUCED FROM THE ZILOG® Z80®-PIO Z80A®-PIO TECHNICAL MANUAL.

*Z80® IS A TRADE MARK OF ZILOG, INC..



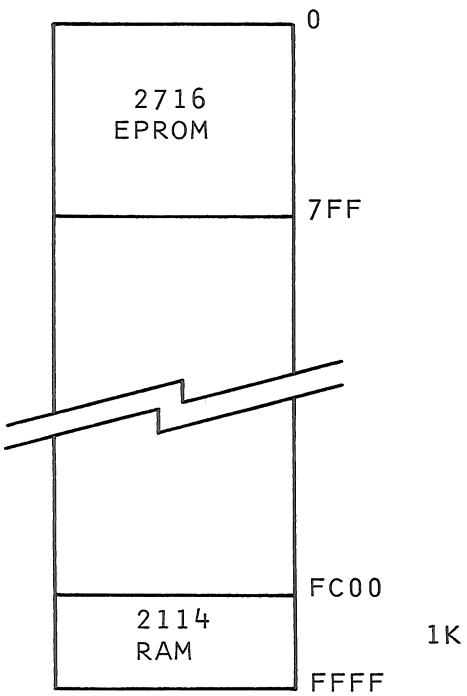
Z80 MICROCOMPUTER

PART # 80-280

EDGE CONNECTOR

PIN #	SIGNAL	PIN #	SIGNAL
2	+5		
25	GND		
3	$\overline{\text{RESET}}$		
1	$\overline{\text{WAIT}}$		
6	DAØ	14	DBØ
7	DA1	15	DB1
8	DA2	16	DB2
9	DA3	17	DB3
10	DA4	18	DB4
11	DA5	19	DB5
12	DA6	20	DB6
13	DA7	21	DB7
4	A-RDY	22	B-RDY
5	A-$\overline{\text{STB}}$	23	B-$\overline{\text{STB}}$

MEMORY MAP



PARTS LIST 80-280

INTEGRATED CIRCUITS

U1	Z80 PIO
U2	Z80 CPU
U3	2716 (NOT INCLUDED)
U4,U7	2114
U5	74LS04
U6	74LS10

CAPACITORS

C1	10MF/16V
C2,C3,C4	.1 DISC
C5,C6	750PF SILVER MICA

RESISTORS - $\frac{1}{4}$ WATT 5%

R1,R2,R3,R4	4.7K
R5,R6	4.7K
R7,R8	470

SOCKETS

2	40 PIN
1	24 PIN
2	18 PIN
2	14 PIN

1 80-280 CIRCUIT BOARD
1 50 PIN EDGE CONNECTOR

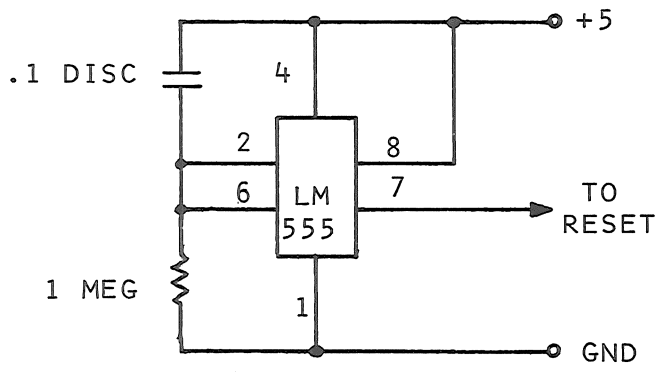
MODIFICATION FOR USE OF 2532 IN PLACE OF 2716

1. CUT BOTH TRACES THAT GO TO PIN 18 OF U3.
2. CONNECT A WIRE FROM PIN 20 OF U3 TO PIN 8 OF U6.
3. CONNECT A WIRE FROM PIN 18 OF U3 TO PIN 1 OF U2.

* PLEASE NOTE THAT ALL MODIFICATIONS SHOULD BE DONE ON THE SOLDER SIDE OF THE CIRCUIT BOARD.

* FUTURE RUNS OF THE 80-280 WILL NOT HAVE TO BE MODIFIED.

PLEASE NOTE THAT C1 (10MF/16V) IS A CHEAP WAY TO DO AUTO RESET. A BETTER RESET CIRCUIT IS SHOWN BELOW. IF THIS CIRCUIT IS USED, REMOVE C1.



AUTO RESET CIRCUIT

```
                00100 ;SIMPLE MEMORY TEST (FC00-FFFF)
                00110 ;ADDRESS BEING TESTED APPEARS
                00120 ;ON OUTPUT PORTS
                00130 ;PORT A = A0 THRU A7
                00140 ;PORT B = A8 THRU A15
                00150 ;HALT PIN 18 ON Z-80 CHIP IS SET LOW
                00160 ;ON MEMORY ERROR.
0000            00170           ORG     0
0000 3E0F       00180           LD      A,0FH   ;SET PIO MODE WORD
0002 D302       00190           OUT     (2),A   ;FOR OUTPUT ON PORT A
0004 D303       00200           OUT     (3),A   ;AND PORT B
0006 2100FC     00210 TEST      LD      HL,0FC00H       ;STARTING ADDRESS
0009 0600       00220 TEST1     LD      B,0             ;DATA TO WRITE TO RAM
000B 70         00230 TEST2     LD      (HL),B          ;WRITE THE DATA
000C 7E         00240           LD      A,(HL)          ;READ IT BACK
000D B8         00250           CP      B               ;IS IT THE SAME ?
000E 2810       00260           JR      Z,FAIL  ; *** BAD MEMORY ***
0010 05         00270           DEC     B               ;NEW DATA BYTE
0011 20F8       00280           JR      NZ,TEST2        ;DO ALL 256 BYTES
0013 7D         00290           LD      A,L
0014 D300       00300           OUT     (0),A           ;PUT ADDRESS ON PORT A
0016 7C         00310           LD      A,H
0017 D301       00320           OUT     (1),A           ;AND ON PORT B TOO !
0019 23         00330           INC     HL              ;NEXT ADDRESS
001A 7C         00340           LD      A,H
001B B7         00350           OR      A               ;IS HL=0 ??
001C 20EB       00360           JR      NZ,TEST1        ;NO. CONTINUE TEST
001E 18E6       00370           JR      TEST            ;START TEST AGAIN
                00380 ;
0020 76         00390 FAIL      HALT                    ;SET PIN 18 LOW
0021 18FD       00400           JR      FAIL            ;TO INDICATE FAIL.
0000            00410           END
00000 Total errors
```

```
0000            00010           ORG     0
                00020 ;OUTPUT A MULTIPLE OF FREQUENCIES ON PORT A & B
                00030 ;PORT A BIT 0 = 45 KHZ
                00040 ;PORT B BIT 7 = 1.5 HZ
                00050 ;PORT A BIT 6 (PIN 13) IS ABOUT 700 HZ FOR SPEAKER
0000 3E0F       00060           LD      A,0FH   ;PIO MODE WORD
0002 D302       00070           OUT     (2),A
0004 D303       00080           OUT     (3),A   ;PORT A & B SET TO OUTPUT
0006 210000     00090           LD      HL,0    ;ZERO 16 BIT COUNTER
0009 7D         00100 LOOP      LD      A,L
000A D300       00110           OUT     (0),A   ;PORT A = L
000C 7C         00120           LD      A,H
000D D301       00130           OUT     (1),A   ;PORT B = H
000F 23         00140           INC     HL      ;INCREMENT COUNTER
0010 18F7       00150           JR      LOOP
0000            00160           END
00000 Total errors
```

1.0 INTRODUCTION

The Z-80 Parallel I/O (PIO) Circuit is a programmable, two port device which provides a TTL compatible interface between peripheral devices and the Z80-CPU. The CPU can configure the Z80-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 Z80-PIO utilizes N channel silicon gate depletion load technology and is packaged in a 40 pin DIP. Major features of the Z80-PIO include:

- 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 5 volt supply and single phase clock are required.

One of the unique freatures 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/O 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.

2.0 PIO ARCHITECTURE

A block diagram of the Z80-PIO is shown in Figure 2.0-1. The internal structure of the Z80-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 Z80-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.

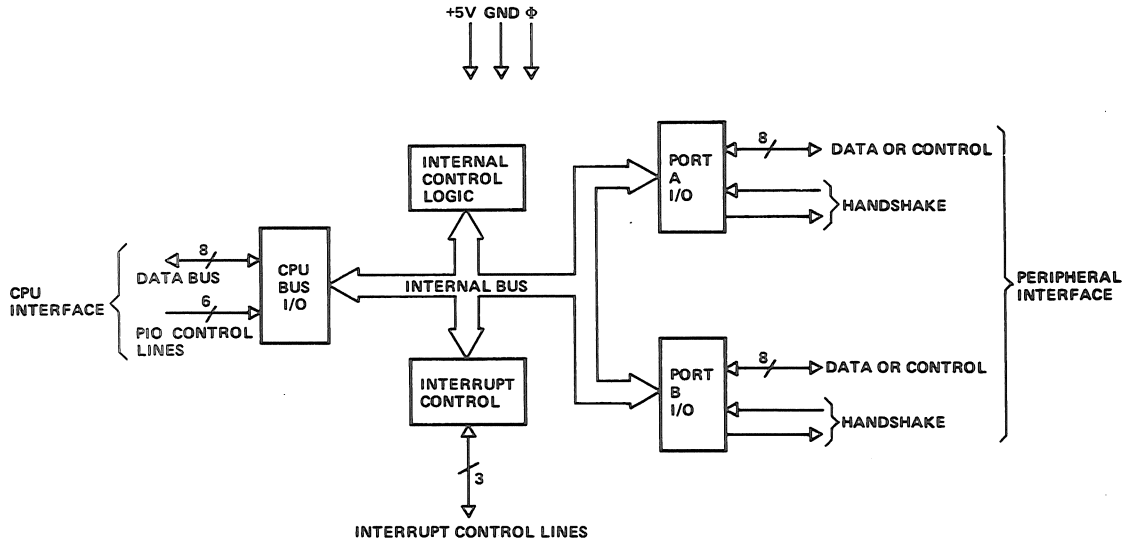


FIGURE 2.0-1
PIO BLOCK DIAGRAM

The Port I/O logic is composed of 6 registers with "handshake" control logic as shown in Figure 2.0-2. 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.

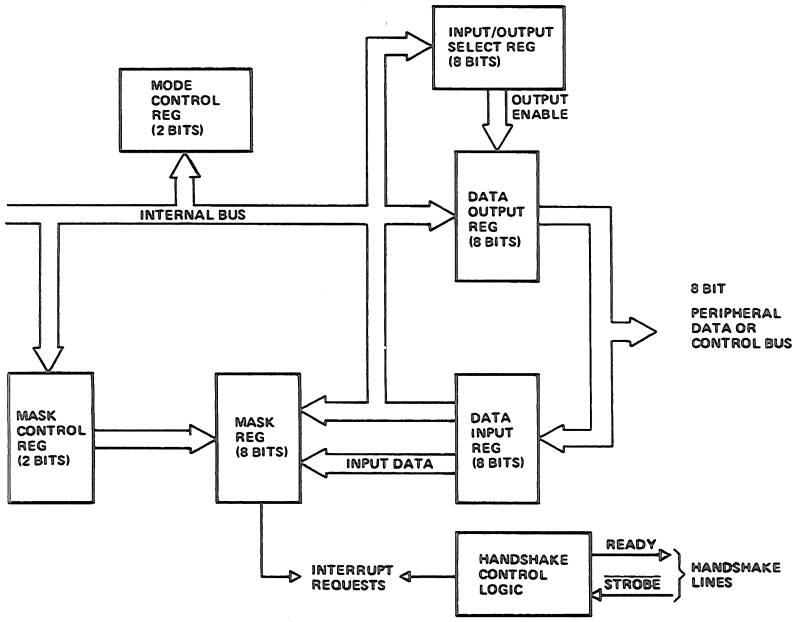


FIGURE 2.0-2
PORT I/O BLOCK DIAGRAM

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 8 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 3 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 than 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 8 bits of the indirect pointer while the I Register in the CPU provides the most significant 8 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 locations 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.

PIN DESCRIPTION

A diagram of the Z80-PIO pin configuration is shown in Figure 3.0-1. This section describes the function of each pin.

D_7-D_0 — Z80-CPU Data Bus (bidirectional, tristate)
This bus is used to transfer all data and commands between the Z80-CPU and the Z80-PIO. D_0 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 Z80-PIO. A low level on this pin selects Port A while a high level selects Port B. Often Address bit A_0 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 PIO. A high level on this pin during a CPU write to the PIO causes the Z-80 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 he Z-80 data bus is being used to transfer data between the CPU and the PIO. Often Address bit A_1 from the CPU will be used for this function.

$\overline{CE}$ — Chip Enable (input, active low)
A low level on this pin enables the PIO to 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 port A and B, data and control.

Φ — System Clock (input)
The Z80-PIO uses the standard Z-80 system clock to synchronize certain signals internally. This is a single phase clock.

$\overline{M1}$ — 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 $\overline{M1}$ is active and the $\overline{RD}$ signal is active, the Z80-CPU is fetching an instruction from memory. Conversely, when $\overline{M1}$ is active and $\overline{IORQ}$ is active, the CPU is acknowledging an interrupt. In addition, the $\overline{M1}$ signal has two other functions within the Z80-PIO.

1. $\overline{M1}$ synchronizes the PIO interrupt logic.
2. When $\overline{M1}$ occurs without an active $\overline{RD}$ or $\overline{IORQ}$ signal the PIO logic enters a reset state.

$\overline{IORQ}$ — Input/Output Request from Z80-CPU (input, active low)
The $\overline{IORQ}$ signal is used in conjunction with the B/A Select, C/D Select, $\overline{CE}$, and $\overline{RD}$ signals to transfer commands and data between the Z80-CPU and the Z80-PIO. When $\overline{CE}$, $\overline{RD}$ and $\overline{IORQ}$ are active, the port addressed by B/A will transfer data to the CPU (a read operation). Conversely, when $\overline{CE}$ and $\overline{IORQ}$ are active but $\overline{RD}$ 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 $\overline{IORQ}$ and $\overline{M1}$ 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.

$\overline{RD}$ — Read Cycle Status from the Z80-CPU (input, active low)
If $\overline{RD}$ is active a MEMORY READ or I/O READ operation is in progress. The $\overline{RD}$ signal is used with B/A Select, C/D Select, $\overline{CE}$, and $\overline{IORQ}$ signals to transfer data from the Z80-PIO to the Z80-CPU.

IEI — Interrupt Enable In (input, 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 devices 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 PIO. Thus this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.

$\overline{\text{INT}}$ — Interrupt Request (output, open drain, active low)
When $\overline{\text{INT}}$ is active the Z80-PIO is requesting an interrupt from the Z80-CPU.

$A_0 - A_7$ — Port A Bus (bidirectional, tristate)
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. A_0 is the least significant bit of the Port A data bus.

$\overline{\text{A STB}}$ — 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 strobe 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 Port 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.

A RDY — 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 A 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 not placed on the Port A data bus unless $\overline{\text{A STB}}$ is active.
4) Control mode: This signal is disabled and forced to a low state.

$B_0 - B_7$ — Port B Bus (bidirectional, tristate)
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 1.5ma @ 1.5V to drive Darlington transistors. B_0 is the least significant bit of the bus.

$\overline{\text{B STB}}$ — Port B Strobe Pulse from Peripheral Device (input, active low)
The meaning of this signal is similar to that of $\overline{\text{A STB}}$ with the following exception:
In the Port A bidirectional mode this signal strobes data from the peripheral device into the Port A input register.

B RDY — 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.

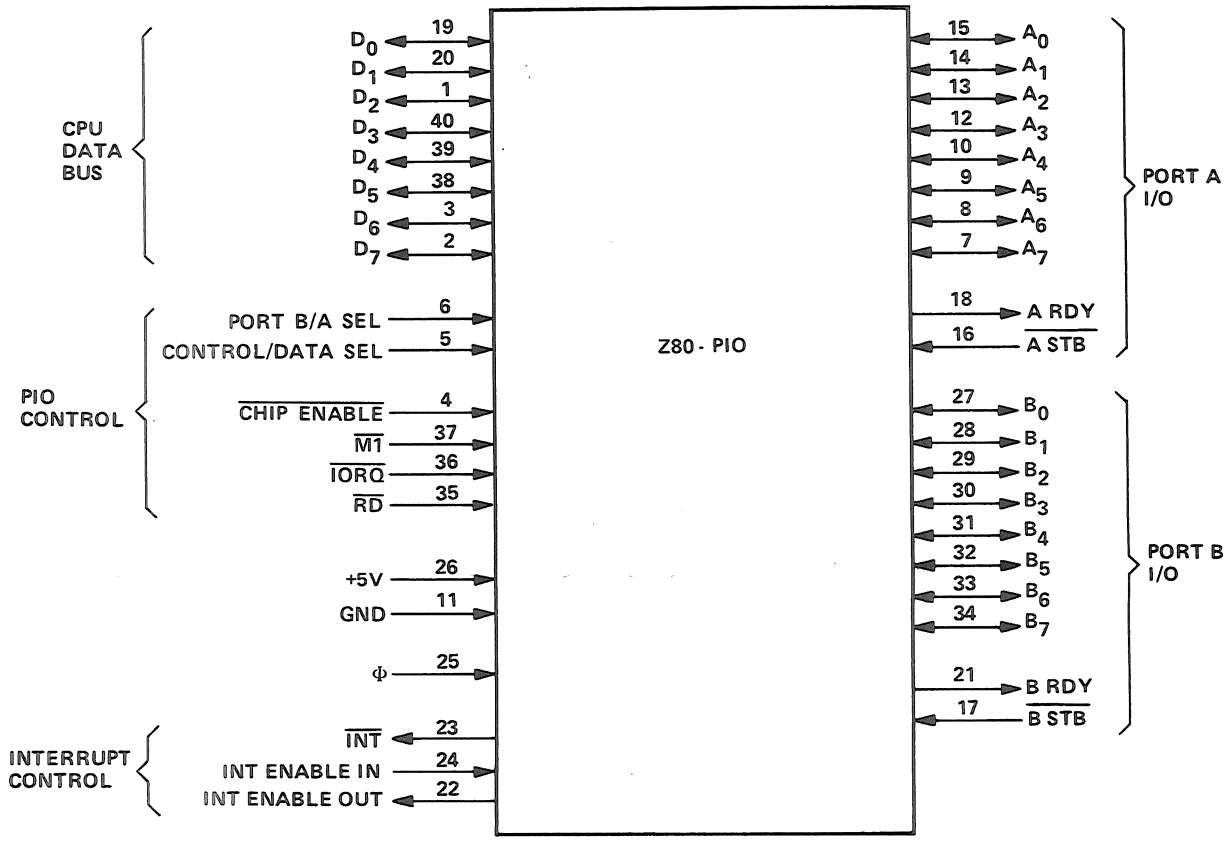


FIGURE 3.0-1
PIO PIN CONFIGURATION

4.0 PROGRAMMING THE PIO

4.1 RESET

The Z80-PIO automatically enters a reset state when power is applied. The reset state performs the following functions:

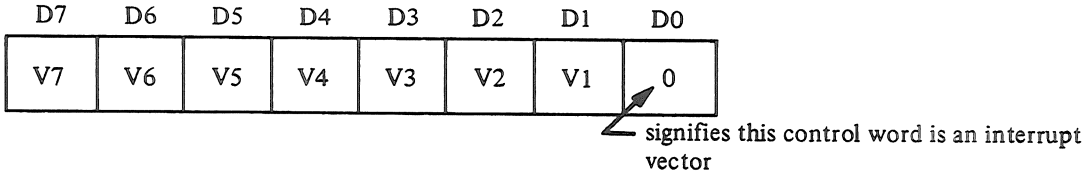
1) Both port mask registers are reset to inhibit all port data bits.
2) Port data bus lines are set to a high impedance state and the Ready "handshake" signals are inactive (low). Mode 1 is automatically selected.
3) The vector address registers are *not* reset.
4) Both port interrupt enable flip flops 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 $\overline{M1}$ signal without the presence of a $\overline{RD}$ or $\overline{IORQ}$ signal. If no $\overline{RD}$ or $\overline{IORQ}$ is detected during $\overline{M1}$ the PIO will enter the reset state immediately after the $\overline{M1}$ 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.

4.2 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 Z-80 data bus during an interrupt acknowledge cycle by the highest priority device requesting service at that time. (Refer to the Z80-CPU Technical Manual 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:

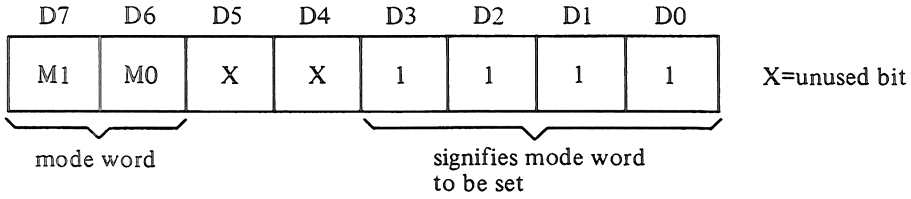


D0 is used in this case as a flag bit which when low causes V7 thru V1 to be loaded into the vector register. At interrupt acknowledge time, the vector of the interrupting port will appear on the Z-80 data bus exactly as shown in the format above.

4.3 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; i.e. 0=Out, 1=In, 2=Bidirectional. Port B can operate in any of these modes except Mode 2.

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



Bits D7 and D6 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".

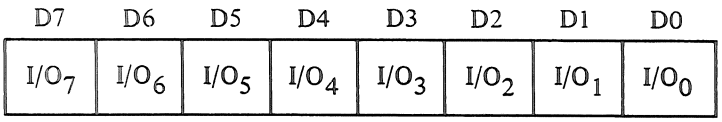
Selecting Mode 0 enables any data written to the port output register by the CPU to be enabled onto the port data bus. The contents of the output register may be changed at any time by the CPU simply by writing a new data word to the port. Also the current contents of the output register may be read 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 rising 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 strobe line. Again, the rising edge of the strobe causes an interrupt request (if 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 A RDY and B RDY 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 $\overline{\text{A STB}}$ 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:



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 strobe 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.

4.4 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 Interrupt	AND/ OR	High/ Low	Masks 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/O 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 D5=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
MB_7	MB_6	MB_5	MB_4	MB_3	MB_2	MB_1	MB_0

Only those port lines whose mask bit is zero will be monitored for generating an interrupt.

8.0 PROGRAMMING SUMMARY

8.1 LOAD INTERRUPT VECTOR

V7	V6	V5	V4	V3	V2	V1	0

8.2 SET MODE

M1	M0	X	X	1	1	1	1

M_1	M_0	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:

I/O_7	I/O_6	I/O_5	I/O_4	I/O_3	I/O_2	I/O_1	I/O_0

I/O = 1 Sets bit to Input

I/O = 0 Sets bit to Output

8.3 SET INTERRUPT CONTROL

Int Enable	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:

MB_7	MB_6	MB_5	MB_4	MB_3	MB_2	MB_1	MB_0

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:

Int Enable	X	X	X	0	0	1	1