

# **APPENDIX F**

Please report errors and omissions to: [max-80@d30.info](mailto:max-80@d30.info)

PDF document created by John Ray - 02/08/2008 - V1R0

# Z8420 Z80<sup>®</sup> PIO Parallel Input/Output Controller



## Product Specification

March 1981

- Features**
- Provides a direct interface between Z-80 microcomputer systems and peripheral devices.
  - Both ports have interrupt-driven handshake for fast response.
  - Four programmable operating modes: byte input, byte output, byte input/output (Port A only), and bit input/output.

- Programmable interrupts on peripheral status conditions.
- Standard Z-80 Family bus-request and prioritized interrupt-request daisy chains implemented without external logic.
- The eight Port B outputs can drive Darlington transistors (1.5 mA at 1.5 V).

**General  
Description**

The Z-80 PIO Parallel I/O Circuit is a programmable, dual-port device that provides a TTL-compatible interface between peripheral devices and the Z-80 CPU. The CPU configures the Z-80 PIO to interface with a wide range of peripheral devices with no other external logic. Typical peripheral devices that are compatible with the Z-80 PIO include most keyboards, paper tape readers and punches, printers, PROM programmers, etc.

One characteristic of the Z-80 peripheral controllers that separates them from other interface controllers is that all data transfer between the peripheral device and the CPU is

accomplished under interrupt control. Thus, the interrupt logic of the PIO permits full use of the efficient interrupt capabilities of the Z-80 CPU during I/O transfers. All logic necessary to implement a fully nested interrupt structure is included in the PIO.

Another feature of the PIO is the ability to interrupt the CPU upon 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 time the processor must spend in polling peripheral status.

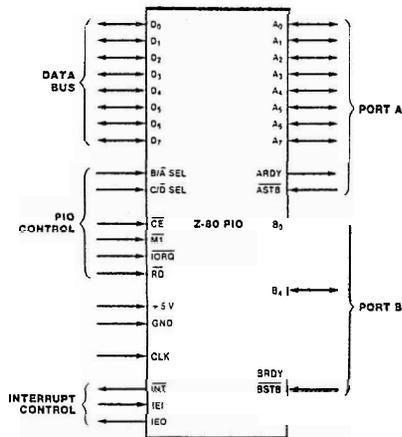


Figure 1. Pin Functions

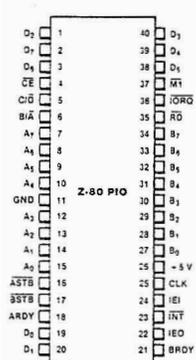


Figure 2. Pin Assignments



## Internal Structure

The internal structure of the Z-80 PIO consists of a Z-80 CPU bus interface, internal control logic, Port A I/O logic, Port B I/O logic, and interrupt control logic (Figure 4). The CPU bus interface logic allows the Z-80 PIO to interface directly to the Z-80 CPU with no other external logic. 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.

**Port Logic.** Each port contains separate input and output registers, handshake control logic, and the control registers shown in Figure 5. All data transfers between the peripheral unit and the CPU use the data input and output registers. The handshake logic associated with each port controls the data transfers through the input and the output registers. The mode control register (two bits) selects one of the four programmable operating modes.

The control mode (Mode 3) uses the remaining registers. The input/output control register specifies which of the eight data bits in the port are to be outputs and enables these bits; the remaining bits are inputs. The mask register and the mask control register control Mode 3 interrupt conditions. The mask register specifies which of the bits in the port are active and which are masked or inactive.

The mask control register specifies two conditions: first, whether the active state of the input bits is High or Low, and second, whether an interrupt is generated when any one unmasked input bit is active (OR condition) or if the interrupt is generated when *all* unmasked input bits are active (AND condition).

**Interrupt Control Logic.** The interrupt control logic section handles all CPU interrupt protocol for nested-priority interrupt structures. Any device's physical location in a daisy-chain configuration determines its priority. Two lines (IEI and IEO) 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 the peripheral requests a new byte transfer. 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 routines completed by the CPU. Higher priority devices may interrupt the servicing of lower priority devices.

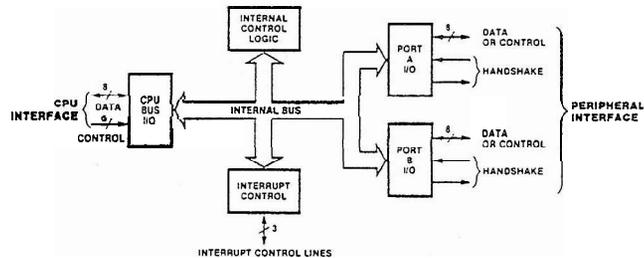


Figure 4. Block Diagram

**Internal Structure**  
(Continued)

If the CPU (in interrupt Mode 2) accepts an interrupt, the interrupting device must provide an 8-bit interrupt vector for the CPU. This vector forms a pointer to a location in 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 0 within the PIO because the pointer must point to two adjacent memory locations for a complete 16-bit address.

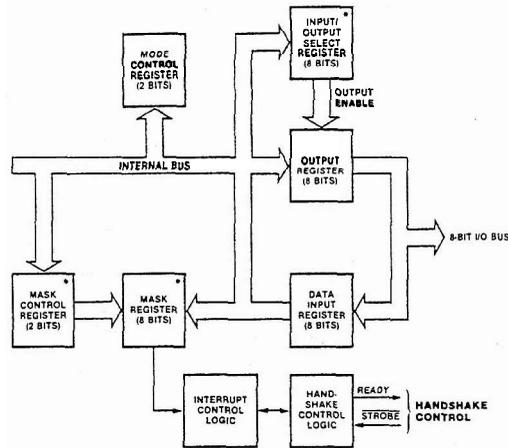
Unlike the other Z-80 peripherals, the PIO does not enable interrupts immediately after programming. It waits until  $\overline{MI}$  goes Low (e.g., during an opcode fetch). This condition is unimportant in the Z-80 environment but might not be if another type of CPU is used.

The PIO decodes the RETI (Return From

Interrupt) instruction directly from the CPU data bus so that each P.I.O. in the system knows at all times whether it is being serviced by the CPU interrupt service routine. No other communication with the CPU is required.

**CPU Bus I/O Logic.** The CPU bus interface logic interfaces the Z-80 PIO directly to the Z-80 CPU, so no external logic is necessary. For large systems, however, address decode and/or buffers may be necessary.

**Internal Control Logic.** This logic receives control words for each port during programming and, in turn, controls the operating functions of the Z-80 PIO. The control logic synchronizes the port operations, controls the  $\overline{CS}$  mode, port addressing, and issues appropriate commands to the ports and the interrupt logic. The Z-80 P does not receive a write input from the CPU; instead, the RD, CE, C.D and IORQ signals generate the write input internally.



\*Used in the bit mode only to allow generation of an interrupt if the peripheral I/O pins go to the specified state.

Figure 5. Typical Port I/O Block Diagram

**Programming Mode 0, 1, or 2.** (*Byte Input, Output, or Bidirectional*). Programming a port for Mode 0, 1, or 2 requires two words per port. These words are:

**A Mode Control Word.** Selects the port operating mode (Figure 6). This word may be written any time.

**An Interrupt Vector.** The Z-80 PIO is designed for use with the Z-80 CPU in interrupt Mode 2 (Figure 7). When interrupts are enabled, the PIO must provide an interrupt vector.

**Mode 3.** (*Bit Input/Output*). Programming a port for Mode 3 operation requires a control word, a vector (if interrupts are enabled), and three additional words, described as follows:

**I/O Register Control.** When Mode 3 is selected, the mode control word must be followed by another control word that sets the I/O control register, which in turn defines which port lines are inputs and which are outputs (Figure 8).

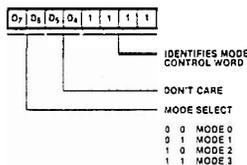


Figure 6. Mode Control Word

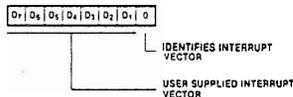


Figure 7. Interrupt Vector Word

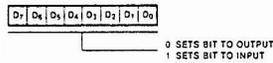
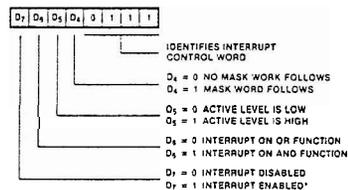


Figure 8. I/O Register Control Word

**Interrupt Control Word.** In Mode 3, handshake is not used. Interrupts are generated as a logic function of the input signal levels. The interrupt control word sets the logic conditions and the logic levels required for generating an interrupt. Two logic conditions or functions are available: AND (if all input bits change to the active level, an interrupt is triggered), and OR (if any one of the input bits changes to the active level, an interrupt is triggered). Bit D<sub>6</sub> sets the logic function, as shown in Figure 9. The active level of the input bits can be set either High or Low. The active level is controlled by Bit D<sub>5</sub>.

**Mask Control Word.** This word sets the mask control register, allowing any unused bits to be masked out. If any bits are to be masked, then D<sub>4</sub> must be set. When D<sub>4</sub> is set the next word written to the port must be a mask control word (Figure 10).

**Interrupt Disable.** There is one other control word which can be used to enable or disable a port interrupt. It can be used without changing the rest of the interrupt control word (Figure 11).



\*NOTE: THE PORT IS NOT ENABLED UNTIL THE INTERRUPT ENABLE IS FOLLOWED BY AN ACTIVE INT.

Figure 9. Interrupt Control Word

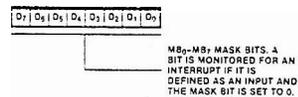


Figure 10. Mask Control Word

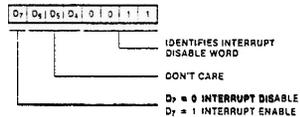


Figure 11. Interrupt Disable Word

Pin Description	
	<p><b>A<sub>0</sub>-A<sub>7</sub>. Port A Bus</b> (bidirectional, 3-state). This 8-bit bus transfers data, status, or control information between Port A of the PIO and a peripheral device. A<sub>0</sub> is the least significant bit of the Port A data bus.</p> <p><b>ARDY. Register A Ready</b> (output, active High). The meaning of this signal depends on the mode of operation selected for Port A as follows:</p> <p><b>Output Mode.</b> 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.</p> <p><b>Input Mode.</b> This signal is active when the Port A input register is empty and ready to accept data from the peripheral device.</p> <p><b>Bidirectional Mode.</b> 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 <math>\overline{ASTB}</math> is active.</p> <p><b>Control Mode.</b> This signal is disabled and forced to a Low state.</p> <p><b><math>\overline{ASTB}</math>. Port A Strobe Pulse From Peripheral Device</b> (input, active Low). The meaning of this signal depends on the mode of operation selected for Port A as follows:</p> <p><b>Output Mode.</b> The positive edge of this strobe is issued by the peripheral to acknowledge the receipt of data made available by the PIO.</p> <p><b>Input Mode.</b> 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.</p> <p><b>Bidirectional Mode.</b> When this signal is active, data from the Port A output register is gated onto the Port A bidirectional data bus. The positive edge of the strobe acknowledges the receipt of the data.</p> <p><b>Control Mode.</b> The strobe is inhibited internally.</p> <p><b>B<sub>0</sub>-B<sub>7</sub>. Port B Bus</b> (bidirectional, 3-state). This 8-bit bus transfers data, status, or control information between Port B and a peripheral device. The Port B data bus can supply 1.5 mA at 1.5 V to drive Darlington transistors. B<sub>0</sub> is the least significant bit of the bus.</p> <p><b>B/<math>\overline{A}</math>. Port B Or A Select</b> (input, High = B). This pin defines which port is accessed during a data transfer between the CPU and the PIO. A Low on this pin selects Port A; a High selects Port B. Often address bit A<sub>0</sub> from the CPU is used for this selection function.</p> <p><b>BRDY. Register B Ready</b> (output, active High). This signal is similar to ARDY, except that 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.</p> <p><b><math>\overline{BSTB}</math>. Port B Strobe Pulse From Peripheral Device</b> (input, active Low). This signal is similar to <math>\overline{ASTB}</math>, except that in the Port A bidirectional mode this signal strobes data from the peripheral device into the Port A input register.</p>
	<p><b>C/<math>\overline{D}</math>. Control Or Data Select</b> (input, High = C). This pin defines the type of data transfer to be performed between the CPU and the PIO. A High on this pin during a CPU write to the PIO causes the Z-80 data bus to be interpreted as a <i>command</i> for the port selected by the B/<math>\overline{A}</math> Select line. A Low on this pin means that the Z-80 data bus is being used to transfer data between the CPU and the PIC. Often address bit A<sub>1</sub> from the CPU is used for this function.</p> <p><b><math>\overline{CE}</math>. Chip Enable</b> (input, active Low). A Low 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 decoded from four I/O port numbers for Port A and B, data, and control.</p> <p><b>CLK. System Clock</b> (input). The Z-80 PIO uses the standard single-phase Z-80 system clock.</p> <p><b>D<sub>0</sub>-D<sub>7</sub>. Z-80 CPU Data Bus</b> (bidirectional, 3-state). This bus is used to transfer all data and commands between the Z-80 CPU and Z-80 PIO. D<sub>0</sub> is the least significant bit.</p> <p><b>IEI. Interrupt Enable In</b> (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 of this pin indicates that no other devices of higher priority are being serviced by a CPU interrupt service routine.</p> <p><b>IEO. Interrupt Enable Out</b> (output, active High). The IEO signal is the other signal required to form a daisy chain priority scheduler. It is High only if IEI is High and the CPU is not servicing an interrupt from this PIO. This signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.</p> <p><b><math>\overline{INT}</math>. Interrupt Request</b> (output, open drain, active Low). When <math>\overline{INT}</math> is active the Z-80 CPU is requesting an interrupt from the Z-80 CPU.</p> <p><b><math>\overline{IORQ}</math>. Input/Output Request</b> (input from Z-80 CPU, active Low). <math>\overline{IORQ}</math> is used in conjunction with B/<math>\overline{A}</math>, C/<math>\overline{D}</math>, <math>\overline{CE}</math>, and <math>\overline{RD}</math> to transfer commands and data between the Z-80 CPU and the Z-80 PIO. When <math>\overline{CE}</math>, <math>\overline{RD}</math>, and <math>\overline{IORQ}</math> are active, the port addressed by B/<math>\overline{A}</math> transfers data to the CPU (a read operation). Conversely, when <math>\overline{CE}</math> and <math>\overline{IORQ}</math> are active but <math>\overline{RD}</math> is not, the port addressed by B/<math>\overline{A}</math> is written into from the CPU with either data or control information, as specified by C/<math>\overline{D}</math>. Also, if <math>\overline{IORQ}</math> and <math>\overline{MI}</math> are active simultaneously, the CPU is acknowledging an interrupt; the interrupting port automatically places its interrupt vector on the CPU data bus if it is the high priority device requesting an interrupt.</p>

**Pin Description**  
(Continued)

**MI.** Machine Cycle (input from CPU, active Low). This signal is used as a sync pulse to control several internal PIO operations. When both the MI and RD signals are active, the Z-80 CPU is fetching an instruction from memory. Conversely, when both MI and IORQ are active, the CPU is acknowledging an interrupt. In addition, MI has two other functions within the Z-80 PIO: it synchronizes

the PIO interrupt logic; when MI occurs without an active RD or IORQ signal, the PIO is reset.

**RD.** Read Cycle Status (input from Z-80 CPU, active Low); if RD is active, or an I/O operation is in progress, RD is used with BA, CD, CE, and IORQ to transfer data from the Z-80 PIO to the Z-80 CPU.

**Timing**

The following timing diagrams show typical timing in a Z-80 CPU environment. For more precise specifications refer to the composite ac timing diagram.

**Write Cycle.** Figure 12 illustrates the timing for programming the Z-80 PIO or for writing data to one of its ports. No Wait states are allowed for writing to the PIO other than the automatically inserted TWA. The PIO does not receive a specific write signal; it internally generates its own from the lack of an active RD signal.

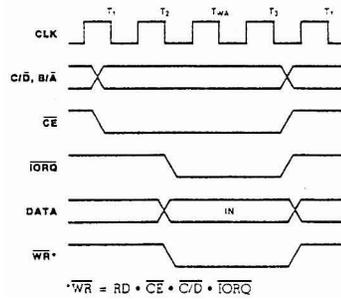


Figure 12. Write Cycle Timing

**Read Cycle.** Figure 13 illustrates the timing for reading the data input from an external device to one of the Z-80 PIO ports. No Wait states are allowed for reading the PIO other than the automatically inserted TWA.

**Output Mode (Mode 0).** An output cycle (Figure 14) is always started by the execution of an output instruction by the CPU. The WR\* pulse from the CPU latches the data from the CPU data bus into the selected port's output register. The WR\* pulse sets the Ready flag after a Low-going edge of CLK, indicating data is available. Ready stays active until the positive edge of the strobe line is received, indicating that data was taken by the peripheral. The positive edge of the strobe pulse generates an INT if the interrupt enable flip-flop has been set and if this device has the highest priority.

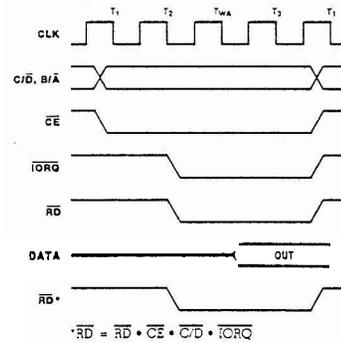


Figure 13. Read Cycle Timing

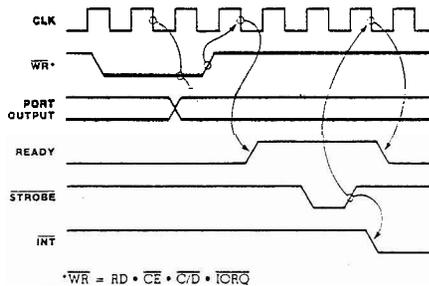


Figure 14. Mode 0 Output Timing

Timing  
(Continued)

**Input Mode (Mode 1).** When **STROBE** goes Low, data is loaded into the selected port input register (Figure 15). The next rising edge of strobe activates **INT**, if Interrupt Enable is set and this is the highest-priority requesting device. The following falling edge of **CLK** resets Ready to an inactive state, indicating

that the input register is full and cannot accept any more data until the CPU completes a read. When a read is complete, the positive edge of **RD** sets Ready at the next Low-going transition of **CLK**. At this time new data can be loaded into the PIO.

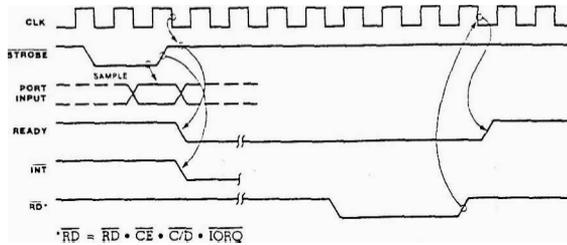


Figure 15. Mode 1 Input Timing

**Bidirectional Mode (Mode 2).** This is a combination of Modes 0 and 1 using all four handshake lines and the eight Port A I/O lines (Figure 16). Port B must be set to the bit mode and its inputs must be masked. The Port A handshake lines are used for output control and the Port B lines are used for input control.

If interrupts occur, Port A's vector will be during port output and Port B's will be during port input. Data is allowed out on Port A bus only when **ASTB** is Low. The edge of this strobe can be used to latch data into the peripheral.

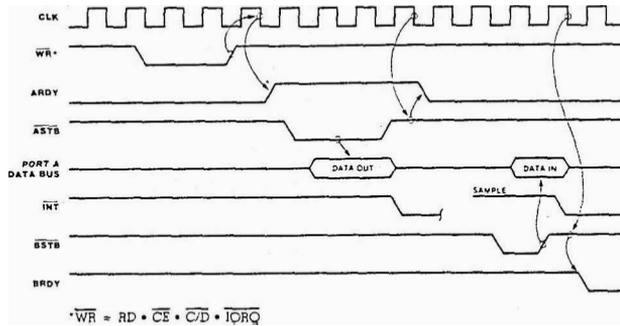


Figure 16. Mode 2 Bidirectional Timing

**Timing**  
(Continued)

**Bit Mode (Mode 3).** The bit mode does not utilize the handshake signals, and a normal port write or port read can be executed at any time. When writing, the data is latched into the output registers with the same timing as the output mode (Figure 17).

When reading the PIO, the data returned to the CPU is 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 contains data that was present immediately prior to the falling edge of  $\overline{RD}$ . An interrupt is generated if interrupts from the port are enabled and the data on the port data lines satisfy the logical equation defined by the 8-bit mask and 2-bit mask control registers. However, if Port A is programmed in bidirectional mode, Port B does not issue an interrupt in bit mode and must therefore be poiled.

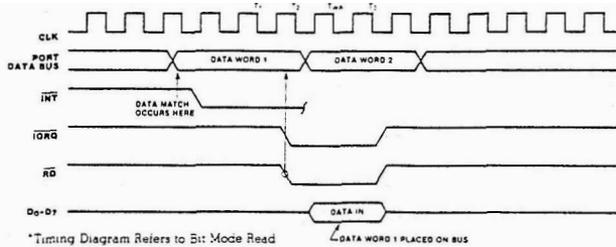


Figure 17. Mode 3 Bit Mode Timing

**Interrupt Acknowledge Timing.** During  $\overline{M1}$  time, peripheral controllers are inhibited from changing their interrupt enable status, permitting the Interrupt Enable signal to ripple through the daisy chain. The peripheral with IEI High and IEO Low during  $\overline{INTACK}$  places a preprogrammed 8-bit interrupt vector on the data bus at this time (Figure 18). IEO is held Low until a Return From Interrupt (RETI) instruction is executed by the CPU while IEI is High. The 2-byte RETI instruction is decoded internally by the PIO for this purpose.

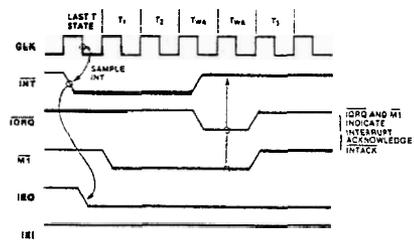


Figure 18. Interrupt Acknowledge Timing

**Return From Interrupt Cycle.** If a Z-80 peripheral has no interrupt pending and is not under service, then its  $\overline{IEO} = \overline{IEI}$ . If it has an interrupt under service (i.e., it has already interrupted and received an interrupt acknowledge) then its  $\overline{IEO}$  is always Low, inhibiting lower priority devices from interrupting. If it has an interrupt pending which has not yet been acknowledged,  $\overline{IEO}$  is Low unless an "ED" is decoded as the first byte of a 2-byte opcode (Figure 19). In this case,  $\overline{IEO}$  goes High until the next opcode byte is decoded, whereupon it goes Low again. If the second byte of the opcode was a "4D," then the opcode was an RETI instruction.

After an "ED" opcode is decoded, only the peripheral device which has interrupted and is currently under service has its  $\overline{IEI}$  High and its

$\overline{IEO}$  Low. This device is the highest-priority device in the daisy chain that has received an interrupt acknowledge. All other peripherals have  $\overline{IEI} = \overline{IEO}$ . If the next opcode byte decoded is "4D," this peripheral resets its "interrupt under service" condition.

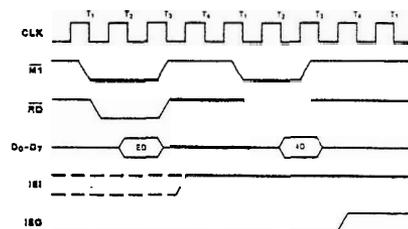
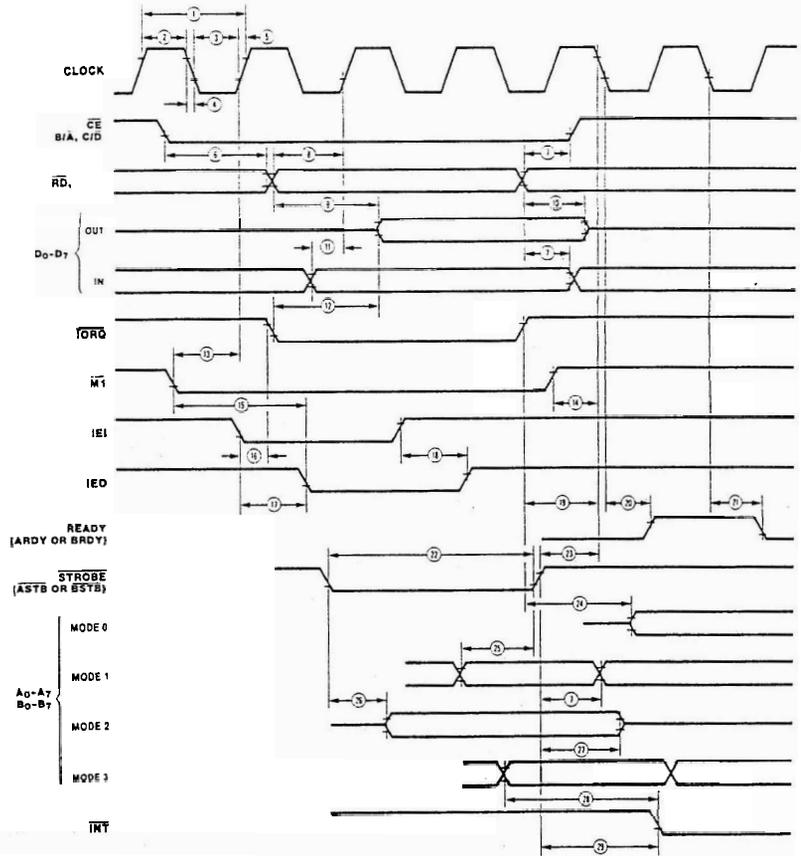


Figure 19. Return From Interrupt

**AC  
Charac-  
teristics**



Number	Symbol	Parameter	Z-80 PIO		Z-80A PIO		Z-80B PIO <sup>[9]</sup>		Comment
			Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)	
1	TcC	Clock Cycle Time	400	[1]	250	[1]	165	[1]	
2	TwCh	Clock Width (High)	170	2000	105	2000	65	2000	
3	TwCl	Clock Width (Low)	170	2000	105	2000	65	2000	
4	TfC	Clock Fall Time		30		30		20	
5	TrC	Clock Rise Time		30		30		20	
6	TsCS(RI)	$\overline{CE}$ , $B/\overline{A}$ , $C/\overline{D}$ to $\overline{RD}$ , $\overline{IORQ}$ ↑ Setup Time	50		50		50		[6]
7	Th	Any Hold Times for Specified Setup Time	0		0		0	0	
8	TsRI(C)	$\overline{RD}$ , $\overline{IORQ}$ to Clock ↑ Setup Time	115		115		70		
9	TdRI(DO)	$\overline{RD}$ , $\overline{IORQ}$ ↓ to Data Out Delay	430		380		300		[2]
10	TdRI(DOs)	$\overline{RD}$ , $\overline{IORQ}$ ↓ to Data Out Float Delay		160		110		70	
11	TsDI(C)	Data In to Clock ↑ Setup Time	50		50		40		CL = 50 pF
12	TdIO(DOI)	$\overline{IORQ}$ ↓ to Data Out Delay (INTACK Cycle)	340		160		120		[3]
13	TsMI(Cr)	$\overline{MI}$ ↓ to Clock ↑ Setup Time	210		90		70		
14	TsMI(Cl)	$\overline{MI}$ ↓ to Clock ↑ Setup Time (MI Cycle)	0		0		0		[8]
15	TdMI(IEO)	$\overline{MI}$ ↓ to IEO ↓ Delay (Interrupt Immediately Preceding $\overline{MI}$ ↓)	300		190		100		[5, 7]
16	TsIEI(IO)	IEI to $\overline{IORQ}$ ↓ Setup Time (INTACK Cycle)	140		140		100		[7]
17	TdIEI(IEOf)	IEI ↓ to IEO ↓ Delay	190		130		120		[5] CL = 50 pF
18	TdIEI(IEOr)	IEI ↓ to IEO ↓ Delay (after ED Decode)	210		160		160		[5]
19	TcIO(C)	$\overline{IORQ}$ ↓ to Clock ↑ Setup Time (To Activate READY on Next Clock Cycle)	220		200		170		
20	TdC(RDYr)	Clock ↓ to READY ↓ Delay	200		190		170		[5] CL = 50 pF
21	TdC(RDYf)	Clock ↓ to READY ↓ Delay	150		140		120		[5]
22	TwSTB	STROBE Pulse Width	150		150		120		[4]
23	TsSTB(C)	STROBE ↑ to Clock ↑ Setup Time (To Activate READY on Next Clock Cycle)	220		220		150		[5]
24	TdIO(PD)	$\overline{IORQ}$ ↓ to PORT DATA Stable Delay (Mode 0)	200		180		160		[5]
25	TsPD(STB)	PORT DATA to STROBE ↑ Setup Time (Mode 1)	260		230		190		
26	TdSTB(PD)	STROBE ↓ to PORT DATA Stable (Mode 2)	230		210		180		[5]
27	TdSTB(PDr)	STROBE ↓ to PORT DATA Float Delay (Mode 2)	200		180		160		CL = 50 pF
28	TdPD(INT)	PORT DATA Match to $\overline{INT}$ ↓ Delay (Mode 3)	540		490		430		
29	TdSTB(INT)	STROBE ↓ to $\overline{INT}$ ↓ Delay	490		440		350		

NOTES:

- [1]  $TcC = TwCh + TwCl + TrC + TfC$ .
- [2] Increase TdRI(DO) by 10 ns for each 50 pF increase in load up to 200 pF max.
- [3] Increase TdIO(DOI) by 10 ns for each 50 pF increase in loading up to 200 pF max.
- [4] For Mode 2:  $TwSTB > TsPD(STB)$ .
- [5] Increase these values by 2 ns for each 10 pF increase in loading up to 100 pF max.

- [6] TsCS(RI) may be reduced. However, the time subtracted from TsCS(RI) will be added to TdRI(DO).
- [7]  $2.5 TcC > (N-2)TdIEI(IECh) - TdMI(IEO) - TsIEI(IO) - TTL$  Buffer Delay, if any.
- [8]  $\overline{MI}$  must be active for a minimum of two clock cycles to reset the PIO.
- [9] Z80B PIO numbers are preliminary and subject to change.

**Absolute Maximum Ratings**  
 Voltages on all inputs and outputs with respect to GND . . . . . -0.3 V to +7.0 V  
 Operating Ambient Temperature . . . . . As Specified in Ordering Information  
 Storage Temperature . . . . . -65°C to +150°C

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

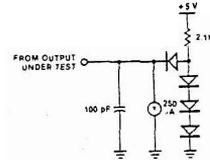
**Test Conditions**  
 The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0 V). Positive current flows into the referenced pin. Available operating temperature ranges are:

- 0° to +70°C,  
+4.75 V ≤ V<sub>CC</sub> ≤ +5.25 V
- -40°C to +85°C,  
+4.75 V ≤ V<sub>CC</sub> ≤ +5.25 V
- -55° to +125°C,  
+4.75 V ≤ V<sub>CC</sub> ≤ +5.5 V

The product number for each operating temperature range may be found in the

Ordering Information section.

All ac parameters assume a load capacitance of 100 pF max. Timing references between two output signals assume a load difference of 50 pF max.



DC Characteristics	Symbol	Parameter	Min	Max	Unit	Test Cond
	V <sub>ILC</sub>	Clock Input Low Voltage	-0.3	+0.45	V	
	V <sub>IHC</sub>	Clock Input High Voltage	V <sub>CC</sub> -0.6	+5.5	V	
	V <sub>IL</sub>	Input Low Voltage	-0.3	+0.8	V	
	V <sub>IH</sub>	Input High Voltage	+2.0	+5.5	V	
	V <sub>OL</sub>	Output Low Voltage		+0.4	V	I <sub>OL</sub> = 2.0
	V <sub>OH</sub>	Output High Voltage	+2.4		V	I <sub>OH</sub> = -2
	I <sub>I1</sub>	Input Leakage Current	-10.0	+10.0	μA	0 < V <sub>IN</sub> < 1
	I <sub>Z</sub>	3-State Output/Data Bus Input Leakage Current	-10.0	+10.0	μA	0 < V <sub>IN</sub> < 1
	I <sub>CC</sub>	Power Supply Current		100.0	mA	V <sub>OH</sub> = 1.5
	I <sub>OHD</sub>	Darlington Drive Current	-1.5	3.8	mA	R <sub>EXT</sub> = 3

Over specified temperature and voltage range.

Capacitance	Symbol	Parameter	Min	Max	Unit	Test Con
	C	Clock Capacitance		10	pF	Unmeas-
	C <sub>IN</sub>	Input Capacitance		5	pF	pins retu-
	C <sub>OUT</sub>	Output Capacitance		10	pF	to ground

Over specified temperature range; f = 1MHz

Ordering Information								
	Product Number	Package/Temp	Speed	Description	Product Number	Package/Temp	Speed	Description
	Z8420	CE	2.5 MHz	Z80 PIO (40-pin)	Z8420A	DE	4.0 MHz	Z80A PIO (40-pin)
	Z8420	CM	2.5 MHz	Same as above	Z8420A	DS	4.0 MHz	Same as above
	Z8420	CMB	2.5 MHz	Same as above	Z8420A	PE	4.0 MHz	Same as above
	Z8420	CS	2.5 MHz	Same as above	Z8420A	PS	4.0 MHz	Same as above
	Z8420	DE	2.5 MHz	Same as above	Z8420B	CE	6.0 MHz	Z80B PIO (40-pin)
	Z8420	DS	2.5 MHz	Same as above	Z8420B	CM	6.0 MHz	Same as above
	Z8420	PE	4.0 MHz	Same as above	Z8420B	CMB	6.0 MHz	Same as above
	Z8420	PS	4.0 MHz	Same as above	Z8420B	CS	6.0 MHz	Same as above
	Z8420A	CE	4.0 MHz	Z80A PIO (40-pin)	Z8420B	DE	6.0 MHz	Same as above
	Z8420A	CM	4.0 MHz	Same as above	Z8420B	DS	6.0 MHz	Same as above
	Z8420A	CMB	4.0 MHz	Same as above	Z8420B	PE	6.0 MHz	Same as above
	Z8420A	CS	4.0 MHz	Same as above	Z8420B	PS	6.0 MHz	Same as above

NOTES: C = Ceramic, D = Cerdip, P = Plastic; E = -40°C to +85°C, M = -55°C to +125°C, MB = 55°C to +125°C with MIL-STD-883 Class B processing, S = 0°C to +70°C.

Z80 PIO

