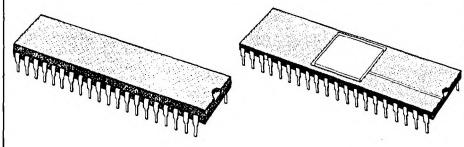
SGS-THOMSON MICROELECTRONICS

Z84C20

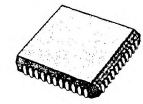
Z80C PIO CMOS VERSION

- PROVIDES A DIRECT INTERFACE BETWEEN Z80 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 Z80 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)
- SINGLE 5 V ± 10 % POWER SUPPLY
- LOW POWER CONSUMPTION :
  - 2 mA typ. at 4 MHz
  - 3 mA typ. at 6 MHz
  - less than 10 µA in Power Down mode
- EXTENDED OPERATING TEMPERATURE :
  - 40 °C TO + 85 ° C



B
DIP-40
(Plastic)

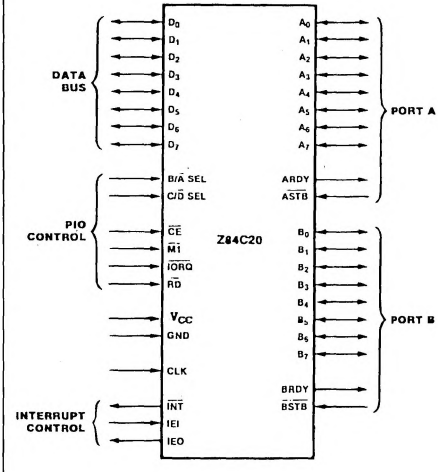
D
DIP-40
(Ceramic)



C
PLCC44
(Plastic)

(Ordering Information at the end of the datasheet)

LOGIC FUNCTIONS



DESCRIPTION

The Z80C PIO Parallel I/O Circuit is a programmable, dual-port device that provides a TTL-compatible interface between peripheral devices and the CPU. The CPU configures the PIO to interface with a wide range of peripheral devices with no other external logic. Typical peripheral devices that are compatible with the PIO include most keyboards, paper tape readers and punches, printers, PROM programmers, etc.

One characteristic of the Z80 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 CPU during I/O transfers. All logic necessary to implement a fully nested interrupt structure is included in the PIO.

Figure 1 : Dual in Line Pin Configuration.

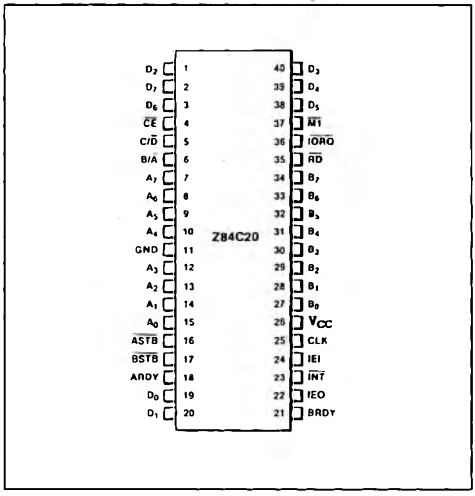
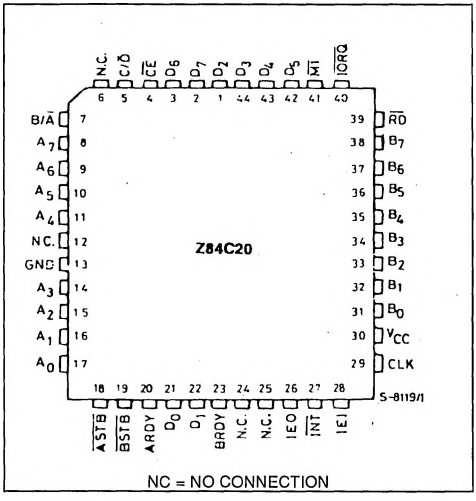


Figure 2 : Chip Carrier Pin Configuration.



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.

The Z80C PIO interfaces to peripherals via two independent general-purpose I/O ports, designated Port A and Port B. Each port has eight data bits and two handshake signals, Ready and Strobe, which control data transfer. The Ready output indicates to the peripheral that the port is ready for a data transfer. Strobe is an input from the peripheral that indicates when a data transfer has occured.

OPERATING MODES

The PIO ports can be programmed to operate in four modes : byte output (Mode 0), byte input (Mode 1), byte input/output (Mode 2) and bit input/output (Mode 3).

In Mode 0, either Port A or Port B can be programmed to output data. Both ports have output registers that are individually addressed by the CPU ; data can be written to either port at any time. When data is written to a port, an active Ready output indicates to the external device that data is available at the associated port and is ready for transfer to the external device. After the data transfer, the external device responds with an active Strobe input, which generates an interrupt, if enabled.

In Mode 1, either Port A or Port B can be configured in the input mode. Each port has an input register addressed by the CPU. When the CPU reads data from a port, the PIO sets the Ready signal, which is detected by the external device. The external device then places data on the I/O lines and strobes the I/O port, which latches the data into the Port Input Register, resets Ready, and triggers the Interrupt Request, if enabled. The CPU can read the input data at any time, which again sets Ready.

Mode 2 is bidirectional and uses Port A, plus the interrupts and handshake signals from both ports. Port B must be set to Mode 3 and masked off. In operation, Port A is used for both data input and output. Output operation is similar to Mode 0 except that data is allowed out onto the Port A bus only when ASTB is Low. For input, operation is similar to Mode 1, except that the data input uses the Port B handshake signals and the Port B interrupt (if enabled).

Both ports can be used in Mode 3. In this mode, the individual bits are defined as either input or output bits. This provides up to eight separate, individually defined bits for each port. During operation, Ready and Strobe are not used. Instead, an interrupt is generated if the condition of one input changes, or if all inputs change. The requirements for generating an interrupt are defined during the programming operation ; the active level is specified as either High or Low, and the logic condition is specified as either one input active (OR) or all inputs active (AND). For example, if the port is programmed for active Low inputs and the logic function is AND, then all inputs at the specified port must go Low to generate an interrupt.

Data outputs are controlled by the CPU and can be written or changed at any time.

- Individual bits can be masked off
- The handshake signals are not used in Mode 3, Ready is held Low, and Strobe is disabled
- When using PIO interrupts, the CPU interrupt mode must be set to Mode 2.

INTERNAL STRUCTURE

The internal structure of the Z80C PIO consists of a CPU bus interface, internal control logic, Port A I/O logic, Port B I/O logic, and interrupt control logic (figure 3). The CPU bus interface logic allows the PIO to interface directly to the 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 4. 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.

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

Figure 3 : Block Diagram.

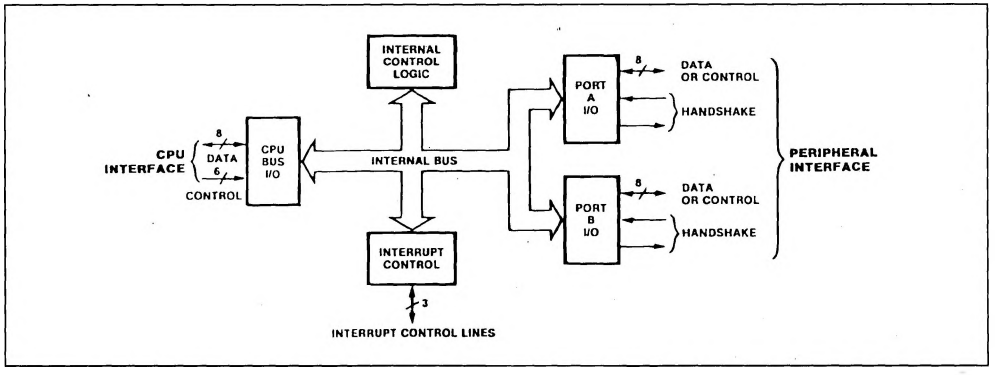
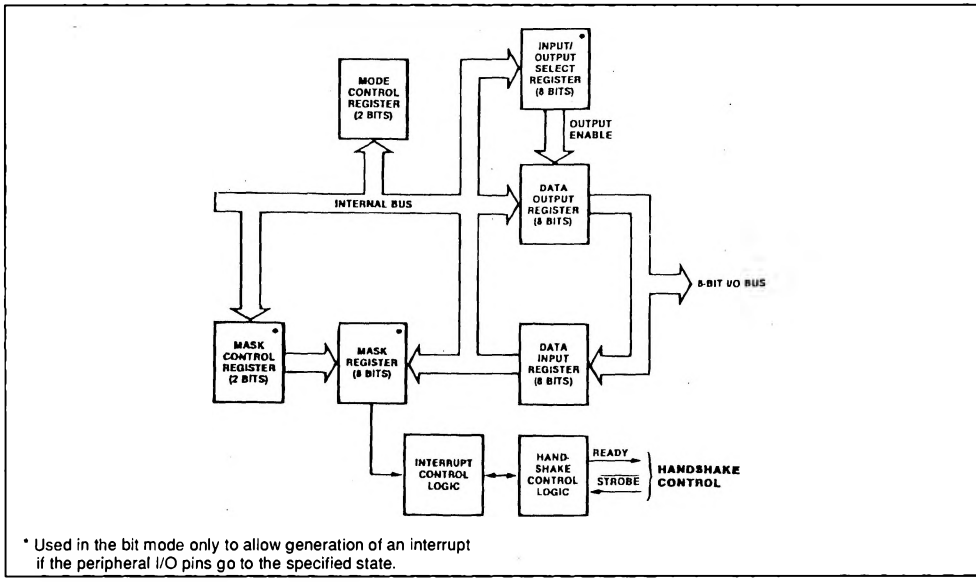


Figure 4 : Typical Port I/O Block Diagram.



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

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 Z80C peripherals, the PIO does not enable interrupts immediately after programming. It waits until $\overline{M1}$ goes Low (e.g., during an opcode fetch). This condition is unimportant in the Z80C 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 PIO 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 PIO directly to the CPU so no external logic is necessary. For large systems, however, address decoders and/or buffers may be necessary.

INTERNAL CONTROL LOGIC

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

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 5). This word may be written any time.

AN INTERRUPT VECTOR. The Z80C PIO is designed for use with the Z80C CPU in interrupt Mode 2 (figure 6). 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

Figure 5 : Mode Control Word.

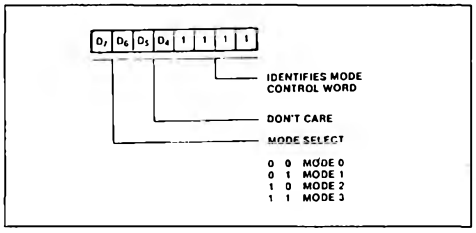


Figure 6 : Interrupt Vector Word.

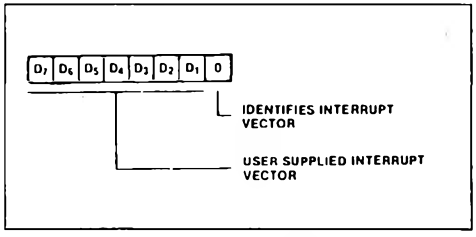


Figure 7 : I/O Register Control Word.

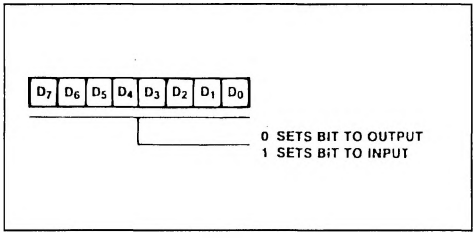
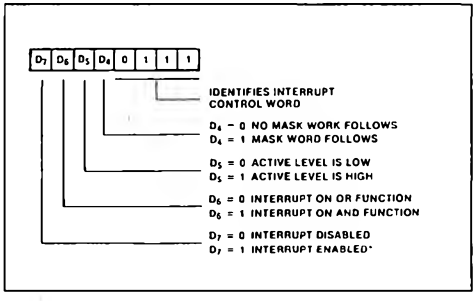


Figure 8 : Interrupt Control Word.



another control word that sets the I/O control register, which in turn defines which port lines are inputs and which are outputs (figure 7).

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_6 sets the logic function, as shown in figure 8. The active level of the input bits can be set either High or Low. The active level is controlled by Bit D_5.

MASK CONTROL WORD. This word sets the mask control register, allowing any unused bits to be masked off. If any bits are to be masked, then D_4 must be set. When D_4 is set, the next word written to the port must be a mask control word (figure 9).

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 10).

Figure 9 : Mask Control Word.

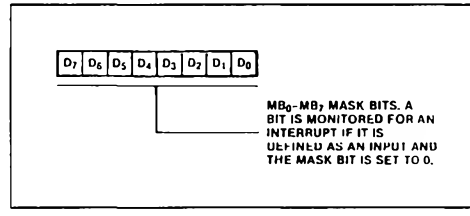
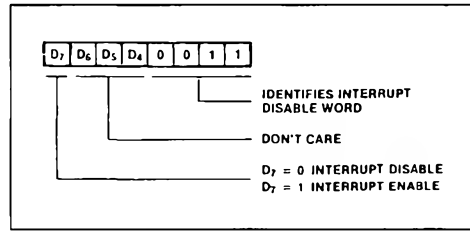


Figure 10 : Interrupt Disable Word.



PIN DESCRIPTIONS

A_0-A_7. *Port A Bus* (Bidirectional, 3-state). This 8-bit bus transfers data, status, or control information between Port A of the PIO and a peripheral device. A_0 is the least significant bit of the Port A data bus.

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

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.

INPUT MODE. This signal is active when the Port A input register is empty and ready to accept data from the peripheral device.

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 ASTB is active.

CONTROL MODE. This signal is disabled and forced to a Low state.

$\overline{\text{ASTB}}$. *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 :

OUTPUT MODE. The positive edge of this strobe is issued by the peripheral to acknowledge the receipt of data made available by the PIO.

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.

BIDIRECTIONAL MODE. 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.

CONTROL MODE. The strobe is inhibited internally.

B_0-B_7. *Port B Bus* (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 1.5mA at 1.5V to drive Darlington transistors. B_0 is the least significant bit of the bus.

B/$\overline{\text{A}}$. *Port B Or A Select* (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_0 from the CPU is used for this selection function.

BRDY. *Register B Ready* (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.

$\overline{\text{BSTB}}$. *Port B Strobe Pulse From Peripheral Device* (Input, Active Low). This signal is similar to ASTB, except that in the Port A bidirectional mode this signal strobes data from the peripheral device into the Port A input register.

C/$\overline{\text{D}}$. *Control Or Data Select* (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 data bus to be interpreted as a command for the port selected by the B/$\overline{\text{A}}$ Select line. A Low on this pin means that the data bus is being used to transfer data between the CPU and the PIO. Often address bit A_1 from the CPU is used for this function.

$\overline{\text{CE}}$. *Chip Enable* (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 Ports A and B, data, and control.

CLK. *System Clock* (Input). The Z80C PIO uses the standard single-phase Z80C system clock.

D_0-D_7. *CPU Data Bus* (Bidirectional, 3-state). This bus is used to transfer all data and commands between the CPU and the PIO. D_0 is the least significant bit.

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 PIO is requesting an interrupt from the CPU.

$\overline{\text{IORQ}}$. *Input/Output Request* (Input from CPU, Active Low). $\overline{\text{IORQ}}$ is used in conjunction with B/$\overline{\text{A}}$, C/$\overline{\text{D}}$, $\overline{\text{CE}}$, and $\overline{\text{RD}}$ to transfer commands and data be-

tween the CPU and the PIO. When $\overline{CE}$, $\overline{RD}$, and IORQ are active, the port addressed by B/A transfers data to the CPU (a read operation). Conversely, when $\overline{CE}$ and IORQ are active but $\overline{RD}$ is not, the port addressed by B/A is written into from the CPU with either data or control information, as specified by C/D. Also, if IORQ and $\overline{M1}$ 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 highest priority device requesting an interrupt.

$\overline{M1}$. *Machine Cycle* (Input from CPU, Active Low). This signal is used as a sync pulse to control several internal PIO operations. When both the $\overline{M1}$ and $\overline{RD}$ signals are active, the CPU is fetching an instruction from memory. Conversely, when both $\overline{M1}$ and IORQ are active, the CPU is acknowledging an interrupt. In addition, $\overline{M1}$ has two other functions within the PIO : it synchronizes the PIO interrupt logic ; when $\overline{M1}$ occurs without an active $\overline{RD}$ or IORQ signal, the PIO is reset.

$\overline{RD}$. *Read Cycle Status* (Input from CPU, Active Low). If $\overline{RD}$ is active, or an I/O operation is in progress, $\overline{RD}$ is used with B/A, C/D, $\overline{CE}$, and IORQ to transfer data from the PIO to the CPU.

TIMING

The following timing diagrams show typical timing in a Z80C CPU environment. For more precise specifications refer to the composite ac timing diagram.

WRITE CYCLE

Figure 11 illustrates the timing for programming the Z80C 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 T_{WA}. The PIO does not receive a specific write signal ; it internally generates its own from the lack of an active $\overline{RD}$ signal.

READ CYCLE

Figure 12 illustrates the timing for reading the data input from an external device to one of the PIO ports. No Wait states are allowed for reading the PIO other than the automatically inserted T_{WA}.

OUTPUT MODE (MODE 0)

An output cycle (figure 13) is always started by the execution of an output instruction by the CPU. The $\overline{WR}$* pulse from the CPU latches the data from the CPU data bus into the selected port's output register. The $\overline{WR}$* 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.

INPUT MODE (MODE 1)

When $\overline{STROBE}$ goes Low, data is loaded into the selected port input register (figure 14). 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 $\overline{RD}$ sets Ready at the next Low-going transition of CLK. At this time new data can be loaded into the PIO.

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 15). 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 used during port output and Port B's will be used during port input. Data is allowed out onto the Port A bus only when $\overline{ASTB}$ is Low. The rising edge of this strobe can be used to latch the data into the peripheral.

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 16).

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 polled.

Figure 11 : Write Cycle Timing.

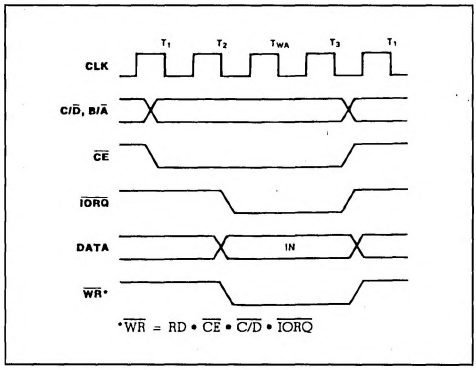


Figure 12 : Read Cycle Timing.

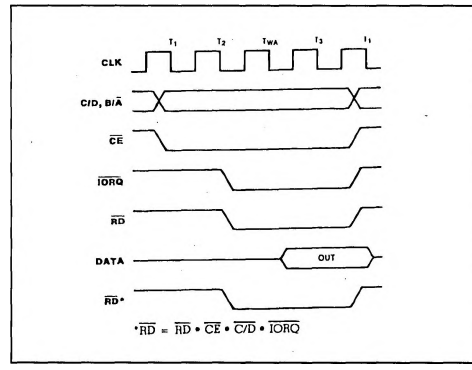
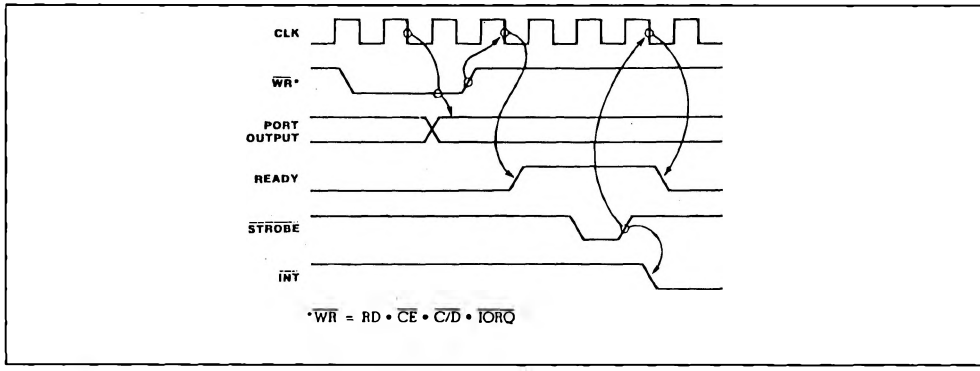


Figure 13 : Mode 0 Output Timing.



INTERRUPT ACKNOWLEDGE TIMING

During 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 INTACK places a preprogrammed 8-bit interrupt vector on the data bus at this time (figure 17). 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.

RETURN FROM INTERRUPT CYCLE

If a Z80C peripheral has no interrupt pending and is not under service, then its IEO = IEI. If it has an interrupt under service (i.e., it has already interrupted and received an interrupt acknowledge) then its IEO is always Low, inhibiting lower priority devices from interrupting. If it has an interrupt pending which has not yet been acknowledged, IEO is Low unless an "ED" is decoded as the first byte of a 2-byte opcode (figure 18). In this case, 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 IEI High and its IEO Low. This device is the highest-priority device in the daisy chain that has received an interrupt acknowledge. All other peripherals have IEI = IEO. If the next opcode byte decoded is "4D", this peripheral device resets its "interrupt under service" condition.

Figure 14 : Mode 1 Input Timing.

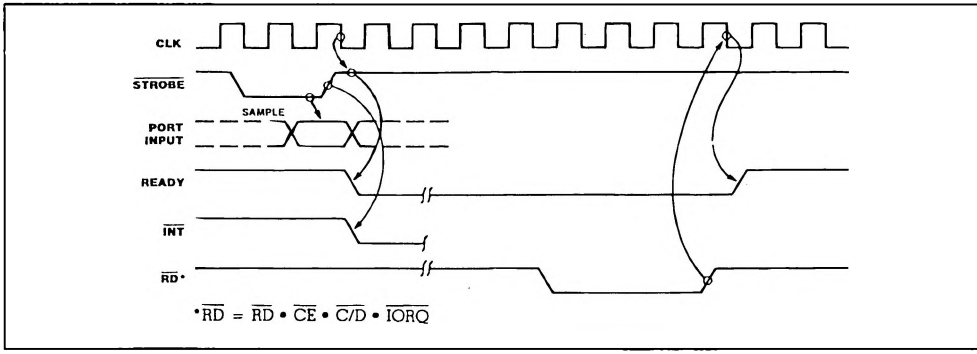


Figure 15 : Mode 2 Bidirectional Timing.

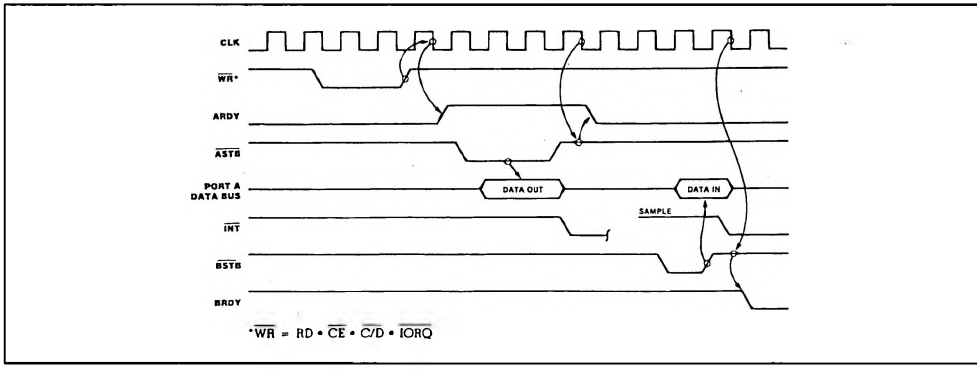
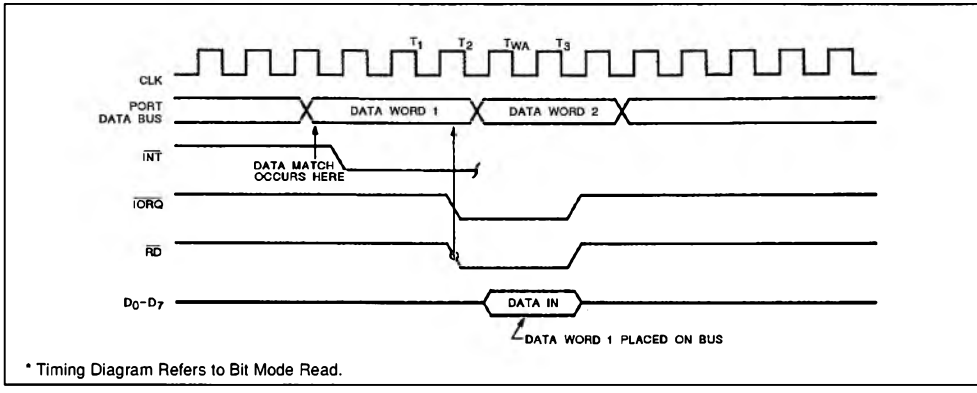


Figure 16 : Mode 3 Bit Mode Timing.



* Timing Diagram Refers to Bit Mode Read.

Figure 17 : Interrupt Acknowledge Timing.

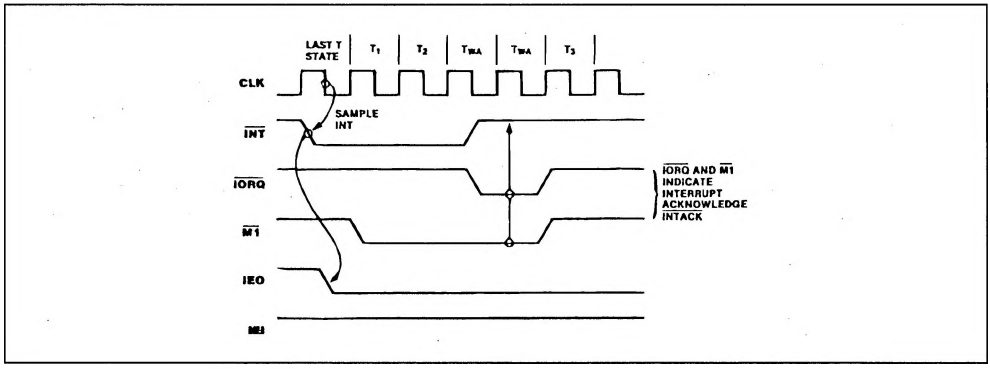
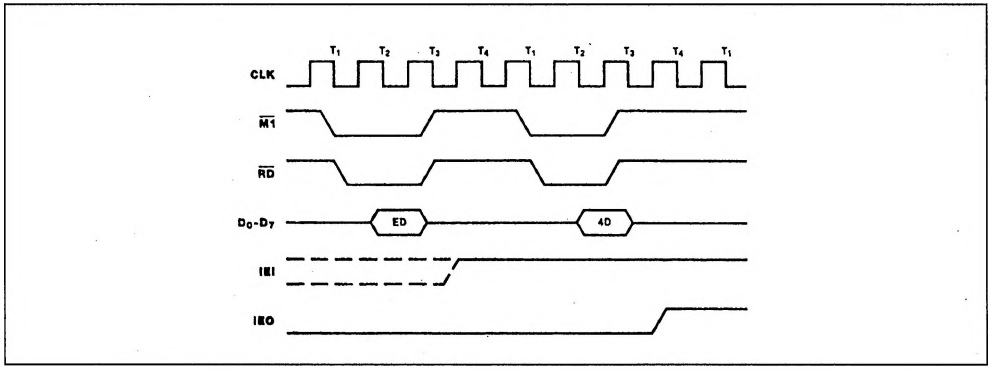
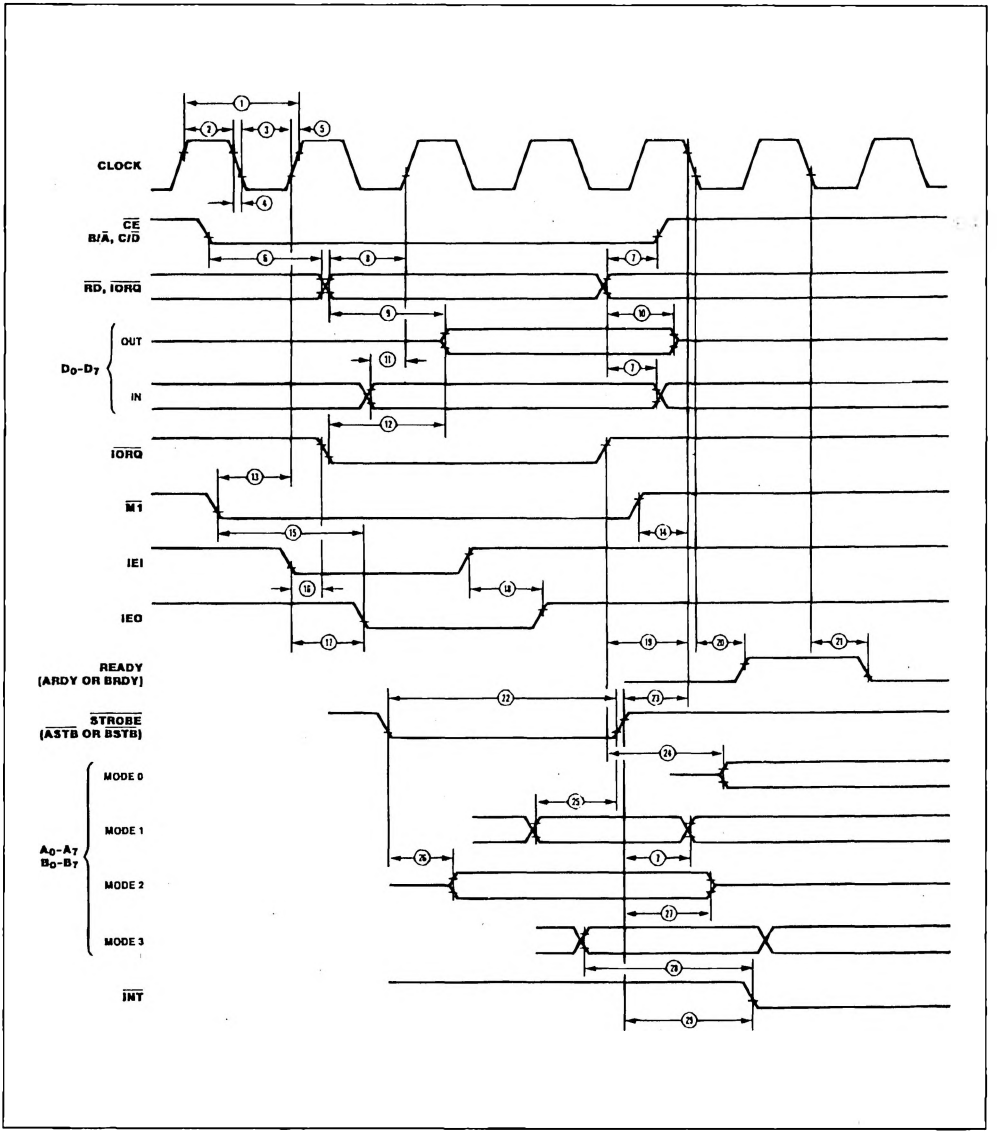


Figure 18 : Return From Interrupt.



AC CHARACTERISTICS



AC CHARACTERISTICS (continued)

N°	Symbol	Parameter	Z84C20A		Z84C20B	
			Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)
1	TcC	Clock Cycle Time	250		165	
2	TwCh	Clock Width (high)	105		65	
3	TwCl	Clock Width (low)	105		65	
4	TfC	Clock Fall Time		30		20
5	TrC	Clock Rise Time		30		20
6	TsCS(Rl)	$\overline{CE}$, B/$\overline{A}$, C/$\overline{D}$ to $\overline{RD}$, $\overline{IORQ}$ ↓ Setup Time	50		50	
7	Th	Any Hold Times for Specified Setup Time	40		40	
8	TsRI(C)	$\overline{RD}$, $\overline{IORQ}$ to Clock ↑ Setup Time	115		70	
9	TdRI(DO)	$\overline{RD}$, $\overline{IORQ}$ ↓ to Data Out Delay		380		300
10	TdRI(DOs)	$\overline{RD}$, $\overline{IORQ}$ ↑ to Data Out Float Delay		110		70
11	TsDI(C)	Data In to Clock ↑ Setup Time	50		40	
12	TdIO(DOI)	$\overline{IORQ}$ ↓ to Data Out Delay (INTACK Cycle)		160		120
13	TsMI(Cr)	$\overline{MI}$ ↓ to Clock ↑ Setup Time	90		70	
14	TsMI(Cf)	$\overline{MI}$ ↑ to Clock ↓ Setup Time ($\overline{MI}$ Cycle)	0		0	
15	TdMI(IEO)	$\overline{MI}$ ↓ to IEO ↓ Delay (interrupt immediately preceding $\overline{MI}$ ↓)		190		100
16	TsIEI(IO)	IEI ↓ to $\overline{IORQ}$ ↓ Setup Time (INTACK Cycle)	140		100	
17	TdIEI(IEOf)	IEI ↓ to IEO ↓ Delay		130		120
18	TdIEI(IEOr)	IEI ↑ to IEO ↑ Delay (after ED decode)		160		150
19	TcIO(C)	$\overline{IORQ}$ ↑ to Clock ↓ Setup Time (to activate READY on Next Clock Cycle)	200		170	
20	TdC(RDYr)	Clock ↓ to READY ↑ Delay		190		170
21	TdC(RDYf)	Clock ↓ to READY ↓ Delay		140		120
22	TwSTB	$\overline{STROBE}$ Pulse Width	150		120	
23	TsSTB(C)	$\overline{STROBE}$ ↑ to Clock ↓ Setup Time (to activate READY on Next Clock Cycle)	220		150	
24	TdIO(PD)	$\overline{IORQ}$ ↑ to PORT DATA Stable Delay (Mode 0)		180		160
25	TsPD(STB)	PORT DATA to $\overline{STROBE}$ ↑ Setup Time (Mode 1)	230		190	
26	TdSTB(PD)	$\overline{STROBE}$ ↓ to PORT DATA Stable (Mode 2)		210		180
27	TdSTB(PDr)	$\overline{STROBE}$ ↑ to PORT DATA Float Delay (Mode 2)		180		160
28	TdPD(INT)	PORT DATA Match to $\overline{INT}$ ↓ Delay (Mode 2)		490		430
29	TdSTB(INT)	$\overline{STROBE}$ ↑ to $\overline{INT}$ ↓ Delay		440		350

Note : * Not compatible with NMOS specifications.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	V_{CC} Supply Voltage with Respect to V_{SS}	– 0.5 to 7	V
V_{IN}	Input Voltage	– 0.5 to V_{CC} + 0.5	V
P_D	Power Dissipation (T_A = 85 °C)	250	mW
T_{SOLDER}	Soldering Temperature (soldering time 10 sec)	260	°C
T_{STG}	Storage Temperature	– 65 to 150	°C
T_{OPR}	Operating Temperature	– 40 to 85	°C

DC CHARACTERISTICS (1)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_{ILC}	Clock Input Low Voltage		– 0.3	–	0.6	V
V_{IHC}	Clock Input High Voltage		V_{CC} – 0.6	–	V_{CC} + 0.3	V
V_{IL}	Input Low Voltage (except CLK)		– 0.5	–	0.8	V
V_{IH}	Input High Voltage (except CLK)		2.2	–	V_{CC}	V
V_{OL}	Output Low Voltage	I_{OL} = 2.0 mA	–	–	0.4	V
V_{OH1}	Output High Voltage (1)	I_{OH} = – 1.6 mA	2.4	–	–	V
V_{OH2}	Output High Voltage (2)	I_{OH} = – 250 µA	V_{CC} – 0.8	–	–	V
I_{LI}	Input Leakage Current	$V_{SS} \le V_{IN} \le V_{CC}$	–	–	± 10	µA
I_{OL},	3-State Output Leakage Current in Float	V_{SS} + 0.4 ≤ $V_{OUT} \le V_{CC}$	–	–	± 10	µA
I_{CC1}	Operating Supply Current : 4 MHz 6 MHz	 V_{CC} = 5 V, CLK = 4 MHz $V_{IH} = V_{CC}$ – 0.2 V, V_{IL} = 0.2 V	 – –	 2 3	 5 8	 mA mA
I_{CC2}	Stand-by Supply Current	V_{CC} = 5 V, CLK = V_{CC} $V_{IH} = V_{CC}$ – 0.2 V V_{IL} = 0.2 V	–	0.5	10	µA
*I_{OHD}	Darlington Drive Current	V_{OH} = 1.5 V, R_{EXT} = 1.1 kΩ	– 1.5	–	– 5.0	mA

Notes : 1. * Applied to Port B only.
2. Typical value is specified at 25 °C.

TEST CONDITIONS

T_A = – 40 °C to + 85 °C
V_{CC} = 5V ± 10%
V_{SS} = 0V

AC TEST CONDITIONS

- Inputs except CLK (clock) are driven at 2.4V for a logic "1" and 0.4V for a logic "0". Clock input is driven at V_{CC} – 0.6V for a logic "1" and 0.6V for a logic "0".
- Timing measurements are made at 2.2V for a logic "1" and 0.8V for a logic "0".

All AC parameters assume a load capacitance of 100pF.

ORDERING INFORMATION

Type	Package	Temp.	Clock	Description
Z84C20AB6	DIP-40 (plastic)	−40/+ 85°C	4 MHz	Z80C Parallel I/O Unit CMOS Version
Z84C20AD6	DIP-40 (ceramic)	−40/+ 85°C		
Z84C20AD2	DIP-40 (ceramic)	−55/+125°C		
Z84C20AC6	PLCC44 (plastic chip-carrier)	−40/+ 85°C		
Z84C20BB6	DIP-40 (plastic)	−40/+ 85°C	6 MHz	
Z84C20BD6	DIP-40 (ceramic)	−40/+ 85°C		
Z84C20BD2	DIP-40 (ceramic)	−55/+125°C		
Z84C20BC6	PLCC44 (plastic chip-carrier)	−40/+ 85°C		