Z8420/Z84C20 NMOS/CMOS Z80® PIO Parallel Input/Output

FEATURES

- Provides a direct interface between Z80 microcomputer systems and peripheral devices.
- Two ports with interrupt-driven handshake for fast response.
- Four programmable operating modes: Output, Input, Bidirectional (Port A only), and Bit Control
- Programmable interrupts on peripheral status conditions. (1.5 mV @ 1.5V)
- **NMOS version for** cost sensitive **performance solutions.**
- **CMOS version for the designs** requiring high speed and low power consumption
- **NMOS Z0842004 - 4 MHz, Z0842006 - 6.17 MHz.**
- CMOS Z84C2006 - DC to 6.17 MHz, Z84C2008 - DC to 8 MHz
- 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 1.5V).
- **6 MHz version supports 6.144 MHz CPU clock operation.**

GENERAL DESCRIPTION

The Z80 PIO Parallel I/O Circuit (hereinafter referred to as the Z80 PIO or PIO) is a programmable, dual-port device that provides a TTL-compatible interface between peripheral devices and the Z80 CPU (Figures 1 and 2). Note the LQFP package is only available in CMOS version. The CPU Configures the Z80 PIO to interface with a wide range of peripheral devices that are compatible with the Z80 PIO include most keyboards, paper tape readers and punches, printers, and PROM programmers.

One characteristic of the Z80 peripheral controllers that separates them from other interface controllers is that all

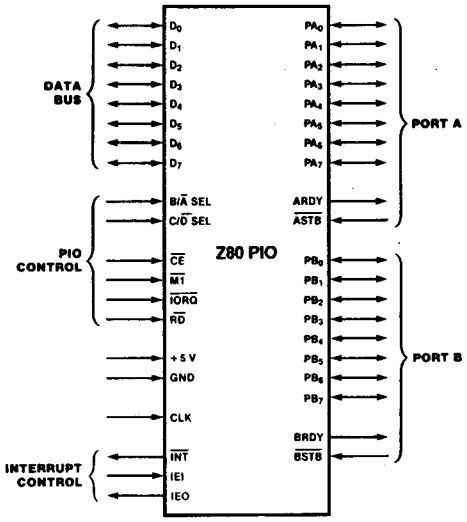


Figure 1. Pin Functions

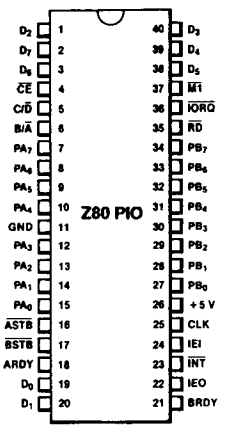


Figure 2a. 40-pin Dual-In-Line Package (DIP), Pin Assignments

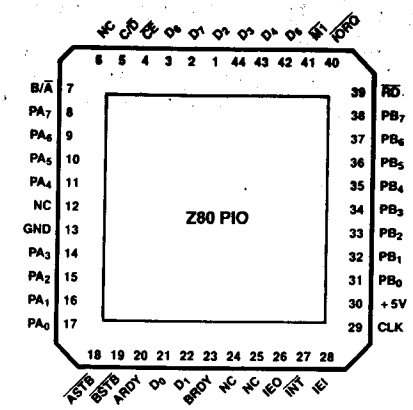


Figure 2b. 44-pin Chip Carrier, Pin Assignments

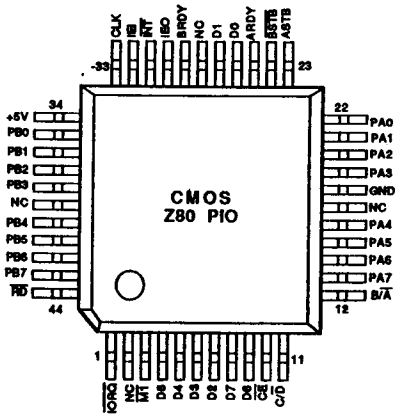


Figure 2c. 44-pin LQFP Pin Assignments

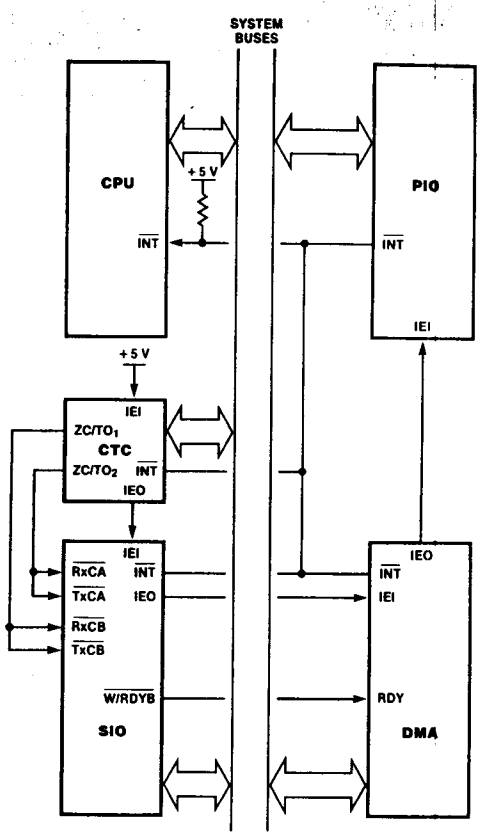


Figure 3. PIO in a Typical Z80 Family Environment

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 Z80 CPU during I/O transfers. All logic necessary to implement a fully nested interrupt structure is included in the PIO (Figure 3).

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 Z80 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 occurred.

Operating Modes. The Z80 PIO ports can be programmed to operate in four modes: Output (Mode 0), Input (Mode 1), Bidirectional (Mode 2) and Bit Control (Mode 3).

Either Port A or Port B can be programmed to output data in Mode 0. 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.

Either Port A or Port B can be programmed to input data in Mode 1. 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 only Port A, plus the interrupts and handshake signals from both ports. Port B must be set to Mode 3 and masked off from generating interrupts. 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 $\overline{ASTB}$ 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 the Z80 PIO interrupts, the Z80 CPU interrupt mode must be set to Mode 2.

INTERNAL STRUCTURE

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 (Figure 4). The CPU bus interface logic allows the Z80 PIO to interface directly to the Z80 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 Bit 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 govern 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

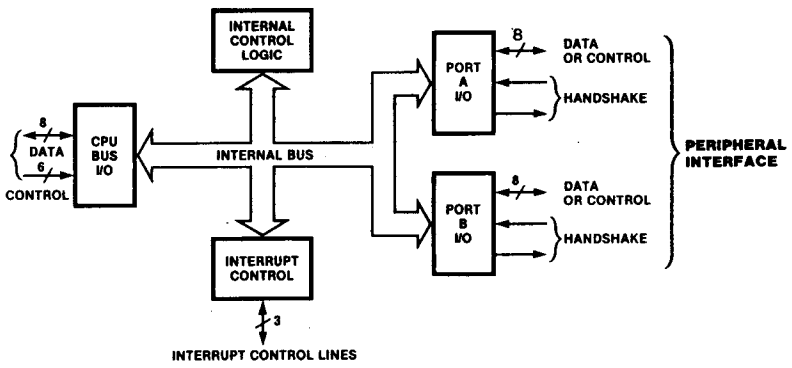


Figure 4. Block Diagram

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.

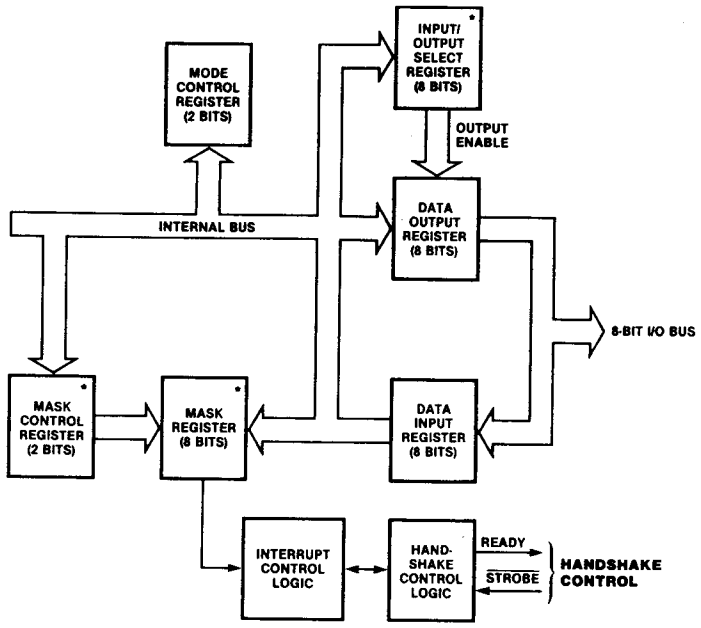
If the CPU (in interrupt Mode 2) accepts an interrupt, the interrupting device must provide an 8-bit interrupt vector for the CPU. This vector forms a pointer to a location in memory where the address of the interrupt service routine is located. The 8-bit vector from the interrupting device forms the least significant eight bits of the indirect pointer while the I Register in the CPU provides the most significant eight 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 Z80 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 Z80 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 Z80 PIO directly to the Z80 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 Z80 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 Z80 PIO does not receive a write input from the CPU; instead, the $\overline{RD}$, $\overline{CE}$, C/$\overline{D}$ and $\overline{IORQ}$ signals internally generate the write input.



*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. (Input, Output, or Bidirectional). Programming a port for Mode 0, 1, or 2 requires at least one, and up to three, control words per port. These words are:

Mode Control Word (Figure 6). Selects the port operating mode. This word is required and may be written at any time.

Interrupt Vector Word (Figure 7). The Z80 PIO is designed for use with the Z80 CPU in interrupt Mode 2. This word must be programmed if interrupts are to be used.

Interrupt Control Word (Figure 9) or ***Interrupt Disable Word*** (Figure 11). Controls the enable or disable of the PIO interrupt function.

Mode 3 (Bit Control). Programming a port for Mode 3 requires at least two, and up to four, control words.

Mode Control Word (Figure 6). Selects the port operating mode. This word is required and may be written at any time.

I/O Register Control Word (Figure 8). When Mode 3 is selected, the Mode Control Word must be followed by the I/O Control Word. This word configures the I/O control register, which defines which port lines are inputs or outputs. This word is required.

Interrupt Vector Word (Figure 7). The Z80 PIO is designed for use with the Z80 CPU in interrupt Mode 2. This word must be programmed if interrupts are to be used.

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_5 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_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 10).

Interrupt Disable Word. This control word 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).

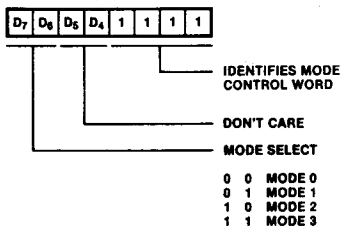


Figure 6. Mode Control Word

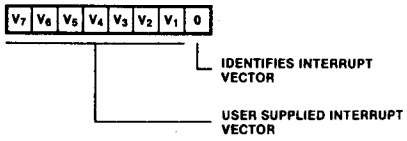


Figure 7. Interrupt Vector Word

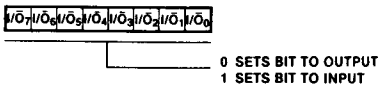


Figure 8. I/O Register Control Word

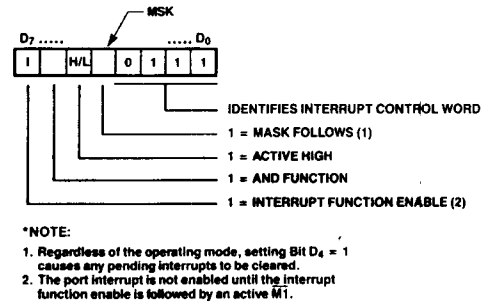


Figure 9. Interrupt Control Word

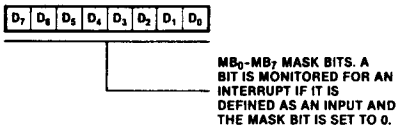


Figure 10. Mask Control Word

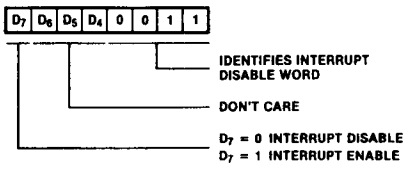


Figure 11. Interrupt Disable Word

PIN DESCRIPTION

PA_0-PA_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. PA_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 $\overline{ASTB}$ is active.

Control Mode. This signal is disabled and forced to a Low state.

$\overline{ASTB}$. *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.

PB_0-PB_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 1.5 mA at 1.5V to drive Darlington transistors. PB_0 is the least significant bit of the bus.

B/$\overline{A}$. *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.

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

C/$\overline{D}$. *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 Z80 data bus to be interpreted as a *command* for the port selected by the B/$\overline{A}$ Select line. A Low on this pin means that the Z80 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.

$\overline{CE}$. *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 Z80 PIO uses the standard single-phase Z80 system clock.

D_0-D_7. *Z80 CPU Data Bus* (bidirectional, 3-state). 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.

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.

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

$\overline{IORQ}$. *Input/Output Request* (input from Z80 CPU, active Low). $\overline{IORQ}$ is used in conjunction with B/$\overline{A}$, C/$\overline{D}$, $\overline{CE}$, and $\overline{RD}$ 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/$\overline{A}$ transfers data to the CPU (a read operation). Conversely, when $\overline{CE}$ and $\overline{IORQ}$ are active but $\overline{RD}$ is not, the port addressed by B/$\overline{A}$ is written into from the CPU with either data or control information, as specified by C/$\overline{D}$. Also, if $\overline{IORQ}$ 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 Z80 CPU is fetching an instruction from memory. Conversely, when both $\overline{M1}$ and $\overline{IORQ}$ are active, the CPU is acknowledging an interrupt. In addition, $\overline{M1}$ has two other functions within the Z80 PIO: it synchronizes the PIO interrupt logic; when $\overline{M1}$ occurs without an active $\overline{RD}$ or $\overline{IORQ}$ signal, the PIO is reset.

$\overline{RD}$. *Read Cycle Status* (input from Z80 CPU, active Low). If $\overline{RD}$ is active, or an I/O operation is in progress, $\overline{RD}$ is used with B/$\overline{A}$, C/$\overline{D}$, $\overline{CE}$, and $\overline{IORQ}$ to transfer data from the Z80 PIO to the Z80 CPU.

TIMING

The following timing diagrams show typical timing in a Z80 CPU environment. For more precise specifications refer to the composite ac timing diagram.

Write Cycle. Figure 12 illustrates the timing for programming the Z80 PIO or for writing data to one of its ports. 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 13 illustrates the timing for reading the data input from an external device to one of the Z80 PIO ports.

Output Mode (Mode 0). An output cycle (Figure 14) 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 $\overline{INT}$ if the interrupt enable flip-flop has been set and if this device has the highest priority.

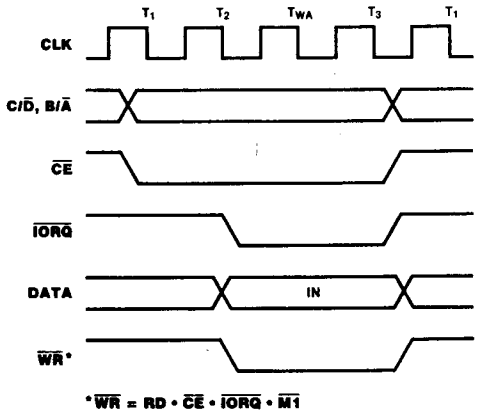


Figure 12. Write Cycle Timing

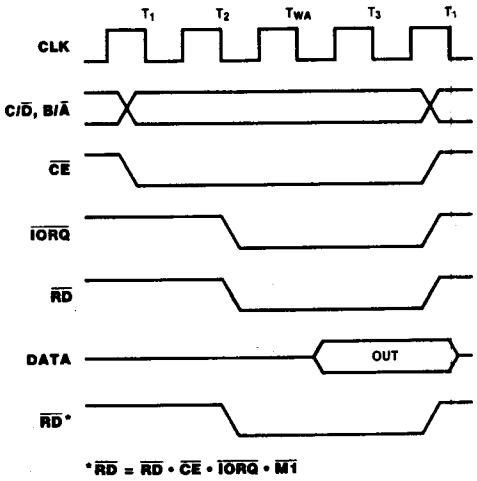


Figure 13. Read Cycle Timing

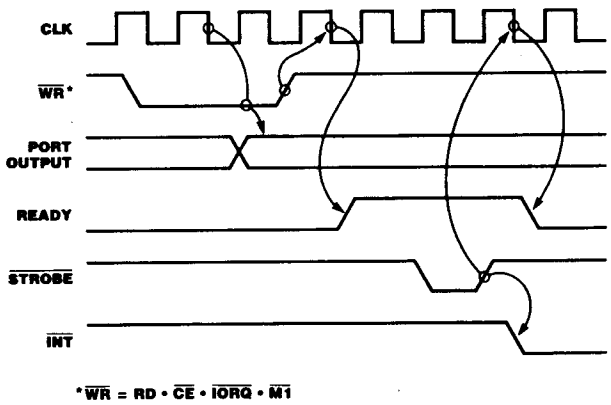


Figure 14. Mode 0 Output Timing

Input Mode (Mode 1). When $\overline{\text{STROBE}}$ goes from Low to High, data is latched into the selected port input register (Figure 15). While $\overline{\text{STROBE}}$ is Low, the input data latches are transparent. The next rising edge of $\overline{\text{STROBE}}$ activates $\overline{\text{INT}}$, 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 $\overline{\text{RD}}$ 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 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 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 $\overline{\text{ASTB}}$ is Low. The rising edge of this strobe can be used to latch the data into the peripheral.

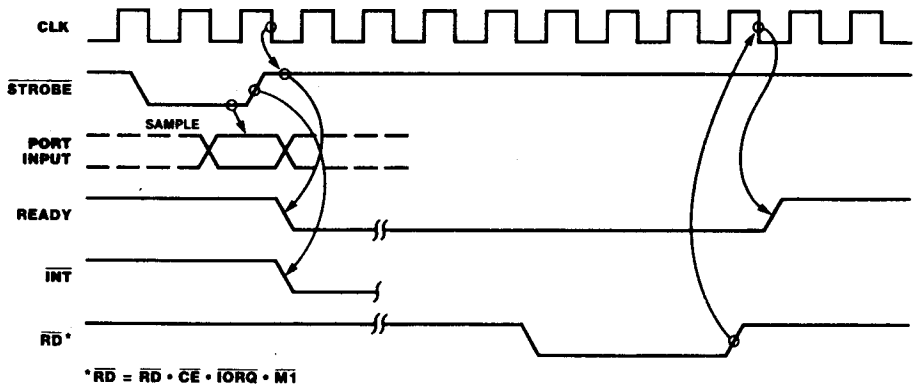


Figure 15. Mode 1 Input Timing

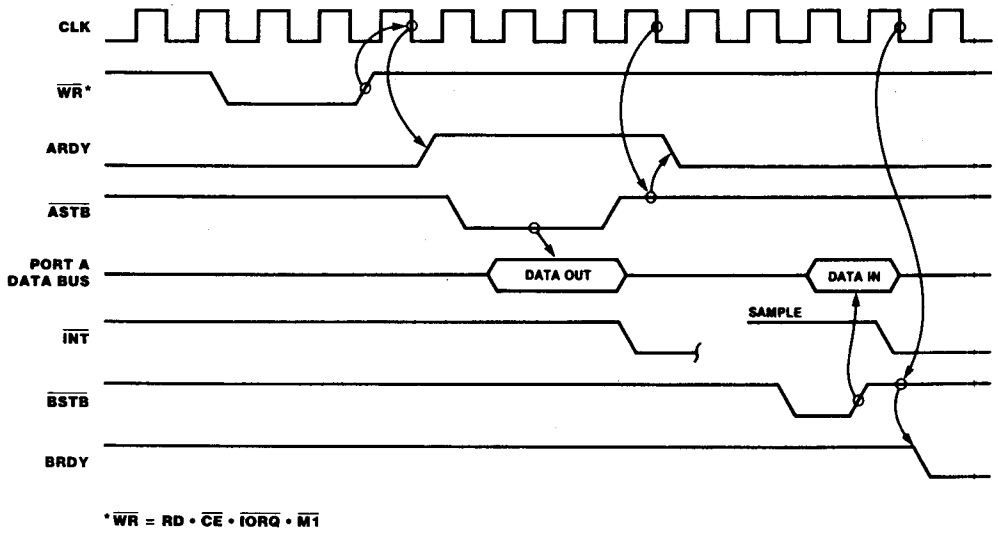


Figure 16. Mode 2 Bidirectional Timing

Bit Control 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.

When reading (Figure 17) 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.

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 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.

Return From Interrupt Cycle. If a Z80 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 19). 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.

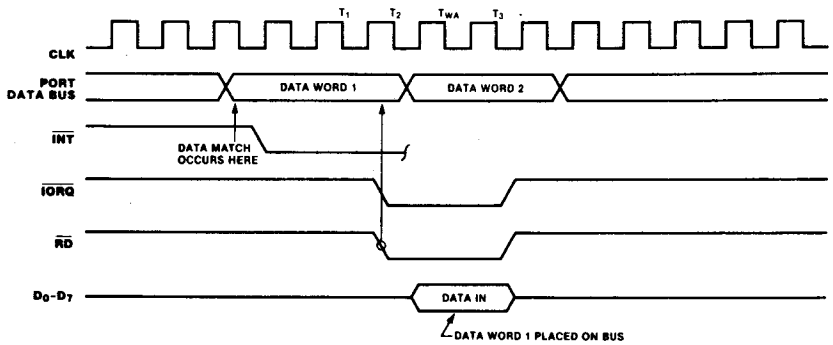


Figure 17. Mode 3 Bit Control Mode Timing, Bit Mode Read

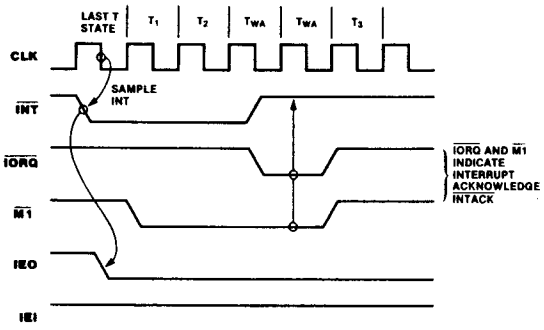


Figure 18. Interrupt Acknowledge Timing

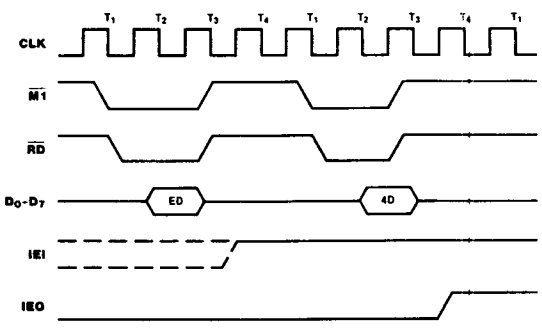


Figure 19. Return From Interrupt

ABSOLUTE MAXIMUM RATINGS

Voltages on V_{CC} with respect to V_{SS} . . . . . – 0.3V to + 7.0V
Voltages on all inputs with respect to V_{SS} . . . . . – 0.3V to V_{CC} + 0.3V
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 these indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

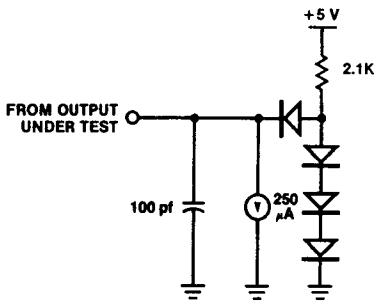
STANDARD TEST CONDITIONS

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

- **S = 0°C to +70°C, V_{CC} Range**
  NMOS: $+4.75V \leq V_{CC} \leq +5.25V$
  CMOS: $+4.50V \leq V_{CC} \leq +5.50V$

- E= -40°C to 100°C, $+4.50V \leq V_{CC} \leq +5.50V$

The Ordering Information section lists package temperature ranges and product numbers. Refer to the Literature List for additional documentation. Package drawings are in the Package Information section.



CAPACITANCE

Symbol	Parameter	Min	Max	Unit
C	Clock Capacitance		10	pf
C_{IN}	Input Capacitance		5	pf
C_{OUT}	Output Capacitance		15	pf

Over specified temperature range; f = 1 MHz.
Unmeasured pins returned to ground.

DC CHARACTERISTICS (Z84C20/CMOS Z80 PIO)

V_{CC}=5.0V ± 10%, unless otherwise specified

Symbol	Parameter	Min	Max	Typ	Unit	Condition
V_{ILC}	Clock Input Low Voltage	-0.3	+0.45		V	
V_{IHC}	Clock Input High Voltage	V_{CC}-0.6	V_{CC}+0.3		V	
V_{IL}	Input High Voltage	2.2	V_{CC}		V	
V_{IH}	Input Low Voltage	-0.3	0.8		V	
V_{OL}	Output Low Voltage		0.4		V	I_{LO}=2.0mA
V_{OH1}	Output High Voltage	2.4			V	I_{OH}=-1.6mA
V_{OH2}	Output High Voltage	V_{CC}-0.8			V	I_{OH}=-250µA
I_{LI}	Input Leakage Current	-10	10		µA	V_{IN}=0.4V to V_{CC}
I_{LO}	3-state Output Leakage Current in Float	-10	10		µA	V_{OUT}=0.4V to V_{CC}
I_{CC1}	Power Supply Current - 4MHz		5	2	mA	V_{CC}=5V
	- 6MHz		6		mA	CLK=4,6,8,10MHz
	- 8MHz		7		mA	V_{IH}=V_{CC}-0.2V
	- 10MHz		12		mA	V_{IL}=0.2V
I_{CC2}	Standby Supply Current		10	0.5	µA	V_{CC}=5V CLK=(0) V_{IH}=V_{CC}-0.2V V_{IL}=0.2V
I_{OHD}	Darlington Drive Current (Port B Only)	-1.5	-5.0		mA	V_{OH}=1.5V REXT=1.1K ohm

Note:
[1] Measurements made with outputs floating.

AC CHARACTERISTICS (Z84C20/CMOS Z80 PIO)

No	Symbol	Parameter	Z84C2004* Min	Z84C2004* Max	Z84C2006 Min	Z84C2006 Max	Z84C2008 Min	Z84C2008 Max	Z84C2010 Min	Z84C2010 Max	Note
1	TcC	Clock Cycle Time	250	[1]	162	[1]	125	[1]	100	[1]	
2	TwCh	Clock Pulse Width (High)	110	DC	65	DC	55	DC	42	DC	
3	TwCl	Clock Pulse Width (Low)	110	DC	65	DC	55	DC	42	DC	
4	TfC	Clock Fall Time		30		20		10		10	
5	TrC	Clock Rise Time		30		20		10		10	
6	TsCS(RI)	/CE,B//A,C//D to /RD, /IORQ Fall Setup Time	50		50		40		35		[6]
7	Th	Any Hold Times for Specified Setup Time	40		35		15		15		
8	TsRI(C)	/RD, /IORQ to Clock Rise Setup Time	115		70		60		40		
9	TdRI(DO)	/RD, /IORQ Fall to Data Out Delay		380		300		200		120	[2]
10	TdRI(DOs)	/RD, /IORQ Rise to Data Out Float Delay		110		70		60		50	
11	TsDI(C)	Data In to Clock Rise Setup Time	50		40		30		20		CL=50pF
12	TdIO(DOI)	/IORQ Falling to Data Out Delay (INTACK Cycle)		160		120		80		80	[3]
13	TsM1(Cr)	/M1 Falling to Clock Rising Setup Time	90		70		50		40		
14	TsM1(Cf)	/M1 Fall to Clock Rise Setup Time (/M1 cycle)	0		0		0		-20		
15	TdM1(IEO)	/M1 Fall to IEO Fall Delay (Interrupt Immediately Preceding /M1 Fall)		190		100		70		70	[5,7]
16	TsIEI(IO)	IEI to /IORQ Falling Setup Time (/INTACK Cycle)	140		100		80		60		[7]
17	TdIEI(IEOf)	IEI Fall to IEO Fall Delay		130		120		70		70	[5]
18	TdIEI(IEOr)	IEI Rise to IEO Rise Delay		160		150		70		70	CL=50pF
19	TcIO(C)	/IORQ Rise to Clock Fall Setup Time (To Activate RDY on Next Clock Cycle)	200		170		140		120		
20	TdC(RDYr)	Clock Fall to RDY Rise Delay		190		170		150		130	[5], CL=50pF
21	TdC(RDYf)	Clock Fall to RDY Fall Delay		140		120		100		85	[5]
22	TwSTB	/STB Pulse Width	150		120		100		80		[4]
23	TsSTB(C)	/STB Rise to Clock Fall Setup Time (To Activate RDY on Next Clock Cycle)	220		150		120		100		[5]
24	TdIO(PD)	/IORQ Rise to Port Data Stable Delay (Mode 0)		180		160		140		120	[5]
25	TsPD(STB)	Port Data to /STB Rise Setup Time (Mode 1)	230		190		140		75		
26	TdSTB(PD)	/STB Fall to Port Data Stable (Mode 2)		210		180		150		120	[5]

*4 MHz CMOS 84C20 is obsoleted and replaced by 6 MHz.

Z84C20 AC CHARACTERISTICS (Continued)

No	Symbol	Parameter	Z84C2004* Min	Z84C2004* Max	Z84C2006 Min	Z84C2006 Max	Z84C2008 Min	Z84C2008 Max	Z84C2010 Min	Z84C2010 Max	Note
27	TdSTB(PDr)	/STB Rise to Port Data Float Delay (Mode 2)		180		160		140		120	CL=50pF
28	TdPD(INT)	Port Data Match to /INT Fall Delay (Mode 3)		490		430		360		200	
29	TdSTB(INT)	/STB Rise to /INT Fall Delay		440		350		29		220	

* All parameters in nanosecond, unless otherwise specified.
* 4 MHz Z84C30 is obsoleted and replaced by 6 MHz

Notes:
[1] TcC = TwCh + TwCl + TrC + TfC.
[2] Increase TdRI(DO) by 10nS for each 50pF increasing in Load up to 200pF max.
[3] Increase TdIO(DOI) by 10nS for each 50pF increasing in Load up to 200pF max.
[4] For Mode 2: TwSTB>TsPD(STB).
[5] Increase these values by 2nS for 10pF increase in loading up to 100pF Max.
[6] TsCS(RI) may be reduced. However, the time subtracted from TsCS(RI) will be added to TdRI(DO).
[7] 2.5TcT > (N-2)TdIEI(IEOf) + TdM1(IEO) + TsIEI(IO) + TTL Buffer Delay, if any.
[8] M1 must be active for a minumum of two clock cycles to reset the PIO.

DC CHARACTERISTICS (Z8420/NMOS Z80 PIO)

Symbol	Parameter	Min	Max	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	−0.3	+0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC} −0.6	V_{CC} +0.3	V	
V_{IL}	Input Low Voltage	−0.3	+0.8	V	
V_{IH}	Input High Voltage	+2.0	V_{CC}	V	
V_{OL}	Output Low Voltage		+0.4	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	+2.4		V	I_{OH} = −250 μA
I_{LI}	Input Leakage Current		±10	μA	V_{IN} = 0 to V_{CC}
I_{LO}	3-State Output Leakage Current in Float		±10	μA	V_{OUT} = 0.4V to V_{CC}
I_{CC}	Power Supply Current		100	mA	
I_{OHD}	Darlington Drive Current Port B Only	−1.5		mA	V_{OH} = 1.5V R_{EXT} = 390 Ω

Over specified temperature and voltage range.

AC CHARACTERISTICS† (Z8420/NMOS Z80 PIO)

Number	Symbol	Parameter	Z0842004 Min	Z0842004 Max	Z0842006 Min	Z0842006 Max	Notes
1	TcC	Clock Cycle Time	250	[1]	162	[1]	
2	TwCh	Clock Width (High)	105	2000	65	2000	
3	TwCl	Clock Width (Low)	105	2000	65	2000	
4	TfC	Clock Fall Time		30		20	
5	TrC	Clock Rise Time		30		20	
6	TsCS(RI)	$\overline{CE}$, B/$\overline{A}$, C/$\overline{D}$ to $\overline{RD}$, $\overline{IORQ}$ ↓ Setup Time	50		50		[6]
7	Th	Any Hold Times for Specified Setup Time	0		0	0	
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	[2]
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		CL = 50 pf
12	TdIO(DOI)	$\overline{IORQ}$ ↓ to Data Out Delay (INTACK Cycle)		200		120	[3]
13	TsM1(Cr)	$\overline{M1}$ ↓ to Clock ↑ Setup Time	90		70		
14	TsM1(Cf)	$\overline{M1}$ ↑ to Clock ↓ Setup Time ($\overline{M1}$ Cycle)	0		0		[8]
15	TdM1(IEO)	$\overline{M1}$ ↓ to IEO ↓ Delay (Interrupt Immediately Preceding $\overline{M1}$ ↓)		190		100	[5,7]
16	TsIEI(IO)	IEI to $\overline{IORQ}$ ↓ Setup Time (INTACK Cycle)	140		100		[7]
17	TdIEI(IEOf)	IEI ↓ to IEO ↓ Delay		130		120	[5] CL = 50 pf
18	TdIEI(IEOr)	IEI ↑ to IEO ↑ Delay (after ED Decode)		160		150	[5]
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	[5] CL = 50 pf
21	TdC(RDYf)	Clock ↓ to READY ↓ Delay		140		120	[5]
22	TwSTB	$\overline{STROBE}$ Pulse Width	150		120		[4]
23	TsSTB(C)	$\overline{STROBE}$ ↑ to Clock ↓ Setup Time (To Activate READY on Next Clock Cycle)	220		150		[5]
24	TdIO(PD)	$\overline{IORQ}$ ↑ to PORT DATA Stable Delay (Mode 0)		180		160	[5]
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	[5]
27	TdSTB(PDr)	$\overline{STROBE}$ ↑ to PORT DATA Float Delay (Mode 2)		180		160	CL = 50 pf
28	TdPD(INT)	PORT DATA Match to $\overline{INT}$ ↓ Delay (Mode 3)		490		430	
29	TdSTB(INT)	$\overline{STROBE}$ ↑ to $\overline{INT}$ ↓ Delay		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).

* $\overline{M1}$ must be active for a minimum of two clock cycles to reset the PIO.

† Units in nanoseconds (ns).

AC TIMING DIAGRAM