Farnell
COMPONENTS

Z80 Family Data

Z8400/Z84C00 NMOS/CMOS Z80® CPU Central Processing Unit

1

FEATURES

The extensive instruction set contains 158 instructions, including the 8080A instruction set as a subset.

- NMOS version for low cost high performance solutions, CMOS version for high performance low power designs.
- NMOS Z0840004 - 4 MHz, Z0840006 - 6.17 MHz, Z0840008 - 8 MHz.
- CMOS Z84C0006 - DC to 6.17 MHz, Z84C008 - DC to 8 MHz, Z84C0010 - DC to 10 MHz, Z84C0020 - DC - 20 MHz
- 6 MHz version can be operated at 6.144 MHz clock.
- The Z80 microprocessors and associated family of peripherals can be linked by a vectored interrupt system. This system can be daisy-chained to allow implementation of a priority interrupt scheme.
- Duplicate set of both general-purpose and flag registers.
- Two sixteen-bit index registers.
- Three modes of maskable interrupts:
  Mode 0—8080A similar;
  Mode 1—Non-Z80 environment, location 38H;
  Mode 2—Z80 family peripherals, vectored interrupts.
- On-chip dynamic memory refresh counter.

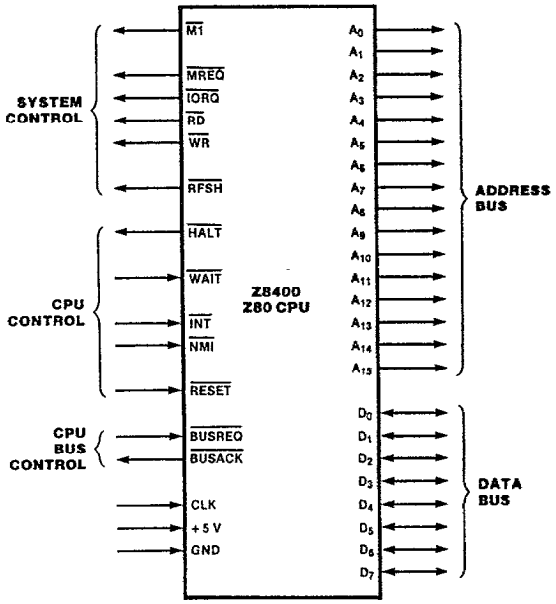


Figure 1. Pin Functions

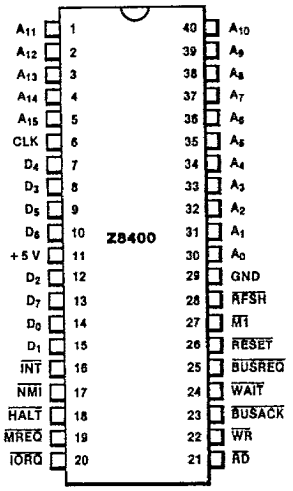
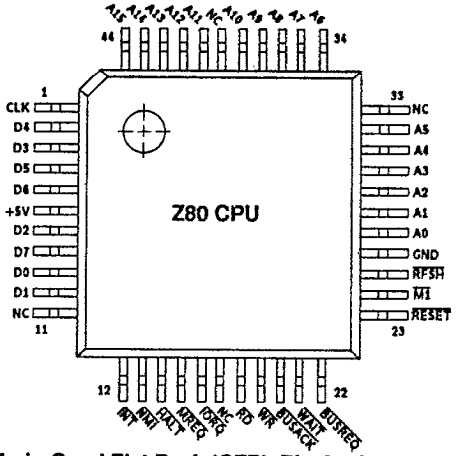


Figure 2. 40-pin Dual-In-Line (DIP), Pin Assignments



44 pin Quad Flat Pack (QFP), Pin Assignments (Only available for 84C00)

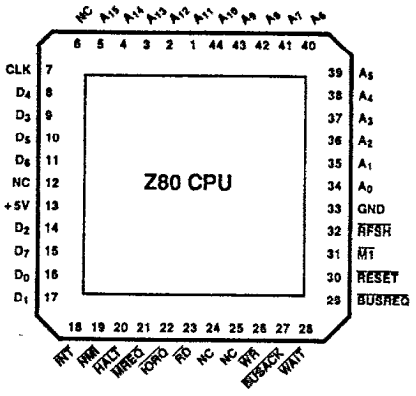


Figure 2b. 44-Pin Chip Carrier Pin Assignments

GENERAL DESCRIPTION

The CPUs are fourth-generation enhanced microprocessors with exceptional computational power. They offer higher system throughput and more efficient memory utilization than comparable second- and third-generation microprocessors. The internal registers contain 208 bits of read/write memory that are accessible to the programmer. These registers include two sets of six general-purpose registers which may be used individually as either 8-bit registers or as 16-bit register pairs. In addition, there are two sets of accumulator and flag registers. A group of "Exchange" instructions makes either set of main or alternate registers accessible to the programmer. The alternate set allows operation in foreground-background mode or it may be reserved for very fast interrupt response.

The CPU also contains a Stack Pointer, Program Counter, two index registers, a Refresh register (counter), and an Interrupt register. The CPU is easy to incorporate into a system since it requires only a single + 5V power source. All output signals are fully decoded and timed to control standard memory or peripheral circuits; the CPU is supported by an extensive family of peripheral controllers. The internal block diagram (Figure 3) shows the primary functions of the processors. Subsequent text provides more detail on the I/O controller family, registers, instruction set, interrupts and daisy chaining, and CPU timing.

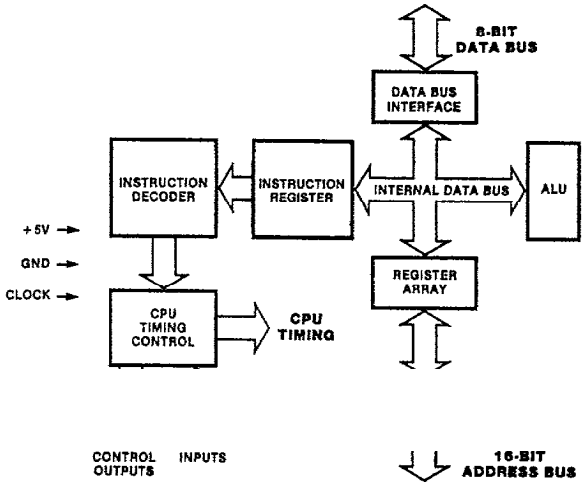


Figure 3. Z80C CPU Block Diagram

Table 1. Z80C CPU Registers

Register		Size (Bits)	Remarks
A, A'	Accumulator	8	Stores an operand or the results of an operation.
F, F'	Flags	8	See Instruction Set.
B, B'	General Purpose	8	Can be used separately or as a 16-bit register with C.
C, C'	General Purpose	8	Can be used separately or as a 16-bit register with C.
D, D'	General Purpose	8	Can be used separately or as a 16-bit register with E.
E, E'	General Purpose	8	Can be used separately or as a 16-bit register with E.
H, H'	General Purpose	8	Can be used separately or as a 16-bit register with L.
L, L'	General Purpose	8	Can be used separately or as a 16-bit register with L.
			Note: The (B,C), (D,E), and (H,L) sets are combined as follows: B — High byte C — Low byte D — High byte E — Low byte H — High byte L — Low byte
I	Interrupt Register	8	Stores upper eight bits of memory address for vectored interrupt processing.
R	Refresh Register	8	Provides user-transparent dynamic memory refresh. Automatically incremented and placed on the address bus during each instruction fetch cycle.
IX	Index Register	16	Used for indexed addressing.
IY	Index Register	16	Used for indexed addressing
SP	Stack Pointer	16	Holds address of the top of the stack. See Push or Pop in instruction set.
PC	Program Counter	16	Holds address of next instruction.
IFF_1-IFF_2	Interrupt Enable	Flip-Flops	Set or reset to indicate interrupt status (see Figure 4).
IMFa-IMFb	Interrupt Mode	Flip-Flops	Reflect Interrupt mode (see Figure 4).

1

failure has been detected. After recognition of the $\overline{NMI}$ signal (providing $\overline{BUSREQ}$ is not active), the CPU jumps to restart location 0066H. Normally, software starting at this address contains the interrupt service routine.

Maskable Interrupt ($\overline{INT}$). Regardless of the interrupt mode set by the user, the CPU response to a maskable interrupt is identical to an 8080 microprocessor interrupt. The interrupt processing cycle begins. This is a special fetch ($\overline{M1}$) cycle in which $\overline{IORQ}$ becomes active rather than $\overline{MREQ}$, as in a normal $\overline{M1}$ cycle. In addition, this special $\overline{M1}$ cycle is automatically extended by two $\overline{WAIT}$ states, to allow for the time required to acknowledge the interrupt request.

8080 microprocessor interrupt service procedures. The interrupting device places an instruction on the data bus. This is normally a Restart instruction, which will initiate a call to the selected one of eight restart locations in page zero of memory. Unlike the 8080, the Z80 CPU responds to the Call instruction with only one interrupt acknowledge cycle followed by two memory read cycles.

Mode 1 Interrupt Operation. Mode 1 operation is very designed to most effectively utilize the capabilities of the Z80 microprocessor and its associated peripheral family. The interrupting peripheral device selects the starting address of the interrupt service routine. It does this by placing an 8-bit vector on the data bus during the interrupt acknowledge [illegible] 8 bits and the contents of the I register as the upper 8 bits. This points to an entry in a table of addresses for interrupt service routines. The CPU then jumps to the routine at that

address. This flexibility in selecting the interrupt service routine address allows the peripheral device to use several different types of service routines. These routines may be located at any available location in memory. Since the interrupting device supplies the low-order byte of the 2-byte vector, bit 0 (A_0) must be a zero.

Interrupt Enable/Disable Operation. Two flip-flops, IFF_1 and IFF_2, referred to in the register description, are used to signal the CPU interrupt status. Operation of the two flip-flops is described in Table 2. For more details, refer to the *Z80 CPU Technical Manual* (03-0029-01) and *Z80 Assembly Language Programming Manual* (03-0002-01).

Table 2. State of Flip-Flops

Action	IFF_1	IFF_2	Comments
CPU Reset	0	0	Maskable interrupt $\overline{INT}$ disabled
DI instruction execution	0	0	Maskable interrupt $\overline{INT}$ disabled
EI instruction execution	1	1	Maskable interrupt $\overline{INT}$ enabled
LD A,I instruction execution	•	•	IFF_2 → Parity flag
LD A,R instruction execution	•	•	IFF_2 → Parity flag
Accept $\overline{NMI}$	0	•	Maskable interrupt $\overline{INT}$ disabled
RETN instruction execution	IFF_2	•	IFF_2 → IFF_1 at completion of an $\overline{NMI}$ service routine.

INSTRUCTION SET

The microprocessor has one of the most powerful and versatile instruction sets available in any 8-bit microprocessor. It includes such unique operations as a block move for fast, efficient data transfers within memory, or between memory and I/O. It also allows operations on any bit in any location in memory.

The following is a summary of the instruction set which shows the assembly language mnemonic, the operation, the flag status, and gives comments on each instruction. For an explanation of flag notations and symbols for mnemonic tables, see the Symbolic Notations section which follows these tables. The *Z80 CPU Technical Manual* (03-0029-01), the *Programmer's Reference Guide* (03-0012-03), and *Assembly Language Programming Manual* (03-0002-01) contain significantly more details for programming use.

The instructions are divided into the following categories:

- 8-bit loads
- 16-bit loads
- Exchanges, block transfers, and searches
- 8-bit arithmetic and logic operations
- General-purpose arithmetic and CPU control
- 16-bit arithmetic operations
- Rotates and shifts
- Bit set, reset, and test operations
- Jumps
- Calls, returns, and restarts
- Input and output operations

A variety of addressing modes are implemented to permit efficient and fast data transfer between various registers, memory locations, and input/output devices. These addressing modes include:

- Immediate
- Immediate extended
- Modified page zero
- Relative
- Extended
- Indexed
- Register
- Register indirect
- Implied
- Bit

8-BIT LOAD GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD r, r'	r ← r'	•	•	X	•	•	•	01	r	r'		1	1	4	r, r' Reg.
LD r, n	r ← n	•	•	X	•	•	•	00	r	110		2	2	7	000 B
									← n →						001 C
LD r, (HL)	r ← (HL)	•	•	X	•	•	•	01	r	110		1	2	7	010 D
LD r, (IX+d)	r ← (IX+d)	•	•	X	•	•	•	11	011	101	DD	3	5	19	011 E
								01	r	110					100 H
									← d →						101 L
LD r, (IY+d)	r ← (IY+d)	•	•	X	•	•	•	11	111	101	FD	3	5	19	111 A
								01	r	110					
									← d →						
LD (HL), r	(HL) ← r	•	•	X	•	•	•	01	110	r		1	2	7	
LD (IX+d), r	(IX+d) ← r	•	•	X	•	•	•	11	011	101	DD	3	5	19	
								01	110	r					
									← d →						
LD (IY+d), r	(IY+d) ← r	•	•	X	•	•	•	11	111	101	FD	3	5	19	
								01	110	r					
									← d →						
LD (HL), n	(HL) ← n	•	•	X	•	•	•	00	110	110	36	2	3	10	
									← n →						
LD (IX+d), n	(IX+d) ← n	•	•	X	•	•	•	11	011	101	DD	4	5	19	
								00	110	110	36				
									← d →						
									← n →						

8-BIT LOAD GROUP (Continued)

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD (IY+d), n	(IY+d) ← n	•	•	X	•	X	•	•	•	11	111	101	FD	4	5	19	
										00	110	110	36				
											← d →						
											← n →						
LD A, (BC)	A ← (BC)	•	•	X	•	X	•	•	•	00	001	010	0A	1	2	7	
LD A, (DE)	A ← (DE)	•	•	X	•	X	•	•	•	00	011	010	1A	1	2	7	
LD A, (nn)	A ← (nn)	•	•	X	•	X	•	•	•	00	111	010	3A	3	4	13	
											← n →						
											← n →						
LD (BC), A	(BC) ← A	•	•	X	•	X	•	•	•	00	000	010	02	1	2	7	
LD (DE), A	(DE) ← A	•	•	X	•	X	•	•	•	00	010	010	12	1	2	7	
LD (nn), A	(nn) ← A	•	•	X	•	X	•	•	•	00	110	010	32	3	4	13	
											← n →						
											← n →						
LD A, I	A ← I	‡	‡	X	0	X	IFF	0	•	11	101	101	ED	2	2	9	
										01	010	111	57				
LD A, R	A ← R	‡	‡	X	0	X	IFF	0	•	11	101	101	ED	2	2	9	
										01	011	111	5F				
LD I, A	I ← A	•	•	X	•	X	•	•	•	11	101	101	ED	2	2	9	
										01	000	111	47				
LD R, A	R ← A	•	•	X	•	X	•	•	•	11	101	101	ED	2	2	9	
										01	001	111	4F				

NOTE: IFF, the content of the interrupt enable flip-flop, (IFF_2), is copied into the P/V flag.

16-BIT LOAD GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD dd, nn	dd ← nn	•	•	X	•	X	•	•	•	00	dd0	001		3	3	10	dd Pair
											← n →						00 BC
											← n →						01 DE
LD IX, nn	IX ← nn	•	•	X	•	X	•	•	•	11	011	101	DD	4	4	14	10 HL
										00	100	001	21				11 SP
											← n →						
											← n →						
LD IY, nn	IY ← nn	•	•	X	•	X	•	•	•	11	111	101	FD	4	4	14	
										00	100	001	21				
											← n →						
											← n →						
LD HL, (nn)	H ← (nn+1)	•	•	X	•	X	•	•	•	00	101	010	2A	3	5	16	
	L ← (nn)										← n →						
											← n →						
LD dd, (nn)	dd_H ← (nn+1)	•	•	X	•	X	•	•	•	11	101	101	ED	4	6	20	
	dd_L ← (nn)									01	dd1	011					
											← n →						
											← n →						

NOTE: $(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively. e.g., BC_L = C, AF_H = A.

16-BIT LOAD GROUP (Continued)

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD IX, (nn)	$IX_H \leftarrow (nn+1)$ $IX_L \leftarrow (nn)$	•	•	X	X	•	•	11 00	011 101 ←n→ ←n→	101 010	DD 2A	4	6	20	
LD IY, (nn)	$IY_H \leftarrow (nn+1)$ $IY_L \leftarrow (nn)$	•	•	X	X	•	•	11 00	111 101 ←n→ ←n→	101 010	FD 2A	4	6	20	
LD (nn), HL	$(nn+1) \leftarrow H$ $(nn) \leftarrow L$	•	•	X	X	•	•	00	100 ←n→ ←n→	010	22	3	5	16	
LD (nn), dd	$(nn+1) \leftarrow dd_H$ $(nn) \leftarrow dd_L$	•	•	X	X	•	•	11 01	101 dd0 ←n→ ←n→	101 011	ED	4	6	20	
LD (nn), IX	$(nn+1) \leftarrow IX_H$ $(nn) \leftarrow IX_L$	•	•	X	X	•	•	11 00	011 100 ←n→ ←n→	101 010	DD 22	4	6	20	
LD (nn), IY	$(nn+1) \leftarrow IY_H$ $(nn) \leftarrow IY_L$	•	•	X	X	•	•	11 00	111 100 ←n→ ←n→	101 010	FD 22	4	6	20	
LD SP, HL	$SP \leftarrow HL$	•	•	X	X	•	•	11	111	001	F9	1	1	6	
LD SP, IX	$SP \leftarrow IX$	•	•	X	X	•	•	11 11	011 111	101 001	DD F9	2	2	10	
LD SP, IY	$SP \leftarrow IY$	•	•	X	X	•	•	11 11	111 111	101 001	FD F9	2	2	10	
PUSH qq	$(SP-2) \leftarrow qq_L$ $(SP-1) \leftarrow qq_H$ $SP \rightarrow SP-2$	•	•	X	X	•	•	11	qq0	101		1	3	11	
PUSH IX	$(SP-2) \leftarrow IX_L$ $(SP-1) \leftarrow IX_H$ $SP \rightarrow SP-2$	•	•	X	X	•	•	11 11	011 100	101 101	DD E5	2	4	15	
PUSH IY	$(SP-2) \leftarrow IY_L$ $(SP-1) \leftarrow IY_H$ $SP \rightarrow SP-2$	•	•	X	X	•	•	11 11	111 100	101 101	FD E5	2	4	15	
POP qq	$qq_H \leftarrow (SP+1)$ $qqL \leftarrow (SP)$ $SP \rightarrow SP+2$	•	•	X	X	•	•	11	qq0	001		1	3	10	
POP IX	$IX_H \leftarrow (SP+1)$ $IX_L \leftarrow (SP)$ $SP \rightarrow SP+2$	•	•	X	X	•	•	11 11	011 100	101 001	DD E1	2	4	14	
POP IY	$IY_H \leftarrow (SP+1)$ $IY_L \leftarrow (SP)$ $SP \rightarrow SP+2$	•	•	X	X	•	•	11 11	111 100	101 001	FD E1	2	4	14	

qq	Pair
00	BC
01	DE
10	HL
11	AF

NOTE: $(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively, e.g., BC_L = C, AF_H = A.

EXCHANGE, BLOCK TRANSFER, BLOCK SEARCH GROUPS

Mnemonic	Symbolic Operation	S	Z		H		P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
EX DE, HL	DE ↔ HL	•	•	X	•	X	•	•	•	11	101	011	EB	1	1	4	
EX AF, AF′	AF ↔ AF′	•	•	X	•	X	•	•	•	00	001	000	08	1	1	4	
EXX	BC ↔ BC′ DE ↔ DE′ HL ↔ HL′	•	•	X	•	X	•	•	•	11	011	001	D9	1	1	4	Register bank and auxiliary register bank exchange
EX (SP), HL	H ↔ (SP + 1) L ↔ (SP)	•	•	X	•	X	•	•	•	11	100	011	E3	1	5	19	
EX (SP), IX	IX_H ↔ (SP + 1) IX_L ↔ (SP)	•	•	X	•	X	•	•	•	11 11	011 100	101 011	DD E3	2	6	23	
EX (SP), IY	IY_H ↔ (SP + 1) IYL ↔ (SP)	•	•	X	•	X	•	•	•	11 11	111 100	101 011	FD E3	2	6	23	
LDI	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC − 1	•	•	X	0	X	↕ ①	0	•	11 10	101 100	101 000	ED A0	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
LDIR	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC − 1 Repeat until BC = 0	•	•	X	0	X	0 ②	0	•	11 10	101 110	101 000	ED B0	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
LDD	(DE) ← (HL) DE ← DE − 1 HL ← HL − 1 BC ← BC − 1	•	•	X	0	X	↕ ①	0	•	11 10	101 101	101 000	ED A8	2	4	16	
LDDR	(DE) ← (HL) DE ← DE − 1 HL ← HL − 1 BC ← BC − 1 Repeat until BC = 0	•	•	X	0	X	0 ②	0	•	11 10	101 111	101 000	ED B8	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
CPI	A − (HL) HL ← HL + 1 BC ← BC − 1	↕	↕ ③	X	↕	X	↕ ①	1	•	11 10	101 100	101 001	ED A1	2	4	16	

NOTE: ① P/V flag is 0 if the result of BC − 1 = 0, otherwise P/V = 1.
② P/V flag is 0 only at completion of instruction.
③ Z flag is 1 if A = HL , otherwise Z = 0.

EXCHANGE, BLOCK TRANSFER, BLOCK SEARCH GROUPS (Continued)

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
			③		①										
CPIR	A – (HL)	‡	‡	X	‡	X	‡	1	•	11	101	101	ED	2	5	21	If BC ≠ 0 and A ≠ (HL)
	HL ← HL + 1 BC ← BC – 1 Repeat until A = (HL) or BC = 0									10	110	001	B1	2	4	16	If BC = 0 or A = (HL)
			③		①										
CPD	A – (HL) HL ← HL – 1 BC ← BC – 1	‡	‡	X	‡	X	‡	1	•	11 10	101 101	101 001	ED A9	2	4	16	
			③		①										
CPDR	A – (HL)	‡	‡	X	‡	X	‡	1	•	11	101	101	ED	2	5	21	If BC ≠ 0 and A ≠ (HL)
	HL ← HL – 1 BC ← BC – 1 Repeat until A = (HL) or BC = 0									10	111	001	B9	2	4	16	If BC = 0 or A = (HL)

NOTE: ① P/V flag is 0 if the result of BC – 1 = 0, otherwise P/V = 1.
② P/V flag is 0 only at completion of instruction.
③ Z flag is 1 if A = (HL), otherwise Z = 0.

8-BIT ARITHMETIC AND LOGICAL GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD A, r	A ← A + r	‡	‡	X	‡	X	V	0	‡	10	[000]	r		1	1	4	r Reg.
ADD A, n	A ← A + n	‡	‡	X	‡	X	V	0	‡	11	[000]	110		2	2	7	000 B
											← n →						001 C
																	010 D
ADD A, (HL)	A ← A + (HL)	‡	‡	X	‡	X	V	0	‡	10	[000]	110		1	2	7	011 E
ADD A, (IX + d)	A ← A + (IX + d)	‡	‡	X	‡	X	V	0	‡	11	011	101	DD	3	5	19	100 H
										10	[000]	110					101 L
											← d →						111 A
ADD A, (IY + d)	A ← A + (IY + d)	‡	‡	X	‡	X	V	0	‡	11	111	101	FD	3	5	19	
										10	[000]	110					
											← d →						
ADC A, s	A ← A + s + CY	‡	‡	X	‡	X	V	0	‡		[001]						s is any of r, n, (HL), (IX + d), (IY + d) as shown for ADD instruction. The indicated bits replace the [000] in the ADD set above.
SUB s	A ← A – s	‡	‡	X	‡	X	V	1	‡		[010]						
SBC A, s	A ← A – s – CY	‡	‡	X	‡	X	V	1	‡		[011]						
AND s	A ← A > s	‡	‡	X	1	X	P	0	0		[100]						
OR s	A ← A > s	‡	‡	X	0	X	P	0	0		[110]						
XOR s	A ← Aes	‡	‡	X	0	X	P	0	0		[101]						
CP s	A – s	‡	‡	X	‡	X	V	1	‡		[111]						

8-BIT ARITHMETIC AND LOGICAL GROUP (Continued)

Mnemonic	Symbolic Operation	S	Z		H		P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
INC r	r ← r + 1	‡	‡	X	‡	X	V	0	•	00	r	100		1	1	4	
INC (HL)	(HL) ←																
	(HL) + 1	‡	‡	X	‡	X	V	0	•	00	110	100		1	3	11	
INC (IX + d)	(IX + d) ←	‡	‡	X	‡	X	V	0	•	11	011	101	DD	3	6	23	
	(IX + d) + 1									00	110	100					
											← d →						
INC (IY + d)	(IY + d) ←	‡	‡	X	‡	X	V	0	•	11	111	101	FD	3	6	23	
	(IY + d) + 1									00	110	100					
											← d →						
DEC m	m ← m − 1	‡	‡	X	‡	X	V	1	•			101					

NOTE: m is any of r, (HL), (IX + d), (IY + d) as shown for INC. DEC same format and states as INC. Replace 100 with 101 in opcode.

GENERAL-PURPOSE ARITHMETIC AND CPU CONTROL GROUPS

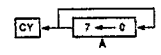
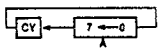
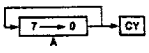
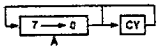
Mnemonic	Symbolic Operation	S	Z		H		P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
DAA	@	‡	‡	X	‡	X	P	•	‡	00	100	111	27	1	1	4	Decimal adjust accumulator.
CPL	A ← A	•	•	X	1	X	•	1	•	00	101	111	2F	1	1	4	Complement accumulator (one's complement).
NEG	A ← 0 − A	‡	‡	X	‡	X	V	1	‡	11	101	101	ED	2	2	8	Negate acc. (two's complement).
										01	000	100	44				
CCF	CY ← CY	•	•	X	X	X	•	0	‡	00	111	111	3F	1	1	4	Complement carry flag.
SCF	CY ← 1	•	•	X	0	X	•	0	1	00	110	111	37	1	1	4	Set carry flag.
NOP	No operation	•	•	X	•	X	•	•	•	00	000	000	00	1	1	4	
HALT	CPU halted	•	•	X	•	X	•	•	•	01	110	110	76	1	1	4	
DI ★	IFF ← 0	•	•	X	•	X	•	•	•	11	110	011	F3	1	1	4	
EI ★	IFF ← 1	•	•	X	•	X	•	•	•	11	111	011	FB	1	1	4	
IM 0	Set interrupt mode 0	•	•	X	•	X	•	•	•	11	101	101	ED	2	2	8	
										01	000	110	46				
IM 1	Set interrupt mode 1	•	•	X	•	X	•	•	•	11	101	101	ED	2	2	8	
										01	010	110	56				
IM 2	Set interrupt mode 2	•	•	X	•	X	•	•	•	11	101	101	ED	2	2	8	
										01	011	110	5E				

NOTES: @ converts accumulator content into packed BCD following add or subtract with packed BCD operands.
IFF indicates the interrupt enable flip-flop.
CY indicates the carry flip-flop.
★ indicates interrupts are not sampled at the end of EI or DI.

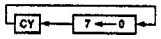
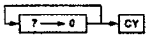
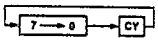
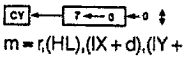
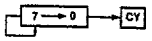
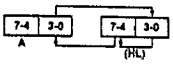
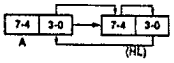
16-BIT ARITHMETIC GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD HL, ss	HL ← HL + ss	•	•	X	X	X	•	0	‡	00	ss1	001		1	3	11	ss Reg.
																	00 BC
ADC HL, ss	HL ←																01 DE
	HL + ss + CY	‡	‡	X	X	X	V	0	‡	11	101	101	ED	2	4	15	10 HL
										01	ss1	010					11 SP
SBC HL, ss	HL ←																
	HL − ss − CY	‡	‡	X	X	X	V	1	‡	11	101	101	ED	2	4	15	
										01	ss0	010					
ADD IX, pp	IX ← IX + pp	•	•	X	X	X	•	0	‡	11	011	101	DD	2	4	15	pp Reg.
										01	pp1	001					00 BC
																	01 DE
																	10 IX
																	11 SP
ADD IY, rr	IY ← IY + rr	•	•	X	X	X	•	0	‡	11	111	101	FD	2	4	15	rr Reg.
										00	rr1	001					00 BC
INC ss	ss ← ss + 1	•	•	X	•	X	•	•	•	00	ss0	011		1	1	6	01 DE
INC IX	IX ← IX + 1	•	•	X	•	X	•	•	•	11	011	101	DD	2	2	10	10 IY
										00	100	011	23				11 SP
INC IY	IY ← IY + 1	•	•	X	•	X	•	•	•	11	111	101	FD	2	2	10	
										00	100	011	23				
DEC ss	ss ← ss − 1	•	•	X	•	X	•	•	•	00	ss1	011		1	1	6	
DEC IX	IX ← IX − 1	•	•	X	•	X	•	•	•	11	011	101	DD	2	2	10	
										00	101	011	2B				
DEC IY	IY ← IY − 1	•	•	X	•	X	•	•	•	11	111	101	FD	2	2	10	
										00	101	011	2B				

ROTATE AND SHIFT GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLCA		•	•	X	0	X	•	0	‡	00	000	111	07	1	1	4	Rotate left circular accumulator.
RLA		•	•	X	0	X	•	0	‡	00	010	111	17	1	1	4	Rotate left accumulator.
RRCA		•	•	X	0	X	•	0	‡	00	001	111	0F	1	1	4	Rotate right circular accumulator.
RRA		•	•	X	0	X	•	0	‡	00	011	111	1F	1	1	4	Rotate right accumulator.

ROTATE AND SHIFT GROUP (Continued)

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLC r		‡	‡	X	0	X	P	0	•	‡	11 00	001 000	011 r	CB	2	2	8	Rotate left circular register r
RLC (IX+d)	r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡	11 11 00	011 001 ←d→ 000	101 011 110	DD CB	4	6	23	010 D 011 E 001 H 101 L 111 A
RLC (IY+d)		‡	‡	X	0	X	P	0	‡	11 11 00	111 001 ←d→ 000	101 011 110	FD CB	4	6	23	
RL m	 m = r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡		010						Instruction format and states are as shown for RLCs. To form new opcode replace 000 or RLCs with shown code.
RRC m	 m = r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡		001						
RR m	 m = r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡		011						
SLA m	 m = r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡		100						
SRA m	 m = r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡		101						
SRL m	 m = r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡		111						
RLD		‡	‡	X	0	X	P	0	•	11 01	101 101	101 111	ED 6F	2	5	18	Rotate digit left and right between the accumulator and location (HL). The content of the upper half of the accumulator is unaffected.
RRD		‡	‡	X	0	X	P	0	•	11 01	101 100	101 111	ED 67	2	5	18	

BIT SET, RESET AND TEST GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
BIT b, r	$Z \leftarrow r_b$	X	‡	X 1	X	X	0	• 11	001	011	CB	2	2	8	r Reg.
								01	b	r					000 B
BIT b, (HL)	$Z \leftarrow (HL)_b$	X	‡	X 1	X	X	0	• 11	001	011	CB	2	3	12	001 C
								01	b	110					010 D
BIT b, (IY + d)	$Z \leftarrow (IY+d)_b$	X	‡	X 1	X	X	0	• 11	011	101	DD	4	5	20	011 E
									← d →						101 L
								01	b	110					111 A
															b Bit Tested
BIT b, $(IY+d)_b$	$Z \leftarrow (IY+d)_b$	X	‡	X 1	X	X	0	• 11	111	101	FD	4	5	20	000 0
								11	001	011	CB				001 1
									← d →						010 2
								01	b	110					011 3
SET b, r	$r_b \leftarrow 1$	•	•	X •	X	•	•	• 11	001	011	CB	2	2	8	100 4
								11	b	r					101 5
SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	X •	X	•	•	• 11	001	011	CB	2	4	15	110 6
								11	b	110					111 7
SET b, (IX + d)	$(IX+d)_b \leftarrow 1$	•	•	X •	X	•	•	• 11	011	101	DD	4	6	23	
								11	001	011	CB				
									← d →						
								11	b	110					
SET b, (IY + d)	$(IY+d)_b \leftarrow 1$	•	•	X •	X	•	•	• 11	111	101	FD	4	6	23	
								11	001	011	CB				
									← d →						
								11	b	110					
RES b, m	$m_b \leftarrow 0$ m ≡ r, (HL), (IX + d), (IY + d)	•	•	X •	X	•	•	• 10							To form new opcode replace 11 of SET b, s with 10. Flags and time states for SET instruction.

NOTE: The notation m_b indicates location m, bit b (0 to 7).

JUMP GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
JP nn	PC ← nn	•	•	X	•	X	•	•	•	11	000	011	C3	3	3	10	cc Condition
										← n →							000 NZ (non-zero)
										← n →							001 Z (zero)
JP cc, nn	If condition cc is true PC ← nn, otherwise continue	•	•	X	•	X	•	•	•	11	cc	010		3	3	10	010 NC (non-carry)
										← n →							011 C (carry)
										← n →							100 PO (parity odd)
																	101 PE (parity even)
JR e	PC ← PC + e	•	•	X	•	X	•	•	•	00	011	000	18	2	3	12	110 P (sign positive)
										← e − 2 →							111 M (sign negative)
JR C, e	If C = 0, continue	•	•	X	•	X	•	•	•	00	111	000	38	2	2	7	If condition not met.
										← e − 2 →							
	If C = 1, PC ← PC + e													2	3	12	If condition is met.
JR NC, e	IF C = 1, [illegible] PC ← PC + e	•	•	X	•	X	•	•	•	00	110	000	30	2	2	7	If condition not met.
JP Z, e	If Z = 0 continue	•	•	X	•	X	•	•	•	00	101	000	28	2	2	7	If condition not met.
										← e − 2 →							
	If Z = 1, PC ← PC + e													2	3	12	If condition is met.
JR NZ, e	If Z = 1, continue	•	•	X	•	X	•	•	•	00	100	000	20	2	2	7	If condition not met.
										← e − 2 →							
	If Z = 0, PC ← PC + e													2	3	12	If condition is met.
JP (HL)	PC ← HL	•	•	X	•	X	•	•	•	11	101	001	E9	1	1	4	
JP (IX)	PC ← IX	•	•	X	•	X	•	•	•	11	011	101	DD	2	2	8	
										11	101	001	E9				
JP (IY)	PC ← IY	•	•	X	•	X	•	•	•	11	111	101	FD	2	2	8	
										11	101	001	E9				
DJNZ, e	B ← B − 1 [illegible]	•	•	X	•	X	•	•	•	00	010	000	10	2	2	8	If B = 0
	If B = 0, continue									← e − 2 →							
	If B≠0, PC ← PC + e													2	3	13	If B≠0.

NOTES: e represents the extension in the relative addressing mode.
e is a signal two's complement number in the range < − 126, 129 >.
e − 2 in the opcode provides an effective address of pc + e as PC is incremented by 2 prior to the addition of e.

CALL AND RETURN GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/VN	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
CALL nn	$(SP-1)\leftarrow PC_H$ $(SP-2)\leftarrow PC_L$ $PC \leftarrow nn$,	•	•	X	•	X	•	•	•	11	001 ←n→ ←n→	101	CD	3	5	17	
CALL cc,nn	If condition cc is false	•	•	X	•	X	•	•	•	11	cc ←n→ ←n→	100		3	3	10	If cc is false.
	continue, otherwise same as CALL nn													3	5	17	If cc is true.
RET	$PC_L \leftarrow (SP)$ $PC_H \leftarrow (SP+1)$	•	•	X	•	X	•	•	•	11	001	001	C9	1	3	10	
RET cc	If condition cc is false	•	•	X	•	X	•	•	•	11	cc	000		1	1	5	If cc is false.
	continue, otherwise same as RET													1	3	11	If cc is true.
RETI	Return from interrupt	•	•	X	•	X	•	•	•	11 01	101 001	101 101	ED 4D	2	4	14	
RETN[1]	Return from non-maskable interrupt	•	•	X	•	X	•	•	•	11 01	101 000	101 101	ED 45	2	4	14	
RST p	$(SP-1)\leftarrow PC_H$ $(SP-2)\leftarrow PC_L$ $PC_H \leftarrow 0$ $PC_L \leftarrow p$	•	•	X	•	X	•	•	•	11	t	111		1	3	11	

cc	Condition
000	NZ (non-zero)
001	Z (zero)
010	NC (non-carry)
011	C (carry)
100	PO (parity odd)
101	PE (parity even)
110	P (sign positive)
111	M (sign negative)

t	p
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

NOTE: [1]RETN loads $IFF_2 \rightarrow IFF_1$

INPUT AND OUTPUT GROUP

Mnemonic	Symbolic Operation	Flags S	Z	H	P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
IN A, (n)	A ← (n)	•	•	X	•	X	•	•	•	11	011 ←n→	01	DB	2	3	11	n to $A_0 \sim A_7$ Acc. to $A_8 \sim A_{15}$
IN r, (C)	r ← (C) if r = 110 only the flags will be affected	‡	‡	X	‡	X	P	0	•	11 01	101 r	101 000	ED	2	3	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INI	(HL) ← (C) B ← B – 1 HL ← HL + 1	X	‡ ①	X	X	X	X	1	X	11 10	101 100	101 010	ED A2	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INIR	(HL) ← (C) B ← B – 1 HL ← HL + 1 Repeat until B = 0	X	1 ②	X	X	X	X	1	X	11 10	101 110	101 010	ED B2	2 2	5 (If B≠0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
IND	(HL) ← (C) B ← B – 1 HL ← HL – 1	X	‡ ①	X	X	X	X	1	X	11 10	101 101	101 010	ED AA	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INDR	(HL) ← (C) B ← B – 1 HL ← HL – 1 Repeat until B = 0	X	1 ②	X	X	X	X	1	X	11 10	101 111	101 010	ED BA	2 2	5 (If B≠0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUT (n), A	(n) ← A	•	•	X	•	X	•	•	•	11	010 ←n→	011	D3	2	3	11	n to $A_0 \sim A_7$ Acc. to $A_8 \sim A_{15}$
OUT (C), r	(C) ← r	•	•	X	•	X	•	•	•	11 01	101 r	101 001	ED	2	3	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTI	(C) ← (HL) B ← B – 1 HL ← HL + 1	X	‡ ①	X	X	X	X	1	X	11 10	101 100	101 011	ED A3	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTIR	(C) ← (HL) B ← B – 1 HL ← HL + 1 Repeat until B = 0	X	1 ②	X	X	X	X	1	X	11 10	101 110	101 011	ED B3	2 2	5 (If B≠0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTD	(C) ← (HL) B ← B – 1 HL ← HL – 1	X	‡ ①	X	X	X	X	1	X	11 10	101 101	101 011	ED AB	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTDR	(C) ← (HL) B ← B – 1 HL ← HL – 1 Repeat until B = 0	X	1 ②	X	X	X	X	1	X	11 10	101 111	101 011	ED BB	2 2	5 (If B≠0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$

NOTES: ① If the result of B – 1 is zero, the Z flag is set; otherwise it is reset.
② Z flag is set upon instruction completion only.

SUMMARY OF FLAG OPERATION

Instructions	D_7 S	Z		H		P/V	N	D_0 C	Comments
ADD A, s; ADC A, s	‡	‡	X	‡	X	V	0	‡	8-bit add or add with carry.
SUB s; SBC A, s; CP s; NEG	‡	‡	X	‡	X	V	1	‡	8-bit subtract, subtract with carry, compare and negate accumulator.
AND s	‡	‡	X	1	X	P	0	0	Logical operation.
OR s, XOR s	‡	‡	X	0	X	P	0	0	Logical operation.
INC s	‡	‡	X	‡	X	V	0	•	8-bit increment.
DEC s	‡	‡	X	‡	X	V	1	•	8-bit decrement.
ADD DD, ss	•	•	X	X	X	•	0	‡	16-bit add.
ADC HL, ss	‡	‡	X	X	X	V	0	‡	16-bit add with carry.
SBC HL, ss	‡	‡	X	X	X	V	1	‡	16-bit subtract with carry.
RLA; RLCA; RRA; RRCA	•	•	X	0	X	•	0	‡	Rotate accumulator.
RL m; RLC m; RR m; RRC m; SLA m; SRA m; SRL m	‡	‡	X	0	X	P	0	‡	Rotate and shift locations.
RLD; RRD	‡	‡	X	0	X	P	0	•	Rotate digit left and right.
DAA	‡	‡	X	‡	X	P	•	‡	Decimal adjust accumulator.
CPL	•	•	X	1	X	•	1	•	Complement accumulator.
SCF	•	•	X	0	X	•	0	1	Set carry.
CCF	•	•	X	X	X	•	0	‡	Complement carry.
IN r (C)	‡	‡	X	0	X	P	0	•	Input register indirect.
INI; IND; OUTI; OUTD	X	‡	X	X	X	X	1	•	Block input and output. Z = 1 if B ≠ 0, otherwise Z = 0.
INIR; INDR; OTIR; OTDR	X	1	X	X	X	X	1	•	Block input and output. Z = 1 if B ≠ 0, otherwise Z = 0.
LDI; LDD	X	X	X	0	X	‡	0	•	Block transfer instructions. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
LDIR; LDDR	X	X	X	0	X	0	0	•	Block transfer instructions. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
CPI; CPIR; CPD; CPDR	X	‡	X	X	X	‡	1	•	Block search instructions. Z = 1 if A = (HL), otherwise Z = 0. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
LD A; I, LD A, R	‡	‡	X	0	X	IFF	0	•	IFF, the content of the interrupt enable flip-flop, (IFF_2), is copied into the P/V flag.
BIT b, s	X	‡	X	1	X	X	0	•	The state of bit b of location s is copied into the Z flag.

SYMBOLIC NOTATION

Symbol	Operation
S	Sign flag. S = 1 if the MSB of the result is 1.
Z	Zero flag. Z = 1 if the result of the operation is 0.
P/V	Parity or overflow flag. Parity (P) and overflow (V) share the same flag. Logical operations affect this flag with the parity of the result while arithmetic operations affect this flag with the overflow of the result. If P/V holds parity: P/V = 1 if the result of the operation is even; P/V = 0 if result is odd. If P/V holds overflow, P/V = 1 if the result of the operation produced an overflow. If P/V does not hold overflow, P/V = 0.
H*	Half-carry flag. H = 1 if the add or subtract operation produced a carry into, or borrow from, bit 4 of the accumulator.
N*	Add/Subtract flag. N = 1 if the previous operation was a subtract.
C	Carry/Link flag. C = 1 if the operation produced a carry from the MSB of the operand or result.

Symbol	Operation
‡	The flag is affected according to the result of the operation.
•	The flag is unchanged by the operation.
0	The flag is reset by the operation.
1	The flag is set by the operation.
X	The flag is indeterminate.
V	P/V flag affected according to the overflow result of the operation.
P	P/V flag affected according to the parity result of the operation.
r	Any one o the CPU registers A, B, C, D, E, H, L.
s	Any 8-bit location for all the addressing modes allowed for the particular instruction.
ss	Any 16-bit location for all the addressing modes allowed for that instruction.
ii	Any one of the two index registers IX or IY.
R	Refresh counter.
n	8-bit value in range < 0, 255 >.
nn	16-bit value in range < 0, 65535 >.

*H and N flags are used in conjunction with the decimal adjust instruction (DAA) to properly correct the result into packed BCD format following addition or subtraction using operands with packed BCD format.

CPU REGISTERS

Figure 4 shows three groups of registers within the CPU. The first group consists of duplicate sets of 8-bit registers: a principal set and an alternate set [designated by ' (prime), e.g., A']. Both sets consist of the Accumulator register, the Flag register, and six general-purpose registers. Transfer of data between these duplicate sets of registers is accomplished by use of "Exchange" instructions. The result is faster response to interrupts and easy, efficient implementation of such versatile programming techniques as background-foreground data processing. The second set of registers consists of six registers with assigned functions. These are the I (Interrupt register), the R (Refresh register), the IX and IY (Index registers), the SP (Stack Pointer), and the PC (Program Counter). The third group consists of two interrupt status flip-flops, plus an additional pair of flip-flops which assists in identifying the interrupt mode at any particular time. Table 1 provides further information on these registers.

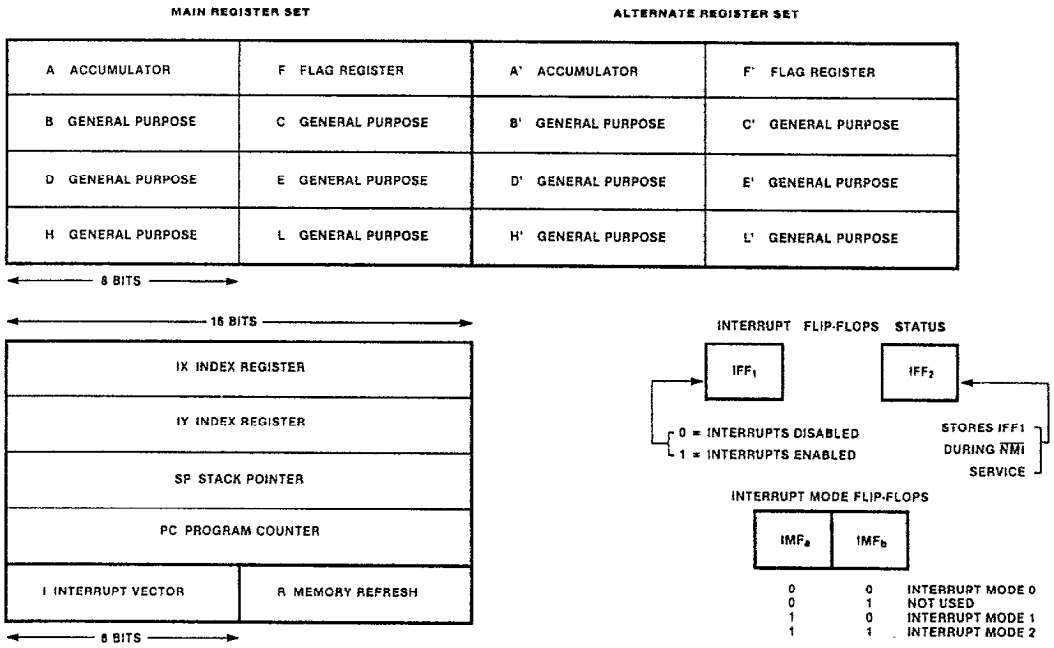


Figure 4. CPU Registers

INTERRUPTS: GENERAL OPERATION

The CPU accepts two interrupt input signals: $\overline{NMI}$ and $\overline{INT}$. The $\overline{NMI}$ is a non-maskable interrupt and has the highest priority. $\overline{INT}$ is a lower priority interrupt and it requires that interrupts be enabled in software in order to operate. $\overline{INT}$ can be connected to multiple peripheral devices in a wired-OR configuration.

The Z80 has a single response mode for interrupt service on the non-maskable interrupt. The maskable interrupt, $\overline{INT}$, has three programmable response modes available. These are:

- Mode 0 — similar to the 8080 microprocessor.
- Mode 1 — Peripheral Interrupt service, for use with non-8080/Z80 systems.
- **Mode 2 - a vectored interrupt scheme, usually daisy-chained, for use with the Z80 Family and compatible peripheral devices.**

The CPU services interrupts by sampling the $\overline{NMI}$ and $\overline{INT}$ signals at the rising edge of the last clock of an instruction. Further interrupt service processing depends upon the type of interrupt that was detected. Details on interrupt responses are shown in the CPU Timing Section.

Non-Maskable Interrupt ($\overline{NMI}$). The nonmaskable interrupt cannot be disabled by program control and therefore will be accepted at all times by the CPU. $\overline{NMI}$ is usually reserved for servicing only the highest priority type interrupts, such as that for orderly shutdown after power

PIN DESCRIPTIONS

A_0-A_{15}. *Address Bus* (output, active High, 3-state). A_0-A_{15} form a 16-bit address bus. The Address Bus provides the address for memory data bus exchanges (up to 64K bytes) and for I/O device exchanges.

$\overline{\text{BUSACK}}$. *Bus Acknowledge* (output, active Low). Bus Acknowledge indicates to the requesting device that the CPU address bus, data bus, and control signals $\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$ have entered their high-impedance states. The external circuitry can now control these lines.

$\overline{\text{BUSREQ}}$. *Bus Request* (input, active Low). Bus Request has a higher priority than $\overline{\text{NMI}}$ and is always recognized at the end of the current machine cycle. $\overline{\text{BUSREQ}}$ forces the CPU address bus, data bus, and control signals $\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, RD, and WR to go to a high-impedance state so that other devices can control these lines. $\overline{\text{BUSREQ}}$ is normally wired-OR and requires an external pullup for these applications. Extended $\overline{\text{BUSREQ}}$ periods due to extensive DMA operations can prevent the CPU from properly refreshing dynamic RAMs.

D_0-D_7. *Data Bus* (input/output, active High, 3-state). D_0-D_7 constitute an 8-bit bidirectional data bus, used for data exchanges with memory and I/O.

$\overline{\text{HALT}}$. *Halt State* (output, active Low). $\overline{\text{HALT}}$ indicates that the CPU has executed a Halt instruction and is awaiting either a nonmaskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOPs to maintain memory refresh.

$\overline{\text{INT}}$. *Interrupt Request* (input, active Low). Interrupt Request is generated by I/O devices. The CPU honors a request at the end of the current instruction if the internal software-controlled interrupt enable flip-flop (IFF) is enabled. $\overline{\text{INT}}$ is normally wired-OR and requires an external pullup for these applications.

$\overline{\text{IORQ}}$. *Input/Output Request* (output, active Low, 3-state). $\overline{\text{IORQ}}$ indicates that the lower half of the address bus holds a valid I/O address for an I/O read or write operation. $\overline{\text{IORQ}}$ is also generated concurrently with $\overline{\text{M1}}$ during an interrupt acknowledge cycle to indicate that an interrupt response vector can be placed on the data bus.

$\overline{\text{M1}}$. *Machine Cycle One* (output, active Low). $\overline{\text{M1}}$, together with $\overline{\text{MREQ}}$, indicates that the current machine cycle is the opcode fetch cycle of an instruction execution. $\overline{\text{M1}}$, together with $\overline{\text{IORQ}}$, indicates an interrupt acknowledge cycle.

$\overline{\text{MREQ}}$. *Memory Request* (output, active Low, 3-state). $\overline{\text{MREQ}}$ indicates that the address bus holds a valid address for a memory read or memory write operation.

$\overline{\text{NMI}}$. *Non-Maskable Interrupt* (input, negative edge-triggered). $\overline{\text{NMI}}$ has a higher priority than $\overline{\text{INT}}$. $\overline{\text{NMI}}$ is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop, and automatically forces the CPU to restart at location 0066H.

$\overline{\text{RD}}$. *Read* (output, active Low, 3-state). $\overline{\text{RD}}$ indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.

$\overline{\text{RESET}}$. *Reset* (input, active Low). $\overline{\text{RESET}}$ initializes the CPU as follows: it resets the interrupt enable flip-flop, clears the PC and Registers I and R, and sets the interrupt status to Mode 0. During reset time, the address and data bus go to a high-impedance state, and all control output signals go to the inactive state. Note that $\overline{\text{RESET}}$ must be active for a minimum of three full clock cycles before the reset operation is complete.

$\overline{\text{RFSH}}$. *Refresh* (output, active Low). $\overline{\text{RFSH}}$, together with $\overline{\text{MREQ}}$, indicates that the lower seven bits of the system's address bus can be used as a refresh address to the system's dynamic memories.

$\overline{\text{WAIT}}$. *Wait* (input, active Low). $\overline{\text{WAIT}}$ indicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter a Wait state as long as this signal is active. Extended $\overline{\text{WAIT}}$ periods can prevent the CPU from properly refreshing dynamic memory.

$\overline{\text{WR}}$. *Write* (output, active Low, 3-state). $\overline{\text{WR}}$ indicates that the CPU data bus holds valid data to be stored at the addressed memory or I/O location.

CPU TIMING

The Z80 CPU executes instructions by proceeding through a specific sequence of operations:

- Memory read or write
- I/O device read or write
- Interrupt acknowledge

The basic clock period is referred to as a T time or cycle, and three or more T cycles make up a machine cycle (M1, M2 or M3 for instance). Machine cycles can be extended either by the CPU automatically inserting one or more Wait states or by the insertion of one or more Wait states by the user.

Instruction Opcode Fetch. The CPU places the contents of the Program Counter (PC) on the address bus at the start of the cycle (Figure 5). Approximately one-half clock cycle later, $\overline{MREQ}$ goes active. When active, $\overline{RD}$ indicates that the memory data can be enabled onto the CPU data bus.

The CPU samples the $\overline{WAIT}$ input with the falling edge of clock state T_2. During clock states T_3 and T_4 of an $\overline{M1}$ cycle, dynamic RAM refresh can occur while the CPU starts decoding and executing the instruction. When the Refresh Control signal becomes active, refreshing of dynamic memory can take place.

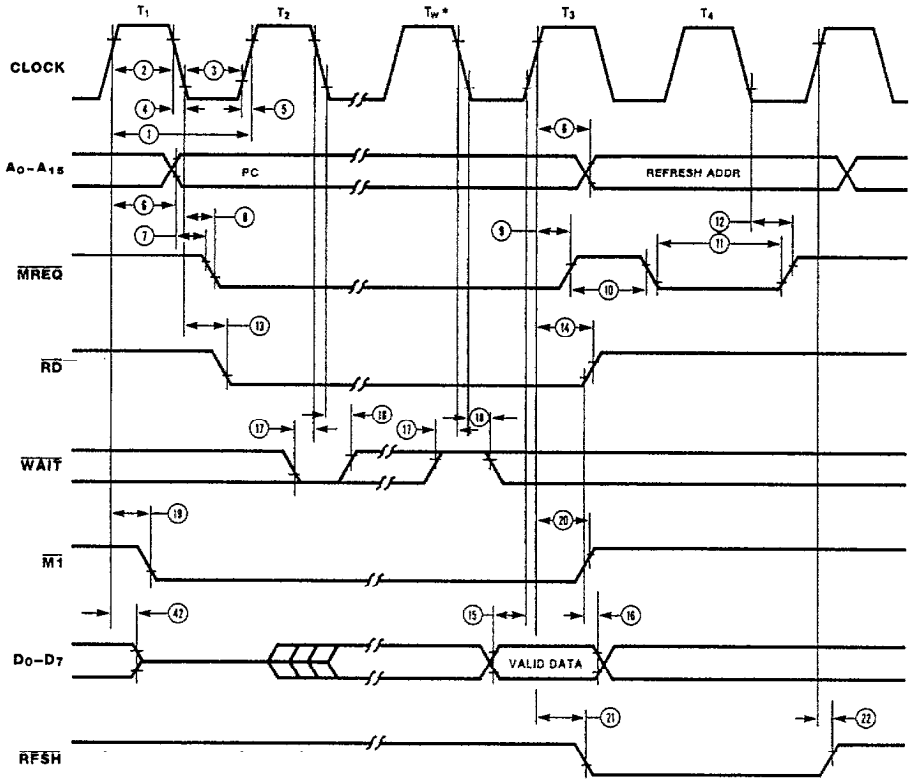


Figure 5. Instruction Opcode Fetch

Memory Read or Write Cycles. Figure 6 shows the timing of memory read or write cycles other than an opcode fetch ($\overline{M1}$) cycle. The $\overline{MREQ}$ and $\overline{RD}$ signals function exactly as in the fetch cycle. In a memory write cycle, $\overline{MREQ}$ also becomes active when the address bus is stable. The $\overline{WR}$ line is active when the data bus is stable, so that it can be used directly as an R/$\overline{W}$ pulse to most semiconductor memories.

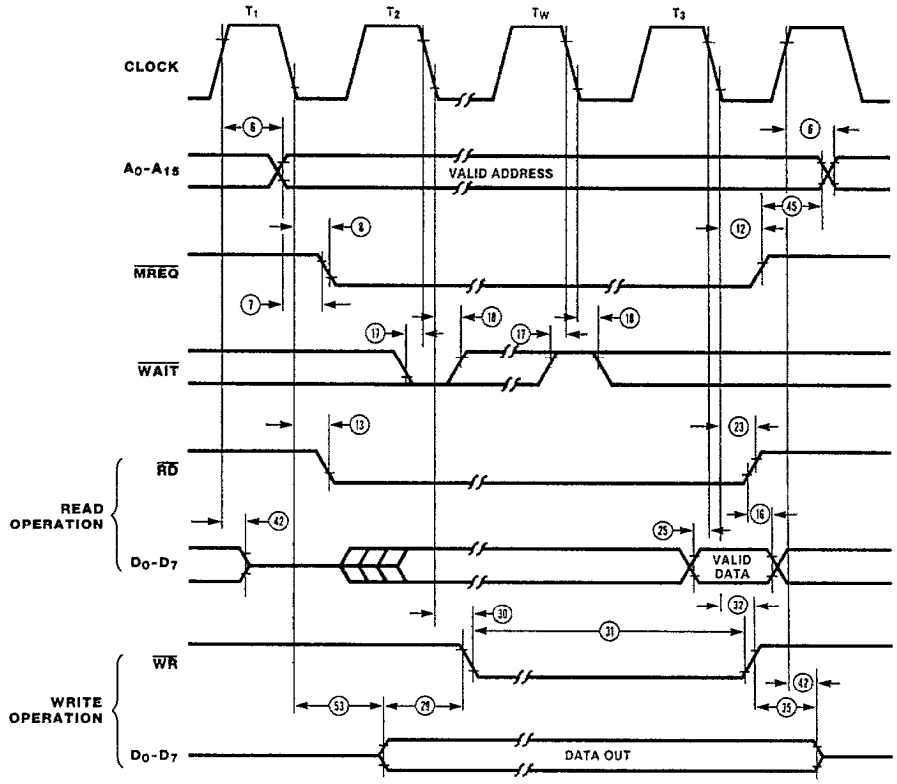
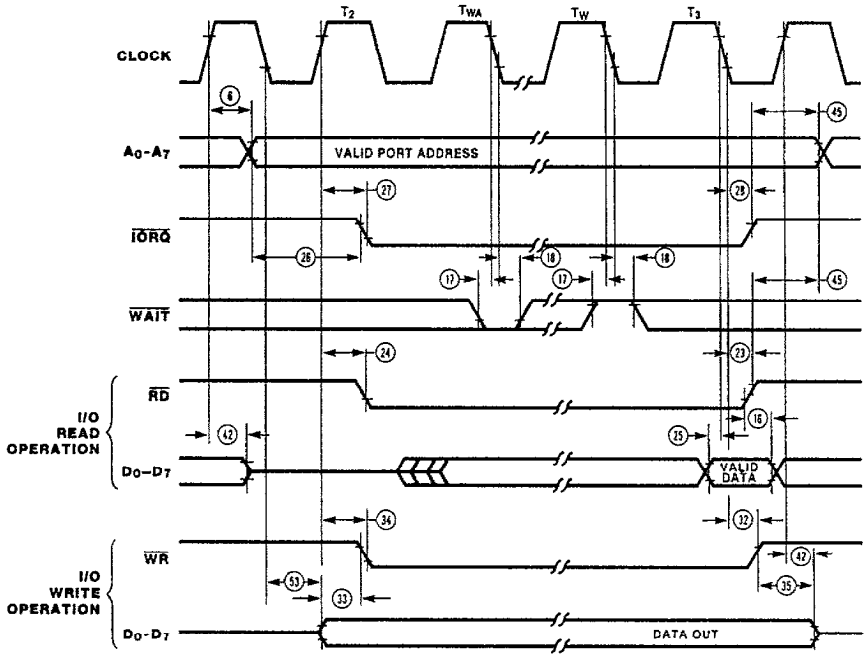


Figure 6. Memory Read or Write Cycles

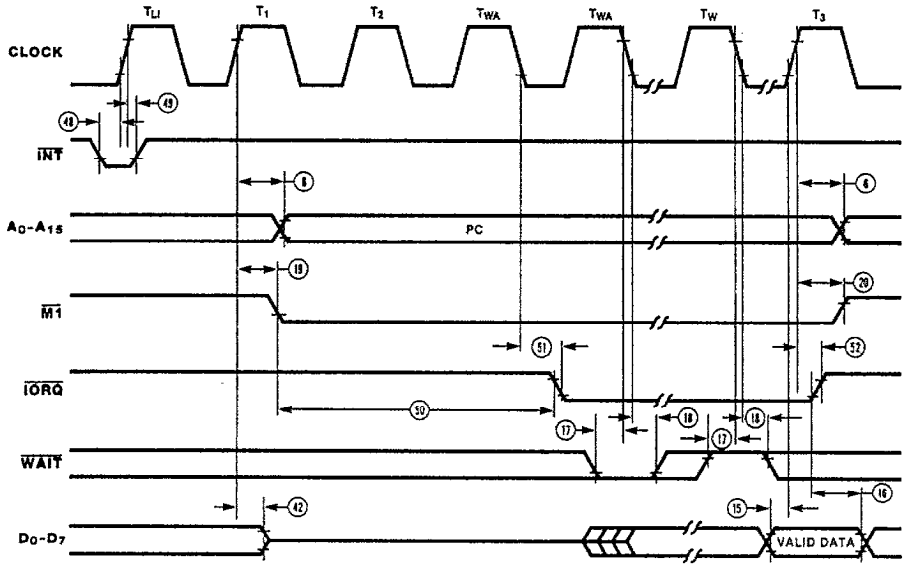
Input or Output Cycles. *Figure 7* shows the timing for an I/O read or I/O write operation. During I/O operations, the CPU automatically inserts a single Wait state (T_{WA}). This extra Wait state allows sufficient time for an I/O port to decode the address from the port address lines.



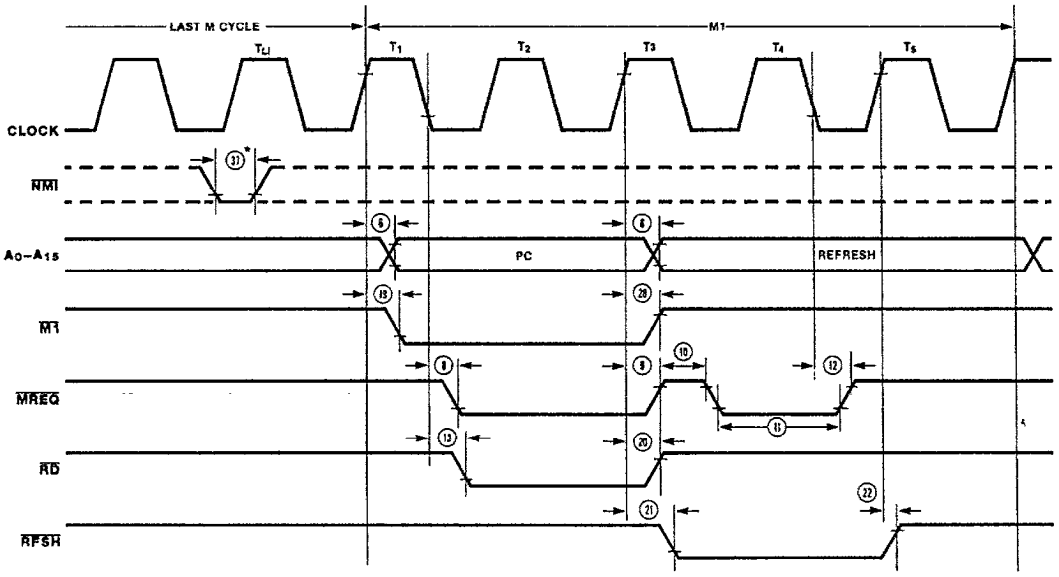
T_{WA} = One wait cycle automatically inserted by CPU.

Figure 7. Input or Output Cycles

Interrupt Request/Acknowledge Cycle. The CPU samples the interrupt signal with the rising edge of the last clock cycle at the end of any instruction (Figure 8). When an interrupt is accepted, a special $\overline{M1}$ cycle is generated. During this $\overline{M1}$ cycle, $\overline{IORQ}$ becomes active (instead of $\overline{MREQ}$) to indicate that the interrupting device can place an 8-bit vector on the data bus. The CPU automatically adds two Wait states to this cycle.



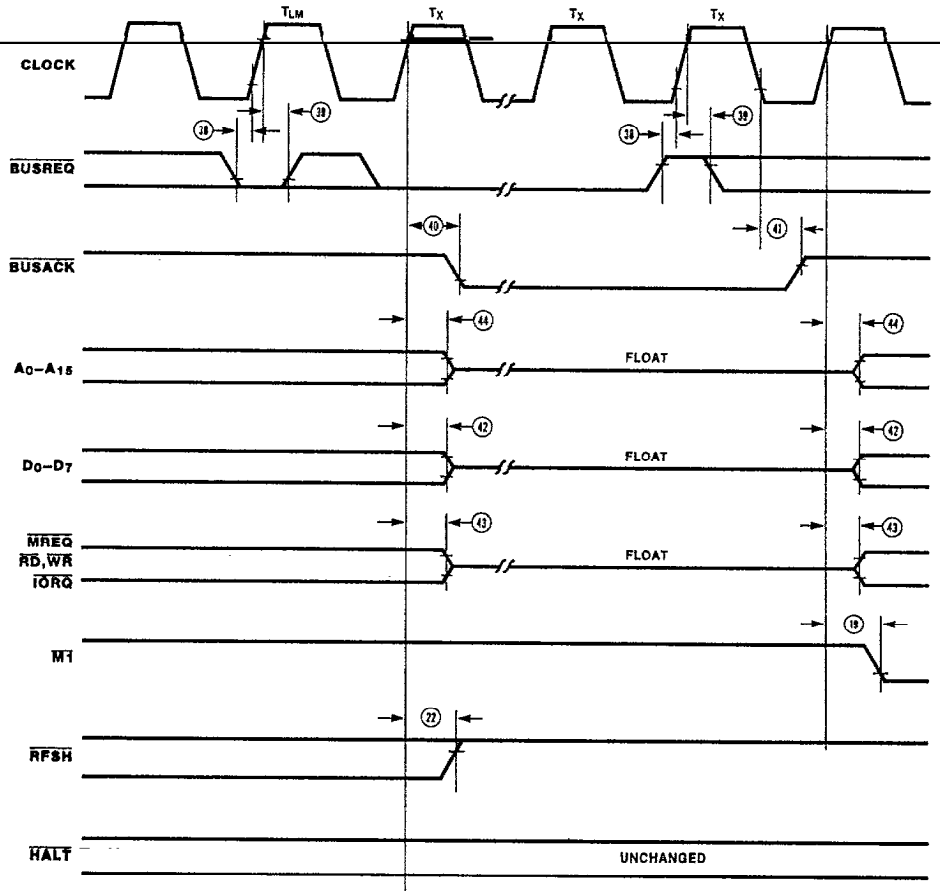
Non-Maskable Interrupt Request Cycle. $\overline{NMI}$ is sampled at the same time as the maskable interrupt input $\overline{INT}$ but has higher priority and cannot be disabled under software control. The subsequent timing is similar to that of a normal memory read operation except that data put on the bus by the memory is ignored. The CPU instead executes a restart (RST) operation and jumps to the $\overline{NMI}$ service routine located at address 0066H (Figure 9).



*Although $\overline{NMI}$ is an asynchronous input, to guarantee its being recognized on the following machine cycle, $\overline{NMI}$'s falling edge must occur no later than the rising edge of the clock cycle preceding the last state of any instruction cycle (T_{LI}).

Figure 9. Non-Maskable Interrupt Request Operation

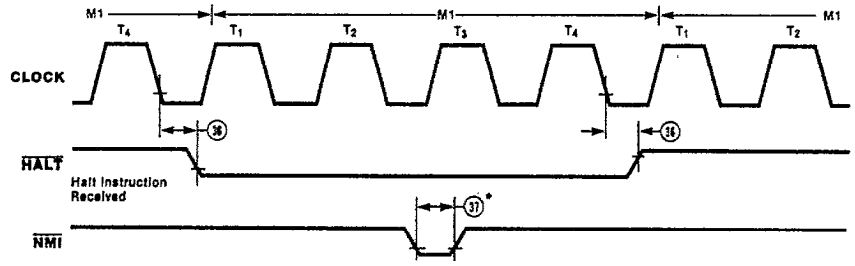
Bus Request/Acknowledge Cycle. The CPU samples $\overline{BUSREQ}$ with the rising edge of the last clock period of any machine cycle (Figure 10). If $\overline{BUSREQ}$ is active, the CPU sets its address, data, and $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$ lines to a high-impedance state with the rising edge of the next clock pulse. At that time, any external device can take control of these lines, usually to transfer data between memory and I/O devices.



NOTES: 1) T_{LM} = Last state of any M cycle.
2) T_X = An arbitrary clock cycle used by requesting device.

Figure 10. BUS Request/Acknowledge Cycle

Halt Acknowledge Cycle. When the CPU receives a HALT instruction, it executes NOP states until either an $\overline{INT}$ or $\overline{NMI}$ input is received. When in the Halt state, the $\overline{HALT}$ output is active and remains so until an interrupt is received (Figure 11). $\overline{INT}$ will also force a Halt exit.



*Although $\overline{NMI}$ is an asynchronous input, to guarantee its being recognized on the following machine cycle, $\overline{NMI}$'s falling edge must occur no later than the rising edge of the clock cycle preceding the last state of any instruction cycle (T_{LI}).

Figure 11. Halt Acknowledge

Reset Cycle. $\overline{RESET}$ must be active for at least three clock cycles for the CPU to properly accept it. As long as $\overline{RESET}$ remains active, the address and data buses float, and the control outputs are inactive. Once $\overline{RESET}$ goes inactive, two internal T cycles are consumed before the CPU resumes normal processing operation. $\overline{RESET}$ clears the PC register, so the first opcode fetch will be to location 0000H (Figure 12).

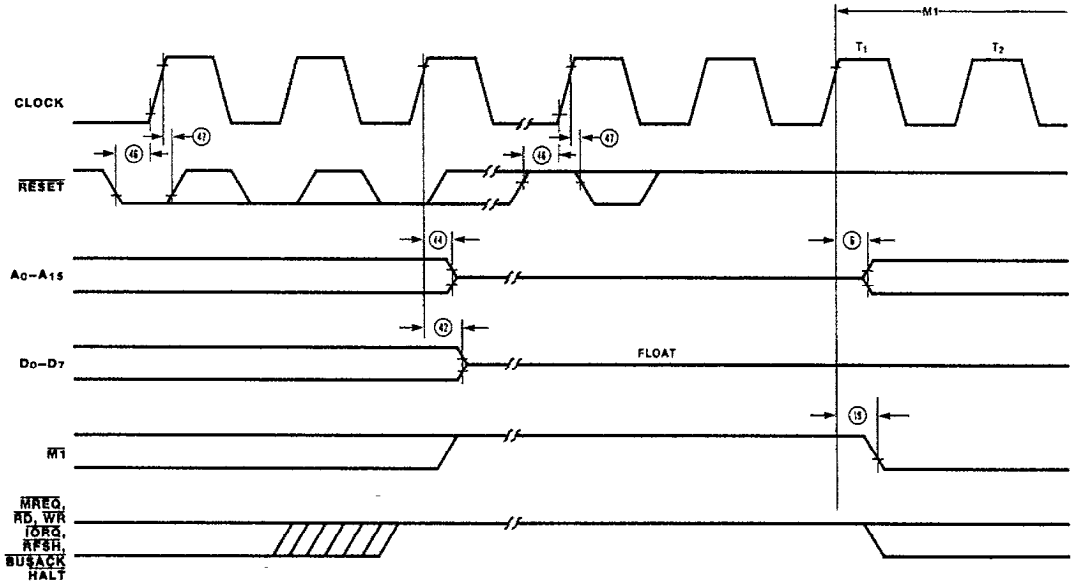


Figure 12. Reset Cycle

Power-Down mode of operation (Only applies to CMOS Z80 CPU).

CMOS Z80 CPU supports Power-Down mode of operation. This mode is also referred to as the "standby mode", and supply current for the CPU goes down as low as 10 uA (Where specified as Icc_2).

Power-Down Acknowledge Cycle. When the clock input to the CPU is stopped at either a High or Low level, the CPU stops its operation and maintains all registers and control signals. However, I_{CC2} (standby supply current) is guaranteed only when the system clock is stopped at a Low level during T_4 of the machine cycle following the execution of the HALT instruction. The timing diagram for the power-down function, when implemented with the HALT instruction, is shown in Figure 13.

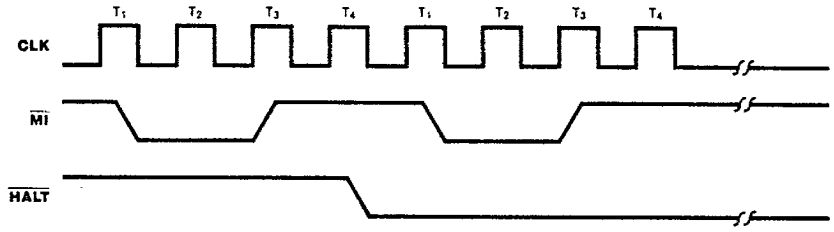


Figure 13. Power-Down Acknowledge

Power-Down Release Cycle. The system clock must be supplied to the CPU to release the power-down state. When the system clock is supplied to the CLK input, the CPU restarts operations from the point at which the power-down state was implemented.

The timing diagrams for the release from power-down mode are shown in Figure 14.

NOTES:

1) When the external oscillator has been stopped to enter the power-down state, some warm-up time may be required to obtain a stable clock for the release.
2) When the HALT instruction is executed to enter the power-down state, the CPU will also enter the Halt state. An interrupt signal (either $\overline{NMI}$ or $\overline{INT}$) or a $\overline{RESET}$ signal must be applied to the CPU after the system clock is supplied in order to release the power-down state.

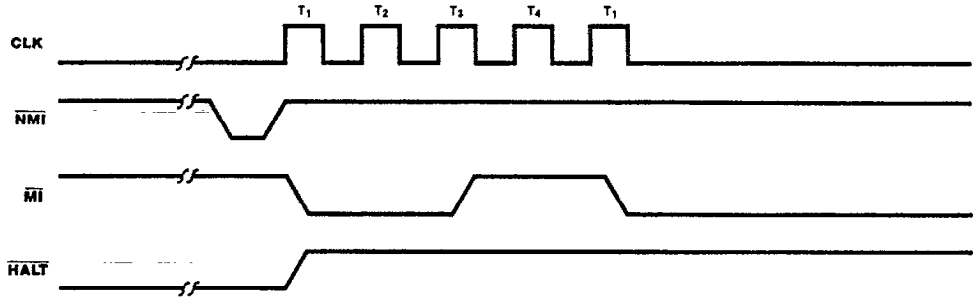


Figure 14a.

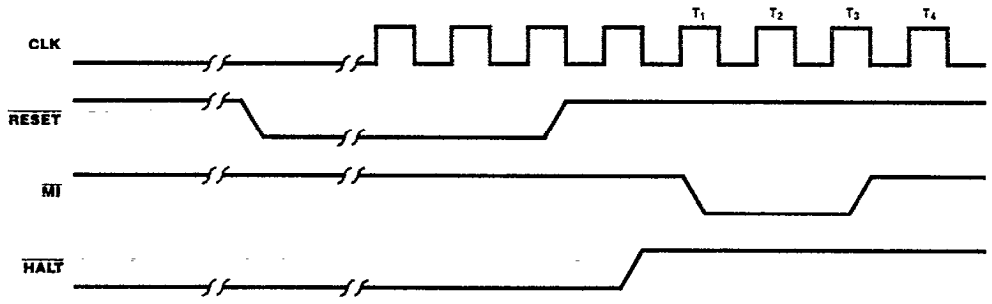


Figure 14b.

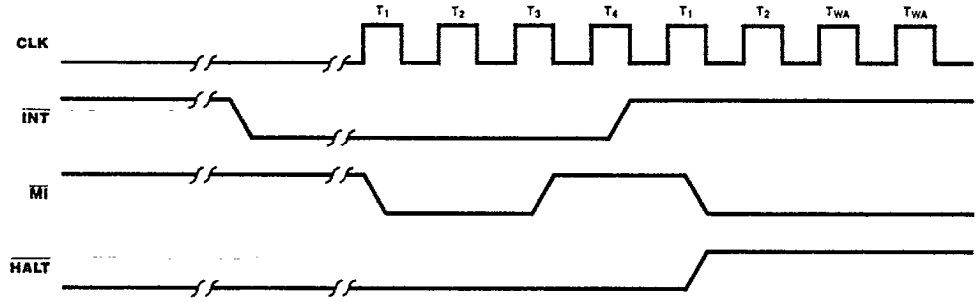


Figure 14c.

Figure 13. Power-Down Release

ABSOLUTE MAXIMUM RATINGS

Voltages on all inputs with respect to V_{SS} ..... −0.3V to V_{CC} + 0.3V
Operating Ambient Temperature ..... See Ordering Information
Storage Temperature ..... −65°C to +150°C

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.

1

STANDARD TEST CONDITIONS

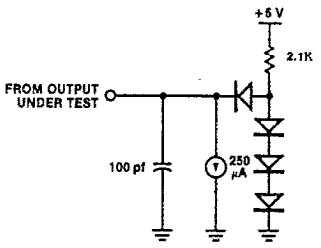
The DC Characteristics and capacitance sections below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0V).

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

Available operating temperature ranges are:

- **S = 0°C to +70°C**
  Voltage Supply Range:
  NMOS: +4.75V ≤ VCC ≤ +5.25V
  CMOS: +4.50V ≤ VCC ≤ +5.50V
- E= -40°C to 100°C, +4.50V ≤ VCC ≤ +5.50V

All ac parameters assume a load capacitance of 100 pf. Add 10 ns delay for each 50 pf increase in load up to a maximum of 200 pf for the data bus and 100 pf for address and control lines. AC timing measurements are referenced to 1.5 volts (except for clock, which is referenced to the 10% and 90%



DC CHARACTERISTICS (Z84C00/CMOS Z80 CPU)

Symbol	Parameter	Min	Max	Unit	Condition
V_{ILC}	Clock Input Low Voltage	−0.3	0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC} −.6	V_{CC} +.3	V	
V_{IL}	Input Low Voltage	−0.3	0.8	V	
V_{IH}	Input High Voltage	2.2	V_{CC}	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.0 mA
V_{OH_1}	Output High Voltage	2.4		V	I_{OH} = −1.6 mA
V_{OH_2}	Output High Voltage	V_{CC} −0.8		V	I_{OH} = −250 μA
I_{CC_1}	Power Supply Current **4 MHz**		**20**	**mA**	V_{CC} = 5V
	6 MHz		**30**	**mA**	V_{IH} = V_{CC} − 0.2V
	8 MHz		**40**	**mA**	V_{IL} = 0.2V
	10 MHz		**50**	**mA**	
	20 MHz		100	mA	V_{cc} = 5V
I_{CC_2}	Standby Supply Current		10	μA	V_{CC} = 5V
					$\overline{CLK}$ = (0)
					V_{IH} = V_{CC} − 0.2V
					V_{IL} = 0.2V
I_{LI}	Input Leakage Current	−10	10	μA	V_{IN} = 0.4 to V_{CC}
I_{LO}	3-State Output Leakage Current in Float	−10	10[2]	μA	V_{OUT} = 0.4 to V_{CC}

1. Measurements made with outputs floating.
2. A_{15}-A_0, D_7-D_0, $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$.
3. I_{CC_2} standby supply current is guaranteed only when the supplied clock is stopped at a low level during T_4 of the machine cycle immediately following the execution of a HALT instruction.

CAPACITANCE

Symbol	Parameter	Min	Max	Unit
C_{OUT}	Output Capacitance		15	pf

T_A = 25°C, f = 1 MHz.
Unmeasured pins returned to ground.

AC CHARACTERISTICS† (Z84C00/CMOS Z80 CPU)

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

No	Symbol	Parameter	Z84C0004** Min	Z84C0004** Max	Z84C0006 Min	Z84C0006 Max	Z84C0008 Min	Z84C0008 Max	Z84C0010 Min	Z84C0010 Max	Z84C0020[1] Min	Z84C0020[1] Max	Unit	Note
1	TcC	Clock Cycle time	250*	DC	162*	DC	125*	DC	100*	DC	50*	DC	nS	
2	TwCh	Clock Pulse width (high)	110	DC	65	DC	55	DC	40	DC	20	DC	nS	
3	TwCl	Clock Pulse width (low)	110	DC	65	DC	55	DC	40	DC	20	DC	nS	
4	TfC	Clock Fall time		30		20		10		10		10	nS	
5	TrC	Clock Rise time		30		20		10		10		10	nS	
6	TdCr(A)	Address valid from Clock Rise		110		90		80		65		57	nS	[2]
7	TdA(MREQf)	Address valid to /MREQ Fall	65*		35*		20*		5*		-15*		nS	
8	TdCf(MREQf)	Clock Fall to /MREQ Fall delay		85		70		60		55		40	nS	
9	TdCr(MREQr)	Clock Rise to /MREQ Rise delay		85		70		60		55		40	nS	
10	TwMREQh	/MREQ pulse width (High)	110*		65*		45**		30*		10*		nS	[3]
11	TwMREQl	/MREQ pulse width (low)	220*		132*		100*		75*		25*		nS	[3]
12	TdCf(MERQr)	Clock Fall to /MREQ Rise delay		85		70		60		55		40	nS	
13	TdCf(RDf)	Clock Fall to /RD Fall delay		95		80		70		65		40	nS	
14	TdCr(RDr)	Clock Rise to /RD Rise delay		85		70		60		55		40	nS	
15	TsD(Cr)	Data setup time to Clock Rise	35		30		30		25		12		nS	
16	ThD(RDr)	Data hold time after /RD Rise	0		0		0		0		0		nS	
17	TsWAIT(Cf)	/WAIT setup time to Clock Fall	70		60		50		20		7.5		nS	
18	ThWAIT(Cf)	/WAIT hold time after Clock Fall	10		10		10		10		10		nS	
19	TdCr(M1f)	Clock Rise to /M1 Fall delay		100		80		70		65		45	nS	
20	TdCr(M1r)	Clock Rise to /M1 Rise delay		100		80		70		65		45	nS	
21	TdCr(RFSHf)	Clock Rise to /RFSH Fall delay		130		110		95		80		60	nS	
22	TdCr(RFSHr)	Clock Rise to /RFSH Rise delay		120		100		85		80		60	nS	
23	TdCf(RDr)	Clock Fall to /RD Rise delay		85		70		60		55		40	nS	
24	TdCr(RDf)	Clock Rise to /RD Fall delay		85		70		60		55		40	nS	
25	TsD(Cf)	Data setup to Clock Fall during M2, M3, M4 or M5 cycles	50		40		30		25		12		nS	
26	TdA(IORQf)	Address stable prior to /IORQ Fall	180*		107*		75*		50*		0*		nS	
27	TdCr(IORQf)	Clock Rise to /IORQ Fall delay		75		65		55		50		40	nS	
28	TdCf(IORQr)	Clock Fall to /IORQ Rise delay		85		70		60		55		40	nS	
29	TdD(WRf)Mw	Data stable prior to /WR Fall	80*		22*		5*		40*		-10*		nS	
30	TdCf(WRf)	Clock Fall to /WR Fall delay		80		70		60		55		40	nS	
31	TwWR	/WR pulse width	220*		132*		100*		75*		25*		nS	
33	TdD(WRf)IO	Data stable prior to /WR Fall	-10*		-55*		-55*		-10*		-30*		nS	
34	TdCr(WRf)	Clock Rise to /WR Fall delay		65		60		60		50		40	nS	
35	TdWRr(D)	Data stable from /WR Rise	60*		30*		15*		10*		0*		nS	
36	TdCf(HALT)	Clock Fall to /HALT 'L' or 'H'		300		260		225		90		70	nS	
37	TwNMI	/NMI pulse width	80		60		60		60		60		nS	
38	TsBUSREQ (Cr)	/BUSREQ setup time to Clock Rise	50		50		40		30		15		nS	

*For clock periods other than the minimums shown, calculate parameters using the table on the following page. Calculated values above assumed TrC = TfC = 20 ns.

†Units in nanoseconds (ns).

†† For loading ≥ 50 pf. Decrease width by 10 ns for each additional 50 pf.

**4 MHz CMOS Z80 is obsoleted and replaced by 6 MHz

AC CHARACTERISTICS† (Z84C00/CMOS Z80 CPU; Continued)

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

No	Symbol	Parameter	Z84C0004** Min	Z84C0004** Max	Z84C0006 Min	Z84C0006 Max	Z84C0008 Min	Z84C0008 Max	Z84C0010 Min	Z84C0010 Max	Z84C0020[1] Min	Z84C0020[1] Max	Unit	Note
39	ThBUSREQ (Cr)	/BUSREQ hold time after Clock Rise	10		10		10		10		10		nS	
40	TdCr (BUSACKf)	Clock Rise to /BASACK Fall delay		100		90		80		75		40	nS	
41	TdCf (BUSACKr)	Clock Fall to /BASACK Rise delay		100		90		80		75		40	nS	
42	TdCr(Dz)	Clock Rise to Data float delay		90		80		70		65		40	nS	
43	TdCr(CTz)	Clock Rise to Control Outputs Float Delay (/MREQ, /IORQ, /RD and /WR)		80		70		60		65		40	nS	
44	TdCr(Az)	Clock Rise to Address float delay		90		80		70		75		40	nS	
45	TdCTr(A)	Address Hold time from /MREQ, /IORQ, /RD or /WR	80*		35*		20*		20*		0*		nS	
46	TsRESET(Cr)	/RESET to Clock Rise setup time	60		60		45		40		15		nS	
47	ThRESET(Cr)	/RESET to Clock Rise Hold time	10		10		10		10		10		nS	
48	TsINTf(Cr)	/INT Fall to Clock Rise Setup Time	80		70		55		50		15		nS	
49	ThINTr(Cr)	/INT Rise to Clock Rise Hold Time	10		10		10		10		10		nS	
50	TdM1f (IORQf)	/M1 Fall to /IORQ Fall delay	565*		359*		270*		220*		100*		nS	
51	TdCf(IORQf)	/Clock Fall to /IORQ Fall delay		85		70		60		55		45	nS	
52	TdCf(IORQr)	Clock Rise to /IORQ Rise delay		85		70		60		55		45	nS	
53	TdCf(D)	Clock Fall to Data Valid delay		150		130		115		110		75	nS	

Notes:

* For Clock periods other than the minimum shown, calculate parameters using the following table.
Calculated values above assumed TrC = TfC = maximum.

** 4 MHz CMOS Z80 is obsoleted and replaced by 6 MHz

[1] Z84C0020 parameters are guuaranteed with 50pF load Capacitance.

[2] If Capacitive Load is other than 50pF, please use Figure 1. to calculate the value.

[3] Increasing delay by 10nS for each 50pF increase in loading, 200pF max for data lines, and 100pF for control lines.

FOOTNOTES TO AC CHARACTERISTICS

No	Symbol	Parameter	Z84C0004**	Z84C0006	Z84C0008	Z84C0010	Z84C0020
1	TcC	TwCh + TwCl + TrC + TfC					
7	TdA(MREQf)	TwCh + TfC	-65	-50	-45	-45	-45
10	TwMREQh	TwCh + TfC	-20	-20	-20	-20	-20
11	TwMREQl	TcC	-30	-30	-25	-25	-25
26	TdA(IORQf)	TcC	-70	-55	-50	-50	-50
29	TdD(WRf)	TcC	-170	-140	-120	-60	-60
31	TwWR	TcC	-30	-30	-25	-25	-25
33	TdD(WRf)	TwCl + TrC	-140	-140	-120	-60	-60
35	TdWRr(D)	TwCl + TrC	-70	-55	-50	-40	-25
45	TdCTr(A)	TwCl + TrC	-50	-50	-45	-30	-30
50	TdM1f(IORQf)	2TcC + TwCh + TfC	-65	-50	-45	-30	-30

AC Test Conditions: V_{IH} = 2.0 V, V_{IL} = 0.8 V; V_{OH} = 1.5 V, V_{OL} = 1.5 V; V_{IHC} = V_{CC} −0.6 V, V_{ILC} = 0.45 V; FLOAT = ±0.5 V

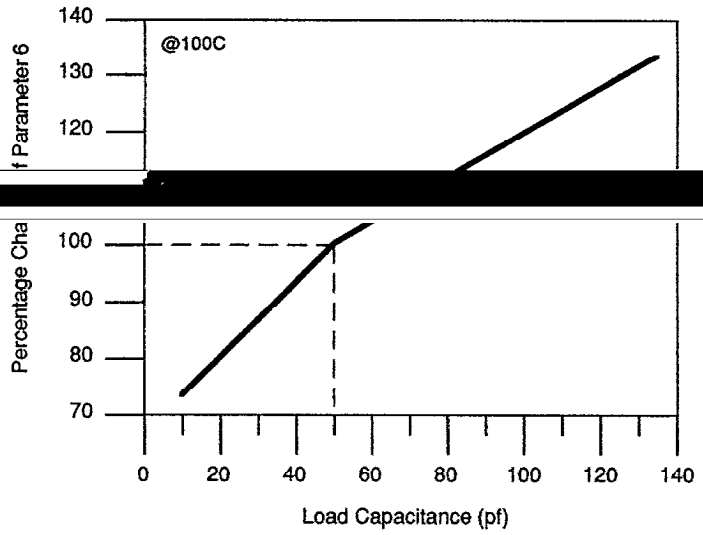


Figure 1. Address Delay Characteristics
(Parameter 6)

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} − .6	V_{CC} + .3	V	
V_{IL}	Input Low Voltage	−0.3	0.8	V	
V_{IH}	Input High Voltage	2.0[1]	V_{CC}	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	2.4[1]		V	I_{OH} = −250 μA
I_{CC}	Power Supply Current		200	mA	Note 3
I_{LI}	Input Leakage Current		10	μA	V_{IN} = 0 to V_{CC}
I_{LO}	3-State Output Leakage Current in Float	−10	10[2]	μA	V_{OUT} = 0.4 to V_{CC}

1. For military grade parts, refer to the Z80 Military Electrical Specification.
2. A_{15}-A_0, D_7-D_0, $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$.
3. Measurements made with outputs floating.

CAPACITANCE

Guaranteed by design and characterization.

Symbol	Parameter	Min	Max	Unit
C_{CLOCK}	Clock Capacitance		35	pf
C_{IN}	Input Capacitance		5	pf
C_{OUT}	Output Capacitance		15	pf

NOTES:
T_A = 25°C, f = 1 MHz.
Unmeasured pins returned to ground.

AC CHARACTERISTICS† (Z8400/NMOS Z80 CPU)

Number	Symbol	Parameter	Z0840004		Z0840006		Z0840008	
2	TwCh	Clock Pulse Width (High)	110	2000	65	2000	55	2000
3	TwCl	Clock Pulse Width (Low)	110	2000	65	2000	55	2000
4	TfC	Clock Fall Time		30		20		10
5	TrC	Clock Rise Time		30		20		10
6	TdCr(A)	Clock ↑ to Address Valid Delay		110		90		80
7	TdA(MREQf)	Address Valid to $\overline{MREQ}$ ↓ Delay	65*		35*		20*	
8	TdCf(MREQf)	Clock ↓ to $\overline{MREQ}$ ↓ Delay		85		70		60
9	TdCr(MREQr)	Clock ↑ to $\overline{MREQ}$ ↑ Delay		85		70		60
10	TwMREQh	$\overline{MREQ}$ Pulse Width (High)	110*‡		65*‡		45*‡	
11	TwMREQl	$\overline{MREQ}$ Pulse Width (Low)	220*‡		135*‡		100*‡	
12	TdCf(MREQr)	Clock ↓ to $\overline{MREQ}$ ↑ Delay		85		70		60
13	TdCf(RDf)	Clock ↓ to $\overline{RD}$ ↓ Delay		95		80		70
14	TdCr(RDr)	Clock ↑ to $\overline{RD}$ ↑ Delay		85		70		60
15	TsD(Cr)	Data Setup Time to Clock ↑	35		30		30	
16	ThD(RDr)	Data Hold Time to $\overline{RD}$ ↑		0		0		0
17	TsWAIT(Cf)	$\overline{WAIT}$ Setup Time to Clock ↓	70		60		50	
18	ThWAIT(Cf)	$\overline{WAIT}$ Hold Time after Clock ↓		0		0		0
19	TdCr(M1f)	Clock ↑ to $\overline{M1}$ ↓ Delay		100		80		70
20	TdCr(M1r)	Clock ↑ to $\overline{M1}$ ↑ Delay		100		80		70
21	TdCr(RFSHf)	Clock ↑ to $\overline{RFSH}$ ↓ Delay		130		110		95
22	TdCr(RFSHr)	Clock ↑ to $\overline{RFSH}$ ↑ Delay		120		100		85
23	TdCf(RDr)	Clock ↓ to $\overline{RD}$ ↑ Delay		85		70		60
24	TdCr(RDf)	Clock ↑ to $\overline{RD}$ ↓ Delay		85		70		60
25	TsD(Cf)	Data Setup to Clock ↓ during M_2, M_3, M_4, or M_5 Cycles	50		40		30	
26	TdA(IORQf)	Address Stable prior to $\overline{IORQ}$ ↓	180*		110*		75*	
27	TdCr(IORQf)	Clock ↑ to $\overline{IORQ}$ ↓ Delay		75		65		55
28	TdCf(IORQr)	Clock ↓ to $\overline{IORQ}$ ↑ Delay		85		70		60
29	TdD(WRf)	Data Stable prior to $\overline{WR}$ ↓	80*		25*		5*	
30	TdCf(WRf)	Clock ↓ to $\overline{WR}$ ↓ Delay		80		70		60
31	TwWR	$\overline{WR}$ Pulse Width	220*		135*		100*	
32	TdCf(WRr)	Clock ↓ to $\overline{WR}$ ↑ Delay		80		70		60
33	TdD(WRf)	Data Stable prior to $\overline{WR}$ ↓	−10*		−55*		55*	
34	TdCr(WRf)	Clock ↑ to $\overline{WR}$ ↓ Delay		65		60		55
35	TdWRr(D)	Data Stable from $\overline{WR}$ ↑	60*		30*		15*	
36	TdCf(HALT)	Clock ↓ to $\overline{HALT}$ ↑ or ↓		300		260		225
37	TwNMI	$\overline{NMI}$ Pulse Width	80		70		60*	
38	TsBUSREQ(Cr)	$\overline{BUSREQ}$ Setup Time to Clock ↑	50		50		40	

*For clock periods other than the minimums shown, calculate parameters using the table on the following page. Calculated values above assumed TrC = TfC = 20 ns.

†Units in nanoseconds (ns).

‡ For loading ≥ 50 pf., Decrease width by 10 ns for each additional 50 pf.

AC CHARACTERISTICS† (Z8400/NMOS Z80 CPU; Continued)

Number	Symbol	Parameter	Z0840004 Min	Z0840004 Max	Z0840006 Min	Z0840006 Max	Z0840008 Min	Z0840008 Max
39	ThBUSREQ(Cr)	$\overline{BUSREQ}$ Hold Time after Clock ↑	0		0		0	
40	TdCr(BUSACKf)	Clock ↑ to $\overline{BUSACK}$ ↓ Delay		100		90		80
41	TdCf(BUSACKr)	Clock ↓ to $\overline{BUSACK}$ ↑ Delay		100		90		80
42	TdCr(Dz)	Clock ↑ to Data Float Delay		90		80		70
43	TdCr(CTz)	Clock ↑ to Control Outputs Float Delay ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$)		80		70		60
44	TdCr(Az)	Clock ↑ to Address Float Delay		90		80		70
45	TdCTr(A)	$\overline{MREQ}$ ↑, $\overline{IORQ}$ ↑, $\overline{RD}$ ↑, and $\overline{WR}$ ↑ to Address Hold Time	80*		35*		20*	
46	TsRESET(Cr)	$\overline{RESET}$ to Clock ↑ Setup Time	60		60		45	
47	ThRESET(Cr)	$\overline{RESET}$ to Clock ↑ Hold Time		0		0		0
48	TsINTf(Cr)	$\overline{INT}$ to Clock ↑ Setup Time	80		70		55	
49	ThINTr(Cr)	$\overline{INT}$ to Clock ↑ Hold Time		0		0		0
50	TdM1f(IORQf)	$\overline{M1}$ ↓ to $\overline{IORQ}$ ↓ Delay	565*		365*		270*	
51	TdCf(IORQf)	Clock ↓ to $\overline{IORQ}$ ↓ Delay		85		70		60
52	TdCf(IORQr)	Clock ↑ $\overline{IORQ}$ ↑ Delay		85		70		60
53	TdCf(D)	Clock ↓ to Data Valid Delay		150		130		115

*For clock periods other than the minimums shown, calculate parameters using the following table. Calculated values above assumed TrC = TfC = 20 ns.
†Units in nanoseconds (ns).

FOOTNOTES TO AC CHARACTERISTICS

Number	Symbol	General Parameter	Z0840004	Z0840006	Z0840008
1	TcC	TwCh + TwCl + TrC + TfC			
7	TdA(MREQf)	TwCh + TfC	− 65	−50	−45
10	TwMREQh	TwCh + TfC	− 20	−20	−20
11	TwMREQl	TcC	− 30	−30	−25
26	TdA(IORQf)	TcC	− 70	−55	−50
29	TdD(WRf)	TcC	− 170	−140	−120
31	TwWR	TcC	− 30	−30	−25
33	TdD(WRf)	TwCl + TrC	− 140	−140	−120
35	TdWRr(D)	TwCl + TrC	− 70	−55	−50
45	TdCTr(A)	TwCl + TrC	− 50	−50	−45
50	TdM1f(IORQf)	2TcC + TwCh + TfC	− 65	−50	−45

AC Test Conditions:
V_{IH} = 2.0 V
V_{IL} = 0.8 V
V_{IHC} = V_{CC} − 0.6 V
V_{ILC} = 0.45 V
V_{OH} = 1.5 V
V_{OL} = 1.5 V
FLOAT = ±0.5 V

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

1

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,
- 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
- NMOS version for cost sensitive performance solutions.
- CMOS version for the designs requiring high speed and low power consumption
- 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 that are compatible with the Z80 PIO include most keyboards, paper tape readers and punches, printers, and PROM programmers.

configures the Z80 PIO to interface with a wide range of separates them from other interface controllers is that all

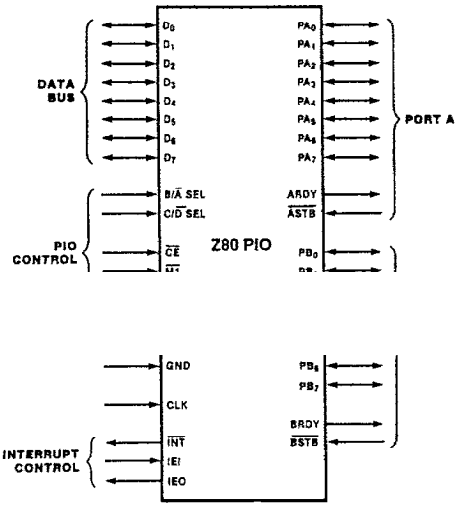
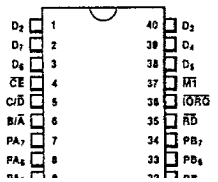


Figure 1. Pin Functions



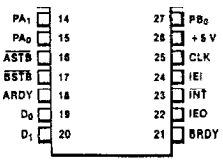


Figure 2a. 40-pin Dual-In-Line Package (DIP), Pin Assignments

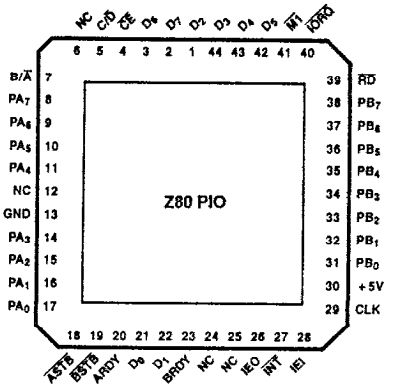
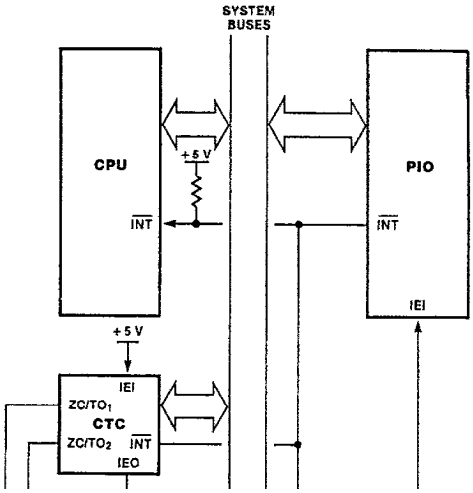
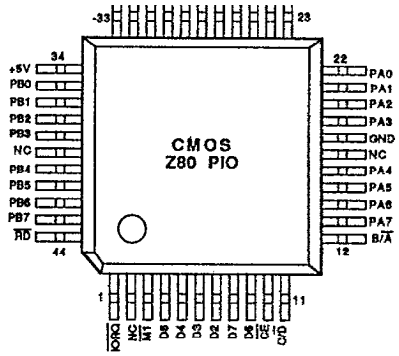


Figure 2b. 44-pin Chip Carrier, Pin Assignments





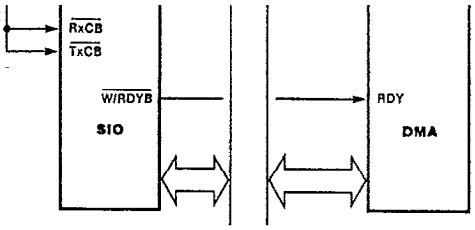


Figure 3. PIO in a Typical Z80 Family Environment

Figure 2c. 44-pin Quad Flat Pack Pin Assignments.

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.

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

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;

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

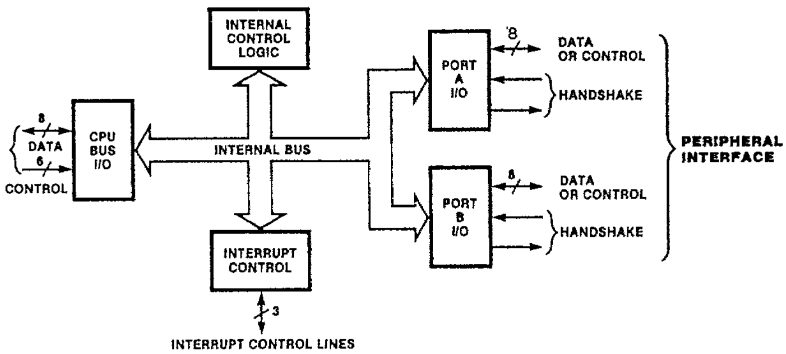


Figure 4. Block Diagram

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.

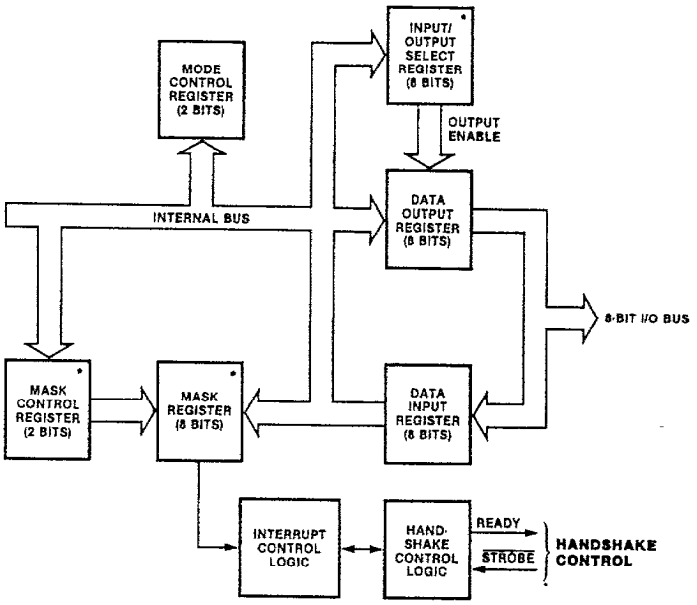
If the CPU (in interrupt Mode 2) accepts an interrupt, the interrupting device must provide an 8-bit interrupt vector for the CPU. This vector forms a pointer to a location in memory where the address of the interrupt service routine is located. The 8-bit vector from the interrupting device forms the 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

[redacted]

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 anytime

[redacted]

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.

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

[redacted]

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).

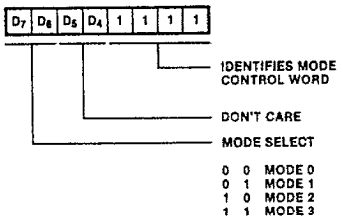


Figure 6. Mode Control Word

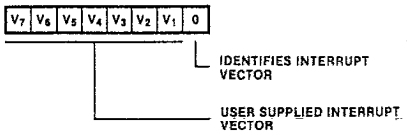


Figure 7. Interrupt Vector Word

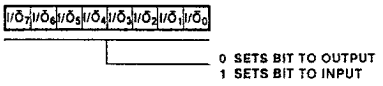


Figure 8. I/O Register Control Word

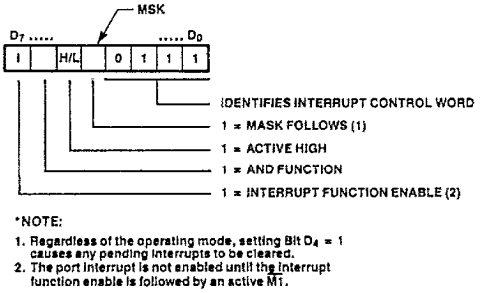


Figure 9. Interrupt Control Word

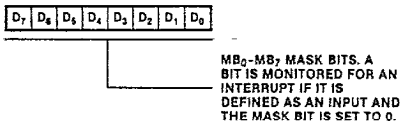


Figure 10. Mask Control Word

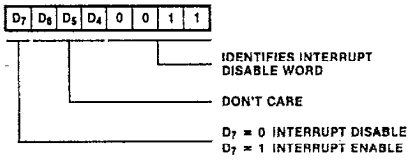


Figure 11. Interrupt Disable Word

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

$\overline{BSTB}$. *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.

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

$\overline{RD}$ 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

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

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.

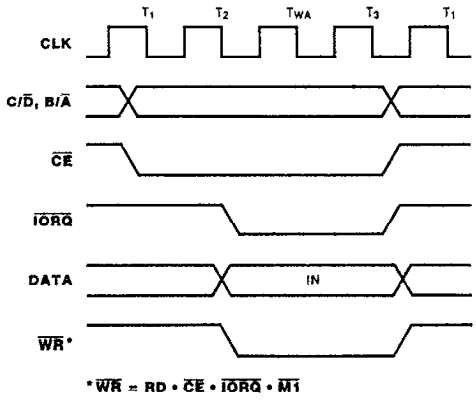


Figure 12. Write Cycle Timing

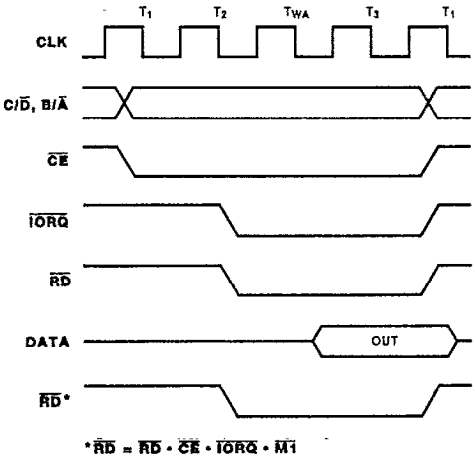


Figure 13. Read Cycle Timing

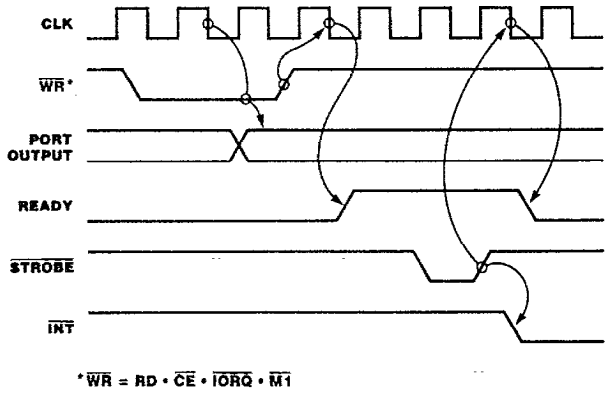


Figure 14. Mode 0 Output Timing

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 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.

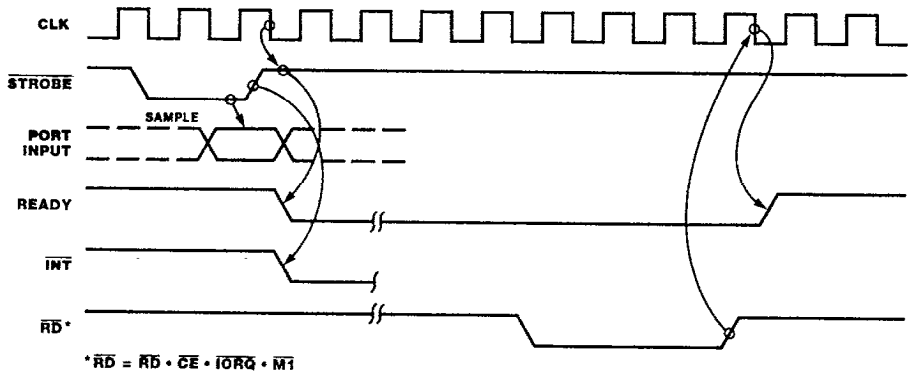


Figure 15. Mode 1 Input Timing

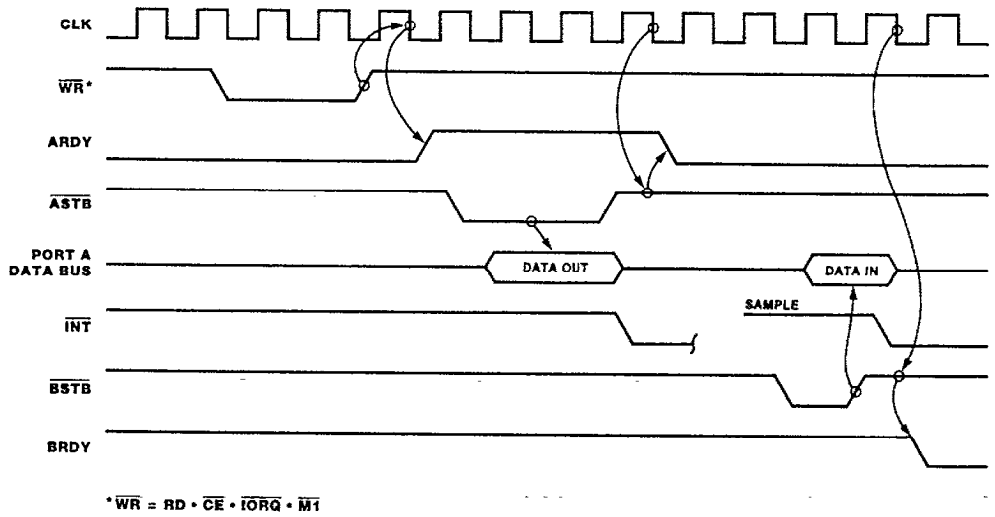


Figure 16. Mode 2 Bidirectional Timing

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.

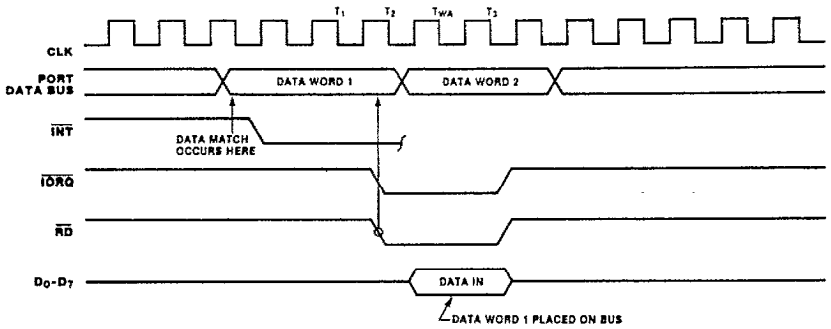


Figure 17. Mode 3 Bit Control Mode Timing, Bit Mode Read

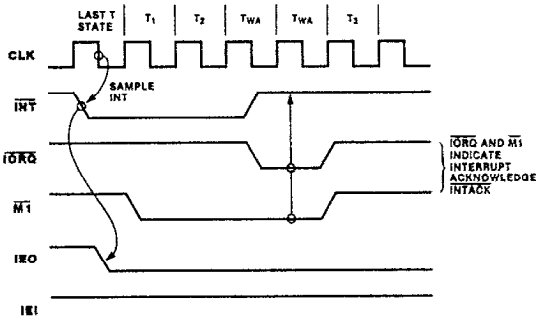


Figure 18. Interrupt Acknowledge Timing

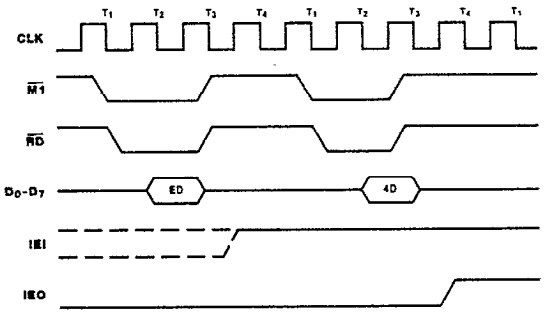


Figure 19. Return From Interrupt

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

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

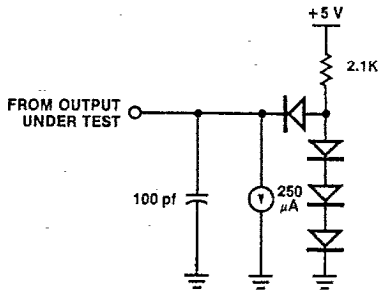
device reliability.

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 ≤ V_{cc} ≤ +5.25V
  CMOS: +4.50V ≤ V_{cc} ≤ +5.50V
- E= -40°C to 100°C, +4.50V ≤ V_{cc} ≤ +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	[redacted]								
13	TsM1(Cr)	/M1 Falling to Clock Rising	[redacted]								
		(/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]
[redacted]											
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	

1

* 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_{IL}	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).

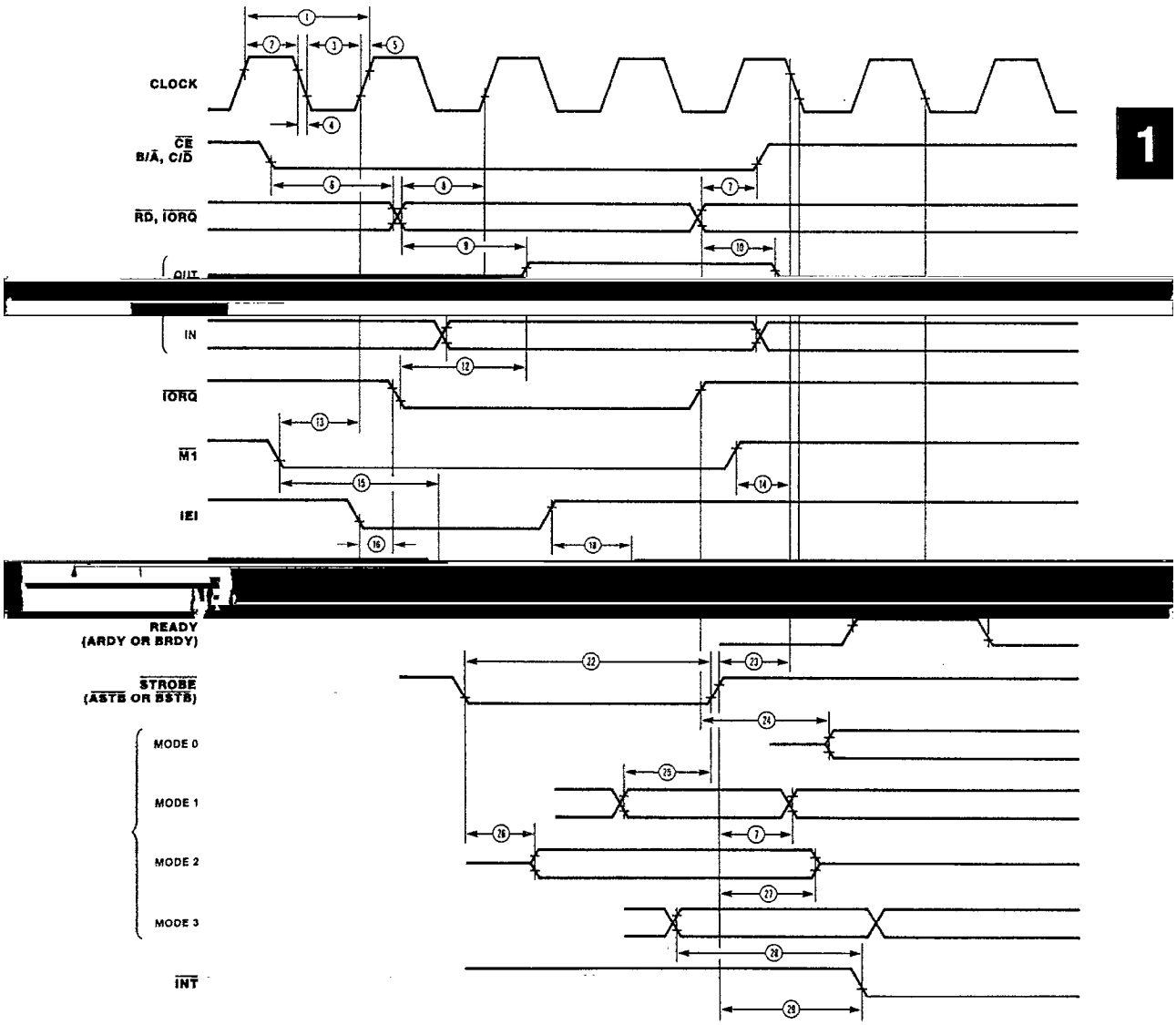
[5] Increase these values by 2 ns for each 10 pf increase in loading up to 100 pf max.

[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



Z8430/Z84C30 NMOS/CMOS Z80 ®CTC Counter/Timer Circuit

1

FEATURES

- Four independently programmable counter/timer channels, each with a readable downcounter and a selectable 16 or 256 prescaler. Downcounters are reloaded automatically at zero count.
- Three channels have Zero Count/Timeout outputs capable of driving Darlington transistors. (1.5 mV @ 1.5V)
- NMOS version for cost sensitive performance solutions.
- CMOS version for the designs requiring low power consumption
- NMOS Z0843004 - 4 MHz, Z0843006 - 6.17 MHz.
- CMOS Z84C3006 - DC to 6.17 MHz, Z84C3008 - DC to 8 MHz, Z84C3010 - DC to 10 MHz

provides fully vectored, prioritized interrupts without external logic. The CTC may also be used as an interrupt controller.

- 6 MHz version supports 6.144 MHz CPU clock operation.

GENERAL DESCRIPTION

The Z80 CTC, hereinafter referred to as Z80 CTC or CTC, software for a broad range of counting and timing applications. The four independently programmable channels of the Z80 CTC satisfy common microcomputer system requirements for event counting, interrupt and interval timing, and general clock rate generation.

System design is simplified because the CTC connects directly to both the Z80 CPU and the Z80 SIO with no additional logic. In larger systems, address decoders and buffers may be required.

Programming the CTC is straightforward: each channel is interrupts are enabled. Once started, the CTC counts down, automatically reloads its time constant, and resumes counting. Software timing loops are completely eliminated. Interrupt processing is simplified because only one vector need be specified; the CTC internally generates a unique vector for each channel.

The Z80 CTC requires a single +5%V power supply and the standard Z80 single-phase system clock. It is packaged in 28-pin DIPs, a 44-pin plastic chip carrier, and a 44-pin Quad Flat Pack. (Figures 2a, 2b, and 2c). Note that the QFP package is only available for CMOS versions.

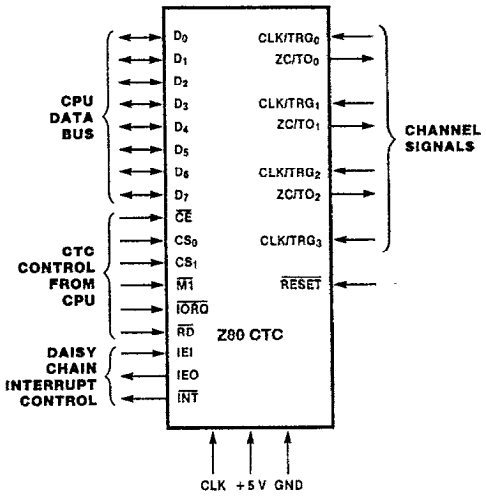


Figure 1. Pin Functions

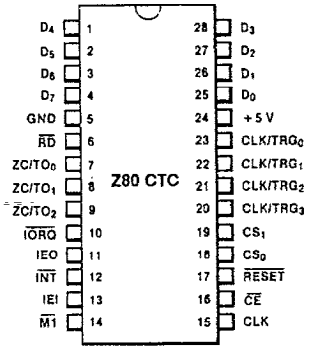


Figure 22a. Pin Assignments

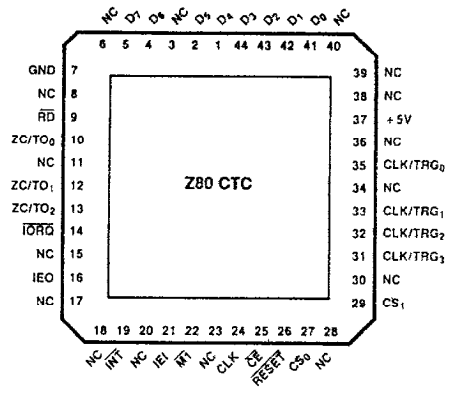


Figure 2b. 44-pin Chip Carrier, Pin Assignments

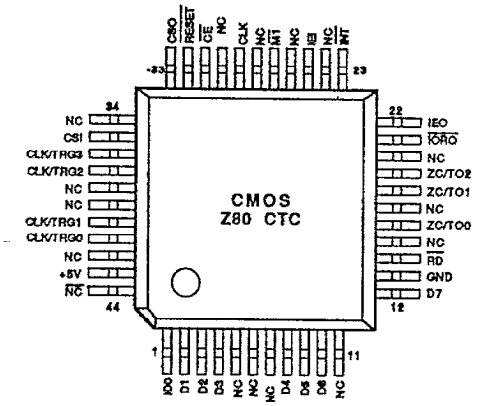


Figure 2C. 44-Pin Quad Flat Pack Pin Assignments

FUNCTIONAL DESCRIPTION

The Z80 CTC has four independent counter/timer channels. Each channel is individually programmed with two words: a control word and a time-constant word. The control word selects the operating mode (counter or timer), enables or disables the channel interrupt, and selects certain other operating parameters. If the timing mode is selected, the control word also sets a prescaler, which divides the system clock by either 16 or 256. The time-constant word is a value from 1 to 256.

During operation, the individual counter channel counts down from the preset time constant value. In counter mode operation the counter decrements on each of the CLK/TRG input pulses until zero count is reached. Each decrement is synchronized by the system clock. For counts greater than 256, more than one counter can be cascaded. At zero count, the down-counter is automatically reset with the time constant value.

The timer mode determines time intervals as small as 2 µs(8 MHz), 3 µs (6 MHz), or 4 µs (4MHz) without additional logic or software timing loops. Time intervals are generated by dividing the system clock with a prescaler that decrements a preset down-counter.

Thus, the time interval is an integral multiple of the clock period, the prescaler value (16 or 256), and the time constant that is preset in the down-counter. A timer is triggered automatically when its time constant value is programmed, or by an external CLK/TRG input.

Three channels have two outputs that occur at zero count. The first output is a zero-count/timeout pulse at the ZC/TO output. The fourth channel (Channel 3) does not have a ZC/TO output; interrupt request is the only output available from Channel 3.

The second output is Interrupt Request ($\overline{INT}$), which occurs if the channel has its interrupt enabled during programming. When the Z80 CPU acknowledges Interrupt Request, the Z80 CTC places an interrupt vector on the data bus.

The four channels of the Z80 CTC are fully prioritized and fit into four contiguous slots in a standard Z80 daisy-chain interrupt structure. Channel 0 is the highest priority and Channel 3 the lowest. Interrupts can be individually enabled (or disabled) for each of the four channels.

INTERNAL STRUCTURE

The CTC has four major elements, as shown in Figure 3.

- CPU bus I/O
- Channel control logic
- Interrupt logic
- Counter/timer circuits

CPU Bus I/O. The CPU bus I/O circuit decodes the address inputs, and interfaces the CPU data and control signals to the CTC for distribution on the internal bus.

Internal Control Logic. The CTC internal control logic controls overall chip operating functions such as the chip enable, reset, and read/write logic.

Interrupt Logic. The interrupt control logic ensures that the CTC interrupts interface properly with the Z80 CPU interrupt system. The logic controls the interrupt priority of the CTC as a function of the IEI signal. If IEI is High, the CTC has priority. During interrupt processing, the interrupt logic holds IEO Low, which inhibits the interrupt operation on lower priority devices. If the IEI input goes Low, priority is relinquished and the interrupt logic drives IEO Low.

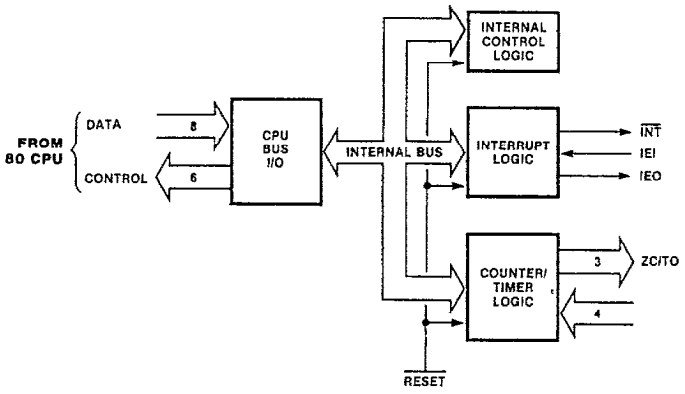


Figure 3. Functional Block Diagram

If a channel is programmed to request an interrupt, the [illegible] the interrupt logic arbitrates the CTC internal priorities, and the interrupt control logic places a unique interrupt vector on the data bus.

If an interrupt is pending, the interrupt logic holds IEO Low. [illegible] (ED_{16}). If the device has a pending interrupt, it raises IEO properly.

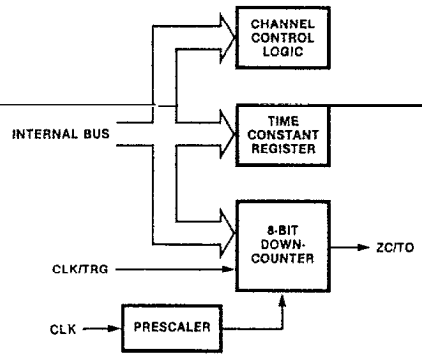


Figure 4. Counter/Timer Block Diagram

Counter/Timer Circuits. The CTC has four independent [illegible]

Channel Control Logic. The channel control logic receives the 8-bit channel control word when the counter/timer channel is programmed. The channel control logic decodes the control word and sets the following operating conditions:

- Operating mode (timer or counter)
- Active slope for CLK/TRG input
- Timer mode trigger (automatic or CLK/TRG input)
- Time constant data word to follow
- Software reset

Time Constant Register. When the counter/timer channel is programmed, the time constant register receives and stores an 8-bit time constant value, which can be anywhere from 1 to 256 (0 = 256). This constant is automatically loaded into the down-counter when the counter/timer channel is initialized, and subsequently after each zero count.

Prescaler. The prescaler, which is used only in timer mode, divides the system clock frequency by a factor of either 16 or 256. The prescaler output clocks the down-counter during timer operation. The effect of the prescaler on the down-counter is a multiplication of the system clock period by 16 or 256. The prescaler factor is programmed by bit 5 of the channel control word.

Down-Counter. Prior to each count cycle, the down-counter is loaded with the time constant register contents. The counter is then decremented one of two ways, depending on operating mode:

- By the prescaler output (timer mode)
- By the trigger pulses into the CLK/TRG input (counter mode)

Without disturbing the down-count, the Z80 CPU can read the count remaining at any time by performing an I/O read operation at the port address assigned to the CTC channel. When the down-counter reaches the zero count, the ZC/TO output generates a positive-going pulse. When the interrupt is enabled, zero count also triggers an interrupt request signal ($\overline{INT}$) from the interrupt logic.

PROGRAMMING

Each Z80 CTC channel must be programmed prior to operation. Programming consists of writing two words to the I/O port that corresponds to the desired channel. The first word is a control word that selects the operating mode and other parameters; the second word is a time constant, which is a binary data word with a value from 1 to 256. A time constant word must be preceded by a channel control word.

After initialization, channels may be reprogrammed at any time. If updated control and time constant words are written to a channel during the count operation, the count continues to zero before the new time constant is loaded into the counter.

If the interrupt on any Z80 CTC channel is enabled, the programming procedure should also include an interrupt [illegible]

A control word is identified by a 1 in bit 0. A 1 in bit 2 indicates a time constant word is to follow. Interrupt vectors are always addressed to Channel 0, and identified by a 0 in bit 0.

Addressing. During programming, channels are addressed with the channel select pins CS_1 and CS_2. A 2-bit binary code selects the appropriate channel as shown in the following table.

Channel	CS_1	CS_0
0	0	0
1	0	1
2	1	0
3	1	1

Reset. The CTC has both hardware and software resets. The hardware reset terminates all down-counts and disables all CTC interrupts by resetting the interrupt bits in the control registers. In addition, the ZC/TO and Interrupt outputs go inactive, IEO reflects IEI, and D_0-D_7 go to the high-impedance state. All channels must be completely reprogrammed after a hardware reset.

The software reset is controlled by bit 1 in the channel control word. When a channel receives a software reset, it stops counting. When a software reset is used, the other bits in the control word also change the contents of the channel control register. After a software reset a new time constant word must be written to the same channel.

If the channel control word has both bits D_1 and D_2 set to 1, [illegible]

operation is triggered automatically when the time constant word is loaded.

Channel Control Word Programming. The channel control word is shown in Figure 5. It sets the modes and parameters described below.

Interrupt Enable. D_7 enables the interrupt, so that an interrupt output ($\overline{INT}$) is generated at zero count. Interrupts may be programmed in either mode and may be enabled or disabled at any time.

Mode. D_6 selects either timer or counter operating mode.

Prescaler Factor. (Timer Mode Only). D_5 selects factor—either 16 or 256.

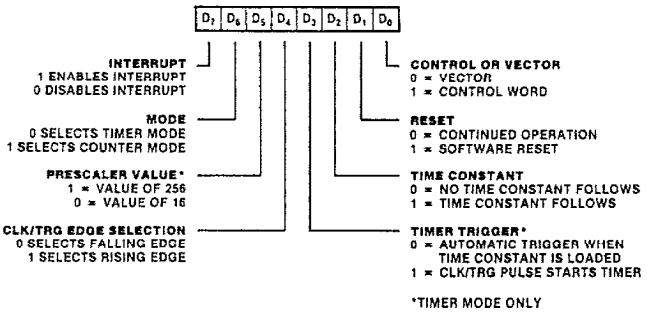


Figure 5. Channel Control Word

Clock/Trigger Edge Selector. D_4 selects the active edge or slope of the CLK/TRG input pulses. Note that reprogramming the CLK/TRG slope during operation is equivalent to issuing an active edge. If the trigger slope is changed by a control word update while a channel is pending operation in timer mode, the result is the same as a CLK/TRG pulse and the timer starts. Similarly, if the channel is in counter mode, the counter decrements.

Timer Trigger (Timer Mode Only). D_3 selects the trigger mode for timer operation. When D_3 is reset to 0, the timer is

machine cycle. At the end of the write operation there is a setup delay of one clock period. The timer starts automatically (decrements) on the rising edge of the second clock pulse (T_2) of the machine cycle following the write operation. Once started, the timer runs continuously. At zero count the timer reloads automatically and continues counting without interruption or delay, until stopped by a reset.

When D_3 is set to 1, the timer is triggered externally through the CLK/TRG input. The time constant word is programmed during an I/O write operation, which takes one machine cycle. The timer is ready for operation on the rising edge of the second clock pulse (T_2) of the following machine cycle. Note that the first timer decrement follows the active edge of the CLK/TRG pulse by a delay time of one clock cycle if a minimum setup time to the rising edge of clock is met. If this minimum is not met, the delay is extended by another clock period. Consequently, for immediate triggering, the CLK/TRG input must precede T_2 by one clock cycle plus its

timer will start on the third clock cycle (T_3).

Once started the timer operates continuously, without interruption or delay, until stopped by a reset.

Time Constant. A 1 in D_2 indicates that the next word addressed to the selected channel is a time constant data word for the time constant register. The time constant word may be written at any time.

A 0 in D_2 indicates no time constant word is to follow. This is ordinarily used when the channel is already in operation and the new channel control word is an update. A channel will not operate without a time constant value. The only way to write a time constant value is to write a control word with D_2 set.

Software Reset. Setting D_1 to 1 causes a software reset, which is described in the Reset section.

Control Word. Setting D_0 to 0 identifies the word as a control word.

Time Constant Programming. Before a channel can start counting it must receive a time constant word from the CPU.

word to indicate that the next word is a time constant. The time constant word can be any value from 1 to 256 (Figure 6). Note that 00_{16} is interpreted as 256.

In timer mode, the time interval is controlled by three factors:

- The system clock period (CLK)
- The prescaler factor (P), which multiplies the interval by either 16 or 256
- The time constant (T), which is programmed into the time constant register

Consequently, the time interval is the product of CLK x P x T. The minimum timer resolution is 16 x CLK (4μs with a 4MHz clock). The maximum timer interval is 256 x CLK x 256 (16.4 ms with a 4MHz clock). For longer intervals timers may be cascaded.

Interrupt Vector Programming. If the Z80 CTC has one or more interrupts enabled, it can supply interrupt vectors to

pre-programmed with the most-significant five bits of the interrupt vector. Programming consists of writing a vector word to the I/O port corresponding to the Z80 CTC Channel 0. Note that D_0 of the vector word is always zero, to distinguish the vector from a channel control word. D_1 and D_2 are not used in programming the vector word. These bits are supplied by the interrupt logic to identify the channel requesting interrupt service with a unique interrupt vector (Figure 7). Channel 0 has the highest priority.

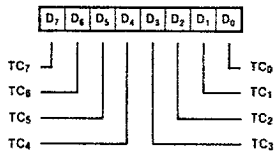


Figure 6. Time Constant Word

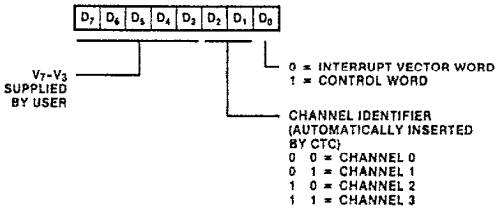


Figure 7. Interrupt Vector Word

PIN DESCRIPTION

$\overline{CE}$. *Chip Enable* (input, active Low). When enabled the CTC accepts control words, interrupt vectors, or time constant data words from the data bus during an I/O write cycle; or transmits the contents of the downcounter to the CPU during an I/O read cycle. In most applications this signal is decoded from the eight least significant bits of the address bus for any of the four I/O port addresses that are mapped to the four counter-timer channels.

CLK. *System Clock* (input). Standard single-phase Z80 system clock.

CLK/TRG$_0$-CLK/TRG$_3$. *External Clock/Timer Trigger* (input, user-selectable active High or Low). Four pins corresponding to the four Z80 CTC channels. In counter mode, every active edge on this pin decrements the downcounter. In timer mode, an active edge starts the timer.

CS$_0$-CS$_1$. *Channel Select* (inputs active High). Two-bit binary address code selects one of the four CTC channels for an I/O write or read (usually connected to A_0 and A_1).

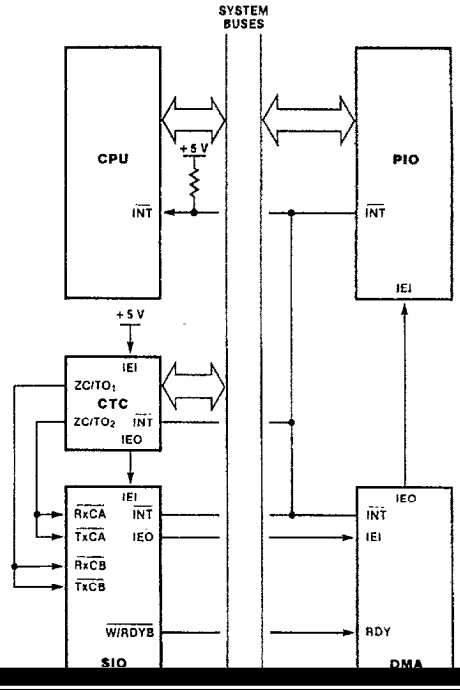


Figure 8. A Typical Z80 Environment

D$_0$-D$_7$. *System Data Bus* (bidirectional, 3-state). Transfers all data and commands between the Z80 CPU and the Z80 CTC.

IEI. *Interrupt Enable In* (input, active High). A High indicates that no other interrupting devices of higher priority in the daisy chain are being serviced by the Z80 CPU.

IEO. *Interrupt Enable Out* (output, active High). High only if IEI is High and the Z80 CPU is not servicing an interrupt from any Z80 CTC channel. IEO blocks lower priority devices from interrupting while a higher priority interrupting device is being serviced.

$\overline{INT}$. *Interrupt Request* (output, open drain, active Low). Low when any Z80 CTC channel that has been programmed to enable interrupts as a zero-count condition in its downcounter.

$\overline{IORQ}$. *Input/Output Request* (input from CPU, active Low). Used with $\overline{CE}$ and $\overline{RD}$ to transfer data and channel control words between the Z80 CPU and the Z80 CTC. During a write cycle, $\overline{IORQ}$ and $\overline{CE}$ are active and $\overline{RD}$ inactive. The Z80 CTC does not receive a specific write signal; rather, it internally generates is own from the inverse of an active $\overline{RD}$ signal. In a read cycle, $\overline{IORQ}$, $\overline{CE}$, and $\overline{RD}$ are active; the contents of the downcounter are read by the Z80 CPU. If $\overline{IORQ}$ and $\overline{M1}$ are both true, the CPU is acknowledging an interrupt request, and the highest priority interrupting channel places its interrupt vector on the Z80 data bus.

$\overline{M1}$. *Machine Cycle One* (input from CPU, active Low). When $\overline{M1}$ and $\overline{IORQ}$ are active, the Z80 CPU is acknowledging an interrupt. The Z80 CTC then places an interupt vector on the data bus if it has highest priority, and if a channel has requested an interrupt ($\overline{INT}$).

$\overline{RD}$. *Read Cycle Status* (input, active Low). Used in conjunction with $\overline{IORQ}$ and $\overline{CE}$ to transfer data and channel control words between the Z80 CPU and the Z80 CTC.

$\overline{RESET}$. *Reset* (input active Low). Terminates all down-counts and disables all interrupts by resetting the interrupt bits in all control registers; the ZC/TO and the interrupt outputs go inactive; IEO reflects IEI; D$_0$-D$_7$ go to the high-impedance state.

ZC/TO$_0$-ZC/TO$_2$. *Zero Count/Timeout* (output, active High). Three ZC/TO pins corresponding to Z80 CTC channels 2 through 0 (Channel 3 has no ZC/TO pin). In both counter and timer modes the output is an active High pulse

Read Cycle Timing. Figure 9 shows read cycle timing. This cycle reads the contents of a down-counter without disturbing the count. During clock cycle T_2, the Z80 CPU initiates a read cycle by driving the following inputs Low: $\overline{RD}$, $\overline{IORQ}$, and $\overline{CE}$. A 2-bit binary code at inputs CS_1 and CS_0 selects the channel to be read. $\overline{M1}$ must be High to distinguish this cycle from an interrupt acknowledge.

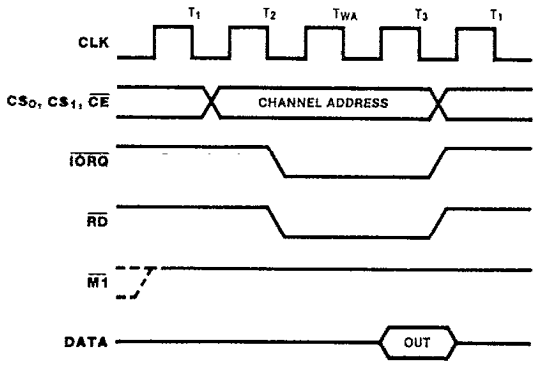


Figure 9. Read Cycle Timing

Write Cycle Timing. Figure 10 shows write cycle timing for loading control, time constant, or vector words.

The CTC does not have a write signal input, so it generates one internally when the read ($\overline{RD}$) input is High during T_1. During T_2 $\overline{IORQ}$ and $\overline{CE}$ inputs are Low. $\overline{M1}$ must be High to distinguish a write cycle from an interrupt acknowledge. A 2-bit binary code at inputs CS_1 and CS_0 selects the channel to be addressed, and the word being written is placed on the Z80 data bus. The data word is latched into the appropriate register with the rising edge of clock cycle T_3.

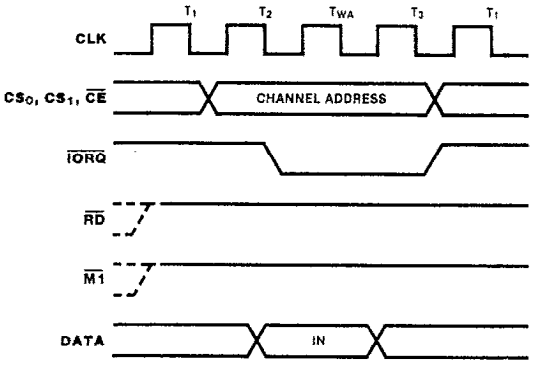


Figure 10. Write Cycle Timing

Timer Operation. In the timer mode, a CLK/TRG pulse input starts the timer (Figure 11) on the second succeeding rising edge of CLK. The trigger pulse is asynchronous, and it must have a minimum width. A minimum lead time (210 ns) is required between the active edge of the CLK/TRG and the next rising edge of CLK to enable the prescaler on the following clock edge. If the CLK/TRG edge occurs closer than this, the initiation of the timer function is delayed one clock cycle. This corresponds to the start-up timing discussed in the programming section. The timer can also be started automatically if so programmed by the channel control word.

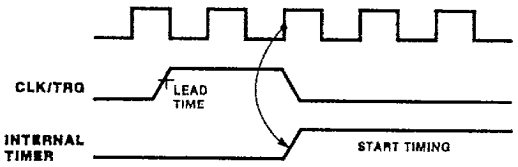


Figure 11. Timer Mode Timing

Counter Operation. In the counter mode, the CLK/TRG pulse input decrements the downcounter. The trigger is asynchronous, but the count is synchronized with CLK. For the decrement to occur on the next rising edge of CLK, the trigger edge must precede CLK by a minimum lead time as shown in Figure 12. If the lead time is less than specified, the count is delayed by one clock cycle. The trigger pulse must have a minimum width, and the trigger period must be at least twice the clock period. If the trigger repetition rate is faster than 1/3 the clock frequency, then TsCTR(Cs), AC Characteristics Specification 26, must be met.

The ZC/TO output occurs immediately after zero count, and follows the rising CLK edge.

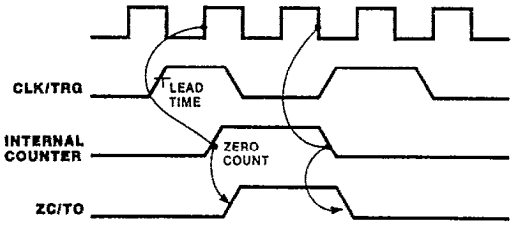


Figure 12. Counter Mode Timing

nested priority interrupts and return from interrupt, wherein the interrupt priority of a peripheral is determined by its location in a daisy chain. Two lines—IEI and IEO—in the CTC connect it to the system daisy chain. The device closest to the +5V supply has the highest priority (Figure 13). For additional information on the Z80 interrupt structure, refer to the *Z80 CPU Product Specification* and the *Z80 CPU Technical Manual*.

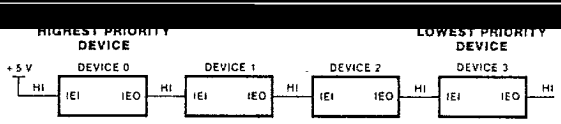


Figure 13. Daisy-Chain Interrupt Priorities

Within the Z80 CTC, interrupt priority is predetermined by channel number: Channel 0 has the highest priority, and Channel 3 the lowest. If a device or channel is being serviced with an interrupt routine, it cannot be interrupted by a device or channel with lower priority until service is complete. Higher priority devices or channels may interrupt the servicing of lower priority devices or channels.

A Z80 CTC channel may be programmed to request an interrupt every time its down counter reaches zero. Note that the CPU must be programmed for interrupt mode 2. Some time after the interrupt request, the CPU sends an interrupt acknowledge. The CTC interrupt control logic determines the highest priority channel that is requesting an interrupt. Then, if the CTC IEI input is High (indicating that it has priority within the system daisy chain) it places an 8-bit bits of this vector were written to the CTC during the programming process; the next two bits are provided by the CTC interrupt control logic as a binary code that identifies the highest priority channel requesting an interrupt; the low-order bit is always zero.

Interrupt Acknowledge Timing. Figure 14 shows interrupt acknowledge timing. After an interrupt request, the Z80 CPU sends the $\overline{M1}$ and $\overline{IORQ}$ request status when M1 is active—about two clock cycles earlier than $\overline{IORQ}$. $\overline{RD}$ is High to distinguish this cycle from an instruction fetch.

The CTC interrupt logic determines the highest priority channel requesting an interrupt. If the CTC interrupt enable input (IEI) is High, the highest priority interrupting channel within the CTC places its interrupt vector on the data bus when $\overline{IORQ}$ goes Low. Two wait states (T_{WA}) are automatically inserted at this time to allow the daisy chain to stabilize. Additional wait states may be added.

Return from Interrupt Timing. At the end of an interrupt service routine the RETI (Return From Interrupt) instruction initializes the daisy chain enable lines for proper control of nested priority interrupt handling. The CTC decodes the 2-byte RETI code internally and determines whether it is RETI timing.

If several Z80 peripherals are in the daisy chain, IEI settles active (High) on the chip currently being serviced when the opcode ED_{16} is decoded. If the following opcode is $4D_{16}$, the peripheral being serviced is released and its IEO becomes active. Additional wait states are allowed.

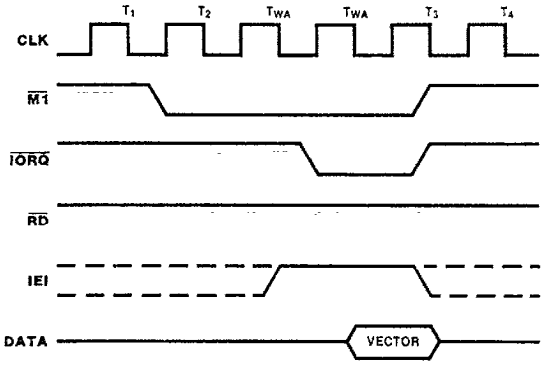


Figure 14. Interrupt Acknowledge Timing

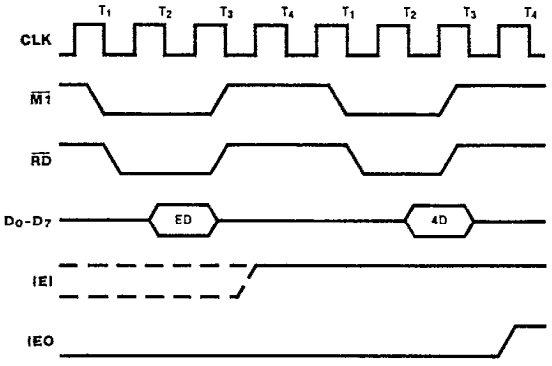


Figure 15. Return From Interrupt Timing

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

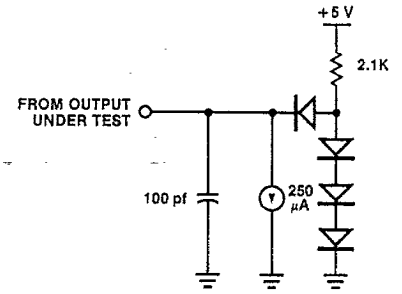
Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above these indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating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



DC CHARACTERISTICS (Z84C30/CMOS Z80 CTC)

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

Symbol	Parameter	Min	Max	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_{IH}	Input High Voltage	2.2	V_{CC}	V	
V_{IL}	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		7 [1]	mA	V_{CC}=5V
	- 6MHz		8 [1]	mA	CLK=4,6,8,10MHz
	- 8MHz		10 [1]	mA	V_{IH}=V_{CC}-0.2V
	- 10MHz		12 [1]	mA	V_{IL}=0.2V
I_{CC2}	Standby Supply Current		10	µA	V_{CC}=5V CLK=(0) V_{IH}=V_{CC}-0.2V V_{IL}=0.2V
I_{OHD}	Darlington Drive Current	-1.5	-5.0	mA	V_{OH}=1.5V REXT=1.1K ohm

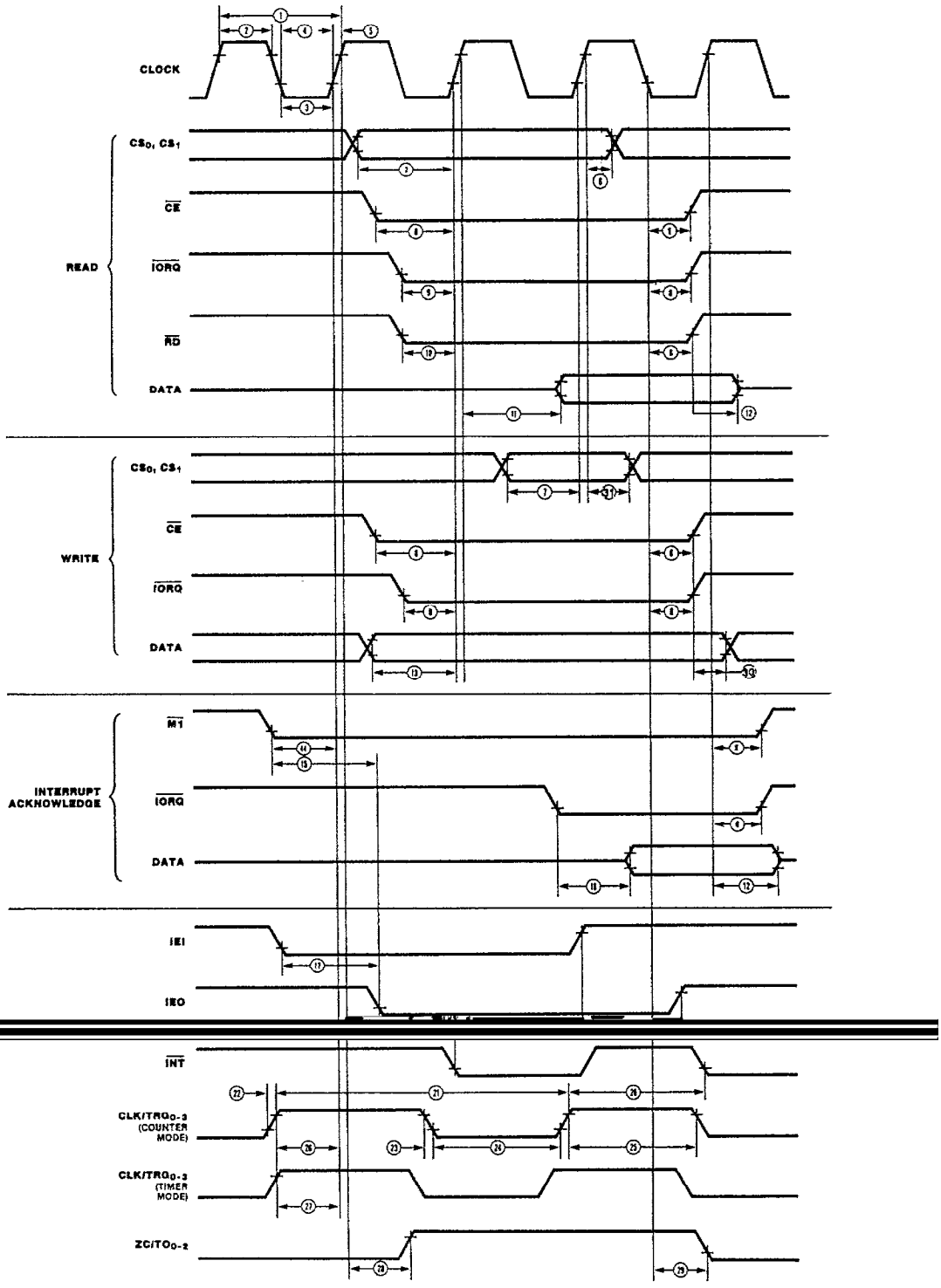
Note: [1] Measurements made with outputs floating.

CAPACITANCE

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

T_A = 25°C, f = 1 MHz
Unmeasured pins returned to ground.

AC CHARACTERISTICS (Z84C30/CMOS Z80 CTC)



AC CHARACTERISTICS (Z84C30/CMOS Z80 CTC Continued)

No	Symbol	Parameter	Z84C3004 * Min	Z84C3004 * Max	Z84C3006 Min	Z84C3006 Max	Z84C3008 Min	Z84C3008 Max	Z84C3010 Min	Z84C3010 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	
5	TrC	Clock Rise Time		30		20		10		10	
6	Th	All Hold Times	0		0		0		0		
7	TsCS(C)	/CS to Clock Rise Setup Time	160		100		50		35		
8	TsCE(C)	/CE to Clock Rise Setup Time	150		100		50		35		
9	TsIO(C)	/IORQ to Clock Rise Setup Time	115		70		40		35		
10	TsRD(C)	/RD Fall to Clock Rise Setup Time	115		70		40		35		
11	TdC(DO)	Clock Rise to Data Out Float Delay	200		130		90		90		[2]
12	TdRIr (DOz)	/RD, /IORQ rising to Data Outime Float Delay		50		40		40		40	
13	TsDI (C)	Data In to Clock rising set-up	50		40		30		30		
14	TsM1(C)	/M1 to Clock Rise Setup Time	90		70		50		40		
15	TdM1(IEO)	/M1 Fall to IEO Fall Delay (Interrupt Immediately Preceding /M1 Fall)		190		130		90		70	[3]
16	TdIO(DIO)	/IORQ Fall to Data Out Delay (/INTACK Cycle)		160		110		80		80	[2,6]
17	TdIEI(IEOf)	IEI Fall to IEO Fall Delay		130		100		70		70	[3]
18	TdIEI(IEOr)	IEI Rise to IEO Rise Delay (After ED Decode)		160		110		70		70	[3]
19	TdC(INT)	Clock Rise to /INT Fall Delay	(TcC+140)		(TcC+120)		(TcC+100)		(TcC+80)		[4]
20	TdCLK(INT)	CLK/TRG Rise to /INT Fall Delay TsCTR(C) Satisfied	(19)+(26)		(19)+(26)		(19)+(26)		(19)+(26)		[5]
		TsCTR(C) Not Satisfied	(1)+(19)+(26)		(1)+(19)+(26)		(1)+(19)+(26)		(1)+(19)+(26)		[5]
21	TcCTR	CLK/TRG Cycle Time	(2TcC)		(2TcC)		(2TcC)		(2TcC)		[5]
22	TrCTR	CLK/TRG Rise Time		50		40		30		30	
23	TfCTR	CLK/TRG Fall Time		50		40		30		30	
24	TwCTRh	CLK/TRG Width (Low)	200		120		90		90		
25	TwCTRl	CLK/TRG Width (High)	200		120		90		90		
26	TsCTR(Cs)	CLK/TRG Rise to Clock Rise Setup Time for Immediate Count	210		150		110		90		[5]
27	TsCTR(Ct)	CLK/TRG Rise to Clock Rise Setup Time for Enabling of Prescaler On Following Clock Rise	210		150		110		90		[4]

* 4 MHz Z84C30 is obsoleted and replaced by 6 MHz

Z84C30 AC CHARACTERISTICS (Continued)

No	Symbol	Parameter	Z84C3004* Min	Z84C3004* Max	Z84C3006 Min	Z84C3006 Max	Z84C3008 Min	Z84C3008 Max	Z84C3010 Min	Z84C3010 Max	Note
28	TdC(ZC/TOr)	Clock Rise to ZC/TO Rise Delay		190		140		100		80	
29	TdC(ZC/TOf)	Clock Fall to ZC/TO Fall Delay		190		140		100		80	
30	ThRIr(D)	/CE, /IORQ Rise to Data Hold	20		20		10		10		
31	ThC(CS)	Clock Rise to /CS Hold	20		20		10		10		

* RESET must be active for a minimum of 3 clock cycles.
Units in Nanoseconds

Notes:
[1] TcC = TwCh + TwCl + TrC + TfC.
[2] Increasing delay by 10nS for each 50pF increase in loading, 200pF max for data lines, and 100pF for control lines.
[3] Increase delay by 2nS for each 10pF increase in loading, 100pF max.
[4] Timer mode.
[5] Counter mode.
[6] 2.5TcT > (N-2)TdIEI(IEOf) + TdM1(IEO) + TsIEI(IO) + TTL Buffer Delay, if any.

* 4 MHz Z84C30 is obsoleted and replaced by 6 MHz

DC CHARACTERISTICS (Z8430/NMOS Z80 CTC)

Symbol	Parameter	Min	Max	Unit	Condition
V_{ILC}	Clock Input Low Voltage	−0.3[c]	+0.45[a]	V	
V_{IHC}	Clock Input High Voltage	V_{CC} −0.6[a]	V_{CC} +0.3[b]	V	
V_{IL}	Input Low Voltage	−0.3[c]	+0.8[a]	V	
V_{IH}	Input High Voltage	+2.2[a]	V_{CC}[b]	V	
V_{OL}	Output Low Voltage		+0.4[a]	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	+2.4[a]		V	I_{OH} = −250 μA
I_{CC}	Power Supply Current:		+120[a]	mA	
I_{LI}	Input Leakage Current		±10[a]	μA	V_{IN} = 0.4 to V_{CC}
I_{LO}	3-State Output Leakage Current in Float		±10[a]	μA	V_{OUT} = 0.4 to V_{CC}
I_{OHD}	Darlington Drive Current	−1.5[a]		mA	V_{OH} = 1.5V R_{EXT} = 390Ω

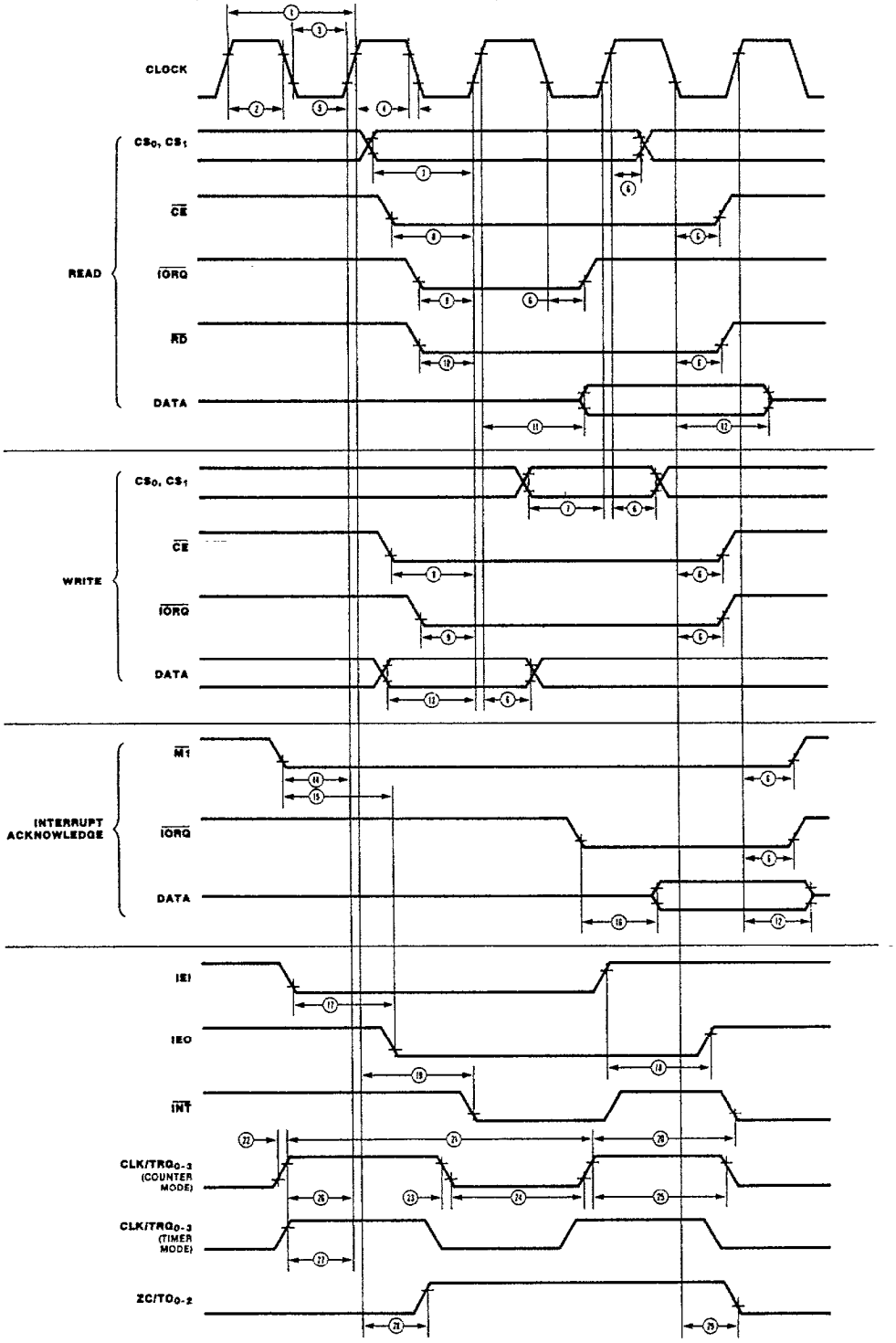
CAPACITANCE

Symbol	Parameter	Max	Unit
CLK	Clock Capacitance	20[c]	pf
C_{IN}	Input Capacitance	5[c]	pf
C_{OUT}	Output Capacitance	15[c]	pf

T_A = 25°C, f = 1 MHz
Unmeasured pins returned to ground.

Parameter Test Status:

[a] Tested
[b] Guaranteed
[c] Guaranteed by characterization/design


CLOCK
READ
CS_0, CS_1
$\overline{CE}$
$\overline{IORQ}$
$\overline{RD}$
DATA
WRITE
CS_0, CS_1
$\overline{CE}$
$\overline{IORQ}$
DATA
INTERRUPT ACKNOWLEDGE
$\overline{M1}$
$\overline{IORQ}$
DATA
IEI
IEO
$\overline{INT}$
CLK/TRG_{0-3} (COUNTER MODE)
CLK/TRG_{0-3} (TIMER MODE)
ZC/TO_{0-2}

AC CHARACTERISTICS (Z8430/NMOS Z80 CTC)

Number	Symbol	Parameter	Z0843004 Min	Z0843004 Max	Z0843006 Min	Z0843006 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	Th	All Hold Times	0		0		
7	TsCS(C)	CS to Clock ↑ Setup Time	160		100		
8	TsCE(C)	$\overline{CE}$ to Clock ↑ Setup Time	150		100		
9	TsIO(C)	$\overline{IORQ}$ ↓ to Clock ↑ Setup Time	115		70		
10	TsRD(C)	$\overline{RD}$ ↓ to Clock ↑ Setup Time	115		70		
11	TdC(DO)	Clock ↑ to Data Out Delay		200		130	[2]
12	TdC(DOz)	Clock ↓ to Data Out Float Delay		110		90	
13	TsDI(C)	Data In to Clock ↑ Setup Time	50		40		
14	TsM1(C)	$\overline{M1}$ to Clock ↑ Setup Time	90		70		
15	TdM1(IEO)	$\overline{M1}$ ↓ to IEO ↓ Delay (Interrupt immediately preceding M1)		190		130	[3]
16	TdIO(DOI)	$\overline{IORQ}$ ↓ to Data Out Delay (INTA Cycle)		160		110	[2]
17	TdIEI(IEOf)	IEI ↓ to IEO ↓ Delay		130		100	[3]
18	TdIEI(IEOr)	IEI ↑ to IEO ↑ Delay (After ED Decode)		160		110	[3]
19	TdC(INT)	Clock ↑ to $\overline{INT}$ ↓ Delay		(1) + 140		(1) + 120	[4,6]
20	TdCLK(INT)	CLK/TRG ↑ to $\overline{INT}$ ↓					
		tsCTR(C) satisfied		(19) + (26)		(19) + (26)	[5,6]
		tsCTR(C) not satisfied		(1) + (19) + (26)		(1) + (19) + (26)	[5,6]
21	TcCTR	CLK/TRG Cycle Time	2TcC		2TcC		[5]
22	TrCTR	CLK/TRG Rise Time		50		40	
23	TfCTR	CLK/TRG Fall Time		50		40	
24	TwCTRl	CLK/TRG Width (Low)	200		120		
25	TwCTRh	CLK/TRG Width (High)	200		120		

NOTES.

[1] TcC = TwCh + TwCl + TrC + TfC.

[2] Increase delay by 10 ns for each 50 pf increase in loading, 200 pf maximum for data lines, and 100 pf for control lines.

[3] Increase delay by 2 ns for each 10 pf increase in loading, 100 pf maximum.

[4] Timer mode

[5] Counter mode.

[6] Parenthetical numbers reference the table number of a parameter e.g., (1) refers to TcC.

† 2.5 TcC > (n − 2) TDIEI(IEOf) + TDM1(IEO) + TsIEI(IO) + TTL buffer delay, if any. $\overline{RESET}$ must be active for a minimum of 3 clock cycles. Units are nanoseconds unless otherwise specified.

AC CHARACTERISTICS (Z8430/NMOS Z80 CTC Continued)

Number	Symbol	Parameter	Z0843004 Min	Z0843004 Max	Z0843006 Min	Z0843006 Max	Notes†
26	TsCTR(Cs)	CLK/TRG ↑ to Clock ↑ Setup Time for Immediate Count	210		150		[5]
27	TsCTR(Ct)	CLK/TRG ↑ to Clock ↑ Setup Time for enabling of Prescaler on following clock ↑	210		150		[4]
28	TdC(ZC/TOr)	Clock ↑ to ZC/TO ↑ Delay		190		140	
29	TdC(ZC/TOf)	Clock ↓ to ZC/TO ↓ Delay		190		140	

NOTES:

[1] TcC = TwCh + TwCl + TrC + TfC.

[2] Increase delay by 10 ns for each 50 pf increase in loading, 200 pf maximum for data lines, and 100 pf for control lines.

[3] Increase delay by 2 ns for each 10 pf increase in loading, 100 pf maximum.

[4] Timer mode

[5] Counter mode.

[6] Parenthetical numbers reference the table number of a parameter. e.g., (1) refers to TcC.

† 2.5 TcC > (n – 2) TDIEI(IEOf) + TDM1(IEO) + TsIEI(IO) + TTL buffer delay, if any. $\overline{\text{RESET}}$ must be active for a minimum of 3 clock cycles. Units are nanoseconds unless otherwise specified.