Welcome to E-XFL.COM

Understanding **Embedded - Microprocessors**

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of **Embedded - Microprocessors**

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details

|	
Product Status	Obsolete
Core Processor	Z80
Number of Cores/Bus Width	1 Core, 8-Bit
Speed	6MHz
Co-Processors/DSP	-
RAM Controllers	-
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	-40°C ~ 100°C (TA)
Security Features	-
Package / Case	40-DIP (0.620", 15.75mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z84c0006pec
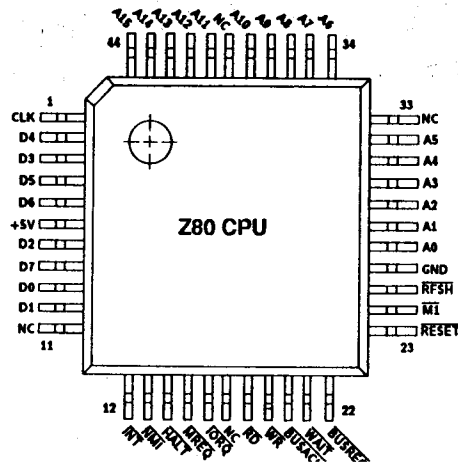
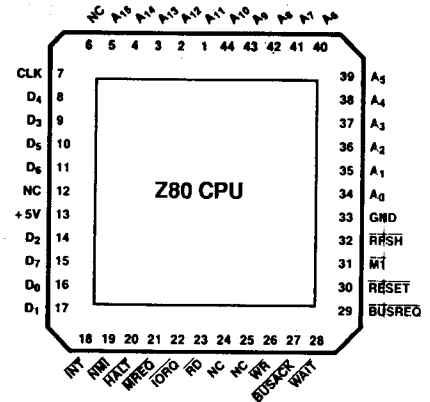Figure 2a. 44-Pin LQFP, Pin Assignments (Only available for 84C00)

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.
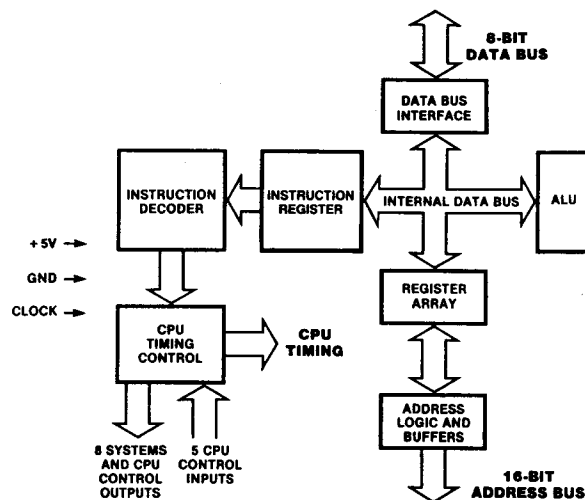

Figure 3. Z80C CPU Block Diagram

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	•	•	$IFF_2 \rightarrow$ Parity flag
LD A,R instruction execution	•	•	$IFF_2 \rightarrow$ Parity flag
Accept $\overline{NMI}$	0	•	Maskable interrupt $\overline{INT}$ disabled
RETN instruction execution	IFF_2	•	$IFF_2 \rightarrow IFF_1$ 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	Flags H	P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments	
LD r, r'	r ← r'	•	•	X	• X	•	•	•	01	r	r'		1	1	4	r, r'	Reg.
LD r, n	r ← n	•	•	X	• X	•	•	•	00	r	110		2	2	7	000	B
								← n →							001	C
LD r, (HL)	r ← (HL)	•	•	X	• X	•	•	•	01	r	110		1	2	7	010	D
LD r, (IX+d)	r ← (IX+d)	•	•	X	• 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	• X	•	•	•	11	111	101	FD	3	5	19	111	A
								01	r	110						
								← d →								
LD (HL), r	(HL) ← r	•	•	X	• X	•	•	•	01	110	r		1	2	7		
LD (IX+d), r	(IX+d) ← r	•	•	X	• X	•	•	•	11	011	101	DD	3	5	19		
								01	110	r						
								← d →								
LD (IY+d), r	(IY+d) ← r	•	•	X	• X	•	•	•	11	111	101	FD	3	5	19		
								01	110	r						
								← d →								
LD (HL), n	(HL) ← n	•	•	X	• X	•	•	•	00	110	110	36	2	3	10		
								← n →								
LD (IX+d), n	(IX+d) ← n	•	•	X	• X	•	•	•	11	011	101	DD	4	5	19		
								00	110	110	36					
								← d →								
								← n →								

16-BIT LOAD GROUP (Continued)

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 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 ←n→ ←n→	011 101	101 010	DD 2A	4	6	20	
LD IY, (nn)	$IY_H \leftarrow (nn+1)$ $IY_L \leftarrow (nn)$	•	•	X	• X	•	•	•	11 00 ←n→ ←n→	111 101	101 010	FD 2A	4	6	20	
LD (nn), HL	$(nn+1) \leftarrow H$ $(nn) \leftarrow L$	•	•	X	• X	•	•	•	00 ←n→ ←n→	100	010	22	3	5	16	
LD (nn), dd	$(nn+1) \leftarrow dd_H$ $(nn) \leftarrow dd_L$	•	•	X	• X	•	•	•	11 01 ←n→ ←n→	101 dd0	101 011	ED	4	6	20	
LD (nn), IX	$(nn+1) \leftarrow IX_H$ $(nn) \leftarrow IX_L$	•	•	X	• X	•	•	•	11 00 ←n→ ←n→	011 100	101 010	DD 22	4	6	20	
LD (nn), IY	$(nn+1) \leftarrow IY_H$ $(nn) \leftarrow IY_L$	•	•	X	• X	•	•	•	11 00 ←n→ ←n→	111 100	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	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'	•	•	X	•	X	•	•	•	11	011	001	D9	1	1	4	Register bank and auxiliary register bank exchange
	DE ↔ DE'																
	HL ↔ HL'																
EX (SP), HL	H ↔ (SP + 1)	•	•	X	•	X	•	•	•	11	100	011	E3	1	5	19	
	L ↔ (SP)																
EX (SP), IX	IX$_H$ ↔ (SP + 1)	•	•	X	•	X	•	•	•	11	011	101	DD	2	6	23	
	IX$_L$ ↔ (SP)									11	100	011	E3				
EX (SP), IY	IY$_H$ ↔ (SP + 1)	•	•	X	•	X	•	•	•	11	111	101	FD	2	6	23	
	IYL ↔ (SP)									11	100	011	E3				
LDI	(DE) ← (HL)	•	•	X	0	X	① ↕	0	•	11	101	101	ED	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
	DE ← DE + 1									10	100	000	A0				
	HL ← HL + 1																
	BC ← BC − 1																
LDIR	(DE) ← (HL)	•	•	X	0	X	② 0	0	•	11	101	101	ED	2	5	21	If BC ≠ 0
	DE ← DE + 1									10	110	000	B0	2	4	16	If BC = 0
	HL ← HL + 1																
	BC ← BC − 1																
	Repeat until																
	BC = 0																
LDD	(DE) ← (HL)	•	•	X	0	X	① ↕	0	•	11	101	101	ED	2	4	16	
	DE ← DE − 1									10	101	000	A8				
	HL ← HL − 1																
	BC ← BC − 1																
LDDR	(DE) ← (HL)	•	•	X	0	X	② 0	0	•	11	101	101	ED	2	5	21	If BC ≠ 0
	DE ← DE − 1									10	111	000	B8	2	4	16	If BC = 0
	HL ← HL − 1																
	BC ← BC − 1																
	Repeat until																
	BC = 0																
CPI	A − (HL)	↕	③ ↕	X	↕	X	① ↕	1	•	11	101	101	ED	2	4	16	
	HL ← HL + 1									10	100	001	A1				
	BC ← BC − 1																

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.

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	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								10	110	001	B1	2	4	16	If BC = 0 or A = (HL)
	BC ← BC – 1															
	Repeat until															
	A = (HL) or															
	BC = 0															
			③		①										
CPD	A – (HL)	‡	‡	X	‡	X ‡	1	•	11	101	101	ED	2	4	16	
	HL ← HL – 1								10	101	001	A9				
	BC ← BC – 1															
			③		①										
CPDR	A – (HL)	‡	‡	X	‡	X ‡	1	•	11	101	101	ED	2	5	21	If BC ≠ 0 and A ≠ (HL)
	HL ← HL – 1								10	111	001	B9	2	4	16	If BC = 0 or A = (HL)
	BC ← BC – 1															
	Repeat until															
	A = (HL) or															
	BC = 0															

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.
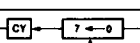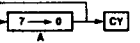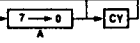③ Z flag is 1 if A = (HL), otherwise Z = 0.

8-BIT ARITHMETIC AND LOGICAL GROUP

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	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,
SUB s	A ← A – s	‡	‡	X	‡	X V	1	‡		[010]						(HL), (IX + d),
SBC A, s	A ← A – s – CY	‡	‡	X	‡	X V	1	‡		[011]						(IY + d) as
AND s	A ← A > s	‡	‡	X	1	X P	0	0		[100]						shown for ADD
OR s	A ← A > s	‡	‡	X	0	X P	0	0		[110]						instruction. The
XOR s	A ← A ⊕ s	‡	‡	X	0	X P	0	0		[101]						indicated bits
CP s	A – s	‡	‡	X	‡	X V	1	‡		[111]						replace the [000] in the ADD set above.

8-BIT ARITHMETIC AND LOGICAL GROUP (Continued)

Mnemonic	Symbolic Operation	S	Z	Flags 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	$\boxed{100}$		1	1	4
INC (HL)	(HL)← (HL)+1	‡	‡	X	‡	X	V	0	•	00	110	$\boxed{100}$		1	3	11
INC (IX+d)	(IX+d)← (IX+d)+1	‡	‡	X	‡	X	V	0	•	11	011	101	DD	3	6	23
								00	110	$\boxed{100}$				
								←d→						
INC (IY+d)	(IY+d)← (IY+d)+1	‡	‡	X	‡	X	V	0	•	11	111	101	FD	3	6	23
								00	110	$\boxed{100}$				
								←d→						
DEC m	m←m−1	‡	‡	X	‡	X	V	1	•			$\boxed{101}$				

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

GENERAL-PURPOSE ARITHMETIC AND CPU CONTROL GROUPS

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

ROTATE AND SHIFT GROUP

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLCA	CY ← [7←0] ← A	•	•	0	•	0	↕	00	000	111	07	1	1	4	Rotate left circular accumulator.
RLA	CY ← [7←0] ← A	•	•	0	•	0	↕	00	010	111	17	1	1	4	Rotate left accumulator.
RRCA	[7→0] → CY	•	•	0	•	0	↕	00	001	111	0F	1	1	4	Rotate right circular accumulator.
RRA	[7→0] → CY	•	•	0	•	0	↕	00	011	111	1F	1	1	4	Rotate right accumulator.

BIT SET, RESET AND TEST GROUP

Mnemonic	Symbolic Operation	S	Z	Flags 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	$Z \leftarrow \overline{r_b}$	X	↕	X	1	X	X	0	•	11	001	011	CB	2	2	8	r	Reg.
										01	b	r					000	B
BIT b, (HL)	$Z \leftarrow \overline{(HL)_b}$	X	↕	X	1	X	X	0	•	11	001	011	CB	2	3	12	001	C
										01	b	110					010	D
BIT b,(IX+d)$_b$	$Z \leftarrow \overline{(IX+d)_b}$	X	↕	X	1	X	X	0	•	11	011	101	DD	4	5	20	011	E
										11	001	011	CB				100	H
										←d→							101	L
										01	b	110					111	A
																	b	Bit Tested
BIT b, (IY+d)$_b$	$Z \leftarrow \overline{(IY+d)_b}$	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).

CALL AND RETURN 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
CALL nn	$(SP-1) \leftarrow PC_H$	•	•	X	•	X	•	11	001	101	CD	3	5	17	
	$(SP-2) \leftarrow PC_L$								← n →						
	$PC \leftarrow nn,$								← n →						
CALL cc,nn	If condition cc is false continue, otherwise same as CALL nn	•	•	X	•	X	•	11	cc	100		3	3	10	If cc is false.
									← n →			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 continue, otherwise same as RET	•	•	X	•	X	•	11	cc	000		1	1	5	If cc is false.
												1	3	11	If cc is true.
RETI	Return from interrupt	•	•	X	•	X	•	11	101	101	ED	2	4	14	
								01	001	101	4D				
RETN[1]	Return from non-maskable interrupt	•	•	X	•	X	•	11	101	101	ED	2	4	14	
								01	000	101	45				
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 $IFF_2 \rightarrow IFF_1$

SUMMARY OF FLAG OPERATION

Instructions	D7 S	Z		H		P/V	N	D0 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 usir... pe ands with packed BCD format.

PIN DESCRIPTIONS

A₀-A₁₅. *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.

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

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

D₀-D₇. *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.

HALT. *Halt State* (output, active Low). $\overline{HALT}$ 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.

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. $\overline{INT}$ is normally wired-OR and requires an external pullup for these applications.

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

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

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

NMI. *Non-Maskable Interrupt* (input, negative edge-triggered). $\overline{NMI}$ has a higher priority than $\overline{INT}$. 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.

RD. *Read* (output, active Low, 3-state). $\overline{RD}$ 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.

RESET. *Reset* (input, active Low). $\overline{RESET}$ 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 $\overline{RESET}$ must be active for a minimum of three full clock cycles before the reset operation is complete.

RFSH. *Refresh* (output, active Low). $\overline{RFSH}$, together with $\overline{MREQ}$, 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.

WAIT. *Wait* (input, active Low). $\overline{WAIT}$ 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 $\overline{WAIT}$ periods can prevent the CPU from properly refreshing dynamic memory.

WR. *Write* (output, active Low, 3-state). $\overline{WR}$ 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.
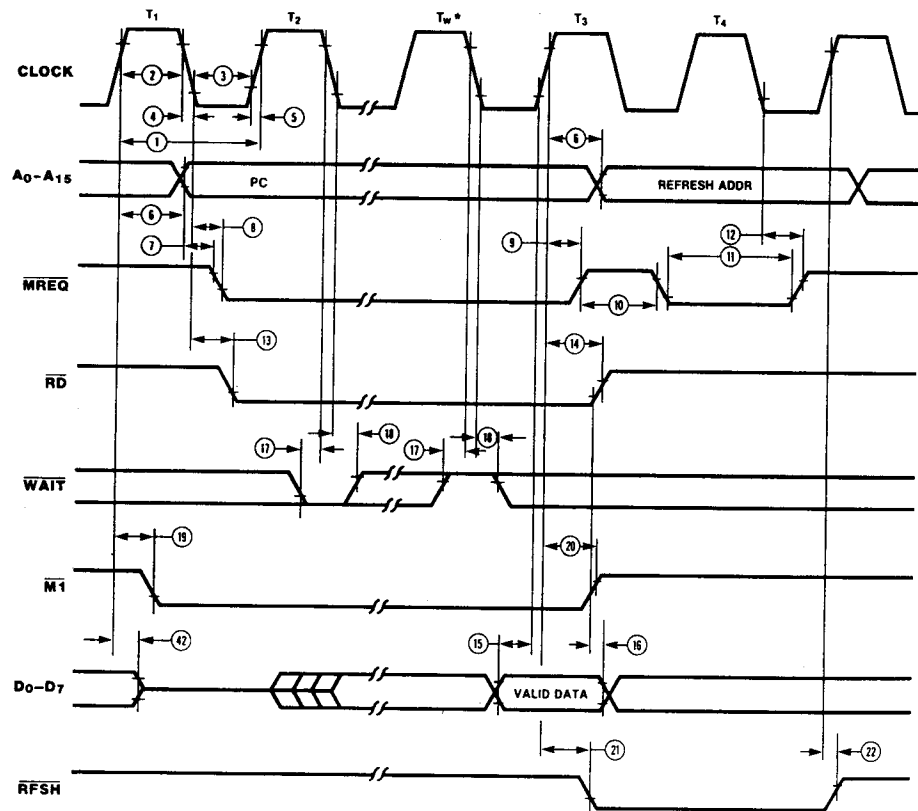
Figure 5. Instruction Opcode Fetch

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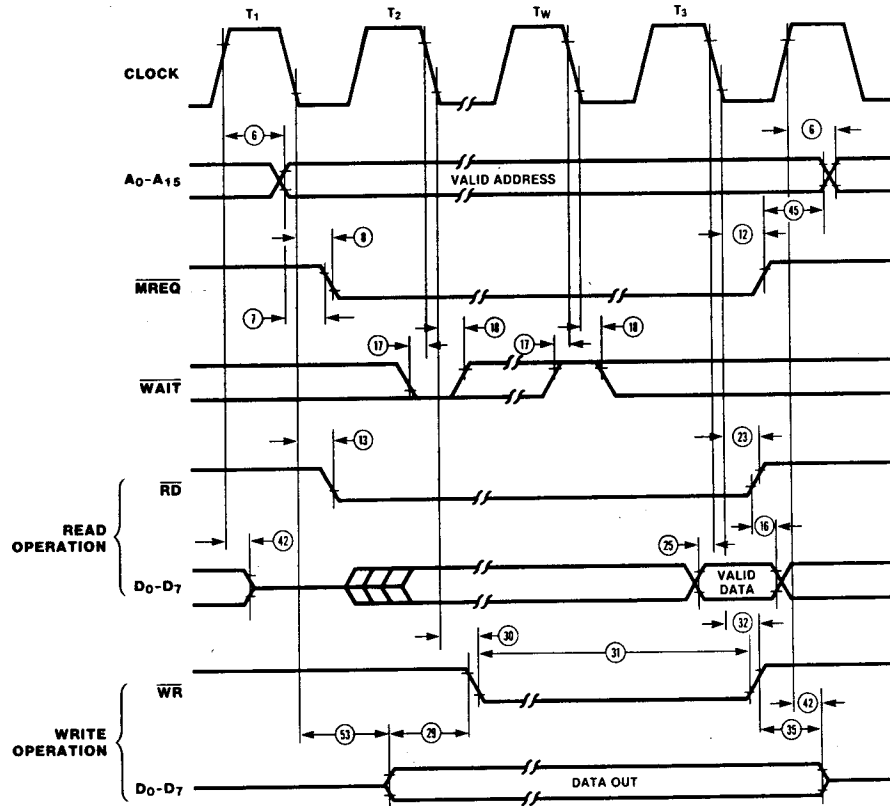
Figure 6. Memory Read or Write Cycles

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.
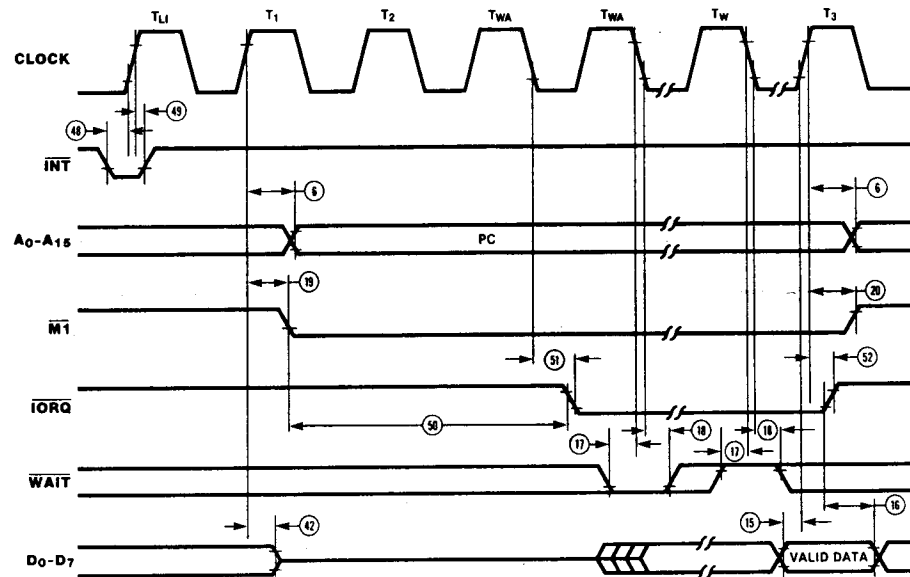
Non-Maskable Interrupt Request Cycle. $\overline{NMI}$ is sampled at the same time as the maskable interrupt input $\overline{INT}$ but has higher priority and cannot be disabled under software control. The subsequent timing is similar to that of a normal memory read operation except that data put on 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).
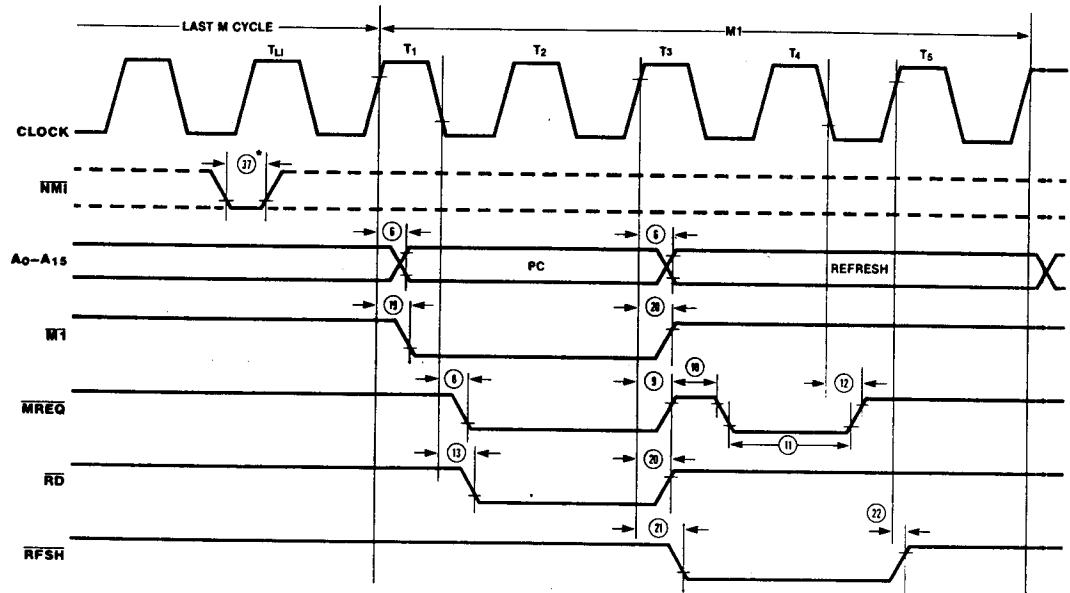
*Although $\overline{NMI}$ is an asynchronous input, to guarantee its being recognized on the following machine cycle, $\overline{NMI}$'s falling edge must occur no later than the rising edge of the clock cycle preceding the last state of any instruction cycle (T_{LI}).

Figure 9. Non-Maskable Interrupt Request Operation

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.
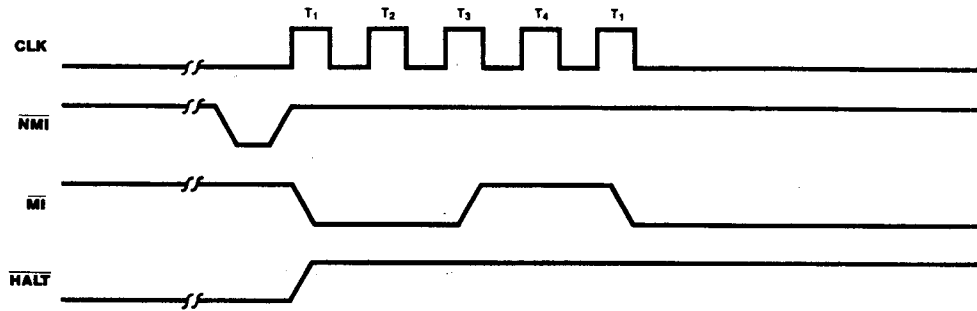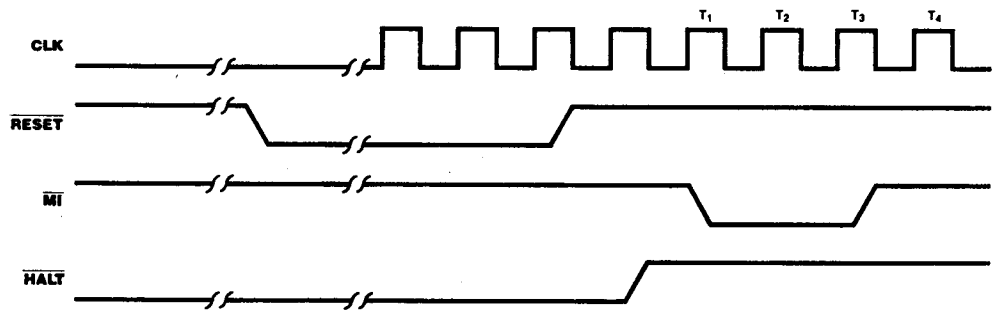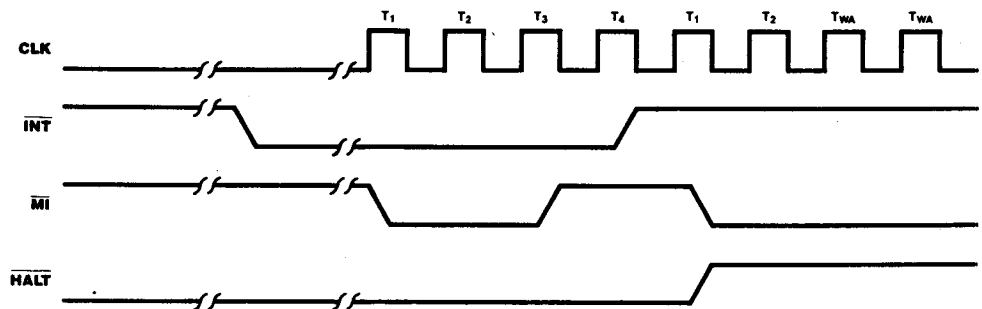

Figure 14a.

Figure 14b.

Figure 14c.

Figure 13. Power-Down Release

AC CHARACTERISTICS† (Z84C00/CMOS Z80 CPU; Continued)
V_{cc}=5.0V ± 10%, unless otherwise specified

No	Symbol	Parameter	Z84C0004** Min	Max	Z84C0006 Min	Max	Z84C0008 Min	Max	Z84C0010 Min	Max	Z84C0020[1] Min	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_{OH} = 1.5 V V_{IHC} = V_{CC} −0.6 V   FLOAT = ±0.5 V
V_{IL} = 0.8 V V_{OL} = 1.5 V V_{ILC} = 0.45 V

AC CHARACTERISTICS[†] (Z8400/NMOS Z80 CPU)

Number	Symbol	Parameter	Z0840004 Min	Z0840004 Max	Z0840006 Min	Z0840006 Max	Z0840008 Min	Z0840008 Max
1	TcC	Clock Cycle Time	250*		162*		125*	
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.

38

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)	BUSREQ Hold Time after Clock ↑	0		0		0	
40	TdCr(BUSACKf)	Clock ↑ to BUSACK ↓ Delay		100		90		80
41	TdCf(BUSACKr)	Clock ↓ to 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 (MREQ, IORQ, RD, and WR)		80		70		60
44	TdCr(Az)	Clock ↑ to Address Float Delay		90		80		70
45	TdCTr(A)	MREQ ↑, IORQ ↑, RD ↑, and WR ↑ to Address Hold Time	80*		35*		20*	
46	TsRESET(Cr)	RESET to Clock ↑ Setup Time	60		60		45	
47	ThRESET(Cr)	RESET to Clock ↑ Hold Time		0		0		0
48	TsINTf(Cr)	INT to Clock ↑ Setup Time	80		70		55	
49	ThINTr(Cr)	INT to Clock ↑ Hold Time		0		0		0
50	TdM1f(IORQf)	M1 ↓ to IORQ ↓ Delay	565*		365*		270*	
51	TdCf(IORQf)	Clock ↓ to IORQ ↓ Delay		85		70		60
52	TdCf(IORQr)	Clock ↑ 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_{OH} = 1.5 V
V_{IL} = 0.8 V V_{OL} = 1.5 V
V_{IHC} = V_{CC} − 0.6 V   FLOAT = ± 0.5 V
V_{ILC} = 0.45 V