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	10MHz
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	44-LCC (J-Lead)
Supplier Device Package	44-PLCC
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z84c0010vec00tr

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).

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 input follows a common timing cycle. After the interrupt has been detected by the CPU (provided that interrupts are enabled and $\overline{BUSREQ}$ is not active) a special 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.

Mode 0 Interrupt Operation. This mode is similar to the 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 similar to that for the $\overline{NMI}$. The principal difference is that the Mode 1 interrupt has only one restart location, 0038H.

Mode 2 Interrupt Operation. This interrupt mode has been 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 cycle. The CPU forms a pointer using this byte as the lower 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	•	•	$IFF_2 \rightarrow$ 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		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	•	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		H		P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments	
LD IX, (nn)	$IX_H \leftarrow (nn+1)$	•	•	X	•	X	•	•	•	11	011	101	DD	4	6	20		
	$IX_L \leftarrow (nn)$									00	101	010	2A					
											← n →							
											← n →							
LD IY, (nn)	$IY_H \leftarrow (nn+1)$	•	•	X	•	X	•	•	•	11	111	101	FD	4	6	20		
	$IY_L \leftarrow (nn)$									00	101	010	2A					
											← n →							
											← n →							
LD (nn), HL	$(nn+1) \leftarrow H$	•	•	X	•	X	•	•	•	00	100	010	22	3	5	16		
	$(nn) \leftarrow L$										← n →							
											← n →							
LD (nn), dd	$(nn+1) \leftarrow dd_H$	•	•	X	•	X	•	•	•	11	101	101	ED	4	6	20		
	$(nn) \leftarrow dd_L$									01	dd0	011						
											← n →							
											← n →							
LD (nn), IX	$(nn+1) \leftarrow IX_H$	•	•	X	•	X	•	•	•	11	011	101	DD	4	6	20		
	$(nn) \leftarrow IX_L$									00	100	010	22					
											← n →							
											← n →							
LD (nn), IY	$(nn+1) \leftarrow IY_H$	•	•	X	•	X	•	•	•	11	111	101	FD	4	6	20		
	$(nn) \leftarrow IY_L$									00	100	010	22					
											← n →							
											← n →							
LD SP, HL	$SP \leftarrow HL$	•	•	X	•	X	•	•	•	11	111	001	F9	1	1	6		
LD SP, IX	$SP \leftarrow IX$	•	•	X	•	X	•	•	•	11	011	101	DD	2	2	10		
										11	111	001	F9					
LD SP, IY	$SP \leftarrow IY$	•	•	X	•	X	•	•	•	11	111	101	FD	2	2	10		
										11	111	001	F9				qq	Pair
PUSH qq	$(SP-2) \leftarrow qq_L$	•	•	X	•	X	•	•	•	11	qq0	101		1	3	11	00	BC
	$(SP-1) \leftarrow qq_H$																01	DE
	$SP \rightarrow SP-2$																10	HL
PUSH IX	$(SP-2) \leftarrow IX_L$	•	•	X	•	X	•	•	•	11	011	101	DD	2	4	15	11	AF
	$(SP-1) \leftarrow IX_H$									11	100	101	E5					
	$SP \rightarrow SP-2$																	
PUSH IY	$(SP-2) \leftarrow IY_L$	•	•	X	•	X	•	•	•	11	111	101	FD	2	4	15		
	$(SP-1) \leftarrow IY_H$									11	100	101	E5					
	$SP \rightarrow SP-2$																	
POP qq	$qq_H \leftarrow (SP+1)$	•	•	X	•	X	•	•	•	11	qq0	001		1	3	10		
	$qqL \leftarrow (SP)$																	
	$SP \rightarrow SP+2$																	
POP IX	$IX_H \leftarrow (SP+1)$	•	•	X	•	X	•	•	•	11	011	101	DD	2	4	14		
	$IX_L \leftarrow (SP)$									11	100	001	E1					
	$SP \rightarrow SP+2$																	
POP IY	$IY_H \leftarrow (SP+1)$	•	•	X	•	X	•	•	•	11	111	101	FD	2	4	14		
	$IY_L \leftarrow (SP)$									11	100	001	E1					
	$SP \rightarrow SP+2$																	

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′	•	•	X	•	X	•	•	•	11	011	001	D9	1	1	4	Register bank
	DE ↔ DE′																and auxiliary
	HL ↔ HL′																register bank
																	exchange
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 ← DE+1									10	100	000	A0				(DE), increment
	HL ← HL+1																the pointers and
	BC ← BC−1																decrement the
																	byte counter
																	(BC)
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.

EXCHANGE, BLOCK TRANSFER, BLOCK SEARCH GROUPS (Continued)

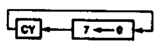
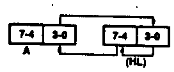
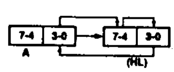
Mnemonic	Symbolic Operation	S	Z		H		P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
			③				①										
CPIR	A − (HL)	↕	↕	X	↕	X	↕	1	•	11	101	101	ED	2	5	21	If BC ≠ 0 and A ≠ (HL)
	HL ← HL + 1									10	110	001	B1	2	4	16	If BC = 0 or A = (HL)
	BC ← BC − 1																
	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		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, (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 ← A ⊕ s	↕	↕	X	0	X	P	0	0		101						
CP s	A − s	↕	↕	X	↕	X	V	1	↕		111						

ROTATE AND SHIFT 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
RLC r		↕	↕	X	0	X	P	0	↕	11 00	001 [000]	011 r	CB	2	2	8	Rotate left circular register r.
RLC (HL)		↕	↕	X	0	X	P	0	↕	11 00	001 000	011 110	CB	2	4	15	r Reg. 000 B 001 C 010 D 011 E 001 H 101 L 111 A
RLC (IX+d)	 r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	11 11 ←d→ 00	011 001 [000]	101 011 110	DD CB	4	6	23	
RLC (IY+d)		↕	↕	X	0	X	P	0	↕	11 11 ←d→ 00	111 001 [000]	101 011 110	FD CB	4	6	23	Instruction format and states are as shown for RLCs. To form new opcode replace [000] or RLCs with shown code.
RL m	 m=r,(HL,(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕		[010]						
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).
RRD		↕	↕	X	0	X	P	0	•	11 01	101 100	101 111	ED 67	2	5	18	The content of the upper half of the accumulator is unaffected.

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
		Flags								Opcode							
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	•	•	X	•	X	•	•	•	11	cc	010		3	3	10	010 NC (non-carry)
	is true PC←nn,										← n →						011 C (carry)
	otherwise										← n →						100 PO (parity add)
	continue																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,	•	•	X	•	X	•	•	•	00	111	000	38	2	2	7	If condition not met.
	continue										← e−2 →						
	If C=1,													2	3	12	If condition is met.
	PC ← PC+e																
JR NC, e	IF C=1,	•	•	X	•	X	•	•	•	00	110	000	30	2	2	7	If condition not met.
	continue										← e−2 →						
	If C=0,													2	3	12	If condition is met.
	PC ← PC+e																
JP Z, e	If Z=0	•	•	X	•	X	•	•	•	00	101	000	28	2	2	7	If condition not met.
	continue										← e−2 →						
	If Z=1,													2	3	12	If condition is met.
	PC ← PC+e																
JR NZ, e	If Z=1,	•	•	X	•	X	•	•	•	00	100	000	20	2	2	7	If condition not met.
	continue										← e−2 →						
	If Z=0,													2	3	12	If condition is met.
	PC ← PC+e																
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	•	•	X	•	X	•	•	•	00	010	000	10	2	2	8	If B=0
	If B=0,										← e−2 →						
	continue																
	If B≠0,													2	3	13	If B≠0.
	PC ← PC+e																

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.

INPUT AND OUTPUT 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
IN A, (n)	A ← (n)	•	•	X	•	X	•	•	•	11	011	01	DB	2	3	11	n to $A_0 \sim A_7$
											← n →						Acc. to $A_8 \sim A_{15}$
IN r, (C)	r ← (C)	↕	↕	X	↕	X	P	0	•	11	101	101	ED	2	3	12	C to $A_0 \sim A_7$
	if r = 110 only									01	r	000					B to $A_8 \sim A_{15}$
	the flags will																
	be affected																
INI	(HL) ← (C)	X	↕ ①	X	X	X	X	1	X	11	101	101	ED	2	4	16	C to $A_0 \sim A_7$
	B ← B − 1									10	100	010	A2				B to $A_8 \sim A_{15}$
	HL ← HL + 1																
INIR	(HL) ← (C)	X	1 ②	X	X	X	X	1	X	11	101	101	ED	2	5	21	C to $A_0 \sim A_7$
	B ← B − 1									10	110	010	B2		(If B≠0)		B to $A_8 \sim A_{15}$
	HL ← HL + 1													2	4	16	
	Repeat until														(If B = 0)		
	B = 0																
IND	(HL) ← (C)	X	↕ ①	X	X	X	X	1	X	11	101	101	ED	2	4	16	C to $A_0 \sim A_7$
	B ← B − 1									10	101	010	AA				B to $A_8 \sim A_{15}$
	HL ← HL − 1																
INDR	(HL) ← (C)	X	1 ②	X	X	X	X	1	X	11	101	101	ED	2	5	21	C to $A_0 \sim A_7$
	B ← B − 1									10	111	010	BA		(If B≠0)		B to $A_8 \sim A_{15}$
	HL ← HL − 1													2	4	16	
	Repeat until														(If B = 0)		
	B = 0																
OUT (n), A	(n) ← A	•	•	X	•	X	•	•	•	11	010	011	D3	2	3	11	n to $A_0 \sim A_7$
											← n →						Acc. to $A_8 \sim A_{15}$
OUT (C), r	(C) ← r	•	•	X	•	X	•	•	•	11	101	101	ED	2	3	12	C to $A_0 \sim A_7$
										01	r	001					B to $A_8 \sim A_{15}$
OUTI	(C) ← (HL)	X	↕ ①	X	X	X	X	1	X	11	101	101	ED	2	4	16	C to $A_0 \sim A_7$
	B ← B − 1									10	100	011	A3				B to $A_8 \sim A_{15}$
	HL ← HL + 1																
OTIR	(C) ← (HL)	X	1 ②	X	X	X	X	1	X	11	101	101	ED	2	5	21	C to $A_0 \sim A_7$
	B ← B − 1									10	110	011	B3		(If B≠0)		B to $A_8 \sim A_{15}$
	HL ← HL + 1													2	4	16	
	Repeat until														(If B = 0)		
	B = 0																
OUTD	(C) ← (HL)	X	↕ ①	X	X	X	X	1	X	11	101	101	ED	2	4	16	C to $A_0 \sim A_7$
	B ← B − 1									10	101	011	AB				B to $A_8 \sim A_{15}$
	HL ← HL − 1																
OTDR	(C) ← (HL)	X	1 ②	X	X	X	X	1	X	11	101	101	ED	2	5	21	C to $A_0 \sim A_7$
	B ← B − 1									10	111	011			(If B≠0)		B to $A_8 \sim A_{15}$
	HL ← HL − 1													2	4	16	
	Repeat until														(If B = 0)		
	B = 0																

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

	D_7							D_0	
Instructions	**S**	**Z**		**H**		**P/V**	**N**	**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.

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.

BUSACK. *Bus Acknowledge* (output, active Low). Bus Acknowledge indicates to the requesting device that the CPU address bus, data bus, and control signals MREQ, IORQ, RD, and 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 NMI and is always recognized at the end of the current machine cycle. BUSREQ forces the CPU address bus, data bus, and control signals MREQ, IORQ, RD, and WR to go to a high-impedance state so that other devices can control these lines. BUSREQ is normally wired-OR and requires an external pullup for these applications. Extended 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.

HALT. *Halt State* (output, active Low). 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. INT is normally wired-OR and requires an external pullup for these applications.

IORQ. *Input/Output Request* (output, active Low, 3-state). IORQ indicates that the lower half of the address bus holds a valid I/O address for an I/O read or write operation. IORQ is also generated concurrently with 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). M1, together with MREQ, indicates that the current machine cycle is the opcode fetch cycle of an instruction execution. M1, together with IORQ, indicates an interrupt acknowledge cycle.

MREQ. *Memory Request* (output, active Low, 3-state). 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). NMI has a higher priority than 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). 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.

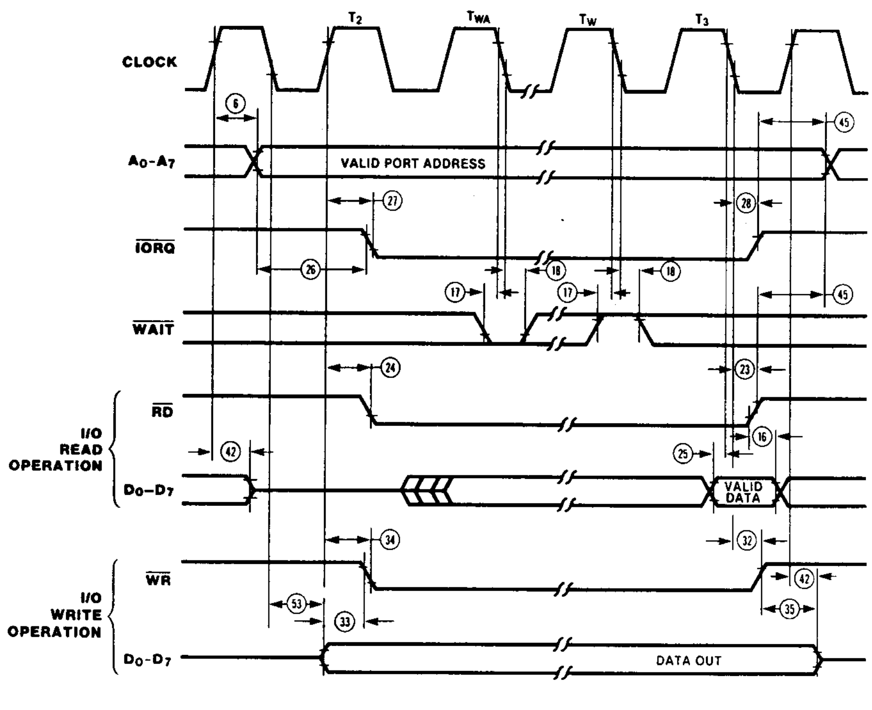
RESET. *Reset* (input, active Low). RESET initializes the CPU as follows: it resets the interrupt enable flip-flop, clears the PC and Registers I and R, and sets the interrupt status to Mode 0. During reset time, the address and data bus go to a high-impedance state, and all control output signals go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

RFSH. *Refresh* (output, active Low). RFSH, together with 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). 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 WAIT periods can prevent the CPU from properly refreshing dynamic memory.

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

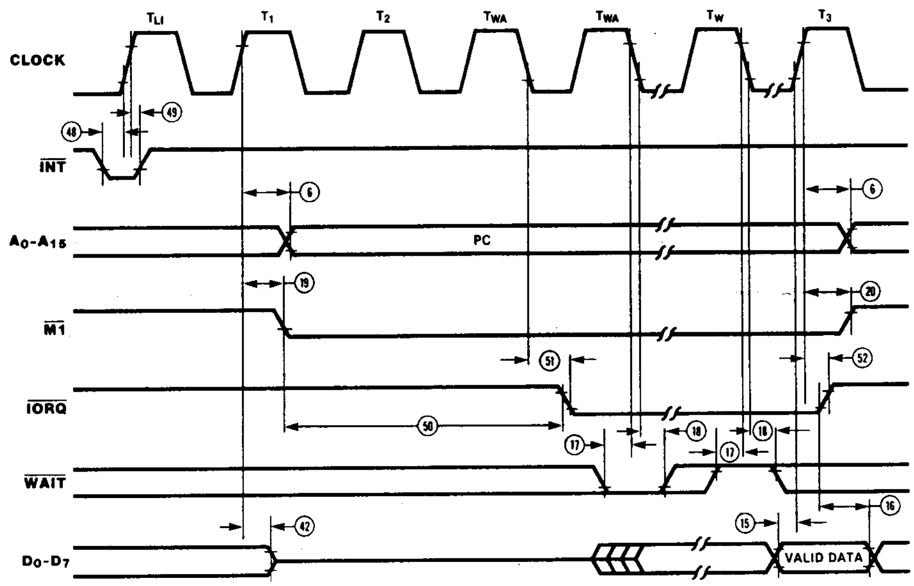
Input or Output Cycles. Figure 7 shows the timing for an I/O read or I/O write operation. During I/O operations, the CPU automatically inserts a single Wait state (T_{WA}). This extra Wait state allows sufficient time for an I/O port to decode the address from the port address lines.



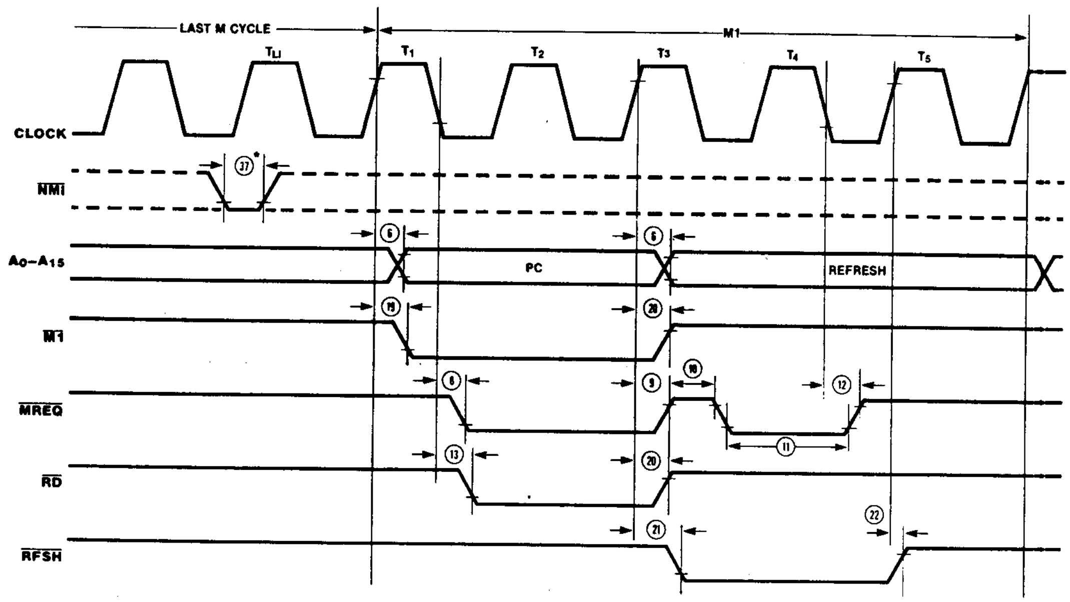
T_{WA} = One wait cycle automatically inserted by CPU.

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. NMI is sampled at the same time as the maskable interrupt input 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 NMI service routine located at address 0066H (Figure 9).



*Although NMI is an asynchronous input, to guarantee its being recognized on the following machine cycle, 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.

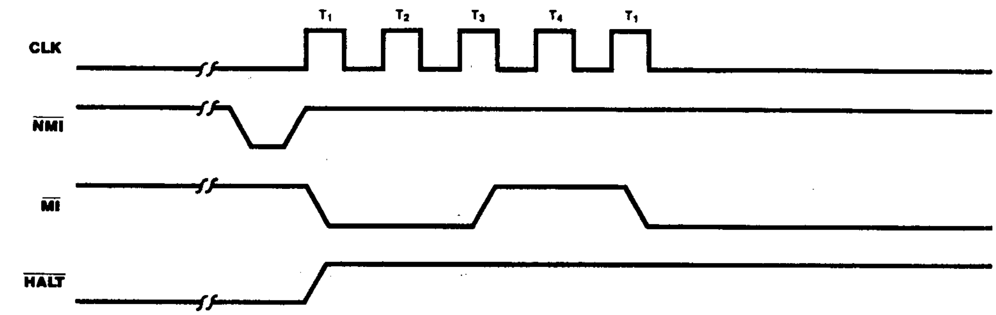


Figure 14a.

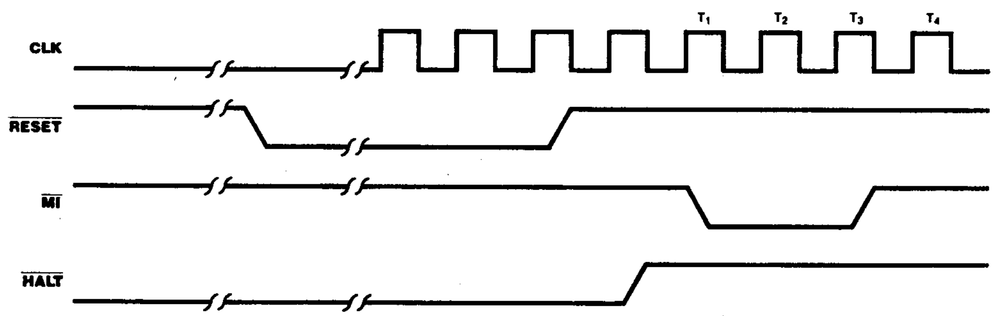


Figure 14b.

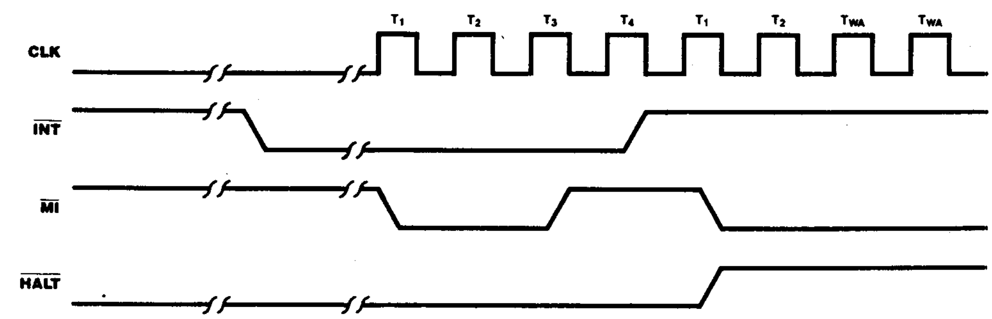


Figure 14c.

Figure 13. Power-Down Release

ABSOLUTE MAXIMUM RATINGS

Voltage on V_{CC} with respect to V_{SS} . . . . – 0.3V to + 7V
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

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

STANDARD TEST CONDITIONS

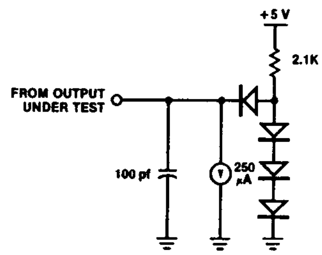
The DC Characteristics and capacitance sections below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0V). Positive current flows into the referenced pin.

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% points).

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



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 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_{CLOCK}	Clock Capacitance		**10**	pf
C_{IN}	Input Capacitance		5	pf
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 vaild 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	
32	TdCf(WRr)	Clock Fall to /WR Rise delay		80		70		60		55		40	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† (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.

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