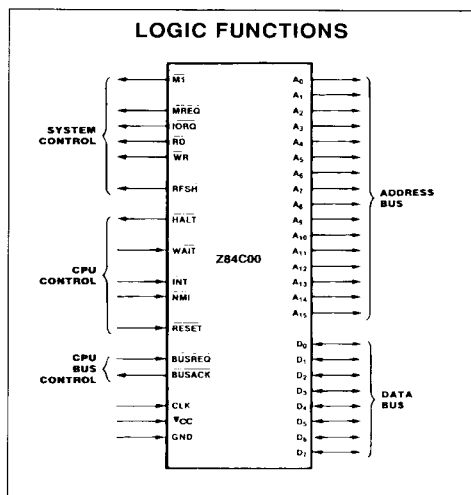
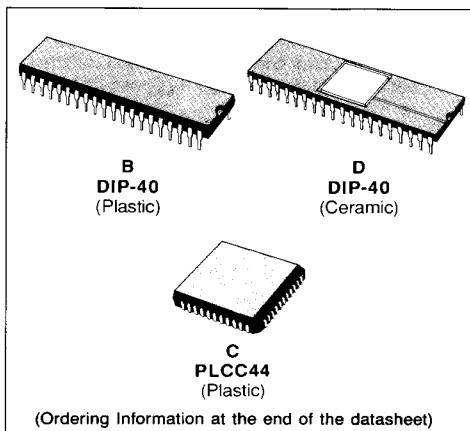


## Z80C CPU CMOS VERSION

- THE INSTRUCTION SET CONTAINS 158 INSTRUCTIONS. THE 78 INSTRUCTIONS OF THE 8080A ARE INCLUDED AS A SUBSET ; 8080A AND Z80 SOFTWARE COMPATIBILITY IS MAINTAINED
- 8 MHz, 6 MHz AND 4 MHz CLOCKS FOR THE Z80CH, Z80CB AND Z80CA, THE Z80C CPU, RESULT IN RAPID INSTRUCTION EXECUTION WITH CONSEQUENT HIGH DATA THROUGHPUT
- THE EXTENSIVE INSTRUCTION SET INCLUDES STRING, BIT, BYTE, AND WORD OPERATIONS. BLOCK SEARCHES AND BLOCK TRANSFERS TOGETHER WITH INDEXED AND RELATIVE ADDRESSING RESULT IN THE MOST POWERFUL DATA HANDLING CAPABILITIES IN THE MICRO-COMPUTER INDUSTRY
- THE Z80C MICROPROCESSORS AND ASSOCIATED FAMILY OF PERIPHERAL CONTROLLERS ARE LINKED BY A VECTORED INTERRUPT SYSTEM. THIS SYSTEM MAY BE DAISY-CHAINED TO ALLOW IMPLEMENTATION OF A PRIORITY INTERRUPT SCHEME. LITTLE, IF ANY, ADDITIONAL LOGIC IS REQUIRED FOR DAISY-CHAINING
- DUPLICATE SETS OF BOTH GENERAL-PURPOSE AND FLAG REGISTERS ARE PROVIDED, EASING THE DESIGN AND OPERATION OF SYSTEM SOFTWARE THROUGH SINGLE-CONTEXT SWITCHING, BACKGROUND-FOREGROUND PROGRAMMING, AND SINGLE-LEVEL INTERRUPT PROCESSING. IN ADDITION, TWO 16-BIT INDEX REGISTERS FACILITATE PROGRAM PROCESSING OF TABLES AND ARRAYS
- THERE ARE THREE MODES OF HIGH SPEED INTERRUPT PROCESSING : 8080 SIMILAR, NON-Z80 PERIPHERAL DEVICE, AND Z80 FAMILY PERIPHERAL WITH OR WITHOUT DAISY CHAIN
- ON-CHIP DYNAMIC MEMORY REFRESH COUNTER
- SINGLE  $5\text{ V} \pm 10\%$  POWER SUPPLY
- LOW POWER CONSUMPTION :
  - 9 mA TYP. AT 4 MHz
  - 15 mA TYP. AT 6 MHz

- 20 mA TYP. AT 8 MHz
- LESS THAN  $10\text{ }\mu\text{A}$  IN POWER DOWN MODE
- EXTENDED OPERATING TEMPERATURE
  - $40\text{ }^{\circ}\text{C}$  TO  $+85\text{ }^{\circ}\text{C}$



## DESCRIPTION

Z80 CMOS Family is fabricated using SGS-THOMSON' CMOS Silicon Gate Technology, which provides low power operation and high performance.

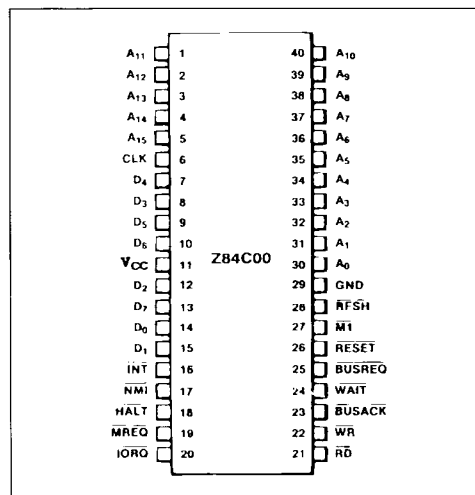
The Z80C CPU is third-generation single-chip microprocessors with exceptional computational power. They offer higher system throughput and more efficient memory utilization than comparable second-and third-generation microprocessors. The internal registers contain 208 bits of read/write memory that are 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 Z80C 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 + 5 V power source, all output signals are fully decoded and timed to control standard memory or peripheral circuits, and is supported by an extensive family of peripheral controllers. The internal block diagram (figure 3) shows the primary functions of the Z80C processors. Subsequent text provides more detail on the Z80C I/O controller family, registers, instruction set, interrupts and daisy chaining, and CPU timing.

**Figure 1 : Dual in Line Pin Configuration.**



**Figure 2 : Chip Carrier Pin Configuration.**

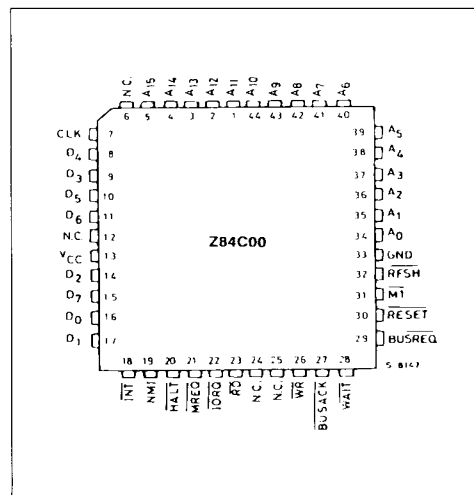
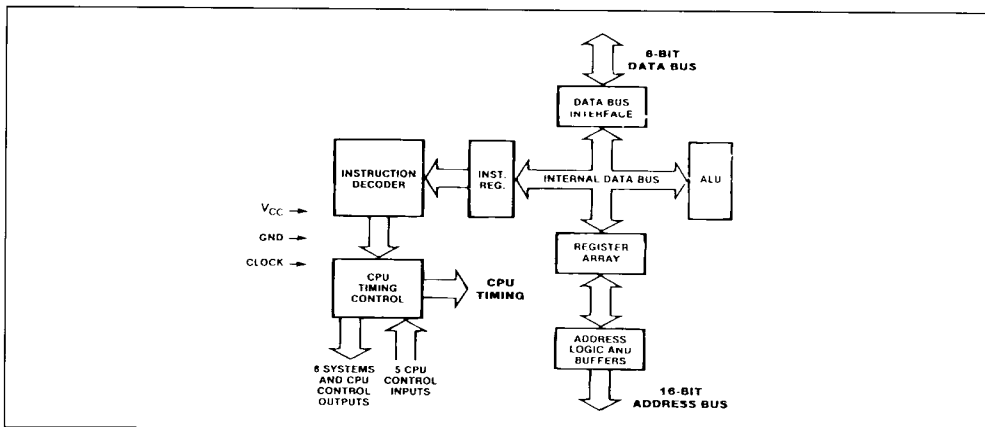


Figure 3 : CPU Block Diagram.



### 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

Figure 4 : CPU Registers.

#### Main Register Set

A Accumulator	F Flag Register
B General Purpose	C General Purpose
D General Purpose	E General Purpose
H General Purpose	L General Purpose

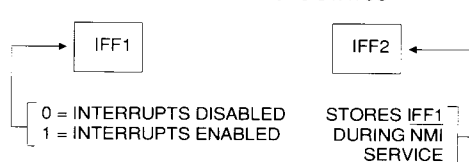
← 8 Bits →

#### Alternate Register Set

A' Accumulator	F' Flag Register
B' General Purpose	C' General Purpose
D' General Purpose	E' General Purpose
H' General Purpose	L' General Purpose

← 16 Bits →	
IX Index Register	
IY Index Register	
SP Stack Pointer	
PC Program Counter	
I Interrupt Vector	R Memory Refresh
← 8 Bits →	

#### INTERRUPT FLIP-FLOPS STATUS



#### INTERRUPT MODE FLIP-FLOPS

IMF <sub>a</sub>	IMF <sub>b</sub>	
0	0	INTERRUPT MODE 0
0	1	NOT USED
1	0	INTERRUPT MODE 1
1	1	INTERRUPT MODE 2

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 and additional pair of flip-flops which assists in identifying the interrupt mode at any particular time. Table 1 provides further information on these registers.

## INTERRUPTS : GENERAL OPERATION

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

The Z80C has a single response mode for interrupt service for the non-maskable interrupt. The maskable interrupt, INT, has three programmable response modes available.

These are :

- Mode 0 - compatible with 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 Z80 Family and compatible peripheral devices.

The CPU services interrupts by sampling the  $\overline{\text{NMI}}$  and 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 ( $\overline{\text{NMI}}$ )

The non-maskable interrupt cannot be disabled by program control and therefore will be accepted at all times by the CPU. NMI is usually reserved for servicing only the highest priority type interrupts, such as that for orderly shut-down after power failure has been detected.

**Table 1.** 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	See B, above.
D, D'	General Purpose	8	Can be used separately or as a 16-bit register with E.
E, E'	General Purpose	8	See D, above.
H, H'	General Purpose	8	Can be used separately or as a 16-bit register with L.
L, L'	General Purpose	8	See H, above.
			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. Lower seven bits are automatically incremented and all eight are placed on the address bus during each instruction fetch cycle refresh time.
IX	Index Register	16	Used for indexed addressing.
IY	Index Register	16	Same as IX, above.
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 <sub>1</sub> —IFF <sub>2</sub>	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).

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

### MASKABLE INTERRUPT ( $\overline{\text{INT}}$ )

Regardless of the interrupt mode set by the user, the Z80C 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  $\overline{\text{BUSREQ}}$  is not active) a special interrupt processing cycle begins. This is a special fetch (M1) cycle in which  $\overline{\text{IORQ}}$  becomes active rather than  $\overline{\text{MREQ}}$ , as in normal M1 cycle.

In addition, this special M1 cycle is automatically extended by two WAIT states, to allow for the time required to acknowledge the interrupt request.

### MODE 0 INTERRUPT OPERATION

This mode is similar with 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  $\overline{\text{NMI}}$ . The principal difference is that the Mode 1 interrupt has a restart location of 0038H only.

### MODE 2 INTERRUPT OPERATION

This interrupt mode has been designed to utilize most effectively the capabilities of the Z80C 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 sup-

plies the low-order byte of the 2-byte vector, bit 0 ( $\text{Ao}$ ) must be a zero.

### INTERRUPT PRIORITY (Daisy Chaining and Nested Interrupts).

The interrupt priority of each peripheral device is determined by its physical location within a daisy-chain configuration. Each device in the chain has an interrupt enable input line (IEI) and an interrupt enable output line (IEO), which is fed to the next lower priority device. The first device in the daisy chain has its IEI input hardwired to a High level. The first device has highest priority, while each succeeding device has a corresponding lower priority. This arrangement permits the CPU to select the highest priority interrupt from several simultaneously interrupting peripherals.

The interrupting device disables its IEO line to the next lower priority peripheral until it has been serviced. After servicing, its IEO line is raised, allowing lower priority peripherals to demand interrupt servicing.

The Z80C CPU will nest (queue) any pending interrupts or interrupts received while a selected peripheral is being serviced.

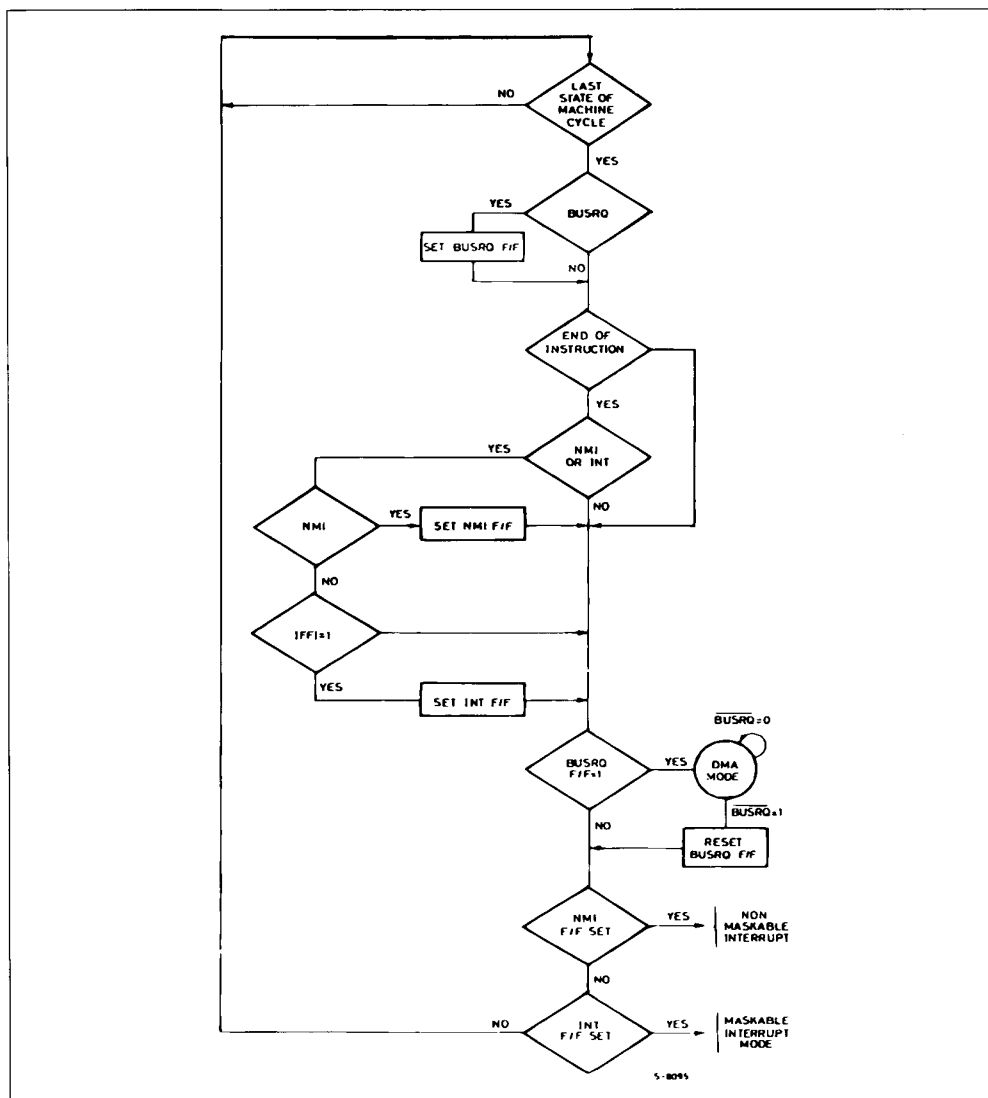
### INTERRUPT ENABLE/DISABLE OPERATION

Two flip-flops,  $\text{IFF}_1$  and  $\text{IFF}_2$ , 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*.

**Table 2.** State of Flip-Flops

Action	$\text{IFF}_2$	$\text{IFF}_1$	Comments
CPU Reset	0	0	Maskable Interrupt INT Disabled
DI Instruction Execution	0	0	Maskable Interrupt INT Disabled
EI Instruction Execution	1	1	Maskable Interrupt INT Enabled
LD A, I Instruction Execution	•	•	$\text{IFF}_2 \rightarrow$ Parity Flag
LD A, R Instruction Execution	•	•	$\text{IFF}_2 \rightarrow$ Parity Flag
Accept $\overline{\text{NMI}}$	0	$\text{IFF}_1$	$\text{IFF}_1 \rightarrow \text{IFF}_2$ (maskable interrupt INT disabled)
RETN Instruction Execution	$\text{IFF}_2$	•	$\text{IFF}_2 \rightarrow \text{IFF}_1$ at Completion of an NMI Service Routine.

## CPU INTERRUPT SEQUENCE



- Notes :
1.  $\overline{\text{INT}}$  and  $\overline{\text{NMI}}$  are always acted on at the end of an instruction.
  2.  $\text{BUSRQ}$  is acted on at the end of a machine cycle.
  3. While the CPU is in the DMA MODE, it will not respond to active inputs on  $\overline{\text{INT}}$  or  $\overline{\text{NMI}}$ .
  4. These three inputs are acted on in the following order of priority.
    - 1)  $\text{BUSRQ}$  -- highest
    - 2)  $\overline{\text{NMI}}$
    - 3)  $\overline{\text{INT}}$  -- lowest.

## INSTRUCTION SET

The Z80C 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 Z80C instruction set and shows the assembly language mnemonic, the operation, the flag status, and gives comments on each instruction. The *Z80 CPU Technical Manual* and *Z80 CPU Programming Manual* 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 SHIFT
- 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

### 16-BIT ARITHMETIC GROUP

Symbol	Symbolic Operation	Flags					Opcode					Hex	N° of Bytes	N° of M Cycles	N° of T States	Comments
		S	Z	H	P/V	N	C	7	6	5	4	3	2	1	0	
ADD HL, ss	HL ← HL + ss	•	•	X	X	X	•	0	↑	00	ss1	001	1	3	11	ss Reg.
ADC HL, ss	HL ← HL + ss + CY	↑	↑	X	X	X	V	0	↑	11	101	101	2	4	15	00 BC 01 DE 10 IX 11 SP
SBC HL, ss	HL ← HL + ss - CY	↑	↑	X	X	X	V	1	↑	11	101	101	2	4	15	01 DE 10 IX 11 SP
ADD IX, pp	IX ← + pp	•	•	X	X	X	•	0	↑	11	011	101	2	4	15	pp Reg.
ADD IY, rr	IY ← IY + rr	•	•	X	X	X	•	0	↑	11	111	101	2	4	15	00 BC 01 DE 10 IX 11 SP
INC ss	ss ← ss + 1	•	•	X	•	X	•	•	•	00	ss0	011	1	1	6	10 IX
INC IX	IX ← IX + 1	•	•	X	•	X	•	•	•	11	011	101	2	2	10	rr Reg.
INC IY	IY ← IY + 1	•	•	X	•	X	•	•	•	00	100	011	2	2	10	00 BC 01 DE 10 IX 11 SP
DEC ss	ss ← ss - 1	•	•	X	•	X	•	•	•	00	ss1	011	1	1	6	10 IY
DEC IX	IX ← IX - 1	•	•	X	•	X	•	•	•	11	011	101	2	2	10	11 SP
DEC IY	IY ← IY - 1	•	•	X	•	X	•	•	•	11	111	101	2	2	10	
								00	101	011			2B			

**Notes :** ss is any of the register pairs BC, DE, HL, SP.  
pp is any of the register pairs BC, DE, IX, SP.  
rr is any of the register pairs BC, DE, IY, SP.

## INSTRUCTION SET (continued)

## 8-BIT LOAD GROUP

Symbol	Symbolic Operation	Flags						Opcode				Hex	N° of Bytes	N° of M Cycles	N° of T States	Comments	
		S	Z	H	P/V	N	C	76	543	210							
LD r, r' LD r, n	$r \leftarrow r'$ $r \leftarrow n$	•	•	X	•	X	•	•	•	01	r	r'		1	1	4	r, r' Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
		•	•	X	•	X	•	•	•	00	r	110		2	2	7	
											$\leftarrow n \rightarrow$						
LD r, (HL) LD r, (IX+d)	$r \leftarrow (HL)$ $r \leftarrow (IX + d)$	•	•	X	•	X	•	•	•	01	r	110	DD	1	2	7	
		•	•	X	•	X	•	•	•	11	011	101		3	5	19	
										01	r	101					
											$\leftarrow d \rightarrow$						
LD r, (IY+d)	$r \leftarrow (IY + d)$	•	•	X	•	X	•	•	•	11	111	101	FD	3	5	19	
										01	r	110					
											$\leftarrow d \rightarrow$						
LD(HL), r LD(IX+d), r	$(HL) \leftarrow r$ $(IX + d) \leftarrow r$	•	•	X	•	X	•	•	•	01	110	r	DD	1	2	7	
		•	•	X	•	X	•	•	•	11	011	101		3	5	19	
										01	110	r					
											$\leftarrow d \rightarrow$						
LD(IY+d), r	$(IY + d) \leftarrow r$	•	•	X	•	X	•	•	•	11	111	101	FD	3	5	19	
										01	110	r					
											$\leftarrow d \rightarrow$						
LD(HL), n	$(HL) \leftarrow n$	•	•	X	•	X	•	•	•	00	110	110	36	2	3	10	
											$\leftarrow n \rightarrow$						
LD(IX+d), n	$(IX + d) \leftarrow n$	•	•	X	•	X	•	•	•	11	011	101	DD	4	5	19	
										00	110	110	36				
											$\leftarrow d \rightarrow$						
											$\leftarrow n \rightarrow$						
LD(IY+d), n	$(IY + d) \leftarrow n$	•	•	X	•	X	•	•	•	11	111	101	FD	4	5	19	
										00	110	110	36				
											$\leftarrow d \rightarrow$						
											$\leftarrow n \rightarrow$						
LD A, (BC) LD A, (DE) LD A, (nn)	$A \leftarrow (BC)$ $A \leftarrow (DE)$ $A \leftarrow (nn)$	•	•	X	•	X	•	•	•	00	001	010	0A	1	2	7	
		•	•	X	•	X	•	•	•	00	011	010	1A	1	2	7	
		•	•	X	•	X	•	•	•	00	111	010	3A	3	4	13	
											$\leftarrow n \rightarrow$						
											$\leftarrow n \rightarrow$						
LD(BC), A LD(DE), A LD(nn), A	$(BC) \leftarrow A$ $(DE) \leftarrow A$ $(nn) \leftarrow A$	•	•	X	•	X	•	•	•	00	000	010	02	1	2	7	
		•	•	X	•	X	•	•	•	00	010	010	12	1	2	7	
		•	•	X	•	X	•	•	•	00	110	010	32	3	4	13	
											$\leftarrow n \rightarrow$						
											$\leftarrow n \rightarrow$						
LD A, I LD A, R LD I, A LD R, A	$A \leftarrow I$ $A \leftarrow R$ $I \leftarrow A$ $R \leftarrow A$	•	•	X	0	X	IFF	0	•	11	101	101	ED	2	2	9	
										01	010	111	57				
		•	•	X	0	X	IF	0	•	11	101	101	ED	2	2	9	
										01	011	111	5F				
		•	•	X	•	X	•	•	•	11	101	101	ED	2	2	9	
										01	000	111	47				
		•	•	X	•	X	•	•	•	11	101	101	ED	2	2	9	
										01	001	111	4F				

Notes : r, r' means any of the registers A, B, C, D, E, H, L.

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

For an explanation of flag notation and symbols for mnemonic tables, see Symbolic Notation section following tables.



## INSTRUCTION SET (continued)

## 16-BIT LOAD GROUP

Symbol	Symbolic Operation	Flags						Opcode				Hex	N° of Bytes	N° of M Cycles	N° of T States	Comments	
		S	Z		H	P/V	N	C	76	543	210						
LD dd, nn	dd ← nn	•	•	X	•	X	•	•	•	00	dd0	001		3	3	10	dd Pair 00 BC 01 DE 10 HL 11 SP
										← n →							
LD IX, nn	IX ← nn	•	•	X	•	X	•	•	•	11	011	101	DD 21	4	4	14	
										00	100	001					
										← n →							
LD IY, nn	IY ← nn	•	•	X	•	X	•	•	•	11	111	101	FD 21	4	4	14	
										00	100	001					
										← n →							
										← n →							
LD HL, (nn)	H ← (nn + 1) L ← (nn)	•	•	X	•	X	•	•	•	00	101	010	2A	3	5	16	
										← n →							
										← n →							
LD dd, (nn)	dd <sub>H</sub> ← (nn + 1) dd <sub>L</sub> ← (nn)	•	•	X	•	X	•	•	•	11	101	101	ED	4	6	20	
										01	dd1	011					
										← n →							
										← n →							
LD IX, (nn)	IX <sub>H</sub> ← (nn + 1) IX <sub>L</sub> ← (nn)	•	•	X	•	X	•	•	•	11	011	101	DD 2A	4	6	20	
										01	101	010					
										← n →							
										← n →							
LD IY, (nn)	IY <sub>H</sub> ← (nn + 1) IY <sub>L</sub> ← (nn)	•	•	X	•	X	•	•	•	11	111	101	FD 2A	4	6	20	
										00	101	010					
										← n →							
										← n →							
LD (nn), HL	(nn + 1) ← H (nn) ← L	•	•	X	•	X	•	•	•	00	100	010	22	3	5	16	
										← n →							
										← n →							
LD (nn), dd	(nn + 1) ← dd <sub>H</sub> (nn) ← dd <sub>L</sub>	•	•	X	•	X	•	•	•	11	101	101	ED	4	6	20	
										01	dd0	011					
										← n →							
										← n →							
LD (nn), IX	(nn + 1) ← IX <sub>H</sub> (nn) ← IX <sub>L</sub>	•	•	X	•	X	•	•	•	11	011	101	DD 22	4	6	20	
										00	100	010					
										← n →							
										← n →							
LD (nn), IY	(nn + 1) ← IY <sub>H</sub> (nn) ← IY <sub>L</sub>	•	•	X	•	X	•	•	•	11	111	101	FD 22	4	6	20	
										00	100	010					
										← n →							
										← n →							

Notes : dd is any of the register pairs BC, DE, HL, SP.

qq is any of the registers pairs AF, BC, DE, HL.

(PAIR)<sub>H</sub>, (PAIR)<sub>L</sub> refer to high order and low order eight bits of the register pair respectively,  
e.g., BC<sub>L</sub> = C, AF<sub>H</sub> = A.

**INSTRUCTION SET** (continued)

**16-BIT LOAD GROUP** (continued)

Symbol	Symbolic Operation	Flags						Opcode				Hex	N° of Bytes	N° of M Cycles	N° of T States	Comments
		S	Z	H	P/V	N	C	76	543	210						
LD SP, HL	SP ← HL	•	•	X	•	X	•	•	•	11 111 001	F9	1	1	6	qq Pair 00 BC 01 DE 10 HL 11 AF	
LD SP, IX	SP ← IX	•	•	X	•	X	•	•	•	11 011 101	DD	2	2	10		
LD SP, IY	SP ← IY	•	•	X	•	X	•	•	•	11 111 001	F9	2	2	10		
										11 111 101	FD					
PUSH qq	(SP - 2) ← qq <sub>L</sub> (SP - 1) ← qq <sub>H</sub> SP → SP - 2	•	•	X	•	X	•	•	•	11 111 001	F9	1	3	11		
										11 111 101	F9					
PUSH IX	(SP - 2) ← IX <sub>L</sub> (SP - 1) ← IX <sub>H</sub> SP → SP - 2	•	•	X	•	X	•	•	•	11 111 101	FD	2	4	15		
										11 100 101	E5					
PUSH IY	(SP - 2) ← IY <sub>L</sub> (SP - 1) ← IY <sub>H</sub> SP → SP - 2	•	•	X	•	X	•	•	•	11 111 101	FD	2	4	15		
										11 100 101	E5					
POP qq	qq <sub>H</sub> ← (SP + 1) qq <sub>L</sub> ← (SP) SP → SP + 2	•	•	X	•	X	•	•	•	11 qq0 001		1	3	10		
POP IX	IX <sub>H</sub> ← (SP + 1) IX <sub>L</sub> ← (SP) SP → SP + 2	•	•	X	•	X	•	•	•	11 011 101	DD	2	4	14		
										11 100 001	E1					
POP IY	IY <sub>H</sub> ← (SP + 1) IY <sub>L</sub> ← (SP) SP → SP + 2	•	•	X	•	X	•	•	•	11 111 101	FD	2	4	14		
										11 100 001	E1					

**Notes :** dd is any of the register pairs BC, DE, HL, SP.  
 qq is any of the registers pairs AF, BC, DE, HL.  
 (PAIR)<sub>H</sub>, (PAIR)<sub>L</sub> refer to high order and low order eight bits of the register pair respectively,  
 e.g., BC<sub>L</sub> = C, AF<sub>H</sub> = A.

**EXCHANGE, BLOCK TRANSFER, BLOCK SEARCH GROUPS**

Symbol	Symbolic Operation	Flags						Opcode			Hex	N° of Bytes	N° of M Cycles	N° of T States	Comments		
		S	Z	H		P/V	N	C	76	543						210	
EX DE, HL, EX AF, AF' EXX	DE ↔ HL,	•	•	X	•	X	•	•	•	11	101	011	EB	1	1	4	Register Bank and Auxiliary Register Bank Exchange
	AF ↔ AF'	•	•	X	•	X	•	•	•	00	001	000	08	1	1	4	
	BC ↔ BC'	•	•	X	•	X	•	•	•	11	011	001	D9	1	1	4	
	DE ↔ DE'																
	HL ↔ HL'																
EX (SP), HL	H ↔ (SP + 1) L ↔ (SP)	•	•	X	•	X	•	•	•	11	100	011	E3	1	5	19	
EX (SP), IX	IX <sub>H</sub> ↔ (SP + 1) IX <sub>L</sub> ↔ (SP)	•	•	X	•	X	•	•	•	11	011	101	DD	2	6	23	
										11	100	011	E3				
EX (SP), IY	IY <sub>H</sub> ↔ (SP + 1) IY <sub>L</sub> ↔ (SP)	•	•	X	•	X	•	•	•	11	111	101	FD	2	6	23	
										11	100	011	E3				

**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.

## INSTRUCTION SET (continued)

## EXCHANGE, BLOCK TRANSFER, BLOCK SEARCH GROUPS (continued)

Symbol	Symbolic Operation	Flags						Opcode				Hex	N° of Bytes	N° of M Cycles	N° of T States	Comments
		S	Z	H	P/V	N	C	76	543	210						
LDI	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC - 1	•	•	X	0	X	① ↓	0	•	11 101 101 10 100 000	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	① 0	0	•	11 101 101 10 110 000	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	•	11 101 101 10 101 000	ED A8	2	4	16		
LDDR	(DE) ← (HL) DE ← DE - 1 HL ← HL - 1 BC ← BC - 1 Repeat Until BC = 0	•	•	X	0	X	② 0	0	•	11 101 101 10 111 000	ED B8	2 2	5 4	21 16		If BC ≠ 0 If BC = 0
CPI	A ← (HL) HL ← HL + 1 BC ← BC - 1	↓	② ↓	X	↓	X	① ↓	1	•	11 101 101 10 100 001	ED A1	2	4	16		
CPIR	A ← (HL) HL ← HL + 1 BC ← BC - 1 Repeat Until A = (HL) or BC = 0	↓	② ↓	X	↓	X	① ↓	1	•	11 101 101 10 110 001	ED B1	2 2	5 4	21 16		If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)
CPD	A ← (HL) HL ← HL + 1 BC ← BC - 1	↓	② ↓	X	↓	X	① ↓	1	•	11 101 101 10 101 001	ED A9	2	4	16		
CPDR	A ← (HL) HL ← HL + 1 BC ← BC - 1 Repeat Until A = (HL) or BC = 0	↓	② ↓	X	↓	X	① ↓	1	•	11 101 101 10 111 001	ED B9	2 2	5 4	21 16		If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)

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.

## INSTRUCTION SET (continued)

## 8-BIT ARITHMETIC AND LOGICAL GROUP

Symbol	Symbolic Operation	Flags						Opcode			Hex	N° of Bytes	N° of M Cycles	N° of T States	Comments
		S	Z	H	P/V	N	C	7 6	5 4 3	2 1 0					
ADD A, r ADD A, n	$A \leftarrow A + r$ $A \leftarrow A + n$	$\downarrow$	$\downarrow$	X	$\downarrow$	X	V	0	$\downarrow$	$\downarrow$		1	1	4	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
ADD A, (HL) ADD A, (IX+d)	$A \leftarrow A + (HL)$ $A \leftarrow A + (IX + d)$	$\downarrow$	$\downarrow$	X	$\downarrow$	X	V	0	$\downarrow$	$\downarrow$		1	2	7	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.
		$\downarrow$	$\downarrow$	X	$\downarrow$	X	V	0	$\downarrow$	$\downarrow$	DD	3	5	19	
ADD A, (IY+d)	$A \leftarrow A + (IY + d)$	$\downarrow$	$\downarrow$	X	$\downarrow$	X	V	0	$\downarrow$	$\downarrow$	FD	3	5	19	
ADC A, s SUB s SBC A, s AND s OR s XOR s CP s INC r INC (HL) INC (IX + d)	$A \leftarrow A + s + CY$ $A \leftarrow A - s$ $A \leftarrow A - s - CY$ $A \leftarrow A \wedge s$ $A \leftarrow A \vee s$ $A \leftarrow A \oplus s$ $A \leftarrow s$ $r \leftarrow r + 1$ $(HL) \leftarrow (HL) + 1$ $(IX + d) \leftarrow (IX + d) + 1$	$\downarrow$	$\downarrow$	X	$\downarrow$	X	V	0	$\downarrow$	$\downarrow$					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.
		$\downarrow$	$\downarrow$	X	$\downarrow$	X	V	1	$\downarrow$	$\downarrow$					
		$\downarrow$	$\downarrow$	X	$\downarrow$	X	V	1	$\downarrow$	$\downarrow$					
		$\downarrow$	$\downarrow$	X	1	X	P	0	0						DD
		$\downarrow$	$\downarrow$	X	0	X	P	0	0						
		$\downarrow$	$\downarrow$	X	0	X	P	0	0						
		$\downarrow$	$\downarrow$	X	$\downarrow$	X	V	1	$\downarrow$	$\downarrow$		1	1	4	DD
		$\downarrow$	$\downarrow$	X	$\downarrow$	X	V	0	$\downarrow$	$\downarrow$		1	3	11	
		$\downarrow$	$\downarrow$	X	$\downarrow$	X	V	0	$\downarrow$	$\downarrow$		3	6	23	
INC (IY+d)	$(IY + d) \leftarrow (IY + d) + 1$	$\downarrow$	$\downarrow$	X	$\downarrow$	X	V	0	$\downarrow$	$\downarrow$	FD	3	6	23	
DEC m	$m \leftarrow m - 1$	$\downarrow$	$\downarrow$	X	$\downarrow$	X	V	1	$\downarrow$	$\downarrow$					

## INSTRUCTION SET (continued)

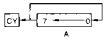
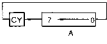

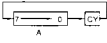
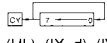
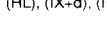
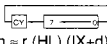
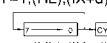
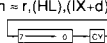
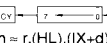

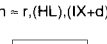
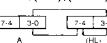

## GENERAL-PURPOSE ARITHMETIC AND CPU CONTROL GROUPS

Symbol	Symbolic Operation	Flags						Opcode			Hex	N° of Bytes	N° of M Cycles	N° of T States	Comments
		S	Z	H	P/V	N	C	7	6	5 4 3 2 1 0					
DAA	Converters acc ; content into packed BCD following add or subtract with packed BCD operands	↑	↑	X	↑	X	P	*	↑	00 100 111	27	1	1	4	Decimal Adjust Accumulator.
CPL	$A \leftarrow \bar{A}$	*	*	X	1	X	*	1	*	00 101 111	2F	1	1	4	Complement Accumulator (one's complement)
NEG	$A \leftarrow 0 - A$	↑	↑	X	↑	X	V	1	↓	11 101 101 01 000 100	ED 44	2	2	8	Negate Acc. (two's complement).
CCF	$CY \leftarrow \overline{CY}$	*	*	X	X	X	*	0	↓	00 111 111	3F	1	1	4	Complement Carry Flag.
SCF	$CY \leftarrow 1$	*	*	X	0	X	*	0	1	00 110 111	37	1	1	4	Set Carry Flag.
NOP	No Operation	*	*	X	0	X	*	0	1	00 000 000	00	1	1	4	
HALT	CPU Halted	*	*	X	*	X	*	*	*	01 110 110	76	1	1	4	
DI*	$IFF \leftarrow 0$	*	*	X	*	X	*	*	*	11 110 011	F3	1	1	4	
EI*	$IFF \leftarrow 1$	*	*	X	*	X	*	*	*	11 111 011	FB	1	1	4	
IM 0	Set Interrupt Mode 0	*	*	X	*	X	*	*	*	11 101 101 01 000 110	ED 46	2	2	8	
IM 1	Set Interrupt Mode 1	*	*	X	*	X	*	*	*	11 101 101 01 010 110	ED 56	2	2	8	
IM2	Set Interrupt Mode 2	*	*	X	*	X	*	*	*	11 101 101 01 011 110	ED 5E	2	2	8	

Notes : 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.

## INSTRUCTION SET (continued)

## ROTATE AND SHIFT GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	Opcode 76 543 210 Hex	No. of Bytes	No. of Cycles	No. of T States	Comments
RLCA		•	•	X	0	X	•	0 000 111 07	1	1	4	Rotate left circular accumulator.
RLA		•	•	X	0	X	•	00 010 111 17	1	1	4	Rotate left accumulator.
RRCA		•	•	X	0	X	•	00 001 111 0F	1	1	4	Rotate right circular accumulator.
RRA		•	•	X	0	X	•	00 011 111 1F	1	1	4	Rotate right accumulator.
RLC r		•	•	X	0	X	P	0 001 011 00	2	2	8	Rotate left circular register r.
RLC (HL)		•	•	X	0	X	P	0 001 011 00	2	4	15	Rotate left circular register r.
RLC (IX + d)		•	•	X	0	X	P	0 001 011 00	4	6	23	Rotate left circular register r.
RLC (IY + d)		•	•	X	0	X	P	0 001 011 00	4	6	23	Rotate left circular register r.
RL m		•	•	X	0	X	P	0 001 011 00	2	4	15	Rotate left circular register r.
RRC m		•	•	X	0	X	P	0 001 011 00	2	4	15	Rotate right circular register r.
RR m		•	•	X	0	X	P	0 001 011 00	2	4	15	Rotate right register r.
SLA m		•	•	X	0	X	P	0 001 011 00	2	4	15	Rotate left shift register r.
SRA m		•	•	X	0	X	P	0 001 011 00	2	4	15	Rotate right shift register r.
SRL m		•	•	X	0	X	P	0 001 011 00	2	4	15	Rotate right shift register r.
RLD		•	•	X	0	X	P	0 001 011 00	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	0 001 011 00	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.

## INSTRUCTION SET (continued)

## BIT SET, RESET AND TEST GROUP

Symbol	Symbolic Operation	Flags						Opcode			Hex	N° of Bytes	N° of M Cycles	N° of T States	Comments
		S	Z	H	P/V	N	C	76	543	210					
BIT b, r	$Z \leftarrow r_b$	X	↓	X	1	X	X	0	•	11 001 011 01 b r	CB	2	2	8	r, Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
BIT b, (HL)	$Z \leftarrow (\overline{HL})_b$	X	↓	X	1	X	X	0	•	11 001 011 01 b 110	CB	2	3	12	
BIT b, (IX+d) <sub>b</sub>	$Z \leftarrow (\overline{IX+d})_b$	X	↓	X	1	X	X	0	•	11 011 101 11 001 011 ← d → 01 b 110	DD CB	4	5	20	
BIT b, (IY+d) <sub>b</sub>	$Z \leftarrow (\overline{IY+d})_b$	X	↓	X	1	X	X	0	•	11 111 101 11 001 011 ← d → 01 b 110	FD CB	4	5	20	b Bit Tested 000 0 001 1 010 2 011 3 100 4 101 5 110 6 111 7
SET b, r	$r_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 001 011 11 b r	CB	2	2	8	
SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 001 011 11 b 110	CB	2	4	15	
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 011 101 11 001 011 ← d → 11 b 110	DD CB	4	6	23	
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 111 101 11 001 011 ← d → 11 b 110	FD CB	4	6	23	
RES b, m	$m_b \leftarrow 0$ $m \equiv r, (HL),$ $(IX+d),$ $(IY+d)$	•	•	X	•	X	•	•	•	10					To form new opcode replace 11 of SET b, s with 10. Flags and time states for SET instruction.

**Notes :** The notation  $m_b$  indicates bit b (0 to 7) of location m.

## INSTRUCTION SET (continued)

## JUMP GROUP

Symbol	Symbolic Operation	Flags						Opcode			Hex	N° of Bytes	N° of M Cycles	N° of T States	Comments			
		S	Z	H	P/V	N	C	7	6	5						4	3	2
JP nn	PC ← nn	•	•	X	•	X	•	•	•	11	000	011		C3	3	3	10	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
JP cc, nn	If condition cc is true PC ← nn, otherwise continue	•	•	X	•	X	•	•	•	11	cc	010			3	3	10	
JR e	PC ← PC + e	•	•	X	•	X	•	•	•	00	011	000		18	2	3	12	
JR C, e	If C = 0, continue If C = 1, PC ← PC+e	•	•	X	•	X	•	•	•	00	111	000		38	2	2	7	If condition not met.
															2	3	12	If condition is met.
JR NC, e	If C = 1, continue If C = 0, PC ← PC+e	•	•	X	•	X	•	•	•	00	110	000		30	2	2	7	If condition not met.
															2	3	12	If condition is met.
JP Z, e	If Z = 0 continue If Z = 1, PC ← PC+e	•	•	X	•	X	•	•	•	00	101	000		28	2	2	7	If condition not met.
															2	3	12	If condition is met.
JR NZ, e	If Z = 1, continue If Z = 0, PC ← PC+e	•	•	X	•	X	•	•	•	00	100	000		20	2	2	7	If condition not met.
															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 If B = 0, continue If B ≠ 0, PC ← PC+e	•	•	X	•	X	•	•	•	00	010	000		10	2	2	8	If B = 0.
															2	3	13	If B ≠ 0.

**Notes :** e represents the extension in the relative addressing mode. e is signed 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.



## INSTRUCTION SET (continued)

## CALL AND RETURN GROUP

Symbol	Symbolic Operation	Flags						Opcode				Hex	N° of Bytes	N° of M Cycles	N° of T States	Comments	
		S	Z	H	P/V	N	C	7	6	5	4						3
CALL nn	(SP - 1) ← PC <sub>H</sub> (SP - 2) ← PC <sub>L</sub> PC ← nn	•	•	X	•	X	•	•	•	11	001	101	CD	3	5	17	If cc is false.  If cc is true.
CALL cc, nn	If condition cc is false continue, otherwise same as CALL nn	•	•	X	•	X	•	•	•	11	cc	100		3	3	10	
										← n →							
										← n →				3	5	17	
RET	PC <sub>L</sub> ← (SP) PC <sub>H</sub> ← (SP + 1)	•	•	X	•	X	•	•	•	11	001	001	C9	1	3	10	If cc is false.  If cc is true cc Condition
RET cc	If condition cc is false continue, otherwise same as RET	•	•	X	•	X	•	•	•	11	cc	000		1	1	5	
														1	3	11	
RETI	Return from interrupt	•	•	X	•	X	•	•	•	11	101	101	ED	2	4	14	000 NZ Non-zero
										01	001	101	4D				001 Z Zero
RETN <sup>1</sup>	Return from non-maskable interrupt	•	•	X	•	X	•	•	•	11	101	101	ED	2	4	14	010 NC Non-carry
										01	000	101	45				011 C Carry
																	100 PO Parity Odd
																	101 PE Parity Even
																	110 P Sign Positive
																	111 M Sign Negative
																	t p
																	000 0H
																	001 08H
																	010 10H
																	011 18H
																	100 20H
																	101 28H
																	110 30H
																	111 38H

Note : RETN loads IFF<sub>2</sub> – IFF<sub>1</sub>.

## INSTRUCTION SET (continued)

## INPUT AND OUTPUT GRUP

Symbol	Symbolic Operation	Flags						Opcode				Hex	N° of Bytes	N° of M Cycles	N° of T States	Comments	
		S	Z	H	P/V	N	C	7	6	5	4						3
IN A, (n)	$A \leftarrow (n)$	•	•	X	•	X	•	•	•	11	011	011	DB	2	3	11	n to A <sub>0</sub> - A <sub>7</sub> Acc. to A <sub>8</sub> - A <sub>15</sub>
IN r, (C)	$r \leftarrow (C)$ If r = 110 only the flags will be affected	•	•	X	•	X	P	0	•	11	101	101	ED	2	3	12	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>
INI	$(HL) \leftarrow (C)$ $B \leftarrow B - 1$ $HL \leftarrow HL + 1$	X	①	X	X	X	X	1	X	11	101	101	ED A2	2	4	16	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>
INIR	$(HL) \leftarrow (C)$ $B \leftarrow B - 1$ $HL \leftarrow HL + 1$ Repeat until B = 0	X	1	X	X	X	X	1	X	11	101	101	ED B2	2 2	5 (if B≠0) 4 (if B=0)	21 16	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>
IND	$(HL) \leftarrow (C)$ $B \leftarrow B - 1$ $HL \leftarrow HL - 1$	X	①	X	X	X	X	1	X	11	101	101	ED AA	2	4	16	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>
INDR	$(HL) \leftarrow (C)$ $B \leftarrow B - 1$ $HL \leftarrow HL - 1$ Repeat until B = 0	X	1	X	X	X	X	1	X	11	101	101	ED BA	2 2	5 (if B≠0) 4 (if B=0)	21 16	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>
OUT (n), A	$(n) \leftarrow A$	•	•	X	•	X	•	•	•	11	010	011	D3	2	3	11	n to A <sub>0</sub> - A <sub>7</sub> Acc. to A <sub>8</sub> - A <sub>15</sub>
OUT (C), r	$(C) \leftarrow r$	•	•	X	•	X	•	•	•	11	101	101	ED	2	3	12	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>
OUTI	$(C) \leftarrow (HL)$ $B \leftarrow B - 1$ $HL \leftarrow HL + 1$	X	①	X	X	X	X	1	X	11	101	101	ED A3	2	4	16	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>
OTIR	$(C) \leftarrow (HL)$ $B \leftarrow B - 1$ $HL \leftarrow HL + 1$ Repeat until B = 0	X	1	X	X	X	X	1	X	11	101	101	ED B3	2 2	5 (if B≠0) 4 (if B=0)	21 16	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>
OUTD	$(C) \leftarrow (HL)$ $B \leftarrow B - 1$ $HL \leftarrow HL - 1$	X	①	X	X	X	X	1	X	11	101	101	ED AB	2	4	16	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>
OTDR	$(C) \leftarrow (HL)$ $B \leftarrow B - 1$ $HL \leftarrow HL + 1$ Repeat until B = 0	X	1	X	X	X	X	1	X	11	101	101	ED	2 2	5 (if B≠0) 4 (if B=0)	21 16	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>

**Note :** 1. If the result of B - 1 is zero the Z flag is set, otherwise it is reset.

## SUMMARY OF FLAG OPERATION

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.
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.
H & N	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.
C	Carry/Link Flag. C = 1 if the operation produced a carry from the MSB of the operand or result.
↓	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 a "don't care".
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 of 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 >

## SYMBOLIC NOTATION

Instruction	D <sub>7</sub> S	Z		H		P/V	N	D <sub>0</sub> 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 OR s, XOR s	↕ ↕ ↕	↕ ↕ ↕	X X X	1 0 0	X X X	P P P	0 0 0	0 0 0	Logical Operations
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 INIR ; INDR ; OTIR ; OTDR	X X	↕ 1	X X	X X	X X	X X	1 1	• •	Block Input and Output. Z = 0 if B ≠ 0 otherwise Z = 0
LDI ; LDD LDIR ; LDDR	X X	X X	X X	0 0	X X	↕ 0	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	•	The content of the interrupt enable flip-flop (IFF) is copied into P/V flag.
BIT b, s	X	↕	X	1	X	X	0	•	The state of bit b of location is copied into the Z flag.

## PIN DESCRIPTIONS

**A<sub>0</sub>-A<sub>15</sub>. Address Bus** (Output, Active High, 3-state). A<sub>0</sub>-A<sub>15</sub> form a 16-bit address bus. The Address Bus provides the address for memory data bus exchanges (up to 64 K 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 wire-ORed 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<sub>0</sub>-D<sub>7</sub>. Data Bus** (Input/Output), active High, 3-state). D<sub>0</sub>-D<sub>7</sub> 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 non-maskable or a maskable interrupt (with the mask enabled) before operation can be resumed.

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 wire-ORed 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 refreshing dynamic memory properly.

**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.

## CPU TIMING

The Z80C 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.

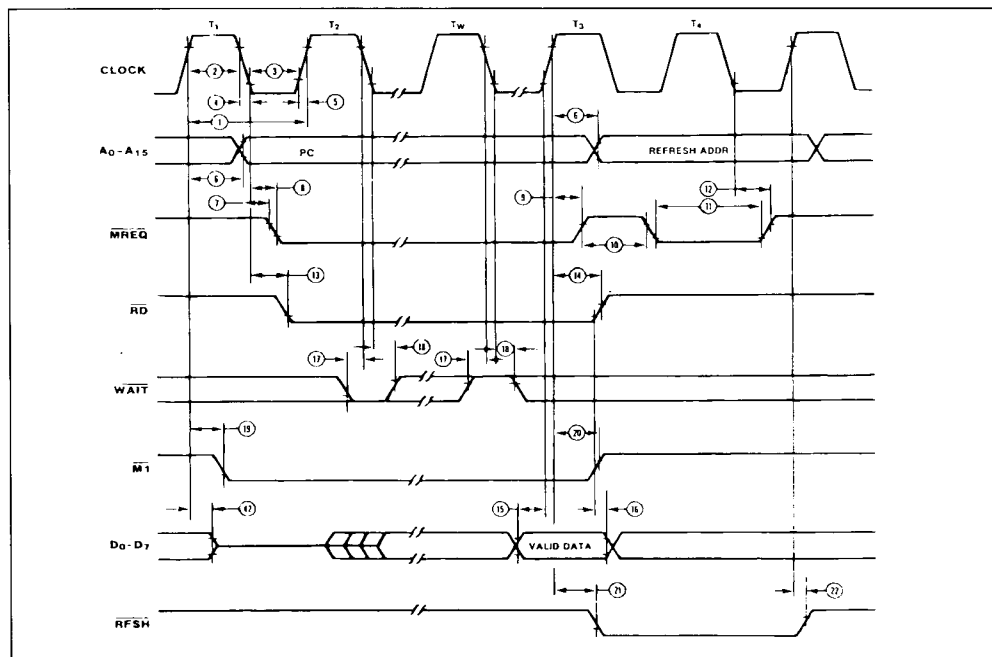
### INSTRUCTION OPCODE FETCH

The CPU places the contents of the Program

**Figure 5 : Instruction Opcode Fetch.**

Counter (PC) on the address bus at the start of the cycle (figure 5). Approximately one-half clock cycle later, MREQ goes active. When active, RD indicates that the memory data can be enabled onto the CPU data bus.

The CPU samples the  $\overline{\text{WAIT}}$  input with the falling edge of clock state T<sub>2</sub>. During clock states T<sub>3</sub> and T<sub>4</sub> of an 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.



**Note :** T<sub>w</sub>-Wait cycle added when necessary for slow ancillary devices.

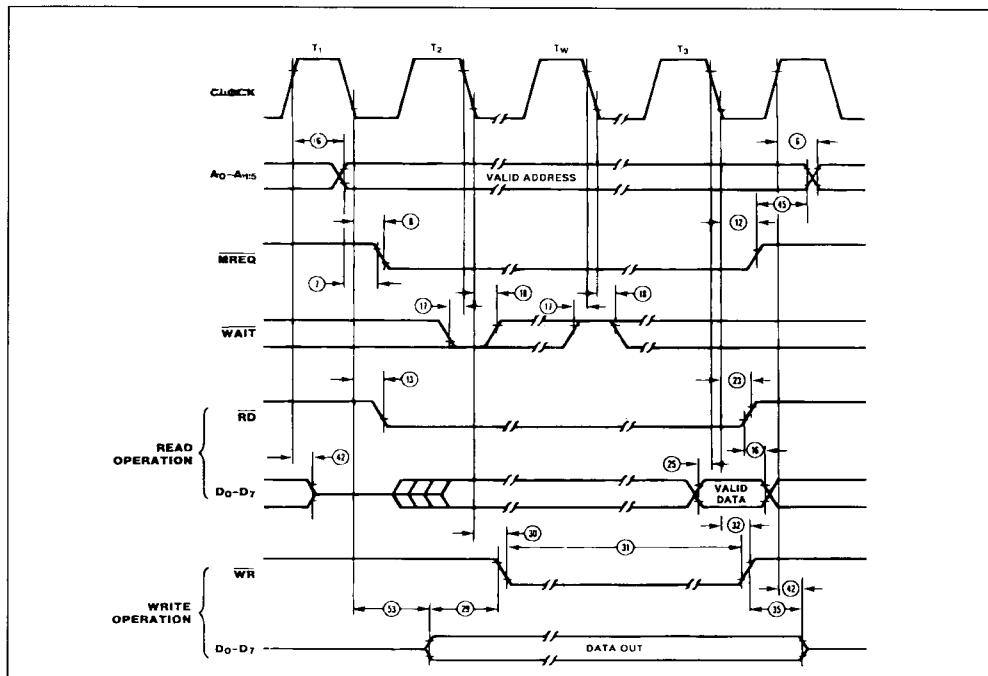
## MEMORY READ OR WRITE CYCLES

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  $\overline{\text{WR}}$  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.

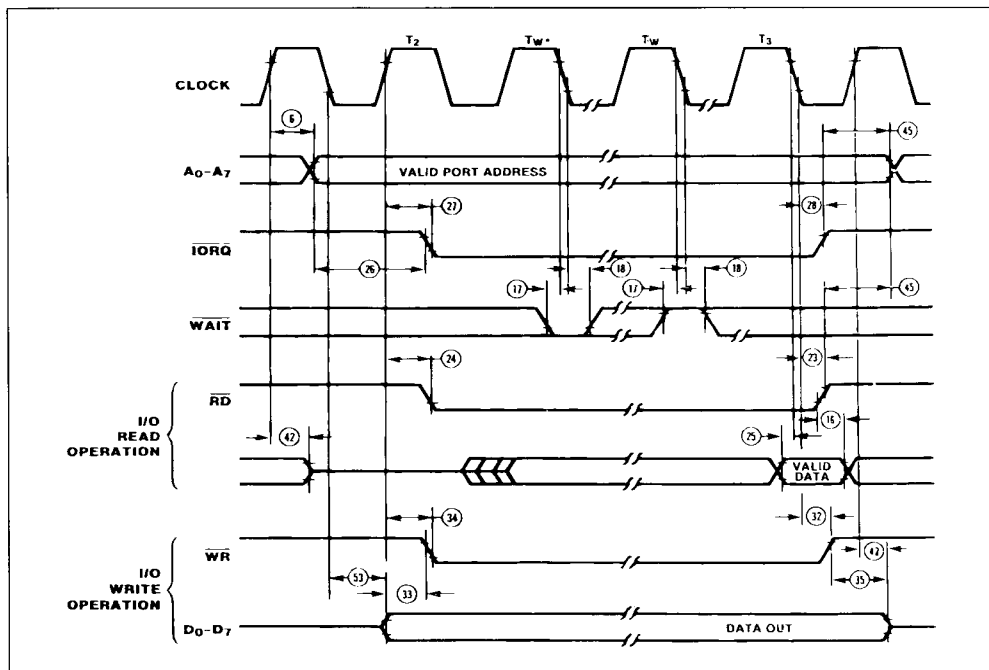


## INPUT OR OUTPUT CYCLES

Figure 7 shows the timing for an I/O read or I/O write operation. During I/O operations, the CPU automati-

cally inserts a single Wait state ( $T_w$ ). This extra Wait state allows sufficient time for an I/O port to decode the address from the port address lines.

**Figure 7 :** Input or Output Cycles.



Note :  $T_w^*$  = One Wait cycle automatically inserted by CPU.



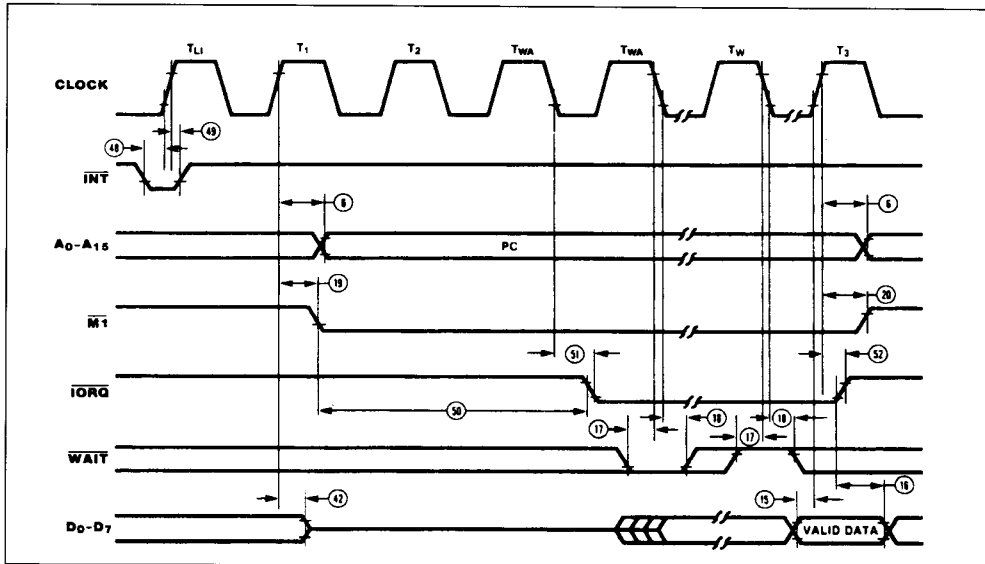
# INTERRUPT REQUEST/ACKNOWLEDGE CYCLE

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.

**Figure 8 :** Interrupt Request/Acknowledge Cycle.



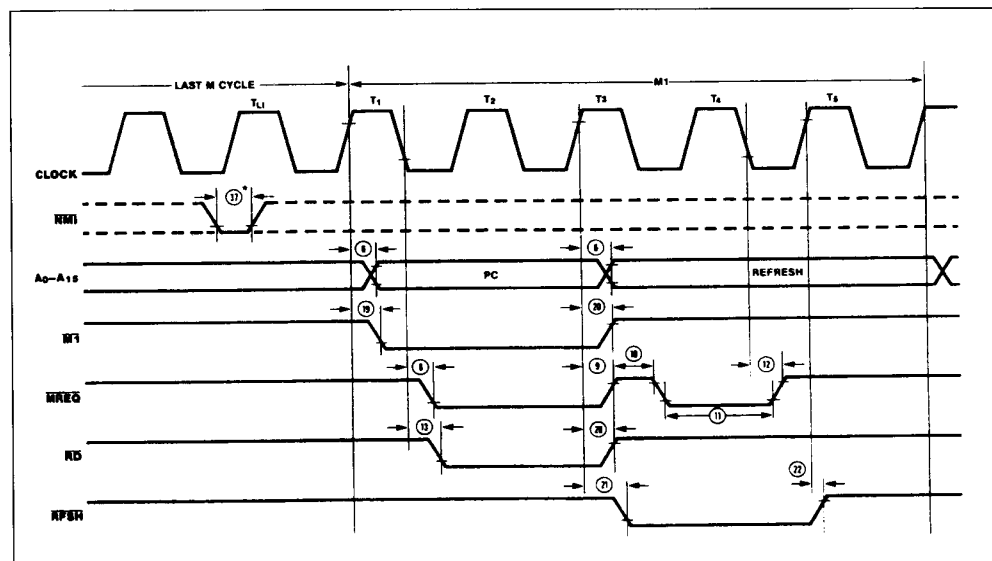
- Notes :
1.  $T_L$  = Last state of previous instruction.
  2. Two Wait cycles automatically inserted by CPU (\*).

## NON-MASKABLE INTERRUPT REQUEST CYCLE

NMI is sampled at the same time as the maskable interrupt input INT bus has higher priority and cannot be disabled under software control. The sub-

sequent timing is similar to that of a normal instruction fetch 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).

**Figure 9 : Non-maskable Interrupt Request Operation.**



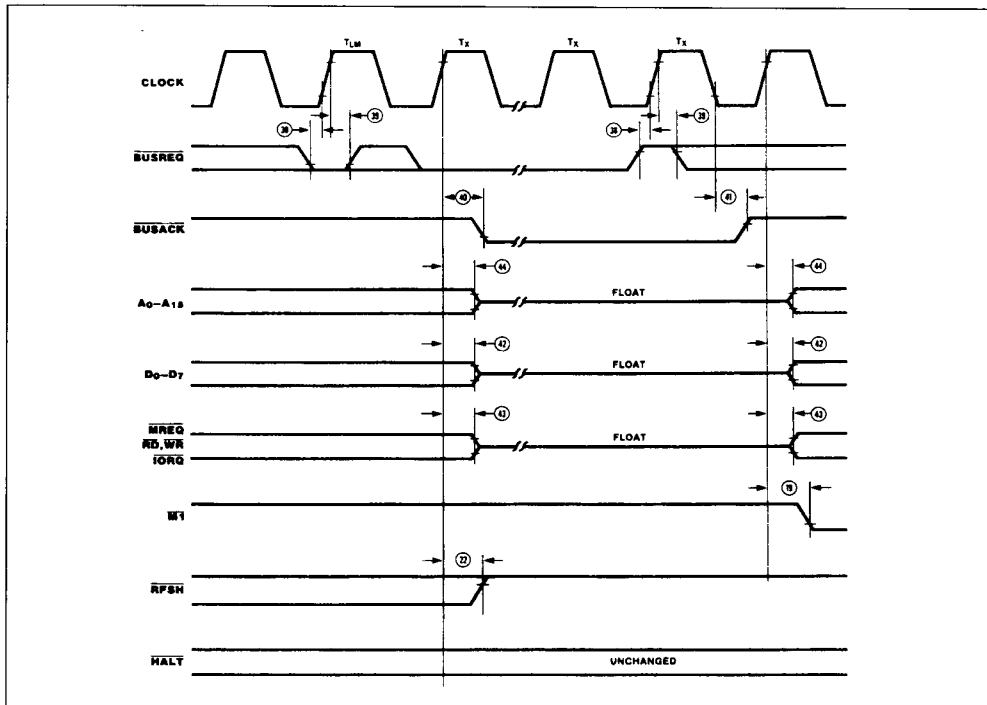
\* Although NMI is an asynchronous input, to guarantee its being recognized on the following machine cycle, NMI's falling edge must occur no later than the rising edge of the clock cycle preceding  $T_{LAST}$ .

## BUS REQUEST/ACKNOWLEDGE CYCLE

The CPU samples BUSREQ with the rising edge of the last clock period of any machine cycle (figure 10). If BUSREQ is active, the CPU sets its address, data, and MREQ, IORQ, RD, and 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.

**Figure 10 : Z-Bus Request/Acknowledge Cycle.**



- Notes :**
1.  $T_L$  = Last state of any M cycle.
  2.  $T_X$  = An arbitrary clock cycle used by requesting device.

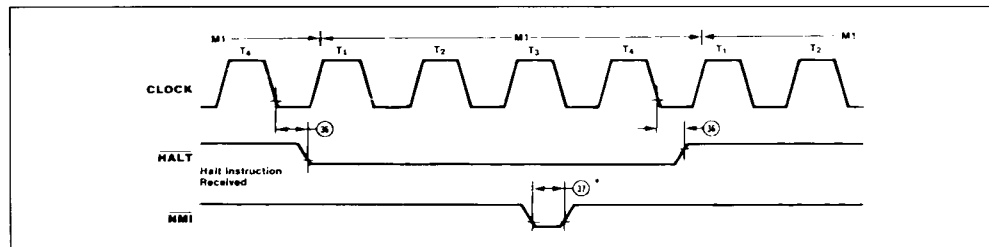
## HALT ACKNOWLEDGE CYCLE

When the CPU receives an Halt instruction, it executes NOP states until either an INT or NMI input is received. When in the Halt state, the HALT output is active and remains so until an interrupt is processed (figure 11).

## RESET CYCLE

RESET must be active for at least three clock cycles

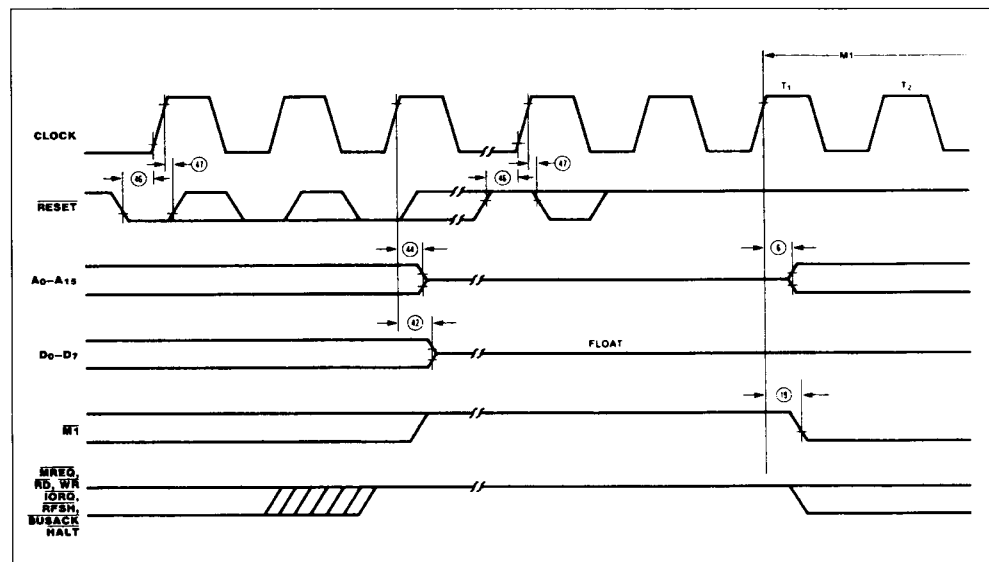
**Figure 11 :** Halt Acknowledge Cycle.



**Note :** INT will also force a Halt exit.

\* See note, Figure 9.

**Figure 12 :** Reset Cycle.



## POWER DOWN

When the CPU system clock is stopped at either a high or low level, the CPU stops its operation and maintains registers and control signals.

However  $I_{CC2}$  Stand-by Supply Current is guaranteed only when the supplied system clock is stopped at a low level during T4 state of the following machine cycle (actually that is M1 cycle and executes NOP instruction) next to OPcode fetch cycle of HALT instruction. The timing diagram when POWER DOWN function is implemented by HALT instruction is shown in figure 13.

This function can be easily realized when a clock generator controller is connected with the CPU.

## RELEASE FROM POWER DOWN STATE

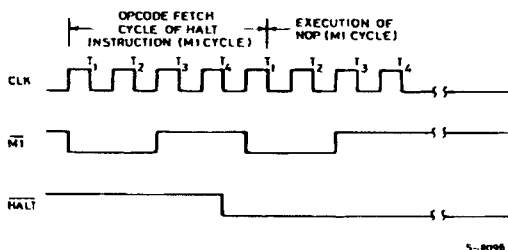
The system clock must be supplied to the CPU to release power down state.

When the system clock is supplied to the CPU CLK terminal, CPU restarts operation continuously from the state when power down function has been implemented.

Note the followings when release from power down state.

- (1) When external oscillator has been stopped to enter power down state, some warming-up time may be required to obtain precious and stable system clock for release from power down state.
- (2) When HALT instruction is executed to enter power down state, the CPU will enter HALT state. An interrupt signal (NMI or INT) or RESET signal must be generated after the system clock is supplied to release power down state. Otherwise the CPU is still in HALT state even if the system clock is supplied.

Figure 13 : Timing Diagram of Power Down Function by Halt Instruction.



## AC CHARACTERISTICS

N°	Symbol	Parameter	Z84C00A		Z84C00B		Z84C00H	
			Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)
1	T <sub>c</sub>	Clock Cycle Time	250	DC	165	DC	125	DC
2	T <sub>w</sub> Ch	Clock Pulse Width (high)	110	DC	65	DC	55	DC
3	T <sub>w</sub> Cl	Clock Pulse Width (low)	110	DC	65	DC	55	DC
4	T <sub>f</sub> C	Clock Fall Time		30		20		10
5	T <sub>r</sub> C	Clock Rise Time		30		20		10
6	T <sub>d</sub> Cr(A)	Clock ↑ to Address Valid Delay		110		90		80
7	T <sub>d</sub> A(MREQf)	Address Valid to MREQ ↓ Delay	65		35		20	
8	T <sub>d</sub> Cf(MREQf)	Clock ↓ to MREQ ↓ Delay		85		70		60
9	T <sub>d</sub> Cr(MREQr)	Clock ↑ to MREQ ↑ Delay		85		70		60
10	T <sub>w</sub> MREQh	MREQ Pulse Width (high)	110		65		45	
11	T <sub>w</sub> MREQl	MREQ Pulse Width (low)	220		135		100	
12	T <sub>d</sub> Cf(MREQr)	Clock ↓ to MREQ ↑ Delay		85		70		60
13	T <sub>d</sub> Cf(RDf)	Clock ↓ to RD ↓ Delay		95		80		70
14	T <sub>d</sub> Cr(RDr)	Clock ↑ to RD ↑ Delay		85		70		60
15	T <sub>s</sub> D(Cr)	Data Setup Time to Clock ↑	35		30		30	
16	T <sub>h</sub> D(RDr)	Data Hold Time to RD ↑	0		0		0	
17	T <sub>s</sub> WAIT(Cf)	WAIT Setup Time to Clock ↓	70		60		50	
18*	T <sub>h</sub> WAIT(Cf)	WAIT Hold Time after Clock ↓	10		10		10	
19	T <sub>d</sub> Cr(MIf)	Clock ↑ to MI ↓ Delay		100		80		70
20	T <sub>d</sub> Cr(MIr)	Clock ↑ to MI ↑ Delay		100		80		70
21	T <sub>d</sub> Cr(RFSHf)	Clock ↑ to RFSH ↓ Delay		130		110		95
22	T <sub>d</sub> Cr(RFSHr)	Clock ↑ to RFSH ↑ Delay		120		100		85
23	T <sub>d</sub> Cf(RDr)	Clock ↓ to RD ↑ Delay		85		70		60
24	T <sub>d</sub> Cr(RDf)	Clock ↑ to RD ↓ Delay		85		70		60
25	T <sub>s</sub> D(Cf)	Data Setup to Clock ↓ during M <sub>2</sub> , M <sub>3</sub> , M <sub>4</sub> or M <sub>5</sub> Cycles	50		40		30	
26	T <sub>d</sub> A(IORQf)	Address Stable Prior to IORQ ↓	180		110		75	
27	T <sub>d</sub> Cr(IOROf)	Clock ↑ to IORQ ↓ Delay		75		65		55
28	T <sub>d</sub> Cf(IORQr)	Clock ↓ to IORQ ↑ Delay		85		70		60
29	T <sub>d</sub> Cf(WRf)	Data Stable Prior to WR ↓	80		25		5	
30	T <sub>d</sub> Df(WRf)	Clock ↓ to WR ↓ Delay		80		70		60
31	T <sub>w</sub> WR	WR Pulse Width	220		135		100	
32	T <sub>d</sub> Cf(WRr)	Clock ↓ to WR ↑ Delay		80		70		60
33	T <sub>d</sub> D(WRf)	Data Stable Prior to WR ↓	- 10		- 55		- 55	
34	T <sub>d</sub> Cr(WRf)	Clock ↑ to WR ↓ Delay		65		60		55
35	T <sub>d</sub> WRr(D)	Data Stable from WR ↑	60		30		15	
36	T <sub>d</sub> Cf(HALT)	Clock ↓ to HALT ↑ or ↓		300		260		225

Note : \* Not compatible with NMOS Specifications.

## AC CHARACTERISTICS (continued)

N°	Symbol	Parameter	Z84C00A		Z84C00B		Z84C00H	
			Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)
37	TwNMI	NMI Pulse Width		80		70		60
38	TsBUSREQ(Cr)	BUSREQ Setup Time to Clock ↑	50		50		40	
39*	TcBUSUREQ(Cr)	BUSREQ Hold Time after Clock ↑	10		10		10	
40	TdCr(BUSACKf)	Clock ↑ to BUSACK ↓ Delay		100		90		80
41	TdCf(BUSACKr)	Clock ↓ to BUSACK ↑ Delay		100		90		80
42	TdCr(Tz)	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	10		10		10	
48	TsINTf(Cr)	INT to Clock ↑ Setup Time	80		70		55	
49*	ThINTr(Cr)	INT to Clock ↑ Hold Time	10		10		10	
50	TdMf(IORQf)	MI ↓ to IORQ ↓ Delay	565		365		270	
51	TdCf(IORQf)	Clock ↓ to IORQ ↓ Delay		85		70		60
52	TdCf(IORQr)	Clock ↑ to IORQ ↑ Delay		85		70		60
53	TdCf(D)	Clock ↓ to Data Valid Delay		150		130		115

Note : \* Not compatible with NMOS Specification.

## ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V <sub>CC</sub>	V <sub>CC</sub> Supply Voltage with Respect to V <sub>SS</sub>	- 0.5 to 7	V
V <sub>IN</sub>	Input Voltage	- 0.5 to V <sub>CC</sub> + 0.5	V
P <sub>D</sub>	Power Dissipation (T <sub>A</sub> = 85 °C)	250	mW
T <sub>SOLDER</sub>	Soldering Temperature (soldering time 10 sec)	260	°C
T <sub>STG</sub>	Storage Temperature	- 65 to 150	°C
T <sub>OPR</sub>	Operating Temperature	- 40 to 85	°C

## DC CHARACTERISTICS

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$V_{ILC}$	Clock Input Low Voltage		- 0.3	-	0.6	V
$V_{IHC}$	Clock Input High Voltage		$V_{CC} - 0.6$	-	$V_{CC} + 0.3$	V
$V_{IL}$	Input Low Voltage (except CLK)		- 0.5	-	0.8	V
$V_{IH}$	Input High Voltage (except CLK)		2.2	-	$V_{CC}$	V
$V_{OL}$	Output Low Voltage	$I_{OL} = 2.0 \text{ mA}$	-	-	0.4	V
$V_{OH1}$	Output High Voltage (1)	$I_{OH} = - 1.6 \text{ mA}$	2.4	-	-	V
$V_{OH2}$	Output High Voltage (2)	$I_{OH} = - 250 \mu\text{A}$	$V_{CC} - 0.8$	-	-	V
$I_{LI}$	Input Leakage Current	$V_{SS} \leq V_{IN} \leq V_{CC}$	-	-	10	$\mu\text{A}$
$I_{LO}$	3-State Output Leakage Current in Float	$V_{SS} + 0.4 \leq V_{OUT} \leq V_{CC}$	- 10	-	10	$\mu\text{A}$
$I_{CC1}$	Operating Supply Current 4 MHz 6 MHz 8 MHz	$V_{CC} = 5 \text{ V}, V_{IL} = 0.2 \text{ V}$	-	9	15	mA
		$V_{IH} = V_{CC} - 0.2 \text{ V}$	-	15	22	mA
			-	20	25	mA
$I_{CC2(1)}$	Stand-by Supply Current	$V_{CC} = 5 \text{ V}$ CLK = (1) $V_{IL} = V_{CC} - 0.2 \text{ V}$ $V_{IH} = 0.2 \text{ V}$	-	0.5	10	$\mu\text{A}$

**Note :** 1.  $I_{CC2}$  Stand-by Current is guaranteed only when the supplied clock is stopped at a low level during T4 state of the following machine cycle (M1) next to OPcode fetch cycle of HALT instruction.

## TEST CONDITIONS

$T_A = - 40^\circ\text{C}$  to  $+ 85^\circ\text{C}$

$V_{CC} = 5 \text{ V} \pm 10 \%$

$V_{SS} = 0 \text{ V}$

AC test conditions

- Inputs except CLK (clock) are driven at 2.4 V for a logic "1" and 0.4 V for a logic "0". Clock input is

driven at  $V_{CC} - 0.6 \text{ V}$  for a logic "1" and 0.6 V for a logic "0".

- Timing measurements are made at 2.2 V for a logic "1" and 0.8 V for a logic "0".

All AC parameters assume a load capacitance of 100 pF.

## ORDERING INFORMATION

Type	Package	Temp.	Clock	Description
<b>Z84C00AB6</b>	DIP-40 (plastic)	-40/+ 85°C	4 MHz	Z80C Central Processing Unit CMOS Version
<b>Z84C00AD6</b>	DIP-40 (ceramic)	-40/+ 85°C		
<b>Z84C00AD2</b>	DIP-40 (ceramic)	-55/+ 125°C		
<b>Z84C00AC6</b>	PLCC44 (plastic chip-carrier)	-40/+ 85°C		
<b>Z84C00BB6</b>	DIP-40 (plastic)	-40/+ 85°C	6 MHz	
<b>Z84C00BD6</b>	DIP-40 (ceramic)	-40/+ 85°C		
<b>Z84C00BD2</b>	DIP-40 (ceramic)	-55/+ 125°C		
<b>Z84C00BC6</b>	PLCC44 (plastic chip-carrier)	-40/+ 85°C		
<b>Z84C00HB6</b>	DIP-40 (plastic)	-40/+ 85°C	8 MHz	
<b>Z84C00HD6</b>	DIP-40 (ceramic)	-40/+ 85°C		
<b>Z84C00HC6</b>	PLCC44 (plastic chip-carrier)	-40/+ 85°C		