Z8400

Z80 CPU Central Process Unit

- The instruction set contains 158 instructions. The 78 instructions of the 8080A are included as a subset; 8080A and Z80* software compatibility is maintained.
- 8MHz, 6MHz, 4MHz and 2.5 MHz clocks for the Z80H, Z80B, Z80A and Z80 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 microcomputer industry.
- The Z80 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.

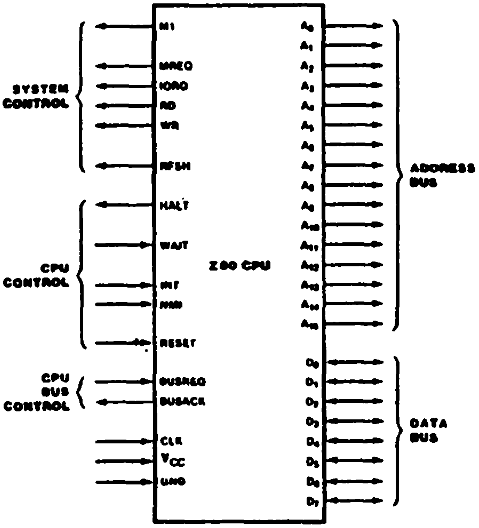

SYSTEM CONTROL
M1
MREQ
IORQ
RD
WR
RFSH
CPU CONTROL
HALT
WAIT
INT
NMI
RESET
CPU BUS CONTROL
BUSREQ
BUSACK
CLK
V_{CC}
GND
Z80 CPU
A_0
A_1
A_2
A_3
A_4
A_5
A_6
A_7
A_8
A_9
A_{10}
A_{11}
A_{12}
A_{13}
A_{14}
A_{15}
ADDRESS BUS
D_0
D_1
D_2
D_3
D_4
D_5
D_6
D_7
DATA BUS

Figure 1. Logic Functions

Z8400

General Description

The Z80, Z80A, Z80B and Z80H CPUs are 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 Z80 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 it is supported by an extensive family of peripheral controllers. The internal block diagram (Figure 3) shows the primary functions of the Z80 processors. Subsequent text provides more detail on the Z80 I/O controller family, registers, instruction set, interrupts and daisy chaining, and CPU timing.

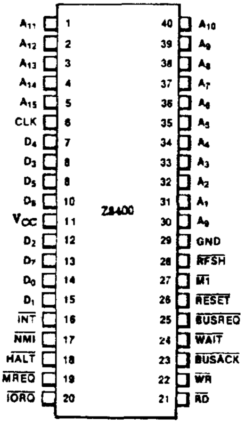
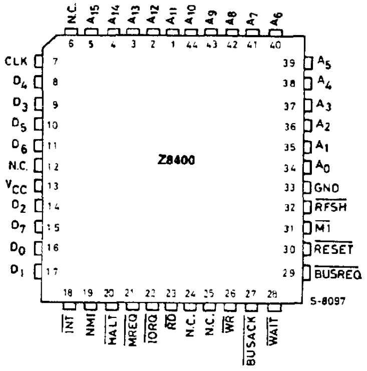

Z8400
A11 1
A12 2
A13 3
A14 4
A15 5
CLK 6
D4 7
D3 8
D5 9
D6 10
VCC 11
D2 12
D7 13
D0 14
D1 15
INT 16
NMI 17
HALT 18
MREQ 19
IORQ 20
40 A10
39 A9
38 A8
37 A7
36 A6
35 A5
34 A4
33 A3
32 A2
31 A1
30 A0
29 GND
28 RFSH
27 M1
26 RESET
25 BUSREQ
24 WAIT
23 BUSACK
22 WR
21 RD

Figure 2. Pin Configuration


Z8400
N.C. 6
A15 5
A14 4
A13 3
A12 2
A11 1
A10 44
A9 43
A8 42
A7 41
A6 40
CLK 7
D4 8
D3 9
D5 10
D6 11
N.C. 12
VCC 13
D2 14
D7 15
D0 16
D1 17
39 A5
38 A4
37 A3
36 A2
35 A1
34 A0
33 GND
32 RFSH
31 M1
30 RESET
29 BUSREQ
18 INT
19 NMI
20 HALT
21 MREQ
22 IORQ
23 RD
24 N.C.
25 N.C.
26 WR
27 BUSACK
28 WAIT
S-8097

N.C.=NO CONNECTION

Figure 2a. Chip Carrier Pin Configuration

General Description (Continued)

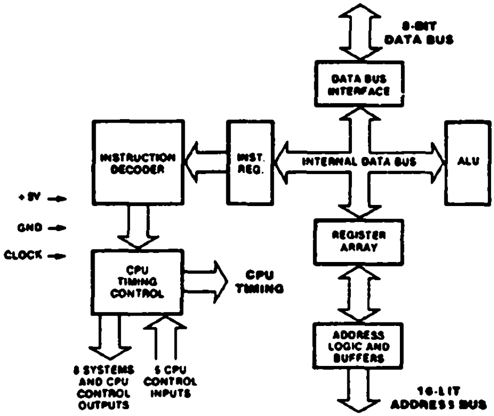

8-BIT DATA BUS
DATA BUS INTERFACE
INSTRUCTION DECODER
INST. REG.
INTERNAL DATA BUS
ALU
+5V
GND
CLOCK
CPU TIMING CONTROL
CPU TIMING
REGISTER ARRAY
ADDRESS LOGIC AND BUFFERS
8 SYSTEMS AND CPU CONTROL OUTPUTS
5 CPU CONTROL INPUTS
16-BIT ADDRESS BUS

Figure 3. CPU Block Diagram

Z80 Microprocessor Family

The Z80, Z80A, Z80B and Z80H microprocessor is the central element of a comprehensive microprocessor product family. This family works together in most applications with minimum requirements for additional logic, facilitating the design of efficent and cost-effective microcomputer-base systems.

Five components to provide extensive support for the Z80 microprocessor. These are:

- The CTC (Couter/Timer Circuit) features four programmable 8-bit counter/timers, each of which has an 8-bit prescaler. Each of the four channels may be configurated to operate in either counter or timer mode.
- The PIO (Parallel Input/Output) operates in both data-byte I/O transfer mode (with handshaking) and in bit mode (without handshaking). The PIO may be configured to interface with standard parallel periperal devices such as printers, tape punches, and keyboards.
- The DMA (Direct Memory Access) controller provides dual port data transfer operations and the ability to teminate data transfer as a result of a pattern match.
- The SIO (Serial Input/Output) controller offers two channels. It is capable of operating in a variety of programmable medes for both synchronous and asynchronous communication, including Bi-Synch and SDLC.
- The DART (Dual Asynchronous Receiver/Transmitter) device provides low cost asynchronous serial communication. It has two channels and a full modem control interface.

Z80 CPU Registers

Figure 4 shows three groups of registers within the Z80 CPU. The first group consists of duplicate sets of 8-bit registers: a principal set and an alternate set (designated by ' [prime], e.g., A'). Both sets consist of the Accumulator Register, the Flag Register, and six general-purpose registers. Transfer of data between these duplicate sets of registers is accomplished by use of "Exchange" instructions. The result is faster response to interrupts and easy, efficient implementation of such versatile programming techniques as background-foreground data processing. The second set of registers consists of six registers with assigned functions. These are the I (Interrupt Register), the R (Refresh Register), the IX and IY (Index Registers), the SP (Stack Pointer), and the PC (Program Counter). The third group consists of two interrupt status flip-flops, plus 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.

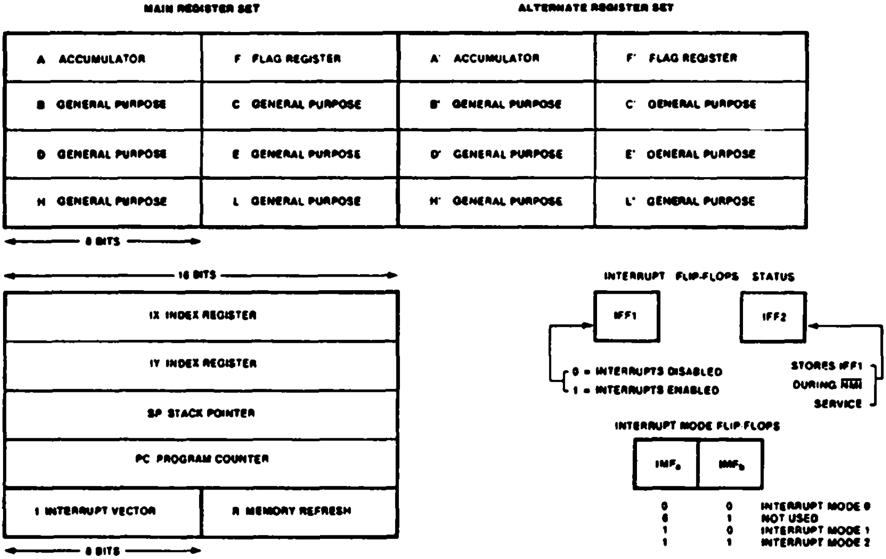

MAIN REGISTER SET
ALTERNATE REGISTER SET
A ACCUMULATOR
F FLAG REGISTER
A' ACCUMULATOR
F' FLAG REGISTER
B GENERAL PURPOSE
C GENERAL PURPOSE
B' GENERAL PURPOSE
C' GENERAL PURPOSE
D GENERAL PURPOSE
E GENERAL PURPOSE
D' GENERAL PURPOSE
E' GENERAL PURPOSE
H GENERAL PURPOSE
L GENERAL PURPOSE
H' GENERAL PURPOSE
L' GENERAL PURPOSE
8 BITS
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
IFF1
IFF2
0 = INTERRUPTS DISABLED
1 = INTERRUPTS ENABLED
STORES IFF1 DURING NMI SERVICE
INTERRUPT MODE FLIP-FLOPS
IMF$_a$
IMF$_b$
0 0 INTERRUPT MODE 0
0 1 NOT USED
1 0 INTERRUPT MODE 1
1 1 INTERRUPT MODE 2

Fig. 4. CPU Registers

CPU Registers (Continued)

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-trasparent 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_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).

Table 1. CPU Registers

Interrupts: General Operation

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

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

- Mode 0 — similar 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 $\overline{\text{INT}}$ 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. $\overline{\text{NMI}}$ is usually reserved for servicing only the highest priority type interrupts, such as that for orderly shut-down after power failure has been detected.
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 Z80 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 (MI) cycle in which $\overline{\text{IORQ}}$ becomes active rather than $\overline{\text{MREQ}}$, as in normal $\overline{\text{MI}}$ cycle. In addition, this special $\overline{\text{MI}}$ cycle is automatically extended by two $\overline{\text{WAIT}}$ 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 $\overline{\text{NMI}}$. 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 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 Priority (Daisy Chaining and Nested Interrupts). The interrupt priority of each peripheral device is determined by its physical location within a daisy-chain

Interrupts: General Operation (Continued)

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 succeding 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 Z80 CPU will nest (queue) any pending interrupts or interrupts received while a selected peripheral is being serviced.

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.*

Action	IFF$_2$	IFF$_2$	Comments
CPU Reset	0	0	Maskable interrupt $\overline{\text{INT}}$ disabled
DI instruction execution	0	0	Maskable interrupt $\overline{\text{INT}}$ disabled
EI instruction execution	1	1	Maskable interrupt $\overline{\text{INT}}$ enabled
LD A, I instruction execution	•	•	IFF$_2$→Parity flag
LD A, R instruction execution	•	•	IFF$_2$→Parity flag
Accept $\overline{\text{NMI}}$	0	IFF$_1$	IFF$_1$→IFF$_2$ (Maskable interrupt $\overline{\text{INT}}$disabled)
RETN instruction execution	IFF$_2$	•	IFF$_2$→IFF$_1$ at completion of an $\overline{\text{NMI}}$ service routine.

Table 2. State of Flip-Flops

Instruction Set

The Z80 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 Z80 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

8-Bit Load Group

Mnemonic	Symbolic Operation	S	Z		Flags H		P/V	N	C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD r, r'	r ← r'	•	•	X	•	X	•	•	•	01 r r'		1	1	4	r, r' Reg.
LD r, n	r ← n	•	•	X	•	X	•	•	•	00 r 110 – n –		2	2	7	000 B 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 01 r 110 – d –	DD	3	5	19	011 E 100 H 101 L
LD r, (IY + d)	r ← (IY + d)	•	•	X	•	X	•	•	•	11 111 101 01 r 110 – d –	FD	3	5	19	111 A
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 01 110 r – d –	DD	3	5	19	
LD (IY + d), r	(IY + d) ← r	•	•	X	•	X	•	•	•	11 111 101 01 110 r – d –	FD	3	5	19	
LD (HL), n	(HL) ← n	•	•	X	•	X	•	•	•	00 110 110 – n –	36	2	3	10	
LD (IX + d), n	(IX + d) ← n	•	•	X	•	X	•	•	•	11 011 101 00 110 110 – d – – n –	DD 36	4	5	19	
LD (IY + d), n	(IY + d) ← n	•	•	X	•	X	•	•	•	11 111 101 00 110 110 – d – – n –	FD 36	4	5	19	
LD A, (BC)	A ← (BC)	•	•	X	•	X	•	•	•	00 001 010	0A	1	2	7	
LD A, (DE)	A ← (DE)	•	•	X	•	X	•	•	•	00 011 010	1A	1	2	7	
LD A, (nn)	A ← (nn)	•	•	X	•	X	•	•	•	00 111 010 – n – – n –	3A	3	4	13	
LD (BC), A	(BC) ← A	•	•	X	•	X	•	•	•	00 000 010	02	1	2	7	
LD (DE), A	(DE) ← A	•	•	X	•	X	•	•	•	00 010 010	12	1	2	7	
LD (nn), A	(nn) ← A	•	•	X	•	X	•	•	•	00 110 010 – n – – n –	32	3	4	13	
LD A, I	A ← I	↕	↕	X	0	X	IFF	0	•	11 101 101 01 010 111	ED 57	2	2	9	
LD A, R	A ← R	↕	↕	X	0	X	IFF	0	•	11 101 101 01 011 111	ED 5F	2	2	9	
LD I, A	I ← A	•	•	X	•	X	•	•	•	11 101 101 01 000 111	ED 47	2	2	9	
LD R, A	R ← A	•	•	X	•	X	•	•	•	11 101 101 01 001 111	ED 4F	2	2	9	

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.

16-Bit Load Group

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD dd, nn	dd ← nn	•	•	X	•	X	•	•	•	00 dd0 001 ← n → ← n →		3	3	10	dd Pair 00 BC 01 DE 10 HL 11 SP
LD IX, nn	IX ← nn	•	•	X	•	X	•	•	•	11 011 101 00 100 001 ← n → ← n →	DD 21	4	4	14	
LD IY, nn	IY ← nn	•	•	X	•	X	•	•	•	11 111 101 00 100 001 ← n → ← n →	FD 21	4	4	14	
LD HL, (nn)	H ← (nn + 1) L ← (nn)	•	•	X	•	X	•	•	•	00 101 010 ← n → ← n →	2A	3	5	16	
LD dd, (nn)	dd_H ← (nn + 1) dd_L ← (nn)	•	•	X	•	X	•	•	•	11 101 101 01 dd1 011 ← n → ← n →	ED	4	6	20	
LD IX, (nn)	IX_H ← (nn + 1) IX_L ← (nn)	•	•	X	•	X	•	•	•	11 011 101 00 101 010 ← n → ← n →	DD 2A	4	6	20	
LD IY, (nn)	IY_H ← (nn + 1) IY_L ← (nn)	•	•	X	•	X	•	•	•	11 111 101 00 101 010 ← n → ← n →	FD 2A	4	6	20	
LD (nn), HL	(nn + 1) ← H (nn) ← L	•	•	X	•	X	•	•	•	00 100 010 ← n → ← n →	22	3	5	16	
LD (nn), dd	(nn + 1) ← dd_H (nn) ← dd_L	•	•	X	•	X	•	•	•	11 101 101 01 dd0 011 ← n → ← n →	ED	4	6	20	
LD (nn), IX	(nn + 1) ← IX_H (nn) ← IX_L	•	•	X	•	X	•	•	•	11 011 101 00 100 010 ← n → ← n →	DD 22	4	6	20	
LD (nn), IY	(nn + 1) ← IY_H (nn) ← IY_L	•	•	X	•	X	•	•	•	11 111 101 00 100 010 ← n → ← n →	FD 22	4	6	20	
LD SP, HL	SP ← HL	•	•	X	•	X	•	•	•	11 111 001	F9	1	1	6	
LD SP, IX	SP ← IX	•	•	X	•	X	•	•	•	11 011 101 11 111 001	DD F9	2	2	10	
LD SP, IY	SP ← IY	•	•	X	•	X	•	•	•	11 111 101 11 111 001	FD F9	2	2	10	
PUSH qq	(SP - 2) ← qq_L (SP - 1) ← qq_H SP ← SP - 2	•	•	X	•	X	•	•	•	11 qq0 101		1	3	11	qq Pair 00 BC 01 DE 10 HL 11 AF
PUSH IX	(SP - 2) ← IX_L (SP - 1) ← IX_H SP ← SP - 2	•	•	X	•	X	•	•	•	11 011 101 11 100 101	DD E5	2	4	15	
PUSH IY	(SP - 2) ← IY_L (SP - 1) ← IY_H SP ← SP - 2	•	•	X	•	X	•	•	•	11 111 101 11 100 101	FD E5	2	4	15	
POP qq	qq_H ← (SP + 1) qq_L ← (SP) SP ← SP + 2	•	•	X	•	X	•	•	•	11 qq0 001		1	3	10	
POP IX	IX_H ← (SP + 1) IX_L ← (SP) SP ← SP + 2	•	•	X	•	X	•	•	•	11 011 101 11 100 001	DD E1	2	4	14	
POP IY	IY_H ← (SP + 1) IY_L ← (SP) SP ← SP + 2	•	•	X	•	X	•	•	•	11 111 101 11 100 001	FD E1	2	4	14	

NOTES: dd is any of the register pairs BC, DE, HL, SP.
qq is any of the register pairs AF, BC, DE, HL.
$(PAIR)_H$, $(PAIR)_L$ 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

EX DE, HL	DE ↔ HL	•	•	X	•	X	•	•	•	11 101 011	EB	1	1	4	
EX AF, AF'	AF ↔ AF'	•	•	X	•	X	•	•	•	00 001 000	08	1	1	4	
EXX	BC ↔ BC' DE ↔ DE' HL ↔ HL'	•	•	X	•	X	•	•	•	11 011 001	D9	1	1	4	Register bank and auxiliary register bank exchange
EX (SP), HL	H ↔ (SP+1) L ↔ (SP)	•	•	X	•	X	•	•	•	11 100 011	E3	1	5	19	
EX (SP), IX	IX_H ↔ (SP+1) IX_L ↔ (SP)	•	•	X	•	X	•	•	•	11 011 101 11 100 011	DD E3	2	6	23	
EX (SP), IY	IY_H ↔ (SP+1) IY_L ↔ (SP)	•	•	X	•	X	•	•	•	11 111 101 11 100 011	FD E3	2	6	23	
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)

NOTE: 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.

8-Bit Arithmetic and Logical Group

Mnemonic	Symbolic Operation	Flags								Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
		S	Z		H		P/V	N	C						
ADD A, r	A ← A + r	↕	↕	X	↕	X	V	0	↕	10 $\boxed{000}$ r		1	1	4	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
ADD A, n	A ← A + n	↕	↕	X	↕	X	V	0	↕	11 $\boxed{000}$ 110 ← n →		2	2	7	
ADD A, (HL)	A ← A + (HL)	↕	↕	X	↕	X	V	0	↕	10 $\boxed{000}$ 110		1	2	7	
ADD A, (IX+d)	A ← A + (IX+d)	↕	↕	X	↕	X	V	0	↕	11 011 101 10 $\boxed{000}$ 110 ← d →	DD	3	5	19	
ADD A, (IY+d)	A ← A + (IY+d)	↕	↕	X	↕	X	V	0	↕	11 111 101 10 $\boxed{000}$ 110 ← d →	FD	3	5	19	
ADC A, s	A ← A+s+CY	↕	↕	X	↕	X	V	0	↕	$\boxed{001}$					s is any of r, n, (HL), (IX + d), (IY + d) as shown for ADD instruction. The indicated bits replace the $\boxed{000}$ in the ADD set above.
SUB s	A ← A−s	↕	↕	X	↕	X	V	1	↕	$\boxed{010}$					
SBC A, s	A ← A−s−CY	↕	↕	X	↕	X	V	1	↕	$\boxed{011}$					
AND s	A ← A ∧ s	↕	↕	X	1	X	P	0	0	$\boxed{100}$					
OR s	A ← A ∨ s	↕	↕	X	0	X	P	0	0	$\boxed{110}$					
XOR s	A ← A ⊕ s	↕	↕	X	0	X	P	0	0	$\boxed{101}$					
CP s	A−s	↕	↕	X	↕	X	V	1	↕	$\boxed{111}$					
INC r	r ← r + 1	↕	↕	X	↕	X	V	0	•	00 r $\boxed{100}$		1	1	4	
INC (HL)	(HL) ← (HL) + 1	↕	↕	X	↕	X	V	0	•	00 110 $\boxed{100}$		1	3	11	
INC (IX + d)	(IX + d) ← (IX + d) + 1	↕	↕	X	↕	X	V	0	•	11 011 101 00 110 $\boxed{100}$ ← d →	DD	3	6	23	
INC (IY + d)	(IY + d) ← (IY + d) + 1	↕	↕	X	↕	X	V	0	•	11 111 101 00 110 $\boxed{100}$ ← d →	FD	3	6	23	
DEC m	m ← m − 1	↕	↕	X	↕	X	V	1	•	$\boxed{101}$					m is any of r, (HL), (IX + d), (IY + d) as shown for INC. DEC same format and states as INC. Replace $\boxed{100}$ with $\boxed{101}$ in opcode.

General-Purpose Arithmetic and CPU Control Groups

Mnemonic	Symbolic Operation	S	Z		Flags H		P/V	N	C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
DAA	Converts 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 ← $\bar{A}$	•	•	X	1	X	•	1	•	00 101 111	2F	1	1	4	Complement accumulator (one's complement).
NEG	A ← 0 – A	↕	↕	X	↕	X	V	1	↕	11 101 101 01 000 100	ED 44	2	2	8	Negate acc. (two's complement).
CCF	CY ← $\overline{CY}$	•	•	X	X	X	•	0	↕	00 111 111	3F	1	1	4	Complement carry flag.
SCF	CY ← 1	•	•	X	0	X	•	0	1	00 110 111	37	1	1	4	Set carry flag.
NOP	No operation	•	•	X	•	X	•	•	•	00 000 000	00	1	1	4	
HALT	CPU halted	•	•	X	•	X	•	•	•	01 110 110	76	1	1	4	
DI *	IFF ← 0	•	•	X	•	X	•	•	•	11 110 011	F3	1	1	4	
EI *	IFF ← 1	•	•	X	•	X	•	•	•	11 111 011	FB	1	1	4	
IM 0	Set interrupt mode 0	•	•	X	•	X	•	•	•	11 101 101 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	
IM 2	Set interrupt mode 2	•	•	X	•	X	•	•	•	11 101 101 01 011 110	ED 5E	2	2	8	

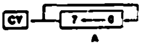
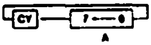
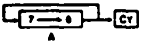
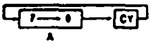
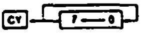
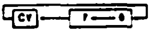
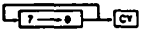
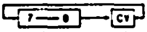
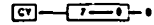
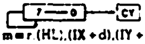
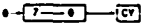
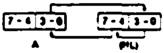
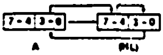
NOTES IFF indicates the interrupt enable flip-flop.
CY indicates the carry flip-flop.
* indicates interrupts are not sampled at the end of EI or DI.

16-Bit Arithmetic Group

ADD HL, ss	HL ← HL+ss	•	•	X	X	X	•	0	↕	00 ss1 001		1	3	11	ss Reg. 00 BC 01 DE 10 HL 11 SP
ADC HL, ss	HL ← HL+ss+CY	↕	↕	X	X	X	V	0	↕	11 101 101 01 ss1 010	ED	2	4	15	
SBC HL, ss	HL ← HL–ss–CY	↕	↕	X	X	X	V	1	↕	11 101 101 01 ss0 010	ED	2	4	15	
ADD IX, pp	IX ← IX + pp	•	•	X	X	X	•	0	↕	11 011 101 01 pp1 001	DD	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY, rr	IY ← IY + rr	•	•	X	X	X	•	0	↕	11 111 101 00 rr1 001	FD	2	4	15	rr Reg. 00 BC 01 DE 10 IY 11 SP
INC ss	ss ← ss + 1	•	•	X	•	X	•	•	•	00 ss0 011		1	1	6	
INC IX	IX ← IX + 1	•	•	X	•	X	•	•	•	11 011 101 00 100 011	DD 23	2	2	10	
INC IY	IY ← IY + 1	•	•	X	•	X	•	•	•	11 111 101 00 100 011	FD 23	2	2	10	
DEC ss	ss ← ss – 1	•	•	X	•	X	•	•	•	00 ss1 011		1	1	6	
DEC IX	IX ← IX – 1	•	•	X	•	X	•	•	•	11 011 101 00 101 011	DD 2B	2	2	10	
DEC IY	IY ← IY – 1	•	•	X	•	X	•	•	•	11 111 101 00 101 011	FD 2B	2	2	10	

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.

Rotate and Shift Group

Mnemonic	Symbolic Operation	S	Z		Flags H		P/V	N	C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLCA		•	•	X	0	X	•	0	‡	00 000 111	07	1	1	4	Rotate left circular accumulator.
RLA		•	•	X	0	X	•	0	‡	00 010 111	17	1	1	4	Rotate left accumulator.
RRCA		•	•	X	0	X	•	0	‡	00 001 111	0F	1	1	4	Rotate right circular accumulator.
RRA		•	•	X	0	X	•	0	‡	00 011 111	1F	1	1	4	Rotate right accumulator.
RLC r		‡	‡	X	0	X	P	0	‡	11 001 011 00 $\boxed{000}$ r	CB	2	2	8	Rotate left circular register r.
RLC (HL)		‡	‡	X	0	X	P	0	‡	11 001 011 00 $\boxed{000}$ 110	CB	2	4	15	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC (IX+d)	 r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡	11 011 101 11 001 011 – d – 00 $\boxed{000}$ 110	DD CB	4	6	23	
RLC (IY+d)		‡	‡	X	0	X	P	0	‡	11 111 101 11 001 011 – d – 00 $\boxed{000}$ 110	FD CB	4	6	23	
RL m	 m ≡ r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡	$\boxed{010}$					Instruction format and states are as shown for RLC's. To form new opcode replace $\boxed{000}$ or RLC's with shown code.
RRC m	 m ≡ r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡	$\boxed{001}$					
RR m	 m ≡ r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡	$\boxed{011}$					
SLA m	 m ≡ r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡	$\boxed{100}$					
SRA m	 m ≡ r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡	$\boxed{101}$					
SRL m	 m ≡ r,(HL),(IX+d),(IY+d)	‡	‡	X	0	X	P	0	‡	$\boxed{111}$					
RLD		‡	‡	X	0	X	P	0	•	11 101 101 01 101 111	ED 6F	2	5	18	Rotate digit left and right between the accumulator and location (HL).
RRD		‡	‡	X	0	X	P	0	•	11 101 101 01 100 111	ED 67	2	5	18	The content of the upper half of the accumulator is unaffected

Bit Set, Reset and Test Group

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

NOTES: The notation m_b indicates bit b (0 to 7) or location m.

Jump 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
JP nn	$PC \leftarrow nn$	•	•	X	•	X	•	•	•	11 000 011 – n – – n –	C3	3	3	10	
JP cc, nn	If condition cc is true $PC \leftarrow nn$, otherwise continue	•	•	X	•	X	•	•	•	11 cc 010 – n – – n –		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
JR e	$PC \leftarrow PC + e$	•	•	X	•	X	•	•	•	00 011 000 – e–2 –	18	2	3	12	
JR C, e	If C = 0, continue	•	•	X	•	X	•	•	•	00 111 000 – e–2 –	38	2	2	7	If condition not met.
	If C = 1, $PC \leftarrow PC + e$											2	3	12	If condition is met.
JR NC, e	If C = 1, continue	•	•	X	•	X	•	•	•	00 110 000 – e–2 –	30	2	2	7	If condition not met.
	If C = 0, $PC \leftarrow PC + e$											2	3	12	If condition is met.
JP Z, e	If Z = 0 continue	•	•	X	•	X	•	•	•	00 101 000 – e–2 –	28	2	2	7	If condition not met.
	If Z = 1, $PC \leftarrow PC + e$											2	3	12	If condition is met.
JR NZ, e	If Z = 1,	•	•	X	•	X	•	•	•	00 100 000 – e–2 –	20	2	2	7	If condition not met.

Jump Grup (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
	continue If Z = 0, PC ← PC + e													2	3	12	If condition is met.
JP (HL)	PC ← HL	•	•	X	•	X	•	•	•	11	101	001	E9	1	1	4	
JP (IX)	PC ← IX	•	•	X	•	X	•	•	•	11 11	011 101	101 001	DD E9	2	2	8	
JP (IY)	PC ← IY	•	•	X	•	X	•	•	•	11 11	111 101	101 001	FD E9	2	2	8	
DJNZ, e	B ← B − 1 If B = 0, continue	•	•	X	•	X	•	•	•	00 –	010 e − 2	000 –	10	2	2	8	If B = 0.
	If B ≠ 0, PC ← PC + e													2	3	13	If B ≠ 0.

NOTES: e represents the extension in the relative addressing mode.
e is a 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.

Call and Return 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
CALL nn	(SP − 1) ← PC_H (SP − 2) ← PC_L PC ← nn	•	•	X	•	X	•	•	•	11 – –	001 n n	101 – –	CD	3	5	17	
CALL cc, nn	If condition cc is false continue,	•	•	X	•	X	•	•	•	11 –	cc n	100 –		3	3	10	If cc is false.
	otherwise same as CALL nn									–	n	–		3	5	17	If cc is true.
RET	PC_L ← (SP) PC_H ← (SP + 1)	•	•	X	•	X	•	•	•	11	001	001	C9	1	3	10	
RET cc	If condition cc is false	•	•	X	•	X	•	•	•	11	cc	000		1	1	5	If cc is false.
	continue, otherwise same as RET													1	3	11	If cc is true.
RETI	Return from interrupt	•	•	X	•	X	•	•	•	11 01	101 001	101 101	ED 4D	2	4	14	
RETN[1]	Return from non-maskable interrupt	•	•	X	•	X	•	•	•	11 01	101 000	101 101	ED 45	2	4	14	
RST p	(SP − 1) ← PC_H (SP − 2) ← PC_L PC_H ← 0 PC_L ← p	•	•	X	•	X	•	•	•	11	t	111		1	3	11	

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

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

NOTE: [1] RETN loads IFF_2 → IFF_1

Input and Output Grup

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

NOTE ① If the result of B – 1 is zero the Z flag is set, otherwise it is reset.

Summary of Flag Operation

Instruction	D7 S	Z		H		P/V	N	D0 C	Comments
ADD A, s; ADC A, s	↕	↕	X	↕	X	V	0	↕	8-bit add or add with carry.
SUB s; SBC A, s; CP s; NEG	↕	↕	X	↕	X	V	1	↕	8-bit subtract, subtract with carry, compare and negate accumulator.
AND s	↕	↕	X	1	X	P	0	0	Logical operations.
OR s, XOR s	↕	↕	X	0	X	P	0	0	
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 = 0 if B ≠ 0 otherwise Z = 0.
INIR; INDR; OTIR; OTDR	X	1	X	X	X	X	1	•	
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	•	
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 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.

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.

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 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 >.

Pin Descriptions

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

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

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

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

$\overline{\mathrm{HALT}}$. *Halt State* (output, active Low). $\overline{\mathrm{HALT}}$ 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 resume. While halted, the CPU executes NOPs to maintain memory refresh.

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

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

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

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

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

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

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

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

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

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

CPU Timing

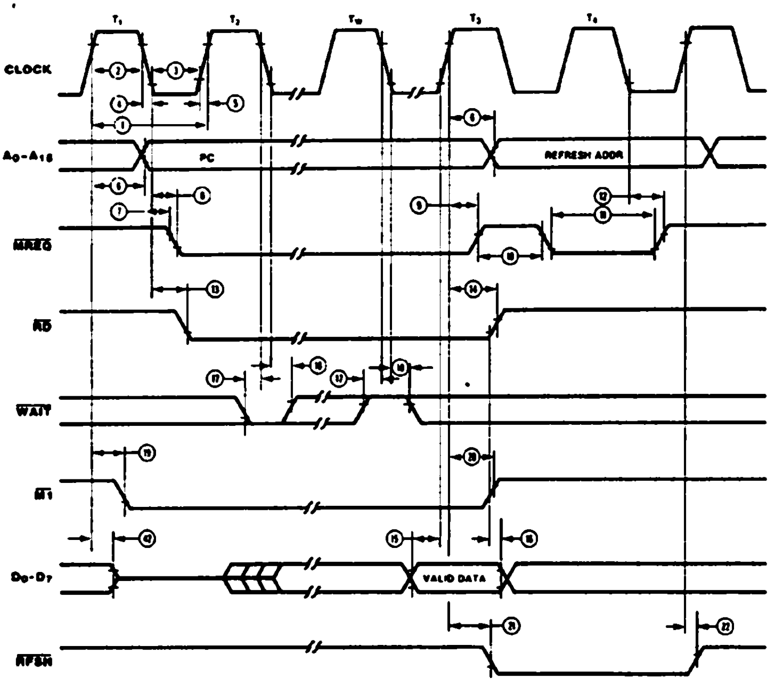
The Z80 CPU executes instructions by proceeding through a specific sequence of operations:

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

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

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

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



NOTE: T_w-Wait cycle added when necessary for slow ancillary devices.

Figure 5. Instruction Opcode Fetch

CPU Timing (Continued)

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

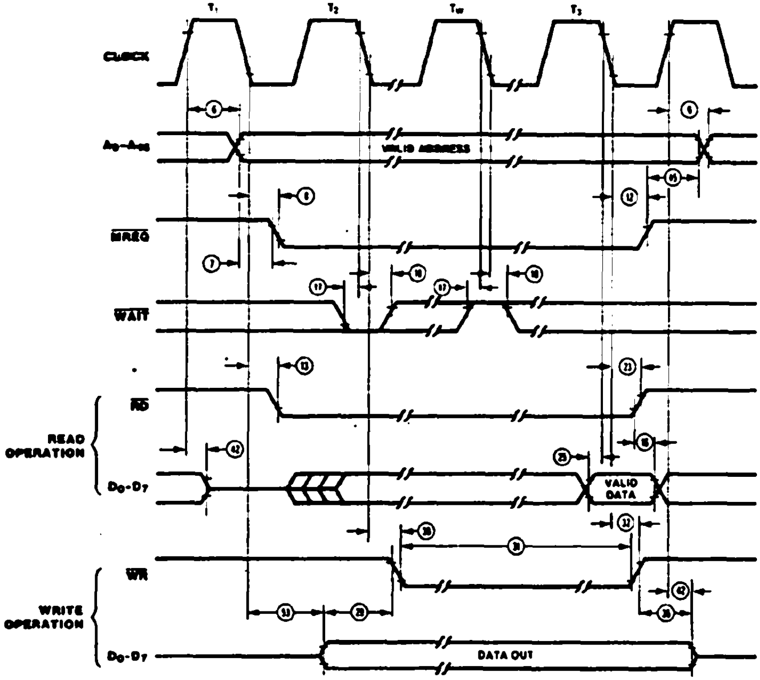
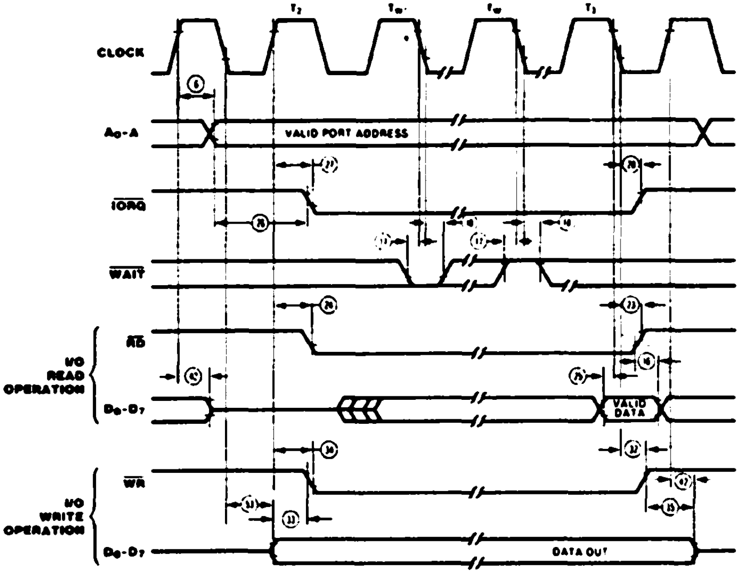

T1
T2
Tw
T3
CLOCK
A0-A15
VALID ADDRESS
MREQ
WAIT
RD
READ OPERATION
D0-D7
VALID DATA
WR
WRITE OPERATION
D0-D7
DATA OUT

Figure 6. Memory Read or Write Cycles

CPU Timing (Continued)

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_w). This extra Wait state allows sufficient time for an I/O port to decode the address from the port address lines.


T2
Tw*
Tw
T3
CLOCK
⑥
A0-A
VALID PORT ADDRESS
IORQ
WAIT
RD
I/O READ OPERATION
D0-D7
VALID DATA
WR
I/O WRITE OPERATION
D0-D7
DATA OUT

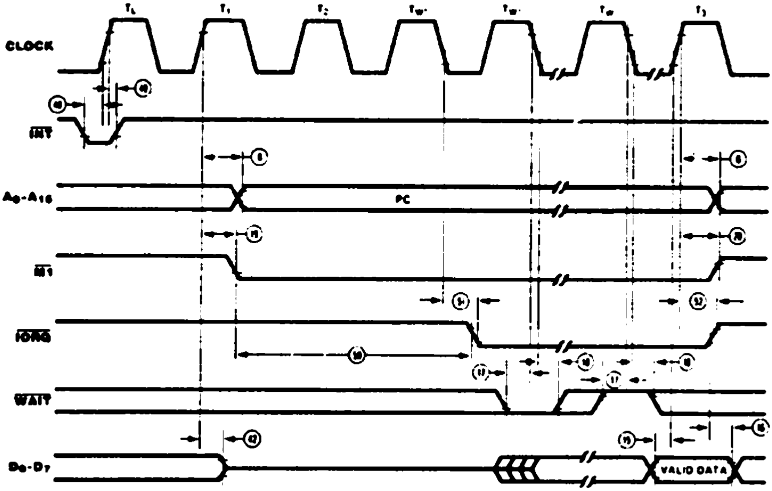
NOTE. T_w* = One Wait cycle automatically inserted by CPU.

Figure 7. Input or Output Cycles

CPU Timing (Continued)

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 $\overline{\text{MI}}$ cycle is generated.

During this $\overline{\text{MI}}$ cycle, $\overline{\text{IORQ}}$ becomes active (instead of $\overline{\text{MREQ}}$) 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.



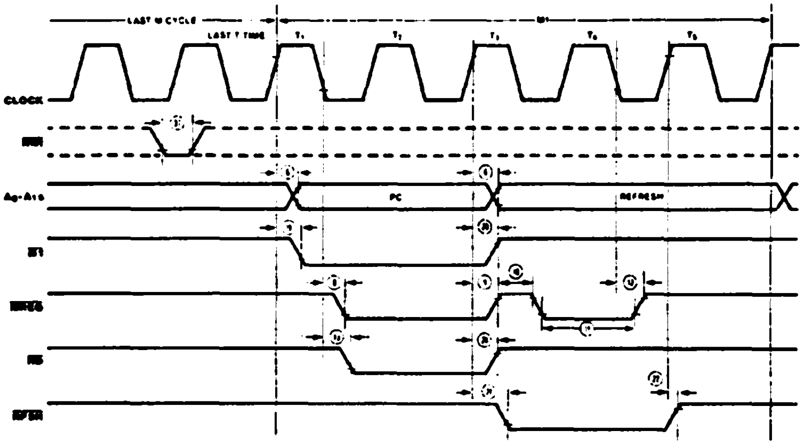
NOTE: 1) T_L = Last state of previous instruction.

2) Two Wait cycles automatically inserted by CPU(*).

Figure 8. Interrupt Request/Acknowledge Cycle

CPU Timing (Continued)

Non-Maskable Interrupt Request Cycle. $\overline{\text{NMI}}$ is sampled at the same time as the maskable interrupt input $\overline{\text{INT}}$ but has higher priority and cannot be disabled under software control. The subsequent 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 $\overline{\text{NMI}}$ service routine located at address 0066H (Figure 9).

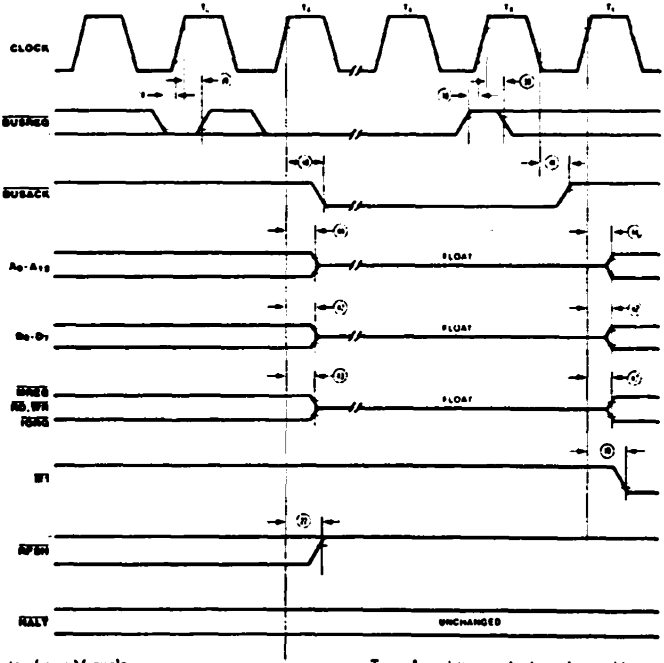


*Although $\overline{\text{NMI}}$ is an asynchronous input, to guarantee its being recognized on the following machine cycle, $\overline{\text{NMI}}$'s falling edge must occur no later than the rising edge of the clock cycle preceding T_{LAST}.

Figure 9. Non-Maskable Interrupt Request Operation

CPU Timing (Continued)

Bus Request/Acknowledge Cycle. The CPU samples $\overline{\text{BUSREQ}}$ with the rising edge of the last clock period of any machine cycle (Figure 10). If $\overline{\text{BUSREQ}}$ is active, the CPU sets its address, data, and $\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$ 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.



CLOCK
BUSREQ
BUSACK
A0-A15
FLOAT
D0-D7
FLOAT
MREQ
RD, WR
IORQ
FLOAT
M1
RFSH
HALT
UNCHANGED

NOTE: T_L = Last state of any M cycle.

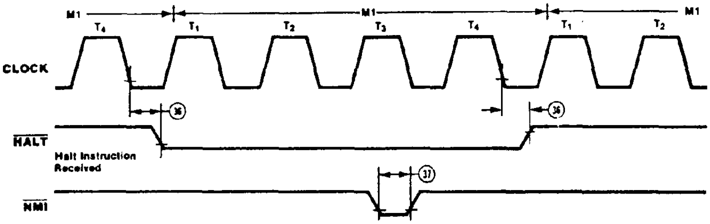
T_X = An arbitrary clock cycle used by requesting device.

Figure 10. Z-Bus Request/Acknowledge Cycle

CPU Timing (Continued)

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

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


M1
T4
T1
T2
M1
T3
T4
T1
M1
T2
CLOCK
㊱
㊱
HALT
Halt Instruction Received
㊲
NMI

NOTE: $\overline{\text{INT}}$ will also force a Halt exit.

*See note, Figure 9.

Figure 11. Halt Acknowledge Cycle

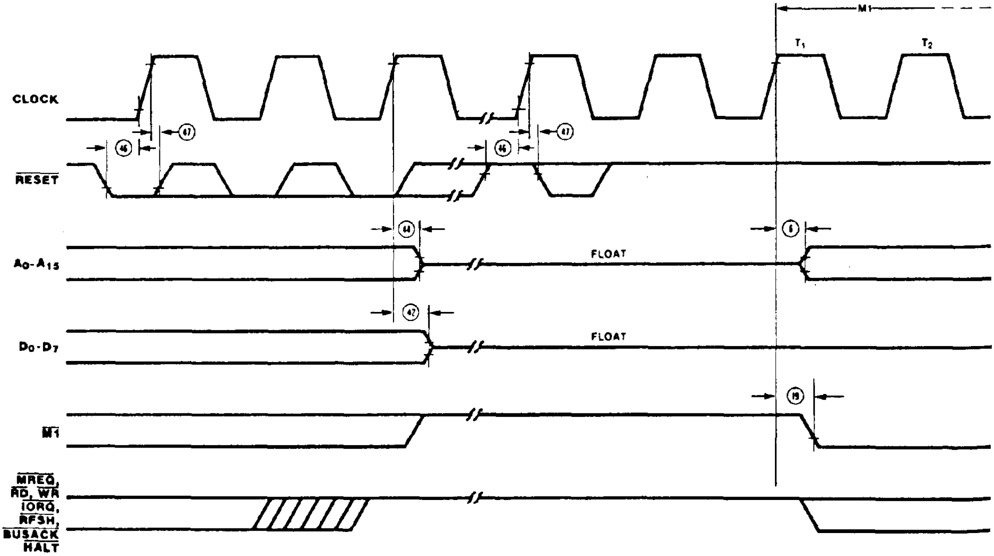

M1
T1
T2
CLOCK
㊼
㊻
㊼
㊻
RESET
㊹
⑥
A0-A15
FLOAT
㊷
D0-D7
FLOAT
⑲
M1
MREQ, RD, WR IORQ, RFSH, BUSACK HALT

Figure 12. Reset Cycle

AC Characteristics

Number	Symbol	Parameter	Z8400 Min (ns)	Z8400 Max (ns)	Z8400A Min (ns)	Z8400A Max (ns)	Z8400B Min (ns)	Z8400B Max (ns)	Z8400H Min (ns)	Z8400H Max (ns)
1	TcC	Clock Cycle Time	400*		250*		165*		125*	
2	TwCh	Clock Pulse Width (High)	180*	—	110*	—	65*	—	55*	
3	TwCl	Clock Pulse Width (Low)	180	2000	110	2000	65	2000	55	2000
4	TfC	Clock Fall Time	—	30	—	30	—	20	—	10
5	TrC	Clock Rise Time		30		30		20		10
6	TdCr(A)	Clock ↑ to Address Valid Delay	—	145	—	110	—	90	—	80
7	TdA(MREQf)	Address Valid to $\overline{MREQ}$ ↓ Delay	125*	—	65*	—	35*	—	20*	—
8	TdCf(MREQf)	Clock ↓ to $\overline{MREQ}$ ↓ Delay	—	100	—	85	—	70	—	60
9	TdCr(MREQr)	Clock ↑ to $\overline{MREQ}$ ↑ Delay	—	100	—	85	—	70	—	60
10	TwMREQh	$\overline{MREQ}$ Pulse Width (High)	170*		110*		65*		45*	
11	TwMREQl	$\overline{MREQ}$ Pulse Width (Low)	360*	—	220*	—	135*	—	100*	—
12	TdCf(MREQr)	Clock ↓ to $\overline{MREQ}$ ↑ Delay	—	100	—	85	—	70	—	60
13	TdCf(RDf)	Clock ↓ to $\overline{RD}$ ↓ Delay	—	130	—	95	—	80	—	70
14	TdCr(RDr)	Clock ↑ to $\overline{RD}$ ↑ Delay	—	100	—	85	—	70	—	60
15	TsD(Cr)	Data Setup Time to Clock ↑	50		35		30		30	
16	ThD(RDr)	Data Hold Time to $\overline{RD}$ ↑	—	0	—	0	—	0	—	0
17	TsWAIT(Cf)	$\overline{WAIT}$ Setup Time to Clock ↓	70	—	70	—	60	—	50	—
18	ThWAIT(Cf)	$\overline{WAIT}$ Hold Time after Clock ↓	—	0	—	0	—	0	—	0
19	TdCr(M1f)	Clock ↑ to $\overline{M1}$ ↓ Delay	—	130	—	100	—	80	—	70
20	TdCr(M1r)	Clock ↑ to $\overline{M1}$ ↑ Delay		130		100		80		70
21	TdCr(RFSHf)	Clock ↑ to $\overline{RFSH}$ ↓ Dealy	—	180	—	130	—	110	—	95
22	TdCr(RFSHr)	Clock ↑ to $\overline{RFSH}$ ↑ Delay	—	150	—	120	—	100	—	85
23	TdCf(RDr)	Clock ↓ to $\overline{RD}$ ↑ Delay	—	110	—	85	—	70	—	60
24	TdCr(RDf)	Clock ↑ to $\overline{RD}$ ↓ Dealy	—	110	—	85	—	70	—	60
25	TsD(Cf)	Data Setup to Clock ↓ during M_2, M_3, M_4 or M_5 Cycles	60		50		40		30	
26	TdA(IORQf)	Address Stable prior to $\overline{IORQ}$ ↓	320*	—	180*	—	110*	—	75*	—
27	TdCr(IORQf)	Clock ↑ to $\overline{IORQ}$ ↓ Delay	—	90	—	75	—	65	—	55
28	TdCf(IORQr)	Clock ↓ to $\overline{IORQ}$ ↑ Delay	—	110	—	85	—	70	—	60
29	TdCf(WRf)	Data Stable prior to $\overline{WR}$ ↓	190*	—	80*	—	25*	—	5*	—
30	TdDf(WRf)	Clock ↓ to $\overline{WR}$ ↓ Delay		90		80		70		60
31	TwWR	$\overline{WR}$ Pulse Width	360*	—	220*	—	135*	—	100*	—
32	TdCf(WRr)	Clock ↓ to $\overline{WR}$ ↑ Delay	—	100	—	80	—	70	—	60
33	TdD(WRf)	Data Stable prior to $\overline{WR}$ ↓	20*	—	-10*	—	-55*	—	55*	—
34	TdCr(WRf)	Clock ↑ to $\overline{WR}$ ↓ Delay	—	80	—	65	—	60	—	55
35	TdWRr(D)	Data Stable from $\overline{WR}$ ↑	120*	—	60*	—	30*	—	15*	—

* For clock periods other than the minimums shown in the table, calculate parameters using the expressions in the table on the following page.

All timings are preliminary and subject to change.

AC Characteristics (Continued)

			Z8400		Z8400A		Z8400B		Z8400H	
Number	Symbol	Parameter	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)
36	TdCf(HALT)	Clock ↓ to $\overline{\text{HALT}}$ ↑ or ↓	—	300	—	300	—	260	—	225
37	TwNMI	$\overline{\text{NMI}}$ Pulse Width	80	—	80	—	70	—	60*	—
38	TsBUSREQ(Cr)	$\overline{\text{BUSREQ}}$ Setup Time to Clock ↑	80	—	50	—	50	—	40	—
39	TcBUSUREQ(Cr)	$\overline{\text{BUSREQ}}$ Hold Time after Clock ↑	0	—	0	—	0	—	0	—
40	TdCr(BUSACKf)	Clock ↑ to $\overline{\text{BUSACK}}$ ↓ Delay		120		100		90		80
41	TdCf(BUSACKr)	Clock ↓ to $\overline{\text{BUSACK}}$ ↑ Delay	—	110	—	100	—	90	—	80
42	TdCr(Tz)	Clock ↑ to Data Float Delay	—	90	—	90	—	80	—	70
43	TdCr(CTz)	Clock ↑ to Control Outputs Float Delay ($\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$)	—	110	—	80	—	70	—	60
44	TdCr(Az)	Clock ↑ to Address Float Delay	—	110	—	90	—	80	—	70
45	TdCTr(A)	$\overline{\text{MREQ}}$ ↑, $\overline{\text{IORQ}}$ ↑, $\overline{\text{RD}}$ ↑, and $\overline{\text{WR}}$ ↑ to Address Hold Time	160*		80*		35*		20*	
46	TsRESET(Cr)	$\overline{\text{RESET}}$ to Clock ↑ Setup Time	90	—	60	—	60	—	45	—
47	ThRESET(Cr)	$\overline{\text{RESET}}$ to Clock ↑ Hold Time	—	0	—	0	—	0	—	0
48	TsINTf(Cr)	$\overline{\text{INT}}$ to Clock ↑ Setup Time	80	—	80	—	70	—	55	—
49	ThINTr(Cr)	$\overline{\text{INT}}$ to Clock ↑ Hold Time	—	0	—	0	—	0	—	0
50	TdMlf(IORQf)	$\overline{\text{MI}}$ ↓ to $\overline{\text{IORQ}}$ ↓ Delay	920*		565*		365*		270*	
51	TdCf(IORQf)	Clock ↓ to $\overline{\text{IORQ}}$ ↓ Delay	—	110	—	85	—	70	—	60
52	TdCf(IORQr)	Clock ↑ to $\overline{\text{IORQ}}$ ↑ Delay	—	100	—	85	—	70	—	60
53	TdCf(D)	Clock ↓ to Data Valid Delay	—	230	—	150	—	130	—	115

* For clock periods other than the minimums shown in the table, calculate parameters using the expressions in the table on the following page.

All timings are preliminary and subject to change.

Footnotes to AC Characteristics

Number	Symbol	Z8400	Z8400A	Z8400B
1	TcC	TwCh + TwCl + TrC + TfC	TwCh + TwCl + TrC + TfC	TwCh + TwCl + TrC + TfC
2	TwCh	Although static by design, TwCh of greater than 200 μs is not guaranteed	Although static by design, TwCh of greater than 200 μs is not guaranteed	Although static by design, TwCh of greater than 200 μs is not guaranteed
7	TdA(MREQf)	TwCh + TfC − 75	TwCh + TfC − 65	TwCh + TfC − 50
10	TwMREQh	TwCh + TfC − 30	TwCh + TfC − 20	TwCh + TfC − 20
11	TwMREQl	TcC − 40	TcC − 30	TcC − 30
26	TdA(IORQf)	TcC − 80	TcC − 70	TcC − 55
29	TdD(WRf)	TcC − 210	TcC − 170	TcC − 140
31	TwWR	TcC − 40	TcC − 30	TcC − 30
33	TdD(WRf)	TwCl + TrC − 180	TwCl + TrC − 140	TwCl + TrC − 140
35	TdWRr(D)	TwCl + TrC − 80	TwCl + TrC − 70	TwCl + TrC − 55
45	TdCTr(A)	TwCl + TrC − 40	TwCl + TrC − 50	TwCl + TrC − 50
50	TdMI1f(IORQf)	2TcC + TwCh + TfC − 80	2TcC + TwCh + TfC − 65	2TcC + TwCh + TfC − 50

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
$V_{OH} = 2.0$ V
$V_{OL} = 0.8$ V
FLOAT = ±0.5 V

Absolute Maximum Ratings

Storage Temperature ... −65°C to +150°C
Temperature under Bias ..... Specified operating range
Voltages on all inputs and outputs with respect to GND ... −0.3 V to +7.0 V
Power Dissipation .... 1.5 W

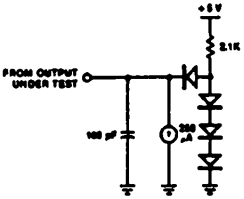
Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating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

Standard Test Conditions

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

- 0°C to +70°C, $+4.75\text{ V} \le V_{CC} \le +5.25\text{ V}$
- −40°C to +85°C, $+4.75\text{ V} \le V_{CC} \le +5.25\text{ V}$
- −55°C to +125°C, $+4.75\text{ V} \le V_{CC} \le +5.25\text{ V}$

All ac parameters assume a load capacitance of 50 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.



DC Characteristics

Symbol	Parameter	Min.	Max	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3	0.45	V	
V_{IHC}	Clock Input High Voltage	$V_{CC}-0.6$	$V_{CC}+0.3$	V	
V_{IL}	Input Low Voltage	-0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage		0.4	V	$I_{OL} = 1.8$ mA
V_{OH}	Output High Voltage	2.4		V	$I_{OH} = -250$ μA
I_{CC}	Power Supply Current				
	Z80		150[1]	mA	
	Z80A		200[2]	mA	
	Z80B		200	mA	
	Z80H		200	mA	
I_{LI}	Input Leakage Current	—	10	μA	$V_{IN} = 0$ to V_{CC}
I_{LO}	3-State Output Leakage Current in Float	-10	10[3]	μA	$V_{OUT} = 0.4$ to V_{CC}

1. For military grade parts, I_{CC} is 200 mA.
2. Typical rate for Z8400A is 90 mA.
3. A_{15}-A_0, D_7-D_0, $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$.

Capacitance

Symbol	Parameter	Min.	Max	Unit	Note
C_{CLOCK}	Clock Capacitance		35	pF	Unmeasured pins returned to ground
C_{IN}	Input Capacitance		5	pF	
C_{OUT}	Output Capacitance		10	pF	

$T_A = 25°C$, $f = 1$ MHz

Ordering Information

Type	Package	Temp.	Clock	Description
Z8400 B1	Plastic	0/+70°C	2.5 MHz	Z80 Central Processing Unit
Z8400 B6	Plastic	−40/+85°C		
Z8400 F1	Frit Seal	0/+70°C		
Z8400 F6	Frit Seal	−40/+85°C		
Z8400 D1	Ceramic	0/+70°C		
Z8400 D6	Ceramic	−40/+85°C		
Z8400 D2	Ceramic	−55/+125°C		
Z8400 C1	Plastic Chip-Carrier	0/+70°C		
Z8400 C6	Plastic Chip-Carrier	−40/+85°C		
Z8400 K1	Ceramic Chip-Carrier	0/+70°C		
Z8400 K6	Ceramic Chip-Carrier	−40/+85°C		
Z8400 K2	Ceramic Chip-Carrier	−55/+125°C		
Z8400A B1	Plastic	0/+70°C	4.0 MHz	
Z8400A B6	Plastic	−40/+85°C		
Z8400A F1	Frit Seal	0/+70°C		
Z8400A F6	Frit Seal	−40/+85°C		
Z8400A D1	Ceramic	0/+70°C		
Z8400A D6	Ceramic	−40/+85°C		
Z8400A D2	Ceramic	−55/+125°C		
Z8400A C1	Plastic Chip-Carrier	0/+70°C		
Z8400A C6	Plastic Chip-Carrier	−40/+85°C		
Z8400A K1	Ceramic Chip-Carrier	0/+70°C		
Z8400A K6	Ceramic Chip-Carrier	−40/+85°C		
Z8400A K2	Ceramic Chip-Carrier	−55/+125°C		
Z8400B B1	Plastic	0/+70°C	6.0 MHz	
Z8400B B6	Plastic	−40/+85°C		
Z8400B F1	Frit Seal	0/+70°C		
Z8400B F6	Frit Seal	−40/+85°C		
Z8400B D1	Ceramic	0/+70°C		
Z8400B D6	Ceramic	−40/+85°C		
Z8400B D2	Ceramic	−55/+125°C		
Z8400B C1	Plastic Chip-Carrier	0/+70°C		
Z8400B C6	Plastic Chip-Carrier	−40/+85°C		
Z8400B K1	Ceramic Chip-Carrier	0/+70°C		
Z8400B K6	Ceramic Chip-Carrier	−40/+85°C		
Z8400B K2	Ceramic Chip-Carrier	−55/+125°C		
Z8400H B1	Plastic	0/+70°C	8.0 MHz	
Z8400H B6	Plastic	−40/+85°C		
Z8400H F1	Frit Seal	0/+70°C		
Z8400H F6	Frit Seal	−40/+85°C		
Z8400H D1	Ceramic	0/+70°C		
Z8400H D6	Ceramic	−40/+85°C		
Z8400H C1	Plastic Chip-Carrier	0/+70°C		
Z8400H C6	Plastic Chip-Carrier	−40/+85°C		
Z8400H K1	Ceramic Chip-Carrier	0/+70°C		
Z8400H K6	Ceramic Chip-Carrier	−40/+85°C		