



## 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.

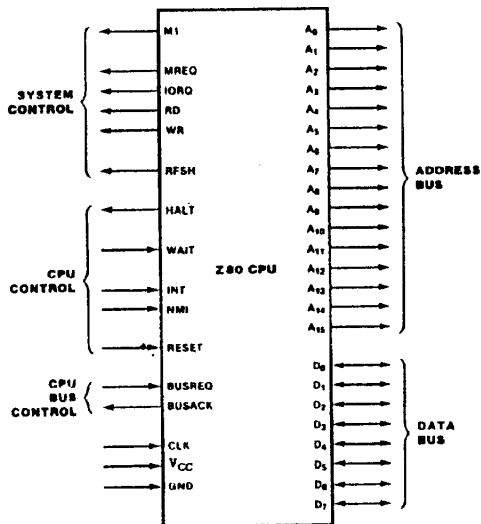


Figure 1. Logic Functions

18-1862  
766464

**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.

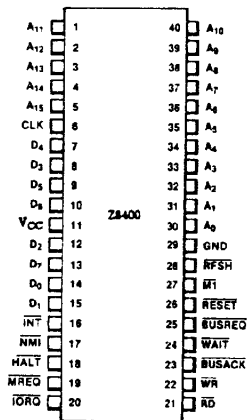


Figure 2. Pin Configuration

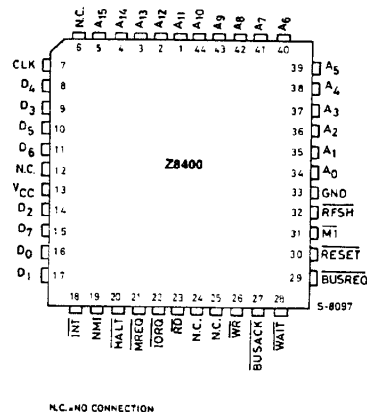


Figure 2a. Chip Carrier Pin Configuration

**General Description (Continued)**

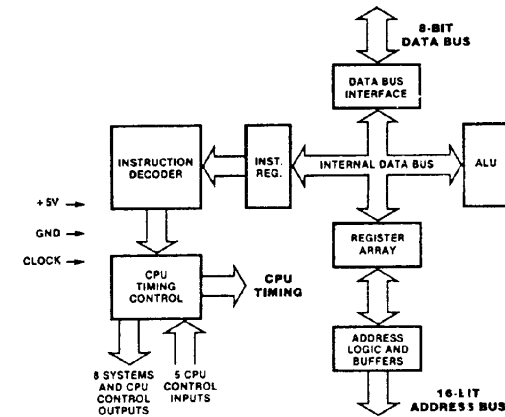


Figure 3. CPU Block Diagram

**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 efficient and cost-effective microcomputer-base systems.

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

- The CTC (Counter/Timer Circuit) features four programmable 8-bit counter/timers, each of which has an 8-bit prescaler. Each of the four channels may be configured 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 peripheral devices such as printers, tape punches, and keyboards.

- The DMA (Direct Memory Access) controller provides dual port data transfer operations and the ability to terminate 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 modes for both synchronous and asynchronous communication, including Bi-Synch and SDLIC.
- 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 an additional pair of flip-flops which assists in identifying the interrupt mode at any particular time. Table 1 provides further information on these registers.

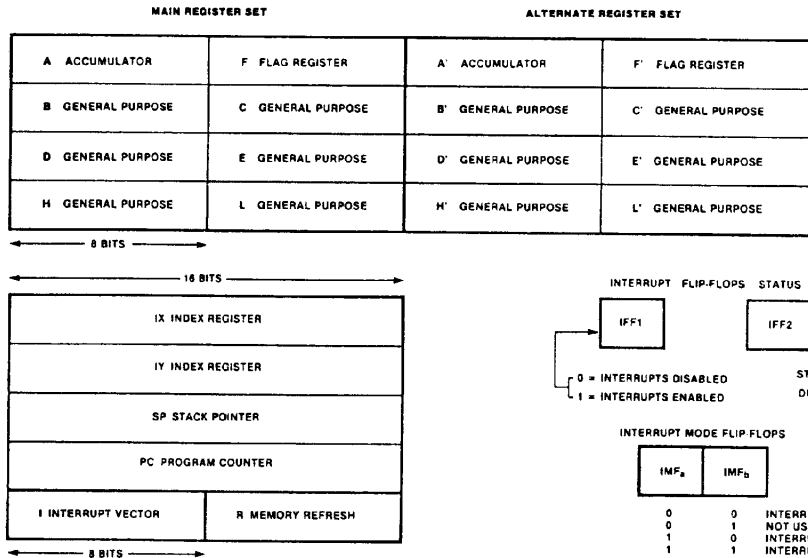
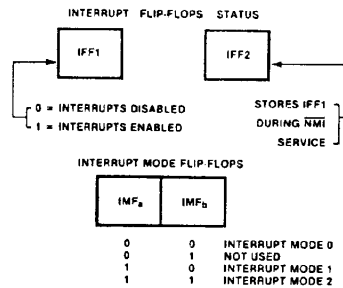


Fig. 4. CPU Registers

**CPU Registers (Continued)**

Register	Size (Bits)	Remarks
A, A'	8	Accumulator Stores an operand or the results of an operation
F, F'	8	Flags See Instruction Set.
B, B'	8	General Purpose Can be used separately or as a 16-bit register with C
C, C'	8	General Purpose See B, above.
D, D'	8	General Purpose Can be used separately or as a 16-bit register with E.
E, E'	8	General Purpose See D, above
H, H'	8	General Purpose Can be used separately or as a 16-bit register with L.
L, L'	8	General Purpose 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	8	Interrupt Register Stores upper eight bits of memory address for vectored interrupt processing.
R	8	Refresh Register 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	16	Index Register Used for indexed addressing.
IY	16	Index Register Same as IX, above.
SP	16	Stack Pointer Holds address of the top of the stack. See Push or Pop in instruction set.
PC	16	Program Counter Holds address of next instruction.
IFF <sub>1</sub> -IFF <sub>2</sub>	Flip-Flops	Interrupt Enable Set or reset to indicate interrupt status (see Figure 4).
IMF <sub>a</sub> -IMF <sub>b</sub>	Flip-Flops	Interrupt Mode Reflect Interrupt mode (see Figure 4).

Table 1. CPU Registers





### Interrupts: General Operation

The CPU accepts two interrupt input signals: 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 Z80 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 — 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 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 (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.

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

**Maskable Interrupt (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 BUSREQ is not active) a special interrupt

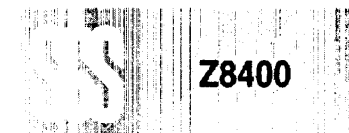
processing cycle begins. This is a special fetch (MI) cycle in which IORQ becomes active rather than MREQ, as in normal MI cycle. In addition, this special MI cycle is automatically extended by two WAIT states, to allow for the time required to acknowledge the interrupt request.

**Mode 0 Interrupt Operation.** This mode is similar 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 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<sub>0</sub>) 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 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 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<sub>1</sub> and IFF<sub>2</sub>, 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 <sub>2</sub>	IFF <sub>1</sub>	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	•	•	IFF <sub>2</sub> → Parity flag
LD A, R instruction execution	•	•	IFF <sub>2</sub> → Parity flag
Accept NMI	0	IFF <sub>1</sub>	IFF <sub>1</sub> → IFF <sub>2</sub> (Maskable interrupt INT disabled)
RETN instruction execution	IFF <sub>2</sub>	•	IFF <sub>2</sub> → IFF <sub>1</sub> at completion of an 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		P/V	N	C	Opcode		No. of Bytes	No. of M Cycles	No. of T States	Comments
				H	L				78	543 210				
LD r, r'	r ← r'	.	.	X	.	X	.	.	.	01 r r'	1	1	4	r, r' Reg.
LD r, n	r ← n	.	.	X	.	X	.	.	.	00 r 110	2	2	7	000 B 001 C 010 D 011 E 100 H 101 L 111 A
LD r, (HL)	r ← (HL)	.	.	X	.	X	.	.	.	01 r 110	1	2	7	
LD r, (IX+d)	r ← (IX+d)	.	.	X	.	X	.	.	.	11 011 101 DD	3	5	19	
LD r, (IY+d)	r ← (IY+d)	.	.	X	.	X	.	.	.	01 r 101 - d - 11 111 101 FD	3	5	19	
LD (HL), r	(HL) ← r	.	.	X	.	X	.	.	.	01 110 r	1	2	7	
LD (IX+d), r	(IX+d) ← r	.	.	X	.	X	.	.	.	11 011 101 DD	3	5	19	
LD (IY+d), r	(IY+d) ← r	.	.	X	.	X	.	.	.	01 110 r - d - 11 111 101 FD	3	5	19	
LD (HL), n	(HL) ← n	.	.	X	.	X	.	.	.	00 110 110 36	2	3	10	
LD (IX+d), n	(IX+d) ← n	.	.	X	.	X	.	.	.	11 011 101 DD 00 110 110 36 - d - - n -	4	5	19	
LD (IY+d), n	(IY+d) ← n	.	.	X	.	X	.	.	.	11 111 101 FD 00 110 110 36 - d - - n -	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 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 32	3	4	13	
LD A, I	A ← I	1	1	X	0	X	IFF	0	.	11 101 101 ED 01 010 111 57	2	2	9	
LD A, R	A ← R	1	1	X	0	X	IFF	0	.	11 101 101 ED 01 011 111 5F	2	2	9	
LD I, A	I ← A	.	.	X	.	X	.	.	.	11 101 101 ED 01 000 111 47	2	2	9	
LD R, A	R ← A	.	.	X	.	X	.	.	.	11 101 101 ED 01 001 111 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	S	Z	Flags H	P/V	M	C	78	543	210	Hex	No. of Bytes	No. of Cycles	No. of States	Comments
LD dd, nn	dd - nn	.	.	X	.	X	.	.	.	.	00 dd0 001	3	3	10	dd Pair 00 BC 01 DE 10 HL 11 SP
LD IX, nn	IX - nn	.	.	X	.	X	.	.	.	.	11 011 101 DD	4	4	14	
LD IY, nn	IY - nn	.	.	X	.	X	.	.	.	.	11 111 101 FD	4	4	14	
LD HL, (nn)	H - (nn+1) L - (nn)	.	.	X	.	X	.	.	.	.	00 101 010 2A	3	5	16	
LD dd, (nn)	ddH - (nn+1) ddL - (nn)	.	.	X	.	X	.	.	.	.	11 101 101 ED	4	6	20	
LD IX, (nn)	IXH - (nn+1) IXL - (nn)	.	.	X	.	X	.	.	.	.	11 011 101 DD	4	6	20	
LD IY, (nn)	IYH - (nn+1) IYL - (nn)	.	.	X	.	X	.	.	.	.	11 111 101 FD	4	6	20	
LD (nn), HL	(nn+1) - H (nn) - L	.	.	X	.	X	.	.	.	.	00 100 010 22	3	5	16	
LD (nn), dd	(nn+1) - ddH (nn) - ddL	.	.	X	.	X	.	.	.	.	11 101 101 ED	4	6	20	
LD (nn), IX	(nn+1) - IXH (nn) - IXL	.	.	X	.	X	.	.	.	.	11 011 101 DD	4	6	20	
LD (nn), IY	(nn+1) - IYH (nn) - IYL	.	.	X	.	X	.	.	.	.	11 111 101 FD	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 DD	2	2	10	
LD SP, IY	SP - IY	.	.	X	.	X	.	.	.	.	11 111 101 FD	2	2	10	
PUSH qq	(SP-2) - qqL (SP-1) - qqH 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) - IXL (SP-1) - IXH SP - SP - 2	.	.	X	.	X	.	.	.	.	11 011 101 DD	2	4	15	
PUSH IY	(SP-2) - IYL (SP-1) - IYH SP - SP - 2	.	.	X	.	X	.	.	.	.	11 111 101 FD	2	4	15	
POP qq	qqH - (SP+1) qqL - (SP) SP - SP + 2	.	.	X	.	X	.	.	.	.	11 qq0 001	1	3	10	
POP IX	IXH - (SP+1) IXL - (SP) SP - SP + 2	.	.	X	.	X	.	.	.	.	11 011 101 DD	2	4	14	
POP IY	IYH - (SP+1) IYL - (SP) SP - SP + 2	.	.	X	.	X	.	.	.	.	11 111 101 FD	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)<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

EX DE, HL	DE - HL	.	.	X	.	X	.	.	.	.	11 101 011 EB	1	1	4	Register bank and auxiliary register bank exchange
EX AF, AF'	AF - AF'	.	.	X	.	X	.	.	.	.	00 001 000 08	1	1	4	
EXX	BC - BC'	.	.	X	.	X	.	.	.	.	11 011 001 D9	1	1	4	
	DE - DE' HL - HL'	.	.	X	.	X	.	.	.	.	11 100 011 E3	1	5	19	
EX (SP), HL	H - (SP+1) L - (SP)	.	.	X	.	X	.	.	.	.	11 100 011 E3	1	5	19	
EX (SP), IX	IXH - (SP+1) IXL - (SP)	.	.	X	.	X	.	.	.	.	11 011 101 DD	2	6	23	
EX (SP), IY	IYH - (SP+1) IYL - (SP)	.	.	X	.	X	.	.	.	.	11 111 101 FD	2	6	23	
LDI	(DE) - (HL) DE - DE+1 HL - HL+1 BC - BC-1	.	.	X	0	X	1	0	.	.	11 101 101 ED	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
LDIR	(DE) - (HL) DE - DE+1 HL - HL+1 BC - BC-1 Repeat until BC = 0	.	.	X	0	X	0	0	.	.	11 101 101 ED	2	5	21	If BC ≠ 0
		.	.	X	0	X	0	0	.	.	10 110 000 B0	2	4	16	If BC = 0
LDD	(DE) - (HL) DE - DE-1 HL - HL-1 BC - BC-1	.	.	X	0	X	1	0	.	.	11 101 101 ED	2	4	16	
LDDR	(DE) - (HL) DE - DE-1 HL - HL-1 BC - BC-1 Repeat until BC = 0	.	.	X	0	X	0	0	.	.	11 101 101 ED	2	5	21	If BC ≠ 0
		.	.	X	0	X	0	0	.	.	10 111 000 B8	2	4	16	If BC = 0
CPH	A - (HL) HL - HL+1 BC - BC-1	1	1	X	1	X	1	1	.	.	11 101 101 ED	2	4	16	
CPHR	A - (HL) HL - HL+1 BC - BC-1 Repeat until A = (HL) or BC = 0	1	1	X	1	X	1	1	.	.	11 101 101 ED	2	5	21	If BC ≠ 0 and A ≠ (HL)
		1	1	X	1	X	1	1	.	.	10 110 001 B1	2	4	16	If BC = 0 or A = (HL)
CPD	A - (HL) HL - HL-1 BC - BC-1	1	1	X	1	X	1	1	.	.	11 101 101 ED	2	4	16	
CPDR	A - (HL) HL - HL-1 BC - BC-1 Repeat until A = (HL) or BC = 0	1	1	X	1	X	1	1	.	.	11 101 101 ED	2	5	21	If BC ≠ 0 and A ≠ (HL)
		1	1	X	1	X	1	1	.	.	10 111 001 B9	2	4	16	If BC = 0 or A = (HL)

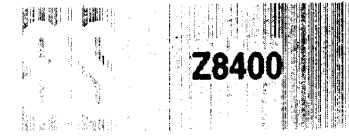
NOTE: ① If the result of B-1 is zero the Z flag is set, otherwise it is reset.  
② Z flag is set upon instruction completion only.



### 8-Bit Arithmetic and Logical Group

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 78 543 210 Hex	No. of Bytes	No. of Cycles	No. of M States	No. of T States	Comments
ADD A, r	A ← A + r	1	1	X	1	X	V	0	1	1	4	r Reg.	
ADD A, n	A ← A + n	1	1	X	1	X	V	0	1	2	2	7	000 B 001 C 010 D 011 E
ADD A, (HL)	A ← A + (HL)	1	1	X	1	X	V	0	1	1	2	7	100 H
ADD A, (IX+d)	A ← A + (IX+d)	1	1	X	1	X	V	0	1	3	5	19	101 L 111 A
ADD A, (IY+d)	A ← A + (IY+d)	1	1	X	1	X	V	0	1	3	5	19	
ADC A, s	A ← A + s + CY	1	1	X	1	X	V	0	1	1	1	4	s is any of r, n, (HL), (IX+d), (IY+d) as shown for ADD instruction. The indicated bits replace the [00] in the ADD set above.
SUB s	A ← A - s	1	1	X	1	X	V	1	1	1	1	4	
SBC A, s	A ← A - s - CY	1	1	X	1	X	V	1	1	1	1	4	
AND s	A ← A ∧ s	1	1	X	1	X	P	0	0	1	1	4	
OR s	A ← A ∨ s	1	1	X	0	X	P	0	0	1	1	4	
XOR s	A ← A ⊕ s	1	1	X	0	X	P	0	0	1	1	4	
CP s	A ← s	1	1	X	1	X	V	1	1	1	1	4	
INC r	r ← r + 1	1	1	X	1	X	V	0	*	1	1	4	
INC (HL)	(HL) ← (HL) + 1	1	1	X	1	X	V	0	*	1	3	11	
INC (IX+d)	(IX+d) ← (IX+d) + 1	1	1	X	1	X	V	0	*	3	6	23	
INC (IY+d)	(IY+d) ← (IY+d) + 1	1	1	X	1	X	V	0	*	3	6	23	
DEC m	m ← m - 1	1	1	X	1	X	V	1	*	1	1	4	m is any of r, (HL), (IX+d), (IY+d) as shown for INC. DEC same format and states as INC. Replace [00] with [01] in opcode.

m is any of r, (HL), (IX+d), (IY+d) as shown for INC. DEC same format and states as INC. Replace [00] with [01] in opcode.



### General-Purpose Arithmetic and CPU Control Groups

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 78 543 210 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.	1	1	X	1	X	P	*	1	1	1	4	Decimal adjust accumulator.	
CPL	A ← $\bar{A}$	*	*	X	1	X	*	1	*	1	1	4	Complement accumulator (one's complement).	
NEG	A ← 0 - A	1	1	X	1	X	V	1	1	2	2	8	Negate acc. (two's complement).	
CCF	CY ← $\bar{CY}$	*	*	X	X	X	*	0	1	1	1	4	Complement carry flag.	
SCF	CY ← 1	*	*	X	0	X	*	0	1	1	1	4	Set carry flag.	
NOP	No operation	*	*	X	*	X	*	*	*	0	0	0	4	
HALT	CPU halted	*	*	X	*	X	*	*	*	1	1	1	4	
DI *	IFF ← 0	*	*	X	*	X	*	*	*	1	1	1	4	
EI *	IFF ← 1	*	*	X	*	X	*	*	*	1	1	1	4	
IM 0	Set interrupt mode 0	*	*	X	*	X	*	*	*	2	2	8		
IM 1	Set interrupt mode 1	*	*	X	*	X	*	*	*	2	2	8		
IM 2	Set interrupt mode 2	*	*	X	*	X	*	*	*	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	1	1	3	11	ss Reg. 00 BC
ADC HL, ss	HL ← HL + ss + CY	1	1	X	X	X	V	0	1	2	4	15	01 DE 10 HL 11 SP
SBC HL, ss	HL ← HL - ss - CY	1	1	X	X	X	V	1	1	2	4	15	
ADD IX, pp	IX ← IX + pp	*	*	X	X	X	*	0	1	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY, rr	IY ← IY + rr	*	*	X	X	X	*	0	1	2	4	15	rr Reg. 00 BC 01 DE 10 IY 11 SP
INC ss	ss ← ss + 1	*	*	X	*	X	*	*	*	1	1	6	
INC IX	IX ← IX + 1	*	*	X	*	X	*	*	*	2	2	10	
INC IY	IY ← IY + 1	*	*	X	*	X	*	*	*	2	2	10	
DEC ss	ss ← ss - 1	*	*	X	*	X	*	*	*	1	1	6	
DEC IX	IX ← IX - 1	*	*	X	*	X	*	*	*	2	2	10	
DEC IY	IY ← IY - 1	*	*	X	*	X	*	*	*	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 78 543 210 Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments		
RLCA		.	.	X	0	X	.	00 000 111	07	1	4	Rotate left circular accumulator.		
RLA		.	.	X	0	X	.	00 010 111	17	1	4	Rotate left accumulator.		
RRCA		.	.	X	0	X	.	00 001 111	0F	1	4	Rotate right circular accumulator.		
RRA		.	.	X	0	X	.	00 011 111	1F	1	4	Rotate right accumulator.		
RLC r		1	1	X	0	X	P	0 1	11 001 011	CB	2	8	Rotate left circular register r.	
RLC (HL)		1	1	X	0	X	P	0 1	00 <input type="checkbox"/> r	11 001 011	CB	2	4	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
									00 <input type="checkbox"/> 110					
									00 <input type="checkbox"/> 110					
RLC (IX+d)		1	1	X	0	X	P	0 1	11 011 101	DD	4	6	23	
	r, (HL), (IX+d), (IY+d)								11 001 011	CB				
									00 <input type="checkbox"/> 110					
RLC (IY+d)		1	1	X	0	X	P	0 1	11 111 101	FD	4	6	23	
									11 001 011	CB				
									00 <input type="checkbox"/> 110					
RL m		1	1	X	0	X	P	0 1	00 <input type="checkbox"/> 110					
	m = r, (HL), (IX+d), (IY+d)								<input type="checkbox"/> 110					
RRC m		1	1	X	0	X	P	0 1	00 <input type="checkbox"/> 110					
	m = r, (HL), (IX+d), (IY+d)								<input type="checkbox"/> 110					
RR m		1	1	X	0	X	P	0 1	00 <input type="checkbox"/> 110					
	m = r, (HL), (IX+d), (IY+d)								<input type="checkbox"/> 110					
SLA m		1	1	X	0	X	P	0 1	00 <input type="checkbox"/> 110					
	m = r, (HL), (IX+d), (IY+d)								<input type="checkbox"/> 110					
SRA m		1	1	X	0	X	P	0 1	00 <input type="checkbox"/> 110					
	m = r, (HL), (IX+d), (IY+d)								<input type="checkbox"/> 110					
SRL m		1	1	X	0	X	P	0 1	00 <input type="checkbox"/> 110					
	m = r, (HL), (IX+d), (IY+d)								<input type="checkbox"/> 110					
RLD		1	1	X	0	X	P	0 *	11 101 101	ED	2	5	18	Rotate digit left and right between the accumulator and location (HL).
	A (HL)								01 101 111	6F				The content of the upper half of the accumulator is unaffected.
RRD		1	1	X	0	X	P	0 *	11 101 101	ED	2	5	18	
	A (HL)								01 100 111	67				

Instruction format and states are as shown for RLC's. To form new opcode replace  or RLC's with shown code.

### Bit Set, Reset and Test Group

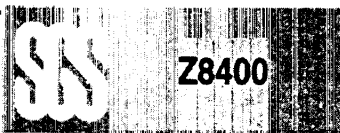
Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 78 543 210 Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments		
BIT b, r	Z = r <sub>b</sub>	X	1	X	1	X	X	0 *	11 001 011	CB	2	2	8	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
BIT b, (HL)	Z = (HL) <sub>b</sub>	X	1	X	1	X	X	0 *	11 001 011	CB	2	3	12	000 B 001 C 010 D 011 E 100 H 101 L 111 A
BIT b, (IX+d) <sub>b</sub>	Z = (IX+d) <sub>b</sub>	X	1	X	1	X	X	0 *	11 011 101	DD	4	5	20	000 0 001 1 010 2 011 3 100 4 101 5 110 6 111 7
									11 001 011	CB				Bit Tested
									01 b 110					
BIT b, (IY+d) <sub>b</sub>	Z = (IY+d) <sub>b</sub>	X	1	X	1	X	X	0 *	11 111 101	FD	4	5	20	000 0 001 1 010 2 011 3 100 4 101 5 110 6 111 7
									11 001 011	CB				
									01 b 110					
SET b, r	r <sub>b</sub> = 1	.	.	X	.	X	.	.	11 001 011	CB	2	2	8	
									<input type="checkbox"/> b r					
SET b, (HL)	(HL) <sub>b</sub> = 1	.	.	X	.	X	.	.	11 001 011	CB	2	4	15	
									<input type="checkbox"/> b 110					
SET b, (IX+d)	(IX+d) <sub>b</sub> = 1	.	.	X	.	X	.	.	11 011 101	DD	4	6	23	
									11 001 011	CB				
									- d -					
									<input type="checkbox"/> b 110					
SET b, (IY+d)	(IY+d) <sub>b</sub> = 1	.	.	X	.	X	.	.	11 111 101	FD	4	6	23	
									11 001 011	CB				
									- d -					
									<input type="checkbox"/> b 110					
RES b, m	m <sub>b</sub> = 0 m = r, (HL), (IX+d), (IY+d)	.	.	X	.	X	.	.	11					To form new opcode replace <input type="checkbox"/> of SET b, s with <input type="checkbox"/> . Flags and time states for SET instruction.

NOTES: The notation m<sub>b</sub> indicates bit b (0 to 7) or location m.

### Jump Group

JP nn	PC = nn	.	.	X	.	X	.	.	11 000 011	C3	3	3	10	
									- n -					
JP cc, nn	If condition cc is true PC = nn, otherwise continue	.	.	X	.	X	.	.	11 cc 010		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
									- n -					
JR e	PC = PC + e	.	.	X	.	X	.	.	00 011 000	18	2	3	12	
									- e-2 -					
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.
									- e-2 -					
JR NC, e	If C = 0, continue If C = 1, PC = PC + e	.	.	X	.	X	.	.	00 110 000	30	2	2	7	If condition is met.
									- e-2 -					
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.
									- e-2 -					
JR NZ, e	If Z = 1, continue If Z = 0, PC = PC + e	.	.	X	.	X	.	.	00 100 000	20	2	2	7	If condition is met.
									- e-2 -					





Z8400

Jump Grup (Continued)

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 76 543 210 Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
	continue If Z = 0, PC - PC + e PC - HL	.	.	X	.	X	.	11 101 001 E9	1	1	4	If condition is met.
JP (HL)		.	.	X	.	X	.	11 011 101 DD 11 101 001 E9	2	2	8	
JP (IX)		.	.	X	.	X	.	11 111 101 FD 11 101 001 E9	2	2	8	
JP (IY)		.	.	X	.	X	.	00 010 000 10 -- e-2 --	2	2	8	If B = 0.
DINZ, e	B - B - 1 If B = 0, continue If B ≠ 0, PC - PC + e	.	.	X	.	X	.		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	Flags 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) - PCH (SP-2) - PCL PC - nn	.	.	X	.	X	.	11 001 101 CD - n - - n -	3	5	17	
CALL cc, nn	If condition cc is false continue, otherwise same as CALL nn	.	.	X	.	X	.	11 cc 100 - n - - n -	3	3	10	If cc is false.
		.	.	X	.	X	.		3	5	17	If cc is true.
RET	PCL - (SP) PCH - (SP+1)	.	.	X	.	X	.	11 001 001 C9	1	3	10	
RET cc	If condition cc is false continue, otherwise same as RET	.	.	X	.	X	.	11 cc 000	1	1	5	If cc is false.
		.	.	X	.	X	.		1	3	11	If cc is true.
RETI	Return from interrupt	.	.	X	.	X	.	11 101 101 ED 01 001 101 4D	2	4	14	
RETI <sup>1</sup>	Return from non-maskable interrupt	.	.	X	.	X	.	11 101 101 ED 01 000 101 45	2	4	14	
RST p	(SP-1) - PCH (SP-2) - PCL PCH - 0 PCL - p	.	.	X	.	X	.	11 t 111	1	3	11	t p 000 00H 001 08H 010 10H 011 18H 100 20H 101 28H 110 30H 111 38H

NOTE: <sup>1</sup>RETI loads IFF<sub>2</sub> - IFF<sub>1</sub>



Z8400

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 DB - n -	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 - (C) if r = 110 only the flags will be affected	1	1	X	1	X	P 0 *	11 101 101 ED 01 r 000	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) - (C) B - B - 1 HL - HL + 1	X	1	X	X	X	1 X	11 101 101 ED 10 100 010 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) - (C) B - B - 1 HL - HL + 1 Repeat until B = 0	X	1	X	X	X	1 X	11 101 101 ED 10 110 010 B2	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) - (C) B - B - 1 HL - HL - 1	X	1	X	X	X	1 X	11 101 101 ED 10 101 010 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) - (C) B - B - 1 HL - HL - 1 Repeat until B = 0	X	1	X	X	X	1 X	11 101 101 ED 10 111 010 BA	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) - A	.	.	X	.	X	.	11 010 011 D3 - n -	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) - r	.	.	X	.	X	.	11 101 101 ED 01 r 001	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) - (HL) B - B - 1 HL - HL + 1	X	1	X	X	X	1 X	11 101 101 ED 10 100 011 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) - (HL) B - B - 1 HL - HL + 1 Repeat until B = 0	X	1	X	X	X	1 X	11 101 101 ED 10 110 011 B3	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) - (HL) B - B - 1 HL - HL - 1	X	1	X	X	X	1 X	11 101 101 ED 10 101 011 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) - (HL) B - B - 1 HL - HL - 1 Repeat until B = 0	X	1	X	X	X	1 X	11 101 101 ED 10 111 011	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

Instruction	D <sub>7</sub> S	Z	H	P/V	N	D <sub>0</sub> C	Comments		
ADD A, s; ADC A, s	1	1	X	1	X	V	0	1	8-bit add or add with carry.
SUB s; SBC A, s; CP s; NEG	1	1	X	1	X	V	1	1	8-bit subtract, subtract with carry, compare and negate accumulator.
AND s	1	1	X	1	X	P	0	0	Logical operations.
OR s; XOR s	1	1	X	0	X	P	0	0	
INC s	1	1	X	1	X	V	0	*	8-bit increment.
DEC s	1	1	X	1	X	V	1	*	8-bit decrement.
ADD DD, ss	*	*	X	X	X	V	0	1	16-bit add.
ADC HL, ss	1	1	X	X	X	V	0	1	16-bit add with carry.
SBC HL, ss	1	1	X	X	X	V	1	1	16-bit subtract with carry.
RLA, RLCA, RRA; RRCA	*	*	X	0	X	*	0	1	Rotate accumulator.
RL m; RLC m; RR m;	1	1	X	0	X	P	0	1	Rotate and shift locations.
RRC m; SLA m;									
SRA m; SRL m									
RLD; RRD	1	1	X	0	X	P	0	*	Rotate digit left and right.
DAA	1	1	X	1	X	P	*	1	Decimal adjust accumulator.
CPL	*	*	X	1	X	*	1	*	Complement accumulator.
SCF	*	*	X	0	X	*	0	1	Set carry.
CCF	*	*	X	X	X	*	0	1	Complement carry.
IN r (C)	1	1	X	0	X	P	0	*	Input register indirect.
INI, IND, OUTI; OUTD	X	1	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	1	0	*	Block transfer instructions. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
LDIR; LDDR	X	X	X	0	X	0	0	*	
CP; CPIR; CPD; CPDR	X	1	X	X	X	1	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	1	1	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	1	X	1	X	X	0	*	The state of bit b of location s is copied into the Z flag.

Symbol	Operation	Symbol	Operation
S	Sign flag. S = 1 if the MSB of the result is 1.	I	The flag is affected according to the result of the operation.
Z	Zero flag. Z = 1 if the result of the operation is 0.	*	The flag is unchanged by the operation.
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.	0	The flag is reset by the operation.
H	Half-carry flag. H = 1 if the add or subtract operation produced a carry into or borrow from bit 4 of the accumulator.	1	The flag is set by the operation.
N	Add/Subtract flag. N = 1 if the previous operation was a subtract.	X	The flag is a "don't care."
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.	V	P/V flag affected according to the overflow result of the operation.
C	Carry/Link flag. C = 1 if the operation produced a carry from the MSB of the operand or result.	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<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 64K bytes) and for I/O device exchanges.

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

**BUSREQ. Bus Request** (input, active Low). Bus Request has a higher priority than NMI and is always recognized at the end of the current machine cycle. BUSREQ forces the CPU address bus, data bus, and control signals MREQ, IORQ, RD, and WR to go to a high-impedance state so that other devices can control these lines. BUSREQ is normally 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 resume. While halted, the CPU executes NOPs to maintain memory refresh.

**INT. Interrupt Request** (input, active Low). Interrupt Request is generated by I/O devices. The CPU honors a request at the end of the current instruction if the internal software-controlled interrupt enable flip-flop (IFF) is enabled. INT is normally 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 MI during an interrupt acknowledge cycle to indicate that an interrupt response vector

can be placed on the data bus. **MI. Machine Cycle One** (output, active Low). MI, together with MREQ, indicates that the current machine cycle is the opcode fetch cycle of an instruction execution. MI, 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

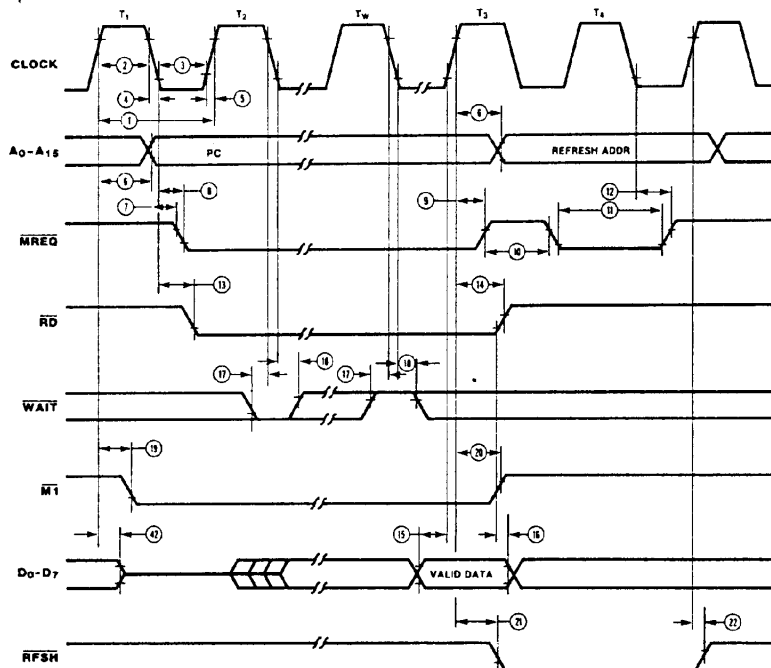
The Z80 CPU executes instructions by proceeding through a specific sequence of operations:

- 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 (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/W pulse to most semiconductor memories.

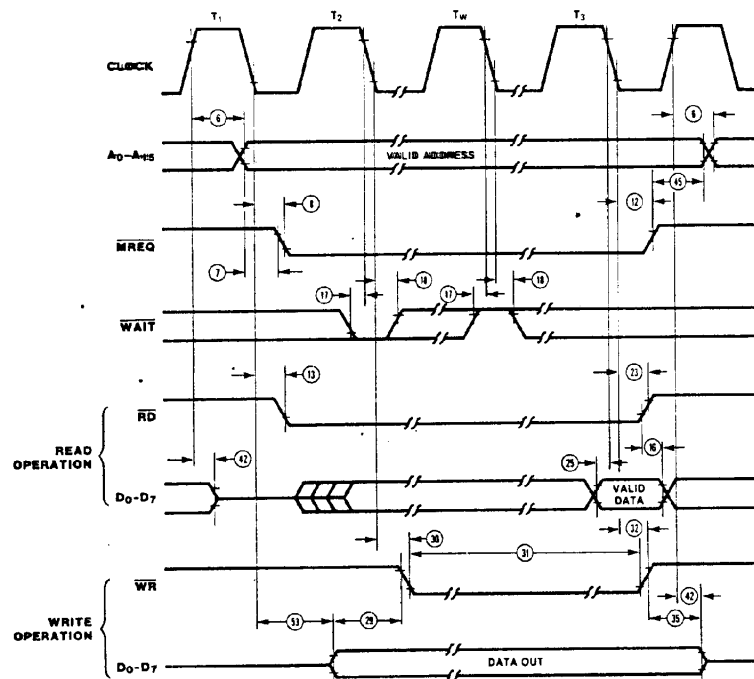
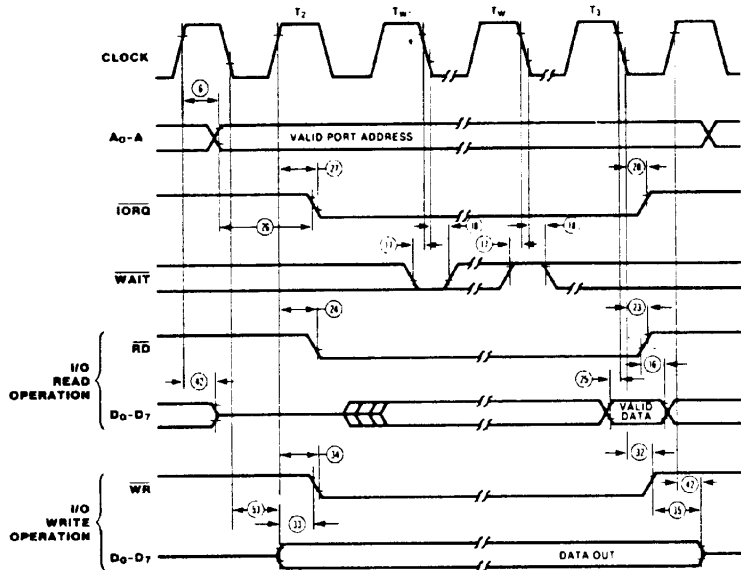


Figure 6. Memory Read or Write Cycles

**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.



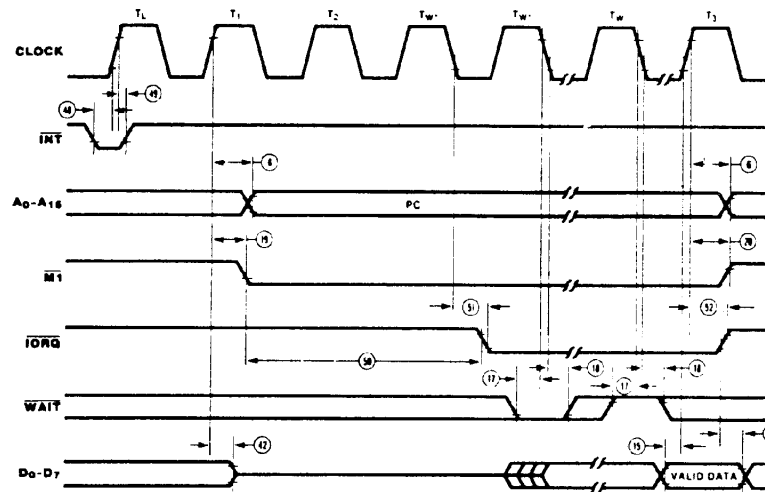
NOTE:  $T_w$  = One Wait cycle automatically inserted by CPU.

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 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.



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

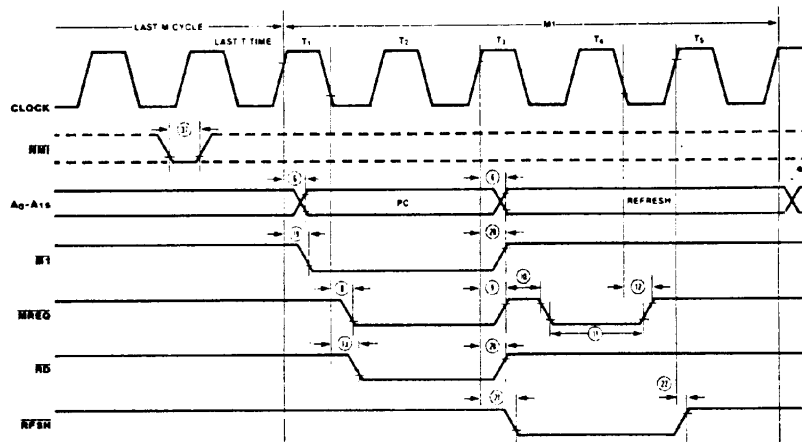
2) Two Wait cycles automatically inserted by CPU(\*).

Figure 8. Interrupt Request/Acknowledge Cycle

CPU Timing (Continued)

**Non-Maskable Interrupt Request Cycle.** NMI is sampled at the same time as the maskable interrupt input INT but has higher priority and cannot be disabled under software control. The subsequent timing is similar to that of a normal 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).



\*Although NMI is an asynchronous input, to guarantee its being recognized on the following machine cycle, NMI's falling edge

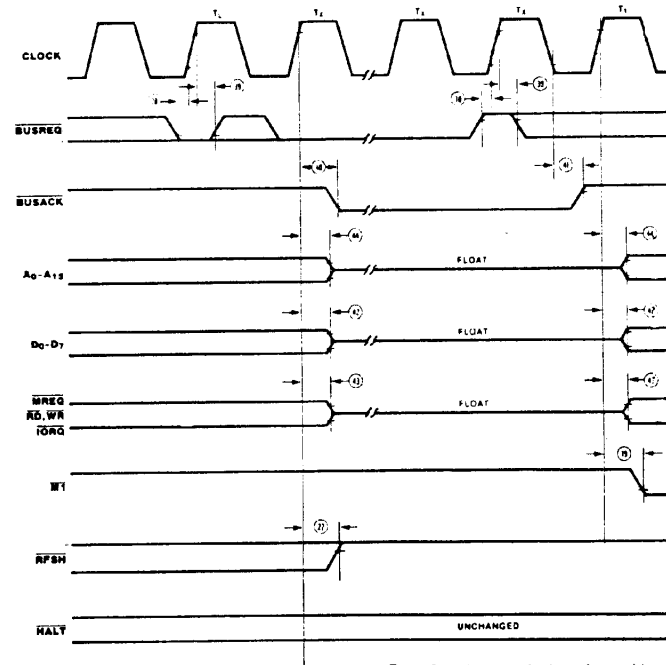
must occur no later than the rising edge of the clock cycle preceding T<sub>LAST</sub>.

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.



NOTE: T<sub>L</sub> = Last state of any M cycle.

T<sub>X</sub> = An arbitrary clock cycle used by requesting device.

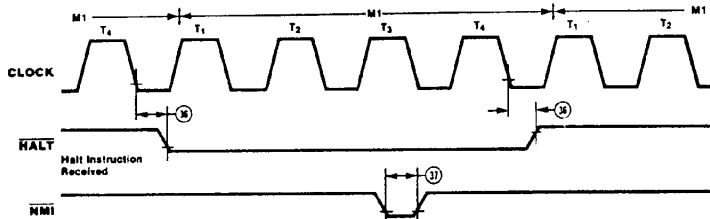
Figure 10. Z-Bus Request/Acknowledge Cycle

CPU Timing (Continued)

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

**Reset Cycle.**  $\overline{RESET}$  must be active for at least three clock cycles for the CPU to

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



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

\*See note, Figure 9.

Figure 11. Halt Acknowledge Cycle

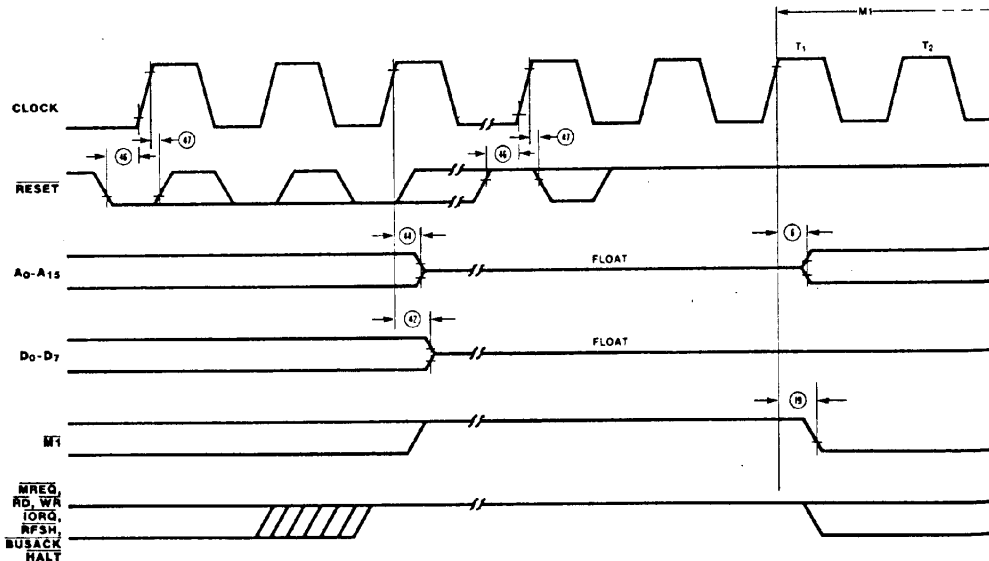
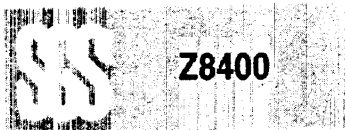


Figure 12. Reset Cycle

AC Characteristics

Number	Symbol	Parameter	Z8400		Z8400A		Z8400B		Z8400H	
			Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)
1	T <sub>c</sub>	Clock Cycle Time	400*	—	250*	—	165*	—	125*	—
2	T <sub>wCh</sub>	Clock Pulse Width (High)	180*	—	110*	—	65*	—	55*	—
3	T <sub>wCl</sub>	Clock Pulse Width (Low)	180	2000	110	2000	65	2000	55	2000
4	T <sub>fC</sub>	Clock Fall Time	—	30	—	30	—	20	—	10
5	T <sub>rC</sub>	Clock Rise Time	—	30	—	30	—	20	—	10
6	T <sub>dCr(A)</sub>	Clock ↑ to Address Valid Delay	—	145	—	110	—	90	—	80
7	T <sub>dA(MREQ)</sub>	Address Valid to $\overline{MREQ}$ ↓ Delay	125*	—	65*	—	35*	—	20*	—
8	T <sub>dC(MREQ)</sub>	Clock ↓ to $\overline{MREQ}$ ↓ Delay	—	100	—	85	—	70	—	60
9	T <sub>dCr(MREQ)</sub>	Clock ↑ to $\overline{MREQ}$ ↑ Delay	—	100	—	85	—	70	—	60
10	T <sub>wMREQh</sub>	$\overline{MREQ}$ Pulse Width (High)	170*	—	110*	—	65*	—	45*	—
11	T <sub>wMREQl</sub>	$\overline{MREQ}$ Pulse Width (Low)	360*	—	220*	—	135*	—	100*	—
12	T <sub>dC(MREQr)</sub>	Clock ↓ to $\overline{MREQr}$ ↑ Delay	—	100	—	85	—	70	—	60
13	T <sub>dCr(RD)</sub>	Clock ↓ to $\overline{RD}$ ↓ Delay	—	130	—	95	—	80	—	70
14	T <sub>dCr(RDr)</sub>	Clock ↑ to $\overline{RD}$ ↑ Delay	—	100	—	85	—	70	—	60
15	T <sub>sD(Cr)</sub>	Data Setup Time to Clock ↑	50	—	35	—	30	—	30	—
16	T <sub>hD(RDr)</sub>	Data Hold Time to $\overline{RD}$ ↑	—	0	—	0	—	0	—	0
17	T <sub>sWAIT(C)</sub>	$\overline{WAIT}$ Setup Time to Clock ↓	70	—	70	—	60	—	50	—
18	T <sub>hWAIT(C)</sub>	$\overline{WAIT}$ Hold Time after Clock ↓	—	0	—	0	—	0	—	0
19	T <sub>dCr(M1)</sub>	Clock ↑ to $\overline{M1}$ ↓ Delay	—	130	—	100	—	80	—	70
20	T <sub>dCr(M1r)</sub>	Clock ↑ to $\overline{M1}$ ↑ Delay	—	130	—	100	—	80	—	70
21	T <sub>dCr(RFSH)</sub>	Clock ↑ to $\overline{RFSH}$ ↓ Delay	—	180	—	130	—	110	—	95
22	T <sub>dCr(RFSHr)</sub>	Clock ↑ to $\overline{RFSH}$ ↑ Delay	—	150	—	120	—	100	—	85
23	T <sub>dC(RD)</sub>	Clock ↓ to $\overline{RD}$ ↑ Delay	—	110	—	85	—	70	—	60
24	T <sub>dCr(RDr)</sub>	Clock ↑ to $\overline{RD}$ ↓ Delay	—	110	—	85	—	70	—	60
25	T <sub>sD(C)</sub>	Data Setup to Clock ↓ during M <sub>2</sub> , M <sub>3</sub> , M <sub>4</sub> or M <sub>5</sub> Cycles	60	—	50	—	40	—	30	—
26	T <sub>dA(IORQ)</sub>	Address Stable prior to $\overline{IORQ}$ ↓	320*	—	180*	—	110*	—	75*	—
27	T <sub>dCr(IORQ)</sub>	Clock ↑ to $\overline{IORQ}$ ↓ Delay	—	90	—	75	—	65	—	55
28	T <sub>dC(IORQr)</sub>	Clock ↓ to $\overline{IORQ}$ ↑ Delay	—	110	—	85	—	70	—	60
29	T <sub>dC(WR)</sub>	Data Stable prior to $\overline{WR}$ ↓	190*	—	80*	—	25*	—	5*	—
30	T <sub>dD(WR)</sub>	Clock ↓ to $\overline{WR}$ ↓ Delay	—	90	—	80	—	70	—	60
31	T <sub>wWR</sub>	$\overline{WR}$ Pulse Width	360*	—	220*	—	135*	—	100*	—
32	T <sub>dC(WRr)</sub>	Clock ↓ to $\overline{WR}$ ↑ Delay	—	100	—	80	—	70	—	60
33	T <sub>dD(WR)</sub>	Data Stable prior to $\overline{WR}$ ↑	20*	—	-10*	—	-55*	—	55*	—
34	T <sub>dCr(WR)</sub>	Clock ↑ to $\overline{WR}$ ↓ Delay	—	80	—	65	—	60	—	55
35	T <sub>dWR(D)</sub>	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.



Z8400

AC Characteristics (Continued)

Number	Symbol	Parameter	Z8400		Z8400A		Z8400B		Z8400H	
			Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)
36	TdC(HALT)	Clock ↓ to HALT ↑ or ↓	—	300	—	300	—	260	—	225
37	TwNMI	NMI Pulse Width	80	—	80	—	70	—	60*	—
38	TsBUSREQ(Cr)	BUSREQ Setup Time to Clock ↑	80	—	50	—	50	—	40	—
39	TcBUSUREQ(Cr)	BUSREQ Hold Time after Clock ↑	0	—	0	—	0	—	0	—
40	TdCr(BUSACKf)	Clock ↑ to BUSACK ↓ Delay	—	120	—	100	—	90	—	80
41	TdC(BUSACKr)	Clock ↓ to 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 (MREQ, IORQ, RD, and WR)	—	110	—	80	—	70	—	60
44	TdCr(Az)	Clock ↑ to Address Float Delay	—	110	—	90	—	80	—	70
45	TdCTr(A)	MREQ ↑, IORQ ↑, RD ↑, and WR ↑ to Address Hold Time	160*	—	80*	—	35*	—	20*	—
46	TsRESET(Cr)	RESET to Clock ↑ Setup Time	90	—	60	—	60	—	45	—
47	ThRESET(Cr)	RESET to Clock ↑ Hold Time	—	0	—	0	—	0	—	0
48	TsINTf(Cr)	INT to Clock ↑ Setup Time	80	—	80	—	70	—	55	—
49	ThINTr(Cr)	INT to Clock ↑ Hold Time	—	0	—	0	—	0	—	0
50	TdMif(IORQf)	MI ↓ to IORQ ↓ Delay	920*	—	565*	—	365*	—	270*	—
51	TdC(IORQf)	Clock ↓ to IORQ ↓ Delay	—	110	—	85	—	70	—	60
52	TdC(IORQr)	Clock ↑ to IORQ ↑ Delay	—	100	—	85	—	70	—	60
53	TdC(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.



Z8400

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, 200 μs is not guaranteed	Although static by design, 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	TdMif(IORQf)	2TcC + TwCh + TfC - 80	2TcC + TwCh + TfC - 65	2TcC + TwCh + TfC - 50

AC Test Conditions:  
 $V_{IH} = 2.0 V$   
 $V_{IL} = 0.8 V$   
 $V_{TIC} = V_{CC} - 0.6 V$

$V_{ILC} = 0.45 V$   
 $V_{OIH} = 2.0 V$   
 $V_{OIL} = 0.8 V$   
 $FLOAT = \pm 0.5 V$

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

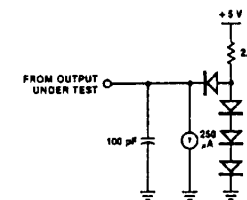
Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating 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 V ≤ V<sub>CC</sub> ≤ +5.25 V
- 40°C to +85°C,  
+4.75 V ≤ V<sub>CC</sub> ≤ +5.25 V
- 55°C to +125°C,  
+4.75 V ≤ V<sub>CC</sub> ≤ +5.25 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.





Z8400

**DC Characteristics**

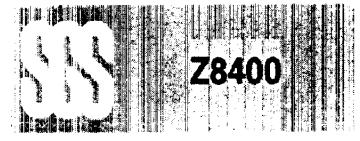
Symbol	Parameter	Min.	Max	Unit	Test Condition
V <sub>ILC</sub>	Clock Input Low Voltage	-0.3	0.45	V	
V <sub>IHC</sub>	Clock Input High Voltage	V <sub>CC</sub> - 0.6	V <sub>CC</sub> + 0.3	V	
V <sub>IL</sub>	Input Low Voltage	-0.3	0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0	V <sub>CC</sub>	V	
V <sub>OL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

1. For military grade parts, I<sub>CC</sub> is 200 mA.
2. Typical rate for Z8400A is 90 mA.
3. A<sub>15</sub>-A<sub>0</sub>, D<sub>7</sub>-D<sub>0</sub>, MREQ, IORQ, RD, and WR.

**Capacitance**

Symbol	Parameter	Min.	Max	Unit	Note
C <sub>CLOCK</sub>	Clock Capacitance		35	pF	
C <sub>IN</sub>	Input Capacitance		5	pf	Unmeasured pins returned to ground
C <sub>OUT</sub>	Output Capacitance		10	pF	

T<sub>A</sub> = 25°C, f = 1 MHz



Z8400

**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		