

Z8300 Low Power Z80[®]L CPU Central Processing Unit

Zilog

Product Specification

April 1985

Features

- The Z80L combines the high performance of the Z80 CPU with extremely low power consumption. It has the identical pinout and instruction set of the Z80. The result is increased reliability and lower system power requirements. This dramatic power savings makes the Z80L a natural choice for both hand-held and battery backup applications.
- The Z80L CPU is offered in two versions: Z8300-1—1.0 MHz clock, 15 mA typical current consumption
Z8300-3—2.5 MHz clock, 25 mA typical current consumption
- The extensive instruction set contains 158 instructions, resulting in sophisticated data handling capabilities. The 78 instructions of the 8080A are included as a subset; 8080A and Z80 Family software compatibility is maintained.
- The Z80L microprocessors and associated family of peripheral controllers are linked by a vectored interrupt system. This system can 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. 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.

Z80 CPU

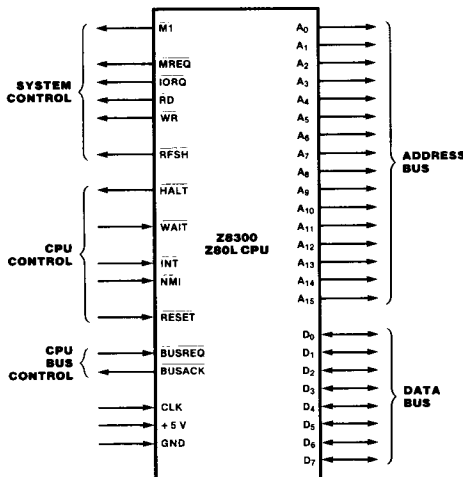


Figure 1. Pin Functions

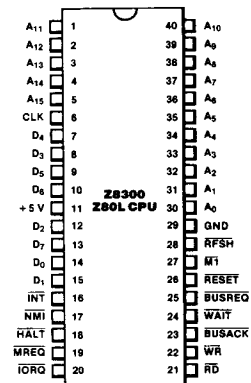


Figure 2. 40-pin Dual-In-Line Package (DIP) Pin Assignments

General Description

The Z80L CPUs are fourth-generation microprocessors with exceptional computational power. They offer high system throughput and efficient memory utilization combined with extremely low power consumption. 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 Z80L 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 Z80L processors. Subsequent text provides more detail on the Z80L I/O controller family, registers, instruction set, interrupts and daisy chaining, CPU timing, and low power requirements.

Z80L Low Power Feature. The Z80L Family offers state-of-the-art microprocessor performance with extremely low power consumption. Its low power requirement rivals comparable CMOS microprocessors. The Z80L Family's lower power consumption provides the ability to reduce system power requirements and enables its use in applications not previously possible. The Z80L is very well suited to battery backup applications or to systems operating primarily on batteries in hand-held or portable systems.

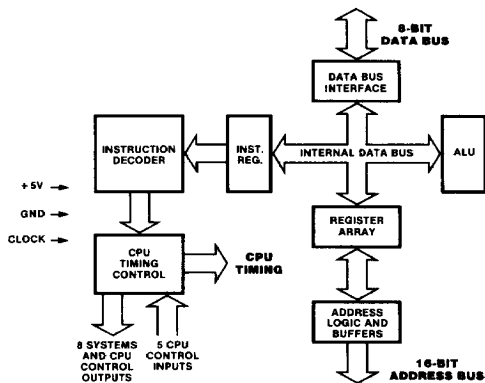


Figure 3. Z80L CPU Block Diagram

Z80L Micro-processor Family

The Zilog Z80L 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-based systems.

The Z80 Family components provide extensive support for the Z80L microprocessor. These are:

- 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 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 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 SDLCL.

Z80L CPU Registers

Figure 4 shows three groups of registers within the Z80L 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.

Z80L CPU

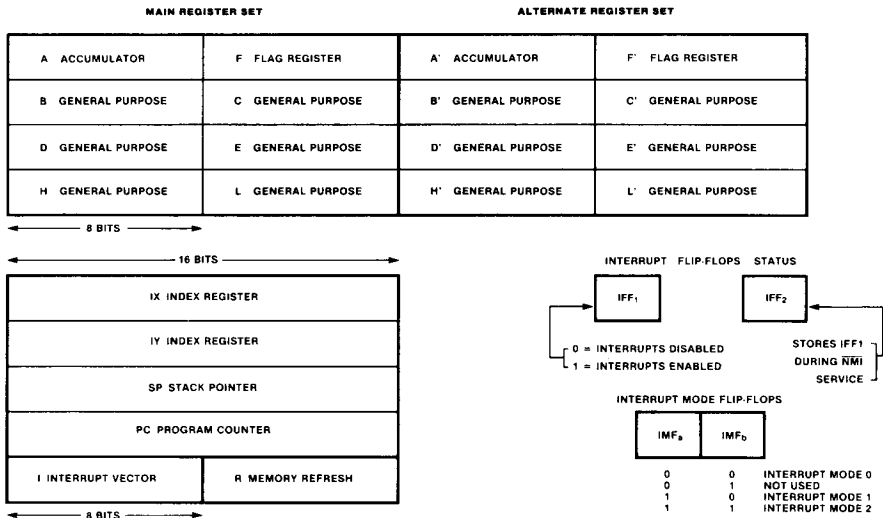


Figure 4. CPU Registers

Z80L 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-transparent dynamic memory refresh. Lower seven bits are automatically incremented and all eight are placed on the address bus during each instruction fetch cycle refresh time.	
IX	Index Register	16	Used for indexed addressing.	
IY	Index Register	16	Same as IX, above.	
SP	Stack Pointer	16	Holds address of the top of the stack. See Push or Pop in instruction set.	
PC	Program Counter	16	Holds address of next instruction.	
IFF 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. Z80L 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 Z80L 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 to the 8080 micro-processor.

- 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.

Interrupts: General Operation (Continued)

Non-Maskable Interrupt (NMI). The non-maskable interrupt cannot be disabled by program control and therefore will be accepted all times by the CPU. NMI is usually reserved for servicing only the highest priority type interrupts, such as that for orderly shutdown 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 Z80L 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 (M1) cycle in which \overline{IORQ} becomes active rather than \overline{MREQ} , as in a normal M1 cycle. In addition, this special M1 cycle is automatically extended by two WAIT states, to allow for the time required to acknowledge the interrupt request and to place the interrupt vector on the bus.

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 vector address of 0038H only.

Mode 2 Interrupt Operation. This interrupt mode has been designed to utilize most effectively the capabilities of the Z80L 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 calls 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 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 Z80L CPU will nest (queue) any pending interrupts or interrupts received while a selected peripheral is being serviced.

Interrupt Enable/Disable Operation. Two flip-flops, IFF₁ and IFF₂, 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* and *Z80 Assembly Language Manual*.

Action	IFF ₁	IFF ₂	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 ₂ — Parity flag
LD A,R instruction execution	•	•	IFF ₂ — Parity flag
Accept \overline{NMI}	0	IFF ₁	IFF ₁ — IFF ₂ (Maskable interrupt INT disabled)
RETN instruction execution	IFF ₂	•	IFF ₂ — IFF ₁ at completion of an NMI service routine.

Table 2. State of Flip-Flops

Instruction Set

The Z80L microprocessor has one of the most powerful and versatile instruction sets available in any 8-bit microprocessor and identical to that of the Z80. 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 Z80L instruction set and shows the assembly language mnemonic, the operation, the flag status, and gives comments on each instruction. The *Z80 CPU Technical Manual (03-0029-XX)* and *Assembly Language Programming Manual (03-0002-XX)* contain significantly more details for programming use.

The instructions in Table 2 are divided into the following categories:

- 8-bit loads
- 16-bit loads
- Exchanges, block transfers, and searches
- 8-bit arithmetic and logic operations
- General-purpose arithmetic and CPU control

- 16-bit arithmetic operations
- Rotates and shifts
- Bit set, reset, and test operations
- Jumps
- Calls, returns, and restarts
- Input and output operations

A variety of addressing modes are implemented to permit efficient and fast data transfer between various registers, memory locations, and input/output devices. These addressing modes include:

- Immediate
- Immediate extended
- Modified page zero
- Relative
- Extended
- Indexed
- Register
- Register indirect
- Implied
- Bit

8-Bit Load Group

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments	
LD r, r'	r ← r'	*	*	X	*	X	*	01 r r'	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 -	DD 36	4	5	19		
LD (IY+d), n	(IY+d) ← n	*	*	X	*	X	*	11 111 101 00 110 110 - d -	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	I	I	X	0	X	IFF	0		2	2	9		
LD A, R	A ← R	I	I	X	0	X	IFF	0		2	2	9		
LD I, A	I ← A	*	*	X	*	X	*	11 101 101 01 010 111	ED 57	2	2	9		
LD R, A	R ← A	*	*	X	*	X	*	01 011 111 11 101 101 01 000 111 11 101 101	ED 47 ED	2	2	9		
								01 001 111	4F					

NOTES: r, r' means any of the registers A, B, C, D, E, H, L.
 IFF the content of the interrupt enable flip-flop. (IFF) is copied into the P/V flag.
 For an explanation of flag notation and symbols for mnemonic tables, see Symbolic Notation section following tables.

16-Bit Load Group

Mnemonic	Symbolic Operation	Flags						Opcode			No. of Bytes	No. of M Cycles	No. of T States	Comments							
		S	Z	H	P/V	N	C	78	543	210				Hex	Pair						
LD dd, nn	dd ← nn	•	•	X	•	X	•	•	•	•	•	•	00	dd0	001	3	3	10	dd 00	Pair BC	
LD IX, nn	IX ← nn	•	•	X	•	X	•	•	•	•	•	•	11	011	101	DD	4	4	14	01	DE
													00	100	001	21				10	HL
													11	—	—	—				11	SP
LD IY, nn	IY ← nn	•	•	X	•	X	•	•	•	•	•	•	11	111	101	FD	4	4	14		
00	100	001	21																		
—	—	—	—																		
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		
01	dd1	011																			
—	—																				
LD IX, (nn)	IXH ← (nn+1) IXL ← (nn)	•	•	X	•	X	•	•	•	•	•	•	11	011	101	DD	4	6	20		
00	101	010	2A																		
—	—																				
LD IY, (nn)	IYH ← (nn+1) IYL ← (nn)	•	•	X	•	X	•	•	•	•	•	•	11	111	101	FD	4	6	20		
00	101	010	2A																		
—	—																				
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		
01	dd0	011																			
—	—																				
LD (nn), IX	(nn+1) ← IXH (nn) ← IXL	•	•	X	•	X	•	•	•	•	•	•	11	011	101	DD	4	6	20		
00	100	010	22																		
—	—																				
LD (nn), IY	(nn+1) ← IYH (nn) ← IYL	•	•	X	•	X	•	•	•	•	•	•	11	111	101	FD	4	6	20		
00	100	010	22																		
—	—																				
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
LD SP, IY	SP ← IY	•	•	X	•	X	•	•	•	•	•	•	11	111	001	F9	2	2	10		
													11	111	101	FD					
PUSH qq	(SP-2) ← qqL (SP-1) ← qqH	•	•	X	•	X	•	•	•	•	•	•	11	qq0	101		1	3	11	qq	Pair
	SP ← SP-2																				
PUSH IX	(SP-2) ← IXL (SP-1) ← IXH	•	•	X	•	X	•	•	•	•	•	•	11	011	101	DD	2	4	15	01	DE
	SP ← SP-2												11	100	101	E5					
PUSH IY	(SP-2) ← IYL (SP-1) ← IYH	•	•	X	•	X	•	•	•	•	•	•	11	111	101	FD	2	4	15	11	AF
	SP ← SP-2												11	100	101	E5					
POP qq	qqH ← (SP+1) qqL ← (SP)	•	•	X	•	X	•	•	•	•	•	•	11	qq0	001		1	3	10		
	SP ← SP+2																				
POP IX	IXH ← (SP+1) IXL ← (SP)	•	•	X	•	X	•	•	•	•	•	•	11	011	101	DD	2	4	14		
	SP ← SP+2												11	100	001	E1					
POP IY	IYH ← (SP+1) IYL ← (SP)	•	•	X	•	X	•	•	•	•	•	•	11	111	101	FD	2	4	14		
	SP ← SP+2												11	100	001	E1					

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		Register bank and auxiliary register bank exchange				
EX AF, AF'	AF ← AF'	•	•	X	•	X	•	•	•	•	•	•	00	001	000	08									
EXX	BC ← BC' DE ← DE' HL ← HL'	•	•	X	•	X	•	•	•	•	•	•	11	011	001	D9									
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									
EX (SP), IY	IYH ← (SP+1) IYL ← (SP)	•	•	X	•	X	•	•	•	•	•	•	11	111	101	FD	2	6	23						
													11	100	011	E3									
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)				
													10	100	000	A0									
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					
													10	110	000	B0							16	If BC = 0	

NOTE: ⊕ P/V flag is 0 if the result of BC-1 = 0, otherwise P/V = 1.

Exchange, Block Transfer, Block Search Groups (Continued)	Mnemonic	Symbolic Operation	Flags				Opcodes 78 543 210 Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
			S	Z	H	P/V N C					
LDD	(DE) - (HL) DE - DE-1 HL - HL-1 BC - BC-1	• • X 0 X 1 0 •	①	11 101 101 ED 10 101 000 A8	2	4	16				
LDDR	(DE) - (HL) DE - DE-1 HL - HL-1 BC - BC-1 Repeat until BC = 0	• • X 0 X 1 0 •	②	11 101 101 ED 10 111 000 B8	2 2	5 4	21 16	If BC ≠ 0 If BC = 0			
CPI	A - (HL) HL - HL+1 BC - BC-1	1 1 X 1 X 1 1 •	③ ①	11 101 101 ED 10 100 001 A1	2	4	16				
CPIR	A - (HL) HL - HL+1 BC - BC-1 Repeat until A = (HL) or BC = 0	1 1 X 1 X 1 1 •	③ ①	11 101 101 ED 10 110 001 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	1 1 X 1 X 1 1 •	③ ①	11 101 101 ED 10 101 001 A9	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 •	③ ①	11 101 101 ED 10 111 001 B9	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)			

NOTES: ① P/V flag is 0 if the result of BC-1 = 0, otherwise P/V = 1.
 ② P/V flag is 0 at completion of instruction only.
 ③ Z flag is 1 if A = (HL), otherwise Z = 0.

8-Bit Arithmetic and Logical Group

ADD A, r	A - A + r	1 1 X 1 X V 0 1	10 000 r	1	1	4	r Reg.
ADD A, n	A - A + n	1 1 X 1 X V 0 1	11 000 110 - n -	2	2	7	000 B 001 C 010 D 011 E 100 H 101 L 111 A
ADD A, (HL)	A - A + (HL)	1 1 X 1 X V 0 1	10 000 110	1	2	7	
ADD A, (IX+d)	A - A + (IX+d)	1 1 X 1 X V 0 1	11 011 101 10 000 110 - d -	DD	3	5	19
ADD A, (IY+d)	A - A + (IY+d)	1 1 X 1 X V 0 1	11 111 101 10 000 110 - d -	FD	3	5	19
ADC A, s	A - A + s + CY	1 1 X 1 X V 0 1	001 010				s is any of r, n, (HL), (IX+d), (IY+d) as shown for ADD instruction. The indicated bits replace the 000 in the ADD set above.
SUB s	A - A - s	1 1 X 1 X V 1 1	011				
SBC A, s	A - A - s - CY	1 1 X 1 X P 0 1	100				
AND s	A - A ∧ s	1 1 X 0 X P 0 0	110				
OR s	A - A ∨ s	1 1 X 0 X P 0 0	101				
XOR s	A - A ⊕ s	1 1 X 0 X P 0 0	111				
CP s	A - s	1 1 X 1 X V 1 1	111				
INC r	r - r + 1	1 1 X 1 X V 0 •	00 r 100	1	1	4	
INC (HL)	(HL) - (HL) + 1	1 1 X 1 X V 0 •	00 110 100	1	3	11	
INC (IX+d)	(IX+d) - (IX+d) + 1	1 1 X 1 X V 0 •	11 011 101 00 110 100 - d -	DD	3	6	23
INC (IY+d)	(IY+d) - (IY+d) + 1	1 1 X 1 X V 0 •	11 111 101 00 110 100 - d -	FD	3	6	23
DEC m	m - m - 1	1 1 X 1 X V 1 •	101				m is any of r, (HL), (IX+d), (IY+d) as shown for INC. DEC same format and states as INC. Replace 100 with 101 in opcode.

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 *	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	$A - 0 - A$	1	1	X	1	X	V	1 1 1	11 101 101 ED 01 000 100 44	2	2	8	Negate acc. (two's complement).
CCF	$CY - \bar{CY}$	*	*	X	X	X	*	0 1	00 111 111 3F	1	1	4	Complement carry flag.
SCF	$CY - 1$	*	*	X	0	X	*	0 1	00 110 111 37	1	1	4	Set carry flag.
NOP	No operation	*	*	X	*	X	*	*	00 000 000 00	1	1	4	
HALT	CPU halted	*	*	X	*	X	*	*	01 110 110 76	1	1	4	
DI *	IFF = 0	*	*	X	*	X	*	*	11 110 011 F3	1	1	4	
EI *	IFF = 1	*	*	X	*	X	*	*	11 111 011 FB	1	1	4	
IM 0	Set interrupt mode 0	*	*	X	*	X	*	*	11 101 101 ED 01 000 110 46	2	2	8	
IM 1	Set interrupt mode 1	*	*	X	*	X	*	*	11 101 101 ED 01 010 110 56	2	2	8	
IM 2	Set interrupt mode 2	*	*	X	*	X	*	*	11 101 101 ED 01 011 110 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 1	00 ss1 001	1	3	11	ss Reg. 00 BC 01 DE 10 HL 11 SP
ADC HL, ss	$HL - HL + ss + CY$	1	1	X	X	X	V	0 1	11 101 101 ED 01 ss1 010	2	4	15	
SBC HL, ss	$HL - HL - ss - CY$	1	1	X	X	X	V	1 1	11 101 101 ED 01 ss0 010	2	4	15	
ADD IX, pp	$IX - IX + pp$	*	*	X	X	X	*	0 1	11 011 101 DD 01 ppl 001	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY, rr	$IY - IY + rr$	*	*	X	X	X	*	0 1	11 111 101 FD 00 rr1 001	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 DD 00 100 011 23	2	2	10	
INC IY	$IY - IY + 1$	*	*	X	*	X	*	*	11 111 101 FD 00 100 011 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 DD 00 101 011 2B	2	2	10	
DEC IY	$IY - IY - 1$	*	*	X	*	X	*	*	11 111 101 FD 00 101 011 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

RLCA		*	*	X	0	X	*	0 1	00 000 111 07	1	1	4	Rotate left circular accumulator.
RLA		*	*	X	0	X	*	0 1	00 010 111 17	1	1	4	Rotate left accumulator.
RRCA		*	*	X	0	X	*	0 1	00 001 111 0F	1	1	4	Rotate right circular accumulator.
RRA		*	*	X	0	X	*	0 1	00 011 111 1F	1	1	4	Rotate right accumulator.
RLC r		1	1	X	0	X	P	0 1	11 001 011 CB 00 000 r	2	2	8	Rotate left circular register r.
RLC (HL)		1	1	X	0	X	P	0 1	11 001 011 CB 00 000 110	2	4	15	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC (IX+d)		1	1	X	0	X	P	0 1	11 011 101 DD 11 001 011 CB - d - 00 000 110	4	6	23	
RLC (IY+d)		1	1	X	0	X	P	0 1	11 111 101 FD 11 001 011 CB - d - 00 000 110	4	6	23	
RL m		1	1	X	0	X	P	0 1	00 000 110 010				Instruction format and states are as shown for RLC's. To form new opcode replace 000 or RLC's with shown code.
RRC m		1	1	X	0	X	P	0 1	001				

Rotate and Shift Group (Continued)

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
RR m	 $m = r.(HL), (IX + d), (IY + d)$	1	1	X	0	X	P 0 1	01					
SLA m	 $m = r.(HL), (IX + d), (IY + d)$	1	1	X	0	X	P 0 1	100					
SRA m	 $m = r.(HL), (IX + d), (IY + d)$	1	1	X	0	X	P 0 1	101					
SRL m	 $m = r.(HL), (IX + d), (IY + d)$	1	1	X	0	X	P 0 1	111					
RLD	 A (HL)	1	1	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	 A (HL)	1	1	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

BIT b, r	$Z - \bar{r}_b$	X	1	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)	$Z - (\overline{HL})_b$	X	1	X	1	X	X 0 *	11 001 011 01 b 110	CB	2	3	12	
BIT b, (IX + d) _b	$Z - (\overline{IX + d})_b$	X	1	X	1	X	X 0 *	11 011 101 11 001 011 - d - 01 b 110	DD CB	4	5	20	101 E 100 H 101 L 111 A
BIT b, (IY + d) _b	$Z - (\overline{IY + d})_b$	X	1	X	1	X	X 0 *	11 111 101 11 001 011 - d - 01 b 110	FD CB	4	5	20	<u>Bit Tested</u> 000 0 001 1 010 2 011 3 100 4 101 5 110 6 111 7
SET b, r	$r_b - 1$	*	*	X	*	X	* * *	11 001 011 b r	CB	2	2	8	
SET b, (HL)	$(HL)_b - 1$	*	*	X	*	X	* * *	11 001 011 b 110	CB	2	4	15	
SET b, (IX + d)	$(IX + d)_b - 1$	*	*	X	*	X	* * *	11 011 101 11 001 011 - d - b 110	DD CB	4	6	23	
SET b, (IY + d)	$(IY + d)_b - 1$	*	*	X	*	X	* * *	11 111 101 11 001 011 - d - b 110	FD CB	4	6	23	
RES b, m	$m_b = 0$ $m = r, (HL), (IX + d), (IY + d)$	*	*	X	*	X	* * *	b					To form new opcode replace [] of SET b, s with [0]. Flags and time states for SET instruction.

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

Jump Group

JP nn	PC - nn	*	*	X	*	X	* * *	11 000 011 - n - - n -	C3	3	3	10	
JP cc, nn	If condition cc is true PC - 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 PC parity odd 101 PE parity even 110 P sign positive 111 M sign negative
JR e	PC - PC + e	*	*	X	*	X	* * *	00 011 000 - e-2 -		2	3	12	
JR C, e	If C = 0, continue If C = 1, PC - PC + e	*	*	X	*	X	* * *	00 111 000 - e-2 -		2	2	7	If condition not met.
JR NC, e	If C = 1, continue If C = 0, PC - PC + e	*	*	X	*	X	* * *	00 110 000 - e-2 -		2	2	7	If condition not met.
JP Z, e	If Z = 0, continue If Z = 1, PC - PC + e	*	*	X	*	X	* * *	00 101 000 - e-2 -		2	2	7	If condition not met.
JR NZ, e	If Z = 1, continue If Z = 0, PC - PC + e	*	*	X	*	X	* * *	00 100 000 - e-2 -		2	2	7	If condition not met.
JP (HL)	PC - HL	*	*	X	*	X	* * *	11 101 001	E9	1	1	4	
JP (IX)	PC - IX	*	*	X	*	X	* * *	11 011 101 11 101 001	DD E9	2	2	8	

Jump Group (Continued)

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
JP (1Y)	PC ← 1Y	•	•	X • X • • • •	11 111 101 FD 11 101 001 E9	2	2	8	
DJNZ, e	B ← B - 1 If B = 0, continue If B ≠ 0, PC ← PC + e	•	•	X • X • • • •	00 010 000 10 — e - 2 —	2	2	8	If B = 0.
						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

CALL nn	(SP - 1) ← PC _H (SP - 2) ← PC _L 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.
						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 continue, otherwise same as RET	•	•	X • X • • • •	11 cc 000	1	1	5	If cc is false.
						1	3	11	If cc is true.
RETI	Return from interrupt	•	•	X • X • • • •	11 101 101 ED 01 001 101 4D	2	4	14	
RETNI ¹	Return from non-maskable interrupt	•	•	X • X • • • •	11 101 101 ED 01 000 101 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: ¹RETNI loads IFF₂ ← IFF₁

Input and Output Group

IN A, (n)	A ← (n)	•	•	X • X • • • •	11 011 011 DB — n —	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	1	1	X 1 X P 0 •	11 101 101 ED 01 r 000	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 ED 10 100 010 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 ED 10 110 010 B2	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 ED 10 101 010 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 ED 10 111 010 BA	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 D3 — n —	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 ED 01 r 001	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 ED 10 100 011 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 ED 10 110 011 B3	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 ED 10 101 011 AB	2	4	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.
 ② Z flag is set upon instruction completion only.

Input and Output Group
(Continued)

Mnemonic	Symbolic Operation	Flags						Opcode				No. of Bytes	No. of M Cycles	No. of T States	Comments		
		S	Z	H	P/V	N	C	76	543	210	Hex						
OTDR	(C) - (HL)	X	1	X	X	X	X	1	X	11	101	101	ED	2	5	21	C to A ₀ - A ₇
	B - B - 1									10	111	011		2	(If B ≠ 0)	4	B to A ₈ - A ₁₅
	HL - HL - 1 Repeat until B = 0															(If B = 0)	16

Summary of Flag Operation

Instruction	D ₇ S	Z	H	P/V	N	D ₀ 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	•	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; RRC m; SLA m; SRA m; SRL m	1	1	X	0	X	P	0	1	Rotate and shift locations.
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; OTD	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	•	
CPI; 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, 1, 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.

Symbolic Notation

Symbol	Operation	Symbol	Operation
S	Sign flag. S = 1 if the MSB of the result is 1.	†	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 indeterminate.
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

A₀-A₁₅. *Address Bus* (output, active High, 3-state). A₀-A₁₅ 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.

D₀-D₇. *Data Bus* (input/output, active High, 3-state). D₀-D₇ 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

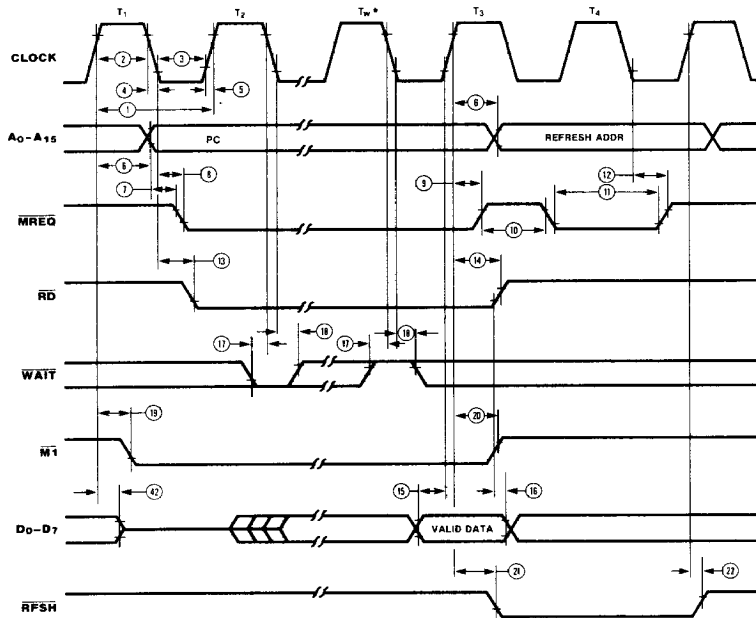
The CPU executes instructions by proceeding through a specific sequence of operations:

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

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 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.

The CPU samples the \overline{WAIT} input with the falling edge of clock state T₂. During clock states T₃ and T₄ of an M1 cycle dynamic RAM refresh can occur while the CPU starts decoding and executing the instruction. When the Refresh Control signal becomes active, refreshing of dynamic memory can take place.



*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/W pulse to most semiconductor memories.

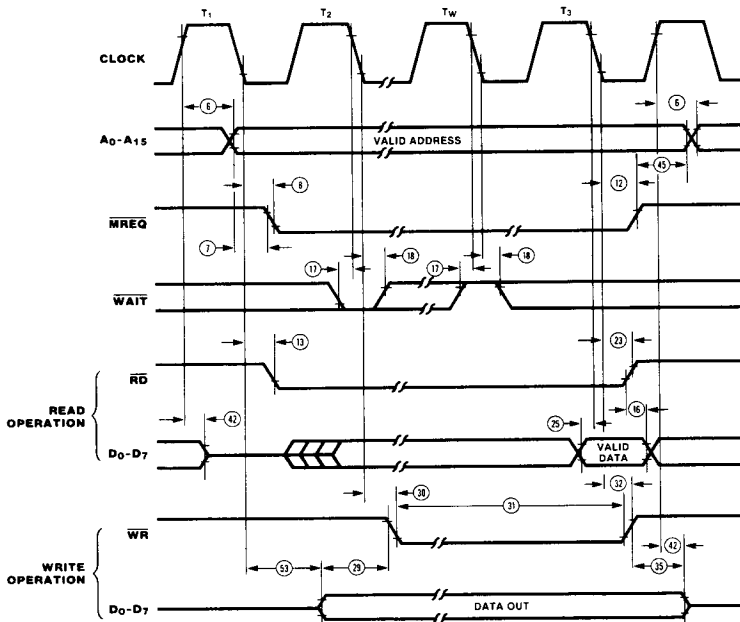


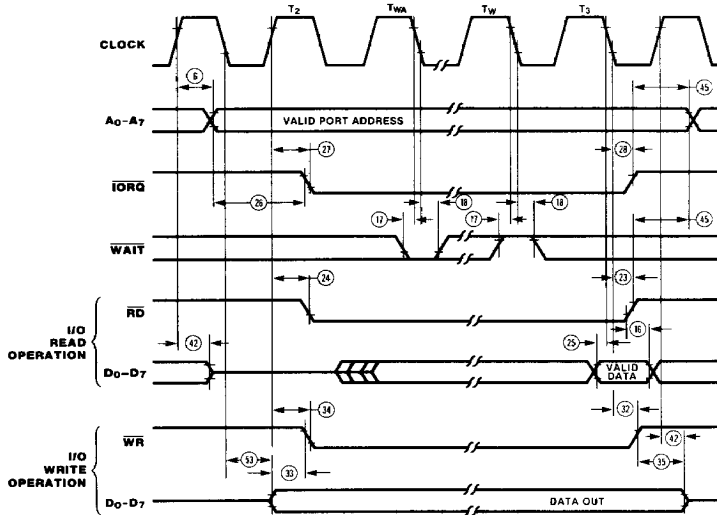
Figure 6. Memory Read or Write Cycles

Z801 CPU

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

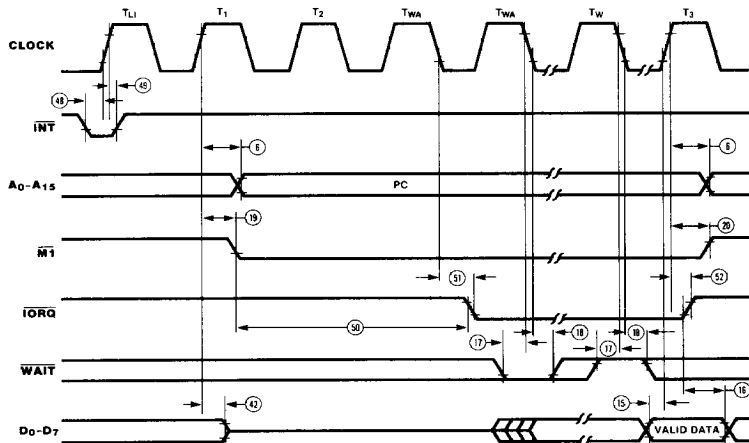


T_{WA} = Wait cycle automatically inserted by CPU.

Figure 7. Input or Output Cycles

Interrupt Request/Acknowledge Cycle. The CPU samples the interrupt signal with the rising edge of the last clock cycle of the last instruction (Figure 8). When an interrupt is accepted, a special $\overline{M1}$ cycle is generated.

During this $\overline{M1}$ cycle, \overline{IORQ} becomes active (instead of \overline{MREQ}) to indicate that the interrupting device can place an 8-bit vector on the data bus. The CPU automatically adds two Wait states to this cycle.



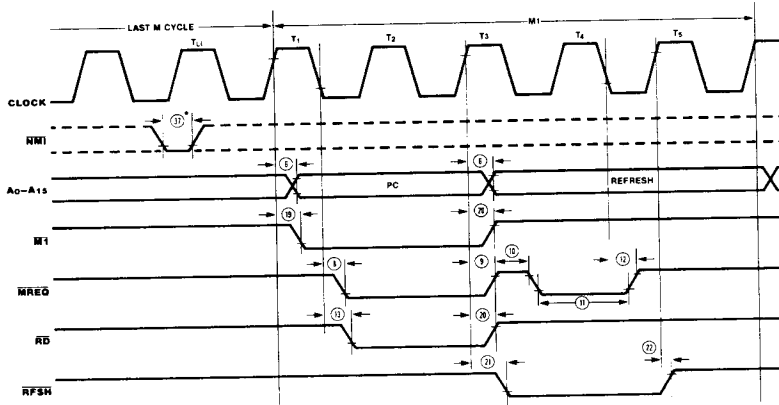
T_{LI} = Last state of any instruction cycle.
 T_{WA} = Wait cycle automatically inserted by CPU.

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 memory read operation except that data put on the bus by the memory is ignored. The CPU instead executes a restart (RST) operation and jumps to the NMI service routine located at address 0066H (Figure 9).

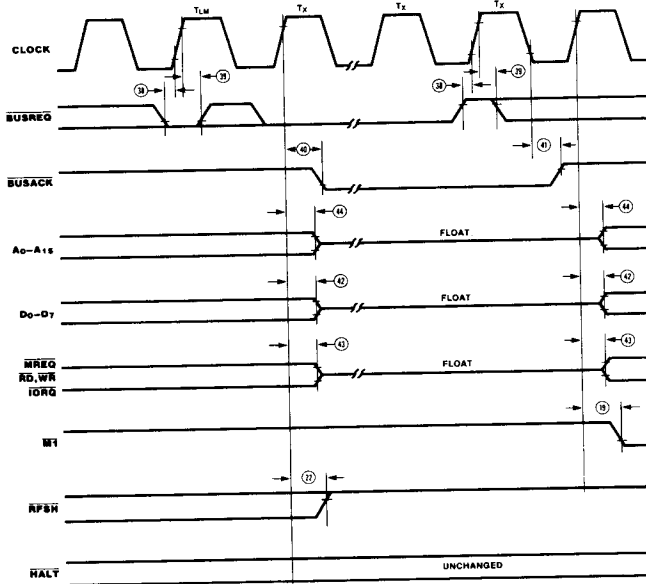


* Although NMI is an asynchronous input, to guarantee its being recognized on the following machine cycle, NMI's falling edge must occur no later than the rising edge of the clock cycle preceding the last state of any instruction cycle (T_{L1}).

Figure 9. Non-Maskable Interrupt Request Operation

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

lines to a high-impedance state with the rising edge of the next clock pulse. At that time, any external device can take control of these lines, usually to transfer data between memory and I/O devices.



NOTES: 1) T_{LM} = Last state of any M cycle.

2) T_X = An arbitrary clock cycle used by requesting device.

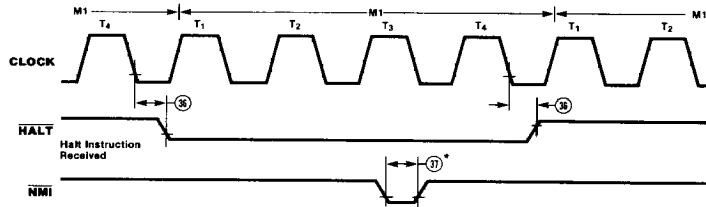
Figure 10. Z-BUS Request/Acknowledge Cycle

Z801 CPU

CPU Timing
(Continued)

Halt Acknowledge Cycle. When the CPU receives a $\overline{\text{HALT}}$ 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 processed (Figure 11). $\overline{\text{INT}}$ will also force a Halt exit.



* 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 the last state of any instruction cycle ($T_{1,T}$).

Figure 11. Halt Acknowledge Cycle

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, two 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).

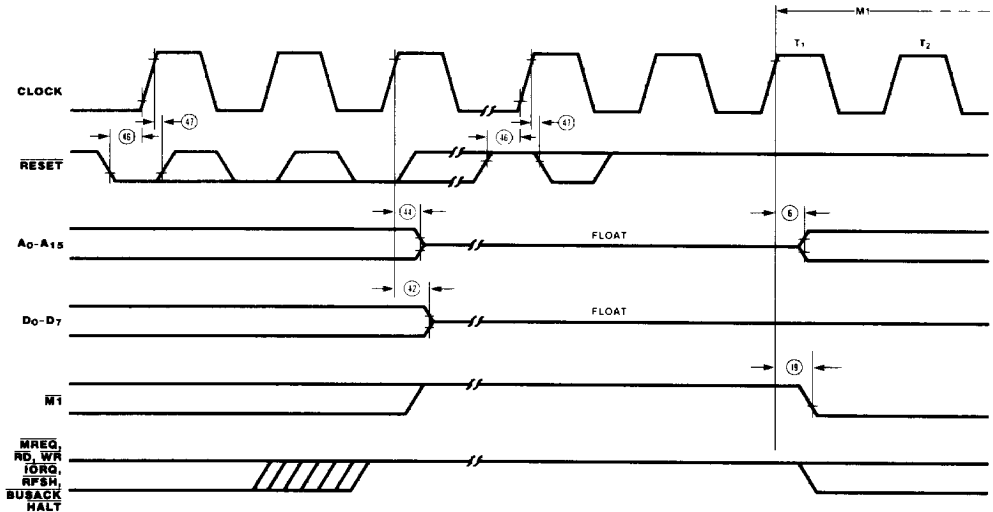


Figure 12. Reset Cycle

AC Characteristics†	Number	Symbol	Parameter	Z8300-1 (1.0 MHz)		Z8300-3 (2.5 MHz)	
				Min (ns)	Max (ns)	Min (ns)	Max (ns)
	1	TcC	Clock Cycle Time	1000*		400*	
	2	TwCh	Clock Pulse Width (High)	470	2000	180	2000
	3	TwCl	Clock Pulse Width (Low)	470	2000	180	2000
	4	TfC	Clock Fall Time	—	30	—	30
	5	TrC	Clock Rise Time	—	30	—	30
	6	TdCr(A)	Clock ↑ to Address Valid Delay	—	380	—	145
	7	TdA(MREQ↑)	Address Valid to $\overline{\text{MREQ}}$ ↓ Delay	370*	—	125*	—
	8	TdCf(MREQf)	Clock ↓ to $\overline{\text{MREQ}}$ ↓ Delay	—	260	—	100
	9	TdCr(MREQr)	Clock ↑ to $\overline{\text{MREQ}}$ ↑ Delay	—	260	—	100
	10	TwMREQh	$\overline{\text{MREQ}}$ Pulse Width (High)	410*	—	170*	—
	11	TwMREQl	$\overline{\text{MREQ}}$ Pulse Width (Low)	890*	—	360*	—
	12	TdCf(MREQr)	Clock ↓ to $\overline{\text{MREQ}}$ ↑ Delay	—	260	—	100
	13	TdCf(RDf)	Clock ↓ to $\overline{\text{RD}}$ ↓ Delay	—	340	—	130
	14	TdCr(RDr)	Clock ↑ to $\overline{\text{RD}}$ ↑ Delay	—	260	—	100
	15	TsD(Cr)	Data Setup Time to Clock ↑	140	—	50	—
	16	ThD(RDr)	Data Hold Time to $\overline{\text{RD}}$ ↑	—	0	—	0
	17	TsWAIT(Cf)	WAIT Setup Time to Clock ↓	190	—	70	—
	18	ThWAIT(Cf)	WAIT Hold Time after Clock ↓	—	0	—	0
	19	TdCr(Mlf)	Clock ↑ to $\overline{\text{M}}$ ↓ Delay	—	340	—	130
	20	TdCr(Mlr)	Clock ↑ to $\overline{\text{M}}$ ↑ Delay	—	340	—	130
	21	TdCr(RFSHf)	Clock ↑ to $\overline{\text{RFSH}}$ ↓ Delay	—	460	—	180
	22	TdCr(RFSHr)	Clock ↑ to $\overline{\text{RFSH}}$ ↑ Delay	—	390	—	150
	23	TdCf(RDr)	Clock ↓ to $\overline{\text{RD}}$ ↓ Delay	—	290	—	110
	24	TdCr(RDf)	Clock ↑ to $\overline{\text{RD}}$ ↓ Delay	—	260	—	100
	25	TsD(Cf)	Data Setup to Clock ↓ during M_2, M_3, M_4 or M_5 Cycles	160	—	60	—
	26	TdA(IRQf)	Address Stable prior to $\overline{\text{IRQ}}$ ↓	790*	—	320*	—
	27	TdCr(IRQf)	Clock ↓ to $\overline{\text{IRQ}}$ ↓ Delay	—	240	—	90
	28	TdCf(IRQr)	Clock ↓ to $\overline{\text{IRQ}}$ ↑ Delay	—	290	—	110
	29	TdD(WRf)	Data Stable prior to $\overline{\text{WR}}$ ↓	470*	—	190*	—
	30	TdCf(WRf)	Clock ↓ to $\overline{\text{WR}}$ ↓ Delay	—	240	—	90
	31	TwWR	$\overline{\text{WR}}$ Pulse Width	890*	—	360*	—
	32	TdCf(WRr)	Clock ↓ to $\overline{\text{WR}}$ ↑ Delay	—	260	—	100
	33	TdD(WRf)	Data Stable prior to $\overline{\text{WR}}$ ↓	-30*	—	30*	—
	34	TdCr(WRf)	Clock ↑ to $\overline{\text{WR}}$ ↓ Delay	—	210	—	80
	35	TdWRr(D)	Data Stable from $\overline{\text{WR}}$ ↑	290*	—	130*	—
	36	TdCf(HALT)	Clock ↓ to $\overline{\text{HALT}}$ ↑ or ↓	—	760	—	300
	37	TwNMI	$\overline{\text{NMI}}$ Pulse Width	210	—	80	—
	38	TsBUSREQ(Cr)	$\overline{\text{BUSREQ}}$ Setup Time to Clock ↑	210	—	80	—

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

†Calculated values above assumed.

TrC = TfC = 20 ns.

†All timings assume equal loading on pins within 50 pf.

AC Characteristics† (Continued)	Number	Symbol	Parameter	Z8300-1		Z8300-3	
				Min (ns)	Max (ns)	Min (ns)	Max (ns)
	39	ThBUSREQ(Cr)	$\overline{\text{BUSREQ}}$ Hold Time after Clock ↑	0	—	0	—
	40	TdCr(BUSACKf)	Clock ↑ to $\overline{\text{BUSACK}}$ ↓ Delay	—	310	—	120
	41	TdCl(BUSACKr)	Clock ↓ to $\overline{\text{BUSACK}}$ ↑ Delay	—	290	—	110
	42	TdCr(Dz)	Clock ↑ to Data Float Delay	—	240	—	90
	43	TdCr(CTz)	Clock ↑ to Control Outputs Float Delay (MREQ, IORQ, RD, and WR)	—	290	—	110
	44	TdCr(Az)	Clock ↑ to Address Float Delay	—	290	—	110
	45	TdCTr(A)	MREQ ↑, IORQ ↑, RD ↑, and WR ↑ to Address Hold Time	400*	—	160*	—
	46	TsRESET(Cr)	$\overline{\text{RESET}}$ to Clock ↑ Setup Time	240	—	90	—
	47	ThRESET(Cr)	$\overline{\text{RESET}}$ to Clock ↑ Hold Time	—	0	—	0
	48	TsINTf(Cr)	$\overline{\text{INT}}$ to Clock ↑ Setup Time	210	—	80	—
	49	ThINTR(Cr)	$\overline{\text{INT}}$ to Clock ↑ Hold Time	—	0	—	0
	50	TdM1H(IORQf)	M1 ↓ to $\overline{\text{IORQ}}$ ↓ Delay	2300*	—	920*	—
	51	TdCf(IORQf)	Clock ↓ to $\overline{\text{IORQ}}$ ↓ Delay	—	290	—	110
	52	TdCf(IORQr)	Clock ↑ to $\overline{\text{IORQ}}$ ↑ Delay	—	260	—	100
	53	TdCf(D)	Clock ↓ to Data Valid Delay	—	290	—	230

*For clock periods other than the minimums shown in the table, calculate parameters using the following expressions. Calculated values above assumed TrC = TtC = 20 ns.
† All timings assume equal loading on pins with 50 pF.

Footnotes to AC Characteristics

Number	Symbol	Z8300-1	Z8300-3
1	TcC	$TwCh + TwCl + TrC + TtC$	$TwCh + TwCl + TrC + TtC$
7	TdA(MREQf)	$TwCh + TtC - 200$	$TwCh + TtC - 75$
10	TwMREQh	$TwCh + TtC - 90$	$TwCh + TtC - 30$
11	TwMREQl	$TcC - 110$	$TcC - 30$
26	TdA(IORQl)	$TcC - 210$	$TcC - 80$
29	TdD(WRf)	$TcC - 540$	$TcC - 210$
31	TwWR	$TcC - 110$	$TcC - 40$
33	TdD(WRf)	$TwCl + TrC - 470$	$TwCl + TrC - 180$
35	TdWRr(D)	$TwCl + TrC - 210$	$TwCl + TrC - 80$
45	TdCTr(A)	$TwCl + TrC - 110$	$TwCl + TrC - 40$
50	TdM1H(IORQf)	$2TcC + TwCh + TtC - 210$	$2TcC + TwCh + TtC - 80$

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 Voltages on all pins with respect to ground. -0.3V to +7V
 Operating Ambient Temperature See Ordering Information
 Storage Temperature. -65°C to +150°C

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

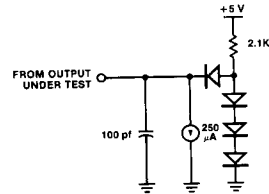
Standard Test Conditions The DC characteristics and capacitance sections listed below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0V). Positive current flows into the referenced pin.

Available operating temperature is:

- S = 0°C to +70°C, +4.75V ≤ V_{CC} ≤ +5.25V

All ac parameters assume a load capacitance of 100 pf. Add 10 ns delay for each 50 pf increase in load up to a maximum of 200 pf for the data bus and 100 pf for address and control lines.

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



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} = 2.0 mA
V _{OH}	Output High Voltage	2.4		V	I _{OH} = -250 μA
I _{LI}	Input Leakage Current		10	μA	V _{IN} = 0 to V _{CC}
I _{LO}	3-State Output Leakage		±10 ¹	μA	V _{OUT} = 0.4 to V _{CC}
I _{CC}	Power Supply Current				

Frequency	Temperature				Unit
	0°C Max	25°C Max	25°C Typical	70°C Max	
Z8300-1 (1.0 MHz)	30	25	15	20	mA
Z8300-3 (2.5 MHz)	45	40	25	35	mA

1. A₁₅-A₀, D₇-D₀, MREQ, IORC, RD, and WR.

Capacitance

Symbol	Parameter	Min	Max	Unit	Note
C _{CLOCK}	Clock Capacitance		35	pF	
C _{IN}	Input Capacitance		5	pF	
C _{OUT}	Output Capacitance		15	pF	

T_A = 25°C, f = 1 MHz.
 Unmeasured pins returned to ground.

ORDERING INFORMATION

Z80L CPU, 1.0 MHz
40-pin DIP
Z8300-1 PS

Z80L CPU, 2.5 MHz
40-pin DIP
Z8300-3 PS

Codes

First letter is for package; second letter is for temperature.

C = Ceramic DIP
P = Plastic DIP
L = Ceramic LCC
V = Plastic PCC

R = Protopack
T = Low Profile Protopack
DIP = Dual-In-Line Package
LCC = Leadless Chip Carrier
PCC = Plastic Chip Carrier (Leaded)

TEMPERATURE

S = 0°C to +70°C
E = -40°C to +85°C
M* = -55°C to +125°C

FLOW

B = 883 Class B

Example: PS is a plastic DIP, 0°C to +70°C.

†Available soon.

*For Military Orders, contact your local Ziilog Sales Office for Military Electrical Specifications.