code	mnemonic		T	S Z AC P C
06	**MVI**	**B,**n	7T	
0E	**MVI**	**C,**n	7T	
16	**MVI**	**D,**n	7T	
1E	**MVI**	**E,**n	7T	
26	**MVI**	**H,**n	7T	
2E	**MVI**	**L,**n	7T	
36	**MVI**	**M,**n	10T	
3E	**MVI**	**A,**n	7T	
40	**MOV**	**B,B**	5T	
41	**MOV**	**B,C**	5T	
42	**MOV**	**B,D**	5T	
43	**MOV**	**B,E**	5T	
44	**MOV**	**B,H**	5T	
45	**MOV**	**B,L**	5T	
46	**MOV**	**B,M**	7T	
47	**MOV**	**B,A**	5T	
48	**MOV**	**C,B**	5T	
49	**MOV**	**C,C**	5T	
4A	**MOV**	**C,D**	5T	
4B	**MOV**	**C,E**	5T	
4C	**MOV**	**C,H**	5T	
4D	**MOV**	**C,L**	5T	
4E	**MOV**	**C,M**	7T	
4F	**MOV**	**C,A**	5T	
0A	**LDAX**	**B**	7T	
1A	**LDAX**	**D**	7T	
2A	**LHLD**	nn	16T	
3A	**LDA**	nn	13T	
02	**STAX**	**B**	7T	
12	**STAX**	**D**	7T	
22	**SHLD**	nn	16T	
32	**STA**	nn	13T	
01	**LXI**	**B,**nn	10T	
11	**LXI**	**D,**nn	10T	
21	**LXI**	**H,**nn	10T	
31	**LXI**	**SP,**nn	10T	
07	**RLC**		4T	X
0F	**RRC**		4T	X
17	**RAL**		4T	X
1F	**RAR**		4T	X
D3	**OUT**	n	10T	
DB	**IN**	n	10T	

code	mnemonic			T	S Z AC P C
78	**MOV**	**A,B**	*move B to A*	5T	
79	**MOV**	**A,C**	*move C to A*	5T	
7A	**MOV**	**A,D**	*move D to A*	5T	
7B	**MOV**	**A,E**	*move E to A*	5T	
7C	**MOV**	**A,H**	*move H to A*	5T	
7D	**MOV**	**A,L**	*move L to A*	5T	
7E	**MOV**	**A,M**	*move memory (HL) to A*	7T	
7F	**MOV**	**A,A**	*move A to A*	5T	
50	**MOV**	**D,B**	*move B to D*	5T	
51	**MOV**	**D,C**	*move C to D*	5T	
52	**MOV**	**D,D**	*move D to D*	5T	
53	**MOV**	**D,E**	*move E to D*	5T	
54	**MOV**	**D,H**	*move H to D*	5T	
55	**MOV**	**D,L**	*move L to D*	5T	
56	**MOV**	**D,M**	*move memory (HL) to D*	7T	
57	**MOV**	**D,A**	*move A to D*	5T	
58	**MOV**	**E,B**	*move B to E*	5T	
59	**MOV**	**E,C**	*move C to E*	5T	
5A	**MOV**	**E,D**	*move D to E*	5T	
5B	**MOV**	**E,E**	*move E to E*	5T	
5C	**MOV**	**E,H**	*move H to E*	5T	
5D	**MOV**	**E,L**	*move L to E*	5T	
5E	**MOV**	**E,M**	*move memory (HL) to E*	7T	
5F	**MOV**	**E,A**	*move A to E*	5T	
80	**ADD**	**B**	*add B to A*	4T	X X X X X
81	**ADD**	**C**	*add C to A*	4T	X X X X X
82	**ADD**	**D**	*add D to A*	4T	X X X X X
83	**ADD**	**E**	*add E to A*	4T	X X X X X
84	**ADD**	**H**	*add H to A*	4T	X X X X X
85	**ADD**	**L**	*add L to A*	4T	X X X X X
86	**ADD**	**M**	*add memory (HL) to A*	7T	X X X X X
87	**ADD**	**A**	*add A to A*	4T	X X X X X
C6	**ADI**	n	*add value to A*	7T	X X X X X
09	**DAD**	**B**	*add BC to HL*	10T	X
19	**DAD**	**D**	*add DE to HL*	10T	X
29	**DAD**	**H**	*add HL to HL*	10T	X
39	**DAD**	**SP**	*add stack pointer to HL*	10T	X
27	**DAA**		*decimal adjust*	4T	X X X X X
2F	**CMA**		*complement A*	4T	
37	**STC**		*set carry flag*	4T	1
3F	**CMC**		*complement carry flag*	4T	X
E3	**XTHL**		*exchange top address in stack with HL*	18T	
EB	**XCHG**		*exchange HL with DE*	4T	

code	mnemonic			T	S Z AC P C
70	**MOV**	**M,B**	*store B to memory (HL)*	7T	
71	**MOV**	**M,C**	*store C to memory (HL)*	7T	
72	**MOV**	**M,D**	*store D to memory (HL)*	7T	
73	**MOV**	**M,E**	*store E to memory (HL)*	7T	
74	**MOV**	**M,H**	*store H to memory (HL)*	7T	
75	**MOV**	**M,L**	*store L to memory (HL)*	7T	
77	**MOV**	**M,A**	*store A to memory (HL)*	7T	
60	**MOV**	**H,B**	*move B to H*	5T	
61	**MOV**	**H,C**	*move C to H*	5T	
62	**MOV**	**H,D**	*move D to H*	5T	
63	**MOV**	**H,E**	*move E to H*	5T	
64	**MOV**	**H,H**	*move H to H*	5T	
65	**MOV**	**H,L**	*move L to H*	5T	
66	**MOV**	**H,M**	*move memory (HL) to H*	7T	
67	**MOV**	**H,A**	*move A to H*	5T	
68	**MOV**	**L,B**	*move B to L*	5T	
69	**MOV**	**L,C**	*move C to L*	5T	
6A	**MOV**	**L,D**	*move D to L*	5T	
6B	**MOV**	**L,E**	*move E to L*	5T	
6C	**MOV**	**L,H**	*move H to L*	5T	
6D	**MOV**	**L,L**	*move L to L*	5T	
6E	**MOV**	**L,M**	*move memory (HL) to L*	7T	
6F	**MOV**	**L,A**	*move A to L*	5T	
88	**ADC**	**B**	*add B to A with carry*	4T	X X X X X
89	**ADC**	**C**	*add C to A with carry*	4T	X X X X X
8A	**ADC**	**D**	*add D to A with carry*	4T	X X X X X
8B	**ADC**	**E**	*add E to A with carry*	4T	X X X X X
8C	**ADC**	**H**	*add H to A with carry*	4T	X X X X X
8D	**ADC**	**L**	*add L to A with carry*	4T	X X X X X
8E	**ADC**	**M**	*add memory (HL) to A with carry*	7T	X X X X X
8F	**ADC**	**A**	*add A to A with carry*	4T	X X X X X
CE	**ACI**	n	*add value to A with carry*	7T	X X X X X
C1	**POP**	**B**	*pop BC from stack*	10T	
D1	**POP**	**D**	*pop DE from stack*	10T	
E1	**POP**	**H**	*pop HL from stack*	10T	
F1	**POP**	**PSW**	*pop PSW (AF) from stack*	10T	X X X X X
C5	**PUSH**	**B**	*push BC to stack*	11T	
D5	**PUSH**	**D**	*push DE to stack*	11T	
E5	**PUSH**	**H**	*push HL to stack*	11T	
F5	**PUSH**	**PSW**	*push PSW (AF) to stack*	11T	
E9	**PCHL**		*jump to address in HL*	5T	
F9	**SPHL**		*load HL to stack pointer*	5T	

code	mnemonic		T	S Z AC P C
04	**INR B**	*increment B*	5T	X X X X
0C	**INR C**	*increment C*	5T	X X X X
14	**INR D**	*increment D*	5T	X X X X
1C	**INR E**	*increment E*	5T	X X X X
24	**INR H**	*increment H*	5T	X X X X
2C	**INR L**	*increment L*	5T	X X X X
34	**INR M**	*increment memory (HL)*	10T	X X X X
3C	**INR A**	*increment A*	5T	X X X X
A0	**ANA B**	*and A with B*	4T	X X X X 0
A1	**ANA C**	*and A with C*	4T	X X X X 0
A2	**ANA D**	*and A with D*	4T	X X X X 0
A3	**ANA E**	*and A with E*	4T	X X X X 0
A4	**ANA H**	*and A with H*	4T	X X X X 0
A5	**ANA L**	*and A with L*	4T	X X X X 0
A6	**ANA M**	*and A with memory (HL)*	7T	X X X X 0
A7	**ANA A**	*and A with A*	4T	X X X X 0
E6	**ANI** n	*and A with value*	7T	X X X X 0
B8	**CMP B**	*compare A with B*	4T	X X X X X
B9	**CMP C**	*compare A with C*	4T	X X X X X
BA	**CMP D**	*compare A with D*	4T	X X X X X
BB	**CMP E**	*compare A with E*	4T	X X X X X
BC	**CMP H**	*compare A with H*	4T	X X X X X
BD	**CMP L**	*compare A with L*	4T	X X X X X
BE	**CMP M**	*compare A with memory (HL)*	7T	X X X X X
BF	**CMP A**	*compare A with A*	4T	X X X X X
FE	**CPI** n	*compare A with value*	7T	X X X X X
C2	**JNZ** nn	*conditional jump (not zero flag)*	10T	
CA	**JZ** nn	*conditional jump (zero flag)*	10T	
D2	**JNC** nn	*conditional jump (not carry flag)*	10T	
DA	**JC** nn	*conditional jump (carry flag)*	10T	
E2	**JPO** nn	*conditional jump (parity odd flag)*	10T	
EA	**JPE** nn	*conditional jump (parity even flag)*	10T	
F2	**JP** nn	*conditional jump (positive flag)*	10T	
FA	**JM** nn	*conditional jump (minus flag)*	10T	
C3	**JMP** nn	*jump to direct address*	10T	
00	**NOP**	*no operation*	4T	
76	**HLT**	*halt*		
F3	**DI**	*disable interrupt*	4T	
FB	**EI**	*enable interrupt*	4T	

code	mnemonic		T	S Z AC P C
05	**DCR B**	*decrement B*	5T	X X X X
0D	**DCR C**	*decrement C*	5T	X X X X
15	**DCR D**	*decrement D*	5T	X X X X
1D	**DCR E**	*decrement E*	5T	X X X X
25	**DCR H**	*decrement H*	5T	X X X X
2D	**DCR L**	*decrement L*	5T	X X X X
35	**DCR M**	*decrement memory (HL)*	10T	X X X X
3D	**DCR A**	*decrement A*	5T	X X X X
B0	**ORA B**	*or A with B*	4T	X X 0 X 0
B1	**ORA C**	*or A with C*	4T	X X 0 X 0
B2	**ORA D**	*or A with D*	4T	X X 0 X 0
B3	**ORA E**	*or A with E*	4T	X X 0 X 0
B4	**ORA H**	*or A with H*	4T	X X 0 X 0
B5	**ORA L**	*or A with L*	4T	X X 0 X 0
B6	**ORA M**	*or A with memory (HL)*	7T	X X 0 X 0
B7	**ORA A**	*or A with A*	4T	X X 0 X 0
F6	**ORI** n	*or A with value*	7T	X X 0 X 0
90	**SUB B**	*subtract B from A*	4T	X X X X X
91	**SUB C**	*subtract C from A*	4T	X X X X X
92	**SUB D**	*subtract D from A*	4T	X X X X X
93	**SUB E**	*subtract E from A*	4T	X X X X X
94	**SUB H**	*subtract H from A*	4T	X X X X X
95	**SUB L**	*subtract L from A*	4T	X X X X X
96	**SUB M**	*subtract memory (HL) from A*	7T	X X X X X
97	**SUB A**	*subtract A from A*	4T	X X X X X
D6	**SUI** n	*subtract value from A*	7T	X X X X X
C4	**CNZ** nn	*conditional call (not zero flag)*	11/17T	
CC	**CZ** nn	*conditional call (zero flag)*	11/17T	
D4	**CNC** nn	*conditional call (not carry flag)*	11/17T	
DC	**CC** nn	*conditional call (carry flag)*	11/17T	
E4	**CPO** nn	*conditional call (parity odd flag)*	11/17T	
EC	**CPE** nn	*conditional call (parity even flag)*	11/17T	
F4	**CP** nn	*conditional call (positive flag)*	11/17T	
FC	**CM** nn	*conditional call (minus flag)*	11/17T	
CD	**CALL** nn	*call to direct address*	17T	

code	mnemonic		T	S Z AC P C
03	**INX B**	*increment BC*	5T	
13	**INX D**	*increment DE*	5T	
23	**INX H**	*increment HL*	5T	
33	**INX SP**	*increment stack pointer*	5T	
0B	**DCX B**	*decrement BC*	5T	
1B	**DCX D**	*decrement DE*	5T	
2B	**DCX H**	*decrement HL*	5T	
3B	**DCX SP**	*decrement stack pointer*	5T	
A8	**XRA B**	*xor A with B*	4T	X X 0 X 0
A9	**XRA C**	*xor A with C*	4T	X X 0 X 0
AA	**XRA D**	*xor A with D*	4T	X X 0 X 0
AB	**XRA E**	*xor A with E*	4T	X X 0 X 0
AC	**XRA H**	*xor A with H*	4T	X X 0 X 0
AD	**XRA L**	*xor A with L*	4T	X X 0 X 0
AE	**XRA M**	*xor A with memory (HL)*	7T	X X 0 X 0
AF	**XRA A**	*xor A with A*	4T	X X 0 X 0
EE	**XRI** n	*xor A with value*	7T	X X 0 X 0
98	**SBB B**	*subtract B from A with carry*	4T	X X X X X
99	**SBB C**	*subtract C from A with carry*	4T	X X X X X
9A	**SBB D**	*subtract D from A with carry*	4T	X X X X X
9B	**SBB E**	*subtract E from A with carry*	4T	X X X X X
9C	**SBB H**	*subtract H from A with carry*	4T	X X X X X
9D	**SBB L**	*subtract L from A with carry*	4T	X X X X X
9E	**SBB M**	*subtract memory (HL) from A with carry*	7T	X X X X X
9F	**SBB A**	*subtract A from A with carry*	4T	X X X X X
DE	**SBI** n	*subtract value from A with carry*	7T	X X X X X
C0	**RNZ**	*conditional return (not zero flag)*	5/11T	
C8	**RZ**	*conditional return (zero flag)*	5/11T	
D0	**RNC**	*conditional return (not carry flag)*	5/11T	
D8	**RC**	*conditional return (carry flag)*	5/11T	
E0	**RPO**	*conditional return (parity odd flag)*	5/11T	
E8	**RPE**	*conditional return (parity even flag)*	5/11T	
F0	**RP**	*conditional return (positive flag)*	5/11T	
F8	**RM**	*conditional return (minus flag)*	5/11T	
C9	**RET**	*return*	10T	
C7	**RST 0**	*restart to 0000h*	11T	
CF	**RST 1**	*restart to 0008h*	11T	
D7	**RST 2**	*restart to 0010h*	11T	
DF	**RST 3**	*restart to 0018h*	11T	
E7	**RST 4**	*restart to 0020h*	11T	
EF	**RST 5**	*restart to 0028h*	11T	
F7	**RST 6**	*restart to 0030h*	11T	
FF	**RST 7**	*restart to 0038h*	11T	