APPENDIX B

8085 Instruction Set

Instructions	Op Code	*T* states	Flags	Main Effects
ACI byte	CE	7	ALL	A ← A + CY + byte
ADC A	8F	4	ALL	A ← A + A + CY
ADC B	88	4	ALL	A ← A + B + CY
ADC C	89	4	ALL	A ← A + C + CY
ADC D	8A	4	ALL	A ← A + D + CY
ADC E	8B	4	ALL	A ← A + E + CY
ADC H	8C	4	ALL	A ← A + H + CY
ADC L	8D	4	ALL	A ← A + L + CY
ADC M	8E	7	ALL	A ← A + M_{HL} + CY
ADD A	87	4	ALL	A ← A + A
ADD B	80	4	ALL	A ← A + B
ADD C	81	4	ALL	A ← A + C
ADD D	82	4	ALL	A ← A + D
ADD E	83	4	ALL	A ← A + E
ADD H	84	4	ALL	A ← A + H
ADD L	85	4	ALL	A ← A + L
ADD M	86	7	ALL	A ← A+ M_{HL}
ADI byte	C6	7	ALL	A ← A + byte
ANA A	A7	4	ALL	A ← A AND A
ANA B	A0	4	ALL	A ← A AND B
ANA C	A1	4	ALL	A ← A AND C
ANA D	A2	4	ALL	A ← A AND D
ANA E	A3	4	ALL	A ← A AND E
ANA H	A4	4	ALL	A ← A AND H
ANA L	A5	4	ALL	A ← A AND L
ANA M	A6	7	ALL	A ← A AND M_{HL}
ANI byte	E6	7	ALL	A ← A AND byte
CALL address	CD	18	NONE	PC ← address
CC address	DC	18/9	NONE	PC ← address if CY=1
CM address	FC	18/9	NONE	PC ← address IF S=1
CMA	2F	4	NONE	A ← $\overline{A}$
CMC	3F	4	CY	CY ← $\overline{CY}$
CMP A	BF	4	ALL	Z ← 1 if A = A
CMP B	B8	4	ALL	Z ← 1 if A = B
CMP C	B9	4	ALL	Z ← 1 if A = C
CMP D	BA	4	ALL	Z ← 1 if A = D

Instructions	Op Code	*T* states	Flags	Main Effects
CMP E	BB	4	ALL	Z←1 if A = E
CMP H	BC	4	ALL	Z←1 if A = H
CMP L	BD	4	ALL	Z←1 if A = L
CMP M	BE	7	ALL	Z←1 if A = M_{HL}
CNC address	D4	18/9	NONE	PC ← address if CY = 0
CNZ address	C4	18/9	NONE	PC ←address if Z = 0
CP address	F4	18/9	NONE	PC ←address if S = 0
CPE address	EC	18/9	NONE	PC ←address if P = 1
CPI byte	FE	7	ALL	Z ←1 if A = byte
CPO address	E4	18/9	NONE	PC ←address if P = 0
CZ address	CC	18/9	NONE	PC ← address if Z = 1
DAA	27	4	ALL	A ←BCD number
DAD B	09	10	CY	HL ← HL + BC
DAD D	19	10	CY	HL ← HL + DE
DAD H	29	10	CY	HL ← HL + HL
DAD SP	39	10	CY	HL ← HL + SP
DCR A	3D	4	ALL BUT CY	A ← A - 1
DCR B	05	4	ALL BUT CY	B ← B - 1
DCR C	0D	4	ALL BUT CY	C ← C - 1
DCR D	15	4	ALL BUT CY	D ← D - 1
DCR E	1D	4	ALL BUT CY	E ← E - 1
DCR H	25	4	ALL BUT CY	H ← H - 1
DCR L	2D	4	ALL BUT CY	L ← L - 1
DCR M	35	10	ALL BUT CY	M_{HL} ← M_{HL} - 1
DCX B	0B	6	NONE	BC ← BC - 1
DCX D	1B	6	NONE	DE ← DE - 1
DCX H	2B	6	NONE	HL ← HL - 1
DCX SP	3B	6	NONE	SP ← SP - 1
DI	F3	4	NONE	Disable interrupts
EI	FB	4	NONE	Enable interrupts
HLT	76	5	NONE	Stop processing
IN byte	DB	10	NONE	A ← byte
INR A	3C	4	ALL BUT CY	A ← A + 1
INR B	04	4	ALL BUT CY	B ← B + 1
INR C	0C	4	ALL BUT CY	C ← C + 1
INR D	14	4	ALL BUT CY	D ← D + 1
INR E	1C	4	ALL BUT CY	E ← E + 1
INR H	24	4	ALL BUT CY	H ← H + 1
INR L	2C	4	ALL BUT CY	L ← L + 1
INR M	34	10	ALL BUT CY	M_{HL} ← M_{HL} + 1
INX B	03	6	NONE	BC ← BC + 1
INX D	13	6	NONE	DE ← DE + 1

Instructions	Op Code	*T* states	Flags	Main Effects
INX H	23	6	NONE	HL ← HL + 1
INX SP	33	6	NONE	SP ← SP + 1
JC address	DA	10/7	NONE	PC ← address if CY = 1
JM address	FA	10/7	NONE	PC ← address if S = 1
JMP address	C3	10	NONE	PC ← address
JNC address	D2	10/7	NONE	PC ← address if CY = 0
JNZ address	C2	10/7	NONE	PC ← address if Z = 0
JP address	F2	10/7	NONE	PC ← address if S = 0
JPE address	EA	10/7	NONE	PC ← address if P = 1
JPO address	E2	10/7	NONE	PC ← address if P = 0
JZ address	CA	10/7	NONE	PC ← address if Z = 1
LDA address	3A	13	NONE	A ← M_{adr}
LDAX B	0A	7	NONE	A ← M_{BC}
LDAX D	1A	7	NONE	A ← M_{DE}
LHLD address	2A	16	NONE	L ← M_{adr} , H ← M_{adr+1}
LXI B,dble	01	10	NONE	BC ← dble
LXI D,dble	11	10	NONE	DE ← dble
LXI H,dble	21	10	NONE	HL ← dble
LXI SP,dble	31	10	NONE	SP ← dble
MOV A,A	7F	4	NONE	A ← A
MOV A,B	78	4	NONE	A ← B
MOV A,C	79	4	NONE	A ← C
MOV A,D	7A	4	NONE	A ← D
MOV A,E	7B	4	NONE	A ← E
MOV A,H	7C	4	NONE	A ← H
MOV A,L	7D	4	NONE	A ← L
MOV A,M	7E	7	NONE	A ← M_{HL}
MOV B,A	47	4	NONE	B ← A
MOV B,B	40	4	NONE	B ← B
MOV B,C	41	4	NONE	B ← C
MOV B,D	42	4	NONE	B ← D
MOV B,E	43	4	NONE	B ← E
MOV B,H	44	4	NONE	B ← H
MOV B,L	45	4	NONE	B ← L
MOV B,M	46	7	NONE	B ← M_{HL}
MOV C,A	4F	4	NONE	C ← A
MOV C,B	48	4	NONE	C ← B
MOV C,C	49	4	NONE	C ← C
MOV C,D	4A	4	NONE	C ← D
MOV C,E	4B	4	NONE	C ← E
MOV C,H	4C	4	NONE	C ← H
MOV C,L	4D	4	NONE	C ← L

Instructions	Op Code	*T* states	Flags	Main Effects
MOV C,M	4E	7	NONE	C ← M_{HL}
MOV D,A	57	4	NONE	D ← A
MOV D,B	50	4	NONE	D ← B
MOV D,C	51	4	NONE	D ← C
MOV D,D	52	4	NONE	D ← D
MOV D,E	53	4	NONE	D ← E
MOV D,H	54	4	NONE	D ← H
MOV D,L	55	4	NONE	D ← L
MOV D,M	56	7	NONE	D ← M_{HL}
MOV E,A	5F	4	NONE	E ← A
MOV E,B	58	4	NONE	E ← B
MOV E,C	59	4	NONE	E ← C
MOV E,D	5A	4	NONE	E ← D
MOV E,E	5B	4	NONE	E ← E
MOV E,H	5C	4	NONE	E ← H
MOV E,L	5D	4	NONE	E ← L
MOV E,M	5E	7	NONE	E ← M_{HL}
MOV H,A	67	4	NONE	H ← A
MOV H,B	60	4	NONE	H ← B
MOV H,C	61	4	NONE	H ← C
MOV H,D	62	4	NONE	H ← D
MOV H,E	63	4	NONE	H ← E
MOV H,H	64	4	NONE	H ← H
MOV H,L	65	4	NONE	H ← L
MOV H,M	66	7	NONE	H ← M_{HL}
MOV L,A	6F	4	NONE	L ← A
MOV L,B	68	4	NONE	L ← B
MOV L,C	69	4	NONE	L ← C
MOV L,D	6A	4	NONE	L ← D
MOV L,E	6B	4	NONE	L ← E
MOV L,H	6C	4	NONE	L ← H
MOV L,L	6D	4	NONE	L ← L
MOV L,M	6E	7	NONE	L ← M_{HL}
MOV M,A	77	7	NONE	M_{HL} ← A
MOV M,B	70	7	NONE	M_{HL} ← B
MOV M,C	71	7	NONE	M_{HL} ← C
MOV M,D	72	7	NONE	M_{HL} ← D
MOV M,E	73	7	NONE	M_{HL} ← E
MOV M,H	74	7	NONE	M_{HL} ← H
MOV M,L	75	7	NONE	M_{HL} ← L
MVI A, byte	3E	7	NONE	A ← byte
MVI B, byte	06	7	NONE	B ← byte

Instructions	Op Code	*T* states	Flags	Main Effects
MVI C, byte	0E	7	NONE	C ← byte
MVI D, byte	16	7	NONE	D ← byte
MVI E, byte	1E	7	NONE	E ← byte
MVI H, byte	26	7	NONE	H ← byte
MVI L, byte	2E	7	NONE	L ← byte
MVI M, byte	36	10	NONE	M_{HL} ← byte
NOP	00	4	NONE	Delay
ORA A	B7	4	ALL	A ←A OR A
ORA B	B0	4	ALL	A ←A OR B
ORA C	B1	4	ALL	A ←A OR C
ORA D	B2	4	ALL	A ←A OR D
ORA E	B3	4	ALL	A ←AOR E
ORA H	B4	4	ALL	A ←A OR H
ORA L	B5	4	ALL	A ←A OR L
ORA M	B6	7	ALL	A ←A OR M_{HL}
ORI byte	F6	7	ALL	A ←A OR byte
OUT byte	D3	10	NONE	Port byte ← A
PCHL	E9	6	NONE	PC ← HL
POP B	C1	10	NONE	B ←M_{stk}
POP D	D1	10	NONE	D ← M_{stk}
POP H	E1	10	NONE	H ← M_{stk}
POP PSW	F1	10	NONE	F ← M_{stk}, A ← M_{stk} – 1
PUSH B	C5	12	NONE	M_{stk} – 1 ← B, M_{stk} – 2 ←C
PUSH D	D5	12	NONE	M_{stk} – 1 ← D, M_{stk} – 2 ←E
PUSH H	E5	12	NONE	M_{stk} – 1 ← H, M_{stk} – 2 ←L
PUSH PSW	F5	12	NONE	M_{stk} – 1 ← A, M_{stk} – 2 ←F
RAL	17	4	CY	Rotate all left
RAR	1F	4	CY	Rotate all right
RC	D8	12/6	NONE	PC ←return address if CY = 1
RET	C9	10	NONE	PC ←return address
RIM	20	4	NONE	A ← I
RLC	07	4	CY	Rotate left with carry
RM	F8	12/6	NONE	PC ← return address if S = 1
RNC	D0	12/6	NONE	PC ← return address if CY = 0
RNZ	C0	12/6	NONE	PC ← return address if Z = 0
RP	F0	12/6	NONE	PC ← return address if S = 0
RPE	E8	12/6	NONE	PC ← return address if P = 1
RPO	E0	12/6	NONE	PC ←return address if P = 0
RRC	0F	4	CY	Rotate right with carry
RST O	C7	12	NONE	PC ←0000H
RST 1	CF	12	NONE	PC ←0008H
RST 2	D7	12	NONE	PC ←0010H

Instructions	Op Code	*T* states	Flags	Main Effects
RST 3	DF	12	NONE	PC ← 0018H
RST 4	E7	12	NONE	PC ← 0020H
RST 5	EF	12	NONE	PC ← 0028H
RST 6	F7	12	NONE	PC ← 0030H
RST 7	FF	12	NONE	PC ← 0038H
RZ	C8	12/6	NONE	PC ← return address if Z = 1
SBB A	9F	4	ALL	A ← A – A – CY
SBB B	98	4	ALL	A ← A – B – CY
SBB C	99	4	ALL	A ← A – C – CY
SBB D	9A	4	ALL	A ← A – D – CY
SBB E	9B	4	ALL	A ← A – E – CY
SBB H	9C	4	ALL	A ← A – H – CY
SBB L	9D	4	ALL	A ← A – L – CY
SBB M	9E	7	ALL	A ← A – M – CY
SBI byte	DE	7	ALL	A ← A – byte – CY
SHLD	22	16	NONE	M_{adr+1} ← H, M_{adr} ← L
SIM	30	4	NONE	I ← A
SPHL address	F9	6	NONE	SP ← HL
STA address	32	13	NONE	M_{adr} ← A
STAX B	02	7	NONE	M_{BC} ← A
STAX D	12	7	NONE	M_{DE} ← A
STC	37	4	CY	CY ← 1
SUB A	97	4	ALL	A ← A – A
SUB B	90	4	ALL	A ← A – B
SUB C	91	4	ALL	A ← A – C
SUB D	92	4	ALL	A ← A – D
SUB E	93	4	ALL	A ← A – E
SUB H	94	4	ALL	A ← A – H
SUB L	95	4	ALL	A ← A – L
SUB M	96	7	ALL	A ← A – M
SUI byte	D6	7	ALL	A ← A – byte
XCHG	EB	4	NONE	HL ↔ DE
XRA A	AF	4	ALL	A ← A XOR A
XRA B	A8	4	ALL	A ← A XOR B
XRA C	A9	4	ALL	A ← A XOR C
XRA D	AA	4	ALL	A ← A XOR D
XRA E	AB	4	ALL	A ← A XOR E
XRA H	AC	4	ALL	A ← A XOR H
XRA L	AD	4	ALL	A ← A XOR L
XRA M	AE	7	ALL	A ← A XOR M
XRI byte	EE	7	ALL	A ← A XOR byte
XTHL	E3	16	NONE	HL ↔stack