Appendix A: 8085 Instruction Set by Opcode

The information in this appendix is reproduced by kind permission of the Intel Corporation. The symbols and abbreviations used are listed below.

Symbol	Meaning
A	Accumulator
B, C, D E, H, L	One of the internal registers
F	Represents the flag register
M	The 16-bit memory address currently held by the register pair H and L
byte	An 8-bit data quantity
dble	A 16-bit (two byte) data quantity
addr	A 16-bit address
port	An 8-bit I/O port address
r, r1, r2	One of the registers A, B, C, D, E, H, L
rp	One of the following register pairs B represents the register pair B and C D represents the register pair D and E H represents the register pair H and L PSW represents the register pair A and F SP represents the 16-bit stack pointer
PC	The 16-bit program counter
CY	Carry flag
P	Parity flag
AC	Auxiliary carry flag
Z	Zero flag
S	Sign flag

Data Transfer Group

These instructions transfer data between registers and memory.
Flags - none affected by instructions in this group.

Move

MOV A,A	7F	MOV B,A	47	MOV C,A	4F
MOV A,B	78	MOV B,B	40	MOV C,B	48
MOV A,C	79	MOV B,C	41	MOV C,C	49
MOV A,D	7A	MOV B,D	42	MOV C,D	4A
MOV A,E	7B	MOV B,E	43	MOV C,E	4B
MOV A,H	7C	MOV B,H	44	MOV C,H	4C
MOV A,L	7D	MOV B,L	45	MOV C,L	4D
MOV A,M	7E	MOV B,M	46	MOV C,M	4E
MOV D,A	57	MOV E,A	5F	MOV H,A	67
MOV D,B	50	MOV E,B	58	MOV H,B	60
MOV D,C	51	MOV E,C	59	MOV H,C	61
MOV D,D	52	MOV E,D	5A	MOV H,D	62
MOV D,E	53	MOV E,E	5B	MOV H,E	63
MOV D,H	54	MOV E,H	5C	MOV H,H	64
MOV D,L	55	MOV E,L	5D	MOV H,L	65
MOV D,M	56	MOV E,M	5E	MOV H,M	66

Move Immediate

MOV L,A	6F	MOV M,A	77	MVI A,byte	3E
MOV L,B	68	MOV M,B	70	MVI B,byte	06
MOV L,C	69	MOV M,C	71	MVI C,byte	0E
MOV L,D	6A	MOV M,D	72	MVI D,byte	16
MOV L,E	6B	MOV M,E	73	MVI E,byte	1E
MOV L,H	6C	MOV M,H	74	MVI H,byte	26
MOV L,L	6D	MOV M,L	75	MVI L,byte	2E
MOV L,M	6E			MVI M,byte	36

Load Immediate (Reg. pair)

LXI B,dble	01
LXI D,dble	11
LXI H,dble	21
LXI SP,dble	31

Load/Store A direct

LDAX B	0A
LDAX D	1A
STAX B	02
STAX D	12

Load/Store A direct

LDA addr	3A
STA addr	32

Load/Store HL direct

LHLD addr	2A
SHLD addr	22

Exchange HL with DE

XCHG	EB

Data Manipulation Group – Arithmetic

Instructions in this group perform arithmetic operations on data in the registers and the memory.

Add*

ADD		ADC	
A	87	A	8F
B	80	B	88
C	81	C	89
D	82	D	8A
E	83	E	8B
H	84	H	8C
L	85	L	8D
M	86	M	8E

Subtract*

SUB		SBB	
A	97	A	9F
B	90	B	98
C	91	C	99
D	92	D	9A
E	93	E	9B
H	94	H	9C
L	95	L	9D
M	96	M	9E

Add/Subtract Immediate*

ADI byte	C6
ACI byte	CE
SUI byte	D6
SBI byte	DE

Double Length Add***

DAD	
B	09
D	19
H	29
SP	39

Increment/Decrement**

INR		DCR	
A	3C	A	3D
B	04	B	05
C	0C	C	0D
D	14	D	15
E	1C	E	1D
H	24	H	25
L	2C	L	2D
M	34	M	35

Increment/Decrement Register Pair****

INX			DCX		
	B	03		B	0B
	D	13		D	1B
	H	23		H	2B
	SP	33		SP	3B

Decimal Adjust A*		**Complement A******	
DAA	27	CMA	2F

Complement/Set CY***		**Arithmetic Immediate***	
CMC	3F	ADI byte	C6
STC	37	ACI byte	CE
		SUI byte	D6
		SBI byte	DE

Notes
* All flags may be affected.
** All flags except CARRY may be affected.
*** Only CARRY FLAG affected.
**** No flags affected.

Data Manipulation Group – Logical

Instructions in this group perform logical operations on data in the registers and the memory.

AND*		**OR***		**Exclusive-OR***	
ANA		ORA		XRA	
A	A7	A	B7	A	AF
B	A0	B	B0	B	A8
C	A1	C	B1	C	A9
D	A2	D	B2	D	AA
E	A3	E	B3	E	AB
H	A4	H	B4	H	AC
L	A5	L	B5	L	AD
M	A6	M	B6	M	AE

Compare*		Rotate***		Logical Immediate*	
CMP A	BF	RLC	07	ANI byte	E6
CMP B	B8	RRC	0F	XRI byte	EE
CMP C	B9	RAL	17	ORI byte	F6
CMP D	BA	RAR	1F	CPI byte	FE
CMP E	BB				
CMP H	BC				
CMP L	BD				
CMP M	BE				

Notes
* All flags may be affected.
*** Only the CARRY flag may be affected.

Transfer of Control Group or Branch Group

This group of instructions alters the sequence of program flow by testing the condition flags.

Jump		Call		Return	
JMP addr	C3	CALL addr	CD	RET	C9
JNZ addr	C2	CNZ addr	C4	RNZ	C0
JZ addr	CA	CZ addr	CC	RZ	C8
JNC addr	D2	CNC addr	D4	RNC	D0
JC addr	DA	CC addr	DC	RC	D8
JPO addr	E2	CPO addr	E4	RPO	E0
JPE addr	EA	CPE addr	EC	RPE	E8
JP addr	F2	CP addr	F4	RP	F0
JM addr	FA	CM addr	FC	RM	F8

Jump Indirect

PCHL	E9

Input/Output Group

This group of instructions performs I/O instructions between the accumulator and a specified port.

IN port	DB
OUT port	D3

Stack and Machine Control Group

This group of instructions maintains the stack and internal control flags.

Stack operations

PUSH	B	C5		POP	B	C1
	D	D5			D	D1
	H	E5			H	E1
	PSW	F5			PSW	F1
XTHL	E3		SPHL	F9		

Interrupt Control

EI	FB
DI	F3
RIM	20
SIM	30

Processor Control

NOP	00
HLT	76

Restart

RST	0	C7
	1	CF
	2	D7
	3	DF
	4	E7
	5	EF
	6	F7
	7	FF

Appendix B: 8085 Instruction Set by Clock Cycles

Mnemonic	*Clock cycles*
MOVE, LOAD AND STORE	
MOV r1,r2	4
MOV M,r	7
MOV r,M	7
MVI r	7
MVI M	10
LXI B	10
LXI D	10
LXI H	10
LXI SP	10
STAX B	7
STAX D	7
LDAX B	7
LDAX D	7
STA	13
LDA	13
SHLD	16
LHLD	16
XCHG	4
STACK OPERATIONS	
PUSH B	12
PUSH D	12
PUSH H	12
PUSH PSW	12
POP B	10
POP D	10
POP H	10
POP PSW	10
XTHL	16
SPHL	6

Mnemonic	*Clock cycles*
RETURN	
RET	10
RC	6/12
RNC	6/12
RZ	6/12
RNZ	6/12
RP	6/12
RM	6/12
RPE	6/12
RPO	6/12
RESTART	
RST	12
INPUT/OUTPUT	
IN	10
OUT	10
INCREMENT AND DECREMENT	
INR r	4
DCR r	4
INR M	10
DCR M	10
INX B	6
INX D	6
INX H	6
INX SP	6
DCX B	6
DCX D	6
DCX H	6
DCX SP	6

JUMP	
JMP	10
JC	7/10
JNC	7/10
JZ	7/10
JNZ	7/10
JP	7/10
JM	7/10
JPE	7/10
JPO	7/10
PCHL	6
CALL	
CALL	18
CC	9/18
CNC	9/18
CZ	9/18
CNZ	9/18
CP	9/18
CM	9/18
CPE	9/18
CPO	9/18
ROTATE	
RLC	4
RRC	4
RAL	4
RAR	4
SPECIALS	
CMA	4
STC	4
CMC	4
DAA	4
CONTROL	
EI	4
DI	4
NOP	4
HLT	5

ADD	
ADD r	4
ADC r	4
ADD M	7
ADC M	7
ADI	7
ACI	7
DAD B	10
DAD D	10
DAD H	10
DAD SP	10
SUBTRACT	
SUB r	4
SBB r	4
SUB M	7
SBB M	7
SUI	7
SBI	7
LOGICAL	
ANA r	4
XRA r	4
ORA r	4
CMP r	4
ANA M	7
XRA M	7
ORA M	7
CMP M	7
ANI	7
XRI	7
ORI	7
CPI	7
INTERRUPT MASK	
RIM	4
SIM	4

Note: Two possible cycle times, for example, 6/12, indicates that the number of instruction cycles involved is dependent on the condition flags.

Solutions to Problems

1.1 (a) 354_8, (b) 011101100_2, (c) 0ECH.
1.2 (a) 111011_2, (b) 1531_8, (c) 1245H.
1.3 (a) 2730, (b) 156, (c) 85.
1.4 (a) 1000100001000_2, (b) 10410_8, (c) 1108H.
1.5 (a) −86, (b) −85.
1.6 (a) 00000101, (b) 11111011, (c) 11100110, (d) 10110000.
1.7 (a) 65 535, (b) +32 767 and −32 768.
1.8 Both the carry and overflow flags are 'set' to '1'.

2.9 4K locations.
2.11 ROM A = ROM B = $\frac{1}{2}$K.
RAM C = RAM D = RAM E = RAM F = $\frac{1}{4}$K.
System expansion blocks = 1K each.

5.5 (a) 2K, (b) (i) 0000H, (b) (ii) 07FFH.
5.6 (a) F800H, (b) FFFFH.

7.2 219_{10} or DBH.
7.3 3.28 ms.

9.15
```
MVI H,00      ; HIGH BYTE OF INDEX ADDRESS = 00H
MVI L,byte    ; LOW BYTE OF INDEX ADDRESS IN  L
LXI D,2000H   ; BASE ADDRESS OF TABLE IN D AND E
DAD D         ; FORM EFFECTIVE ADDRESS OF RESULT
MOV A,M       ; GET PETHERICK CODE FROM TABLE
STA 200FH     ; STORE IT IN 200FH
HLT           ; HALT
```

10.4 COUNT 1 = 219_{10} or DBH, COUNT 2 = 199_{10} or C7H.
(*Note:* in general the value of the product COUNT 1 × COUNT 2 = $43\,581_{10}$.)

11.7

	(a)	(b)	(c)	(d)	(e)	(f)
PC	2060	2080	20A0	208C	2068	2025
SP	20AE	20AC	20AA	20AC	20AE	20B0

12.3

Address	Byte 1 2 3	Label	Instruction mnemonic	Comment
2010	3E 01		MVI A,01H	
2012	D3 20		OUT 20H	; INITIALISE PIO
2014	31 B0 20		LXI SP,20B0H	; INITIALISE STACK POINTER
2017	DB 22		IN 20H	; READ SWITCHES
2019	2F		CMA	; COMPLEMENT DATA
201A	32 00 20	LOOP:	STA 2000H	; STORE DATA IN 2000H
201D	D3 21		OUT 21H	; DISPLAY DATA ON LEDs
201F	CD 09 03		CALL TIME	; CALL TIME DELAY
2022	DB 22		IN 22H	; READ SWITCHES
2024	A7		ANA A	; SET FLAGS
2025	C2 1A 20		JNZ LOOP	; RETURN TO LOOP IF ANY ; SWITCH = 1
2028	D3 21		OUT 21H	; EXTINGUISH LEDS
202A	76		HLT	; HALT PROGRAM

12.4

Label	Instruction mnemonic	Comment
	LXI SP,20B0H	; INITIALISE STACK POINTER
	MVI A,01H	; INITIALISE PIO
	OUT 20H	
START:	IN 22H	; READ STATE OF SWITCHES
	STA 2000H	; STORE DATA IN LOCATION 2000H
	CALL 03DFH	; DISPLAY DATA ON VDU
	LDA 2000H	; RECALL DATA
	XRI 0FH	; INVERT LOW BYTE OF DATA
	OUT 21H	; DISPLAY RESULT ON LEDS
	JMP START	; READ SWITCHES ONCE AGAIN

14.5 0010H, 0020H, 0024H, 003CH.

14.6 Yes.

14.8
```
LXI SP,20B0H    ; INITIALISE STACK POINTER
MVI A,0CH       ; UNMASK RST 6.5 AND 5.5
SIM
EI              ; ENABLE INTERRUPTS
```

14.9

Address	Byte 1 2 3	Label	Instruction mnemonic	Comment
2020	06 04		MVI B,04H	; LOAD COUNTER
2022	21 00 20		LXI H,2000H	; POINTER M = 2000H
2025	97		SUB A	; CLEAR ACCUMULATOR
2026	86	LOOP:	ADD M	; FORM SUM IN ACCUMULATOR
2027	23		INX H	; INCREMENT POINTER
2028	05		DCR B	; DECREMENT COUNTER
2029	C2 26 20		JNZ LOOP	; JUMP IF SUM NOT COMPLETE
202C	32 04 20		STA 2004H	; STORE SUM
202F	76		HLT	; WAIT FOR INTERRUPT
2030	C3 30 20	STOP:	JMP STOP	; TERMINATE PROGRAM
....				
....				
2080	3A 04 20	INTR:	LDA 2004H	; GET SUM
2083	07		RLC	; ROTATE DATA LEFT TWICE
2084	07		RLC	; TO MULTIPLY BY FOUR
2085	E6 FC		ANI FC	; MASK OUT NON-SIGNIFICANT BITS
2087	32 05 20		STA 2005H	; STORE PRODUCT
				; INTERRUPT NEED NOT BE RE-ENABLED
208B	C9		RET	; RETURN TO MAIN PROGRAM

14.10 (a) 5.325 ms (b) 21.3 ms.

Index of Useful Programs

Index