

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	
M	Represents the flag register
byte	The 16-bit memory address currently held by the register pair H and L
dble	An 8-bit data quantity
addr	A 16-bit (two byte) data quantity
port	A 16-bit address
r, r1, r2	An 8-bit I/O port address
rp	One of the registers A, B, C, D, E, H, L
	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
		A,B	78			B,B	40			C,B	48
		A,C	79			B,C	41			C,C	49
		A,D	7A			B,D	42			C,D	4A
		A,E	7B			B,E	43			C,E	4B
		A,H	7C			B,H	44			C,H	4C
		A,L	7D			B,L	45			C,L	4D
		A,M	7E			B,M	46			C,M	4E

MOV	{	D,A	57	MOV	{	E,A	5F	MOV	{	H,A	67
		D,B	50			E,B	58			H,B	60
		D,C	51			E,C	59			H,C	61
		D,D	52			E,D	5A			H,D	62
		D,E	53			E,E	5B			H,E	63
		D,H	54			E,H	5C			H,H	64
		D,L	55			E,L	5D			H,L	65
		D,M	56			E,M	5E			H,M	66

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

Load Immediate (Reg. pair)

LXI	{	B,db1e	01
		D,db1e	11
		H,db1e	21
		SP,db1e	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	{	A	87	ADC	{	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	{	A	97	SBB	{	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	{	A	3C	DCR	{	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	B	03	DCX	B	0B
	D	13		D	1B
	H	23		H	2B
	SP	33		SP	3B

Decimal Adjust A*

DAA 27

Complement A****

CMA 2F

Complement/Set CY***

CMC 3F

STC 37

Arithmetic Immediate*

ADI byte C6

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*

ANA	A	A7
	B	A0
	C	A1
	D	A2
	E	A3
	H	A4
	L	A5
	M	A6

OR*

ORA	A	B7
	B	B0
	C	B1
	D	B2
	E	B3
	H	B4
	L	B5
	M	B6

Exclusive-OR*

XRA	A	AF
	B	A8
	C	A9
	D	AA
	E	AB
	H	AC
	L	AD
	M	AE

Compare*		Rotate***		Logical Immediate*	
CMP	A	BF	RLC 07	ANI byte	E6
	B	B8	RRC 0F	XRI byte	EE
	C	B9	RAL 17	ORI byte	F6
	D	BA	RAR 1F	CPI byte	FE
	E	BB			
	H	BC			
	L	BD			
	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		ADD	
JMP	10	ADD r	4
JC	7/10	ADC r	4
JNC	7/10	ADD M	7
JZ	7/10	ADC M	7
JNZ	7/10	ADI	7
JP	7/10	ACI	7
JM	7/10	DAD B	10
JPE	7/10	DAD D	10
JPO	7/10	DAD H	10
PCHL	6	DAD SP	10
CALL		SUBTRACT	
CALL	18	SUB r	4
CC	9/18	SBB r	4
CNC	9/18	SUB M	7
CZ	9/18	SBB M	7
CNZ	9/18	SUI	7
CP	9/18	SBI	7
CM	9/18	LOGICAL	
CPE	9/18	ANA r	4
CPO	9/18	XRA r	4
ROTATE		ORA r	4
RLC	4	CMP r	4
RRC	4	ANA M	7
RAL	4	XRA M	7
RAR	4	ORA M	7
SPECIALS		CMP M	7
CMA	4	ANI	7
STC	4	XRI	7
CMC	4	ORI	7
DAA	4	CPI	7
CONTROL		INTERRUPT MASK	
EI	4	RIM	4
DI	4	SIM	4
NOP	4		
HLT	5		

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 \times 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			Label	Instruction mnemonic	Comment
	1	2	3			
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			Label	Instruction mnemonic	Comment
	1	2	3			
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

<i>Table</i>	<i>Program</i>	
9.4	Hexadecimal addition program	112
9.5	Decimal addition program	113
9.7	'Squares' program using a lookup table	115
10.1	Program that adds five hex values	120
10.4	Short time delay program	127
10.5	Nested time delay	129
11.6	Simple subroutine	138
11.8	Simple nested subroutines	142
12.2	Digital input and output of data	150
12.3	Flashing light sequence	151
12.5	Direct control of a stepper motor	155
12.6	Stepper motor control using a special 'driver' integrated circuit	158
12.11	Handshake program using two handshake lines	168
13.2	Square wave program	176
13.3	Ramp waveform program	177
13.4	Generating a waveform using a 'lookup' table	179
13.6	Program for a successive approximation ADC	187
14.2	Typical RST 6.5 initialisation procedure	203
14.3	A program containing an interrupt (RST 6.5)	206
14.6	An elapsed time interrupt	213
14.7	A handshake data transfer — polled interrupts	218
14.8	Interrupt-driven handshake, main program	220
14.9	Interrupt-driven handshake, input handshake (RST 5.5)	220
14.10	Interrupt-driven handshake, output handshake (RST 6.5)	220

Index

- Accumulator 28
- Accumulator I/O addressing 24
- Addition 6
 - binary 7
 - double-length 88
 - hexadecimal 8
- Address 18
- Address bus 18
- Address decoding 45–58
- Address field 111
- Address/data bus 32
- Addressing mode 102
 - direct 105
 - extended 105
 - immediate 102
 - implied 104
 - indexed 106, 114
 - indirect 106
 - inherent 104
 - register direct 106
 - register indirect 107
 - relative 109
 - stack 106
 - zero page 106
- Algorithm 82
- Analogue-to-digital convertor (ADC) 179
 - continuous balance 182
 - interfacing to 188–94
 - successive-approximation 183–8
- Architecture of CPU 28
- Arithmetic and logic instructions 96
- Arithmetic and logic unit (ALU) 14, 30
- Arithmetic shift 100
- Assembler statement 110
- Assembly language 94, 110
- Auxiliary carry flag 30
- Base 1
- Baud rate 74
- Bidirectional data bus 18
- Binary digit (bit) 2
- Binary number 1
- Bit time 74
- Branch instructions *see* Jump instructions
- Break point 93
- Buffer 60
- Bus system 18, 26
- Byte 12
- Call instruction 132
- Carry flag 9, 29, 30
 - auxiliary 30
- Central processing unit (CPU) 14
- Chip 15
- Chip enable pin 20, 25
- Chip select pin 20, 46, 47
- Clock oscillator 16, 31, 35
- Code conversion 116
- Comment, in program 111
- Comparison instruction 98
- Complement, binary 9
- Conditional call instructions 132
- Conditional jump instructions 101
- Conditional return instructions 132
- Control and timing section 31
- Control bus 18
- Control unit of CPU 14
- C/S register of PIO 147
 - command aspect of 147, 148, 164
 - status aspect of 166

- Data bus 18
- Data manipulation instructions 95
- Data rate 76
- Data transfer instructions 95
- Debugging 90
- Decimal adjustment instruction 112, 113
- Decoder 46–57
- Dedicated microcomputer 14
- Delay (time) routine 78–80, 125–9
- Device enable pin 20
- Device select pin 20
- Digital computer 14
- Digital input and output 148
- Digital-to-analogue convertor (DAC) 173–9
 - waveform generation using 175
- Direct digital control (DDC) 152
- Displacement 108
- Dual-in-line package (DIP) 15
- Dynamic RAM 44

- Effective address 108, 114
- EPROM 16, 17

- Fetch-execute cycle 33
- Field structure, of assembly language instruction 110–13
- FIFO store 44
- Firmware 18
- Flag 9, 29
- Flag register 9, 29
- Flip-flop 9
- Flowchart 82
- Foldback, on memory map 57
- Full-adder 7

- General-purpose computer 14

- Half-carry flag *see* Auxiliary carry flag
- Handshake 42, 159–69
 - implementing 161
 - input 160, 163
 - output 160, 163
- Hardware 17
- Hexadecimal numbers 3

- Index (in addressing mode) 114
- Index register 108
- Input/output port 17, 20, 23
 - bidirectional 22
 - dedicated 17
 - programmable 17, 145–8
 - unidirectional 22
- Instruction decoder 31
- Instruction register 30
- Instruction set 94, 223, 229
- Integrated circuit 15
- Interface 17, 60
- Interrupt 41, 196
 - handshake 214–21
 - maskable 196
 - multilevel 206
 - non-maskable 41, 196
 - polled 197, 217
 - priority of 200
 - timer (PIO) 209
 - vectored 197, 199
 - 8085 198
- Interrupt flag 197
- Interrupt handling 202
- Interrupt mask (I register) 200
- Interrupt routine 41
- Interval timer 78
- I/O control module 148, 149
- I/O read 39
- I/O timing 39
- I/O write 37

- Jump instructions 101

- Keyboard 67–73

- Label, in program 86, 94, 111
- Latched I/O port 63–7, 70
- LED, 7-segment 63–8
- LIFO store 44, 106, 134
- Listener, in handshake 160

- Logical instructions 96–100
- Logical shift 100
- Lookup table 113–16, 178, 179
- Loop, in program 118

- Machine code instruction 94, 95, 109
- Machine cycle 35
- Mainframe computer 14
- Maskable interrupt 196
- Memory 14
- Memory access time 39
- Memory map 23
- Memory read 38
- Memory register 28
- Memory write 36
- Memory-mapped address 24
- Memory-mapped I/O 24
- Microcomputer 14
- Microcycle 35
- Microprocessor-based system 14, 18
 - minimal 17
- Minicomputer 14
- Monitor program 92
- Multiplexing 32

- Negative flag 30
- Negative number, binary 9, 10
- Non-maskable interrupt 41, 196
- Non-volatile memory 44
- Number conversion 4–6

- Octal number 2
- Offset 108, 114
- One's complement 9, 10
- Opcode 30, 90, 111
- Operand 14
- Operand field, in assembly program 111
- Operation code 30, 90
- Output port 20–3
- Overflow 12
- Overflow flag 12

- Page 23
- Parallel I/O port 20–3
- Parameter passing 143
- Parity flag 29, 30
- Peripheral 17
- Pointer register 28, 105
- Polling 40, 217
- POP instruction 135–7
- Port 17, 20, 33, 145
- Processor status word (PSW) 137
- Program 14, 17, 82
- Program counter 28
- Programmable I/O port 17, 145–8
 - timer in 209
- Programmable timer 80
- Pseudo-operation 112
- PUSH instruction 135–7

- Radix 1
- RAM (random-access memory) 16, 44
- Read signal 31
- Read-write memory 16
- Real-time clock 78
- Register 28
- Return instruction 132
- Rollover (keyboard) protection 70
- ROM (read-only memory) 16, 47
- Rotate instruction 98

- Serial I/O port 20, 23, 73
- Seven-segment display 63–8
- Sign bit 9
- Sign flag 30
- Signed binary number 9
- Single stepping 93
- Software 17
- Stack 134
- Stack pointer 29, 135
- State, timing 35
- Static RAM 44
- Status register *see* Flag register
- Stepper motor 152–9
- Store 14
- Strobed data transfer 159
- Structured programming 82–7

- Subroutine 132, 149
 - nested 140
- Subroutine call 102
- Subroutine return 102
- Subroutine structure 133
- Subtraction, binary 10, 11
- Successive-approximation ADC 183–8
- Symbolic address (label) 86, 94

- Talker, in handshake 159
- Test and branch instructions 101
- Three-state (tri-state) gate 24–6, 60
- Time delay program 78–80, 125–7, 151
 - nested 127–9
- Timer (PIO or hardware) 209
- Timing sequences 78
- Trace routine 90

- Trace table 89, 92, 139
- T-state 35
- Two's complement 10, 11

- UART 73
- Unconditional jump instruction 101
- Unsigned binary number 8, 11

- Vectored interrupt 197, 199
- Video display unit (VDU) 18
- Volatile memory 44

- Write signal 31

- Zero flag 30