THE 8085 INSTRUCTION SET

The 8085 instruction set includes five different types of instructions.

1. Data Transfer Group:

Move data between registers or between memory and register.

2. Arithmetic Group:

Add, subtract increment or decrement data in registers or memory.

3. Logical Group:

AND, OR, EXCLUSIVE-OR, compare, rotate or complement data in registers or in memory.

4. Branch Group:

Conditional and unconditional jump instructions, subroutine call instructions and return instructions.

5. Stack, I/O, and Machine Control Group:

Includes I/O instructions, as well as instructions for maintaining the stack and internal control flags.

Note: only Arithmetic and logical instructions affected the flag register.

The 8085 can operate either on the internal CPU registers (A, B, C, D, E, H, & L) or in the system memory (RAM or ROM). The different addressing capabilities on the 8085 are:

a. Implied:

Meaning that the operation involves an operation on one of the registers. (E.g. ADD B, INR C).

b. Immediate:

Which involves an operation with a type supplied immediately after the instruction. (E.g. ADI 03, ORI 02).

c. Direct:

Which involves an operation with data sound on the address supplied as two bytes after the instruction. (e.g. LDA 2050, STA 2051).

DATA TRANSFER AND ARITHMETIC INSTRUCTIONS

An instruction in the data transfer group, transfers data from a source location to a destination location. Either of these may be a memory location or a register.

The data transfer operations are:

1. MVI rd , byte:- This instruction moves the immediate data given after the instruction into register rd.
2. MOV rd, rs:- This instruction moves data from register rs to register rd.
3. LXI rp, 16-bit:- This instruction moves 16-bit data or address to register pair.

4.XCHG:- The contents of register H are exchanged with the contents of register D, and the contents of register L are exchanged with the contents of register E.

No	Instruction	Type	No. of Bytes	Function	Effect on flags
1.	MVI rd,byte	Data transfer	2	rd=byte	None
2.	MOV rd,rs	Data transfer	1	rd=rs	None
3.	LXI rp,16-bit	Data transfer	3	rp=16-bit (data/ address) Lreg=Lbyte, Hreg=Hbyte	None
4.	XCHG	Data Transfer	1	HL ↔ DE	None

rd: destination register (8-bit), rs: source register (8-bit).
rp: register pair (16-bit): (BC, DE, and HL)

The arithmetic operations are:

The arithmetic instructions perform operations on the data stored in memory or registers and affect the flag register. These instructions are:

1. INR r: - This instruction increments the contents of the register r.

   r=r+1
2. ADD r: - This instruction adds the contents of the register r to the accumulator, and store the results back into the accumulator.

   A=A+r
3. ADI byte: - This instruction adds 8-bit data (byte) to the accumulator, and stores the results back into the accumulator.

   A=A+byte

4. ADC r: - This instruction adds the contents of the register r to the accumulator, but also adds the carry from pervious step, and store the results back into the accumulator.

   A=A+r+CY

5. ACI byte: - This instruction adds 8-bit data to the accumulator, but also adds the carry from pervious step, and store the results back into the accumulator.

   A=A+byte+CY

6. DCR r: - This instruction decrements the contents of the register r.

   r=r-1

7. SUB r: - This instruction subtracts the contents of the register r from the accumulator, and stores the results back into the accumulator.

   A=A-r

8. SUI byte: - This instruction subtracts 8-bit data (byte) from the accumulator, and stores the results back into the accumulator.

   A=A-byte

9. SBB r: - This instruction subtracts the contents of the register r from the accumulator, but also subtracts the carry from pervious step, and store the results back into the accumulator.

   A=A-r-Cy

10. SBI byte: - This instruction subtracts 8-bit data from the accumulator, but also subtracts the carry from pervious step, and store the results back into the accumulator.

    A=A-byte-Cy

11. DAA: - This instruction converts the contents of accumulator from a binary value to two 4-bit binary coded decimal (BCD) digits. This is the only instruction that uses the auxiliary flag to perform the binary to BCD conversion.

The conversion procedure is as follow:-

If the value of the low-order 4-bits in the accumulator is greater than 9 or if the AC flag is set, the instruction adds 6 to the low-order four bits.

If the value of the high-order 4-bits in the accumulator is greater than 9 or if the carry flag is set, the instruction adds 6 to the high-order four bits.

A=BCD number (A)

12. INX rp: - This instruction increments the contents of the register pair rp.

    rp=rp+1

13. DCX rp: - This instruction decrements the contents of the register pair rp.

    rp=rp – 1

14. DAD rp: - This instruction adds the contents of the register pair rp to the contents of register pair HL and store the result back into the HL.

    HL=HL +rp

No	Instruction	Type	No. of Bytes	Function	Effect
1.	INR r	arithmetic	1	r=r+1	All but CY
2.	ADD r	arithmetic	1	A=A+r	All
3.	ADI byte	arithmetic	2	A=A+byte	All
4.	ADC r	arithmetic	1	A=A+r+CY	All
5.	ACI byte	arithmetic	2	A=A+byte+CY	All
6.	DCR r	arithmetic	1	r=r-1	All but CY
7.	SUB r	arithmetic	1	A=A-r	All
8.	SUI byte	arithmetic	2	A=A-byte	All
9.	SBB r	arithmetic	1	A=A-r-CY	All
10.	SBI byte	arithmetic	2	A=A-Byte-CY	All
11.	DAA	arithmetic	1	A=BCD number (A)	All
12.	INX rp	arithmetic	1	rp=rp+1	NONE
13.	DCX rp	arithmetic	1	rp=rp-1	NONE
14.	DAD rp	arithmetic	1	HL=HL+rp	CY

Notes:-

1- Most of the arithmetic instructions affect the contents of an important CPU register; namely the flag register.

2- Most of arithmetic instructions using 8-bit registers are done using the accumulator.

3- The increment or the decrement operations can be performed in any register.

4- For the addition of 8-bit registers, the accumulator is always the 1st operand, but the addition of 16-bit registers, the HL register pair is always the 1st operand.

Example

Find the summation of register B and register C and put the result in register D, when B=5 and C=3

```
Address   Hexcode   Label    Opcode   Operand        Comments

 2000       06      START:   MVI       B, 5           ; B=5
 2001       05
 2002       0E               MVI       C,3            ; C=3
 2003       03
 2004       78               MOV       A,B           ; A=B=5
 2005       81               ADD        C           ; A=A+C=8
                                               ; S=0, Z=0, Ac=0, P=0, Cy=0
 2006       57               MOV       D,A           ; D=A=8
 2007       CF               RST1                     ; End
```

- The address field specifies the address of the respective instruction.
- The hexcode field lists the data after translation of the source code of machine language program. This field contains the data to be entered to the machine.
- The label field specifies the label for the program.
- Opcode field specifies the 8085 instructions to be executed.
- Operand field specifies the data to be operated by the corresponding instruction.
- Comment field is an optional field used to comment lines.

Classification of Instructions according to number of bytes which are taken:

1- One byte instruction: The instruction without data/address translated to machine code and allocate 1 byte from memory

2- Two byte instruction: The instruction which contain 8 bit data Like|:

MVI B,25 or contain 8 bit address Like IN 34 translated to machine code and allocate 2 bytes from memory the 1st byte for the instruction and the 2nd byte for 8 bit data or address.

3- Three byte instruction: The instruction which contain 16 bit data Like|:

LXI H,4532 or contain 16 bit address Like: LDA 2025 translated to machine code and allocate 3 bytes from memory the 1st byte for the instruction and the 2nd byte for the low byte of data/address and 3rd byte for high byte of data/address

Class Work

1) C=(B-10)+(D+H)-1 ; B=30, D=50, H=10

Address	HexCode	Label	Opcode	Operands	Comments
2000 2001	30		MVI	B,30	; B=30
2002 2003	50		MVI	D,50	; D=50
2004 2005	10		MVI	H,10	; H=10
2006			MOV	A,B	; A=B=30
2007 2008	20		SUI	10	;A=A-10=20,S=0,Z=0,AC=0, P=0, CY=0
2009			MOV	C,A	; C=A=20
200A			MOV	A,D	; A=D=50
200B			ADD	H	;A=A+H=60,S=0,Z=0,AC=0, P=1, CY=0
200C			ADD	C	;A=A+C=80,S=1,Z=0,AC=0, P=0, CY=0
200D			DCR	A	; A=A-1=7F, S=0, Z=0,AC=1, P=0
200E			MOV	C,A	; C=A=7F
200F			RST1		; End

2) A=(B-C)-(D+3) ; BC=A502, D=4

Address	HexCode	Label	Opcode	Operands	Comments
2000 2001 2002	02 A5		LXI	B,A502	; B=A5, C=02
2003 2004	04		MVI	D,4	; D=4
2005			MOV	A,D	; A=D=4
2006 2007	03		ADI	3	; A=A+3=7, S=0, Z=0, AC=0, P=0, CY=0
2008			MOV	H,A	; H=A=7
2009			MOV	A,B	; A=B=A5
200A			SUB	C	; A=A-C=A3, S=1, Z=0, AC=0, P=1, CY=0
200B			SUB	H	; A=A-H=9C, S=1, Z=0, AC=1, P=1, CY=0
200C			RST1		; End

3) Exchange DE with HL; DE=15A, HL=8B1C

Address	HexCode	Label	Opcode	Operands	Comments
2000 2001	5A	لا يهـــــم التسلسل	MVI	E,5A	; E=5A
2002 2003	01	لكــن هــذا الحل فقط	MVI	D,01	; D=01
2004 2005	1C	هو المقبول	MVI	L,1C	; L=1C
2006 2007	8B		MVI	H,8B	; H=8B
2008			MOV	A,E	; A=E=5A ممكن استخدام غير A كوسيط
2009			MOV	E,L	; E=L=1C
200A			MOV	L,A	; L=A=5A
200B		ممكن ان	MOV	A,D	; A=D=01
200C		تكون	MOV	D,H	; D=H=8B
200D		قبل	MOV	H,A	; H=A=01
200E			RST1		; End

4) Exchange DE with HL; DE=15A, HL=8B1C using Instructions use Register Pair

Address	HexCode	Label	Opcode	Operands	Comments
2000 2001 2002	 5A 01		LXI	D,15A	; DE=15A
2003 2004 2005	 1C 8B		LXI	H,8B1C	; HL=8B1C
2006			XCHG		; DE=8B1C, HL=15A
2007			RST1		; End

5) HL=BC+DE, BC=2FF, DE=102

Address	HexCode	Label	Opcode	Operands	Comments
2000 2001	 FF		MVI	C,FF	; C=FF
2002 2003	 02		MVI	B,02	; B=02
2004 2005	 02		MVI	E,02	; E=02
2006 2007	 01		MVI	D,01	; D=01
2008			MOV	A,C	; A=C=FF
2009			ADD	E	; A=A+E=01, S=0,Z=0,AC=1, P=0,CY=1
200A			MOV	L,A	; L=A=01
200B			MOV	A,B	; A=B=02
200C			ADC	D	; A=A+D+CY=4, S=0,Z=0,AC=0, P=0,CY=0
200D			MOV	H,A	; H=A=4
200E			RST1		; End

6) HL=BC+DE ; BC=2FF, DE=102 using Instructions use Register Pair

Address	HexCode	Label	Opcode	Operands	Comments
2000 2001 2002	 FF 02		LXI	B,2FF	; BC=2FF
2003 2004 2005	 02 01		LXI	D,102	; DE=102
2006			MOV	L,C	; L=C=FF
2007			MOV	H,B	; H=B=02
2008			DAD	D	; HL=HL+DE=401 , CY=0
2009			RST1		; End

Homework

Write programs with effects

1. E=((A-30)+1)-(H+1A); A=4F, H=3
2. What is the result of each instruction of the following program and its effect?

```
MVI     D,10
MVI     A,50
MVI     L,2F
DCR     A
INR     L
ADD     D
SUB     L
ADI     21
SUI     50
RST1
```