

## MIT Assignment-04

Jeetu Mehra

U24AI032

**Question 1** : Write instruction to increment the contents of a register and show result to memory location

0008H.

**Output:**

Registers			Flag		Load me at	
A	33		S	0		
BC	00	00	Z	0		
DE	00	00	AC	0		
HL	00	00	P	1		
PSW	00	00	C	0		
PC	42	0B				
SP	FF	FF				
Int-Reg	00					

Decimal - Hex Conversion	
Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports	
0	- + 00
<input type="button" value="Update Port Value"/>	

Line	Code
1	
2	; <Program title>
3	
4	jmp start
5	
6	; data
7	
8	
9	; code
10	start: nop
11	
12	MVI A, 32H
13	INR A
14	STA 0008H
15	
16	hlt

Data
Stack
KeyPad
Memory
I/O Ports

Start

Address (Hex)	Address	Data
0008	8	51
0009	9	0
000A	10	0

**Question 2:** Write instruction to increment the contents of a memory location the address of which is in HL Pair.

**Output:**

**Registers**

A	33	
BC	00	00
DE	00	00
HL	20	00
PSW	00	00
PC	42	0B
SP	FF	FF
Int-Reg	00	

**Flag**

S	0
Z	0
AC	0
P	1
C	0

Load me at

```

1
2  ;<Program title>
3
4  jmp start
5
6  ;data
7
8
9  ;code
10 start: nop
11
12 LXI H, 2000H
13 MVI M, 32H
14 INR M
15
16
17
18 hlt

```

**Decimal - Hex Conversion**

Decimal	Hex
<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

**I/O Ports**

<input type="text" value="0"/>	-	<input type="text" value="00"/>
<input type="button" value="Update Port Value"/>		

Data
Stack
Abc KeyPad
**Memory**
I/O Ports

Start

Address (Hex)	Address	Data
2000	8192	51
2001	8193	0
2002	8194	0
----	----	-

**Question 3:** Write instruction to decrement the contents of a register and show result to I/O port address 0009H.

**Output:**

**Registers**

A	31	
BC	00	00
DE	00	00
HL	20	00
PSW	00	00
PC	42	0A
SP	FF	FF
Int-Reg	00	

**Flag**

S	0
Z	0
AC	0
P	0
C	0

Load me at

1	
2	<i>; &lt;Program title&gt;</i>
3	
4	<b>jmp</b> start
5	
6	<i>; data</i>
7	
8	
9	<i>; code</i>
10	<b>start:</b> <b>nop</b>
11	
12	<b>MVI</b> A, 32H
13	<b>DCR</b> A
14	<b>OUT</b> 09H
15	
16	
17	<b>hlt</b>

**Decimal - Hex Conversion**

Decimal	Hex
<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="button" value="→ To Hex"/>	<input type="button" value="← To Dec"/>

**I/O Ports**

<input type="text" value="0"/>	-	+	<input type="text" value="00"/>
<input type="button" value="Update Port Value"/>			

**Question 4 :** Write instruction to decrement the contents of a memory location the address of which is in HL pair and copy that result to register B.

**Output:**

**Registers**

A	31	
BC	31	00
DE	00	00
HL	20	00
PSW	00	00
PC	42	0C
SP	FF	FF
Int-Reg	00	

**Flag**

S	0
Z	0
AC	0
P	0
C	0

Load me at

```

1
2 ;<Program title>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: nop
11 |
12 LXI H, 2000H
13 MVI M, 32H
14 DCR M
15 MOV B, M
16
17
18
19 hlt

```

**Decimal - Hex Conversion**

Decimal	Hex
<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="button" value="→ To Hex"/>	<input type="button" value="← To Dec"/>

**I/O Ports**

Start	<input type="text" value="2000h"/>	<input type="button" value="OK"/>
Address (Hex)	Address	Data
2000	8192	49
2001	8193	0
2002	8194	0
2003	8195	0

**Question 5 :** Write instruction to increment the contents of a register-pair and show result to memory location 003FH.

**Output:**

**Registers**

A	31
BC	31 00
DE	00 00
HL	00 33
PSW	00 00
PC	42 0C
SP	FF FF
Int-Reg	00

**Flag**

S	0
Z	0
AC	0
P	0
C	0

Load me at

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

<Program title>

jmp start

;data

|

;code

start: nop

LXI H, 0032H

INX H

SHLD 003FH

hlt

Decimal - Hex Conversion

Decimal

Hex

0

0

To Hex

To Dec

I/O Ports

Data	Stack	KeyPad	Memory	I/O Ports
Start	003Fh	OK		
Address (Hex)	Address	Data		
003F	63	51		
0040	64	0		
0041	65	0		
0042	66	0		
----	--	-		

**Question 6 :** Write instruction to decrement the contents of a register-pair and show result to memory location 005FH.

**Output:**

**Registers**

A	31	
BC	31	00
DE	00	00
HL	00	31
PSW	00	00
PC	42	0C
SP	FF	FF
Int-Reg	00	

**Flag**

S	0
Z	0
AC	0
P	0
C	0

Load me at

**Decimal - Hex Conversion**

Decimal	Hex
0	0
To Hex	To Dec

I/O Ports

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

<Program title>

jmp start

;data

;code

start: nop

LXI H, 0032H

DCX H

SHLD 005FH

HLT

hlt

Data	Stack	KeyPad	Memory	I/O Ports
Start	005Fh	OK		
Address (Hex)	Address	Data		
005F	95	49		
0060	96	0		
0061	97	0		
0062	98	0		

**Question 7 :** Write instruction to do following operations:

- Load 93H in Accumulator.
- Load B7 H in register C.
- Addition of the above two numbers.

### Output:

The screenshot shows an 8085 assembly simulator interface. The top menu bar includes File, Reset, Assembler, Debug, and Help. Below the menu is a toolbar with icons for file operations, execution, and debugging. The main window is divided into several sections:

- Registers:** A table showing the state of various registers.
 

Register	Value	Register	Value
A	4A		
BC	31	B7	
DE	00	00	
HL	00	31	
PSW	00	00	
PC	42	0A	
SP	FF	FF	
Int-Reg	00		
- Flag:** A section showing the status of various flags.
 

Flag	Status
S	0
Z	0
AC	0
P	0
C	1
- Decimal - Hex Conversion:** A section with two input fields (both containing '0') and two buttons: 'To Hex' and 'To Dec'.
- Assembly Code:** A list of instructions with line numbers.
 

```

1
2 ;<Program title>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: nop
11
12 MVI A, 93H
13 MVI C, 00B7H
14 ADD C
15
16
17 hlt
      
```

**Question 8 :** Write instruction to add the 35H to the sum in the previous example when the carry flag is set.

### Output:

Registers			Flag		Load me at	
A	80		S	1	1	
BC	31	B7	Z	0	2	<Program title>
DE	00	00	AC	1	3	
HL	00	31	P	0	4	jmp start
PSW	00	00	C	0	5	
PC	42	0C			6	;data
SP	FF	FF			7	
Int-Reg	00				8	

💡
**Decimal - Hex Conversion**

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

**I/O Ports**

0	-	+	00
---	---	---	----

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20

```

; <Program title>
jmp start

; data

; code
start: nop

MVI A, 93H
MVI C, 00B7H
ADD C
ACI 35H
HLT

hlt

```

**Question 9** : a. Assume the accumulator holds the data byte FFH. By adding 01H comment about the conditions of all flags.


b. After that increment the content of A and comments about the conditions of flags

**Output:**




Registers			Flag		Load me at	
A	01		S	0		
BC	00	00	Z	0		
DE	00	00	AC	0		
HL	00	00	P	0		
PSW	00	00	C	1		
PC	42	0A				
SP	FF	FF				
Int-Reg	00					


**Decimal - Hex Conversion**

Decimal	Hex
<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>


**I/O Ports**

<input type="text" value="0"/>	<input type="text" value="-"/>	<input type="text" value="00"/>
--------------------------------	--------------------------------	---------------------------------

```

1
2  ;<Program title>
3
4  jmp start
5
6  ;data
7
8
9  ;code
10 start: nop
11
12 MVI A, 0FFH
13 ADI 01H
14 INR A
15
16
17 hlt
  
```

**Question 10 :** Write instruction to do following operations:

- Load 8BH in C.
- Load 6FH in register D.
- Increment the contents of C by one.
- Addition of the contents in C and D and display the sum at the PORT address 001FH.

**Output:**

Registers			Flag		Load me at	
A	FB		S	1		
BC	00	8C	Z	0		
DE	6F	00	AC	1		
HL	00	00	P	0		
PSW	00	00	C	0		
PC	42	0E				
SP	FF	FF				
Int-Reg	00					

**Decimal - Hex Conversion**

Decimal	Hex
31	1F

To Hex

To Dec

**I/O Ports**

**Question 11:** Write instruction to do following operations:

- Load FFH in B.
- Load 01H in A.
- Add the contents of B with A.
- Load 02H in register D.
- Addition of the contents in D with contents of A with carry.

**Output:**

Registers			Flag		Load me at
A	03		S	0	
BC	FF	8C	Z	0	
DE	02	00	AC	0	
HL	00	00	P	1	
PSW	00	00	C	0	
PC	42	0D			
SP	FF	FF			
Int-Reg	00				

💡
**Decimal - Hex Conversion**

Decimal	Hex
31	1F

➡ To Hex
⬅ To Dec

**I/O Ports**

1  
2 ;<Program title>  
3  
4 jmp start  
5  
6 ;data  
7  
8  
9 ;code  
10 start: nop  
11  
12 MVI B, 0FFH  
13 MVI A, 01H  
14  
15 ADD B  
16  
17 MVI D, 02H  
18  
19 ADC D  
20  
21 hlt

**Question 12:** Write instruction to perform following operations:

- Perform Logical AND between the contents of register D (D=54H) with the Contents of Accumulator (A=82H).
- Perform Logical AND between the 8-bit data(97H) with the Contents of Accumulator (A=82H).

c. Perform Logical AND between the contents of memory location with the Contents of Accumulator (A=72H).

### Output:

**Registers**

A	00	
BC	FF	8C
DE	54	00
HL	00	00
PSW	00	00
PC	42	0A
SP	FF	FF
Int-Reg	00	

**Flag**

S	0
Z	1
AC	1
P	1
C	0

Load me at

**Decimal - Hex Conversion**

Decimal	Hex
<input type="text" value="31"/>	<input type="text" value="1F"/>
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

**I/O Ports**

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

```
<Program title>
jmp start

;data

;code
start: nop

MVI A, 82H
MVI D, 54H

ANA D

hlt
```

### Registers

A	82	
BC	FF	8C
DE	54	00
HL	00	00
PSW	00	00
PC	42	09
SP	FF	FF
Int-Reg	00	

### Flag

S	1
Z	0
AC	1
P	1
C	0

Load me at

```

1
2  ;<Program title>
3
4  jmp start
5
6  ;data
7
8
9  ;code
10 start: nop
11
12 MVI A, 82H
13 ANI 97H
14
15 hlt

```

### Decimal - Hex Conversion

Decimal	Hex
<input type="text" value="31"/>	<input type="text" value="1F"/>
<input type="button" value="→ To Hex"/>	<input type="button" value="← To Dec"/>

### Registers

A	32	
BC	FF	8C
DE	54	00
HL	20	00
PSW	00	00
PC	42	0D
SP	FF	FF
Int-Reg	00	

### Flag

S	0
Z	0
AC	1
P	0
C	0

Load me at

```

1
2  ;<Program title>
3
4  jmp start
5
6  ;data
7
8
9  ;code
10 start: nop
11
12 LXI H, 2000H
13 MVI M, 32H
14
15 MVI A, 72H
16
17 ANA M
18
19 hlt

```

### Decimal - Hex Conversion

Decimal	Hex
<input type="text" value="31"/>	<input type="text" value="1F"/>
<input type="button" value="→ To Hex"/>	<input type="button" value="← To Dec"/>

### I/O Ports

<input type="text" value="0"/>	<input type="text" value="-"/>	<input type="text" value="00"/>
--------------------------------	--------------------------------	---------------------------------

**Question 13:** Write instruction to perform following operations:

- a. Perform Logical OR between the contents of register B (D=51H) with the Contents of Accumulator (A=A8H).
- b. Perform Logical OR between the 8-bit data(A6H) with the Contents of Accumulator (A=82H).
- c. Perform Logical OR between the contents of memory location with the Contents of Accumulator (A=C2H).

**Output:**

File
Reset
Assembler
Debug
Help

**Registers**

Register	Value	Flag
A	F2	S 1
BC	51 8C	Z 0
DE	54 00	AC 0
HL	20 00	P 0
PSW	00 00	C 0
PC	42 16	
SP	FF FF	
Int-Reg	00	

**Flag**

Load me at

```

1
2 ;<Program title>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: nop
11
12 ; ----- Part (a) -----
13 ; OR between B (51H) and A (A8H)
14
15 MVI A, 0A8H
16 MVI B, 51H
17 ORA B
18 ; Result → A = F9H
19
20
21 ; ----- Part (b) -----
22 ; OR between 8-bit data (A6H) and A (82H)
23
24 MVI A, 82H
25 ORI 0A6H
26 ; Result → A = A6H
27
28
29 ; ----- Part (c) -----
30 ; OR between memory location and A (A = C2H)
31
32 LXI H, 2000H
33 MVI M, 32H
34
35 MVI A, 0C2H
36 ORA M
37 ; Result → A = F2H
38
39 hlt

```

**Decimal - Hex Conversion**

Decimal	Hex
31	1F

To Hex
To Dec

**I/O Ports**

Port	Value
0	00

Update Port Value

**Memory**

Address	Value
0	00

Update Memory

**Question 14:** Write instruction to perform following operations:

- a. Perform Logical Exclusive-OR between the contents of register B (D=77H) with the  
Contents of Accumulator (A=56H).
- b. Perform Logical Exclusive-OR between the 8-bit data(8FH) with the  
Contents of  
Accumulator (A=A2H).
- c. Perform Logical Exclusive-OR between the contents of memory  
location with the  
Contents of Accumulator (A=4AH).

**Output:**



File Reset Assembler Debug Help

Registers

A	78
BC	51 8C
DE	77 00
HL	20 00
PSW	00 00
PC	42 16
SP	FF FF
Int-Reg	00

Flag

S	0
Z	0
AC	0
P	1
C	0

Load me at

Decimal - Hex Conversion

Decimal	Hex
31	1F
To Hex	To Dec

I/O Ports

0	-	+	00
Update Port Value			

Memory

0	-	+	00
Update Memory			

```

1
2 ;<Program title>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: nop
11
12 ; ----- Part (a) -----
13 ; XOR between D (77H) and A (56H)
14
15 MVI A, 56H
16 MVI D, 77H
17 XRA D
18 ; Result → A = 21H
19
20
21 ; ----- Part (b) -----
22 ; XOR between 8-bit data (8FH) and A (A2H)
23
24 MVI A, 0A2H
25 XRI 08FH
26 ; Result → A = 2DH
27
28
29 ; ----- Part (c) -----
30 ; XOR between memory location and A (A = 4AH)
31
32 LXI H, 2000H
33 MVI M, 32H ; memory value assumed 32H
34
35 MVI A, 4AH
36 XRA M
37 ; Result → A = 78H
38
39
40 hlt

```

**Question 15:** Write instruction to perform following operations:

- Compare the contents of register with the contents of A.
- Compare the 8-bit data with the contents of A.

c. Complement the contents of A.

d. Complement the Carry

Output:

Registers			Flag		Load me at	
A	CD		S	0	1	
BC	32	8C	Z	1	2	<Program title>
DE	77	00	AC	0	3	
HL	20	00	P	1	4	jmp start
PSW	00	00	C	1	5	
PC	42	0E			6	;data
SP	FF	FF			7	
Int-Reg	00				8	

**Decimal - Hex Conversion**

Decimal	Hex
31	1F

[To Hex](#) [To Dec](#)

**I/O Ports**

0	-	+	00
---	---	---	----

[Update Port Value](#)

**Memory**

0	-	+	00
---	---	---	----

[Update Memory](#)

```
1
2 ;<Program title>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: nop
11
12 ; ----- Part (a) -----
13 ; Compare contents of register with A
14 ; Assume A = 32H, B = 32H
15
16 MVI A, 32H
17 MVI B, 32H
18 CMP B
19 ; Only flags change, A unchanged
20
21
22 ; ----- Part (b) -----
23 ; Compare 8-bit data with A
24
25 CPI 32H
26 ; Only flags change
27
28
29 ; ----- Part (c) -----
30 ; Complement contents of A
31
32 CMA
33 ; A becomes complement
34
35
36 ; ----- Part (d) -----
37 ; Complement Carry
38
39 CMC
40
41
42 hlt
```

## Question 16: Perform 16-bit Addition

### Output:

Registers			Flag	Load me at
A	CD		S 0	1
BC	32	8C		2 ;<Program title>
DE	32	10	Z 1	3
HL	64	64		4 jmp start
PSW	00	00	AC 0	5
PC	42	0C	P 1	6 ;data
SP	FF	FF		7
Int-Reg	00		C 0	8

**Decimal - Hex Conversion**

Decimal	Hex
31	1F

**I/O Ports**

0	-	+	00
---	---	---	----

```
1
2 ;<Program title>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: nop
11
12 ; Load first 16-bit number in HL
13 LXI H, 3254H
14
15 ; Load second 16-bit number in DE
16 LXI D, 3210H
17
18 ; 16-bit addition → HL = HL + DE
19 DAD D
20
21 hlt
```