

U18CO018
Shubham Shekhaliya
Sub: MIT
Lab: Assignment-4

1 Write a program to load the data byte A8H in register C. Mask the high-order bits(D7-D4), And display the low-order bits (D3-D0) at an output port.

Code:-

```
mvi c,0A8H
mov a,c
; for the last 4 bit do and with 0FH
mvi b,00FH
ana b
out 01H
hlt
```

Output:-

The screenshot shows the GNUSim8085 - 8085 Microprocessor Simulator interface. The main window displays the assembly code entered in the 'Load me at' field:

```
1 mvi c,0A8H
2 mov a,c
3 ; for the last 4 bit do and with 0FH
4 mvi b,00FH
5 ana b
6 out 01H
7
8 hlt
```

The 'Registers' panel on the left shows the state of the 8085 registers:

Register	Value
A	08
BC	0F A8
DE	00 00
HL	00 00
PSW	00 00
PC	42 09
SP	FF FF
Int-Reg	00

The 'Flag' panel shows the status of the flags:

Flag	Status
S	0
Z	0
AC	1
P	0
C	0

The 'Memory' panel on the right shows the memory dump starting at address 0000:

Address (Hex)	Address	Data
0000	0	0
0001	1	0
0002	2	0
0003	3	0
0004	4	0
0005	5	0
0006	6	0
0007	7	0
0008	8	0
0009	9	0
000A	10	0

The 'I/O Ports' panel shows the output port value as 08.

The 'Assembler Message' panel at the bottom right shows the message: "Program assembled successfully".

Simulator: Idle

2 Write a program to load the data byte 8EH in register D and F7H in register E. Mask the high-order bits (D7-D4) from both the data bytes, Exclusive-OR the low-order bits (D3-D0) and display the answer.

Code:-

```
mvi D,8EH
mvi E,0F7H
mvi b, 0FH
mov a,d
ana b
mov d,a
mov a,e
ana b
mov e,a
mov a,d
xra e      ; accumulator contains the xor of the both number (LSB)
;at port 1
out 01H
hlt
```

Output:-

The screenshot displays the GNUSim8085 - 8085 Microprocessor Simulator interface. The main window shows the assembly code being executed, with line numbers 1 through 15. The code includes instructions for loading data into registers D and E, performing bitwise operations (AND, XOR), and outputting the result to port 1. The simulator's status bar at the bottom indicates "Simulator: Idle".

Registers:

Register	Value
A	09
BC	0F A8
DE	0E 07
HL	00 00
PSW	00 00
PC	42 11
SP	FF FF
Int-Reg	00

Flag:

Flag	Value
S	0
Z	0
AC	0
P	1
C	0

Assembly Code:

```
1 mvi D,8EH
2 mvi E,0F7H
3 mvi b, 0FH
4 mov a,d
5 ana b
6 mov d,a
7 mov a,e
8 ana b
9 mov e,a
10 mov a,d
11 xra e
12 ; accumulator contains the xor of the both number (LSB)
13 ;at port 1
14 out 01H
15 hlt
```

Memory Dump:

Address (Hex)	Address	Data
0001	1	0
0002	2	0
0003	3	0
0004	4	0
0005	5	0
0006	6	0
0007	7	0
0008	8	0
0009	9	0
000A	10	0

Assembler Message:

Line No	Assembler Message
0	Program assembled successfully

3 Write a program to load the bit pattern 91H in register B and 87H in register C. Mask all the bits except D0 from registers B and C.

Code:-

```
mvi b,91H
mvi c,87H
mvi d,01H
mov a,b
ana d
mov b,a ; register b contain the answer
mov a,c
ana d
mov c,a ; register c contain the answer
hlt
```

Output:-

The screenshot displays the GNUSim8085 - 8085 Microprocessor Simulator interface. The main window shows the assembly code being executed, with line numbers 1 through 11. The code is as follows:

```
1 mvi b,91H
2 mvi c,87H
3 mvi d,01H
4 mov a,b
5 ana d
6 mov b,a ; register b contain the answer
7 mov a,c
8 ana d
9 mov c,a ; register c contain the answer
10 hlt
11
```

On the left side, the Registers window shows the current state of the microprocessor registers. The registers A, BC, DE, HL, PSW, PC, SP, and Int-Reg are listed with their values. The PC register is highlighted, showing the value 42. The Flag window shows the status of the flags: S, Z, AC, P, and C.

On the right side, the Memory window shows the memory contents. The memory is organized into a table with columns for Address (Hex), Address, and Data. The memory contents are as follows:

Address (Hex)	Address	Data
0001	1	0
0002	2	0
0003	3	0
0004	4	0
0005	5	0
0006	6	0
0007	7	0
0008	8	0
0009	9	0
000A	10	0

At the bottom right, the Assembler Message window shows the message: "Program assembled successfully".

4 Write a program to clear the CY flag, to load number FFH in register B, and increment B. If the CY flag is set, display 01 at the output port, otherwise, display the contents of register B.

Code:-

```
xra a ;clear the cy flag
mvi b,0FFH
inr b ;it doesn't affect CY flag
jnc show ;if carry then show 01H
mvi a,01H
out 01H
jmp end
show: mov a,b
out 01H
end: hlt
```

Output:-

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers

Register	Value
A	00
BC	00 01
DE	01 07
HL	00 00
PSW	00 00
PC	42 12
SP	FF FF
Int-Reg	00

Flag

Flag	Value
S	0
Z	1
AC	1
P	1
C	0

Decimal - Hex Conversion

Decimal: 0 Hex: 0

→ To Hex ← To Dec

I/O Ports

1 - + 00

Update Port Value

Memory

0 - + 00

Update Memory

Load me at

```
1 xra a ;clear the cy flag
2 mvi b,0FFH
3 inr b ;it doesn't affect CY flag
4 jnc show ;if carry then show 01H
5 mvi a,01H
6 out 01H
7 jmp end
8 show: mov a,b
9 out 01H
10 end: hlt
```

Start 01h OK

Address (Hex)	Address	Data
0001	1	0
0002	2	0
0003	3	0
0004	4	0
0005	5	0
0006	6	0
0007	7	0
0008	8	0
0009	9	0
000A	10	0

Line No Assembler Message

0 Program assembled successfully

Simulator: Idle

5 Write a program to mask lower bit of an 8 bit number.

Code:-

```
mvi a,0A5H
```

```
ani 0F0H ;mask lsb and accumulator contains result
```

```
hlt
```

Output:-

The screenshot displays the GNUSim8085 - 8085 Microprocessor Simulator interface. The main window is divided into several sections:

- Registers:** A table showing the state of various registers. The Accumulator (A) contains 0A5H. The Program Counter (PC) contains 4205H. The Stack Pointer (SP) contains FF FF. The Interrupt Register (Int-Reg) contains 00.
- Flags:** A table showing the state of various flags. The Sign flag (S) is 1. The Zero flag (Z) is 0. The Carry flag (C) is 0.
- Decimal - Hex Conversion:** A section for converting between decimal and hexadecimal values. The decimal input is 0, and the hexadecimal output is 0.
- I/O Ports:** A section for interacting with I/O ports. The port value is 08.
- Memory:** A section for interacting with memory. The memory address is 0, and the memory value is 00.
- Assembly Program:** A text area containing the assembly code:

```
1 mvi a,0A5H
2 ani 0F0H ;mask lsb and accumulator contains result
3 hlt
4
```
- Memory Window:** A table showing the memory contents. The address range is from 0000 to 0009. The data values are all 0.
- Assembler Message:** A log showing the message: "Program assembled successfully".

The simulator status at the bottom indicates "Simulator: Idle".

6 Write a program Load two unsigned numbers in register B and register C respectively. Subtract C from B. If the result is in 2's complement, convert the result in absolute magnitude And display it at PORT 1, otherwise, display the positive result. Execute the program.

Code:-

```
mvi b,042H
mvi c,069H
mov a,b
sub c
jnc display ;if ans -ve then it will produce carry
cma ;for getting magnitude one's complement
inr a ; two 's complement
display: out 01H
hlt
```

Output:-

A. For Set1:B=42H,C=69H

The screenshot displays the GNUSim8085 - 8085 Microprocessor Simulator interface. The main window is titled "GNUSim8085 - 8085 Microprocessor Simulator". The interface includes a menu bar (File, Reset, Assembler, Debug, Help) and a toolbar with various icons. The central area shows the assembly code being executed, with line numbers 1 through 10. The code is as follows:

```
1 mvi b,042H
2 mvi c,069H
3 mov a,b
4 sub c
5 jnc display ;if ans -ve then it will produce carry
6 cma ;for getting magnitude one's complement
7 inr a ; two 's complement
8 display: out 01H
9 hlt
10
```

On the left side, the "Registers" panel shows the current state of the 8085 registers. The "Flag" panel shows the status of the flags. The "Decimal - Hex Conversion" panel shows the conversion of decimal to hex and vice versa. The "I/O Ports" panel shows the current value of port 1, which is 27. The "Memory" panel shows the current memory address and data.

On the right side, the "Memory" panel shows the memory contents. The "Data" panel shows the current data value, which is 0h. The "Stack" panel shows the current stack pointer, which is 0h. The "KeyPad" panel shows the current keypad input, which is 0h. The "I/O Ports" panel shows the current port value, which is 27.

At the bottom, the "Assembler Message" panel shows the message "Program assembled successfully".

B. Set2:B=69H,C=42H

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers

Register	Value
A	27
BC	69 42
DE	00 00
HL	00 00
PSW	00 00
PC	42 0E
SP	FF FF
Int-Reg	00

Flag

Flag	Value
S	0
Z	0
AC	0
P	1
C	0

Load me at

```

1 mvi b,069H
2 mvi c,042H
3 mov a,b
4 sub c
5 jnc display ;if ans -ve then it will produce carry
6 cma ;for getting magnitude one's complement
7 inr a ; two 's complement
8 display: out 01H
9 hlt
10

```

Decimal - Hex Conversion

Decimal: 0 Hex: 0

To Hex To Dec

I/O Ports

1 - + 27

Update Port Value

Memory

0 - + 00

Update Memory

Start 0h OK

Address (Hex)	Address	Data
0000	0	0
0001	1	0
0002	2	0
0003	3	0
0004	4	0
0005	5	0
0006	6	0
0007	7	0
0008	8	0
0009	9	0

Line No Assembler Message

0 Program assembled successfully

Simulator: Idle

C. Set 3: B=F8H,C = 23H

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers

Register	Value
A	D5
BC	F8 23
DE	00 00
HL	00 00
PSW	00 00
PC	42 0E
SP	FF FF
Int-Reg	00

Flag

Flag	Value
S	1
Z	0
AC	0
P	0
C	0

Load me at

```

1 mvi b,0F8H
2 mvi c,023H
3 mov a,b
4 sub c
5 jnc display ;if ans -ve then it will produce carry
6 cma ;for getting magnitude one's complement
7 inr a ; two 's complement
8 display: out 01H
9 hlt
10

```

Decimal - Hex Conversion

Decimal: 0 Hex: 0

To Hex To Dec

I/O Ports

1 - + D5

Update Port Value

Memory

0 - + 00

Update Memory

Start 0h OK

Address (Hex)	Address	Data
0000	0	0
0001	1	0
0002	2	0
0003	3	0
0004	4	0
0005	5	0
0006	6	0
0007	7	0
0008	8	0
0009	9	0

Line No Assembler Message

0 Program assembled successfully

Simulator: Idle