

# MIT Lab 4

## U18CO021: SAHIL BONDRE

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.

```
; <q1>

jmp start
;data
;code
start: nop
mvi C, 0A8H
mvi A, 0FH
ana C
out 00H
hlt
```

Registers			Flag	
A	08		S	0
BC	00	A8	Z	0
DE	0E	07	AC	1
HL	00	00	P	0
PSW	00	00	C	0
PC	42	0C		
SP	FF	FF		
Int-Reg	00			

≡ I/O Ports

0

− +

08

↻ Update Port Value

Output on Port 0

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.

```
; <q2>
jmp start
; data
; code
start: nop
mvi D, 8EH
mvi E, 0F7H
mov A, D
ani 0FH
mov D, A
mov A, E
ani 0FH
mov E, A
xra D
out 00H
hlt
```

Registers			Flag	
A	09		S	0
BC	00	A8	Z	0
DE	0E	07	AC	0
HL	00	00	P	1
PSW	00	00	C	0
PC	42	14		
SP	FF	FF		
Int-Reg	00			

≡ I/O Ports

0

− +

09

↻ Update Port Value

Output of XOR on Port 0

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.

```
; <q3>

jmp start

;data
;code
start: nop
mvi B, 91H
mvi C, 87H
mov A, B
ani 01H
mov B, A
mov A, C
ani 01H
mov C, A

hlt
```

Registers			Flag	
A	01		S	0
BC	01	01	Z	0
DE	0E	07	AC	1
HL	00	00	P	0
PSW	00	00	C	0
PC	42	11		
SP	FF	FF		
Int-Reg	00			

B & C with the everything other than D0 masked

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.

```
; <Program title>
jmp start
; data
; code
start: nop
; reset carry
stc
cmc
mvi B, 0FFH
incr B
mov A, B
jnc skip
mvi A, 01H
skip: nop
out 00H
hlt
```

Registers			Flag	
A	00		S	0
BC	00	00	Z	1
DE	00	00	AC	1
HL	00	00	P	1
PSW	00	00	C	0
PC	42	13		
SP	FF	FF		
Int-Reg	00			

No carry set

I/O Ports

0

− +

00

Update Port Value

Output or B (00H)

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

```
; <q5>

jmp start

; data

; code
start: nop
; 8 bit number
mvi B, 0A5H
mov A, B
ani 0FEH
hlt
```

Registers			Flag	
A	A4		S	1
BC	A5	00	Z	0
DE	00	00	AC	1
HL	00	00	P	0
PSW	00	00	C	0
PC	42	0A		
SP	FF	FF		
Int-Reg	00			

A with lower bit masked off B

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 with the

Following sets of data.

Set1: B=42H, C=69H

Set2: B=69H, C=42H

Set 3: B=F8H, C = 23H

; <q6>

jmp start

; data

; code

start: nop

mvi B, 042H

mvi C, 069H

mov A, B

sub C

jc skip

out 01H

hlt

skip: nop

cma

adi 01H

out 01H

hlt

Registers			Flag	
A	27		S	0
BC	42	69	Z	0
DE	00	00	AC	0
HL	00	00	P	1
PSW	00	00	C	0
PC	42	17		
SP	FF	FF		
Int-Reg	00			

**i** 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="1"/>	<input type="button" value="-"/>	<input type="button" value="+"/>	<input type="text" value="27"/>
<input type="button" value="↻ Update Port Value"/>			

Set 1

Registers

A	27	
BC	69	42
DE	00	00
HL	00	00
PSW	00	00
PC	42	10
SP	FF	FF
Int-Reg	00	

Flag

S	0
Z	0
AC	0
P	1
C	0

Decimal - Hex Conversion

Decimal

0

→ To Hex

Hex

0

← To Dec

I/O Ports

1

- +

27

↻ Update Port Value

Set 2

Registers

A	D5	
BC	F8	23
DE	00	00
HL	00	00
PSW	00	00
PC	42	10
SP	FF	FF
Int-Reg	00	

Flag

S	1
Z	0
AC	0
P	0
C	0

Decimal - Hex Conversion

Decimal

0

→ To Hex

Hex

0

← To Dec

I/O Ports

1

- +

D5

↻ Update Port Value

Set 3