NAME:_____ STD.:_____ DIV.:_____ DATE: PAGE: 13

EXPERIMENT No: 03

ROLL NO: 5139

DATE: 25/3/22

AIM: To find 2's complement using logical instruction.

THEORY:

LOGICAL INSTRUCTIONS

Logical instruction are used to perform the bit operation. Following are the logical instruction of 8086 processor:

1. NOT
2. TEST
3. AND
4. OR
5. X-OR

NOT: Destination

This instruction forms the 1's complement of destination and result stores into destination

Mnemonic: NOT destination

Operation:

Destination ← Destination

Example: MOV CL, 01H

NOT CL

Before Execution CL=01H

0	0	0	0	0	0	0	1

After Execution CL=FEh

1	1	1	1	1	1	1	0

2. AND : Destination, Source

This instruction is used to logically AND's the content of source with destination and result will store into destination

Mnemonic: AND destination, source

Operation:

Destination ← Destination ∧ Source

Example: AND BL, CL

MOV CL, 35h

AND CL, F0h

0 0 1 1	0 1 0 1
1 1 1 1	0 0 0 0
0 0 1 1	0 0 0 0

Clear Lower Nibble

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

3. OR : Destination, Source

This instruction is used to logically OR's the content of source with destination and result will store into destination

Mnemonic: OR destination, Source

Operation:

Destination ← Destination ∨ Source

Example OR BL, CL

NAME: STD.: DIV.: DATE : PAGE : (15)

```
MOV CL, 35h
OR CL, F0h
```

```
  0 0 1 1 | 0 1 0 1
  1 1 1 1 | 0 0 0 0      Set higher Nibble.
  1 1 1 1 | 0 1 0 1
```

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

XOR : Destination, Source

This instruction is used to logically XOR's the content of source with destination and result will store into destination

Mnemonic: XOR destination, Source

Operation:

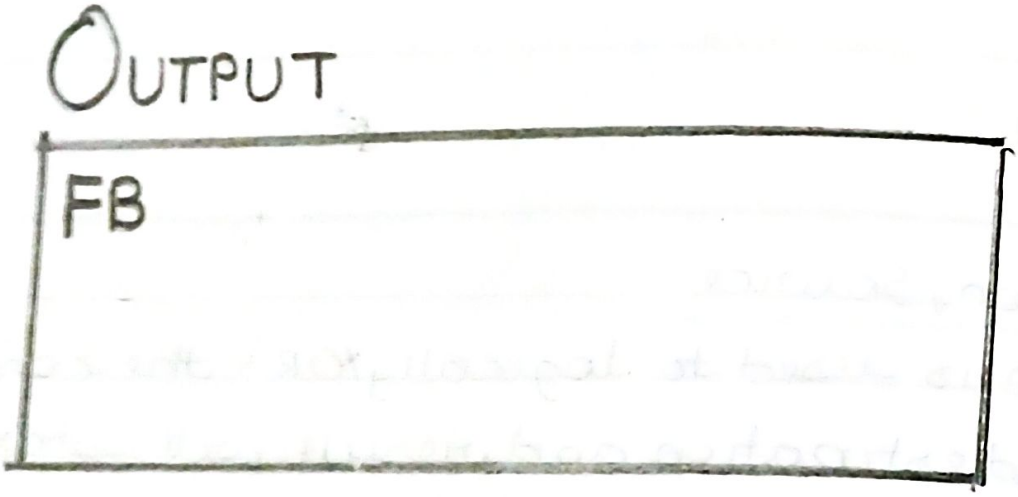
Destination ← Destination ⊕ Source

Example XOR BL, CL

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

```
MOV CL, 35h
XOR CL, 0Fh
```

```
  0 0 1 1 | 0 1 0 1
  0 0 0 0 | 1 1 1 1
  0 0 1 1 | 1 0 1 0
```


OUTPUT
FB

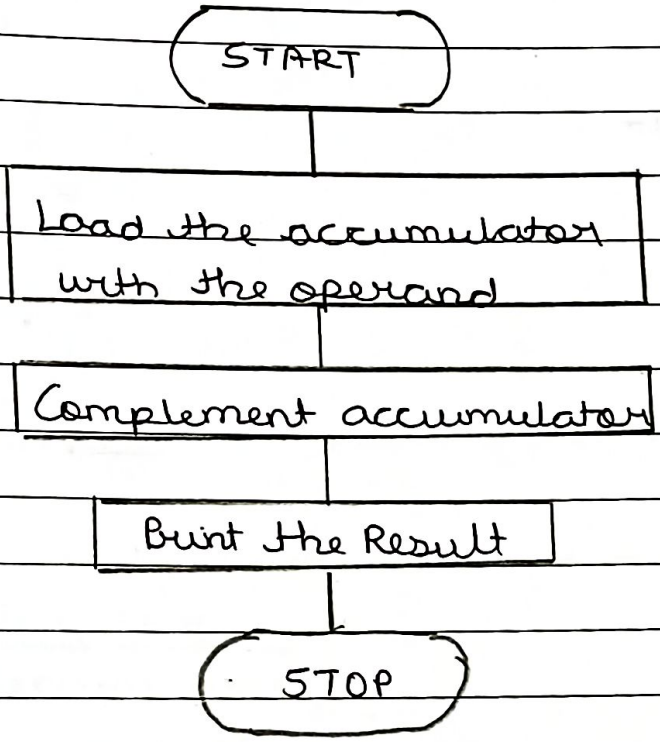
Program 1

Write a program to find 1's complement of given number using logical instruction

→ ALGORITHM:

- Load accumulator with the operand from memory
- Complement the accumulator
- Print the Result.

→ FLOWCHART:



→ PROGRAM:

```
.model small
.data
a db
.code
mov ax, @data
mov ds, ax
mov al, a
not al
mov ch, 02h
mov cl, 04h
```

```
mov bh, al
i2: rol bh, cl
mov dl, bh
and dl, 0fh
cmp dl, 09
jbc i4
add dl, 07h
i4: add dl, 30h
mov ah, 02h
int 21h
dec ch
jnz i2
mov ah, 4ch
int 21h
end
```

Steps to display output:

1. C:\> tasm Filename.asm
2. C:\> tlink filename.obj
3. C:\> filename

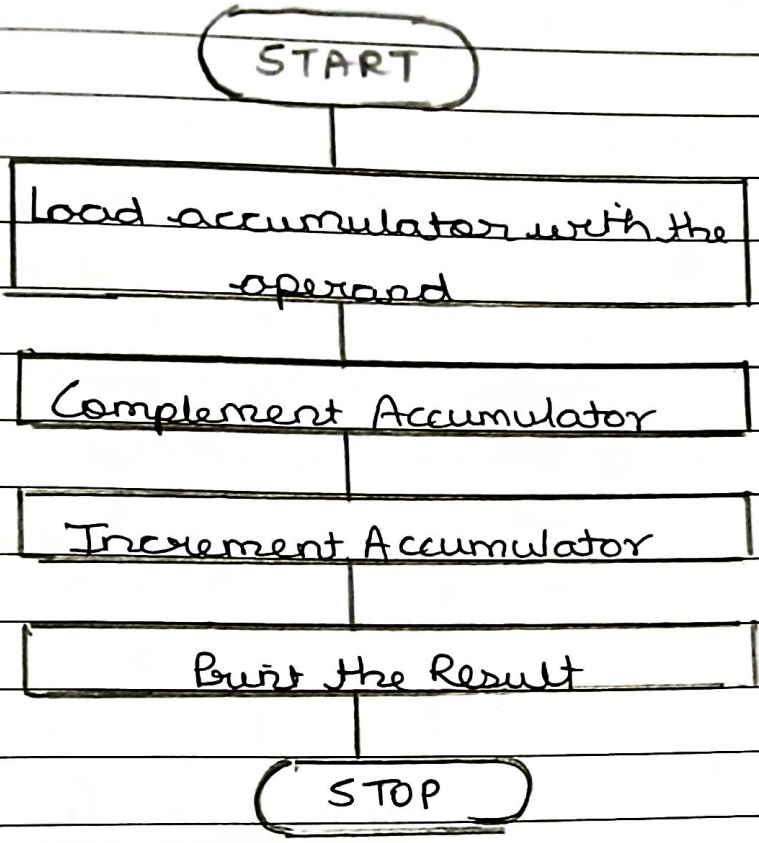
Program 2

- Write a program to find 2's complement of given number using logical instruction.

→ ALGORITHM

- Load accumulator with the 16 or 8 bit operand
- Complement the accumulator
- Increment Accumulator
- Print the Result.

OUTPUT



PROGRAM:

```
.model small
.data
a db 04h
.code
mov ax, @data
mov ds, ax
mov al, a
not al
add al, 1
mov ch, 02h
mov cl, 04h
mov bh, al
i2: rol bh, cl
mov dl, bh
and dl, 0fh
```

```
cmp dl, 09
jbe i4
add dl, 07h
i4: add dl, 30h
mov ah, 02h
int 21h
dec ch
jnz i2
mov ah, 4ch
int 21h
end
```

Steps to display output:

→ C:\> tasm Filename.asm

→ C:\> link Filename.obj

→ C:\> Filename

CONCLUSION

Hence, we implemented various logical instructions and also found 1's complement and 2's complement using logical instruction