Experiment No: 03

Date : ____________

Roll No: ___________

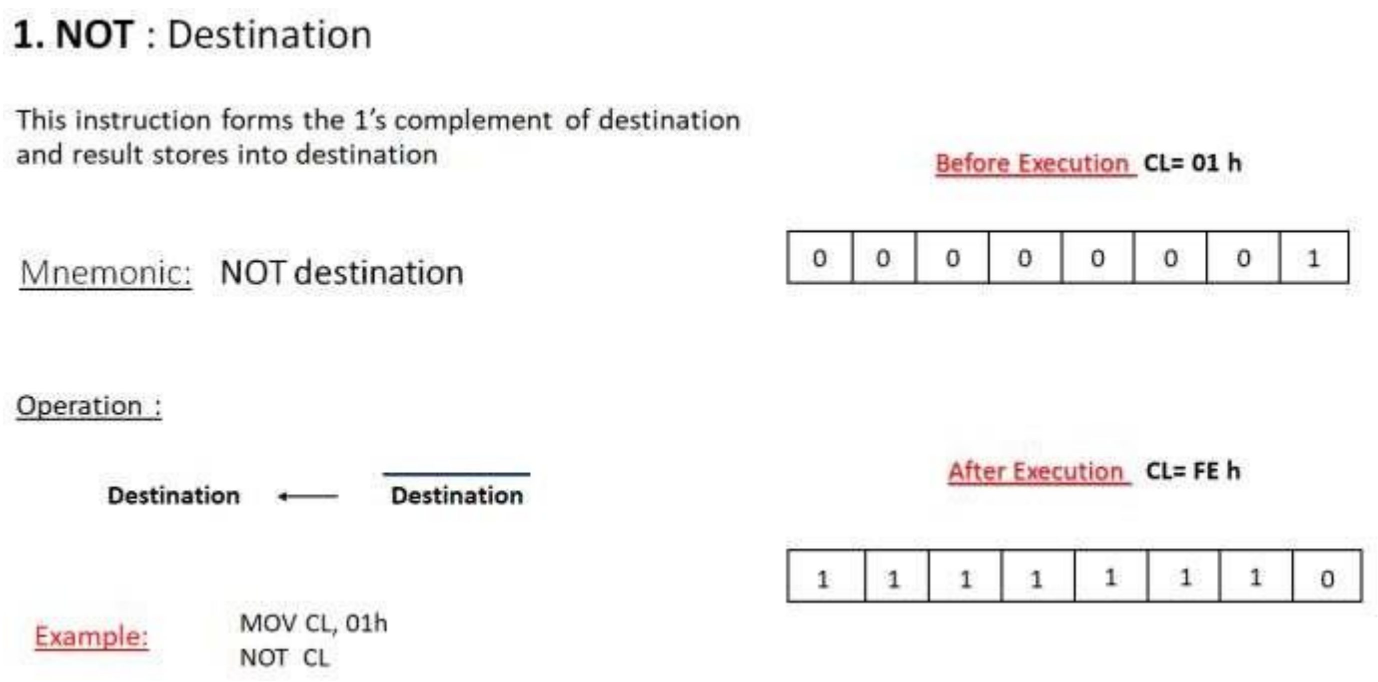
Aim : **To find 2's complement using logical instructions of 8086.**

Theory :

Logical Instructions:

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

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



1. NOT : Destination

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

Mnemonic: NOT destination

Operation :

Destination ← $\overline{\textbf{Destination}}$

Example: MOV CL, 01h
NOT CL

Before Execution **CL= 01 h**

0	0	0	0	0	0	0	1

After Execution **CL= FE h**

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 the destination

Mnemonic: AND destination , Source

Operation :

Destination ← Destination $\wedge$ Source

Example: AND BL, CL

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

Anything AND with 0 will become 0

Anything AND with 1 will remains same

MOV CL, 35 h

AND CL, F0 h

0	0	1	1	0	1	0	1	
1	1	1	1	0	0	0	0	**Clear Lower Nibble**
0	0	1	1	0	0	0	0	

NOTE : data is always starts with number

MOV CL, 35 h

AND CL, 0F0 h

3. **OR** : Destination , Source

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

Mnemonic: OR destination , Source

Operation :

Destination ← Destination $\vee$ Source

Example: OR BL, CL

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

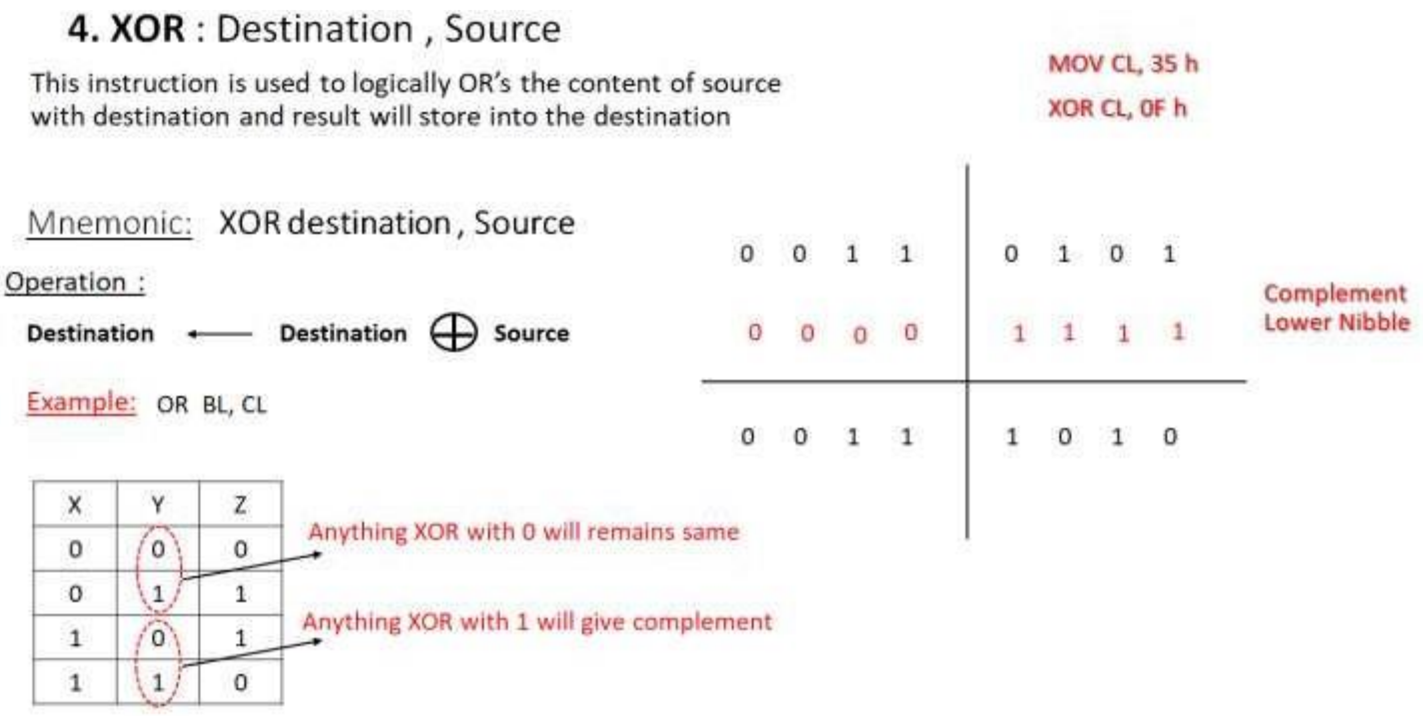
Anything OR with 0 will remains same

Anything OR with 1 will become 1

MOV CL, 35 h

OR CL, 0F0 h

	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



4. XOR : Destination , Source

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

MOV CL, 35 h

XOR CL, 0F h

Mnemonic: XOR destination , Source

Operation :

Destination ⟵ Destination ⊕ Source

Example: OR BL, CL

0	0	1	1	0	1	0	1	
0	0	0	0	1	1	1	1	Complement Lower Nibble
0	0	1	1	1	0	1	0	

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

Anything XOR with 0 will remains same

Anything XOR with 1 will give complement

Program 1: WAP to find 1's complement of given number using logical instruction

Algorithm :

Flowchart :

Program :

Steps to display Output :

Output:

Program 2: WAP to find 2’s complement of given number using logical instruction

(Note : for 2’s complement don’t use NEG instruction)

Algorithm :

Flowchart :

Program :

Steps to display Output :

Output:

Conclusion :