Logic Instructions

- **AND**

  AND destination, source
- **OR**

  OR destination, source
- **XOR**

  XOR destination, source

The results of the above three logic instructions are stored in the destination. The destination must be a register or memory location. The source may be a constant, register or memory location. Memory to memory operations are not allowed.

Effect on flags:

They affect SF, ZF, PF.

AF is undefined.

CF, OF = 0.

Important properties of AND, OR, XOR operations:

b AND 1 = b; b AND 0 = 0;

b OR 1 = 1; b OR 0 = b;

b XOR 1 = -b (complementary) ; b XOR 0 = b;

So –

i) AND can be used to **clear** a specific bit position while preserving the others by using "0" as a mask bit.

ii) OR can be used to **set** a specific bit position while preserving the others by using "1" as a mask bit.

iii) XOR can be used to **complement** a specific bit position while preserving the others by using “1” as a mask bit. A “0” mask bit preserves the other bit positions.

Example-1: Clear the sign bit of AL while leaving the other bits unchanged.

Solution:

AND AL, 7Fh

```
X =   xxxx xxxx
AND 0111 1111  =7F
-------------------------
      0xxx xxxx
```

Example-2: Set the msb and lsb of AL while preserving the other bits.

Solution:

OR AL, 81h

```
AL =   xxxx xxxx
  OR 1000 0001
-------------------------
      1xxx xxx1
```

Example-3: Change the sign bit of DX

Solution:

XOR DX, 8000h

```
X =    xxxx xxxx xxxx xxxx
XOR 1000 0000 0000 0000
--------------------------------------------
      Yxxx xxxx xxxx xxxx
```

- **NOT**

NOT destination

It performs one’s complement of the destination.

It does not affect any flags.

Exercise:

1. MOV BX, 3256H

   MOV CX, 1554H

   AND CX, BX

   HLT

What are the contents of BX and CX?

2. MOV BX, 3A56H

   MOV AX, 1504H

   XOR AX, BX

   HLT

What is the content of AX?

3. MOV AX, 1027H

   MOV BX, 5A27H

   TEST AX, BX

   HLT

What are the contents of AX and BX?

4. SHR BL, 1 where BL = 65H.

0110 0101 >>1 = 0011 0010 = 32

What is the content of BL?

5. Suppose AX = 6165H and BL = 09H.

Write a code in assembly language to complement the 3rd bit of AX and to set the LSB of BL.

```
X =    xxxx xxxx xxxx xxxx
XOR 0000 0000 0000 0100   = 0004 h
--------------------------------------------------------------
       xxxx xxxx xxxx x0xx
```

```
X = xxxx xxxx
OR 0000 0001  =01H
-----------------------------------------
     xxxx xxx1
```

```
XOR AX, 0004H
OR BL, 01H
```

Coding in Assembly Language

Logical instructions:

Logical instructions include NOT, AND, OR, XOR, TEST etc. instructions. Their job is to compare the data values and make results according to logic specified. For example,

- MOV   BX, 30H ; In binary 110000

  NOT   BX   ; In binary 001111

This code takes BX value and then complements all the bits and stores the new value to BX. So it stores 0F value in BX after executing NOT operation. For another example,

-
  ```
  MOV    BX, 70H   ;  In binary 1110000
  MOV    CX, 40H   ;  In binary 1000000
  AND    CX, BX    ;  In binary 1000000
  ```

-
  ```
  MOV    BX, 70H   ;  In binary 1110000
  MOV    CX, 40H   ;  In binary 1000000
  OR     CX, BX    ;  In binary 1110000
  ```

-
  ```
  MOV    BX, 70H   ;  In binary 1110000
  MOV    CX, 40H   ;  In binary 1000000
  XOR    CX, BX    ;  In binary  0110000
  ```

Test operation is a little different from AND operation. It performs bit by bit AND operation but it does not change any operands value.

-
  ```
  MOV    BX, 70H   ;  In binary 1110000
  MOV    CX, 40H   ;  In binary 1000000
  TEST   CX, BX    ;  In binary  CX value is 1000000
  ```

-
  ```
  MOV    BX, 70H   ;  In binary 1110000
  MOV    CX, 40H   ;  In binary 1000000
  AND    CX, BX    ;  In binary  1110000
  ```

Example 1.

```
CODE SEGMENT
ASSUME CS:CODE, DS:CODE
```

```
MOV    BX, 3256H
MOV    CX, 1554H
AND    CX, BX
HLT
CODE ENDS
END
```

Example 2.

```
CODE SEGMENT
ASSUME CS:CODE, DS:CODE
MOV    BX, 3256H
MOV    CX, 1554H
XOR     CX, BX
HLT
CODE ENDS
END
```

Example 3.

```
CODE SEGMENT
ASSUME CS:CODE, DS:CODE
        MOV    AX, 1027H
        MOV    BX, 5A27H
        MOV    CX, 54A5H
        OR     AX, BX
        XOR    AX, CX
        NOT    AX
        TEST   CX, BX
        AND    CX, AX
```

```
      HLT
CODE ENDS
END
```

JUMP Commands:

Sometimes it is necessary to go from one line of program to another line without executing some intermediate lines. For this Jump commands are used. We can explain this with a simple example.

```
MOV    AX, 3254H
MOV    BX, 1F4BH
MOV    CX, 412AH
ADD    AX, CX
JMP    L3T2
SUB    AX, BX
L3T2:    AND    AX, BX
HLT
```

In this example L3T2 is a level. As we can see in fifth line JMP command is used. It makes the program to go from fifth line to L3T2 level that is seventh line. So sixth line is not executed.

There are two types of Jump commands. These are (i) Conditional jump and (ii) Unconditional Jump. Previous example is an unconditional jump. Conditional Jumps are like if statements. If some flags are affected only then these jump instructions executed. We can look at the following example,

```
MOV    AX, 125BH
MOV    BX, 125BH
MOV    CX, 412AH
SUB    AX, BX
JZ     L3T2
```

```
DIV    BX
L3T2:    AND    AX, CX
HLT
```

In fourth line subtraction operation is performed. As both AX and BX have same value. Their subtracted value is 0. So ZF is set to 1. In fifth line JZ L3T2 is written. It means if ZF = 1 then go to L3T2:. Otherwise continue. As ZF = 1, program moves to eighth line. This is a conditional Jump. Some other conditional jumps are,

Command	**Condition of Jump**
JA/JNBE jump if above / jump if not below or equal	CF =0, ZF = 0
JBE/JNA jump if below or equal / jump if not above	CF = 0 or ZF = 0
JNB/JAE/JNC jump if not below / jump if above or equal / jump if not carry	CF = 0
JB/JNAE/JC	CF = 1
JZ/JE	ZF = 1
JNZ/JNE	ZF = 0

Example 4.

```
CODE SEGMENT
ASSUME CS:CODE, DS:CODE
            MOV    AX, 7A24H
            MOV    BX, 15A3H
            SUB    AX, BX
            JMP    L3T2
EEE323: DIV    BX
             JMP    Last
L3T2:      MOV    CX, 45B1H
             AND    AX, CX
             TEST   AX, BX
             JMP    EEE316
```

```
Last:     HLT
CODE ENDS
END
```

Example 5.

```
CODE SEGMENT
ASSUME CS:CODE, DS:CODE
          MOV    AX, 7A24H
          MOV    BX, 95A3H
          ADD    AX, BX
          JC    L3T2
EEE316:  OR    AX, 23H
          JNZ    Last
L3T2:     MOV    CX, 0FC7H
          SUB    AX,CX
          JZ     EEE316
Last:     HLT
CODE ENDS
END
```