3.2.3 Logical Instructions

Logical instructions are bitwise instructions operating on the individual bits. Typical logical operations include logical complement (**NOT**), logical and (**AND**), logical or (**OR**), and logical exclusive or (**XOR**).

i. OR X,Y ; X = X + Y (X **OR** Y).

The truth table of **OR** gate can be shown below:

X	**Y**	**O / P**
0	0	0
0	1	1
1	0	1
1	1	1

Examples:

OR AX, BX

OR CX, 30h

OR BL, DH

OR BYTEPTR[5000h], 7Fh

Important note: Here we can use **OR** for ***setting bits*** of any data by putting **1's** against the bits that want to be set & the others are **0's**.

Example 7: Set the 2'nd bit of reg. AL:

AL = x x x x x x x x

02 = 0 0 0 0 0 0 1 0

we put 1 in the position of the 2'nd bit

So the solution will be:

OR AL, 02h

ii. AND X,Y ; X = X . Y (X **AND** Y).

The truth table of **AND** gate can be shown below:

X	**Y**	**O / P**
0	0	0
0	1	0
1	0	0
1	1	1

Examples:

AND AX, BX

AND CX, 30h

AND BL, DH

AND BYTEPTR[5000h], 7Fh

Important note: Here we can use **AND** for ***resetting(clearing) bits*** of any data by putting **0's** against the bits that want to be reset & the others are **1's**.

Example 8: Reset the 4'th and 11'th bits of reg. CX:

CX = x x x x x x x x x x x x x x x x

FBF7 = 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1

we put 0's in the positions of 4'th and 11'th bits

So the solution will be:

AND CX,FBF7h

iii. XOR X,Y ; X = X ⊕ Y (X **XOR** Y).

The truth table of **XOR** gate can be shown below:

X	Y	O / P
0	0	0
0	1	1
1	0	1
1	1	0

Examples:

XOR AX, BX

XOR CX, 30h

XOR BL, DH

XOR WORDPTR[BX+SI+20h], 756Fh

Important note: Here we can use **XOR** for ***inverting bits*** of any data by putting **1's** against the bits that want to be inverted and the others are **0's**.

Example 9: Invert the 1'st & the 8'th bits of reg. DL:

DL = x x x x x x x x

81h = 1 0 0 0 0 0 0 1

we put 1's in the positions of 1'st & 8'th bits

So the solution will be:

XOR DL,81h

iv. NOT X ; Take the 1's complement of X.

X	O / P
0	1
1	0

Examples:

NOT SI ; invert all bits of reg. SI

NOT BL ; invert all bits of reg. BL

NOT WORDPTR[SI+1234h] ; invert all bits of the content of the 16-bit of the memory locations specified by [SI+1234] as the low byte and [SI+1235] as the high byte.

Example 10:: what is the value of AL after executing the following instruction:

MOV AL, C3h ; AL = 1100 0011 b

NOT AL

AL = 0011 1100 b = 3Ch

Example 11: Rebuild the following instruction without using OR instruction.

OR AL,BL .

We can rebuild the instruction by using **Boolean Algebra** for the expression of or gate:

$$O/P = AL + BL = \overline{\overline{AL} \cdot \overline{BL}}$$

So the solution will be:

NOT AL

NOT BL

AND AL,BL

NOT AL

Important note: the **NOT** instruction does not take the *immediate addressing mode*.

The general formats for the logical instructions are shown in table below:

Mnemonics	Meaning	Format	Operation	Flags affected
OR	Logical OR	OR D, S	(D) . (S) ⟶ (D)	CF, OF, SF, PF, ZF AC undefined
AND	Logical AND	AND D, S	(D) + (S) ⟶ (D)	CF, OF, SF, PF, ZF AC undefined
XOR	Logical Exclusive OR	XOR D, S	(D) ⊕ (S) ⟶ (D)	CF, OF, SF, PF, ZF AC undefined
NOT	Logical NOT	NOT D	(D) ⟶ $\overline{(D)}$	

H.W. 1: trace the following program:

MOV AX, 0324h

MOV BX, 0203h

OR AX, BX

AND BX, 3C2F

XOR AX, DCAB

NOT BH

HLT

Assume all flags (Z, S, P) are initially zeros

H.W. 2: write an A.L.P. to express the following logical expression:

X= $\overline{(Y \oplus Z)}$. (W+55)

Where X is a M.L. with an offset of 0500h. Assume Y=33h, Z=50h, and W=05h.

H.W. 3: write an instructions that do the following:

Rebuild

i. **XOR AX, BX**
ii. **AND CX, SI**
iii. **XOR AX, FF**
iv. **NOT BYTEPTR [3500h]**