-Multiplication instructions (MUL , iMUL)

*in 8086 MP , multiplication can be done on bytes (8 –bit) or on words (16 –bit).

*there are two basic types of multiplication instruction , one for unsigned integer number (MUL) and the other for singed integer numbers (iMUL).

*the product after multiplication is always a double width product if two 8 –bit numbers are multiplied , they generate a 16-bit product . and if two 16 –bits numbers are multiplied they generate a 32 –bit product .

*the destination operand is always implicit within the multiplication instruction and it is always represented by the accumulator register .

Mnemonic	*Meaning*	*Format*	*Operation*	*Flags*
MUL	*Multiply (unsigned)*	*MUL s*	*(AL) . (S8) ⟶ (AL)* *(AX) . (s16) ⟶ (DX)(AX)*	*OF,SF* *CF,ZF*
iMUL	*Multiply (signed)*	*iMUL*	*(AL).(S8) ⟶ (AL)* *(AX).(S16) ⟶ (DX)(AX)*	*AF,PF*

The source operand can be a register or memory location .

EX /

MUL CL (AX)=(AL)*(CL)

iMUL DH (AX)=(AL)*(DH)

MUL BYTE PTR [BX] (AX)=(AL)*((DS)0 + (BX))

MUL Di (DX).(AX)=(AX)*(Di)

iMUL WORD PTR [si] (DX).(AX) = (AX)*((DS)0 + (Si))

example / write an assembly language program that multiply two 8-bits numbers stored in BL and CL and store the result in DX

sol / *MOV AL , CL*

MUL BL

MOV DX , AX

-Division instructions (DIV , IDIV)

***as with multiplication , division occurs on 8 –bits or 16 –bits Numbers . these numbers can be integer signed (iDiv) or unsigned numbers (Div).**

***for division operation again just the source operand is specified the other operand is either the contents of AX for 16-bit dividends or the contents of both DX and AX for 32-bit dividends .**

Mnemonic	*Meaning*	*Format*	*operation*	*Flags*
Div.	*Division (unsigned)*	*DiV S*	*Q((AX)/(S8)) (AL)* *R((AX)/(S8)) (AH)* *Q((DX,AX)/(S16) (AX)* *R((DX,AX)/(S16) (DX)*	*CF* *AF* *ZF* *SF* *OF* *OF*
iDiV	*Integer divide (signed)*	*iDiV S*	*=*	

***where Q stand for quotient and R for remainder .**

***the source operand can be a general purpose register or a memory location contents .**

EX /

Div CL (AL) = Q ((AX)/(CL))

(AH) = R ((AX)/(CL))

iDiv Si (AX) = Q ((DX , AX)/(Si)

(DX) = R ((DX , AX)/(Si))

***a division can result in two different types of errors these are :-**

1- an attempt to divide by zero .

2- a divide overflow that occur when a small number divides into a large number . for example , suppose that AX = 3000 and that it is divided by 2 , because the quotient for an 8-bit division appear in

AL ,the result of 1500 causes a dived overflow because 1500 does not fit into AL .

Example / if the contents of AL is $(-1)_{10}$ and the contents of CL is $(-2)_{10}$, what will the result produced in AX by executing the instructions

MUL CL and iMUL CL

SOL /

The first instruction multiplies AL & CL as unsigned numbers

$-1_{10} = 11111111_2 = FF_{16}$

$-2_{10} = 11111110_2 = FE_{16}$

Thus executing MUL CL result

$(AX) = (11111111)_2 * (11111110)_2 = 1111110100000010$

$= FD02_{16}$

While executing iMUL CL gives

$(AX) = 1_{16} * 2_{16} = 2_{16}$

3-logic instructions

The 8086 microprocessor has instruction for performing the logic operations AND,OR,XOR,AND NOT these instruction performs their respective logic operations bit by bit on the specified source and destination operands.

MNEMONIC	meaning	format	operation	Flags affected
AND	Logical	AND D,S	(S).(D) → (D)	CF,AF
OR	Logical	OR D,S	(S)+(D) → (D)	ZF,SF
XOR	Logical	XOR D,S	(S)⊕(D) → (D)	OF
NOT	logical	NOT D	$\overline{(D)}$ → (D)	PF

The source/ destination operands can be:-

reg ↔ reg

reg ↔ memory *for not instruction the destination

reg ← immediate operand can be a register or a memory

memory ← immediate

EX:-

AND AL,BH (AL)=(AL).(BH)

AND DI , 4FFFH (DI)=(DI).4FFFH

OR CX ,[DI] (CX) = (CX) + ((DI) 0+ (DI))

XOR DH, 0A3H (DH) = (DH)⊕A3H

NOT BX (BX)=($\overline{BX}$)

NOT [SI] ((DS) 0 + (SI)) = $\overline{\text{((DS) 0 + (SI))}}$

***the AND instruction can be used for masking off a specified operand bits by anding with 0's e.g.**

AND AL ,00000011B this will mask off the 6 mast significant bits of Al.

H.W give the following sequence of instructions specify the contents of AL register at each instruction

MOV AL,01010101B

AND AL,00011111B

OR AL,11000000B

XOR AL,00001111B

NOT AL

EX write an assembly language program that performs the following Boolean operation:-

DI=AX.BX$\oplus$ (AX.$\overline{CX}$+DX)

SOL

NOT CX $\overline{CX}$

AND AX,CX AX.$\overline{CX}$

OR AX,DX AX.$\overline{CX}$+DX

AND BX,DX AX.BX

XOR AX,BX AX.BX$\oplus$ (AX.$\overline{CX}$+DX)

MOV DI,AX DI

EX:-

write a program that clear bits 0and 1 of cx ,sets bits 9 and 10 of cx ,and invert bit 12 of cx?

SOL

AND CX,0FFFCH clear bits 0&1

OR CX,0600H set bit 9&10

XOR CX,1000H invert bit 12

NOTE AND with 0 used to clear ,or with1 used to

Set and Xor with 1used to invert