3.2.2.3 Multiplication instructions: MUL and IMUL

The **MUL** (for unsigned numbers) and **IMUL** (for signed numbers) instructions are used to perform the multiplication operation between two operands: the *source* is external and the *destination* is internal. The source may be either register or memory with data either 8-bit or 16-bit. The internal destination could be either **AX** for an 8-bit source operand or the registers pair **DX AX** for a 16-bit source operand. This operation is explained in table below:

Multiplication (MUL or IMUL)	Internal operand	External operand (source)	Result
Byte * Byte	AL	Register or BYTEPTR of memory	AL*source = AX
Word * Word	AX	Register or WORDPTR of memory	AX*source = DX AX

The general format for the multiplication instructions is shown in table below:

Mnemonics	Meaning	Format	Operation	Flags affected
MUL	Multiply	MUL S	AL*S8 → AX AX*S16 → DX AX	CF, OF AC, SF, PF, ZF undefined
IMUL	Integer multiply	IMUL S	AL*S8 → AX AX*S16 → DX AX	CF, OF AC, SF, PF, ZF undefined

Where Q stands for Quotient and R stands for reminder.

Examples:

MUL BL ; AX=AL*BL.

IMUL CH ; AX=AL*CH.

MUL WORDPTR[BX+DI+1234h] ; DX AX=AX*The content of M.L. (16-bit) with an offset of (BX+DI+1234h) as the low byte from ((BX+DI+1234h) and the high byte from (BX+DI+1235h).

IMUL BYTEPTR[5000h] ; AX=AL*The content of M.L. (8-bit) with an offset of (5000h).

MUL CX ; DX AX=AX*CX.

Important note: the **MUL** and **IMUL** instructions do not take the *immediate addressing mode*.

Now to understand the difference between MUL and IMUL instruction, first load **IMUL AH** and put FFFFh in register AX. As a signed byte, FF= -1 so (-1 * -1 = +1). Trace the instruction and see that AX=0001h. now load **MUL AH** and again set AX=FFFFh. Now FF=255, so 255*255=65025 or in Hex. FF*FF=FE01h. tracing this instruction gives AX=FE01h, as expected. Only for byte factors less than 128 do MUL and IMUL give the same results.

Example 4: write an A.L.P. that multiplies two 16-bit numbers stored in BX and CX and store the result in M.L.s specified by [2342h]?

Solution:

```
MOV AX, BX
MUL CX
MOV [2342h], AX
MOV [2344h], DX
HLT
```

H.W. 1: write an A.L.P. that multiplies two 16-bit numbers stored in 2 consecutive M.L.s specified by [BX] then store the result in the next 4 M.L.s followed the 2 consecutive numbers.

3.2.2.4 Division instructions: DIV and IDIV

The **DIV** (for unsigned numbers) and **IDIV** (for signed numbers) instructions are used to perform the division operation between two operands: the *source* is external and the *destination* is internal. The source may be either register or memory with data either 8-bit or 16-bit. The internal destination could be either **AX** for an 8-bit source operand or the registers pair **DX AX** for a 16-bit source operand. This operation is explained in table below:

Division (DIV or IDIV)	**Internal operand**	**External operand (source)**	**Result**
Word / Byte	AX	Register or BYTEPTR of memory	AX / source = AX Q (AL) R(AH)
DWord / Word	DX AX	Register or WORDPTR of memory	(DX AX) / source = DX AX Q (AX) R (DX)

The general format for the multiplication instructions is shown in table below:

Mnemonics	Meaning	Format	Operation	Flags affected
DIV	Divide	DIV S	AX/S8 ⟶ Q(AL) R(AH) DX AX/S16 ⟶ Q(AX) R(DX)	CF, OF, AC, SF, PF, ZF undefined
IDIV	Integer divide	IDIV S	AX/S8 ⟶ Q(AL) R(AH) DX AX/S16 ⟶ Q(AX) R(DX)	CF, OF, AC, SF, PF, ZF undefined

Examples:

DIV BL ; AX/BL ⟶ Q(AL) and R(AH).

IDIV CH ; AX/CH ⟶ Q(AL) and R(AH) .

DIV WORDPTR[BX+DI+1234h] ; (DX AX)/(The content of M.L. (16-bit) with an offset of (BX+DI+1234h) as the low byte from ((BX+DI+1234h) and the high byte from (BX+DI+1235h)) ⟶ The Q(AX) and R(DX).

IDIV BYTEPTR[5000h] ; AX/(The content of M.L. (8-bit) with an offset of (5000h)) ⟶ the Q(AL) and R(AH).

DIV CX ; (DX AX)/CX ⟶ The Q(AX) and R(DX).

Important note: the **DIV** and **IDIV** instructions do not take the *immediate addressing mode*.

The division operation can result in two errors:

1. Division by zero.
2. Division overflow when a large 16-bit no. is divide by a small 8-bit no.. for example, the division of 4000h by 04h, because the result of an 8-bit division must be fill in AL, while the result of the last operation is 1000h that does not fit into AL.

To divide an 8-bit dividend by and an 8-bit divisor by extending the sign bit of AL to fill all bits of AH. This can be done automatically by executing the instruction (**CBW**).

In a similar way, a 16-bit dividend in AX can be divided by 16-bit divisor. In this case the sign bit in AX is extended to fill all bits of DX. The instruction **(CWD)** performs this operation automatically.

Note that **CBW** extends 8-bit in AL to 16 bit in AX while the value in AX will be equivalent to the vale in AL. similarly, **CWD** converts the value in AX to 32-bit in (DX AX) without changing the original value. The general format of thes instructions is shown in table below:

Mnemonics	Meaning	Format	Operation	Flags affected
CBW	Convert byte to word	CBW	**Copy** the MSB of AL to **all bits** of AH	None
CWD	Convert word to double word	CWD	**Copy** the MSB of AX to **all bits** of DX	None

Example 4: write an A.L.P. to divide the 40 by 3 then explain its operation.

Solution:

```
MOV AX, 0028h       ; AX=0028h
MOV BL, 03h         ; BL=03h
DIV BL              ; the explaining is below.
HLT
```

The Q of the division is 13 and in the Hex. it is D so AL=0Dh

The R of the division is 1 so AH=01.

The final result in AX=010Dh.

Example 5: what is the value of AX after executing the following instructions:

```
MOV AX, 0000h
MOV AL, FBh              ; AX=00FBh
CBW
```

Since the MSB of AL=1 then, all bits of AH=1's. So the value of AH=FFh. Then AX=FFFBh

Example 6: what is the value of AX and DX after executing the following instructions:

```
MOV DX, 0000h
MOV AL, FFFBh            ; DX AX = 0000   FFFB
CBW
```

Since the MSB of AX=1 then, all bits of DX=1's. So the value of DX AX= FFFF FFFB.

H.W. 2: trace the following program:

MOV AX, 0324h

MOV BX, 0203h

MUL BL

DIV BH

CWD

HLT

Assume all flags are initially zeros

H.W. 3: write an A.L.P. to express the following equation:

$$X = \left(2*Y + \frac{Z}{5} \right) - W^2$$

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