SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Instruction set of 8086

The Instruction set of 8086:-

1. Data transfer instructions
2. Arithmetic instructions
3. Logical Instructions

Data transfer instructions

A. General purpose instructions:

- **MOV** – Used to copy the byte or word from the provided source to the provided destination.

-syntax- MOV destination,source

- Ex:-MOV AX,BX; contents of bx moved to ax.
- **PUSH** – Used to put a word at the top of the stack.
- Ex:-push AX; contents Ax reg moved to the stack.
- **POP** – Used to get a word from the top of the stack to the provided location.
- Eg:-pop Ax; contents from top of stack moved to AX.
- **XCHG** – Used to exchange the data from two locations.
- Eg:-XCHG ax,bx

Data transfer instructions..contd

B. <u>Instructions for input and output port transfer</u>

- **IN** – Used to read a byte or word from the provided port to the accumulator(AX register).
- **OUT** – Used to send out a byte or word from the accumulator to the provided port.

C. <u>Instructions to transfer the address</u>

- **LEA** – Used to load the address of operand into the provided register.
- **LDS** – Used to load DS register and other provided register from the memory
- **LES** – Used to load ES register and other provided register from the memory.

Data transfer instructions..contd

D. <u>Instructions to transfer flag registers</u>

- **LAHF** – Used to load AH with the low byte of the flag register.
- **SAHF** – Used to store AH register to low byte of the flag register.
- **PUSHF** – Used to copy the flag register at the top of the stack.
- **POPF** – Used to copy a word at the top of the stack to the flag register.

Instruction :POP Destination

- This instruction copies a word from memory location pointed by SS:SP to the destination-LIFO
- The segment register used with SP is SS , and no segment override is possible (use some other **segment** register than the default **segment** for a particular code).
- When it is executed:

  I. Data from memory location (in stack segment) SS:SP is first copied to the destination

  II. And then SP is incremented by 2 (SP=SP+2)

  Note : destination can be a general purpose register, Flag register ,segment register or memory location .

  EX: POP AX (store the value of stack in AX)

  POP DS

Instruction :PUSH Source

When this instruction is executed:

i) SP=SP-2 (i.e. First SP is decremented by 2)

ii) Copy the word from source register to the location in Stack Segment where SP points.

Note :1**.**Source operand (16-bit) can be a general purpose register, Flag register ,segment register or memory .

2. Addressing mode – register Indirect

EX: PUSH AX (content of AX is moved to stack)

PUSH BX

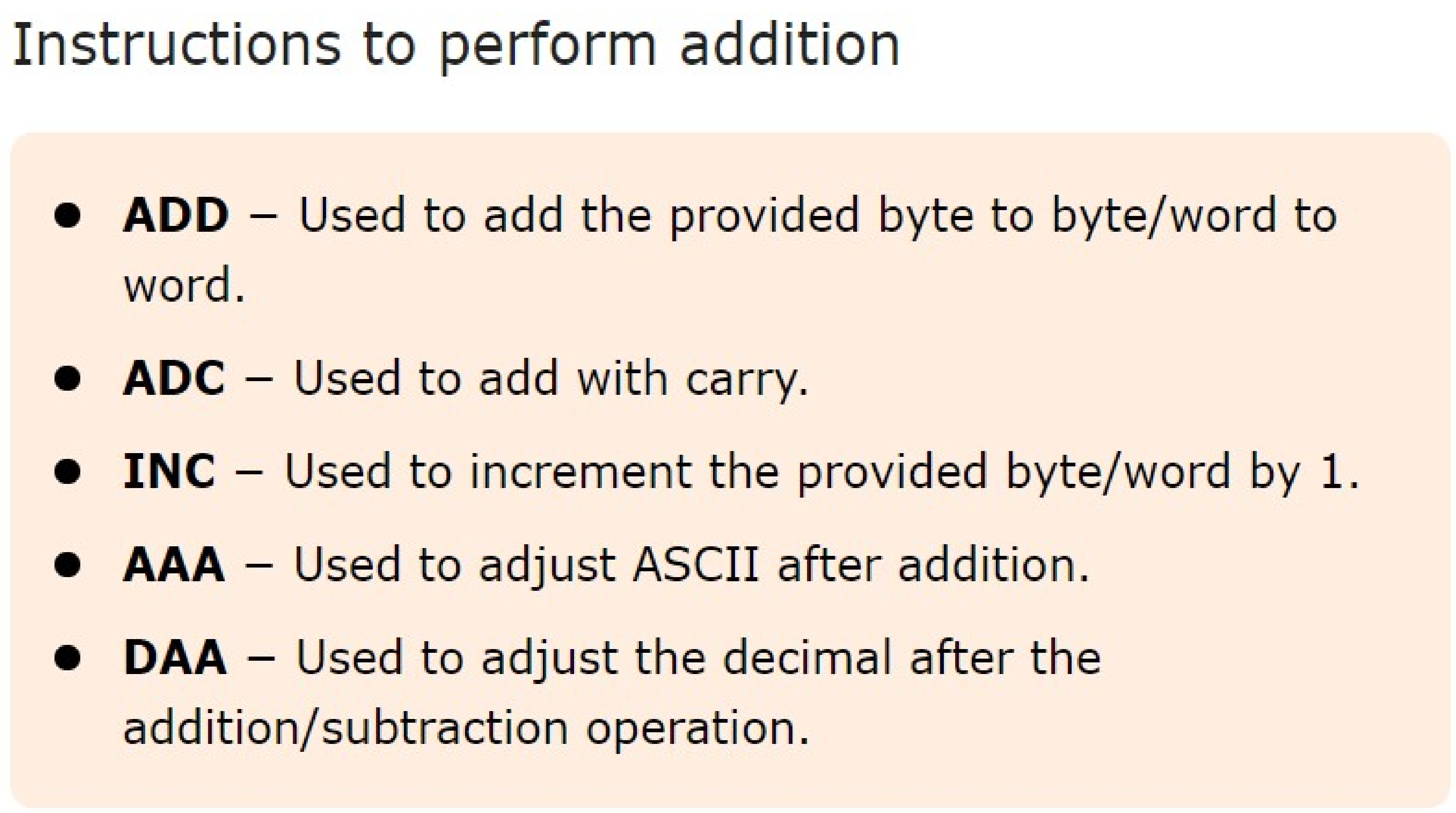
PUSH DS

XCHG

- XCHG Reg2,Reg1
- Contents of Reg1 are exchanged with Reg2.
- Both Reg’s should be of same size.
- Both can be ether 8 bit or 16 bit.
- Eg:-XCHG AL,BL
- XCHG [BX],AX
- BX-pointing register also

ARITHMETIC INSTRUCTIONS

1.instructions to perform addition



Instructions to perform addition

- **ADD** – Used to add the provided byte to byte/word to word.
- **ADC** – Used to add with carry.
- **INC** – Used to increment the provided byte/word by 1.
- **AAA** – Used to adjust ASCII after addition.
- **DAA** – Used to adjust the decimal after the addition/subtraction operation.

ADDITION instructions..cntd

- **ADD**

Ex ADD AX,BX

AX<= AX+BX

- **ADC**

Ex ADD AX,BX

AX<= AX+BX+CF

- **INC**

Ex: INC CL (content of CL +1)

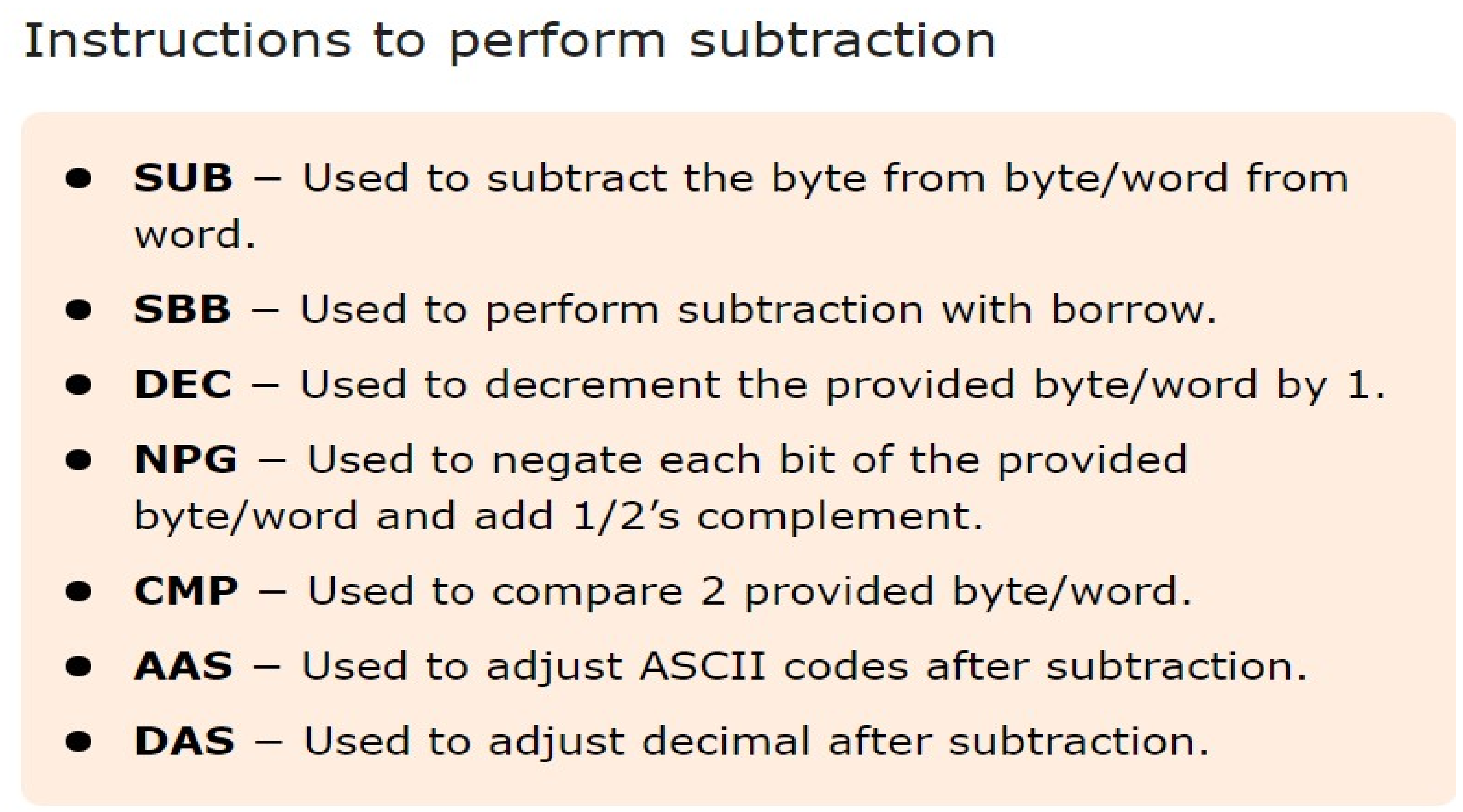
CL<= CL+1

INC BX

BX<= BX+1

- **AAA**
- **DAA**

2.Instructions to perform subtraction



Instructions to perform subtraction

- **SUB** – Used to subtract the byte from byte/word from word.
- **SBB** – Used to perform subtraction with borrow.
- **DEC** – Used to decrement the provided byte/word by 1.
- **NPG** – Used to negate each bit of the provided byte/word and add 1/2's complement.
- **CMP** – Used to compare 2 provided byte/word.
- **AAS** – Used to adjust ASCII codes after subtraction.
- **DAS** – Used to adjust decimal after subtraction.

SUBTRACT INSTRUCTIONS

- 1.SUB
- Eg:-SUB ax, bx ;   Subtracts bx from ax and stores result in ax.
- 2.SBB ax,bx ; Subtracts bx and borrow(carry flag) from ax and stores result in ax.
- 3.CMP ax,bx ; comparison is done by subtracting bx from ax and checking the flags.

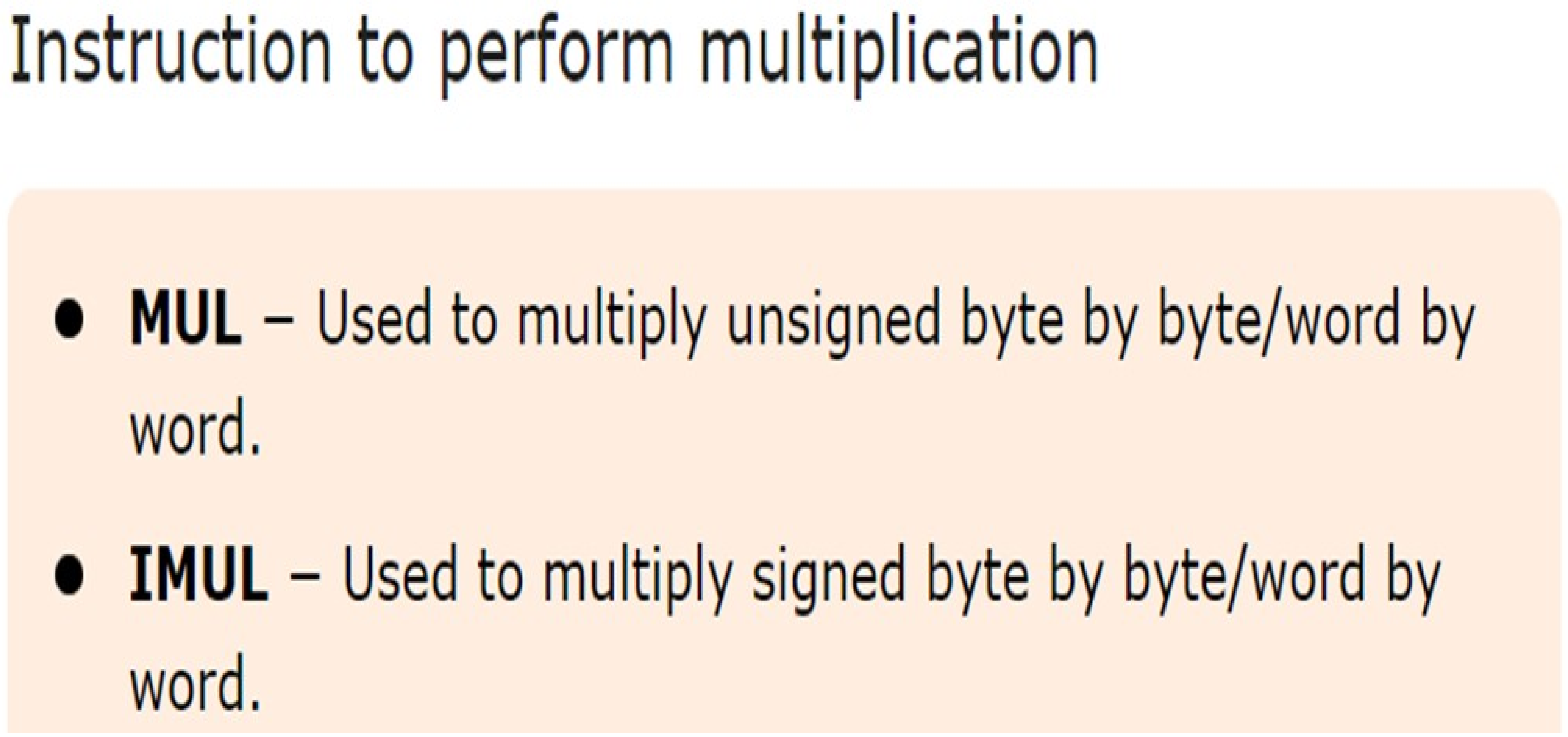
-If ax<bx –carry flag set

-ax=bx, zero flag set

SUBTRACT INSTRUCTIONS

- 4.dec
- Eg: DEC ax- Decrements the value of ax by 1.

3.Instruction to perform Multiplication



Instruction to perform multiplication

- **MUL** – Used to multiply unsigned byte by byte/word by word.
- **IMUL** – Used to multiply signed byte by byte/word by word.

MULTIPLICATION

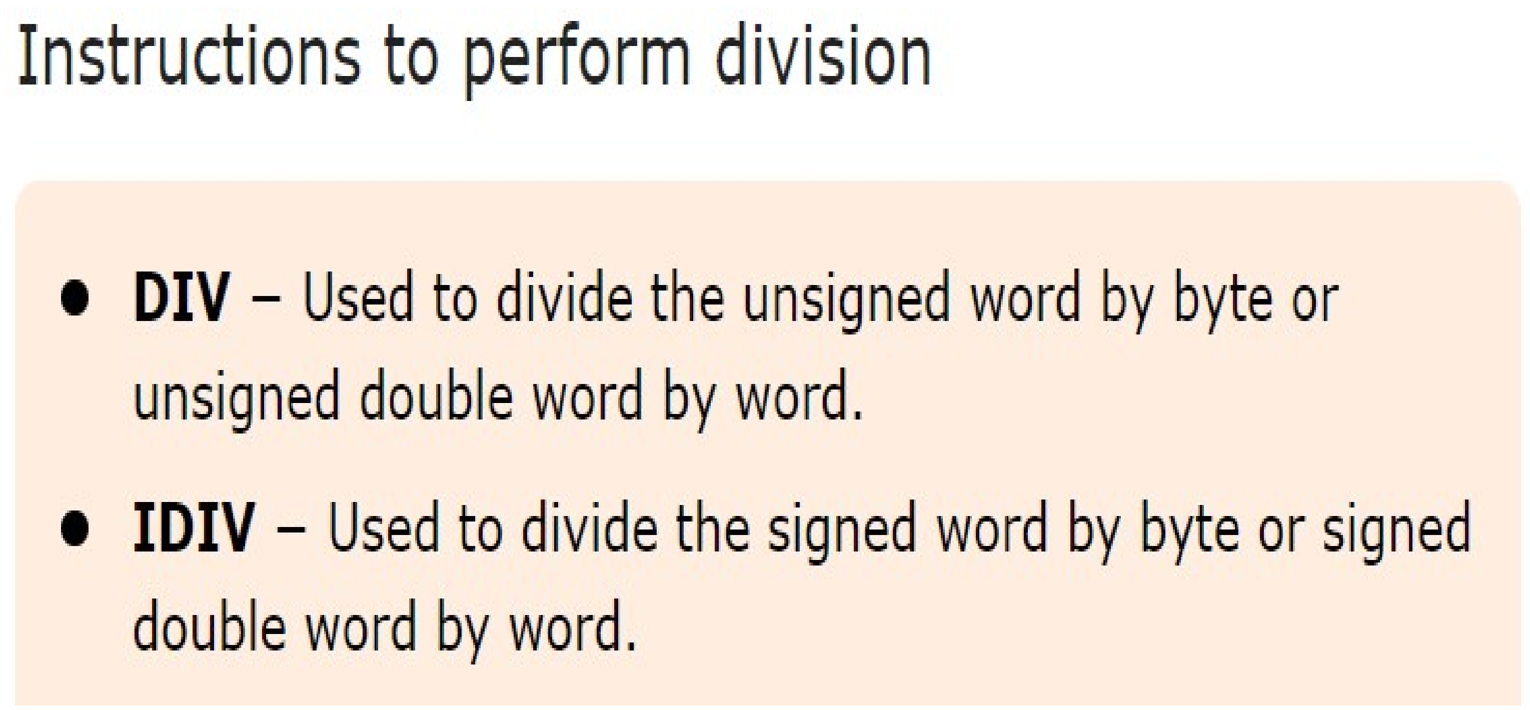
- Eg:-1.Mul bl

Multiplies AL by BL and stores the result in AX.

2.Mul bx

Multiplies ax with bx and stores the lower word of the result in Ax and higher word in DX.

4.Instrutions for Division



Instructions to perform division

- **DIV** – Used to divide the unsigned word by byte or unsigned double word by word.
- **IDIV** – Used to divide the signed word by byte or signed double word by word.

DIVISION

- Eg:-Div BL

-AL is divided by BL and the quotient is stored in AL and the reminder in AH register.

- Eg:-Div BX

-Ax is divided by BX and the Quotient is stored in AX and the reminder in DX.

Logical Instructions

Instructions to perform logical operation

- **NOT** – Used to invert each bit of a byte or word.
- **AND** – Used for adding each bit in a byte/word with the corresponding bit in another byte/word.
- **OR** – Used to multiply each bit in a byte/word with the corresponding bit in another byte/word.
- **XOR** – Used to perform Exclusive-OR operation over each bit in a byte/word with the corresponding bit in another byte/word.