

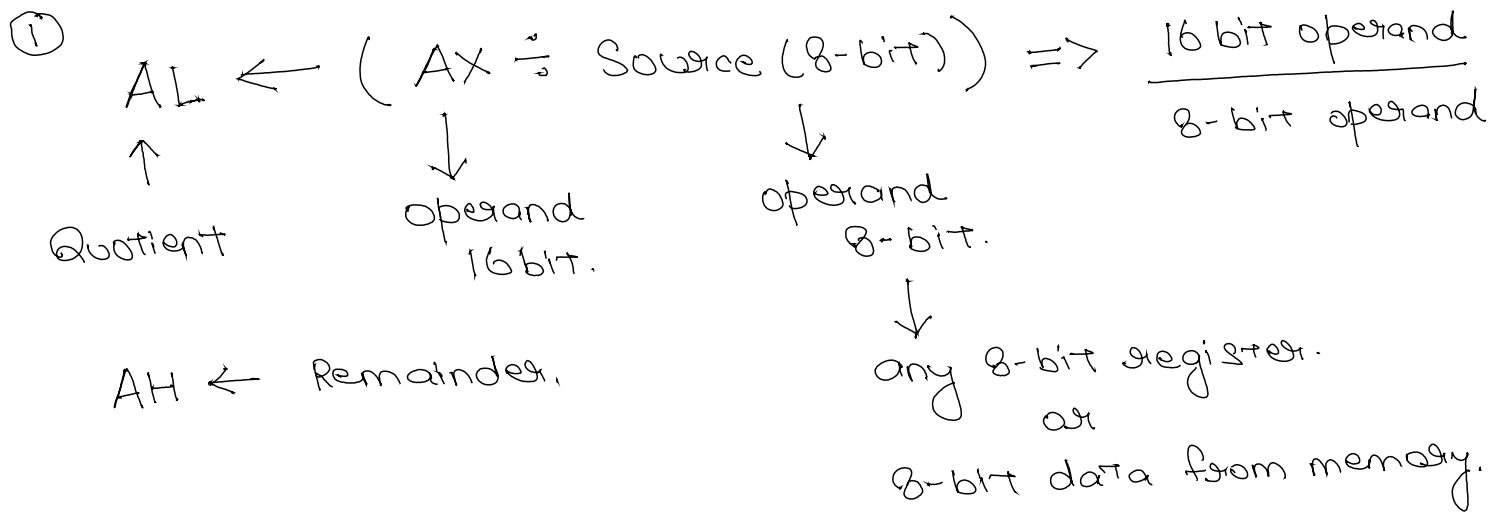
30/12/2021

8085 Microprocessor

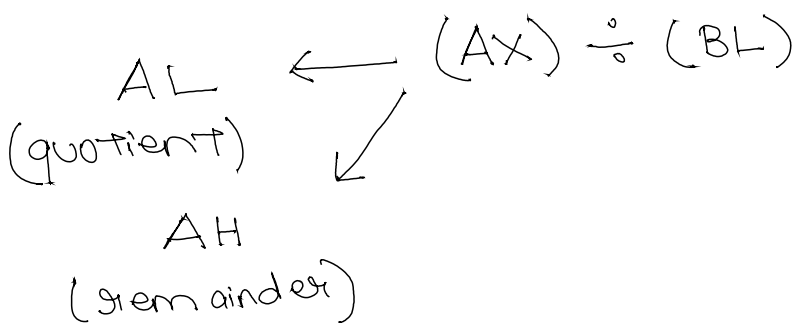
Divide Instruction

DIV Source

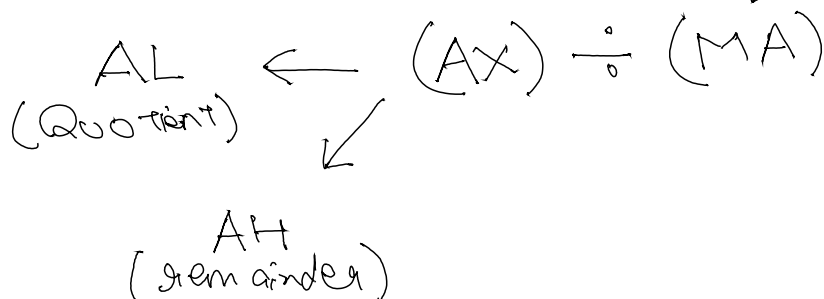
→ Division of 16 bit or 32 bit number by.
8 bit or ~~bit~~ 16 bit number.



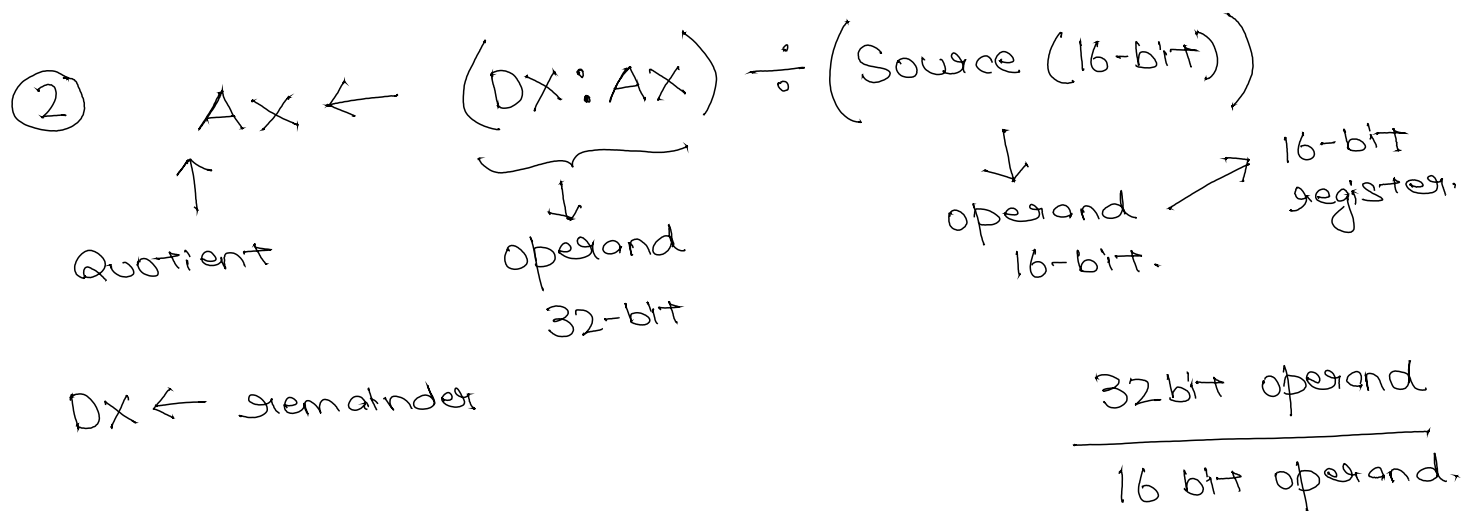
For eg: ① DIV BL



② DIV [0248H]

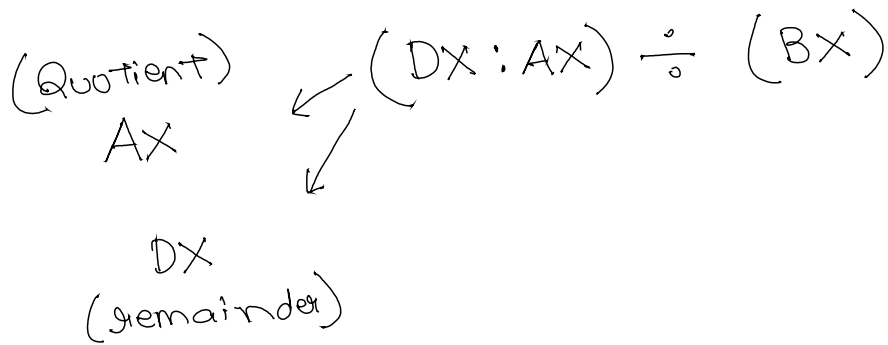


$$MA = BA + EA \\ = DS \times 16_{10} + 0248$$



For eg:

DIV BX



Note :- ① If the quotient is larger than the size of AL register (in case of 16-bit/8-bit) or the size of AX register (in case of 32-bit/16-bit) then a divide by zero condition occurs which is handled by a special interrupt.

② If denominator in a division is zero then also divide by zero interrupt occurs.

→ DIV is used for unsigned division operation.
IDIV is used for signed division operation.

DAA (Decimal adjust after addition)

↳ NO operand

↳ operation: $AL \leftarrow (AL \text{ adjusting for BCD addition})$.

MOV AL, 54H

AL = 54H.

MOV BL, 26H

BL = 26H.

1
54
26
80

ADD AL, BL

DAA

* DAA will always be used after addition instruction only.

0101 0100 → AL

0010 0110 → BL

0111	1010	7AH.
------	------	------

not valid BCD.

+	0000	0110
<hr/>		
1	000	0000
<hr/>		
8	0	

DAS (Decimal adjust after subtraction)

DAS instruction after subtraction instruction to adjust the result for BCD value.

Logical Instructions of 8086

→ Basic logic operations: NOT, AND, OR, XOR

→ Bit by Bit shift operation such as

SHL (Shift logical left)

SHR (Shift logical right)

SAL (Shift arithmetic left)

SAR (Shift arithmetic right)

→ Rotate operations

ROR (rotate right without carry)

ROL (rotate left without carry)

RCR (rotate right through carry)

RCL (rotate left through carry)

→ Compare Instructions.

NOT destination (is complement of destination)

↓
register
memory.

For eg: MOV AL, 0AH
NOT AL

(BE)
AL = 0000 1010

AL = 1111 0101
(AE)

AND destination, Source (Logical AND)

operation: destination operand \leftarrow (destination operand) AND (source operand)

bitwise logical AND operation.

Source operand \rightarrow Immediate data ; Register ; memory.

destination operand \rightarrow Register ; memory.

For eg:

MOV BL, 0FH

MOV CL, 0FH

AND CL, BL

BL = 0000 1111 \rightarrow AND

CL = 1111 0000 \rightarrow

CL = 0000 0000

OR destination, source (Logical OR operation)

operation: destination operand \leftarrow (destination operand) OR (source operand)

\downarrow Register memory \downarrow Immediate data Register memory.

For eg:

MOV CX, 01F1H

OR CX, 0001H

CX = 0000 0001 1111 0001
1111 0000 0000 0001

1111 0001 1111 0001

\swarrow F 1 F 1

CX after OR instruction.

XOR destination, Source (Exclusive logical OR)

operation: $(\text{destination operand}) \leftarrow (\text{destination operand}) \oplus (\text{source operand})$

↓

Register
memory

↓

Immediate data
Register
memory.