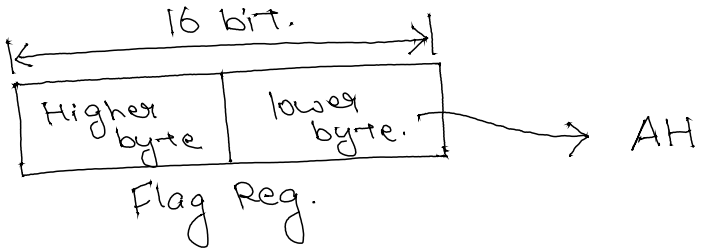
22/12/2021

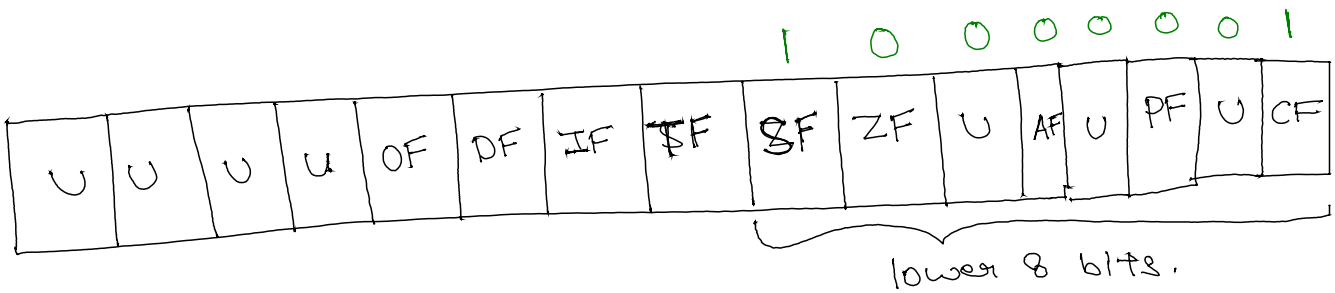
Microprocessor

LAHF → Loads the lower flag register byte into AH register.



→ NO operand, implied addressing.

For eg:



LAHF

After executing

81H → AH

SAHF → Store the AH register value (8-bit) into lower byte of flag register.

AH → lower byte of flag register.

IN Instruction :- (Input data from I/O device)

① IN port-address
↳ 8-bit operation.

AL ← [port-address]

eg: IN 08H
↳ 08H is the address of I/O device.

8-bit value from I/O device with address 08H will be copies to AL register.

② IN [DX]
↳ DX register will be having the address of I/O device.

→ 16-bit operation then value from I/O device goes to AX register.

→ 8-bit operation value from I/O device goes to AL register.

Out Instruction :- (Out the data to I/O device)

* 8 bit data from AL register can be sent to the I/O device whose address is specified directly or indirectly (using DX reg.)
* 16 bit data from AX register can be sent to the I/O device.

LEA Instruction :-

* Load effective address

Eg: LEA AX, [BX]

↳ effective address → AX.

Difference between MOV and LEA.

MOV AX, [BX]

MA = BA + EA.

= DS x 16_{10} + BX

= 2000 x 16_{10} + 0423H.

= 20423 H.

↓

AX.

Value at MA is copied to AX.

LEA AX, [BX].

MA = BA + EA

= DS x 16_{10} + BX

↓

AX.

Eg: LEA CX, [BX+SI+02H]

CX ← EA

EA = BX + SI + 02

MOV CX, [BX+SI+02H]

MA = BA + EA

CX ← (MA)value.

LDS Instruction

→ Load Data segment.

→ LDS instruction stores four consecutive memory locations into a specified destination register and DS register.

→ The 16-bit value (first word) from memory is loaded into the register specified in the instruction and second word from memory is loaded to DS register.

→ LDS reg., memory.

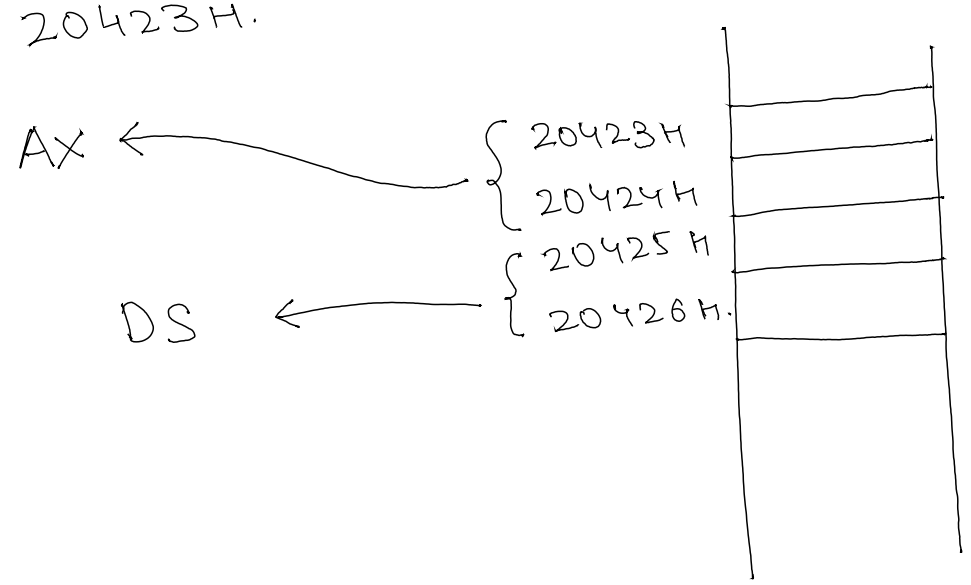
For eg: LDS AX, [BX]

MA = BA + EA

= DSX16$_{10}$ + BX

= 2000X16$_{10}$ + 0423H

= 20423H.



LES Instruction

→ Load Extra Segment

→ Similar to LDS, only difference is instead of Data segment (DS). data will loaded from memory to Extra segment (ES).

XLAT instruction

→ Translate byte in AL register by table look-up.

→ NO operand, belongs to implied addressing mode.

→ The BX register contains the effective address of the lookup-table.

→ AL register contains the displacement.

→ value of MA which is calculated as

DS : BX + AL

is ~~mo~~. ~~copit~~ copied to AL reg.