27/12/2021

MICRO PROCESSOR

STACK OPERATION OF 8086

- * In case of switching between functions, subroutines, procedure, information such as variables, register values, status of accumulator or flag register is stored on the stack.

- * CONTEXT SWITCHING.

- * PUSH → stores value or data onto stack.
- * POP → retrives or load data from stack.

→ 8086 Physicall address generation of stock.

$$MA = BA + EA$$

$$MA = SS \times 16_{10} + SP$$

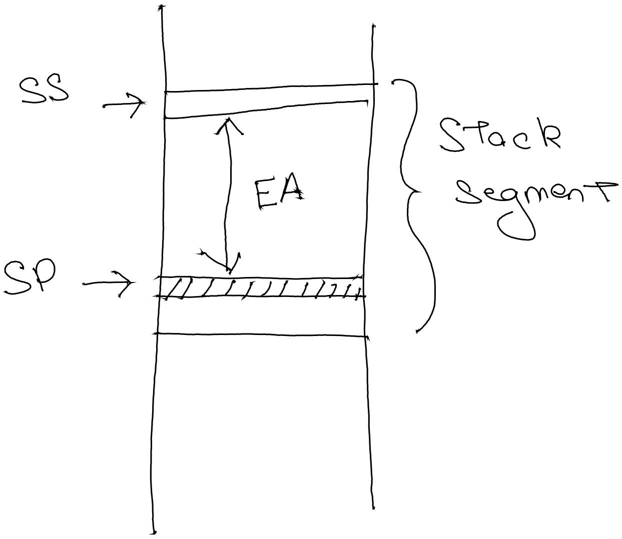
MA : Physical memory add.

BA : Base add.

EA : Effective add.

Eg:- SS = 5000 H.
SP = 2400 H.

$$MA = SS \times 16_{10} + SP$$
$$= 5000H \times 16_{10} + 2400H$$
$$= 50000H + 2400H$$
$$= 52400\ H.$$



PUSH INSTRUCTION → Store 16-bit value in Stack memory.

PUSH Reg.

↳ 16-bit register → AX ; BX ; CX ; DX

$DX = \boxed{DH \mid DL}$

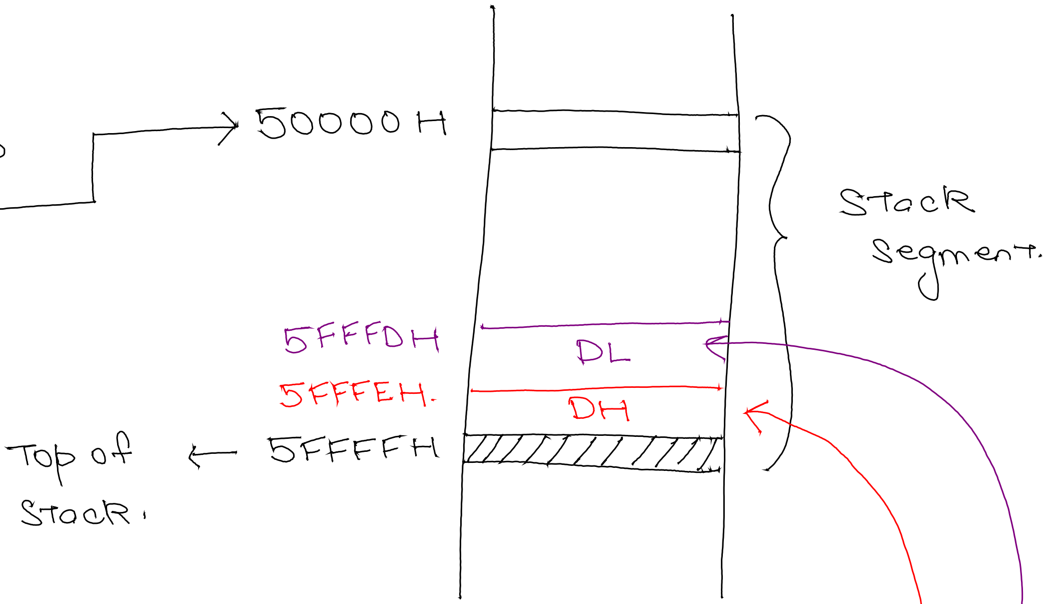
$\boxed{\text{PUSH DX}}$

Let us suppose while executing this instruction SS = 5000H and SP = FFFFH.

SS = 5000H

MA = 5000H × 16_{10}

= 50000H.



MA = BA + EA

= SS × 16_{10} + SP ←

= 50000 + FFFF

= 5FFFF H.

① When PUSH DX instruction is executed,

SP = SP − 1

Top of stack = 5FFFEH.

DH value will be stored.

② SP = SP − 1

Top of stack = 5FFFDH.

DL value will be stored.

After execution → Top of stack = 5FFFDH.

DH → stored in 5FFFEH DL → store in 5FFFDH.

POP Instruction → Get 16-bit value from Stack.

POP Reg.

↳ 16-bit register.

Let us assume that after executing the PUSH DX instruction seen before.

$\boxed{\text{POP AX.}}$

SS = 5000H

MA = 5000×16₁₀

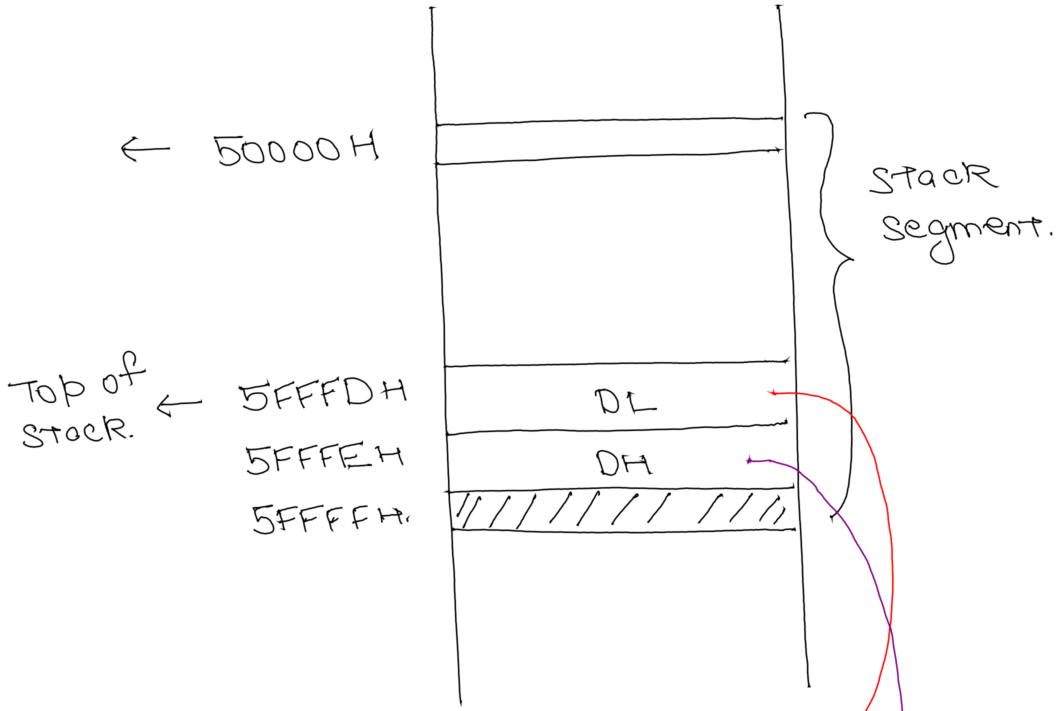
= 50000H

MA = BA + EA

= SS×16₁₀ + SP

= 50000 + FFFDH

= 5FFFDH.



→ When POP AX is executed

① AL ← SS:[SP]

Top of Stack. (5FFFDH)

AL reg.

SP = SP+1

② AH ← SS:[SP]

Top of Stack. (5FFFEH)

AH.

SP = SP+1

After execution : AH ← DH ; AL ← DL ; Top of Stack = 5FFFFH.

→ Only 16-bit operation is supported by PUSH and POP instruction.

→ Register addressing mode.
Direct addressing mode
Indirect addressing mode
Based addressing mode
Indexed addressing mode
Based - indexed " "
} are supported.

PUSH operand.
↓
16-bit
↓
register
memory.

PUSH [BX]
Push [BX + 08H]
Push [BX+SI +20H]

PUSH AX
PUSH DX
PUSH BX

$\underbrace{\text{PUSH [BX]}}$
↓
16-bit value from data segment of memory will be push to stack.

$\underbrace{DS \times 16_{10} + BX}$

MA → Top of stack.

→ SP = SP-1

$\underbrace{DS \times 16_{10} + BX + 1}$

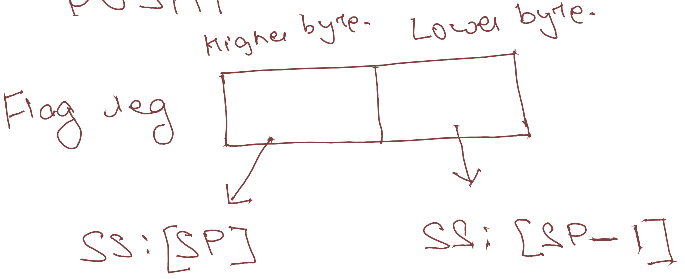
MA → top of stack.

→ SP = SP-1

PUSHF → Store flag reg. in the stack.

→ No operand.

Eg: PUSHF



PUSHA Instruction

→ No operand

→ Push all general purpose register in the stack.

AX
CX
DX
BX
SP
BP
SI
DI.

GPR in this sequence are stored in the stack.

→. Original value of SP is stored (before execution of this instruction).

POPF → Get value from stack into flag register.

POPA → Get value from stack into general purpose register.

DI
SI
BP
SP
BX
DX
CX
AX.

GPR in this sequence are poped.