

## The Stack

The stack is a temporary storage originally used to preserve return addresses when a subroutine is called. It is so convenient for temporary storage that it is used to save the contents of certain registers or some other main program parameters.

The instruction that is used to save parameters on the stack is **PUSH** instruction and the instruction that is used to retrieve them from the stack is the **POP** instruction. The standard formats of these instructions are shown in the following table:

Mnemonics	Meaning	Format	Operation	Flags affected
<b>PUSH</b>	Save word onto stack	PUSH S	S $\longrightarrow$ M.L. specified by SP	None
<b>POP</b>	Retrieve word off stack	POP D	M.L. specified by SP $\longrightarrow$ D	None

### Important Note:

- The allowed operands are (general purpose registers, a segment register (excluding CS), or a storage location in memory).
- PUSH and POP instructions always store or retrieve **WORDS** of data only.

The work of stack is organized in the form of First In Last Out (FILO) or Last In First Out (LIFO).

### Ex.s:

**PUSH AX**

**PUSH [BX+50h]**

**POP DS**

**POP [SI]**

## Offsets of The Stack

### i. SP register

Represents the *top of the stack* and it is affected by every PUSH and POP operations.

For PUSH instruction the SP register will be decremented by 2.

$$SP_{\text{new}} = SP_{\text{old}} - 2$$

**Ex.s: Assume SP=5006h, DX=F736h.**

PUSH DX                               ; DH → (SP-1)  
    ; DL → (SP-2)

	XX	[5002h]
	XX	[5003h]
DL	36	[5004h]
DH	F7	[5005h]
Top of stack (SP)	XX	[5006h]

The new value of the SP will be: SP=5004h.

For POP instruction the SP register will be incremented by 2.

$$SP_{\text{new}} = SP_{\text{old}} + 2$$

**Ex.s: Assume SS=3000h, SP=7000h.**

POP AX                               ; (SP+1) → AH  
    ; (SP+2) → AL

$$\begin{aligned} PA &= SS * 10h + SP \\ &= 3000 * 10 + 7000h \\ &= 37000h \end{aligned}$$

Top of stack (SP)	XX	[37000h]
AH ←	55	[37001h]
AL ←	36	[37002h]
	XX	[37003h]
	XX	[37004h]

AX=5536h

SP=7002h

**ii. BP register**

If anyone wants to reach a certain memory location in the stack (i.e. not the top of the stack) then, the BP register will be used. The BP used for the *random access* of a certain M.L. in the stack.

**Ex. 1:** Write A.L.P. to put 507Dh in register CX. Assume the top of the stack is 6040h in a segment of memory which its start is 15DDh using stack operations.

Sol.

MOV AX, 15DDh

MOV SS, AX

MOV SP, 6040h

MOV AX, 507Dh

PUSH AX

POP CX

HLT

**Ex. 2:** You have AX=1234h, BX=3200h, SP=120Ah. What are the values of AX, BX CX, DX, and SP after executing the following program:

PUSH AX

PUSH BX

POP CX

PUSH BX

POP DX

HLT

AX=1234h      BX=3200h      CX=3200h      DX=3200      SP=1208h.

**Ex. 3:** Write an A.L.P. to store the following sequence in the stack: 0, 1, 4, 5.  
Assume the start of the stack segment and offset is 1000:0500.

Sol.

MOV AX, 1000h

MOV SS, AX

MOV SS, 0500h

MOV BX, 0000h

MOV CL, 02h

ROR BX, CL

JC \*

ROL BX, CL

PUSH BX

\*: INC BX

CMP BX, 0006h

JNZ #

HLT