Stack :

The ***Stack*** is an area of memory for keeping temporary data. The SS register contains the address of the beginning of the stack. Initially, the SP register contains the size of the stack , a value that points to the byte past the end of the stack. The stack differs from other segment in its method of storing data. It begins storing data at the highest location in the segment and stores data downward through memory.

The process of storing the data in the stack is called 'pushing into' the stack and the reverse process of transferring the data back from the stack to the CPU register is known as 'popping off' the stack. The stack is essentially *Last-In-First-Out* (LIFO) data segment. This means that the data which is pushed into the stack last will be on top of stack and will be popped off the stack first.

The maximum value of SP=FFFFH and the segment can have maximum of 64K locations. If the SP starts with an initial value of FFFFh, it will be decremented by two whenever a 16-bit data is pushed onto the stack. After successive push operations, when the stack pointer contains 0000H, any attempt to further push the data to the stack will result in stack overflow.

The Stack Segment register (SS) and Stack pointer register (SP) together address the stack-top.

The PUSH and POP instructions are two of a number of instructions that modify the contents of the SP register and are used for storing data on the stack and retrieving it.

PUSH : Store 16 bit value in the stack.

POP :Get 16 bit value from the stack.

The basic syntax of these instructions:

PUSH operand(source)

POP operand(destination)

The operand is 16 bit. These types of operand :

push	REG
pop	SREG
	memory

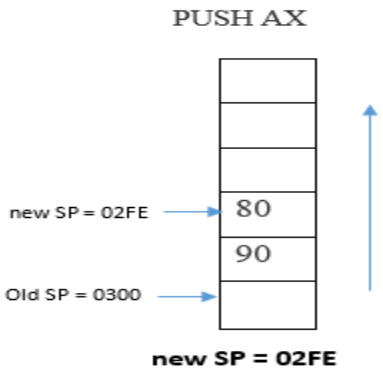
The algorithm of push:

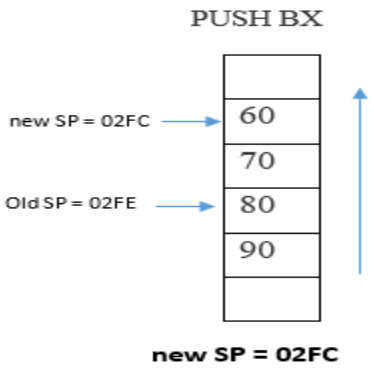
- Subtract 2 from SP register (new SP = old SP – 2) .
- Write the value of operand (source) to the address SS:SP (top of the stack) = operand. Note: write high then low of operand(source).

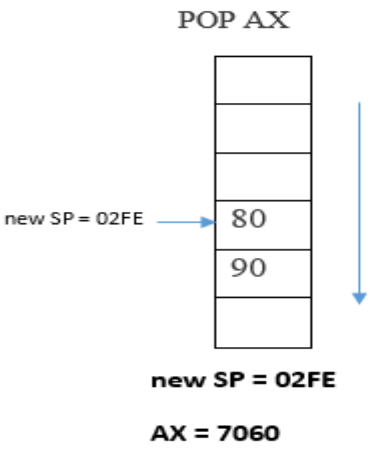
The algorithm of pop:

- Write the value at the address SS:SP (top of the stack)to the operand (destination). Note: write low then high to the destination.
- Add 2 to SP register (SP = SP + 2).

Ex: Find the contains of registers: AX, SB, SP; If you know the value of AX =9080h , BX= 7060h and SP = 0300.

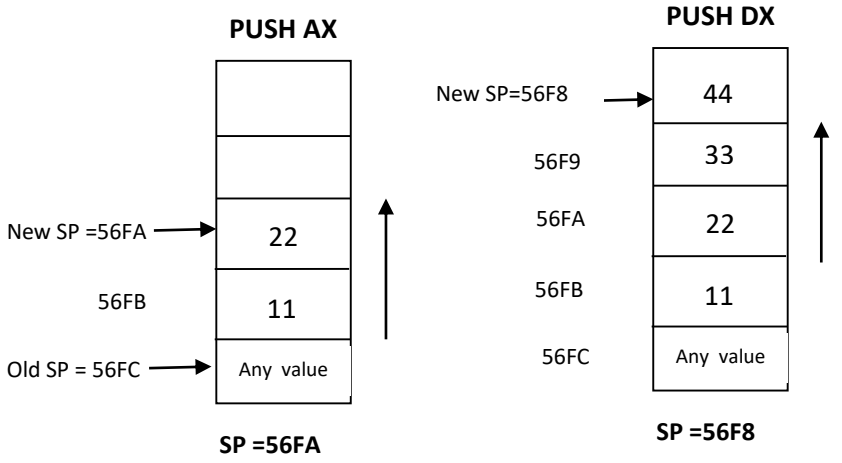






Ex: Find the contains of SP , AX , DX and memory locations (0901h and 0902 h) after execute the following instructions. The value of AX=1122 h , DX=3344 h , SP = 56FC h , SI = 0901 h and the content of 0901and 0902 is FF66 h respectively.

```
PUSH AX
PUSH DX
INC AX
DEC DX
POP AX
POP DX
PUSH AX
POP [SI]
```




```
INC AX      ; AX = 1123
DEC DX      ; DX = 3343
```