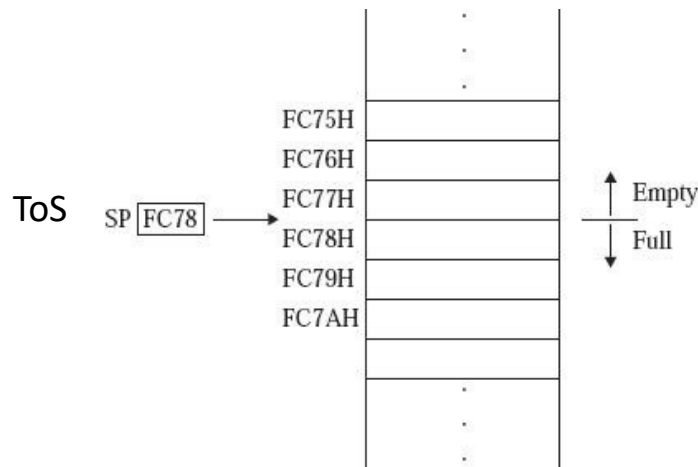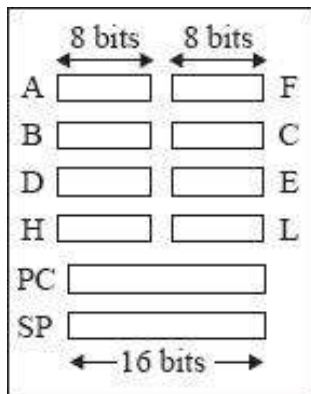The stack is a LIFO (last in, first out) data structure implemented in the RAM area and is used to

a. store addresses and data when the microprocessor branches to a subroutine(could also be a ISR)

b. swap values of two registers and register pairs we use the stack as well.

The Stack Pointer register will hold the address of the top location of the stack.

PUSH the SP register gets **decreased** by 2 and new data item used to insert on to the top of the stack.

POP operation, the data item will have to be deleted from the top of the stack and the SP register will get **increased** by the value of 2.

An example program to exchange content of HL register pair with DE register pair

Steps

1.Initialize stack pointer (SP) by 2FFF.
2.Push the content of H and L register into the stack.
3.Push the content of D and E register into the stack.
4.Pop the upper two bytes from top of stack and place it in HL register.
5.Pop the remaining two bytes from top of stack and place it in DE register.

Program:
2000: LXI H 2050H
2003: LXI D 2A52H
2006: LXI SP 2FFFH
2009: PUSH H ⟶ SP--;[SP] = H;
                SP--;[SP] = L;
200A: PUSH D
200B: POP H
200C: POP D ⟶ [SP] = E; SP++;
                [SP] = D; SP++;
200D: HLT

Alternate Code:
LXI H 2FFFH
SPHL
LXI H 2050H
LXI D 2A52H
PUSH H
PUSH D
POP H
POP D
HLT

USING STACK in a Subroutine

PUSH PSW
```
SP <- SP - 1 M[SP] <- A
SP <- SP - 1 M[SP] <- F
```

POP PSW
```
F <- M[SP] SP <- SP + 1
A <- M[SP] SP <- SP + 1
```

Chalk and Talk

RET
Pops the [SP]
To PC

2000	PUSH PSW	Push value of accumulator and flag in stack
2001	POP H	Pop value from TOP of memory stack in H
2002	MOV C, H	C <- H
2003	MOV H, B	H <- B
2004	MOV B, C	B <- C
2005	PUSH H	Push the value of register H
2006	POP PSW	Pop value of flag register and Accumulator
2007	HLT	END