# Data Transfer Instruction (doesn’t affect flags)

- **Question (Classwork)**
  - Write a program in 8085 to read a content of memory with address 4400H and load the value to register D.

```
LXI H, 4400H
MOV D, M
```

  - Write a program in 8085 to load register B and L with value 3BH and 7FH respectively. Then store the value of registers B and L in memory A010H and B100H respectively.

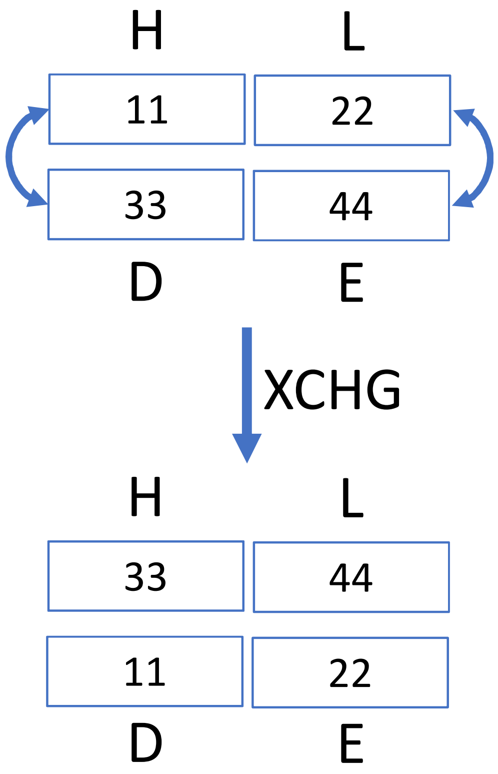
```
MVI B, 3BH
MVI L, 7FH
MOV A, B                        LXI H, A010H
STA A010H;   [A010] <- A        MOV M, B;   [A010] <- B
MOV A, L                        LXI H, B100H
STA B100H;   [B100] <- A        MOV M, L;   [B100] <- L
```

# Data Transfer Instruction (doesn’t affect flags)

- **Exchange the value of HL and DE**
  - Instruction: **XCHG**; H <-> D, L <-> E

    **(1 byte instruction)**

    e.g. LXI H, 1122H; H <- 11H, L <- 22H<br>
    LXI D, 3344H; D <- 33H, E <- 44H<br>
    XCHG; H <-> D, L <->E

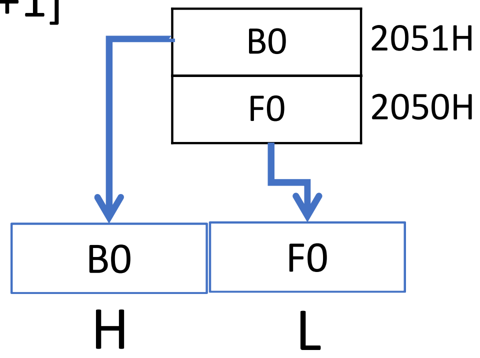


# Data Transfer Instruction (doesn’t affect flags)

- **Load/Store HL directly**
  - Instruction: **LHLD** 16-bit address; L <- [address], H <- [address+1]

    **(3 bytes instruction)**

    e.g. LHLD 2050H; L <- [2050H], H<- [2051H]

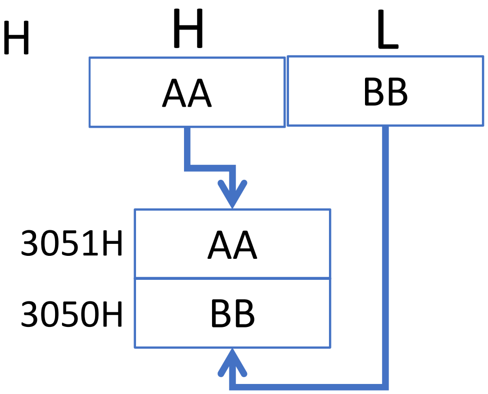


  - Instruction: **SHLD** 16-bit address; [address] <- L, [address+1] <- H

    **(3 bytes instruction)**

    e.g. LXI H, AABBH

    SHLD 2050H; [2050H] <- L, [2051H] <- H



# Data Transfer Instruction (doesn’t affect flags)

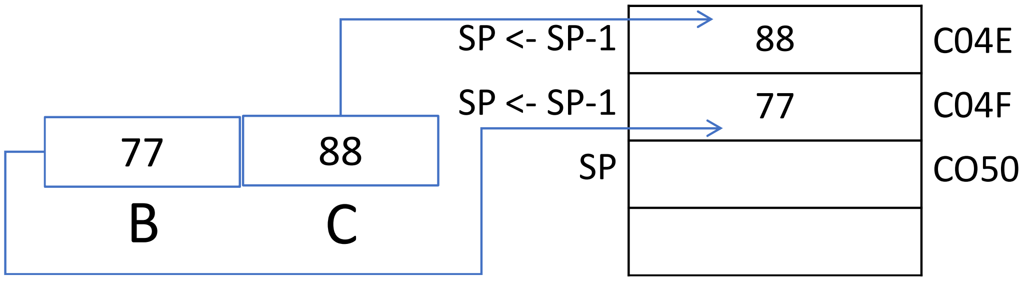
- **Between register and stack memory**
  - Stack memory is a memory that follows LIFO (Last In First Out) principle
  - Stack pointer (SP) holds the address of top of stack
  - Instruction: **PUSH** reg_pair;

    **(1 byte instruction)**

  **Following steps are processed for execution of PUSH instruction**

  - ➢ SP is decrement by 1;   SP <- SP - 1
  - ➢ Higher order register value is copied to memory pointed by SP;  [SP] <- higher register value
  - ➢ SP is again decremented by 1;  SP <- SP - 1
  - ➢ Lower order register value is copied to memory pointed by SP;  [SP] <- lower register value

E.g. LXI B, 7788H;
LXI SP, C050H
PUSH B;



# Data Transfer Instruction (doesn’t affect flags)

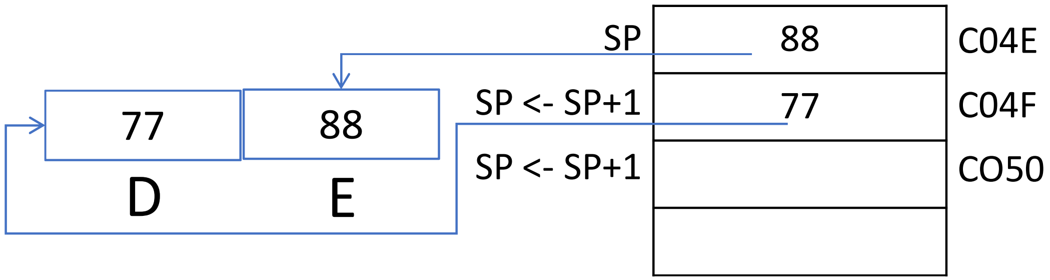
- **Between register and stack memory**
  - Instruction: **POP** reg_pair;

    **(1 byte instruction)**

  **Following steps are processed for execution of POP instruction**

  - ➢ Value of memory pointed by SP is copied to lower order register; lower order register <- [SP]
  - ➢ SP is incremented by 1; SP <- SP + 1
  - ➢ Value of memory pointed by SP is copied to higher order register; higher register value <- [SP]
  - ➢ SP is again incremented by 1; SP <- SP + 1

E.g. POP D;



# Data Transfer Instruction (doesn’t affect flags)

- **Between input port and accumulator**
  - Instruction: **IN** 8-bit port address;   A<- [port address]

    The content of input port is copied to accumulator

    (**2 bytes instruction)**

    E.g.  IN 8CH;  A <- [8CH]

- **Between output port and accumulator**
  - Instruction: **OUT** 8-bit port address;   [port address] <- A

    The content of accumulator is copied to output port

    (**2 bytes instruction)**

    E.g.  OUT 80H;  [80H] <- A

# Data Transfer Instruction (doesn't affect flags)

- **Between stack memory and HL**
  - Instruction: **XTHL**;    [SP] <-> L, [SP+1] <-> H

    Exchanges the content of stack memory pointed by SP with L and SP+1 with H

    (**1 bytes instruction)**

    E.g.  XTHL;

- **Between stack pointer (SP) and HL**
  - Instruction: **SPHL**;   SP <- HL

    Copies the value of register pair HL to SP

    (**1 bytes instruction)**

    E.g.  LXI H, 4455H;  H <- 44H, L <- 55H
    SPHL;  SP <- HL