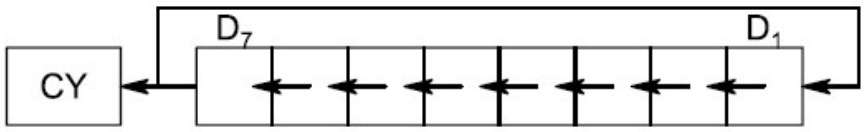
Logical Instruction(affect flags)

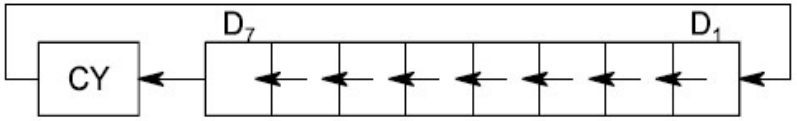
- **Rotate instruction (only carry flags are affected by rotate instruction)**
  - These instructions rotate the content of accumulator by 1-bit towards left or right with/without carry.
  - Instruction: **RLC**; Rotate left without carry



e.g MVI A, 39H; A <- 39H
RLC

39H: 0011 1001
RLC
72H: 0111 0010

  - Instruction: **RAL**; Rotate left with carry

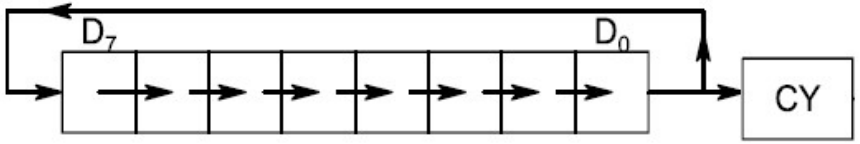


e.g MVI A, 44H; A <- 44H
RAL

(Note: Rotating left can be viewed as multiplying by 2 but it is valid only if MSB is 0)

Logical Instruction(affect flags)

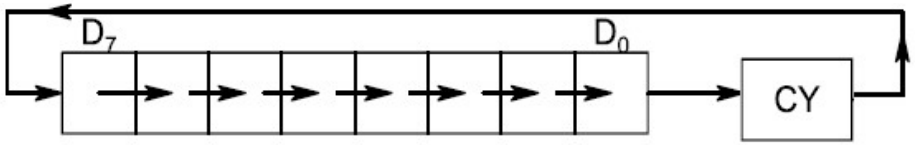
- **Rotate instruction (only carry flags are affected by rotate instruction)**
  - Instruction: **RRC**; Rotate right without carry



e.g MVI A, ACH; A <- ACH
RRC

ACH: 1010 1100
↓ **RRC**
56H: 0101 0110

  - Instruction: **RAR**; Rotate right with carry



e.g MVI A, CCH; A <- CCH
RAR

02H: 0000 0010
04H: 0000 0100
08H: 0000 1000
10H: 0001 0000

(Note: Rotating right can be viewed as dividing by 2 but it is valid only if LSB is 0)

Branching Instructions

- These instructions allow microprocessor to change the sequence of a program to any other location.
- Changes the sequence either conditionally or unconditionally
- Classified into following sub-groups
  - Jump Instructions
  - Call and Return Instructions
  - Restart Instructions

Branching Instructions

- **Jump Instructions**
  - **Unconditional Jump**
  - Instruction: **JMP** 16-bit address
    - Program sequence is transferred to the memory location specified by 16-bit address.

E.g

```
8000H :  MVI A, 20H
8002H:   MVI B, 30H
8004H:   JMP 8050H
8007H :  ……
         ……..
8050H :  ADD B
8051H:   HLT
```

Branching Instructions

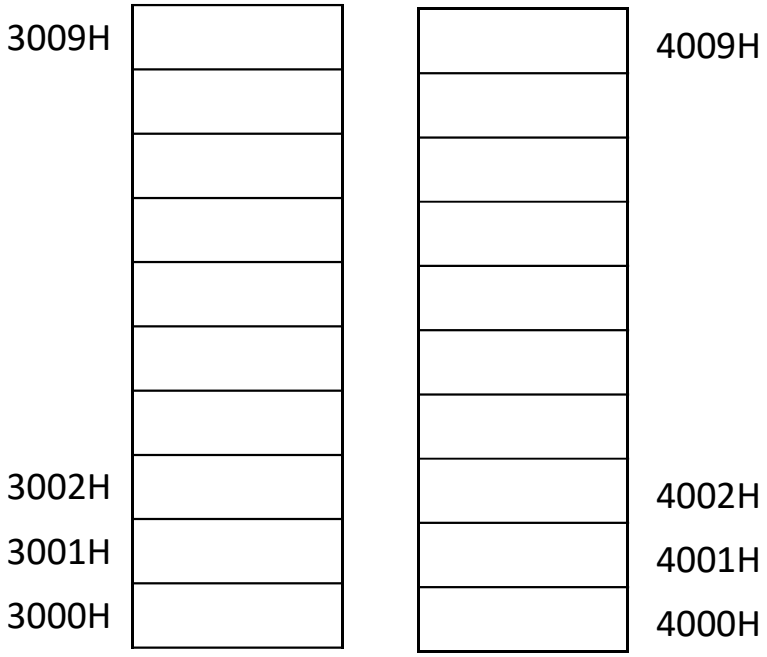
- **Jump Instructions**
  - **Conditional Jump**
  - In conditional jump, the microprocessor makes decision to jump to the next location on the basis of flags
  - These instructions are like if..else statement of C/C++.

Instructions	Comments
JC 16-bit address	Jump on carry(CY=1)
JNC 16-bit address	Jump on not carry(CY=0)
JZ 16-bit address	Jump on zero(Z=1)
JNZ 16-bit address	Jump on not zero (Z=0)
JPE 16-bit address	Jump on parity even (P=1)
JPO 16-bit address	Jump on parity odd (P=0)
JP 16-bit address	Jump on plus (S=0)
JM 16-bit address	Jump on Minus (S=1)

Branching Instructions

- **Jump Instructions**
  - Write a program in 8085 to transfer 10 8-bit numbers stored in memory starting from 3000H to memory starting from 4000H.

```
        LXI H, 3000H; SOURCE TABLE
        LXI B, 4000H;  B<-40, C<-00
        MVI D, 0AH; COUNTER
UP:     MOV B,M;  A<-[HL]
        STAX B;  [BC]<-A
        INX H;  HL<-HL+1
        INX B;  BC<-BC+1
        DCR D; D<-D-1
        JNZ UP
        HLT
```



Branching Instructions

- **Jump Instructions**
  - Write a program in 8085 to read two numbers from memory 2050H and 2060H. If number in 2050H is smaller, add two numbers else subtract the number in 2050H from number in 2060H.

```
        LDA 2060H;  A<-[2060H]
        MOV B,A
        LDA 2050H;  A<-[2050H]
        CMP B;  A-B
        JC PASS
        MOV C,A
        MOV A,B
        MOV B,C
        SUB B
        JMP LAST
PASS:   ADD B
LAST:   HLT
```

Condition	Zero (Z) Flag	Carry (CY) Flag
A > D	0	0
A = D	1	0
A < D	0	1

Branching Instructions

- **Jump Instructions**
  - 10 8-bit numbers are stored in memory starting from D000H. Write a program in 8085 to transfer these numbers to memory starting from D050H only if number of source table is less than 9FH, else store 00H in destination.

```
        LXI H, D000H; SOURCE TABLE
        LXI D, D050H; DESTINATION TABLE
        MVI B, 0AH; COUNTER
UP:     MOV A, M;  A<-[HL]
        CPI 9FH;  A-9FH
        JC PASS
        MVI A, 00H
PASS:   STAX D;   [DE] <- A
        INX H
        INX D
        DCR B
        JNZ UP
        HLT
```