3.3.3 Logical Group Instructions :-

→ These instructions perform various logical operations ex: (AND, OR, Exclusive OR, Rotate, Compare, Complement) with the contents of the accumulator.

Instructions	No of Bytes	Machine Cycle (MC)	T-state	Addressing Mode (A.M.)
1. ANA R	1 Byte	1 M.C. (OF)	4T	Register AM
2. ANA M	1 Byte	2 MC (OF+MR)	4T+3T = 7T	Indirect AM
3. ANI data	2 Byte	2 MC (OF+MR)	4T+3T = 7T	Immediate AM
4. ORA R	1 Byte	1 MC (OF)	4T	Register AM
5. ORA M	1 Byte	2 MC (OF+MR)	4T+3T = 7T	Indirect AM
6. ORI data	2 Byte	2 MC (OF+MR)	4T+3T = 7T	Immediate AM
7. XRA R	1 Byte	1 MC (OF)	4T	Register AM
8. XRA M	1 Byte	2 MC (OF+MR)	4T+3T = 7T	Indirect AM
9. XRI data	2 Byte	2 MC (OF+MR)	4T+3T = 7T	Immediate AM
10. CMA	1 Byte	1 MC (OF)	4T	Implicit A.M.
11. CMC	1 Byte	1 MC (OF)	4T	Implicit AM
12. STC	1 Byte	1 MC (OF)	4T	Implicit AM
13. CMP R	1 Byte	1 MC (OF)	4T	Register AM
14. CMP M	1 Byte	2 MC (OF+MR)	4T+3T = 7T	Indirect AM
15. CPI data	2 Byte	2 MC (OF+MR)	4T+3T = 7T	Immediate AM
16. RLC	1 Byte	1 MC (OF)	4T	Implicit AM.
17. RRC	1 Byte	1 MC (OF)	4T	Implicit AM.
18. RAL	1 Byte	1 MC (OF)	4T	Implicit AM
19. RAR	1 Byte	1 MC (OF)	4T	Implicit AM

Application of OR, AND

OR:- For set the bit

ex:

IN 00H A = D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
ORI 10H = 0 0 0 1 0 0 0 0

D_7 D_6 D_5 1 D_3 D_2 D_1 D_0

AND:- For reset the bit
(Mask the bit)

ex:

IN 00H A = D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
ANI 7FH = 0 1 1 1 1 1 1 1
---
0 D_6 D_5 D_4 D_3 D_2 D_1 D_0

ANA R
ANA M
ANI 8-bit data
} Logically AND the contents of Reg./ Memory/ specified data with accumulator
A ⇐ A AND R/M/data

ORA R
ORA M
ORI 8-bit data
} Logically OR the contents of Reg/Mem/ data with accumulator
A ⇐ A OR R/M/data

XRA R
XRA M
XRI 8-bit data
} Logically XOR the contents of Reg/Mem/data with accumulator
A ⇐ A XOR R/M/data

CMA :- Complement the contents of accumulator
$A \Leftarrow \overline{A}$

CMC :- Complement the CY flag ($CY = \overline{CY}$)

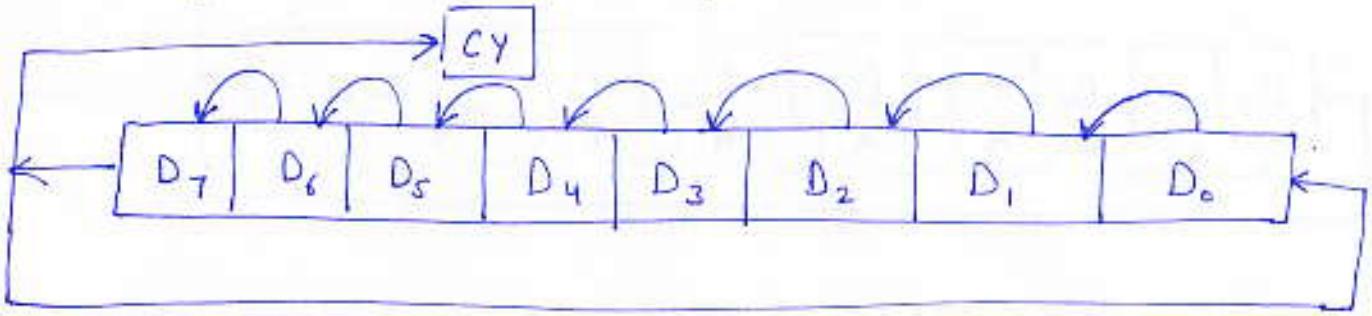
STC :- Set the CY flag (CY = 1)

Logic Operations :-

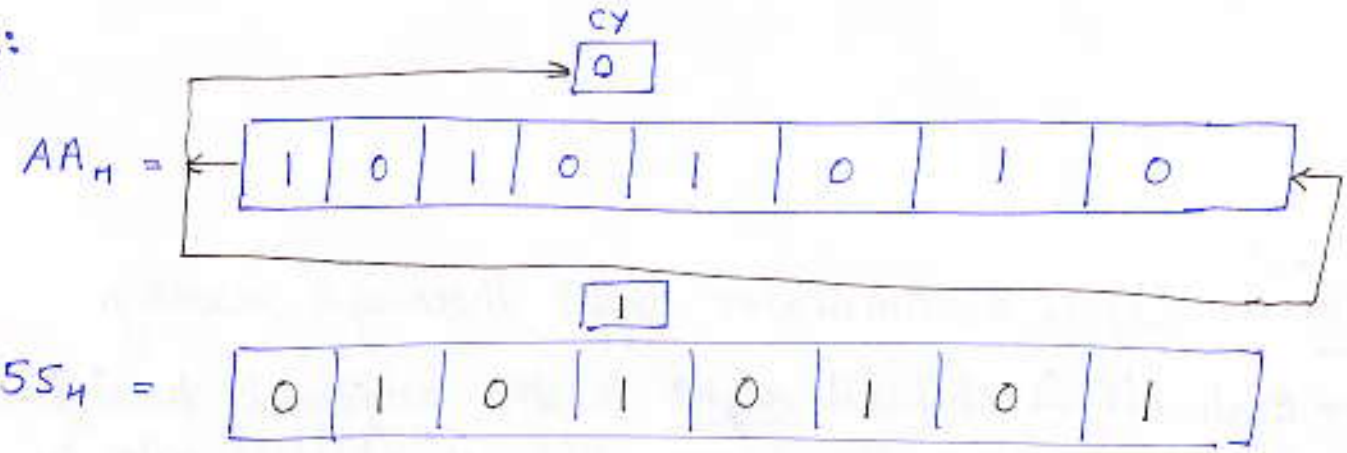
1.) RLC :- Rotate accumulator left

→ each bit is shifted to the adjacent left position. D_7 becomes D_0

→ CY flag is modified according to bit D_7.



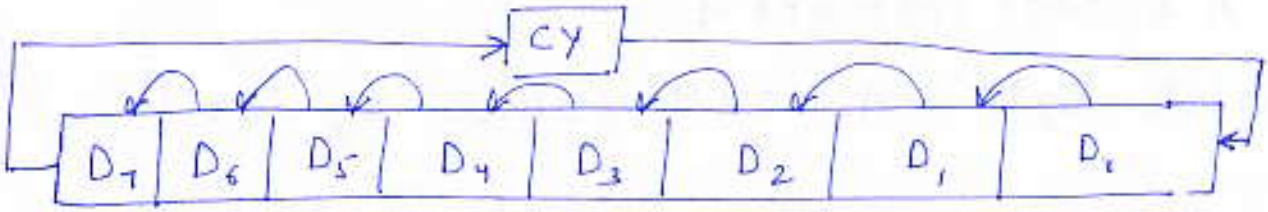
ex:



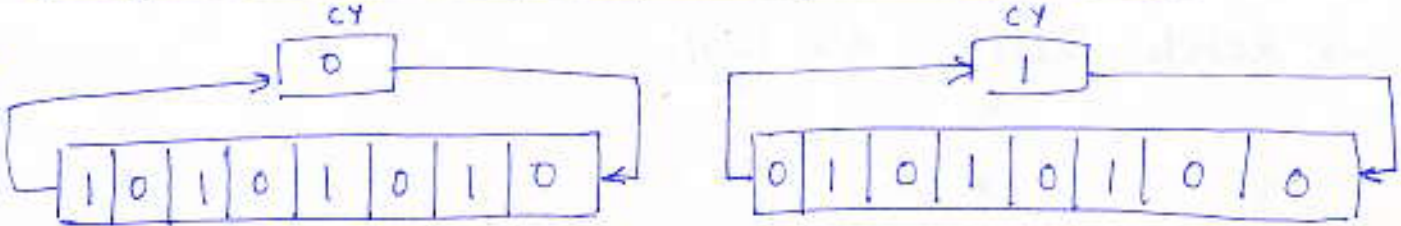
2.) RAL :- Rotate accumulator left through carry

→ each bit is shifted to the adjacent left position. Bit D_7 becomes the carry bit and the carry bit is shifted into D_0.

→ Carry flag is modified to bit D_7.



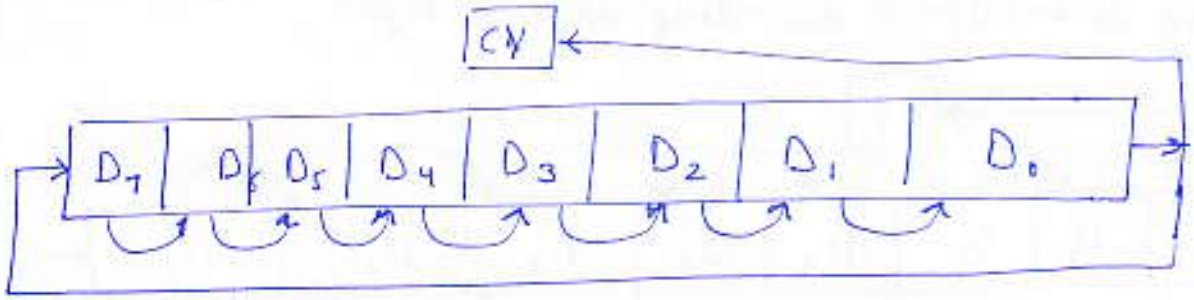
ex:



RRC :- Rotate Accumulator Right

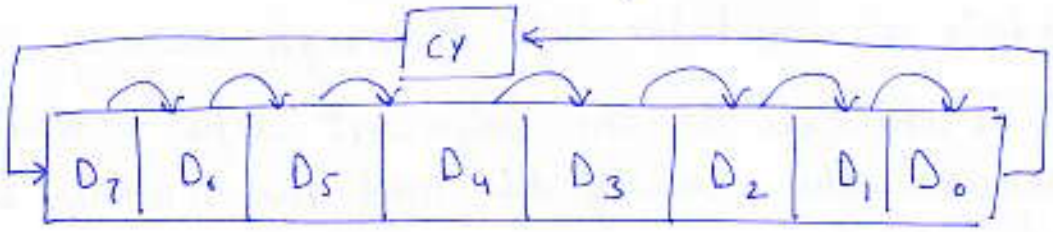
→ each bit is shifted right to adjacent position. Bit D_0 becomes D_7.

→ carry flag is modified according to D_0.



RAR :- Rotate accumulator Right through carry

→ Each bit is shifted right to the adjacent position. Bit D_0 becomes the carry bit is shifted into D_7.



Application :- arithmetic multiply & divide operations.

A = 0000 1000 = 08 H

→ Rotate Right A = 04 H (divide by two)

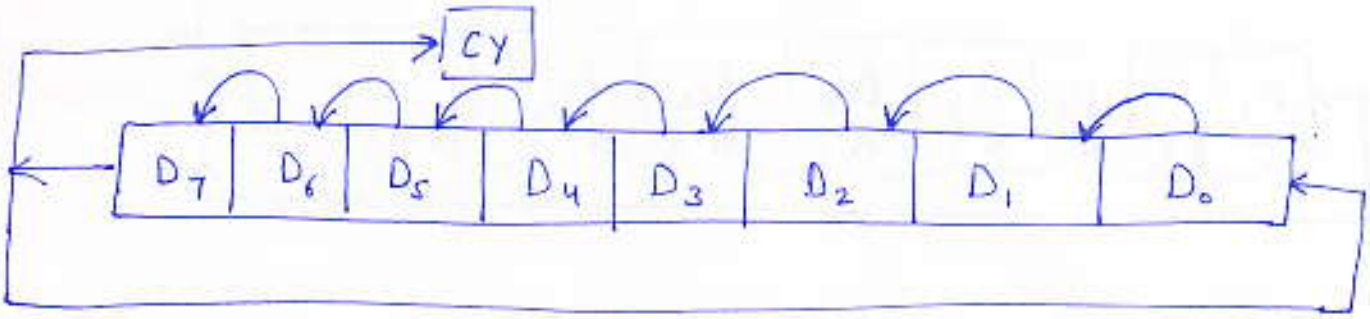
→ Rotate Left A = 10 H

Logic Operations :-

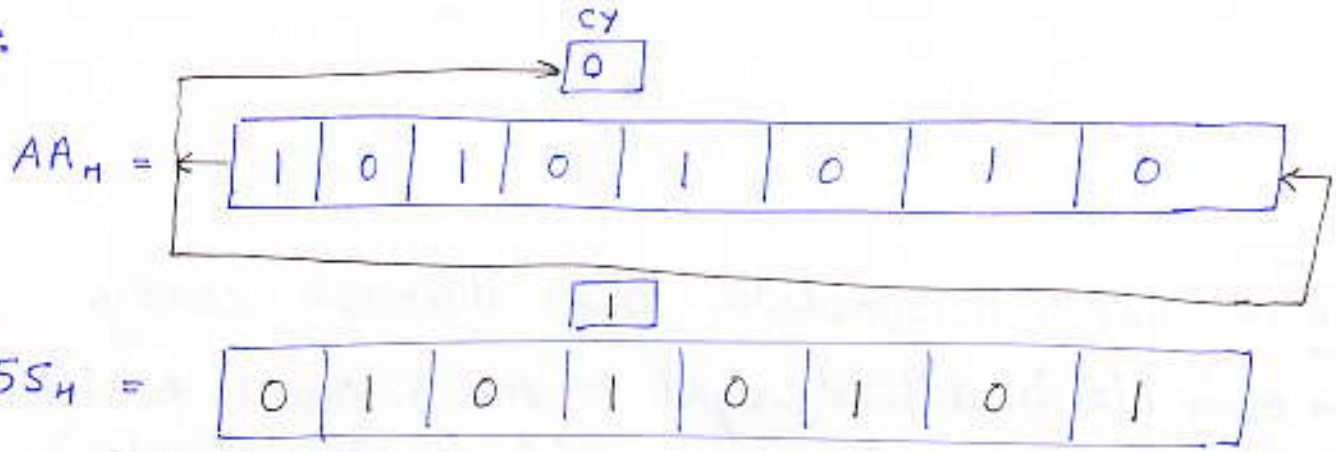
1.) RLC :- Rotate accumulator left

→ each bit is shifted to the adjacent left position. D_7 becomes D_0

→ CY flag is modified according to bit D_7.



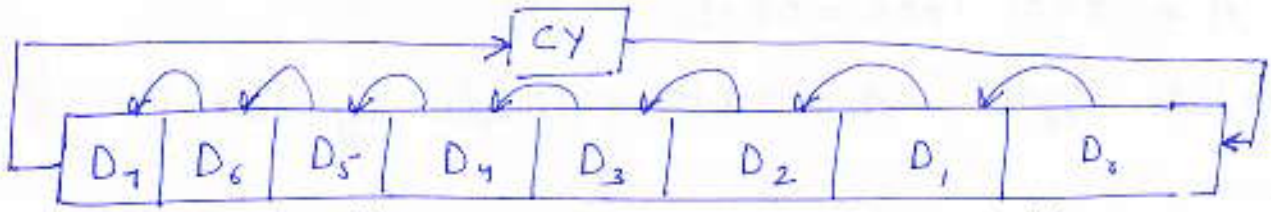
Ex:



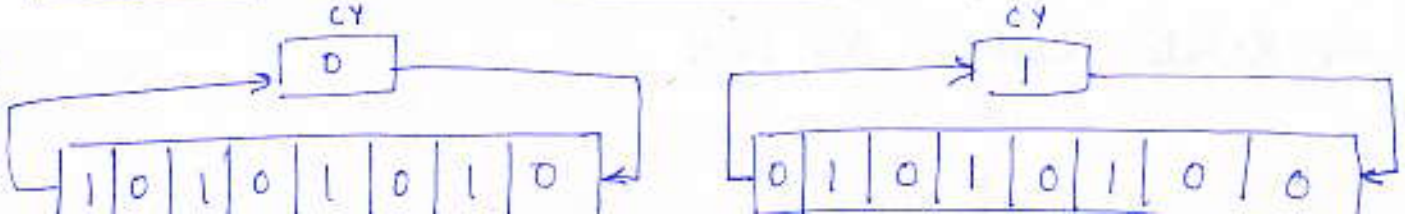
2.) RAL :- Rotate accumulator left through carry

→ each bit is shifted to the adjacent left position. Bit D_7 becomes the carry bit and the carry bit is shifted into D_0.

→ Carry flag is modified to bit D_7.



Ex:



CMP R/M :- Compare (Register or Memory) with Accumulator

→ 1-byte instruction

→ A < (R/M), CY is set & zero flag is reset
A = (R/M), zero flag is set & CY flag is reset
A > (R/M), CY & zero flag are reset

CPI 8-bit :- Compare immediate with accumulator

→ 2-byte instruction

– same. –

ex:- Write an instruction to load the accumulator 64H, & verify data byte in memory location 2050H is equal to accumulator contents.

```
LXI H, 2050H
MVI A, 64H
CMP M
```

ex:- Find 2's complement of a no. stored at memory location 2050H & store the result at 2150H.

```
LDA 2050H
CMA
ADI 01H
STA 2150H

HLT
```