Lakshmi

2's comp (26) = 1101 1010

87 = 0011 0111

1 0001 0001

Data + Borrow = 26 H.

Complement - C, C = 0.

Flags S=0 Z=0, AC=1, P=1, C=0.

11. SBB R (Subtract register from A with borrow)

The content of register and carry status are subtracted from accumulator. After subtraction, result is stored in accumulator.

M/C = 1
States: 4
Flags: All
Length: 1 byte.

Eg SBB B.

12. SBB M (Subtract memory from A with borrow)

The content of memory location addressed by HL pair & carry status are subtracted from content of A. Result is stored in A

M/C = 2 (F, MR.)
States: 7
Flags: All
Length: 1 byte.

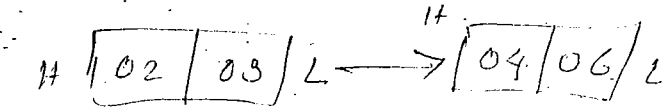
Eg: SBB M.

13. DAD Rp (Added register pair to HL pair)

The contents of register pair Rp are added to the contents of HL pair. & result is placed in HL pair. Only carry flag is affected.

M/C = 3 (F, Bus idle, Bus idle)
States: 10
Flags: carry only
length: 1 byte.

Eg: DAD H



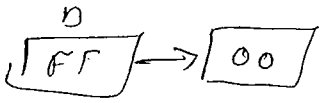
Bus idle machine cycle is executed when external signal needed for any external operation.

14. INR R (Increment Register Content)

The content of Register R is incremented by one. All flags except Carry is modified.

M/C = 1
States 4.
length: 1 byte
Flags All except carry

Eg. INR D



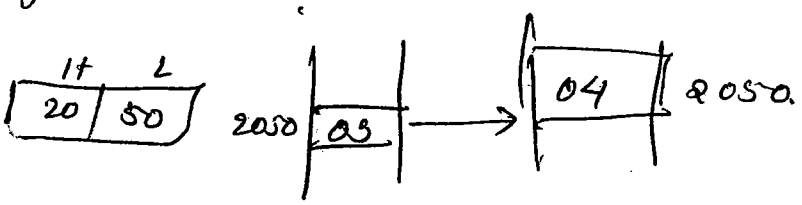
carry not affected.

15. INR M (Increment memory content)

The content of memory location addressed by HL pair is incremented by 1

M/C = 3 (F, MR, MW)
States: 10
Flags: All except carry
length: 1 byte

Eg: INR M

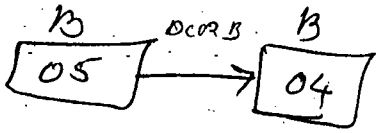


16. DCR R. (Decrement Register content)

The content of register R is decremented by 1

M/C: 1
States: 4
Flags: All except carry
length: 1 byte.

eg. DCR B



17. DCR m (Decrement memory content)

The content of memory location addressed by HL pair is decremented by 1

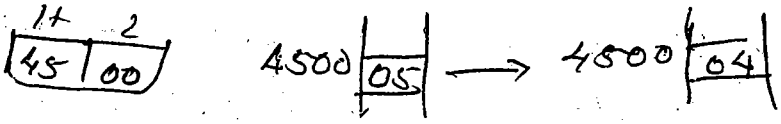
m/c : 3 (F, MR, MW)

States : 10

Length : 1 byte.

Flags : All except Carry

eg. DCR M.



18. INX Rp. (Increment Register pair)

The content of register pair Rp is incremented by 1 no flag is affected

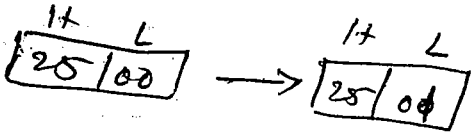
m/c : 1

States : 6

Flags : none

length : 1 byte.

eg : INX H



19. DCX Rp. (Decrement Register pair)

The content of register pair Rp is decremented by 1

m/c : 1

States : 6

Flags : none

length : 1 byte.

eg : DCX D

20. DAA (Decimal Adjust Accumulator)

Instruction DAA is used in the program after ADD, ADC etc. After the execution of these instruction Result is in hexadecimal and placed in accumulator. DAA operates on Result and final Result in decimal form.

Instruction DAA works as follows.

i) IF the value of the low-order 4 bits (D_3-D_0) in accumulator is greater than 9, OR if AC flag is set, the instruction adds 6 to the low-order 4 bits.

ii) If the value of the high order 4 bits (D_4-D_7) in the accumulator is greater than 9, or if carry flag set then the instruction adds 60 to high order 4 bits.

39 + 12
BCD BCD

```
0011 1001
0001 0010
---------
0100 1011
0000 0110
---------
0101 0001   BCD decimal
```

m/c=1
States : 4
Flags : All
Length : 1 byte.

III Logic And Bit manipulation Instructions

1. ANA R. (And register with A)

The contents of register R is ANDed with content of A, and result is placed in A.

[A] ← [A] ∧ [R]

m/c cycle : 1

States : 4

Flags : All AC set to 1 & Carry to 0.

Length : 1 byte

A = 54 0101 0100

B = 82 1000 0010

0000 0000

flags : S=0 Z=1 P=1, AC=1 C=0

[A] = 00H.

Eg : ANA B

2. ANA M. (AND memory with A)

The contents of memory location addressed by HL pair is ANDed with accumulator and the result is placed in accumulator.

[A] ← [A] ∧ [[HL]]

M/c = 2 (F, MR)

States : 7

Flags : All except AC=1, C=0

length : 1 byte

3. ANI 8 bit data (AND immediate data with accumulator)

The 8 bit data is ANDed with contents of accumulator and result is placed in A.

[A] ← [A] ∧ data

m/c = 2 (F, MR)

States : 7

Flags : All AC=1 C=0

Length : 2 byte.

Eg. ANI 97H.

4. ORA R (OR Register with accumulator)

The content of register R is ORed with the content of A. Result is placed in A.

$[A] \leftarrow [A] \vee [R]$

m/c = 1
States: 4
Flags: all C & AC = 0
length: 1 byte

A = 03 0000 0011
C = 81 1000 0001
1000 0011

S = 1 Z = 0 P = 0 C = 0 AC = 0
[A] = 83 H

Eg. ORA C.

5. ORA M (OR memory with A)

The content of memory location addressed by HL pair is ORed with content of A

$[A] \leftarrow [A] \vee [[HL]]$

m/c : 2
States : 7
Flags : all C & AC = 0
length : 1 byte

6. ORI 8 bit data (OR immediate data with A)

The content as 8 bit data is ORed with the content of accumulator

$[A] \leftarrow [A] \vee$ data

m/c : 2
States : 7.
Flags - all, C & AC = 0
length - 2 bytes

Eg ORI 07 H

7. XRA R (Exclusive OR Register with accumulator)

The content of Register R is exclusive ORed with content of A. Result is placed in accumulator.

[A] ← [A] ∀ [R]

m/c : 1
Status : 4
Flags : All e.g AC = 0
Length : 1 byte.

A = 77	0111	0111
D = 56	0101	0110
	0010	0001	21H

eg. XRA D

8. XRA M (Exclusive OR memory with A)

The content of memory location addressed by HL pair is exclusive ORed with accumulator. The result is placed in accumulator. [A] ← [A] ∀ [[HL]]

m/c : 2
Status : 7
Flags : All AC & C = 0
Length : 1 byte.

9. XRI 8 bit data (Exclusive OR Immediate data with accumulator)

The immediate data is Exored with content of accumulator. The result is placed in accumulator.

[A] ← [A] ∀ data.

m/c : 2
Status : 7
Flags : All C & AC = 0
Length : 2 bytes.

eg. XRI A2H.

10. CMP R (Compare Register with accumulator)

The content of register R is subtracted from content of accumulator and status flags are set accordingly to the result. But result is discarded.

i) IF [A] < [R] , carry flag is set.
zero flag is reset.

ii) IF [A] > [R] , carry & zero flags are reset.

iii) IF [A] = [R] zero flag is set, carry flag is reset.

m/c : 1
Status : 4
Flags : All
Length : 1 byte.

Eg: CMP B

A	57	
B	62	

So 57-62.

0110 0010
1001 1101
1
1001 1111 0

2's com (62) = 1001 1110
-57. 0101 0111
0] 1111 0101

Complement carry = C = 1.

So S=1 Z=0 AC=1 P=1, C=1

11. CMP M. (Compare memory with A)

The content of memory location addressed by HL pair is subtracted from content of accumulator. Flags are set according to the result and result is discarded.

[A] - [[HL]]

m/c : 2 (F, MR)
Status : 7
Flags : All.

Length: 1 byte

Eg: CMP M.

i) [A] < [M] C=1, Z=0

ii) [A] < [M] C=0, Z=0

iii) [A] = [M] Z=1, C=0

12. CPI 8bit data (Compare immediate data with A)

The 8bit data is subtracted from A and flags are modified according to the result. content of A remains unchanged. The result is discarded.

[A] - data

M/C : 2

States : 7

Flags : All

length : 2 bytes

Eg CPI 98H

[A] = 89

2's comple (98) 1010 1000

0100 1001

0] 1111 0001

Complement carry C=1

S=1 Z=0, AC=1, P=0 C=1

0101 1000
1010 0111
1
0101 0000
A8

13. CMA (complement Accumulator)

The content of accumulator is complemented

[A] ← $\bar{A}$. no flags are affected

M/C : 1

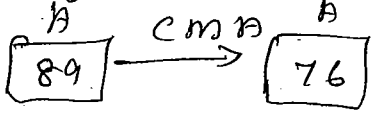
States : 4.

Flags: None

length : 1 byte.

1000 1001 (89)

Complement - 0111 0110 (76)



14. CMC (complement carry)

carry flag will be complemented after the execution of CMC instruction.

mle : 1
states : 4
Flags : only carry
length : 1 byte

15. STC (set carry)

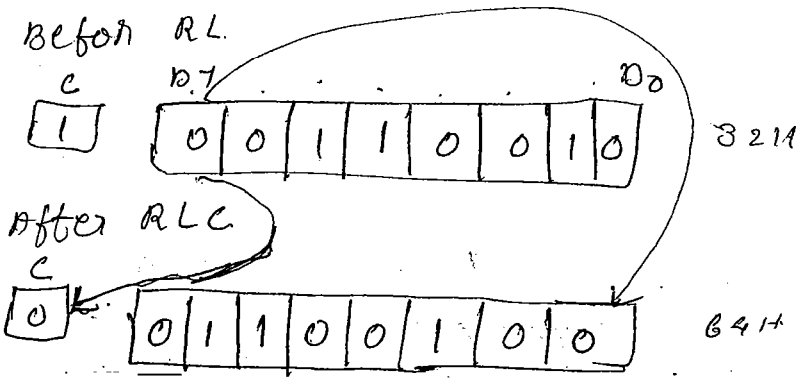
The carry flag is set to 1

mle : 1
states : 4
Flags : carry only
length : 1 byte.

16 RLC (Rotate accumulator left without carry)

The content of accumulator is rotated left by 1 bit and left most bit of accumulator is rotateed to carry. only carry flag is affected

mle : 1
states : 4
Flags : carry only
length : 1 byte



The left most bit of A is placed in carry as well as in right-most bit position D_0.

17. RRC (Rotate accumulator without carry)

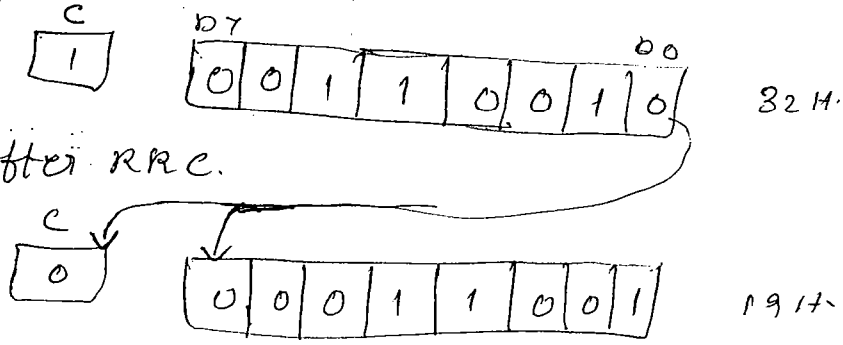
The content of accumulator is rotated right by 1 bit and right most bit of accumulator is rotated to carry.

No. c : 1

States : 4

Flags : carry only

Eg. Before RRC



After RRC.

i.e Right most bit of accumulator is placed on carry as well as left most bit.

18. RAR (Rotate accumulator right through carry)

The content of accumulator along with carry flag is rotated right by 1 bit. Here carry is moved to MSB (D7) position and LSB (D_0) moved to carry

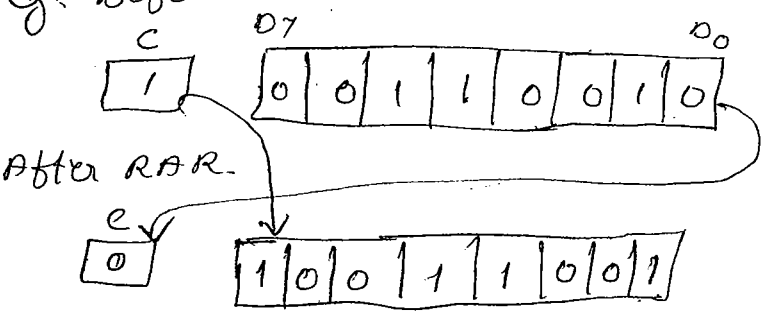
No. c : 1

States : 4

Flags : carry only

Length : 1 byte.

Eg. Before RAR



19. RAL (Rotate Accumulator left through carry)

The content of accumulator along with carry is rotated left by 1 bit. Here carry is moved to LSB and MSB is moved to carry.

m/c : 1

Status : 4

Flags : carry only

Length : 1 byte.

Befor RAL.