8085 Instructions

1) MOV -> MOV Rd RS => 1 Byte, 1 Machine cycle, 4-T state [MOV R]
MOV M => 1 Byte 2 machine cycle, 7-T state.

MOV is used to move 8 bit data content from two register (source and destination) either it in Register specified.

In MOV M, M is the memory address where content is pointed by H,L register pair

---

2) MVI R, 8 bit data -> 2 Byte, 2 machine cyle, 7 T state [Memory mode M]
It is used to move 8 bit data into Register. [move immediate]

MVI Register, 8bit data => move 8 bit data to register immediately

MVI M, data -> to load 8 bit data into a memory location pointed by HL pair immediately. -> 2 Byte, 3 machine cyle, 10 T states.

It is immediate addressing mode

---

3) ADD => ADD R => used to add content of Register specified to Accumulator content. 1 Byte, 1 machine cyle, 4 T-states.

ADD M -> add content specified in memory location pointed by HL pair to content of Accumulator.

1 Byte, 2 machine cyle, 7 T-states. all flags modified.

---

4) ADI => ADI 8 bit data => add an 8 bit value to content of Accumulator [Add immediate] 2 byte, 2 machine cyle, 7 T-state.
all flags modified. Immediate addressing mode.

---

5) ACI => add with carry immediate to Accumulator. ACI 8 bit data.
add 8 bit data to content of Accumulator along with carry.
It is immediate addressing mode. all flags are modified.

2 byte, 2 machine cyle, 7 T-State.

6) ADC => add with carry

ADC R -> add the content of Register specified to content of Accumulator along with carry.

1 Byte, 1 machine Cycle, 4 T state.

ADC M => add the content present at memory location pointed by HL pair to content of Register along with carry.

1 Byte, 2 machine cycle, 7 T-State.

All flags are modified.

---

7) ANA => "AND Accumulator".

ANA R => logically perform "And" operation between content of Register specified and content of Accumulator.

1 Byte, 1 machine cyc, 4 - T state.

ANA M => logically perform "AND" operation with content located at memory address pointed by HL pair and content of Accumulator.

1 Byte, 2 machine cycle, 7 T-State.

Auxiliary flag is Set and carry flag is Reset.

---

8) ANI = And immediate with Accumulator.

ANI 8 bit data => to logically perform AND operation between 8 bit data specified and content of Accumulator.

It is immediate addressing mode.

2 byte, 2 machine cycle, 7 T-State.

carry flag is set to 0

Auxillary flag is set to 1.

9) CMA ⇒ Complement Accumulator.

It performs 1's complement on content of Accumulator, result is stored in Accumulator itself.

No flags modified.

1 byte, 1 machine cycle, 4 T-states.

---

10) CMP ⇒ Compare Accumulator

CMP R ⇒ Used to compare content of Register specified and content of Accumulator. It is basically [A-R] operation.

Flags have value depending on A-R.

If [A-R] result is positive [+ve] ⇒ Non signflag = 0
zeroflag = 0
carryflag = 0

If [A-R] result is negative [-ve] ⇒
signflag ⇒ 1
zeroflag ⇒ 0
carryflag ⇒ 0

If [A-R] result is equal i.e, [A=-R] ⇒ signflag ⇒ 0
Zeroflag ⇒ 1
carryflag ⇒ 0

1 byte, 1 machine cycle 4 T-states

CMP M ⇒ to compare content of Accumulator with content of operand at memory location pointed by HL pair.

1 byte, 2 machine cycle, 7 T-states.

11) DAA => Decimal Adjust Accumulator.

The 8bit number in accumulator is adjusted to form two 4 bit BCD.

Rules. If value of least significant 4 bits of Accumulator is greater than 9 or if AC flag is set, 6 is added to Accumulator.

If value of most significant 4 bits of Accumulator is now greater than 9, or if CY flag is set, 6 is added to the most significant 4bits of the accumulator.

All flags are modified. 1 Byte, 1 machine cycle, 4 T-States.

Implicit addressing mode.

---

12) DCR => Decrement.

DCR R => Used to decrement the content of Register by one.

It is basically subtract 1 from the register specified contents.

decrement value is stored on Register R specified.

all flag except carry flag is modified.

1 byte, 1 machine cycle, 4 T-state.

DCR M -> decrement the content present at memory location pointed by HL register pair by one.

1 byte, 3 machine cycle, 10 T-states.

---

13) DCX => DCX Rp => decrement Register pair, Decrement Extended register pair.

This instruction is used to decrement content of Rp specified by 1.

and result is stored in Register pair itself

no bits are affected.

↳ flag

1 byte, 1 machine cycle, 6 T-states.

14) HLT - halt the microprocessor. This is the last statement of a 8085 program. It tells the END of Execution of instruction in any program.

It executes current instruction and holds any further execution. Reset is necessary to exit from halt state.

1 byte, 2 or more machine cycle, 5 or more T states.

---

15) INR => Increment

INR-R => This instruction increment the value of Register specified by one. It adds '1' to content of Register, and result is stored in Register itself.

All flags except carry flag is modified.

1 byte, 1 machine cycle, 4 T-state.

INR M => It increment the content present at memory location pointed by H-L register pair by one.

1 Byte, 3 machine cycle, 10 T state.

16) INX ⇒ INX RP → Increment Xtended Register pair.

This instruction is used to increment the content of Register pair specified by I and result is stored in Register pair itself.

no flag bits are affected.

1 byte, 1 machine cycle, 6 T-states.

---

17) JMP ⇒ unconditional jump / Unconditional Jump

JMP 16 bit address. The program sequence is transferred to memory address given in the operand.

3 byte, 3 machine cycle, 10 T states.

There is no condition to jump.

---

18) JC ⇒ Jump on carry set. It is a conditional jump.

JC 16 bit address / label. The program sequence is transferred to particular label or 16 bit address, if carry flag is set (1).

3 byte, 3 machine cycle, 10 T states.

---

19) JNC ⇒ Jump on carry reset. It is a conditional jump.

JNC 16 bit address / label. The program sequence is transferred to particular label or 16 bit address if carry flag is reset (0).

[no carry].

3 byte, 3 machine cycle, 10 T states.

---

20) JZ ⇒ Jump on zero. It is a conditional jump.

JZ 16 bit address / label. The program sequence is transferred to particular label / 16 bit address if zero flag is set (1).

3 byte, 3 machine cycle, 10 T-state.

21) JNZ ⇒ Jump on Non Zero. This is a conditional jump.
The program sequence is transferred to particular label of 16 bit address
if zero flag is reset [0]. no zero flag set.

3 byte, 3 machine cycle, 10 T-state

---

22) LDA ⇒ load Accumulator.
LDA 16 bit address. This instruction is used to load the Accumulator with content from 16 bit memory specified.
Accumulator will have 8 bit content from 16 bit address.
Absolute addressing mode.
no flags are modified.

3 byte, 4 machine cycle, 13 T-states.

---

23) LDAX RP ⇒ load accumulator from memory pointed by extended Register pair. It is an indirect addressing mode.
only BC and DE registers pairs are used for LDAX RP.
LDAX M is not provided in 8085, because, it is same as
Mov A, M in its function.
1 byte, 2 machine cycle, 7 T-state.

---

24) LXI ⇒ load register pair immediate
LXI RP, 16 bit data. This instruction loads 16 bit data into the specified register pair.
BC, DE, HL are the register pair.

Immediate addressing mode.
no flags are modified.

3 byte, 3 machine cycle, 10 T states

25) ORA => OR Accumulator.

ORA R -> This instruction performs logical OR operation with specified Register content and content of Register.
The result is stored in Accumulator

1 byte, 1 machine cycle, 4 T-states

ORA M => performs logical OR operation with contents present at memory location pointed by HL register pair.

1 byte, 2 machine cycle, 7 T-states

---

26) ORI => OR immediate with Accumulator.

ORI 8 bit data -> it performs OR operation on 8 bit data specified and with content of Accumulator. The result is stored in Accumulator.

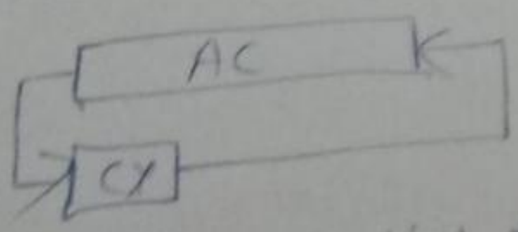
carry flag and AC are reset to 0.

2 bytes, 2 machine cycle, 7 T-states

---

27) RAL => Rotate Accumulator left through carry.

It rotates the Accumulator content to left by 1 bit position.



bit from MSB is copied to CY flag bit, previous CY bit will be moved to least significant bit of Accumulator.

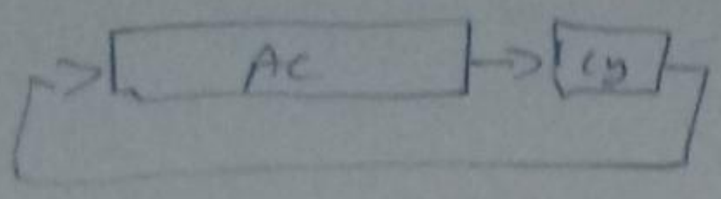
Thus it is 9 bit rotation of Accumulator & CY flag.

only carry flag is affected.

1 Byte, 1 machine cycle, 4 T-States

28) RAR -> Rotate accumulator Right through carry.

It rotate the Accumulator content to Right by 1 bit position.



The least significant bit is coming out and will be copied on CY flag bit and previous cy flag bit will be moved to the most significant bit position of accumulator.

Thus it is a bit Rotation of Accumulator.

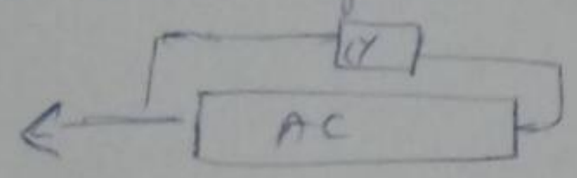
Only carry flag is affected.

1 Byte, 1 machine cycle, 4 T-state.

---

29) RLC => Rotate left Accumulator without carry.

It rotate the content of Accumulator to left bit by 1 position.



MSB of Accumulator will come out and left rotate will create an empty space at LSB and this comes out bit will be copied at empty bit place and also on the carry bit in flag register. Thus carry flag gets a copy of bit moved out from MS bit position.

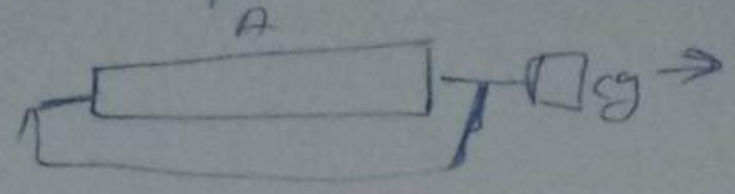
It is only 8 bit rotation of Accumulator content.

only carry flag is affected.

1 Byte, 1 machine cycle, 4 T-State.

30) RRC => Rotate Right accumulator. without carry.

It rotate the content of Accumulator to right by 1 bit position.



LSB will come out from Accumulator and will be copied to carry bit in flag register and also will be copied to MSB position of Accumulator.

It is 8 bit rotation of Accumulator content.

Only carry flag is affected.

1 byte, 1 machine cycle, 4 T-states

---

31) STA => store accumulator.

~~STO~~ STA 16 bit address. This instruction is used to store Accumulator 8 bit content to specified memory location.

This is absolute addressing mode.

No flags are modified

3 Byte, 4 machine cycle, 13 T-states.

---

32) STAX => Store Accumulator content in memory pointed by extended register RP.

STAX RP. This instruction uses register indirect addressing mode for specifying destination.

The content of Accumulator will be written to memory location pointed by 16 bit address as stored in Register pair.

1 Byte, 2 machine cycle, 7 T-states.

33) SUB => subtract.

SUB R => used to subtract content of Register specified from content of Accumulator.

1 machine cycle, 1 Byte, 4 T State. all flag modified.

SUB M => used to subtract content present in memory location pointed by HL pair from content of Accumulator.

1 Byte, 2 machine cycle, 7 T-State.

---

34) SUI => subtract Immediate.

SUI 8 bit data => subtract a 8 bit data from Accumulator content and store result in Accumulator.

all flags modified.

Immediate addressing mode.

2 Byte, 2 machine cycle, 7 T-State.

---

35) SBB => subtract with Borrow.

SBB R => subtract content of specified Register from Accumulator content along with Borrow. the result is stored in Accumulator.

1 Byte, 1 machine cycle, 4 T State

SBB M => Subtract content specified at memory location pointed by HL pair along with borrow.

1 Byte, 2 machine cycle, 7 T-State.

all flags are modified.

36) XRA ⇒ Exclusive OR Accumulator.

XRA R ⇒ perform logical Exclusive OR operation on Register content and Accumulator content.
Result is stored in Accumulator.

1 Byte, 1 machine cycle, 4 T-State.

XRA M ⇒ perform logical Exclusive OR operation on content specified at memory location pointed by HL pair.

1 Byte, 2 machine cycle, 7 T-State.

37) XRI ⇒ Exclusive OR Immediate with Accumulator.

XRI 8 bit data ⇒ this instruction is used to perform EX-OR between 8 bit data and Accumulator content.
The result is stored in Accumulator.

2 Byte, 2 machine cycle, 7-T States