

Arithmetic and Logic Instructions Groups

Lecture objectives: at the end of this lecture the student will able to:

- 1- Define the arithmetic and logic operations that can be executed in 8085 microprocessor.
- 2- Explain the operation of arithmetic instructions in 8085 microprocessor.
- 3- Explain the operation of logic instructions in 8085 microprocessor.
- 4- Write programs to execute the arithmetic and logic operation.

3.1 The arithmetic and logic operations:

3.1.1 Addition

Any 8-bit number, or the contents of register, or the contents of memory location can be added to the contents of accumulator. The result of summation operation is stored in the accumulator. Where, no two other 8-bit registers can be added directly.

3.1.2 Subtraction

Any 8-bit number, or the contents of register, or the contents of memory location can be subtracted from the contents of accumulator. The result is stored in the accumulator, where subtraction operation is performed in 2's complement form. If the result is negative, it is stored in 2's complement form. No two other 8-bit registers can be subtracted directly.

3.1.3 Increment / Decrement

The 8-bit contents of a register or a memory location can be incremented or decremented by 1. Also, the 16-bit contents of a register pair can be incremented or decremented by 1. Increment or decrement can be performed on any register or a memory location.

3.2 Arithmetic instructions

Opcode	Operand	Description
ADD	R M	Add register or memory to accumulator

A- ADD operand

[one byte instruction]

Add (A) to operand and save the result in accumulator. There are two types of **ADD** instruction according to type operand. Operand is one of registers (A, B, C, D, E, H, and L) or memory location which addressed by (HL).

ADD C Add (A) to (C) and save the result in accumulator.
ADD M Add (M_(HL)) to (A) and save the result in accumulator.

Opcode	Operand	Description
ADC	R M	Add register or memory to accumulator with carry

B- ADC operand**[one byte instruction]**

Add (A) and operand and CY and save the result in accumulator. There are two types from ADC instruction according to type operand. Operand is one of registers (A, B, C, D, E, H, and L) or memory location which addressed by (HL).

ADC B Add (B) with (A) with CY and save the result in accumulator.

ADC M Add (M_(HL)) with (A) with CY and save the result in accumulator.

Opcode	Operand	Description
ADI	8-bit data	Add immediate to accumulator

C- ADI, 8-bit data**[two bytes instruction]**

Add 8-bit number with (A) and save the result in accumulator.

ADI 37 Add 37 to (A) and save the result in accumulator.

Opcode	Operand	Description
ACI	8-bit data	Add immediate to accumulator with carry

D- ACI, 8-bit data**[two bytes instruction]**

Add 8-bit number with (A) with CY and save the result in accumulator.

ACI 37 add 37 with (A) with CY and save the result in accumulator.

Opcode	Operand	Description
SUB	R M	Subtract register or memory from accumulator

E- SUB operand**[one byte instruction]**

Subtract operand from (A) and save the result in accumulator. There are two types from SUB instruction according to type operand. Operand is one of registers (A, B, C, D, E, H, and L) or memory location which addressed by (HL).

SUB C subtract (C) from (A) and save the result in accumulator.

SUB M subtract (M_(HL)) from (A) and save the result in accumulator.

Opcode	Operand	Description
SBB	R M	Subtract register or memory from accumulator with borrow

F- SBB operand [one byte instruction]

Subtract operand and borrow from (A) and save the result in accumulator. There are two types of SBB instruction according to type operand. Operand is one of registers (A, B, C, D, E, H, and L) or memory location which addressed by (HL).

SBB B Subtract (B) and borrow from (A) and save the result in accumulator.
SBB M Subtract (M_(HL)) and borrow from (A) and save the result in accumulator.

Opcode	Operand	Description
SUI	8-bit data	Subtract immediate from accumulator

G- SUI, 8-bit data [two bytes instruction]

subtract 8-bit data from (A) and save the result in accumulator.

SUI 37 Subtract 37 from (A) and save the result in accumulator.

Opcode	Operand	Description
SBI	8-bit data	Subtract immediate from accumulator with borrow

H- SBI, 8-bit data [two bytes instruction]

subtract 8-bit data and borrow from (A) and save the result in accumulator.

SBI 37 Subtract 37 and borrow from (A) and save the result in accumulator.

The previous instructions has the following basics:

- 1- Implicitly the accumulator is one of the registers.
- 2- These instructions are modify all flags according to data conditions of results.
- 3- The result of arithmetic operation stored in the accumulator.
- 4- Don't affect the contents of the other register.

Opcode	Operand	Description
INR	R M	Increment register or memory by 1

I- INR operand [one byte instruction]

Increment the contents of operand by 1. There are two types of **INR** instruction according to operand type. Operand is one of registers (**A, B, C, D, E, H, and L**) or memory location which addressed by (**HL**).

INR E increment by one the contents of register E.
INR M increment by one the (**M_(HL)**).

Opcode	Operand	Description
DCR	R M	Decrement register or memory by 1

J- DCR operand [one byte instruction]

Decrement one from contents of operand. There are two types of **DCR** instruction according to operand type. Operand is one of registers (**A, B, C, D, E, H, and L**) or memory location which addressed by (**HL**).

DCR E decrement one from contents of register E.
DCR M decrement one from contents (**M_(HL)**).

The previous two instructions has the following basics:

- 1- This instruction affect the contents of specified register with instruction only.
- 2- This instruction affect all flags except CY flag.

Opcode	Operand	Description
INX	R	Increment register pair by 1

K- INX R_p [one byte instruction]

Increment one to contents of register pair. There is one types of **INX** instruction. Register pair is one of the (**BC, DE, HL**) and 16-bit register **SP**

INX D increment one to contents of register pair **DE**.

Opcode	Operand	Description
DCX	R	Decrement register pair by 1

L- DCX R_p [one byte instruction]

Decrement one from contents of register pair. There is one types of **DCX** instruction. Register pair is one of the (**BC, DE, HL**) and 16-bit register **SP**

DCX B decrement one from contents of register pair **BC**.

Note : **INX** and **DCX** instruction don't affect any flags.

Opcode	Operand	Description
DAD	Reg. pair	Add register pair to H-L pair

M- DAD, R.P. **[one byte instruction]**

Add contents of register pair to contents of **HL** and save the result in **HL**.

DAD B Add (BC) to (HL) and save the result in HL.

Note: this instruction affect CY flag only.

3.3 Logical group instructions:

These instructions perform logical operations on data stored in registers, memory and status flags. The logical operations are (AND, OR, XOR, Rotate, Compare, Complement).

3.3.1 AND, OR, XOR logical operation:

Any 8-bit data, or the contents of register, or memory location can logically have (AND or OR or XOR) operations with the contents of accumulator where, the result is stored in accumulator.

Opcode	Operand	Description
ANA	R M	Logical AND register or memory with accumulator

A- ANA operand **[one byte instruction]**

This instruction execute AND logical operation between (A) and contents of operand and save the result in accumulator. Operand is one of registers (**A, B, C, D, E, H, and L**) or memory location which addressed by (**HL**) as shown below:

ANA D AND logical operation between (**D**) and (**A**) and saving the result in accumulator.

Opcode	Operand	Description
ANI	8-bit data	Logical AND immediate with accumulator

B- ANI, 8-bit data **[two bytes instruction]**

This instruction execute AND logical operation between (A) and 8-bit number and save the result in accumulator.

ANI 6AAND logical operation between **6A** and (**A**) and saving the result in accumulator.

Opcode	Operand	Description
ORA	R M	Logical OR register or memory with accumulator

C- OR operand**[one byte instruction]**

This instruction execute OR logical operation between (**A**) and contents of operand and save the result in accumulator. Operand is one of registers (**A, B, C, D, E, H, and L**) or memory location which addressing by (**HL**) as shown below:

OR COR logical operation between (**C**) and (**A**) and saving the result in accumulator.

Opcode	Operand	Description
ORI	8-bit data	Logical OR immediate with accumulator

D- OR, 8-bit data**[two bytes instruction]**

This instruction execute the OR logical operation between (**A**) and 8-bit number and save the result in accumulator.

ORI 6A OR logical operation between **6A** and (**A**) and saving result in accumulator.

Opcode	Operand	Description
XRA	R M	Logical XOR register or memory with accumulator

E- XRA operand**[one byte instruction]**

this instruction execute Ex-OR logical operation between (**A**) and contents of operand and save the result in accumulator. The operand is one of registers (**A, B, C, D, E, H, and L**) or memory location which addressed by (**HL**) as shown below:

XRA EEx-OR logical operation between (**E**) and (**A**) and saving the result in accumulator.

Opcode	Operand	Description
XRI	8-bit data	XOR immediate with accumulator

F- XRI, 8-bit data**[two bytes instruction]**

This instruction execute Ex- OR logical operation between (**A**) and 8-bit number and save the result in accumulator.

XRI 6A Ex-OR operation between **6A** and (**A**) and saving result in accumulator.

The previous instructions have following features:

- The result is placed in the accumulator.
- S, Z, P are modified to reflect the result.
- CY and AC are reset in OR and EX-OR operations, while in AND operation CY is reset and AC is set.

3.3.2 Rotate operation:

Each bit in the accumulator can be shifted either left or right to the next position.

Opcode	Operand	Description
RAR	None	Rotate accumulator right through carry

A- RAR

[one byte instruction]

Rotate the content of Accumulator to **R**ight with carry. Fig. 3.1 shows the rotate operation.

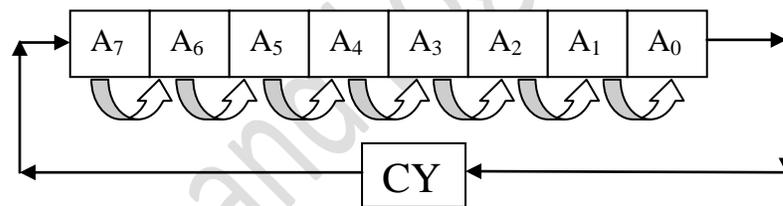


Figure 3.1: RAR instruction

Note: CY is modified according to bit D₀. S, Z, P, AC are not affected.

Opcode	Operand	Description
RAL	None	Rotate accumulator left through carry

B- RAL

[one byte instruction]

Rotate the content of Accumulator to **L**eft with carry. Fig. 3.2 shows the rotate operation.

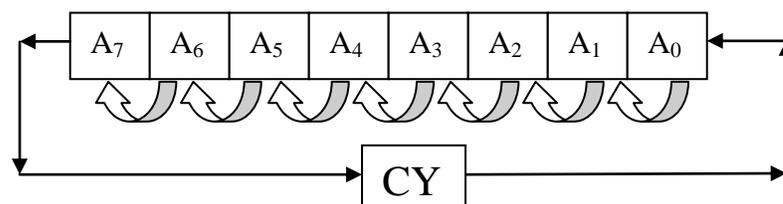


Figure 3.2: RAL instruction

Note: CY is modified according to bit D7, while, S, Z, P, AC are not affected.

Opcode	Operand	Description
RRC	None	Rotate accumulator right

C- RRC

[one byte instruction]

Rotate the content of Accumulator to **R**ight without carry. Fig. 3.3 shows the rotate operation.

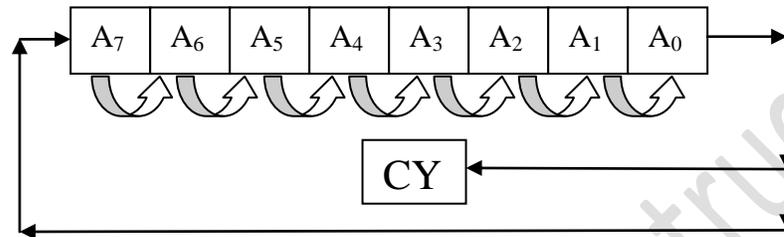


Figure 3.3: RRC instruction.

Note: CY is modified according to bit D0. S, Z, P, AC are not affected.

Opcode	Operand	Description
RLC	None	Rotate accumulator left

D- RLC

[one byte instruction]

Rotate the content of Accumulator to **L**eft without carry. Figure 3.4 below shows the rotate operation.

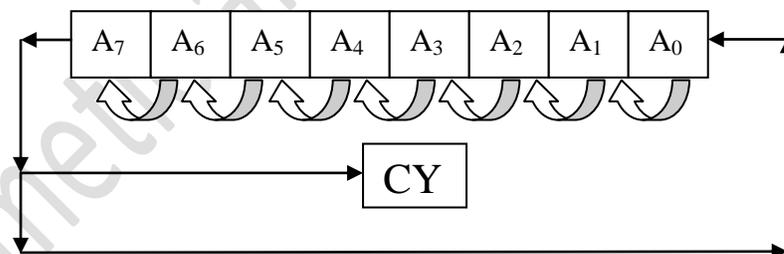


Figure 3.4: RLC instruction.

Note: CY is modified according to bit D7, S, Z, P, AC are not affected.

3.3.3 Complement:

The contents of accumulator can be complemented where, each 0 is replaced by 1 and each 1 is replaced by 0.

Opcode	Operand	Description
CMA	None	Complement accumulator

A- CMA [one byte instruction]

NOT logical operation (complement) to contents of accumulator.

Note: No flags are affected by this instruction.

Opcode	Operand	Description
STC	None	Set carry

B- STC [one byte instruction]

Set CY flag. (CY=1)

Note: No other flags are affected by this instruction.

Opcode	Operand	Description
CMC	None	Complement carry

C- CMC [one byte instruction]

NOT logical operation (complement) to CY flag.

Note: No other flags are affected by this instruction.

3.3.4 Compare:

Any 8-bit data, or the contents of register, or memory location can be compared for:

- Equality
- Greater Than
- Less Than

with the contents of accumulator.

The result is reflected in status flags.

Opcode	Operand	Description
CMP	R M	Compare register or memory with accumulator

A- CMP operand [one byte instruction]

The contents of the operand (register or memory) are compared with the contents of the accumulator. Both contents are preserved. The flags are modified according to cases below:

If (A) > operand: [CY=0,Z=0]

If (A) < operand: [CY=1,Z=0]

If (A) = operand: [CY=0,Z=1]

Operand is one of register (A, B, C, D, E, H, and L) or (M_{HL}).

CMP E compare between (E) and (A).

Opcode	Operand	Description
CPI	8-bit data	Compare immediate with accumulator

B- CPI operand [two bytes instruction]

The 8-bit data is compared with the contents of accumulator. The values being compared remain unchanged. The flags are modified according to cases below:

If (A) > data: [CY=0,Z=0]

If (A) < data: [CY=1,Z=0]

If (A) = data: [CY=0,Z=1]