Arithmetic Instructions(affect flags)

- **Addition**
  - In addition, accumulator(A) is an implicit operand in this instruction and result is always stored in A
  - Instruction: **ADD** reg/M; A <- A+reg/M

    (1 byte instruction)

    e.g. ADD B; A <- A+B

    ADD L;   A <- A+L

    ADD M; A <- A+[HL]

  - Instruction: **ADI** 8-bit value; A <- A+value

    (2 bytes instruction)

    e.g. ADI 33H; A <- A+33H

Arithmetic Instructions(affect flags)

- **Addition**
  - Write a program in 8085 to add A5H and 7CH. Also, examine the status of flag.

```
MVI A, A5H                MVI A, A5H
ADI 7CH                   MVI B, 7CH
HLT                       ADD B
                          HLT
```

```
 A5H:     1010 0101        Flags:
+7CH:     0111 1100        S Z X AC X P X CY
        ___________
       1  0010  0001       0 0 X  1  X 1 X  1
```

  - Write a program in 8085 to add to numbers stored in memory 3050H and 3060H and store the result in 3070H.

```
LDA 3050H;  A <- [3050H]                  LDA 3050H
MOV B, A                                  LXI H, 3060H
LDA 3060H;   A <- [3060H]                 ADD M
ADD B;  A <- A+B                          STA 3070H
STA 3070H;  [3070H] <- A
```

Arithmetic Instructions(affect flags)

- **Addition**
  - Instruction: **ADC** reg/M; A <- A+reg/M+CY

    - Add with carry

    (1 byte instruction)

    e.g. ADC B; A <- A+B+CY

    ADC M; A <- A+[HL]+CY

  - Instruction: **ACI** 8-bit value; A <- A+value+CY

    - Add immediate value with carry

    (2 bytes instruction)

    e.g. ACI A9H; A <- A+A9H+CY

Arithmetic Instructions(affect flags)

- **Addition**
  - Instruction: **DAD** reg_pair; HL <- HL+reg_pair

    - Double Addition.

    - 16-bit contents of the register pair are added to the contents of H-L pair.

    - The result is stored in HL pair.

    - Only carry flag is affected by this instruction

      e.g. DAD B; HL <- HL + BC

Arithmetic Instructions(affect flags)

- **Addition**

Write a program in 8085 to add 2379H and 431FH and store the result in HL pair. Perform this addition with and without using DAD instruction.

Without using DAD

```
  MVI A, 79H
  ADI 1FH
  MOV L, A
  MVI A, 23H
  ACI 43H
  MOV H, A
  HLT
```

Using DAD

```
LXI H, 2379H
LXI B, 431FH
DAD B
HLT
```

Arithmetic Instructions(affect flags)

- **Subtraction**
  - In subtraction, accumulator(A) is an implicit operand in this instruction and result is always stored in A
  - Subtraction is performed using 2’s complement method.
  - Instruction: **SUB** reg/M; A <- A-reg/M

    (1 byte instruction)

    e.g. SUB D; A <- A-D

    SUB H; A <- A-H

    SUB M; A <- A-[HL]
  - Instruction: **SUI** 8-bit value; A <- A-value

    (2 bytes instruction)

    e.g. SUI C8H; A <- A-C8H

Arithmetic Instructions(affect flags)

- **Subtraction**
  - Write a program in 8085 to subtract 29H from 7CH. Also, examine the status of flag.

```
MVI A, 7CH       MVI A, 7CH
MVI B, 29H       SUI 29H
SUB B            HLT
HLT
```

<u>2’s complement method</u>

```
                           7CH: 0111 1100
                          -29H: 0010 1001
2’s complement of 29H:          1101 0111
                  Add 7CH:      0111 1100
                                ---------
                              1 0101 0011   → 53H
```

Flags: S->0, Z->0, AC->1, P->1, CY->0

(Note: Carry is complemented in subtraction)

  - Write a program in 8085 to subtract 8AH from 37H. Also, examine the status of flag.
  - Write a program in 8085 to subtract to numbers stored in memory 3050H and 3060H and store the result in 3070H.

Arithmetic Instructions(affect flags)

- **Subtraction**
  - Instruction: **SBB** reg/M; A <- A-reg/M-CY

    - Subtract with borrow

    (1 byte instruction)

    e.g. SBI B; A <- A-B-CY

    SBI M; A <- A-[HL]-CY

  - Instruction: **SBI** 8-bit value; A <- A-value-CY

    - Subtract immediate value with borrow

    (2 bytes instruction)

    e.g. SBI A9H; A <- A-A9H-CY

Arithmetic Instructions(affect flags)

- **Increment Instruction**
  - Instruction: **INR** reg/M; reg/M <- reg/M+1

    - Increment the content of register or memory by 1
    - Affects all flags except Carry Flag

    (1 byte instruction)

    e.g. INR C; C <- C+1
    INR M; [HL] <- [HL] + 1

  - Instruction: **INX** reg_pair; reg_pair <- reg_pair + 1

    - Increment the content of register pair by 1
    - No flags are affected

    (1 byte instruction)

    e.g. INX B; BC <- BC + 1
    INX H; HL <- HL + 1

Arithmetic Instructions(affect flags)

- **Decrement Instruction**
  - Instruction: **DCR** reg/M; reg/M <- reg/M-1

    - Decrement the content of register or memory by 1

    - Affects all flags except Carry Flag

    (1 byte instruction)

    e.g. DCR L; L <- L-1

    DCR M; [HL] <- [HL] - 1

  - Instruction: **DCX** reg_pair; reg_pair <- reg_pair - 1

    - Decrement the content of register pair by 1

    - No flags are affected

    (1 byte instruction)

    e.g. DCX D; DE <- DE - 1

    DCX H; HL <- HL - 1

Arithmetic Instructions(affect flags)

- **Comparison Instruction**
  - Instruction: **CMP** reg/M; A – reg/M

    - Compares the value of register or memory with the value of accumulator

    - The comparison is performed by subtracting the second operand from accumulator without being altering the content of accumulator.

    - The result of comparison is reflected in flag register by setting/resetting different flags.

    (1 byte instruction)

    e.g. CMP D; A - D

    CMP M; A - [HL]

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

Arithmetic Instructions(affect flags)

- **Comparison Instruction**
  - Instruction: **CPI** 8-bit value; A – value

    - Compares the given value with the value of accumulator

    - Operation is similar as CMP instruction

    (2 bytes instruction)

    e.g. CPI 5BH; A – 5BH