# LAB 4

**AIM:** TO EXPLORE AND ANALYZE BRANCH INSTRUCTIONS IN 8085.

## PRACTICE ASSIGNMENT

### Assignment

Explore the following Branching Instructions of 8085: JP, JM, JPE, JPO.

**JP (Jump if Plus):**

- Syntax: `JP <address>`
- Jump to the given address if the Sign flag is not set meaning value is positive.

**JM (Jump if Minus):**

- Syntax: `JM <address>`
- Jump to the given address if the Sign flag is set meaning value is negative.

**JPE (Jump if Parity Even):**

- Syntax: `JPE <address>`
- Jump to the given address if the Parity flag is set meaning the value is Even.

**JPO (Jump if Parity Odd):**

- Syntax: `JPO <address>`
- Jump to the given address if the Parity flag is not set meaning the value is Odd.

## LAB ASSIGNMENT

### Question - 1

Write an Assembly language program in 8085 to find the sum of a series of numbers. Store 5 numbers starting from memory location 3300H. Store the result at the memory location 3310H.

**Code:**

```
; Initialize the memory with numbers
MVI C, 05H          ; Initialize the counter to 5
MVI A, 01H          ; Initialize the first number to 1
LXI H, 3300H        ; Initialize the memory block starting point to 3300H

INIT_LOOP:
    MOV M, A        ; Store the value of A into memory at address HL
    INR A           ; Increment A for the next number
    INX H           ; Increment HL for the next memory address
    DCR C           ; Decrement the counter
    JNZ INIT_LOOP   ; Loop until the counter becomes 0
```

```
; Calculate the sum of the numbers
MVI C, 05H          ; Initialize the counter to 5
MVI A, 00H          ; Initialize the accumulator with the value 0
LXI H, 3300H        ; Initialize the HL pair to point to the memory location 3300H

SUM_LOOP:
    ADD M           ; Add the number at the memory location to the accumulator
    INX H           ; Increment the Pointer to point to the next number
    DCR C           ; Decrement the counter
    JNZ SUM_LOOP    ; Loop until the counter becomes 0

; Store the sum in memory location 3310H
LXI H, 3310H        ; Load memory location 3310H into HL
MOV M, A            ; Store the value of the accumulator to the location pointed
by HL

HLT                 ; Stop the execution
```

## Question - 2

Write an Assembly Language program in 8085 to store 10 bytes starting from 2500H , subtract two consecutive bytes, if the result is negative store at memory location 2510H onwards.

**INPUT:** `20, 10, 12, 18, 56, 42, 78, A0, 45, 31.`

**OUTPUT:** `FA, D8`

**Code:**

```
; Initialize pointers and counter
LXI H, 2500H     ; HL points to 2500H (start of input)
LXI D, 2510H     ; DE points to 2510H (output start)
MVI C, 09H       ; Initialize counter to 9 (subtraction happens 9 times)

; Process each pair of bytes
NEXT_PAIR:
  MOV A, M        ; Load current byte into A
  INX H           ; Increment HL to point to next byte
  SUB M           ; Subtract next byte from A
  DCR C           ; Decrement counter
  JZ DONE         ; Jump to DONE if counter reaches 0

  JC STORE_RESULT ; If result is negative, store it

  INX H           ; Increment the HL pair to get the next two numbers
  JP NEXT_PAIR    ; If result is not negative, move to next pair

STORE_RESULT:
  STAX D          ; Store the negative result at memory pointed by DE
  INX D           ; Increment DE to point to next result location
  INX H           ; Increment the HL pair to get the next two numbers
```

```
  JP NEXT_PAIR    ; Continue processing next pair

DONE:
HLT               ; Halt the program
```

## Question - 3

Write an Assembly Language program to search a given byte in the list. If the element is found then store the index/location of that element in the memory. If the element is not found then store FFFFH in the memory.

**INPUT Test Case 1:**

```
(8000H): 06H, Store the count of total numbers
(8001H): 55H, The number to be searched
(8002H to 8007H): 11H, 22H, 33H, 44H, 55H, 66H
```

**OUTPUT:**

```
(8010H): 80
(8011H): 06
```

**INPUT Test Case 2:**

```
(8000H): 06H, Store the count of total numbers
(8001H): 77H, The number to be searched
(8002H to 8007H): 11H, 22H, 33H, 44H, 55H, 66H
```

**OUTPUT:**

```
(8010H): FF
(8011H): FF
```

**Code:**

```
; Memory Initialization for search process
LXI H, 8000H        ; HL points to 8000H (First byte - count of total numbers)
MOV A, M            ; Load total count (6) from memory
MOV B, A            ; Copy count to B (B will hold the remaining numbers to
search)
INX H               ; HL now points to 8001H (search number)
MOV C, M            ; Load the number to be searched into C (55H)
INX H               ; HL points to the start of the list (8002H)
```

```
LXI D, 8010H        ; DE points to 8010H where the result (index or FF) will be
stored

; Search Loop
SEARCH_LOOP:
    MOV A, M        ; Load current list number into A
    CMP C           ; Compare A with C (search number)
    JZ FOUND        ; If numbers match, jump to FOUND

    INX H           ; Increment HL to the next number
    DCR B           ; Decrement count in B
    JZ NOT_FOUND    ; If counter reaches zero, jump to NOT_FOUND
    JMP SEARCH_LOOP ; Continue the search loop

; If Found
FOUND:
    MOV A, H        ; Copy the high byte (H) to A
    STAX D          ; Store the high byte of the address at DE (8010H)
    INX D           ; Move DE to point to 8011H
    MOV A, L        ; Copy the low byte (L) to A
    STAX D          ; Store the low byte of the address at DE (8011H)
    JMP END         ; Jump to END

; If Not Found
NOT_FOUND:
    MVI A, 0FFH     ; Store the value FF in the accumulator
    STAX D          ; Store the value of A at the memory location pointed by DE
pair
    INX D           ; Increment the DE pair
    STAX D          ; Store the value of A at the memory location pointed by DE
pair

END:
HLT                 ; Stop the program
```

## Question - 4

Write an Assembly Language program to sum 10 bytes of data, the sum may exceed 8 bits, store the lower byte of the sum at memory location 3000H and upper byte at 3010H.

**INPUT:** `E4, A0, FF, CD, A4, 16, F1, B2, 56, 67` **OUTPUT:**

```
(3000H): 06
(3010H): 6A
```

**Code:**

```
; Initialization
LXI H, 8000H    ; HL points to the starting location 8000H
```

```
MVI B, 00H      ; Clear register B (used for the upper byte of sum)
MVI C, 00H      ; Clear register C (used for the lower byte of sum)
MVI E, 10H      ; Set counter to 10 for the number of bytes

; Sum both the numbers (Handle 16-bit sum: low byte in C, high byte in B)
CONTINUE_SUM:
  MOV A, M       ; Load current byte from memory into A
  ADD C          ; Add the lower byte (C) into A (accumulator)
  MOV C, A       ; Store result of the lower byte sum in C (low byte)
  MVI A, 00H     ; Clear the accumulator
  ADC B          ; Add high byte (B) and carry to A
  MOV B, A       ; Store updated value back to B (high byte)
  INX H          ; Increment HL to point to the next byte of data
  DCR E          ; Decrement the counter
  JNZ CONTINUE_SUM ; Repeat the process until the counter reaches zero

; Store the result at the memory locations 3000H and 3001H
LXI H, 3000H    ; Set HL to 3000H to store the result
MOV M, B        ; Store the lower byte (sum) at 3010H
LXI H, 3010H    ; Change HL to 3001H
MOV M, C        ; Store the upper byte (carry or high byte) at 3000H

HLT             ; End the program
```

## Question - 5

Write an 8085 Assembly language program to multiply two 8-bit numbers stored in memory location and store the 16-bit results into the memory.

**Code:**

```
; Initialization
LXI H, 3000H      ; HL points to memory location 3000H where the inputs are stored
MOV B, M          ; Load the first number (multiplicand) into B
INX H             ; Increment HL to point to the second number
MOV C, M          ; Load the second number (multiplier) into C
MVI D, 00H        ; Clear D (upper byte of the product)
MVI E, 00H        ; Clear E (lower byte of the product)

; Multiply using repeated addition
MULTIPLY:
  MOV A, E        ; Load the lower byte of the result (E) into A
  ADD B           ; Add the multiplicand (B) to A
  MOV E, A        ; Store the result back to E
  MVI A, 00H      ; Clear the accumulator for the upper byte calculation
  ADC D           ; Add the carry (if any) to the upper byte (D)
  MOV D, A        ; Store the result back to D
  DCR C           ; Decrement the multiplier (C)
  JNZ MULTIPLY    ; Repeat until multiplier (C) becomes zero

; Store the result
```

```
LXI H, 3010H      ; HL points to the memory location 3010H
MOV M, D          ; Store the upper byte of the result at 3010H
INX H             ; Increment HL to 3011H
MOV M, E          ; Store the lower byte of the result at 3011H

HLT               ; End of the program
```