# REPORT

**Microprocessor Lab**

**001910501005**

**Atanu Ghosh**

**JU BCSE UG-II Sem-II**

# Assignment – 1

**Q1 : Load the contents of the memory locations 2200H and 2201H into registers. Add these registers and store the result in memory locations 2202H and 2203H.**

**Sol :**

```
;<Program title>

JMP Start                       ; Jump to START

Start: LXI H,2200H      ; Store 22H to H register and 00H to L register
       MOV A,M          ; Copy contents of Location 2200H to accumulator
       INX H            ; Increase the address of H-L register pair by 1
       MVI D,00H        ; Store 00H to register D
       ADD M            ; Add content of accumulator to contents of memory location from HL
       JNC Loop         ; If no carry, enter the LOOP
       INR D            ; Increase the content of register D by 1

Loop:  STA 2203H        ; Store contents of accumulator to memory location 2203H
        MOV A,D                 ; Copy contents of register D to accumulator

       STA 2202H        ; Store contents of accumulator to memory location 2202H

HLT                                     ; Terminates the program
```

**Q2 : Find the sum of N numbers stored in consecutive locations starting from 2500H. The value of N is stored in 2200H. Store the result in locations 2300H and 2301H.**

**Sol :**

```
;<Program title>

JMP Start           ; Jump to START

Start: LDA 2200H    ; Load accumulator with content of location 2200H
       MOV B,A      ; Copy content of accumalator to register B / Setting the Count(N)
       LXI H,2500H ; Store 25H to H register and 00H to L register
       MVI A,00H    ; Store 00H to accumulator / Initializing Sum
       MVI C,00H    ; Store 00H to register C / Initializing Carry

Skip:  ADD M        ; Add content of accumulator with contents of memory location from H-L register pair
           INX H            ; Increase the address of H-L register pair by 1
       JNC Loop     ; If no carry, enter the LOOP
       INR C        ; Increase the content of register C by 1 / Increment the Carry(C = C+1)

Loop:  DCR B        ; Decrease the content of register B by 1 / Decrement the Count(N = N-1)
           JNZ Skip     ; If not zero, jump to SKIP

       STA 2300H    ; Store contents of accumulator to memory location 2300H
       MOV A,C      ; Copy contents of register C to accumulator / Store the Carry
       STA 2301H    ; Store contents of accumulator to memory location 2301H

HLT                         ; Terminates the program
```

**Q3 : Find the sum of the least significant 4 bits and most significant 4 bits of a byte stored in memory location 2500H. Store the result in 2550H.**

**Sol :**

```
;<Program title>

JMP Start                           ; Jump to START

Start:  LXI H, 2500H    ; Store 25H to H register and 00H to L register
        MOV A, M                    ; Copy contents of Location 2500H to accumulator
        ANI 0FH                     ; Perform AND operation of A with 0F and store it to A

        MOV D, A                    ; Copy the content of accumulator to D register
        MOV A, M                    ; Copy the content of Memory to accumulator

        RRC                                 ; Rotate right accumulator
        RRC                                 ; Rotate right accumulator
        RRC                                 ; Rotate right accumulator
        RRC                                 ; Rotate right accumulator

        ANI 0FH                     ; Perform AND operation of A with 0F and store it to A
        ADD D                       ; Add content of D register with accumulator content
    STA 2550H               ; Store contents of accumulator to memory location 2300H

HLT;                            ; Terminates the program
```

**Q4 : Write a program to count the '1's and '0's of a byte stored in 2500H. Store the result in 2610H and 2511H, respectively**

**Sol :**

```
;<Program title>

JMP Start                       ; Jump to START

Start: LXI H, 2500H     ; Store 25H to H register and 00H to L register
       MOV A,M                  ; Copy the content of Memory to accumulator
       MVI B,08H        ; Store 08H to register B
       MVI D,00H        ; Store 00H register D

LOOP:  RLC                      ; Rotate left accumulator
       JNC SKIP                 ; If no carry, jump to SKIP
       INR D                    ; Increase the content of register D by 1

SKIP:  DCR B            ; Decrease the content of register B by 1
       JNZ LOOP                 ; If not zero, jump to LOOP

       MOV A,D                  ;  Copy the content of register D to accumulator
       STA 2610H        ; Store contents of accumulator to memory location 2610H
       MOV B,A                  ; Copy the content of accumulator to register B
       MVI A,08H        ; Store 08H to register A
       SUB B                    ; Subtract content of D register from accumulator content
       STA 2511H        ; Store contents of accumulator to memory location 2511H

HLT                             ; Terminates the program
```

**Q5 : Write a program to sum two 16-bits binary numbers.**

**Sol :**

```
;<Program title>

JMP Start                       ; Jump to START

Start: LHLD 2200H       ; Load HL direct to 22H and 00H respectively
           XCHG                 ; Exchange contents of DE with contents of HL
       LHLD 2210H       ; Load HL direct to 22H and 00H respectively
       DAD D            ; contents of HL = contents of HL + contents of DE
       MVI B,00H        ; Store 00H register B
       JNC Loop         ; If no carry, enter the LOOP
       INR B            ; Increase the content of register B by 1

Loop:  SHLD 2220H       ; Store HL direct to 22H and 20H respectively
           MOV A,B              ; Copy the content of register B to accumulator

       STA 2223H        ; Store contents of accumulator to memory location 2223H

HLT                                 ; Terminates the program
```

# Assignment - 2

**Q1 : Two numbers MNH and KLH are stored in 2050H and 2051H, respectively. Write a program to assemble them as NKH and LMH store them in 2052H and 2053H.**

**Sol :**

```
;<Swap nibbles of two 8 bit numbers and store them in specific addresses>

JMP Start                   ; starts the execution

Start: LXI H, 2050H     ; take the first number as input
          MOV A, M          ; move first number (MNH) to accumulator A
       LXI H, 2051H     ; take the second number as input
       MOV B, M         ; move second number (KLH) to reg B

; calculation for the 2052H position
       ANI 0FH          ; AND immediate with 0FH, result 0NH
       RRC                      ; Rotate right accumulator
       RRC                      ; Rotate right accumulator
       RRC                      ; Rotate right accumulator
       RRC                      ; Rotate right accumulator, result N0H
       MOV D, A         ; move N0H to reg D
       MOV A, B         ; move second number (KLH) to accumulator A
       ANI F0H          ; AND immediate with F0H, result K0H
       RRC                      ; Rotate right accumulator
       RRC                      ; Rotate right accumulator
       RRC                      ; Rotate right accumulator
       RRC                      ; Rotate right accumulator, result 0KH
       ADD D            ; add N0H & 0KH, result NKH
       STA 2052H        ; store NKH at memory address 2052H

; calculation for the 2053H position
       LDA 2050H        ; load accumlator with the first number (MNH)
       ANI F0H          ; AND immediate with F0H, result M0H
       RRC                      ; Rotate right accumulator
       RRC                      ; Rotate right accumulator
       RRC                      ; Rotate right accumulator
       RRC                      ; Rotate right accumulator, result 0MH
       MOV D, A         ; move 0MH to reg D
       MOV A, B         ; move second number (KLH) to accumulator A
       ANI 0FH          ; AND immediate with 0FH, result 0LH
       RRC                      ; Rotate right accumulator
       RRC                      ; Rotate right accumulator
       RRC                      ; Rotate right accumulator
       RRC                      ; Rotate right accumulator, result L0H
       ADD D            ; add L0H & 0MH, result LMH
       STA 2053H        ; store LMH at memory address 2053H

HLT                         ; terminates the execution
```

**Q2 : Two numbers A & B are stored in 2050H and 2051H, respectively. Write a program to perform A×B and store the result in 2052H and 2053H.**

**Sol :**

```
;<Multiplication of two 8 bit numbers with carry >

JMP Start                       ; starts the execution

Start: LXI H, 2050H     ; take the first number as input
           MOV B, M             ; move first number to reg B
       LXI H, 2051H     ; take the second number as input
       MOV C, M         ; move second number to reg C

       MVI A, 00H       ; initialise reg A so that no garbage value resides in it
       MVI D, 00H       ; initialise the value carry in reg D

Loop:  ADD B            ; add numIn B with numIn accumulator A
           JNC Skip             ; if no carry generated jump to SKIP
       INR D            ; increment the carry if carry is generated

Skip:  DCR C            ; decrement the second number by 1
           JNZ Loop             ; if result is not 0 jump to LOOP
       STA 2053H        ; store the number of accumulator at memory address 2053H
       MOV A, D         ; move the carry part from reg D to accumulator
       STA 2052H        ; store the carry of accumulator at memory addess 2052H
           HLT                  ; terminates the execution
```

**Q3 : N numbers are stored in consecutive m/m location starting from 2050H. The value N is stored in 204FH.**

**i) Find the maximum among the N numbers.**

**Sol :**

```
;<Find the MAXIMUM number from a list of numbers>

JMP Start                              ; starts the execution

Start:  LXI H, 204FH    ; taking count of numbers as input
                MOV D, M               ; storing that count in reg D
        LXI H, 2050H    ; taking first input of the list of numbers
        MOV A, M               ; moving taken list input to accumulator
        DCR D                  ; decrement the count of numbers by 1

Loop:   INX H                  ; increment the inputIndexValue by 1
        CMP M                  ; compare content of memory M[HL] with content of accumulator A
        JNC Skip               ; if numIn A >= numIn M[HL] jump to SKIP
        MOV A, M               ; if numIn A < numIn M[HL] move content of M to A

Skip:   DCR D                  ; decrement the count of numbers by 1
                JNZ Loop               ; if count isNotEqual to 0 jump to LOOP
        STA 204DH              ; store the larger number at memory address 204DH
                HLT                            ; terminates the execution

```

**ii) Find the minimum among the N numbers.**

**Sol :**

```
;<Find the MINIMUM number from a list of numbers>

JMP Start                              ; starts the execution

Start:  LXI H, 204FH    ; taking count of numbers as input
                MOV D, M               ; storing that count in reg D
        LXI H, 2050H    ; taking first input of the list of numbers
        MOV A, M               ; moving taken list input to accumulator
        DCR D                  ; decrement the count of numbers by 1

Loop:   INX H                  ; increment the inputIndexValue by 1
        CMP M                  ; compare content of memory M[HL] with content of accumulator A
        JC Skip                ; if numIn A < numIn M[HL] jump to SKIP
        MOV A, M               ; if numIn A >= numIn M[HL] move content of M to A

Skip:   DCR D                  ; decrement the count of numbers by 1
                JNZ Loop               ; if count isNotEqual to 0 jump to LOOP
        STA 204DH              ; store the smaller number at memory address 204DH
                HLT                            ; terminates the execution

```

### iii) Sort the N numbers in ascending order.

**Sol :**

```
;<Sort given N numbers in ascending order>

JMP Start                           ; starts the execution

Start:  LXI H, 204FH    ; taking count of numbers as input
        MOV C, M                    ; storing that count in reg C
        DCR C                   ; decrement the count of numbers by 1
        JZ SKIP1                    ; if no number is present in the list jump to SKIP1

LOOP1:  MOV B, C                ; moving the count to reg B
        LXI H,2050H                 ; taking first input of the list of numbers

LOOP2:  MOV A, M                ; moving current list input to accumulator
        INX H                       ; increment inputIndexValue by 1 to take next input
        MOV E, M                    ; moving the next taken input to reg E
        CMP E                       ; comparing num [i] with num [i + 1]
        JC SKIP2                    ; if num [i] < num [i+1] jump to SKIP2
        DCX H                       ; if num [i] >= num [i+1] return to previous index
        MOV M, E                    ; store the (i+1)-th input at i-th position
        INX H                       ; increment inputIndexValue by 1
        MOV M, A                    ; store the i-th input at (i+1)-th position // swapping complete

SKIP2:  DCR B                   ; decrement the max index of input array by 1
        JNZ LOOP2                   ; if max index isNotEqualTo 0 jump to LOOP2
        DCR C                       ; decrement the count of numbers by 1
        JNZ LOOP1                   ; if count isNotEqualTo 0 jump to LOOP1

SKIP1:  HLT                     ; terminates the execution
```

### iv) Sort the N numbers in descending order.

**Sol :**

```
;<Sort given N numbers in descending order>

JMP Start                           ; starts the execution

Start:  LXI H, 204FH    ; taking count of numbers as input
        MOV C, M                    ; storing that count in reg C
        DCR C                   ; decrement the count of numbers by 1
        JZ SKIP1                    ; if no number is present in the list jump to SKIP1

LOOP1:  MOV B, C                ; moving the count to reg B
        LXI H,2050H                 ; taking first input of the list of numbers

LOOP2:  MOV A, M                ; moving current list input to accumulator
        INX H                       ; increment inputIndexValue by 1 to take next input
        MOV E, M                    ; moving the next taken input to reg E
        CMP E                       ; comparing num [i] with num [i + 1]
        JNC SKIP2                   ; if num [i] >= num [i+1] jump to SKIP2
        DCX H                       ; if num [i] < num [i+1] return to previous index
        MOV M, E                    ; store the (i+1)-th input at i-th position
        INX H                       ; increment inputIndexValue by 1
        MOV M, A                    ; store the i-th input at (i+1)-th position // swapping complete

SKIP2:  DCR B                   ; decrement the max index of input array by 1
        JNZ LOOP2                   ; if max index isNotEqualTo 0 jump to LOOP2
        DCR C                       ; decrement the count of numbers by 1
        JNZ LOOP1                   ; if count isNotEqualTo 0 jump to LOOP1

SKIP1:  HLT                     ; terminates the execution
```

**Q4 : N numbers are stored in consecutive m/m location starting from 2050H. The value N is stored in 204FH. Write a program to copy the even and odd numbers starting from 2100H and 2200H, respectively. Store the total no. of even and odd numbers in 2300H and 2201H, respectively.**

## Sol :

```
;<Store and count number of even and odd numbers from a given list of numbers>

JMP Start                           ; starts the execution

Start: LXI H,204FH      ; taking count of numbers as input
          MOV B,M               ; storing that count in reg B
          LXI H,2050H   ; taking first input of the list of numbers
          MVI C,00H     ; initialise reg C with 00H
          LXI D,2200H   ; load reg pair [DE] with content from adress 2200H

LOOP:  MOV A,M          ; move input number to accumulator A
          RAR                   ; Rotate right through Carry
          JNC SKIP              ; ifNoCarryFound, jump to SKIP
          INR C                 ; increment the count of odd numbers
          RAL                   ; Rotate left through Carry
          XCHG                  ; exchange contents of M[HL] with contents of [DE]
          MOV M,A               ; move accumulator content to M[HL]
          XCHG                  ; exchange contents of M[HL] with contents of [DE]
          INX D                 ; increment the reg pair address of DE by 1

SKIP:  DCR B            ; decrement the count of input numbers by 1
          INX H                 ; take the next input
          JNZ LOOP              ; if count isNotEqualTo 0 jump to LOOP
          MOV A,C               ; move the count of odd numbers to accumulator
          STA 2301H     ; store the oddNumberCount at address 2301H
          LXI H,204FH   ; taking count of numbers as input
          MOV B,M               ; storing that count in reg B
          LXI H,2050H   ; taking first input of the list of numbers
          MVI C,00H     ; initialise reg C with 00H
          LXI D,2100H   ; load reg pair [DE] with content from adress 2100H

LOOP1: MOV A,M          ; move input number to accumulator A
          INX H                 ; take the next input
          RAR                   ; Rotate right through Carry
          JC SKIP1              ; ifCarryFound, jump to SKIP1
          INR C                 ; increment the count of even numbers
          RAL                   ; Rotate left through Carry
          XCHG                  ; exchange contents of M[HL] with contents of [DE]
          MOV M,A               ; move accumulator content to M[HL]
          XCHG                  ; exchange contents of M[HL] with contents of [DE]
          INX D                 ; increment the reg pair address of DE by 1

SKIP1: DCR B            ; decrement the count of input numbers by 1
          JNZ LOOP1     ; if count isNotEqualTo 0 jump to LOOP1
          MOV A,C               ; move the count of even numbers to accumulator
          STA 2300H     ; store the evenNumberCount at address 2301H
          HLT                   ; terminates the execution

```

**Q5 : N numbers are stored in consecutive m/m location starting from 2050H. The value N is stored in 204FH. Write a program to test whether a number stored in 204EH is present in the list. If present, store its position in the list at 204DH; otherwise store FFH.**

**Sol :**

```
;<Find a given number from a list of given numbers>

JMP Start                           ; starts the execution

Start:  LXI H, 204FH    ; taking count of numbers as input
                MOV D, M                ; storing that count in reg D
        LXI H, 204EH    ; taking the number to be searched from the list as input
        MOV E, M                ; storing that number in reg E
        MVI C, 00H              ; initialise position of the number that is to be searched
        LXI H, 2050H    ; taking first input of the list of numbers

Loop:   MOV A, M                ; moving taken list input to accumulator
                CMP E                   ; compare content of reg E with content of accumulator A
        JNZ Skip                ; if numbers in E and A areNotEqual then jump to SKIP
        MOV A, C                ; move the positionIndexValue of the number to accumulator A
        STA 204DH               ; store that positionIndexValue at address 204DH
        HLT                             ; halts the execution

Skip:   MVI A, FFH              ; finalise the position as FFH ifNumberNotMatched
                STA 204DH               ; store that positionIndexValue at address 204DH
        INX H                   ; increment the inputIndexValue by 1
        INR C                   ; increment the positionIndexValue by 1
        DCR D                   ; decrement the count of numbers by 1
        JNZ Loop                ; if count of numbers isNotEqualTo 0 jump to LOOP
                HLT                             ; terminates the execution

```

# Assignment – 3

**Q1 : A set of N data bytes is stored in m/m locations starting from 2501H. The value of N is stored in 2500H. Write a program to store these data bytes from m/m location 2600H if D0 or D7 is 1; otherwise reject the data byte.**

**Sol :**

```
;<Copy numbers either starting or ending with 1 to another location>

JMP Start                           ; starts the execution

Start:  LXI H,2600H             ; to set the location where numbers will be stored after execution
        XCHG                        ; exchange content of H-L reg pair with content of D-E reg pair
        LXI H,2500H             ; taking the count of numbers as input
        MOV B,M                 ; reg B stores the count of numbers

LOOP1:  INX H                   ; taking the input to form a list of numbers
        MOV A,M                 ; move that taken input to accumulator A
        RRC                             ; rotate right accumulator
        JNC LOOP2               ; is D7 isEqualTo 0 jump to LOOP2
        RLC                             ; rotate left accumulator
        XCHG                    ; exchange content of H-L reg pair with content of D-E reg pair
        MOV M,A                 ; move acuumulator content to M [HL]
        INX H                   ; take the next input of numbers
        XCHG                    ; exchange content of H-L reg pair with content of D-E reg pair
        JMP SKIP                ; unconditionally nump to SKIP


LOOP2:  RLC                         ; rotate left accumulator
        RLC                             ; rotate left accumulator
        JNC SKIP                ; if D0 isNotEqualTo 1 jump to SKIP
        RRC                             ; rotate right accumulator  if D0 isEqualTo 1
        XCHG                    ; exchange content of H-L reg pair with content of D-E reg pair
        MOV M,A                 ; move acuumulator content to M [HL]
        INX H                   ; taking the input to form a list of numbers
        XCHG                    ; exchange content of H-L reg pair with content of D-E reg pair

SKIP:   DCR B                   ; decrementing the count of numbers by 1
        JNZ LOOP                ; is count isNotEqualTo 0 jump to LOOP
        HLT                             ; terminates the execution

```

**Q2 : There are N data bytes stored from m/m location 2200H. The value of N is stored in 21FFH. Write an 8085 program to find the sum of integers whose LSB and MSB are 1. Store the result in 2500H and 2501H.**

**Sol :**

```
;<Store those numbers whose sum of LSB and MSB are 1 at certain positions>

JMP Start                           ; starts the execution

Start:  LXI H,21FFH             ; taking the count of numbers as input
        MOV B,M                     ; move the count to reg B
        LXI H,2200H             ; taking the first input of numbers
        MVI C,00H               ; initialising C reg
        MVI A,00H               ; initialising A reg

LOOP:   STA 2600H               ; store accumulator content at address 2600H
        MOV A,M                     ; number taken as input is moved to Accumulator
        ANI 81H                     ; 1000 0001 ANDed
        CPI 81H                     ; check if LSB=1 and MSB=1
        JNZ SKIP1               ; if LSB AND MSB not = 1 then goto SKIP1
        LDA 2600H               ; stores the existing sum
        ADD M                       ; add the new number to sum
        JNC SKIP                ; if no carry found, jump to SKIP
        INR C                       ; increment reg C content
        JMP SKIP                ; unconditionally jump to SKIP

SKIP1:  LDA 2600H               ; load accumulator from address 2600H
        INX H                       ; increment M [HL]
        DCR B                       ; decrement the count by 1
        JNZ LOOP                ; if count isNotEqualTo 0, jump to LOOP
        JMP LAST                ; unconditionally jump to LAST

SKIP:   INX H                   ; increment M [HL]
        DCR B                       ; decrement the count by 1
        JNZ LOOP                ; if count isNotEqualTo 0, jump to LOOP

LAST:   STA 2500H               ; store the count of numbers with SumOfLSB = 1 at address 2500H
        MOV A,C                     ; move the count of numbers having SumOfMSB = 1 to accumulator
        STA 2501H               ; store the count of numbers with SumOfMSB = 1 at address 2501H
        HLT                             ; terminates the execution
```

**Q3 : Write an 8085 program to generate N th fibonacci number using function and store it in 2050H. The value of N (8-bits) is stored in memory 2060H.**

**Sol :**

```
;<Print N-th Fibonacci number using function>

JMP Start                       ; starts the execution

Start:  CALL Func       ; calls the function
                HLT                     ; terminates the execution

Func:   LDA 2060H       ; take count (N) of number as input
                MOV C,A         ; move the count (N) to reg C
                MVI A,00H       ; base condition
                DCR C           ; decrement the count by 1 (N = N-1)
                JZ Skip         ; ifCountEqualsToZero jump to SKIP
                MVI A,01H       ; base condition
                DCR C           ; decrement the count by 1 (N = N-2)
                JZ Skip         ; ifCountEqualsToZero jump to SKIP
        MVI B,00H       ; initialise buffer as 00H

Loop:   MOV D,A         ; move the sum (Fib(N-1)) to reg D from accumulator A
                ADD B           ; calculate the N-th number here (Fib(N) = Fib(N-1) + Fib(N-2))
                MOV B,D         ; move previous number of the series (Fib(N-2)) to reg B
                DCR C           ; decrement the count by 1
                JNZ Loop        ; ifCountNotEqualsToZero, jump to LOOP

Skip:   STA 2050H       ; store the result at 2050H
                RET                     ; return the function
```

**Q4 : Write a program to transfer a block of bytes of size N from location1 to location2 (location2 > location1) when the size of overlap between the two locations is defined by M. The values of N and M are stored in 201EH and 201FH, respectively.**

**Sol :**

```
;<Print N-th Fibonacci number using function>

JMP Start                       ; starts the execution

Start:  LXI H,201EH             ;
        MOV B,M                 ; STORE N AT REG-B
        INR L                   ;
        MOV C,M         ; STORE M AT REG-C
        MOV A,B                 ;
        SUB C                   ; A = (N-M)
        LXI H,2050H             ; HL -> LOCATION1
        ADD L                   ; A -> A+L
        JNC Skip                ;
        INR H                   ;

Skip:   MOV L,A                 ; HL = (HL+A) OR HL->LOCATION2
                MOV C,B                 ;
        MVI B,00H               ;
        DCX B                   ; BC -> N-1
        DAD B                   ;
        XCHG                    ; STORE THE END-ADDRESS OF LOCATION2 IN DE PAIR
        LXI H,2050H             ;
        DAD B                   ; STORE END-ADDRESS OF LOCATION1 IN HL PAIR
        INX B                   ; C = N

LOOP:   MOV A,M                 ;
                XCHG                    ;
        MOV M,A                 ; SWAP CONTENT OF HL WITH DE
        DCX D                   ;
        DCX H                   ;
        XCHG                    ;
        DCR C                   ; DECREMENT N
        JNZ LOOP                ; CONTINUE UNTIL N IS 0

        HLT                             ; terminates the execution
```