11.08.2023

ECC512 Microprocessor and Microcontroller

Assignment -3

CHANDNI JHA | REGISTRATION-690 | ROLL-21030

INDEX

Github-Link: https://github.com/ChandniJha630/CSC512-MIcroprocessor-And-Microcontroller-Lab/tree/main/Assignment%203

PROB.1 FIND THE FACTORIAL OF A GIVEN NUMBER XX, STORED IN 8C00H, AND STORE THE RESULT IN 8C01H.

```
;< CODE>
jmp start
;data
;code
start: nop
LXI H,8C00H
MOV B,M
MVI D,1H
FACTORIAL:   CALL MULTIPLY
             DCR B
             JNZ FACTORIAL
INX H
MOV M,D
HLT

MULTIPLY:    MOV E,B
             MVI A, 00H
             LOOP:        ADD D
                    DCR E
                    JNZ LOOP
             MOV D, A
RET
```

ALGORITHM

The program starts by loading the value from memory into the B register. This value represents the number for which the factorial needs to be calculated.

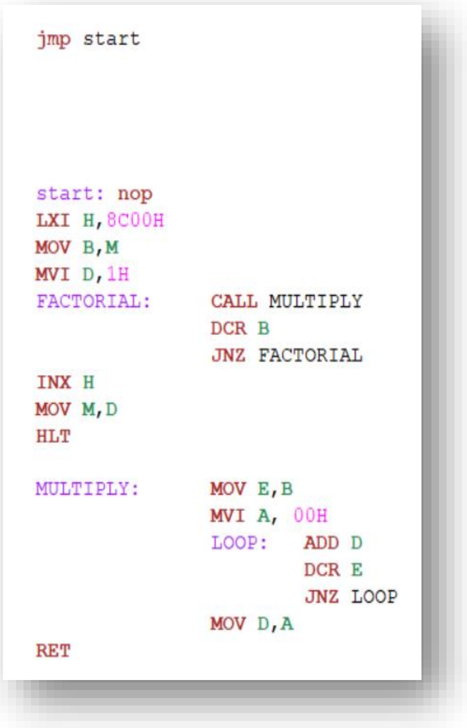
The program then enters a loop labelled as FACTORIAL. It repeatedly calls the MULTIPLY subroutine to calculate the factorial value for the current number stored in the B register.

The MULTIPLY subroutine uses a loop labelled as LOOP to perform multiplication using addition. It initializes an accumulator A to 00H and iterates E times (where E is the value from the B register) to repeatedly add the value in D to the accumulator A. This effectively calculates the product of D and E.

Once the multiplication is complete, the result is stored in the D register.

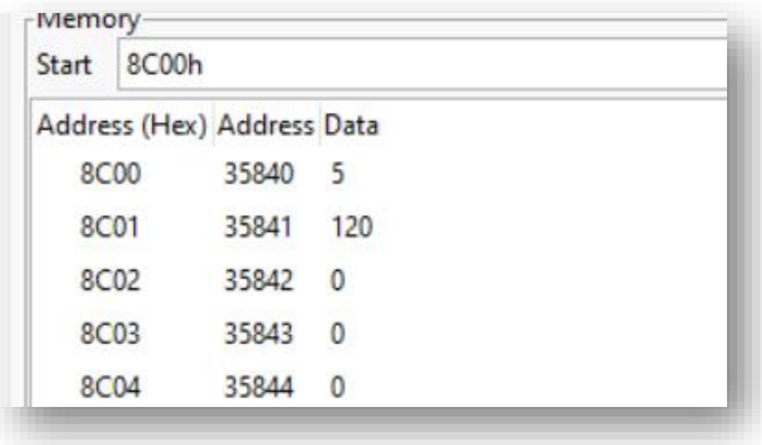
After calculating the factorial, the value in B (representing the current number) is decremented, and the loop continues until B becomes zero.

CODE



```
jmp start




start: nop
LXI H,8C00H
MOV B,M
MVI D,1H
FACTORIAL:      CALL MULTIPLY
                DCR B
                JNZ FACTORIAL
INX H
MOV M,D
HLT

MULTIPLY:       MOV E,B
                MVI A, 00H
                LOOP:   ADD D
                        DCR E
                        JNZ LOOP
                MOV D,A
RET
```

INPUT- OUTPUT



```
Memory
Start  8C00h

Address (Hex) Address Data
    8C00       35840   5
    8C01       35841   120
    8C02       35842   0
    8C03       35843   0
    8C04       35844   0
```

PROB.2 WRITE AN ALP TO FIND THE SQUARE OF A GIVEN NUMBER STORED IN 8550H, AND STORE THE RESULT IN 8551H.

```
;<CODE>
jmp start
;data
;code
start: nop

LXI H,8550H
MOV L,M
MVI H,90H
MOV A,M
STA 8551H
Hlt
```

ALGORITHM

The program then uses the LXI instruction to load the 16-bit memory address 8550H into the HL register pair. This will be used to access data in memory.

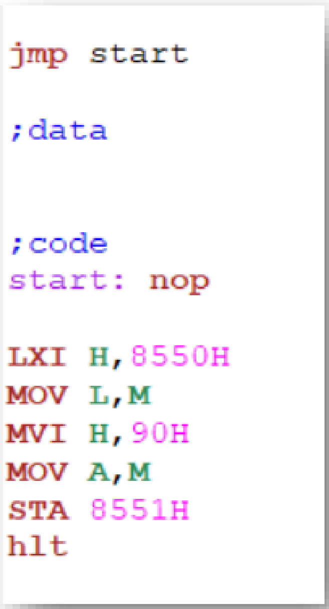
The MOV instruction copies the value stored in the memory location pointed to by the HL register pair into the L register. This effectively loads data from memory into the L register.

The MVI instruction sets an immediate value 90H (i.e, look up table) into the H register. This immediate value is directly loaded into the register.

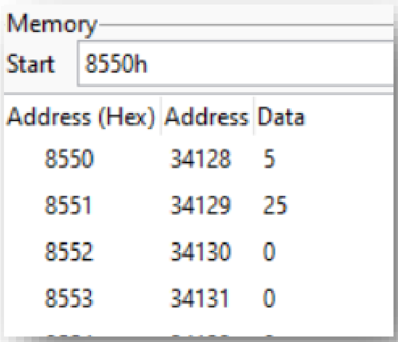
Another MOV instruction copies the value stored in the memory location pointed to by the HL register pair into the A register. This loads another piece of data from memory into the A register.

The STA instruction (Store Accumulator) stores the value currently in the A register into the memory location 8551H.

CODE INPUT / OUTPUT



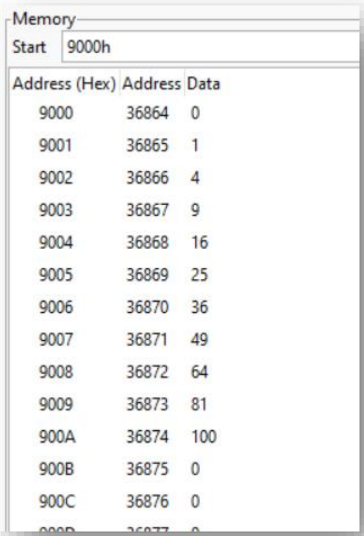
```
jmp start

;data


;code
start: nop

LXI H,8550H
MOV L,M
MVI H,90H
MOV A,M
STA 8551H
hlt
```



Memory
Start 8550h

Address (Hex)	Address	Data
8550	34128	5
8551	34129	25
8552	34130	0
8553	34131	0

LOOKUP TABLE STORED AT 90H



Memory
Start 9000h

Address (Hex)	Address	Data
9000	36864	0
9001	36865	1
9002	36866	4
9003	36867	9
9004	36868	16
9005	36869	25
9006	36870	36
9007	36871	49
9008	36872	64
9009	36873	81
900A	36874	100
900B	36875	0
900C	36876	0

PROB.3 TWO 8 BIT NUMBERS ARE STORED IN MEMORY LOCATIONS 8C00H AND 8C01H. PERFORM DIVISION AND STORE THE QUOTIENT IN 8C02H AND REMAINDER IN 8C03H.

```
;<code>
jmp start
;data
;code
start: nop
LDA 8C01H
MOV D,A
LDA 8C00H

MVI C,0FFH

DIV:    INR C
        SUB D
        JNC DIV
        ADD D
STA 8C03H
MOV A,C
STA 8C02H
Hlt
```

ALGORITHM

The program starts by using the LDA instruction to load the value from memory location 8C01H into the A register. This value in the A register will be used as part of the division process.

The MOV instruction then copies the value in the A register to the D register. This value will be used later in the division process.

Another LDA instruction is used to load the value from memory location 8C00H into the A register. This value will serve as the dividend in the division operation.

The MVI instruction loads the immediate value 0FFH into the C register. This will be the initial value of the counter used in the division process.

The program enters a loop labelled as DIV. Inside this loop:

The INR instruction increments the value in the C register, which acts as a counter for the division operation.

The SUB instruction subtracts the value in the D register (copied from A earlier) from the value in the A register. This is the core of the division process. The carry flag (C flag) is set if a borrow does not occur during the subtraction.

The JNC instruction is a conditional jump that jumps back to the DIV label if the carry flag (C flag) is not set. This effectively repeats the loop until the subtraction can be done without causing a borrow (indicating a successful division step).

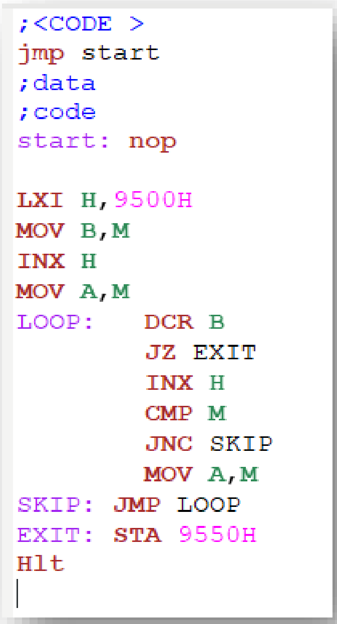
After the loop, the ADD instruction adds the value in the D register back to the A register.

The STA instruction stores the final value in the A register at memory location 8C03H.

After the division loop, the program copies the value in the C register (which contains the quotient of the division) to the A register using the MOV instruction.

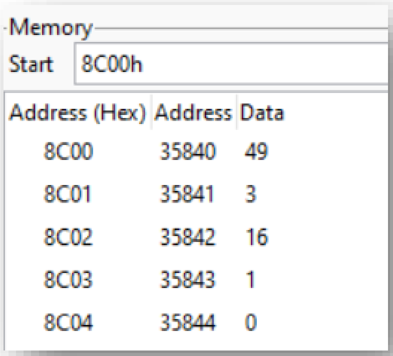
Finally, the STA instruction stores the quotient value in the A register at memory location 8C02H.

CODE:



```
;<CODE >
jmp start
;data
;code
start: nop

LXI H,9500H
MOV B,M
INX H
MOV A,M
LOOP:    DCR B
         JZ EXIT
         INX H
         CMP M
         JNC SKIP
         MOV A,M
SKIP: JMP LOOP
EXIT: STA 9550H
Hlt
```

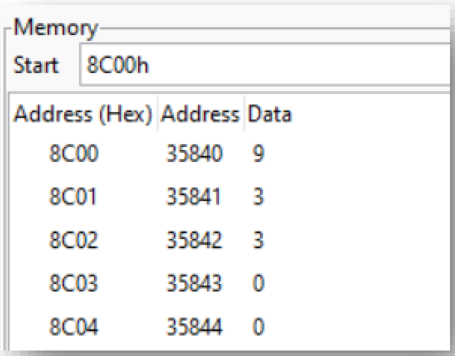
INPUTS AND OUTPUTS



Memory

Start 8C00h

Address (Hex)	Address	Data
8C00	35840	49
8C01	35841	3
8C02	35842	16
8C03	35843	1
8C04	35844	0



Memory

Start 8C00h

Address (Hex)	Address	Data
8C00	35840	9
8C01	35841	3
8C02	35842	3
8C03	35843	0
8C04	35844	0

PROB.4 FIND THE LARGEST NO. FROM A SERIES OF NUMBERS. THE NO. OF DATA IS STORED IN 9500H AND THE DATA IS STORED FROM 9501H. STORE THE RESULT IN 9550H

```
;<CODE >
jmp start
;data
;code
start: nop

LXI H,9500H
MOV B,M
INX H
MOV A,M
LOOP: DCR B
        JZ EXIT
        INX H
        CMP M
        JNC SKIP
        MOV A,M
SKIP: JMP LOOP
EXIT: STA 9550H
Hlt
```

ALGORITHM

The program starts by loading the immediate value 9500H into the HL register pair. This value is likely used to point to a specific memory address where data is stored.

The MOV instruction loads the value at memory location 9500H into the B register.

The INX instruction increments the memory address pointed to by the HL register pair.

The MOV instruction then loads the value at the updated memory location into the A register.

The program enters a loop labeled as LOOP. Inside this loop:

The DCR instruction decrements the value in the B register.

The JZ instruction checks if the result of the decrement operation is zero (if B becomes zero). If B is zero, it jumps to the EXIT label, effectively ending the loop.

The INX instruction increments the memory address pointed to by the HL register pair, effectively moving to the next memory location.

The CMP instruction compares the value at the current memory location (pointed by HL) with the value in the A register.

The JNC instruction checks if there's no carry (if A >= M). If there's no carry, it jumps to the SKIP label, which effectively skips loading the value from memory into the A register.

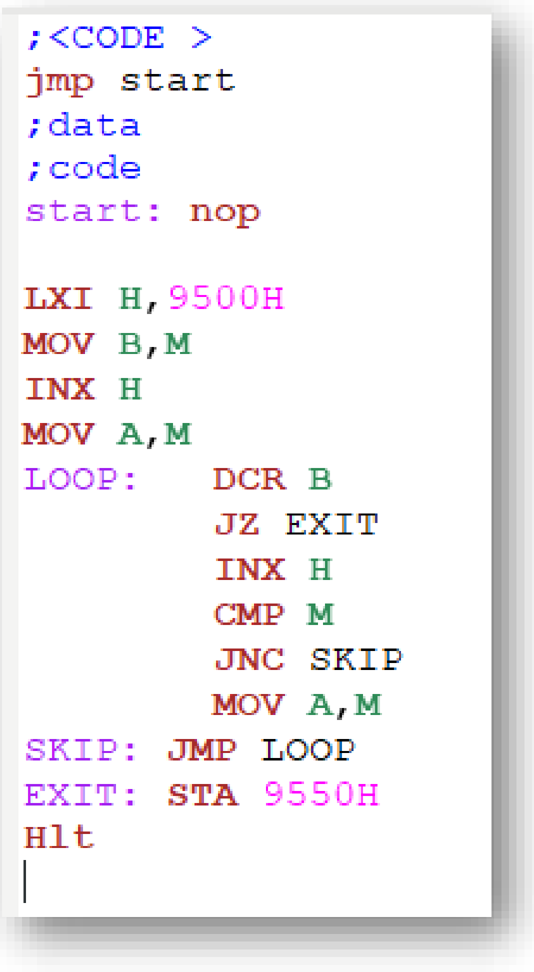
If there's a carry (A < M), the MOV instruction loads the value at the memory location (pointed by HL) into the A register.

The program then unconditionally jumps back to the LOOP label to repeat the process for the next iteration.

When the loop condition is finally met (B becomes zero), the program proceeds to the EXIT label.

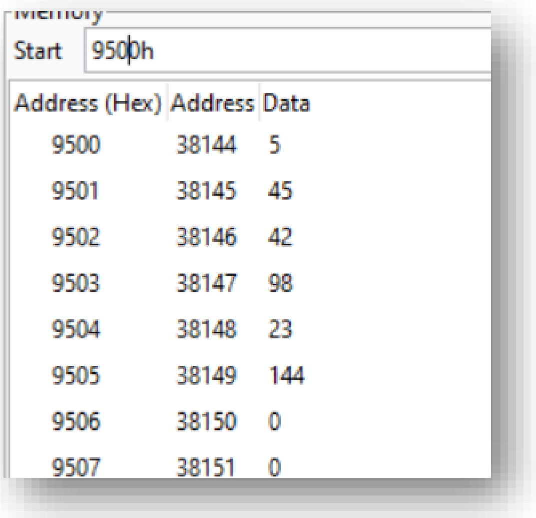
At the EXIT label, the STA instruction stores the value in the A register at memory location 9550H.

CODE



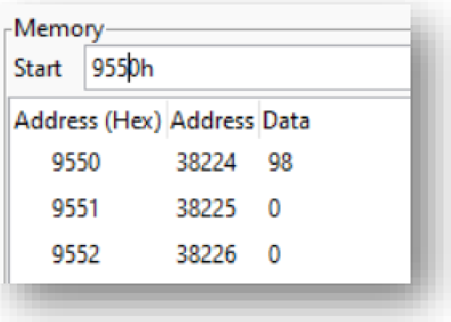
```
;<CODE >
jmp start
;data
;code
start: nop

LXI H,9500H
MOV B,M
INX H
MOV A,M
LOOP:   DCR B
        JZ EXIT
        INX H
        CMP M
        JNC SKIP
        MOV A,M
SKIP: JMP LOOP
EXIT: STA 9550H
Hlt
```

INPUTS AND OUTPUTS



Memory
Start 9500h

Address (Hex)	Address	Data
9500	38144	5
9501	38145	45
9502	38146	42
9503	38147	98
9504	38148	23
9505	38149	144
9506	38150	0
9507	38151	0



Memory
Start 9550h

Address (Hex)	Address	Data
9550	38224	98
9551	38225	0
9552	38226	0