MICROPROCESSOR LAB FILE

Suhaib Ahmad Salmani 2021BITE060

Contents

1 Addition of two 8 bit Numbers

Perform addition of two 8 bit binary Numbers.

```
MVI B, 03H
MVI C, 02H
ADD B
ADD C
hlt
```

- Load value 3 into register B.
- Load value 2 into register C.
- Add the values in registers B and C.
- Halt the program.

The result is in accumulator A, which is the sum of 3 and 2 (i.e., 5).

2 Addition of two 8 bit numbers stored in memory

Perform Addition of two 8-bits numbers. One is stored at memory location 0026 H and the other is taken as input from input port with address 01H. Add these two numbers and store the result and carry at memory location 0027 and 0028 H respectively. Also display the result at output port 05H and carry at port 06H.

```
LXI H, 0026H
MVI C, 00H
MOV B, M
IN 01H
ADD B
JNC NO_CARRY
INR C
NO_CARRY: INX H
STA M
INX H
MOV M, C
OUT 05H
MOV A, C
OUT 06H
hlt
```

- Load the value 0026H into register pair H and L (LXI H, 0026H).
- Set register C to 00H (MVI C, 00H).
- Copy the value from the memory location pointed to by HL to register B (MOV B, M).
- Input a byte from I/O port 01H and store it in the accumulator (IN 01H).
- Add the value in register B to the accumulator (ADD B).
- Check for no carry (JNC NO_CARRY). If there is no carry, jump to the label NO_CARRY.
- Increment the value in register C (INR C).
- Label NO_CARRY: Increment the HL register pair (INX H).
- Store the value in the accumulator at the memory location pointed to by HL (STA M).
- Increment the HL register pair (INX H).
- Copy the value in register C to the memory location pointed to by HL (MOV M, C).
- Output the value in the accumulator to I/O port 05H (OUT 05H).
- Copy the value in register C to the accumulator (MOV A, C).
- Output the value in the accumulator to I/O port 06H (OUT 06H).
- Halt the program (HLT).

3 Multiplication of two 8 bit numbers

Perform the multiplication of two 8 bit numbers stored at memory locations 0035H and 0036H respectively. Store the result at memory location 0037H.

```
LXI H, 0035H
MOV B, M
INX H
MOV C, M
LOOP: ADD B
DCR C
JNZ LOOP
INX H
MOV M, A

HLT
```

- Load the value 06H into accumulator A (MVI A, 06H).

- Load the value 04H into register B (MVI B, 04H).
- Set register C to 00H (MVI C, 00H).
- Load the value 0000H into register pair H and L (MVI H, 0000H).
- Subtract the value in register B from the accumulator (SUB B).
- Check for no carry (JNC SKIP). If there is no carry, jump to the label SKIP.
- Complement the bits in the accumulator (CMA).
- Increment the value in accumulator A (INR A).
- Increment the value in register C (INR C).
- Label SKIP: Copy the value in accumulator A to the memory location pointed to by HL (MOV M, A).
- Increment the HL register pair (INX H).
- Copy the value in register C to the memory location pointed to by HL (MOV M, C).
- Halt the program (HLT).

4 Subtraction of two 8 bit numbers

Perform Subtraction of two 8 bit numbers

```
MVI A, 06H
MVI B, 04H
MVI C, 00H
MVI H, 0000H
SUB B
JNC SKIP
CMA
INR A
INR C
SKIP: MOV M, A
INX H
MOV M, C
HLT
```

- Load the value 06H into accumulator A (MVI A, 06H).
- Load the value 04H into register B (MVI B, 04H).
- Set register C to 00H (MVI C, 00H).

- Load the value 0000H into register pair H and L (MVI H, 0000H).
- Subtract the value in register B from the accumulator (SUB B).
- Check for no carry (JNC SKIP). If there is no carry, jump to the label SKIP.
- Complement the bits in the accumulator (CMA).
- Increment the value in accumulator A (INR A).
- Increment the value in register C (INR C).
- Label SKIP: Copy the value in accumulator A to the memory location pointed to by HL (MOV M, A).
- Increment the HL register pair (INX H).
- Copy the value in register C to the memory location pointed to by HL (MOV M, C).
- Halt the program (HLT).

5 Division of two 8 bit numbers

Perform Division of two 8 bit numbers

```
MVI A, 0AH
MVI B, 03H
MVI C, 00H
LOOP: CMP B
JC SKIP
SUB B
INR C
JMP LOOP
SKIP: OUT 00H
MOV A, C
OUT 01H
HLT
```

- Load the value 0AH into accumulator A (MVI A, 0AH).
- Load the value 03H into register B (MVI B, 03H).
- Set register C to 00H (MVI C, 00H).
- Label LOOP: Compare the value in accumulator A with the value in register B (CMP B).
- Jump to SKIP if there is a carry (JC SKIP).
- Subtract the value in register B from the accumulator (SUB B).

- Increment the value in register C (INR C).
- Jump back to LOOP (JMP LOOP).
- Label SKIP: Output the value in accumulator A to I/O port 00H (OUT 00H).
- Copy the value in register C to accumulator A (MOV A, C).
- Output the value in accumulator A to I/O port 01H (OUT 01H).
- Halt the program (HLT).

6 Largest Number in an Array

Write a program to find the largest number in an array using Microprocessor 8085 instructions.

```
LXI H, 0000H
MVI B, 05H
MOV A, M
LOOP: DCR B
JZ EXIT
INX H
CMP M
JC STORE_BIG
JMP LOOP
STORE_BIG: MOV A, M
JMP LOOP
EXIT: STA 0007H
HLT
```

- Load the value 0000H into register pair H and L (LXI H, 0000H).
- Load the value 05H into register B (MVI B, 05H).
- Copy the value from the memory location pointed to by HL to accumulator A (MOV A, M).
- Label LOOP: Decrement the value in register B (DCR B).
- Check if B is zero (JZ EXIT). If B is zero, jump to the label EXIT.
- Increment the value in register pair H and L (INX H).
- Compare the value in the memory location pointed to by HL with the value in accumulator A (CMP M).
- If there is a carry (JC STORE_BIG), jump to the label STORE_BIG.
- Otherwise, jump back to LOOP (JMP LOOP).

- Label STORE_BIG: Copy the value from the memory location pointed to by HL to accumulator A (MOV A, M).
- Jump back to LOOP (JMP LOOP).
- Label EXIT: Store the value in accumulator A at memory location 0007H (STA 0007H).
- Halt the program (HLT).

7 Smallest Number in an Array

Write a program to find the smallest number in an array using Microprocessor 8085 instructions.

```
LXI H, 0000H
MVI B, 05H
MOV A, M
LOOP: DCR B
JZ EXIT
INX H
CMP M
JNC STORE_SMALL
JMP LOOP
STORE_SMALL: MOV A, M
JMP LOOP
EXIT: STA 0007H
HLT
```

- Load the value 0000H into register pair H and L (LXI H, 0000H).
- Load the value 05H into register B (MVI B, 05H).
- Copy the value from the memory location pointed to by HL to accumulator A (MOV A, M).
- Label LOOP: Decrement the value in register B (DCR B).
- Check if B is zero (JZ EXIT). If B is zero, jump to the label EXIT.
- Increment the value in register pair H and L (INX H).
- Compare the value in the memory location pointed to by HL with the value in accumulator A (CMP M).
- If there is no carry (JNC STORE_SMALL), jump to the label STORE_SMALL.
- Otherwise, jump back to LOOP (JMP LOOP).
- Label STORE_SMALL: Copy the value from the memory location pointed to by HL to accumulator A (MOV A, M).

- Jump back to LOOP (JMP LOOP).
- Label EXIT: Store the value in accumulator A at memory location 0007H (STA 0007H).
- Halt the program (HLT).

8 Sort array in ascending order

Write a program to sort an array in ascending order using Microprocessor 8085 instructions.

```
LXI H, 0026H
MVI C, 00H
MOV B, M
IN 01H
ADD B
JNC NO_CARRY
INR C
NO_CARRY: INX H
STA M
INX H
MOV M, C
OUT 05H
MOV A, C
OUT 06H
hlt
```

- Load the value 0026H into register pair H and L (LXI H, 0026H).
- Set register C to 00H (MVI C, 00H).
- Copy the value from the memory location pointed to by HL to register B (MOV B, M).
- Input a byte from I/O port 01H and store it in the accumulator (IN 01H).
- Add the value in register B to the accumulator (ADD B).
- Check for no carry (JNC NO_CARRY). If there is no carry, jump to the label NO_CARRY.
- Increment the value in register C (INR C).
- Label NO_CARRY: Increment the HL register pair (INX H).
- Store the value in the accumulator at the memory location pointed to by HL (STA M).
- Increment the HL register pair (INX H).
- Copy the value in register C to the memory location pointed to by HL (MOV M, C).
- Output the value in the accumulator to I/O port 05H (OUT 05H).

- Copy the value in register C to accumulator A (MOV A, C).
- Output the value in accumulator A to I/O port 06H (OUT 06H).
- Halt the program (HLT).

9 1's Complement

Write a program to do 1's complement of Contents of Accumulator .

```
MVI A, 00H
CMA
hlt
```

- Load the value 00H into accumulator A (MVI A, 00H).
- Complement (invert) all bits in accumulator A (CMA).
- Halt the program (HLT).

10 2's Complement

Write a program to do 2's complement of Contents of Accumulator .

```
MVI A, 08H
CMA
INR A
hlt
```

- Load the value 08H into accumulator A (MVI A, 08H).
- Complement (invert) all bits in accumulator A (CMA).
- Increment the value in accumulator A by 1 (INR A).
- Halt the program (HLT).

11 Sum of elements of Array

add the first N natural numbers and store the result in memory at location X

```
LXI H, 0000H
MVI B, 04H
MVI A, 00H
```

```
loop: ADD M
INX H
DCR B
JNZ loop
STA 000AH
HLT
```

- Load the value 0000H into register pair H and L (LXI H, 0000H).
- Load the value 04H into register B (MVI B, 04H).
- Set accumulator A to 00H (MVI A, 00H).
- Label loop: Add the value from the memory location pointed to by HL to the accumulator (ADD M).
- Increment the value in register pair H and L (INX H).
- Decrement the value in register B (DCR B).
- Check if B is not zero (JNZ loop). If B is not zero, jump back to the label loop.
- Store the value in accumulator A at memory location 000AH (STA 000AH).
- Halt the program (HLT).

12 Find Factorial

Program to find the factorial of a number

```
LXI H,0050H
MOV B,M
MVI D,01H
Factorial: CALL Multiply
DCR B
JNZ Factorial
INX H
MOV M,D
HLT
Multiply: MOV E,B
MVI A,00H
MULTIPLY_LOOP: ADD D
DCR E
JNZ MULTIPLY_LOOP
MOV D,A
RET
```

- Load the value 0050H into register pair H and L (LXI H, 0050H).
- Copy the value from the memory location pointed to by HL to register B (MOV B, M).
- Load the value 01H into register D (MVI D, 01H).
- Label Factorial: Call a subroutine named Multiply (CALL Multiply).
- Decrement the value in register B (DCR B).
- Check if B is not zero (JNZ Factorial). If B is not zero, jump back to the label Factorial.
- Increment the value in register pair H and L (INX H).
- Copy the value in register D to the memory location pointed to by HL (MOV M, D).
- Halt the program (HLT).
- Subroutine Multiply: Copy the value in register B to register E (MOV E, B).
- Set accumulator A to 00H (MVI A, 00H).
- Label MULTIPLY_LOOP: Add the value in register D to the accumulator (ADD D).
- Decrement the value in register E (DCR E).
- Check if E is not zero (JNZ MULTIPLY_LOOP). If E is not zero, jump back to the label MULTIPLY_LOOP.
- Copy the value in accumulator A to register D (MOV D, A).
- Return from the subroutine (RET).

13 Program to add two BCD numbers

```
LXI H, 0050H
MOV B, M
MVI D, 01H
FACTORIAL:
CALL MULTIPLY
DCR B
JNZ FACTORIAL
INX H
MOV M, D
HLT
MULTIPLY:
MOV E, B
MVI A, 00H
MULTIPLY_LOOP:
```

```
ADD D
DCR E
JNZ MULTIPLY_LOOP
MOV D, A
RET
```

14 Program to find square root of a number

```
MVI A, 01H
MOV B, A
MOV C, A
LDA 2000H
LOOP:
    DIV B
    ADD B
    DCR C
    JZ DONE
    JMP LOOP
DONE:
STA 2001H
HLT
```

15 Program to compare 2 hex numerals for equality

```
MVI A, 01H
MOV B, A
LDA 2000H
MOV C, A
LDA 2001H
CMP C
JZ EQUAL
MOV A, 00H
JMP DONE
EQUAL:
MOV A, 01H
DONE:
STA 2002H
HLT
```

16 Program to search a number in an array of n numbers

```
MVI C, 00H
LDA 2000H
MOV B, A
LDA 2001H
LOOP:
    CMP M
    JZ FOUND
    INX H
    DCR B
    JZ NOT_FOUND
    JMP LOOP
FOUND:
MOV A, C
STA 2002H
HLT
NOT_FOUND:
MVI A, FFH
STA 2002H
HLT
```