

MICROCONTROLLER AND MICROPROCESSOR LAB

EXPERIMENT 3

AIM: To understand the operation of the overflow flag during addition, and subtraction operations in 8051.

SOFTWARE USED: Keil uVision5

Question-1: Write an assembly language program to add two signed 8-bit numbers and observe the overflow flag.

Code:

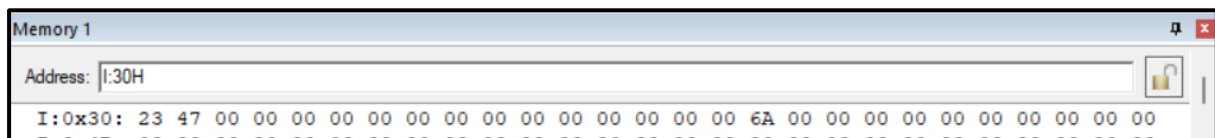
```
ORG 0000H
    MOV A,30H
    MOV B,31H
    ADD A, B
    MOV 40H, A
    END
```

Algorithm:

1. Set the memory location ORG 0000H.
2. Load the hexadecimal value 30H into register A.
3. Load the hexadecimal value 31H into register B.
4. Add the content of register B to register A.
5. Store the result of the addition into memory location 40H.
6. End the program.

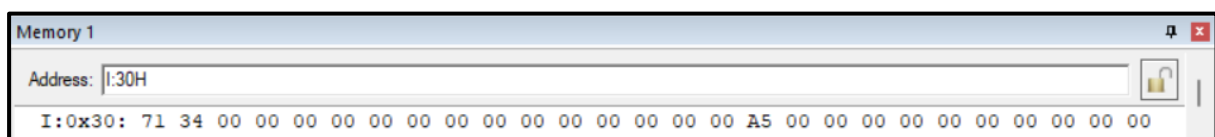
Result:

Case 1: Both numbers are positive.



psw	0x00
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

Case 2: One number is positive and the other is negative. (Overflow Flag is 0)



psw	0x04
p	0
f1	0
ov	1
rs	0
f0	0
ac	0
cy	0

Case 3: One number is positive and the other is negative. (Overflow Flag is 1)

Memory 1	
Address:	I:30H
I:0x30:	AB B3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 5E 00 00 00 00 00 00 00 00 00
I:0x4B:	00 00

psw	0x85
p	1
f1	0
ov	1
rs	0
f0	0
ac	0
cy	1

Case 4: Both numbers are negative.

Memory 1	
Address:	I:30H
I:0x30:	FF FE 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FD 00 00 00 00 00 00 00 00 00
I:0x4B:	00 00

psw	0xc1
p	1
f1	0
ov	0
rs	0
f0	0
ac	1
cy	1

Conclusion:

The assembly language program adds two signed 8-bit numbers, observing the overflow flag in different scenarios. Cases include adding positive numbers (no overflow expected), positive and negative numbers (overflow flag may remain 0 or set to 1), and negative numbers (similar to positive numbers case).