Final Worksheet

Student Name:Rajiv Paul **UID:20BCS1812**

Branch: CSE **Section/Group:607A**

Semester: 4th **Subject Name:MPI Lab**

1) **Aim/Overview of the practical:**

Write a program to subtract two 8-bit numbers with or without borrow where first number is at 2500 memory address and second number is at 2501 memory address and store the result into 2502 and borrow into 2503 memory address.

Apparatus/Simulator used: 8085 simulator

Algorithm:

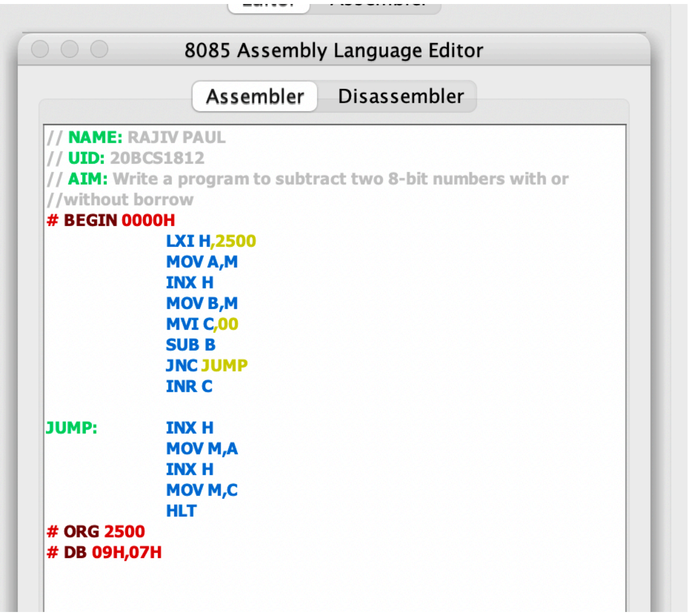
1. LXI H,5000 loaded H-L pair with address 5000H
2. MOV A,M moved the 1st operand from memory to reg. A.
3. INX H incremented H-L pair to point to next memory location.
4. MOV B,M moved the second operand from memory to reg. B.
5. MVI C,00 initialised reg. C with 00H.
6. SUB B subtracted B from A.
7. JNC JUMP jumps to the address of INX H if there is borrow.
8. INR C incremented reg. C
9. INX H increments H-L pair
10. MOV M,C moves borrow from reg. C to memory
11. HLT end of the execution.

Steps for experiment/practical/Code:

```
# BEGIN 0000H

          LXI H,2500

          MOV A,M

          INX H

          MOV B,M

          MVI C,00

          SUB B

          JNC JUMP

          INR C


JUMP:         INX H

          MOV M,A

          INX H

          MOV M,C

          HLT

# ORG 2500

# DB 09H,07H
```

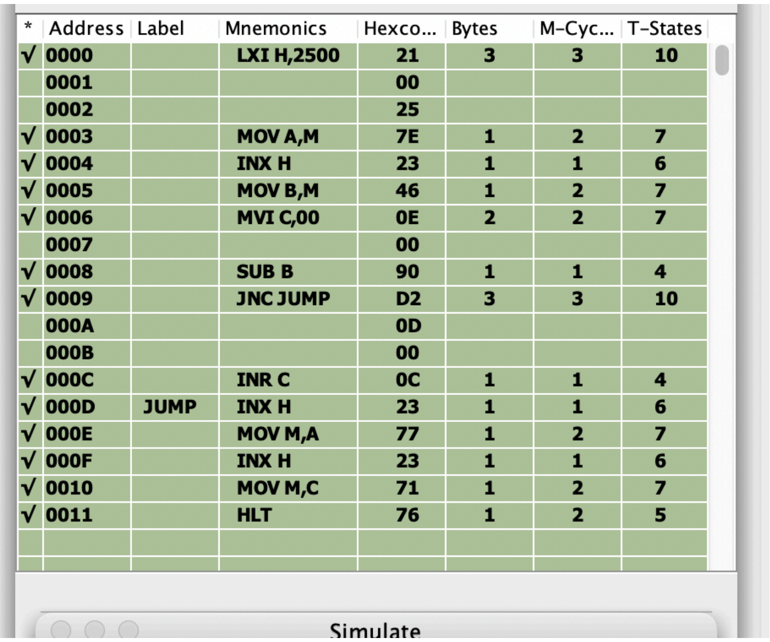
Simulation:

1. CODE IN EDITOR WINDOW:



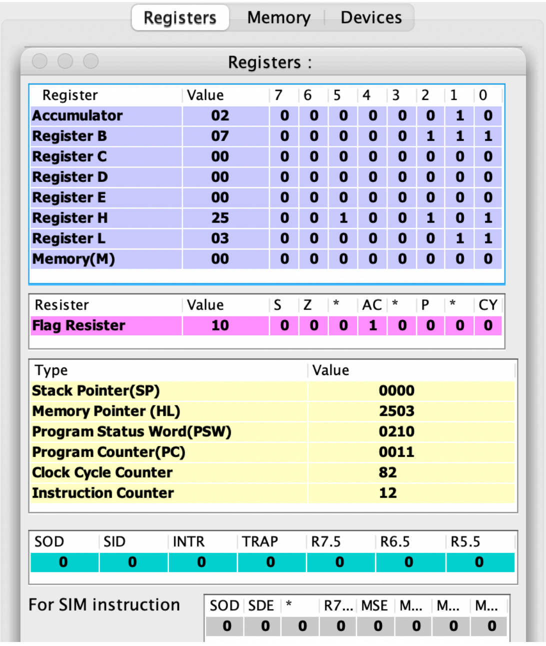
```
// NAME: RAJIV PAUL
// UID: 20BCS1812
// AIM: Write a program to subtract two 8-bit numbers with or
//without borrow
# BEGIN 0000H
                LXI H,2500
                MOV A,M
                INX H
                MOV B,M
                MVI C,00
                SUB B
                JNC JUMP
                INR C

JUMP:           INX H
                MOV M,A
                INX H
                MOV M,C
                HLT
# ORG 2500
# DB 09H,07H
```

2. ASSEMBLER WINDOW:



*	Address	Label	Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0000		LXI H,2500	21	3	3	10
	0001			00			
	0002			25			
√	0003		MOV A,M	7E	1	2	7
√	0004		INX H	23	1	1	6
√	0005		MOV B,M	46	1	2	7
√	0006		MVI C,00	0E	2	2	7
	0007			00			
√	0008		SUB B	90	1	1	4
√	0009		JNC JUMP	D2	3	3	10
	000A			0D			
	000B			00			
√	000C		INR C	0C	1	1	4
√	000D	JUMP	INX H	23	1	1	6
√	000E		MOV M,A	77	1	2	7
√	000F		INX H	23	1	1	6
√	0010		MOV M,C	71	1	2	7
√	0011		HLT	76	1	2	5

3. REGISTERS:



Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	02	0	0	0	0	0	0	1	0
Register B	07	0	0	0	0	0	1	1	1
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	25	0	0	1	0	0	1	0	1
Register L	03	0	0	0	0	0	0	1	1
Memory(M)	00	0	0	0	0	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	10	0	0	0	1	0	0	0	0

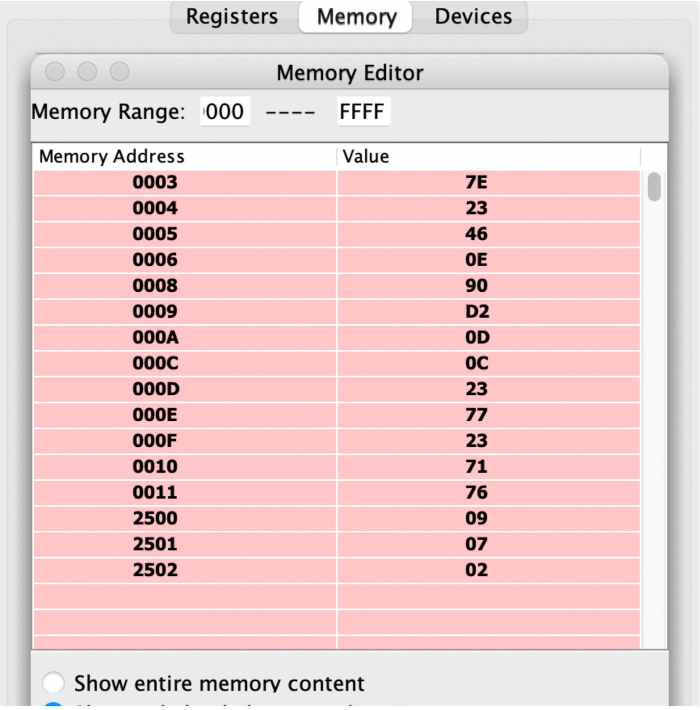
Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	2503
Program Status Word(PSW)	0210
Program Counter(PC)	0011
Clock Cycle Counter	82
Instruction Counter	12

SOD	SID	INTR	TRAP	R7.5	R6.5	R5.5
0	0	0	0	0	0	0

For SIM instruction

SOD	SDE	*	R7...	MSE	M...	M...	M...
0	0	0	0	0	0	0	0

4. MEMORY:



Memory Range: 000 ---- FFFF

Memory Address	Value
0003	7E
0004	23
0005	46
0006	0E
0008	90
0009	D2
000A	0D
000C	0C
000D	23
000E	77
000F	23
0010	71
0011	76
2500	09
2501	07
2502	02

○ Show entire memory content

RESULT

BEFORE EXECUTION:

2500H: 09
2501H: 07

AFTER EXECUTION:

2502H : 02
2503H : 00(no borrow)