Experiment 1.3

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

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

Semester: 4th **Date of Performance: 01/03/2022**

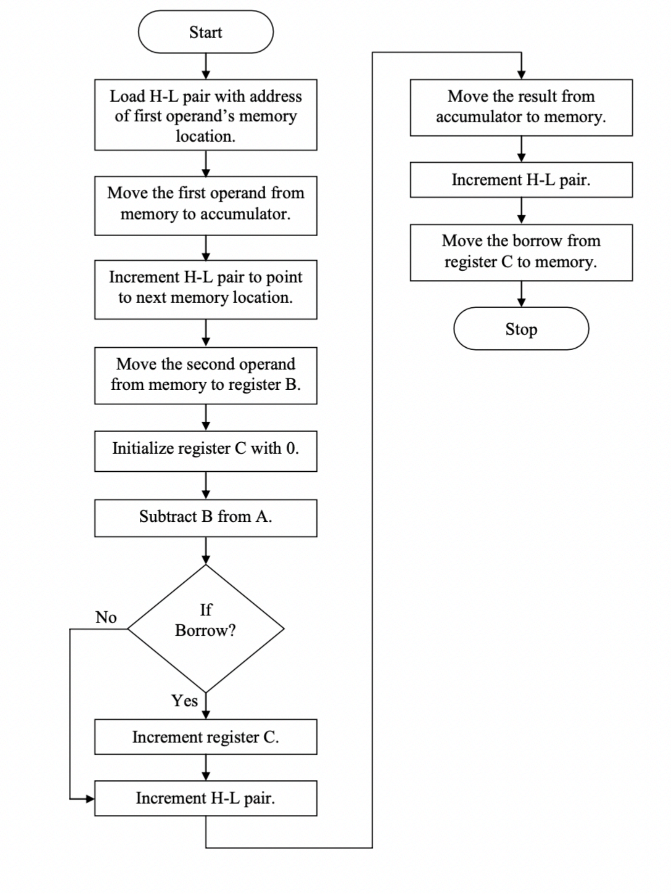
Subject Name:MPI Lab **Subject Code: 22E-20CSP-253**

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

a) Subtraction of two 8 bit numbers along with considering borrow.

Apparatus/Simulator used: 8085 simulator

Flowchart:


Start
Load H-L pair with address of first operand's memory location.
Move the first operand from memory to accumulator.
Increment H-L pair to point to next memory location.
Move the second operand from memory to register B.
Initialize register C with 0.
Subtract B from A.
If Borrow?
No
Yes
Increment register C.
Increment H-L pair.
Move the result from accumulator to memory.
Increment H-L pair.
Move the borrow from register C to memory.
Stop

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

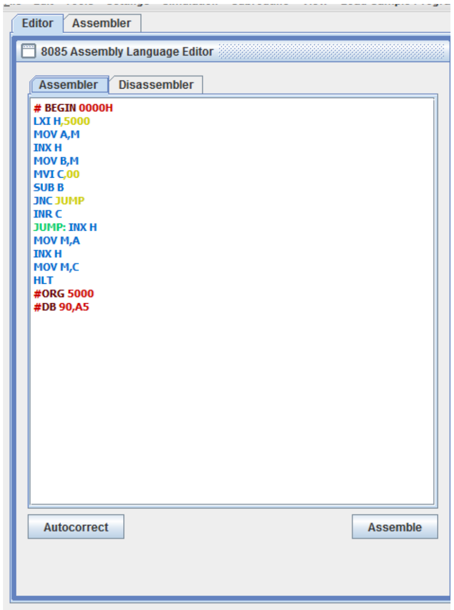
11. HLT end of the execution.

Steps for experiment/practical/Code:

```
# BEGIN 0000H

LXI H,5000

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 5000

#DB 90,A5
```

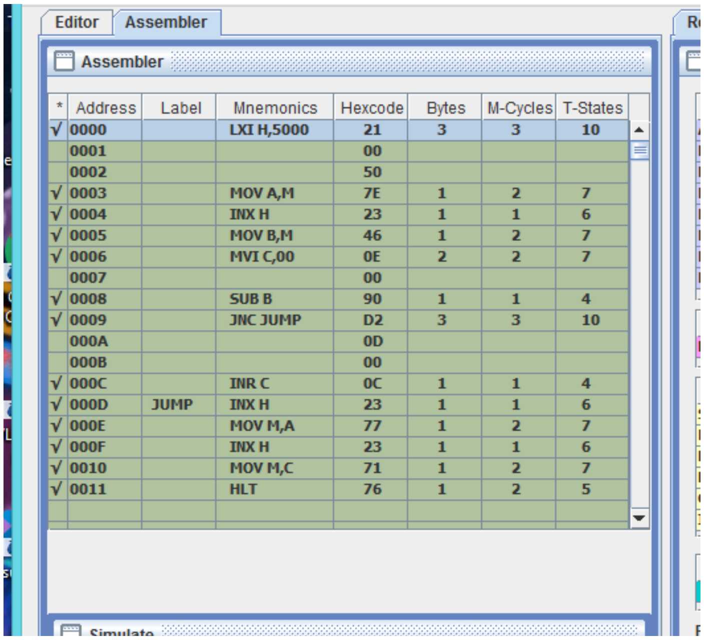
Simulation:

1. CODE IN EDITOR WINDOW:



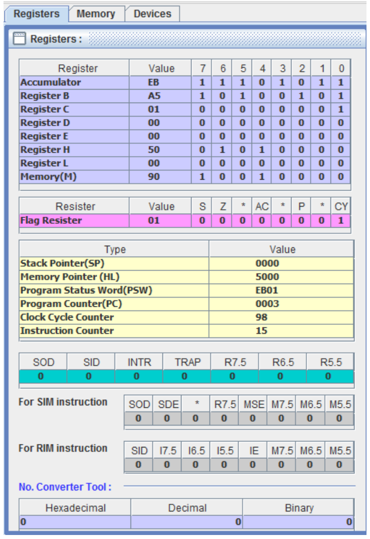
```
# BEGIN 0000H
LXI H,5000
MOV A,M
INX H
MOV B,M
MVI C,00
SUB B
JNC JUMP
INR C
JUMP: INX H
MOV M,A
INX H
MOV M,C
HLT
#ORG 5000
#DB 90,A5
```

2. ASSEMBLER WINDOW:



*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0000		LXI H,5000	21	3	3	10
	0001			00			
	0002			50			
√	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:



Register	Value	7	6	5	4	3	2	1	0
Accumulator	EB	1	1	1	0	1	0	1	1
Register B	A5	1	0	1	0	0	1	0	1
Register C	01	0	0	0	0	0	0	0	1
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	50	0	1	0	1	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	90	1	0	0	1	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	01	0	0	0	0	0	0	0	1

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	5000
Program Status Word(PSW)	EB01
Program Counter(PC)	0003
Clock Cycle Counter	98
Instruction Counter	15

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

For SIM instruction

SOD	SDE	*	R7.5	MSE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

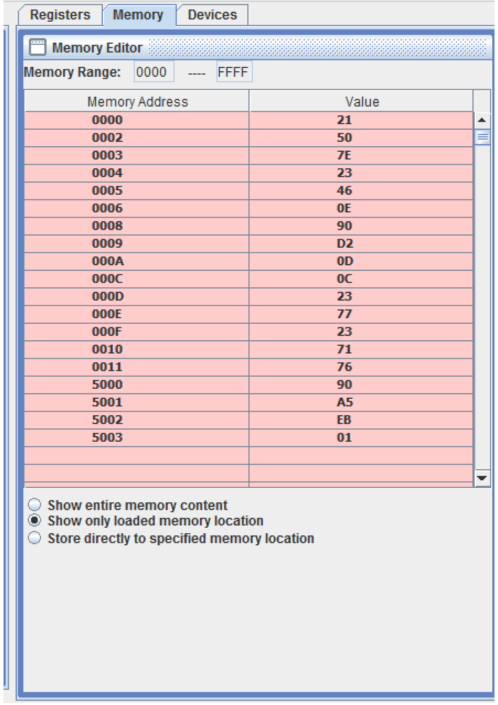
For RIM instruction

SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

No. Converter Tool :

Hexadecimal	Decimal	Binary
0	0	0

4. MEMORY:



Memory Range: 0000 ---- FFFF

Memory Address	Value
0000	21
0002	50
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
5000	90
5001	A5
5002	EB
5003	01

○ Show entire memory content
● Show only loaded memory location
○ Store directly to specified memory location

RESULT

BEFORE EXECUTION:

5000H: 90
5001H: A5

AFTER EXECUTION:

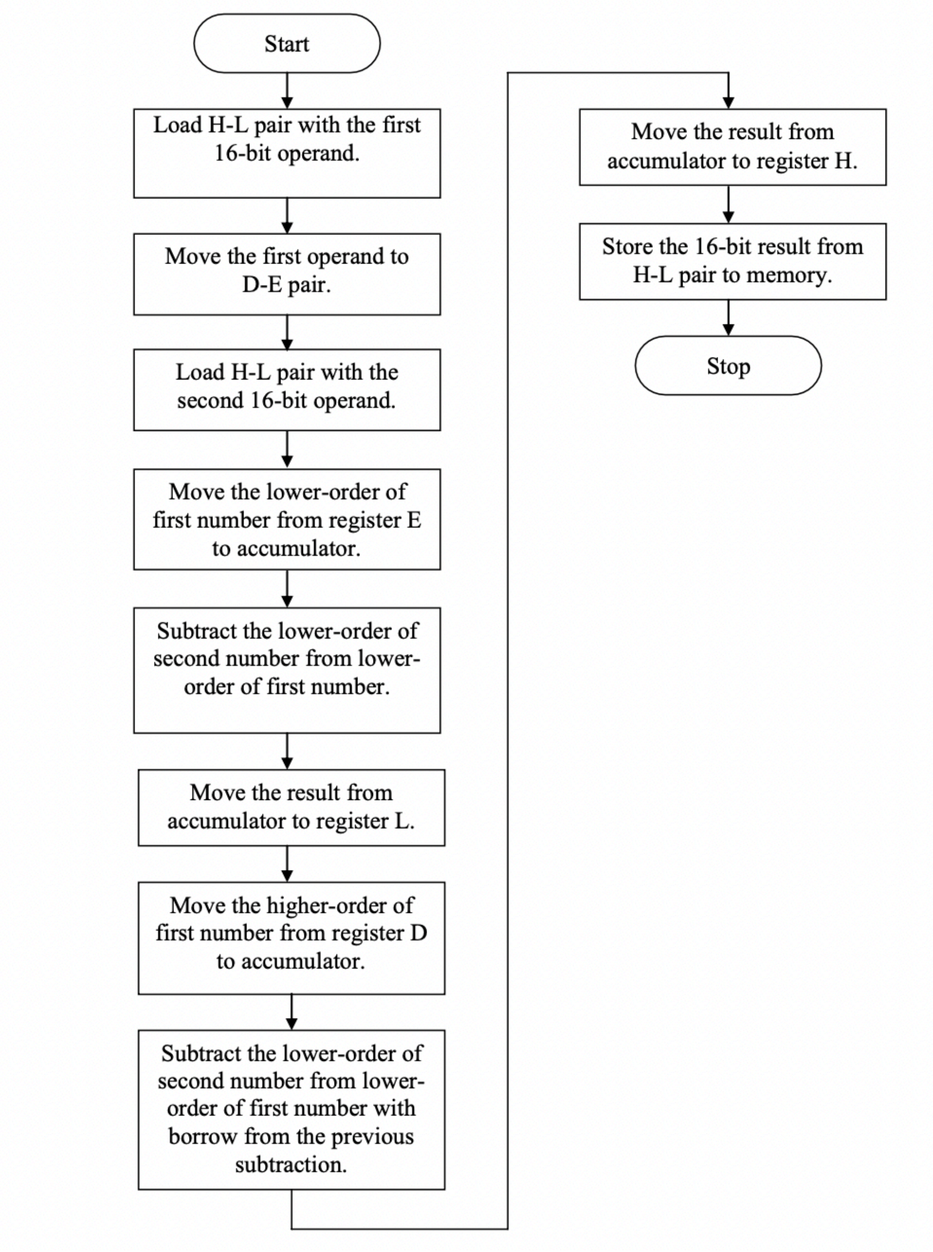
5002H : EB
5003H : 01

Aim/Overview of the practical:

b) Subtraction of two 8 bit numbers along with considering borrow.

Apparatus/Simulator used: 8085 simulator

Flowchart:


Start
Load H-L pair with the first 16-bit operand.
Move the first operand to D-E pair.
Load H-L pair with the second 16-bit operand.
Move the lower-order of first number from register E to accumulator.
Subtract the lower-order of second number from lower-order of first number.
Move the result from accumulator to register L.
Move the higher-order of first number from register D to accumulator.
Subtract the lower-order of second number from lower-order of first number with borrow from the previous subtraction.
Move the result from accumulator to register H.
Store the 16-bit result from H-L pair to memory.
Stop

Algorithm:

1. LHLD 5000 loaded H-L pair with 1st operand from 5000H.

2. XCHG exchanged H-L pair with D-E pair.

3. LHLD 5002 loaded H-L pair with 2nd operand from 5000H.

4. MOV A,E moved the lower order of 1st number from reg. E to A.

5. SUB L subtracted lower order of 2nd number from lower order of 1st number.

6. MOV A,D moved the result from reg. D to A.

7. SBB H subtracted the higher-order of 2nd number from higher-order of 1st number with borrow from the previous subtraction.

8. MOV H,A moved the result from reg. A to H.

9. SHLD 5004 stored the 16-bit result from H-L pair to memory.

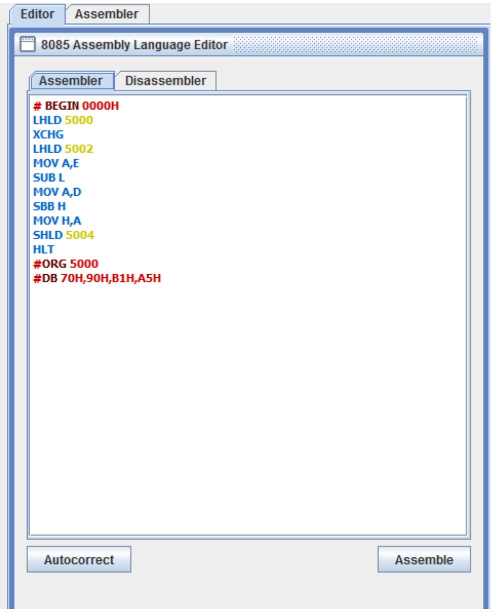
10. HLT end of the execution.

Steps for experiment/practical/Code:

```
# BEGIN 0000H

LHLD 5000

XCHG

LHID 5002

MOV A,E

SUBL

MOV A,D

SBBH

MOV H,A

SHLD 5004

HLT

#ORG 5000

#DB 70H,90H,BIH,A5H
```

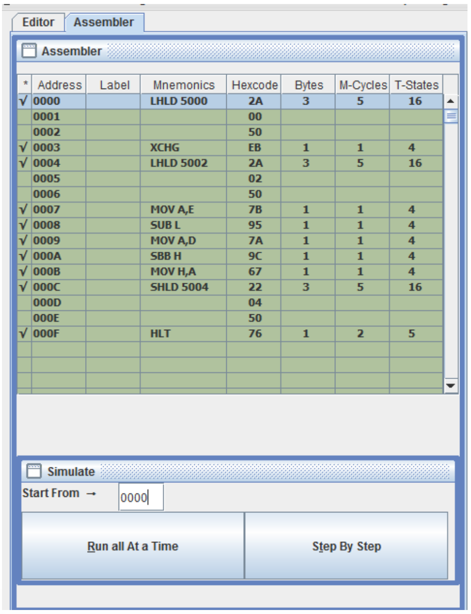
Simulation:

1. CODE IN EDITOR WINDOW:



```
# BEGIN 0000H
LHLD 5000
XCHG
LHLD 5002
MOV A,E
SUB L
MOV A,D
SBB H
MOV H,A
SHLD 5004
HLT
#ORG 5000
#DB 70H,90H,B1H,A5H
```

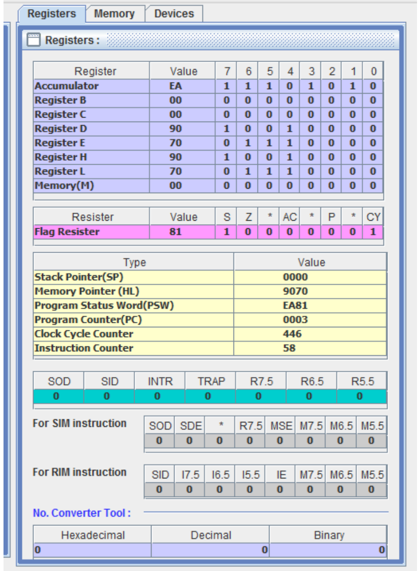
2. ASSEMBLER WINDOW:



*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0000		LHLD 5000	2A	3	5	16
	0001			00			
	0002			50			
√	0003		XCHG	EB	1	1	4
√	0004		LHLD 5002	2A	3	5	16
	0005			02			
	0006			50			
√	0007		MOV A,E	7B	1	1	4
√	0008		SUB L	95	1	1	4
√	0009		MOV A,D	7A	1	1	4
√	000A		SBB H	9C	1	1	4
√	000B		MOV H,A	67	1	1	4
√	000C		SHLD 5004	22	3	5	16
	000D			04			
	000E			50			
√	000F		HLT	76	1	2	5

Start From → 0000

3. REGISTERS:



Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	EA	1	1	1	0	1	0	1	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	90	1	0	0	1	0	0	0	0
Register E	70	0	1	1	1	0	0	0	0
Register H	90	1	0	0	1	0	0	0	0
Register L	70	0	1	1	1	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	81	1	0	0	0	0	0	0	1

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	9070
Program Status Word(PSW)	EA81
Program Counter(PC)	0003
Clock Cycle Counter	446
Instruction Counter	58

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

For SIM instruction

SOD	SDE	*	R7.5	MSE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

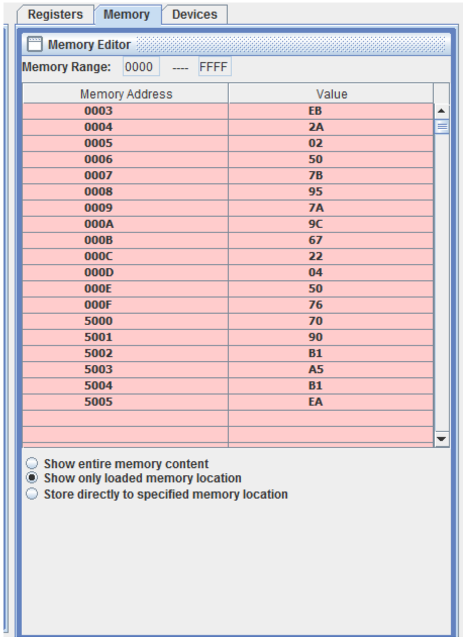
For RIM instruction

SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

No. Converter Tool :

Hexadecimal	Decimal	Binary
0	0	0

4. MEMORY:



Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
0003	EB
0004	2A
0005	02
0006	50
0007	7B
0008	95
0009	7A
000A	9C
000B	67
000C	22
000D	04
000E	50
000F	76
5000	70
5001	90
5002	B1
5003	A5
5004	B1
5005	EA

○ Show entire memory content
● Show only loaded memory location
○ Store directly to specified memory location

RESULT

BEFORE EXECUTION:

```
5000H: 70
5001H: 90
5002H: B1
5003H: A5
```

AFTER EXECUTION:

```
5004H :  B1
5005H :  EA
```

Learning outcomes (What I have learnt):

1.Learnt about 8085 simulator

2. Learnt how to perform 8 bits and 16 bits subtraction.

3.

4.

5.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			