TRIBHUVAN UNIVERSITY

Faculty of Management

Shanker Dev Campus

MICROPROCESSOR AND COMPUTER ARCHITECTURE

A LAB REPORT

Submitted To

Department of BIM
Shanker Dev Campus
Putalisadak, Kathmandu

In partial fulfillment of the requirements for the Third Semester of the Bachelor of Information Management Degree

Submitted By

ASHESH NEUPANE
(TU Exam Roll No: 16295/23)

Under Supervision: **Mr. Narayan GC**

March 2025

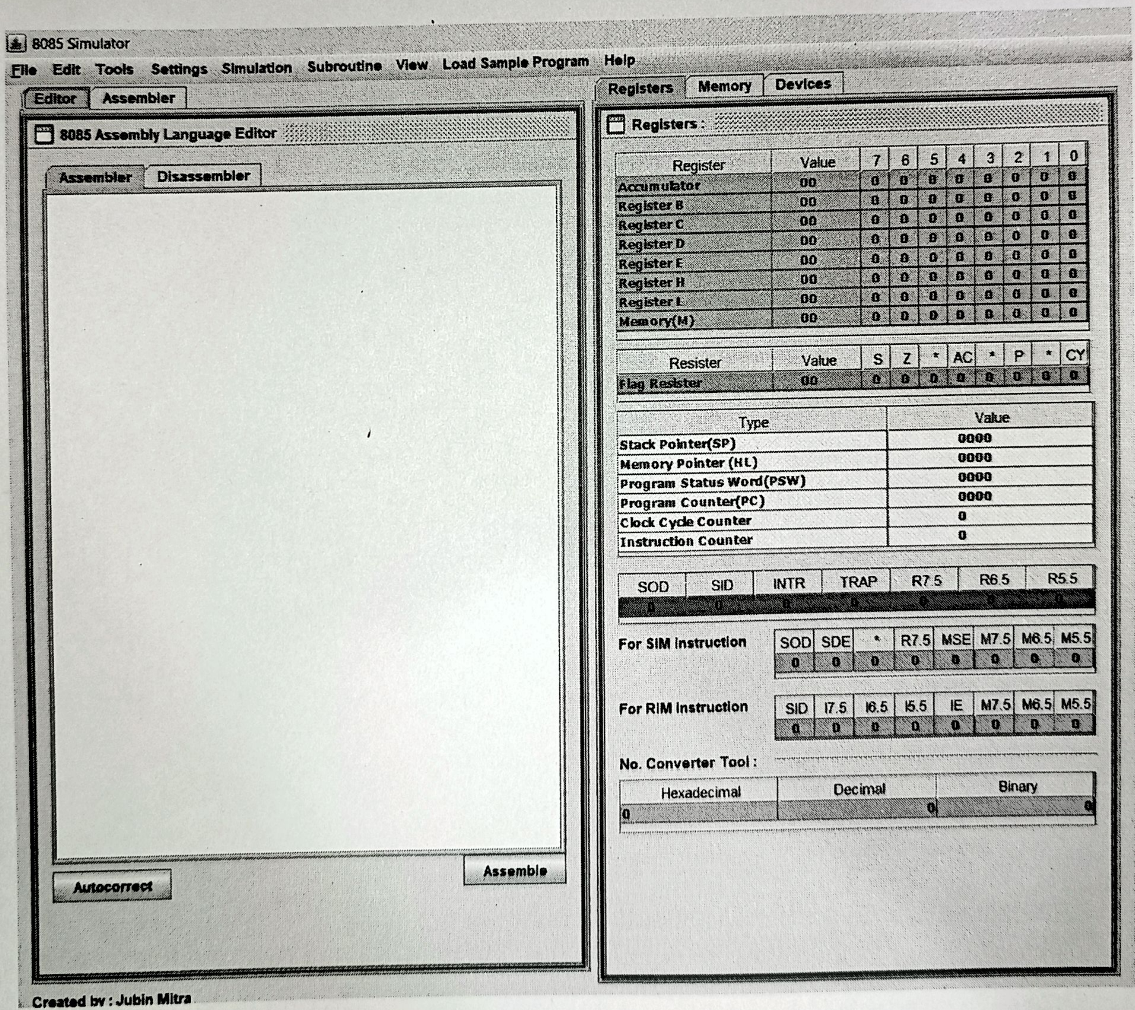
Supervisor's Signature:

Table of Contents

8085 Simulator

8085 simulator IDE is a powerful application that supplies 8085 microprocessor users and educators with user-friendly graphical development environment for windows with integrated simulator, 8085 basic compiler, assembler, disassembler and debugger for the Intel 8085 8 bit microprocessor.

The 8085 assembler is a low level programming language used in the Intel 8085 microprocessor. It allows users to write code that can be directly executed by 8085 microprocessor. It is a powerful tool for programming at a low level and can be used to control hardware and perform specific task.



write an assembly language program that performs the addition of 8 bit numbers.

Steps:

- Store the first number in the memory location C050.
- Store the second number in the memory location C051.
- Result is stored in C052.

Program:

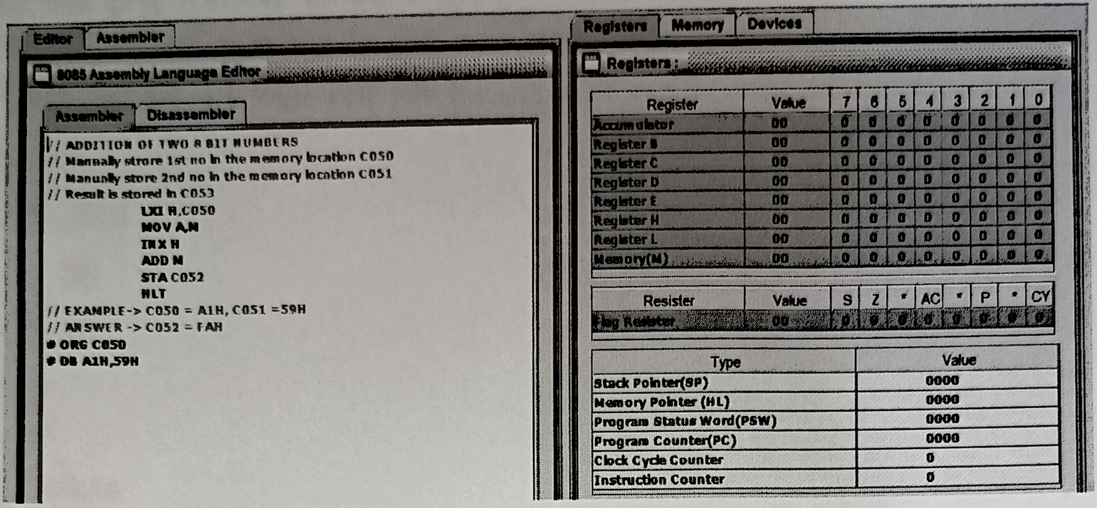
```
LXI H, C050
MOV A,M
INX H
ADD M
STA C052
HLT
#ORG C050
#DB A1H, 59H
```

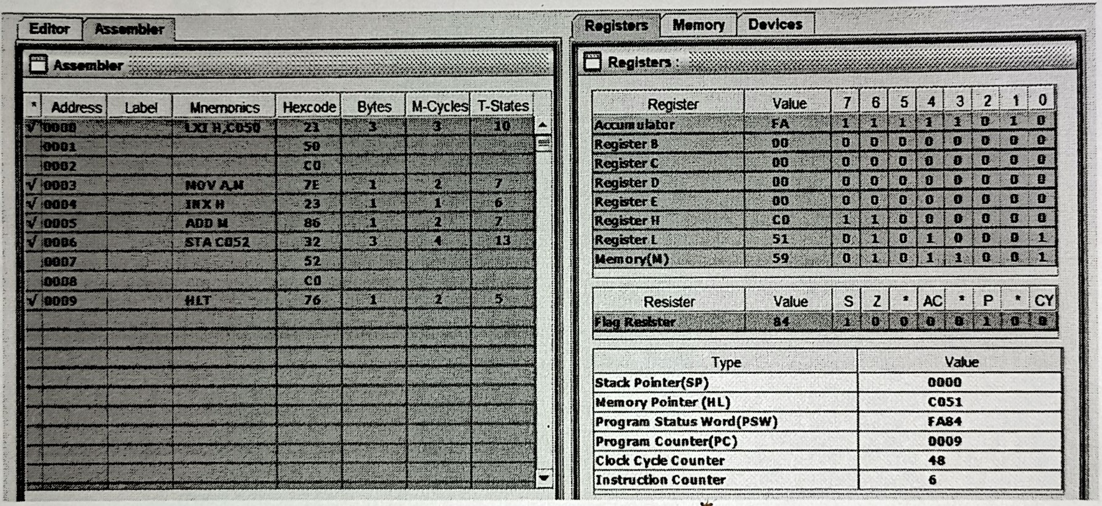
Name: ASHESH NEUPANE

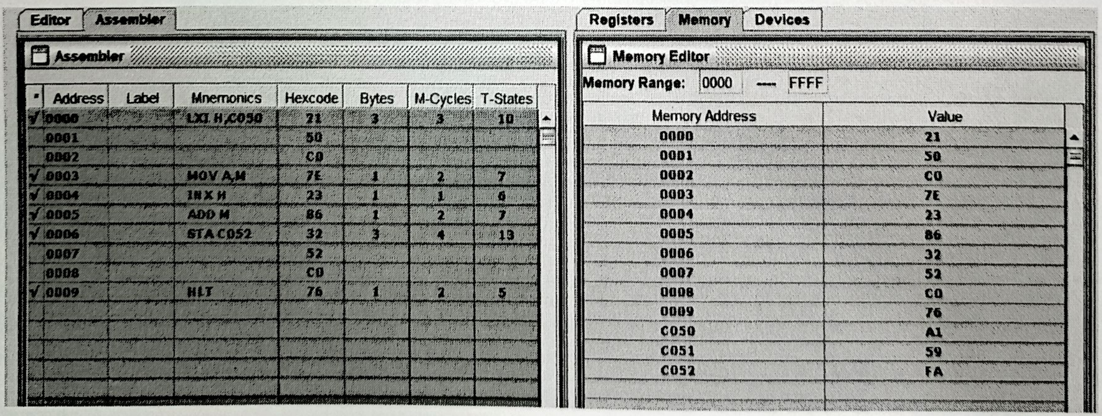
TU Exam Roll No: 16295/23

Laboratory Experiment:

Addition of 8 bit numbers.







Write an assembly language that perform the addition of 8 bit numbers having sum 16 bit.

Steps:

- Store the first number in the memory location C050.
- Store the second number in the memory location C051.
- Result is stored in C052 and C053.

Programs:

```
LXI H, C050
MVI C,00
MOV A,M
INX H
ADD M
JNC AHEAD
INR C
AHEAD: STA C053
       MOV A,C
       STA C052
       HLT
#ORG C050
#DB A1H, 79H
```

Name: ASHESH NEUPANE

TU Exam Roll No: 16295/23

Laboratory Experiment:

Addition of 8 bit numbers having 16 bit sum.

Editor | Assembler

8085 Assembly Language Editor

Assembler | Disassembler

```
// ADDITION OF TWO 8 BIT NUMBERS HAVING 16 BIT SUM
// Manually strore 1st no in the memory location C050
// Manually store 2nd no in the memory location C051
// Result is stored in C052, & C053
            LXI H,C050
            MVI C,00
            MOV A,M
            INX H
            ADD M
            JNC AHEAD
            INR C

AHEAD:      STA C053
            MOV A,C
            STA C052
            HLT
// EXAMPLE -> C050 = A1H, C051 = 69H
// ANSWER-> C052 = 01H, C053 = 1AH
# ORG C050
# DB A1H,79H
```

Registers | Memory | Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	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	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

Editor | Assembler

Assembler

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0000		LXI H,C050	21	3	3	10
	0001			50			
	0002			C0			
√	0003		MVI C,00	0E	2	2	7
	0004			00			
√	0005		MOV A,M	7E	1	2	7
√	0006		INX H	23	1	1	6
√	0007		ADD M	86	1	2	7
√	0008		JNC AHEAD	D2	3	3	10
	0009			0C			
	000A			00			
√	000B		INR C	0C	1	1	4
√	000C	AHEAD	STA C053	32	3	4	13
	000D			53			
	000E			C0			
√	000F		MOV A,C	79	1	1	4
√	0010		STA C052	32	3	4	13
	0011			52			
	0012			C0			

Registers | Memory | Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	01	0	0	0	0	0	0	0	1
Register B	00	0	0	0	0	0	0	0	0
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	C0	1	1	0	0	0	0	0	0
Register L	51	0	1	0	1	0	0	0	1
Memory(M)	79	0	1	1	1	1	0	0	1

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)	C051
Program Status Word(PSW)	0101
Program Counter(PC)	0013
Clock Cycle Counter	83
Instruction Counter	11

Editor | Assembler

Assembler

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0003		MVI C,00	0E	2	2	7
	0004			00			
√	0005		MOV A,M	7E	1	2	7
√	0006		INX H	23	1	1	6
√	0007		ADD M	86	1	2	7
√	0008		JNC AHEAD	D2	3	3	10
	0009			0C			
	000A			00			
√	000B		INR C	0C	1	1	4
√	000C	AHEAD	STA C053	32	3	4	13
	000D			53			
	000E			C0			
√	000F		MOV A,C	79	1	1	4
√	0010		STA C052	32	3	4	13
	0011			52			
	0012			C0			
√	0013		HLT	76	1	2	5

Registers | Memory | Devices

Memory Editor

Memory Range: 0000 — FFFF

Memory Address	Value
0001	50
0002	C0
0003	0E
0005	7E
0006	23
0007	86
0008	D2
0009	0C
000B	0C
000C	32
000D	53
000E	C0
000F	79
0010	32
0011	52
0012	C0
0013	76
C050	A1
C051	79
C052	01
C053	1A

Write an assembly language program that perform addition of two 16 bit numbers having sum 16 bits or more.

Steps:

- Store first 16 bit number in the memory location C050 and C051 in reverse order.
- Store second 16 bit number in the memory location C052 and C053 in reverse order.
- Result is stored in C054 and C055 in reverse order.

Program:

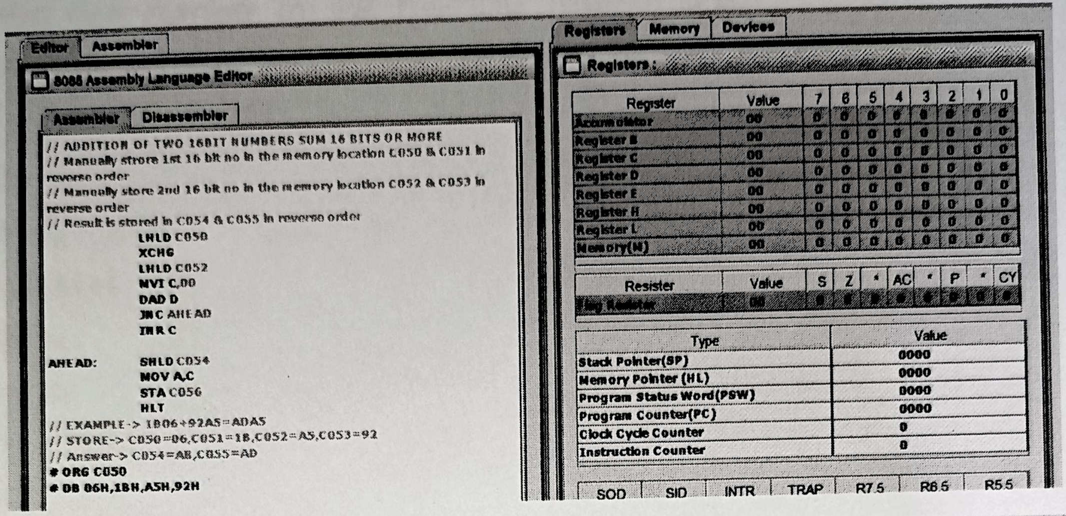
```
LHLD C050
XCHG
LHLD C052
MVI C, 00
DAD D
JNC AHEAD
INR C
AHEAD: SHLD C054
       MOV A,C
       STA C056
       HLT
#ORG C050
#DB 06H, 1BH, A5H, 92H
```

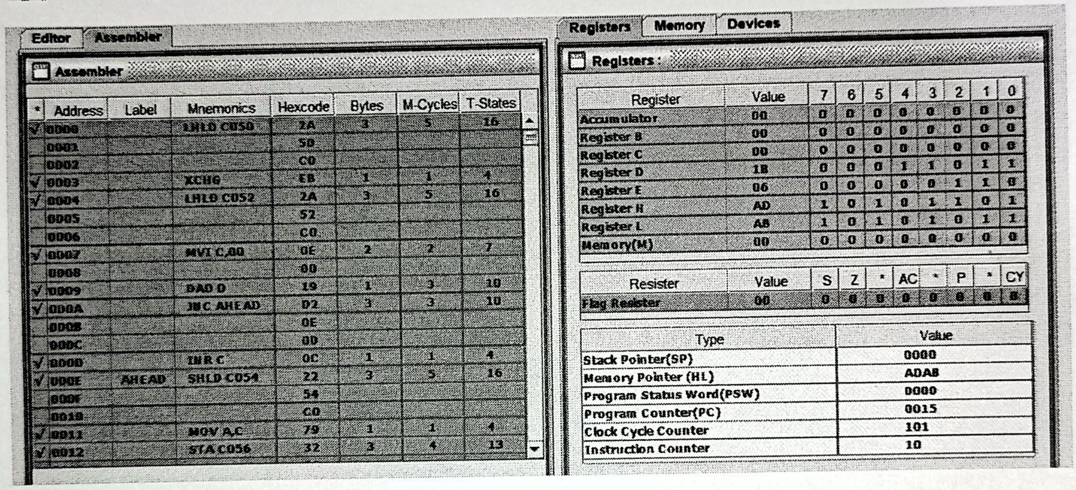
Name: ASHESH NEUPANE

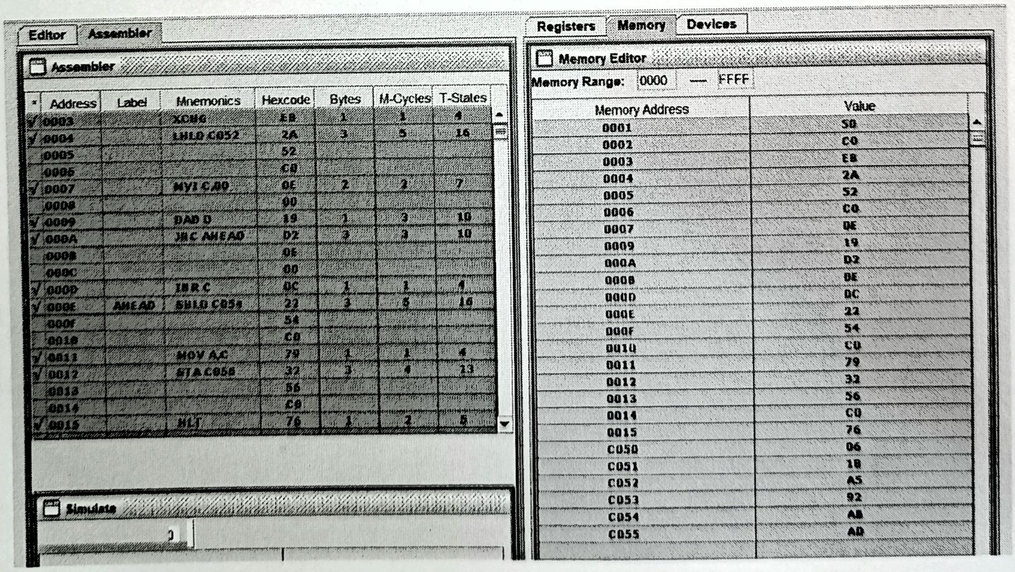
TU Exam Roll No: 16295/23

Laboratory Experiment:

Addition of 16 bit numbers having sum 16 bits or more.







Write an assembly language program that performs subtraction of 8 bit numbers.

Steps:

- Store the first number in the memory location C050.
- Store the second number in the memory location C051.
- Result is stored in C052.

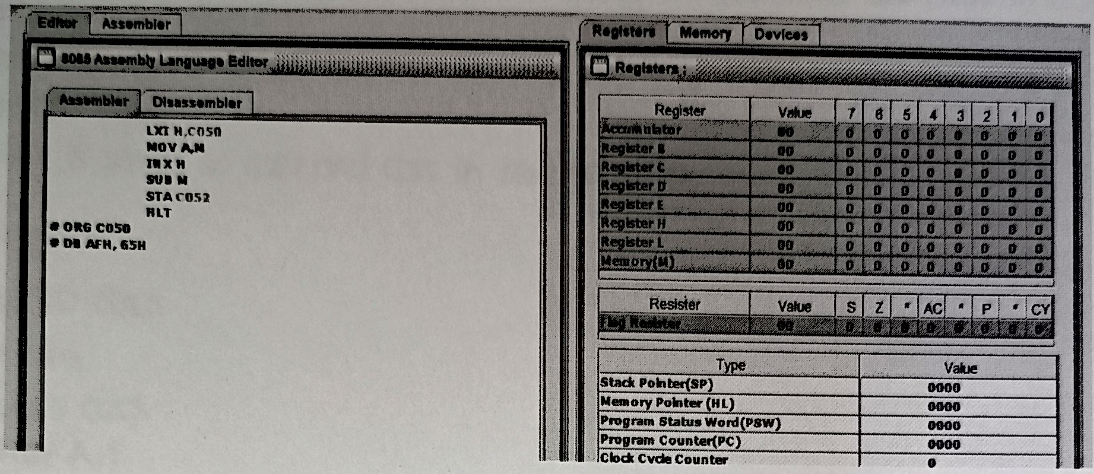
Program:

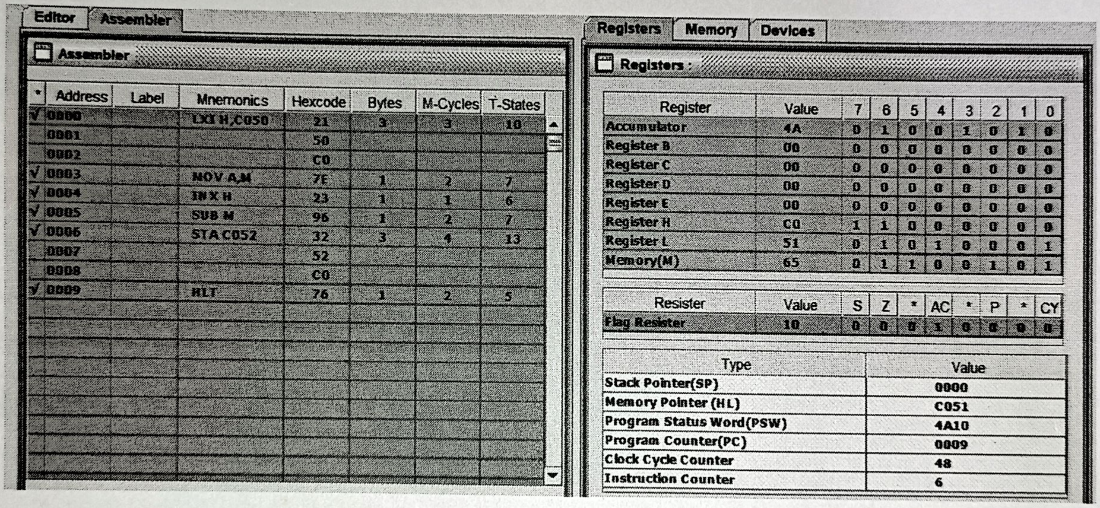
```
LXI H, C050
MOV A, M
INX H
SUB M
STA C052
HLT
#ORG C050
# DB AFH, 65H
```

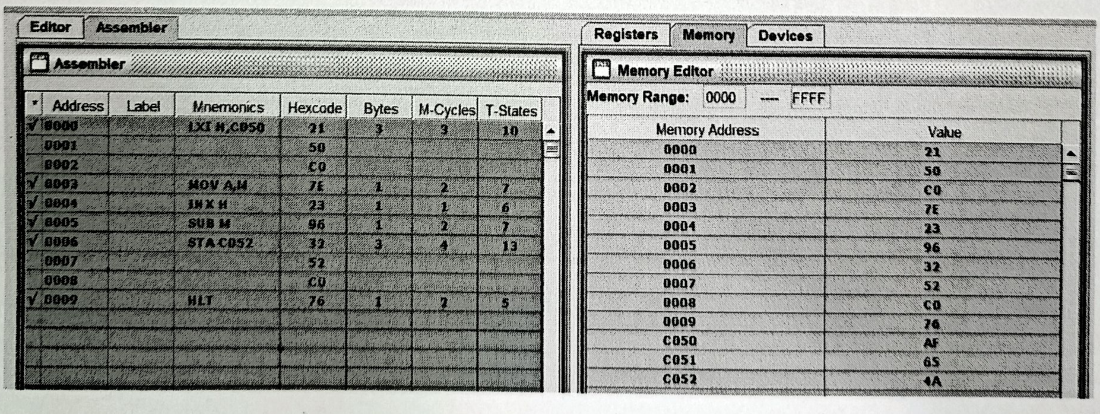
Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

Laboratory Experiment:

Subtraction of 8 bit numbers.







Write an assembly language program that perform subtraction of 16 bit numbers.

Steps:

- Store the first 16 bit number into the memory location C050 and C051 in reverse order.
- Store the second 16 bit number into the memory location C052 and C053 in reverse order.
- Result is stored in C054 and C055 in reverse order.

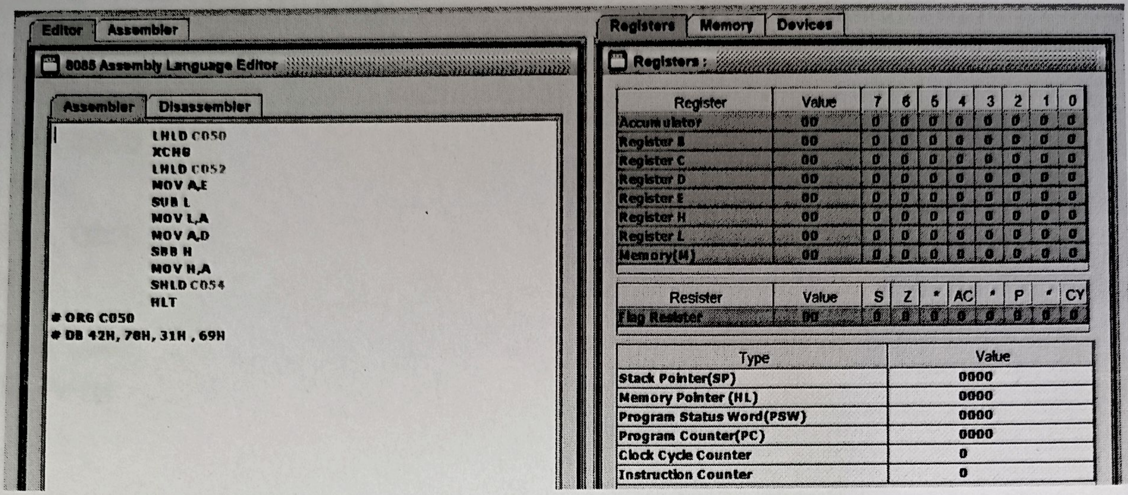
Program:

```
LHLD C050
XCHG
LHLD C052
MOV A,E
SUB L
MOV L,A
MOV A,D
SBB H
MOV H,A
SHLD C054
HLT
#ORG C050
#DB 42H,78H,31H,69H
```

Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

Laboratory Experiment:

Subtraction of 16 bit numbers.



Editor | Assembler — 8085 Assembly Language Editor — Assembler | Disassembler

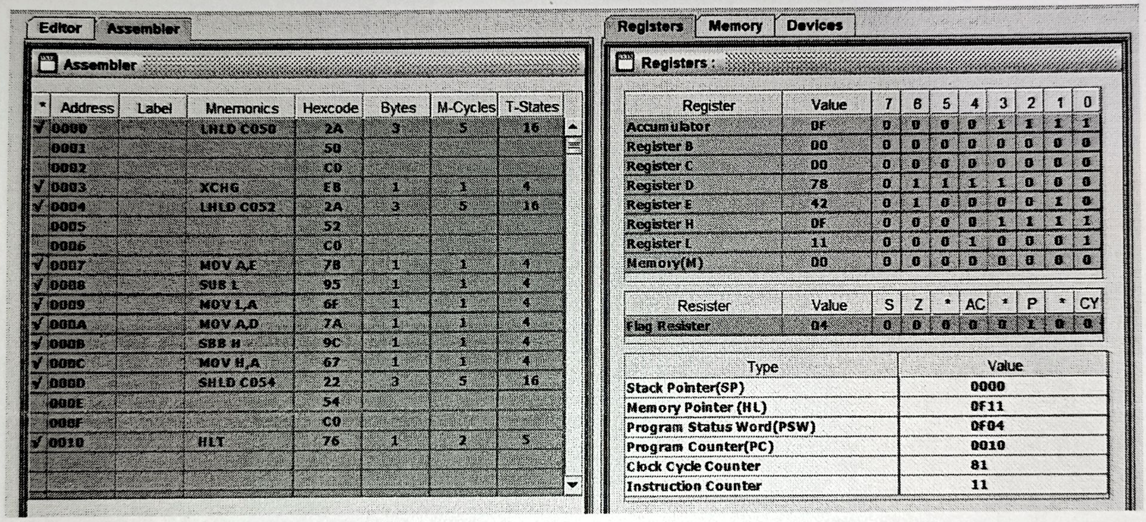
```
        LHLD C050
        XCHG
        LHLD C052
        MOV A,E
        SUB L
        MOV L,A
        MOV A,D
        SBB H
        MOV H,A
        SHLD C054
        HLT
# ORG C050
# DB 42H, 78H, 31H , 69H
```

Registers | Memory | Devices — Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	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	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0



Editor | Assembler — Assembler

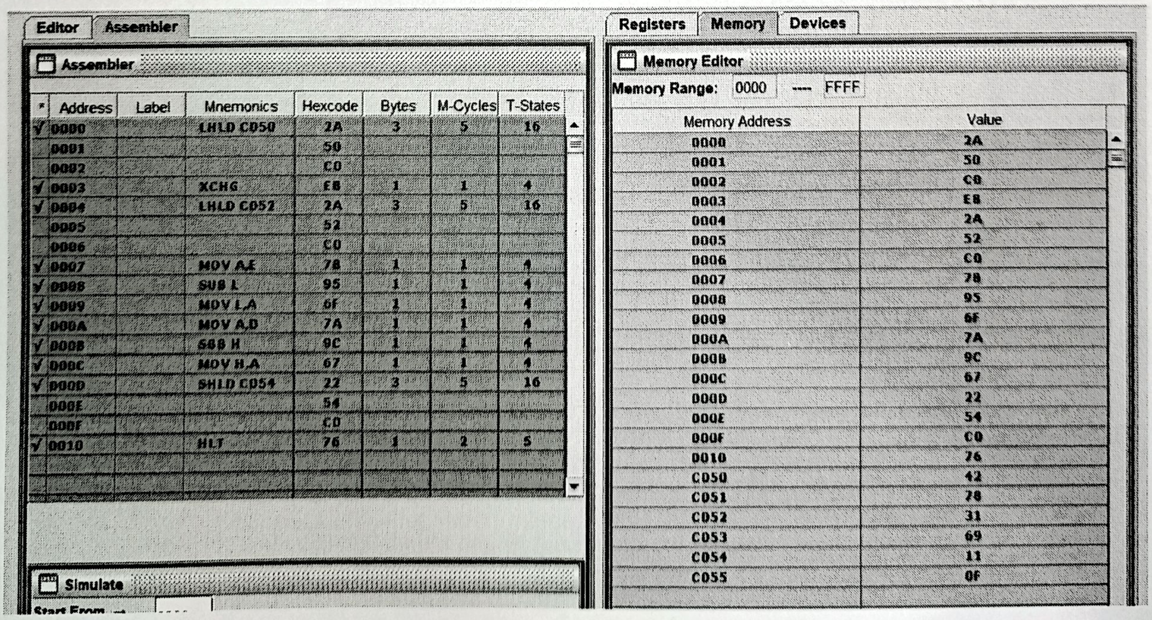
*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0000		LHLD C050	2A	3	5	16
	0001			50			
	0002			C0			
√	0003		XCHG	EB	1	1	4
√	0004		LHLD C052	2A	3	5	16
	0005			52			
	0006			C0			
√	0007		MOV A,E	7B	1	1	4
√	0008		SUB L	95	1	1	4
√	0009		MOV L,A	6F	1	1	4
√	000A		MOV A,D	7A	1	1	4
√	000B		SBB H	9C	1	1	4
√	000C		MOV H,A	67	1	1	4
√	000D		SHLD C054	22	3	5	16
	000E			54			
	000F			C0			
√	0010		HLT	76	1	2	5

Registers | Memory | Devices — Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	0F	0	0	0	0	1	1	1	1
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	78	0	1	1	1	1	0	0	0
Register E	42	0	1	0	0	0	0	1	0
Register H	0F	0	0	0	0	1	1	1	1
Register L	11	0	0	0	1	0	0	0	1
Memory(M)	00	0	0	0	0	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	04	0	0	0	0	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0F11
Program Status Word(PSW)	0F04
Program Counter(PC)	0010
Clock Cycle Counter	81
Instruction Counter	11



Editor | Assembler — Assembler

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0000		LHLD C050	2A	3	5	16
	0001			50			
	0002			C0			
√	0003		XCHG	EB	1	1	4
√	0004		LHLD C052	2A	3	5	16
	0005			52			
	0006			C0			
√	0007		MOV A,E	7B	1	1	4
√	0008		SUB L	95	1	1	4
√	0009		MOV L,A	6F	1	1	4
√	000A		MOV A,D	7A	1	1	4
√	000B		SBB H	9C	1	1	4
√	000C		MOV H,A	67	1	1	4
√	000D		SHLD C054	22	3	5	16
	000E			54			
	000F			C0			
√	0010		HLT	76	1	2	5

Simulate

Start From →

Registers | Memory | Devices — Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
0000	2A
0001	50
0002	C0
0003	EB
0004	2A
0005	52
0006	C0
0007	7B
0008	95
0009	6F
000A	7A
000B	9C
000C	67
000D	22
000E	54
000F	C0
0010	76
C050	42
C051	78
C052	31
C053	69
C054	11
C055	0F

Write an assembly language program that performs 1's complement of an 8 bit number.

Steps:

- The number to be complemented is stored in C050.
- Result is stored in C051.

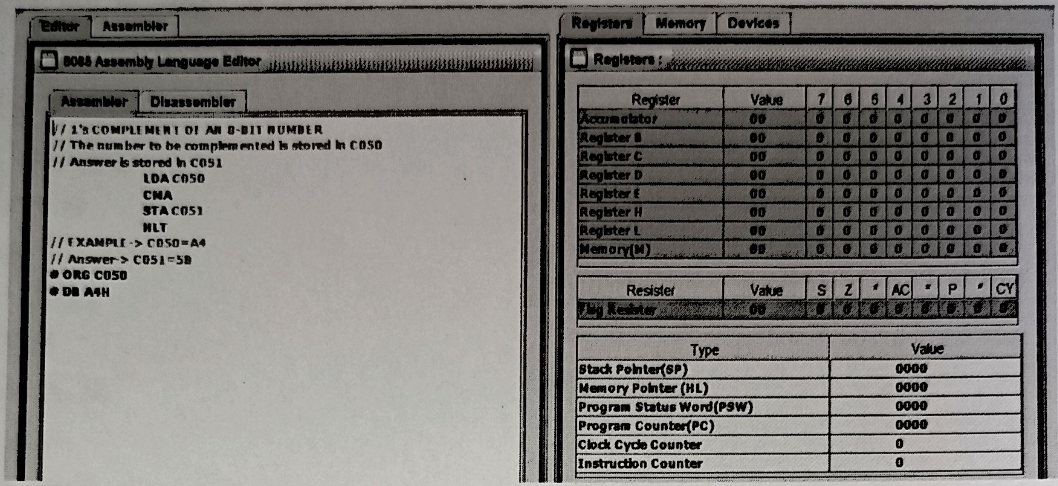
Program:

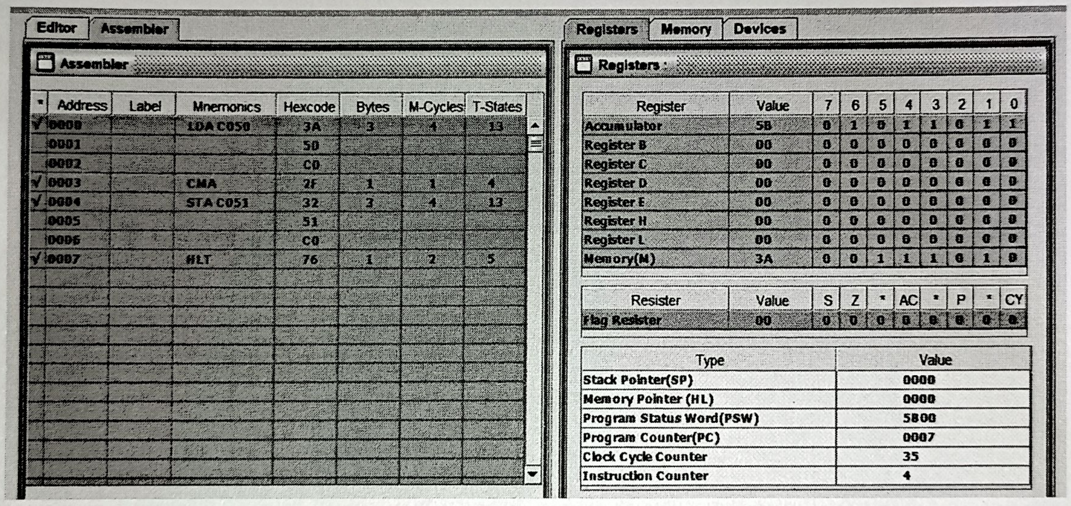
```
LDA C050
CMA
STA C051
HLT
#ORG C050
#DB A4H
```

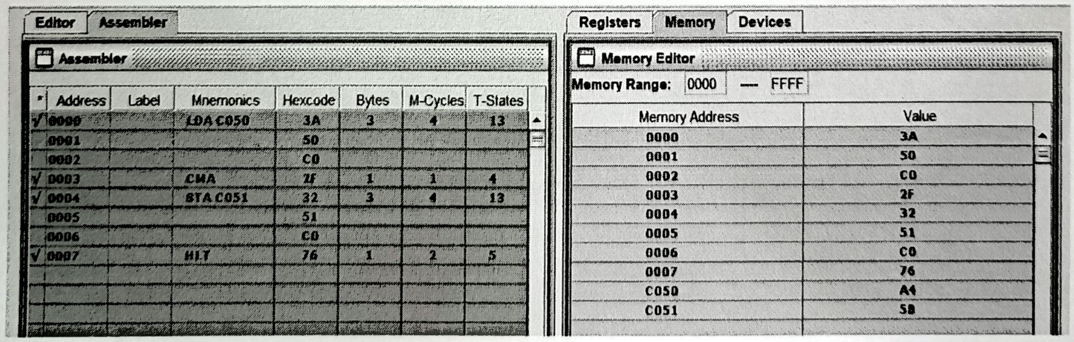
Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

Laboratory Experiment:

1's complement of 8 bit number.







Write an assembly language program that performs 1's complement of a 16 bit number.

Steps:

- The 16 bit number is stored in C050, C051.
- The result is stored in C052, C053.

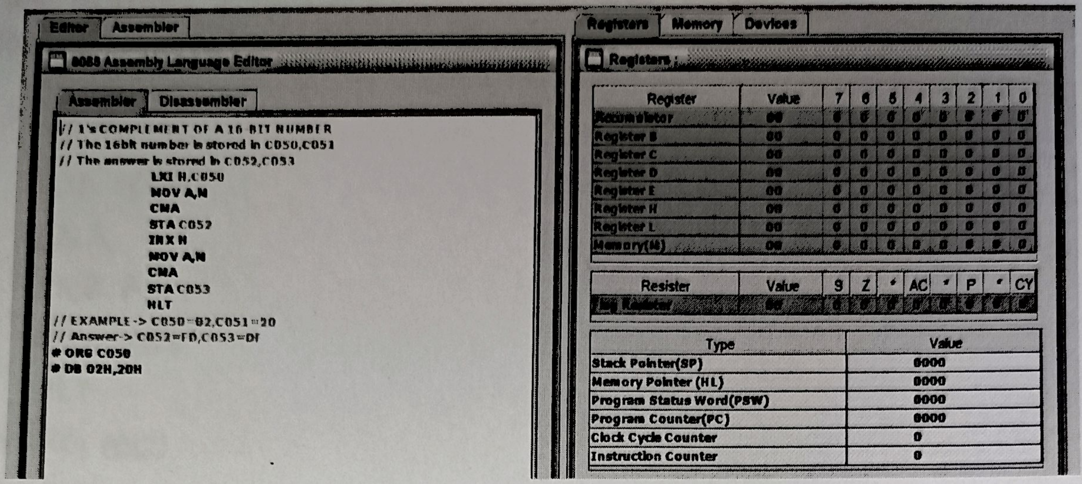
Program:

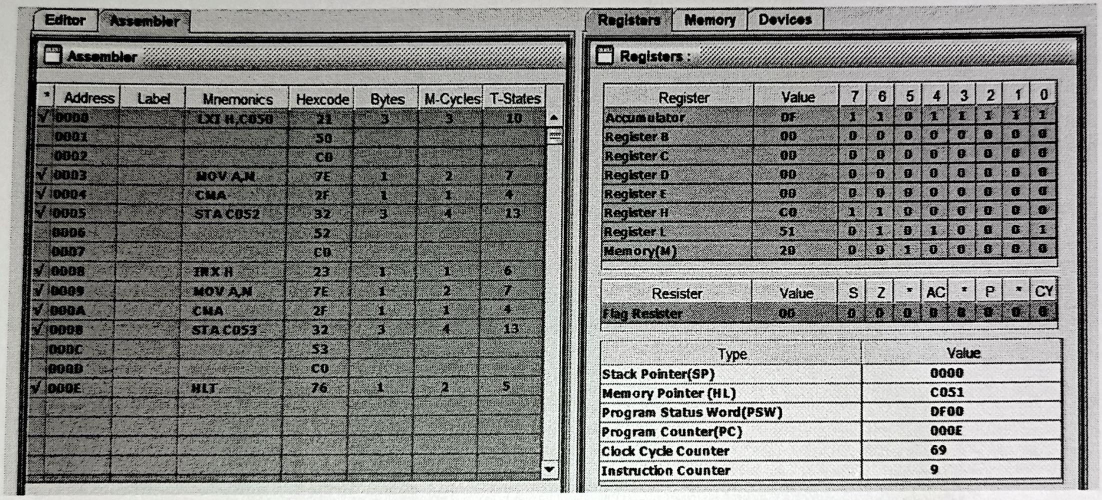
```
LXI H, C050
MOV A, M
CMA
STA C052
INX H
MOV A, M
CMA
STA C053
HLT
# ORG C050
# DB 02H, 20H
```

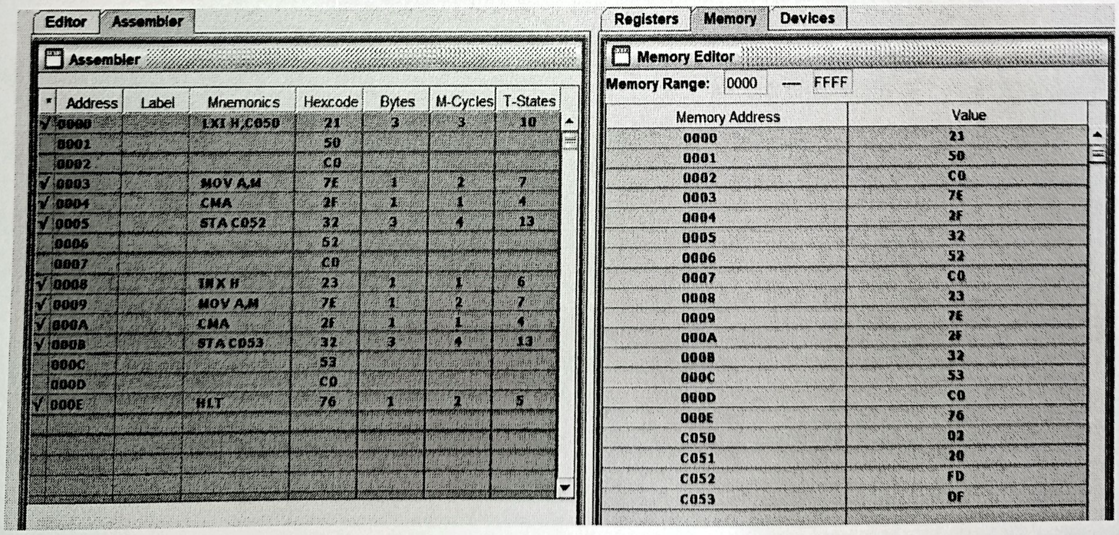
Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

Laboratory Experiment:

1's complement of 16 bit number.







Write an assembly language program that performs 2's complement of 8 bit number.

Steps:

- The number to be complemented is stored in C050.
- Result is stored in C051.

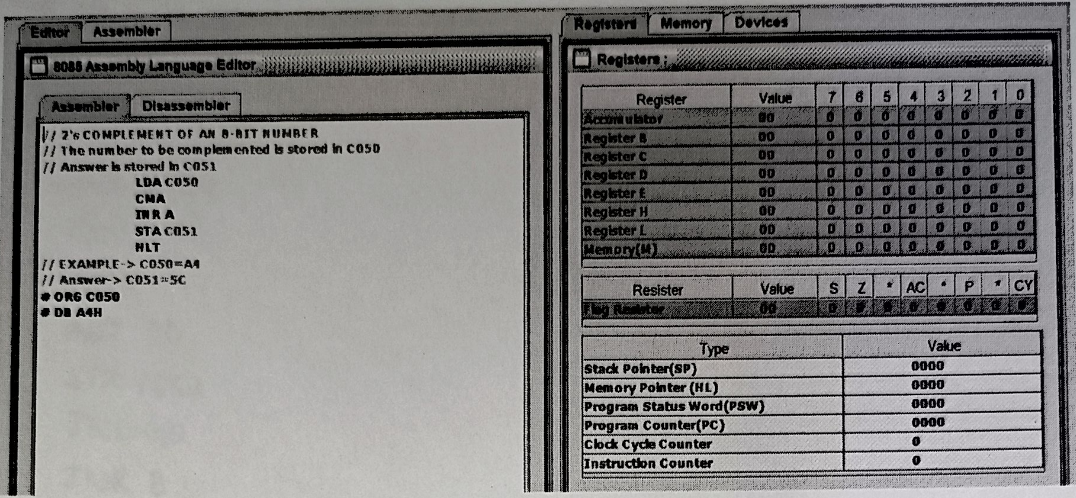
Program:

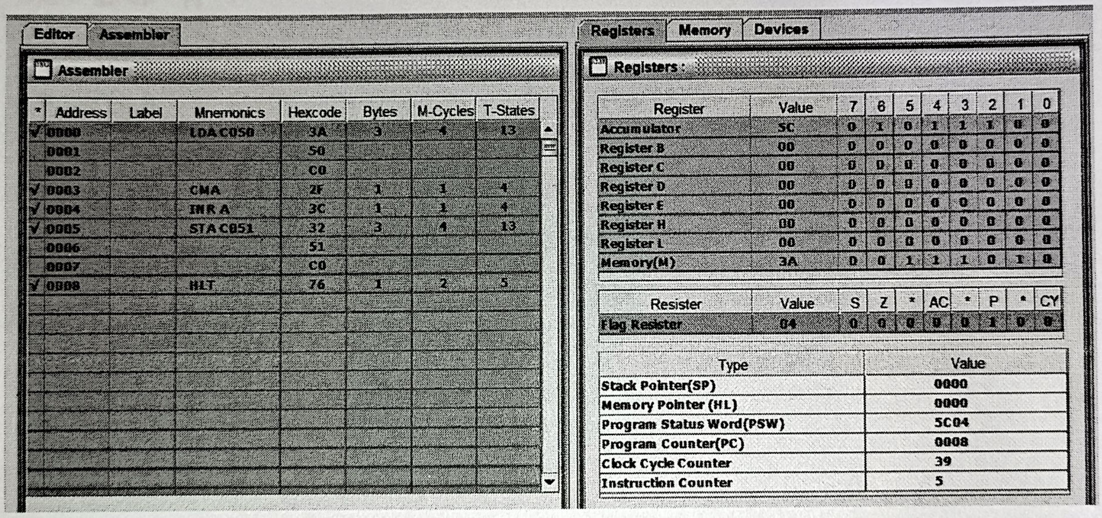
```
   LDA C050
   CMA
   INR A
   STA C051
   HLT
#ORG C050
#DB A4H
```

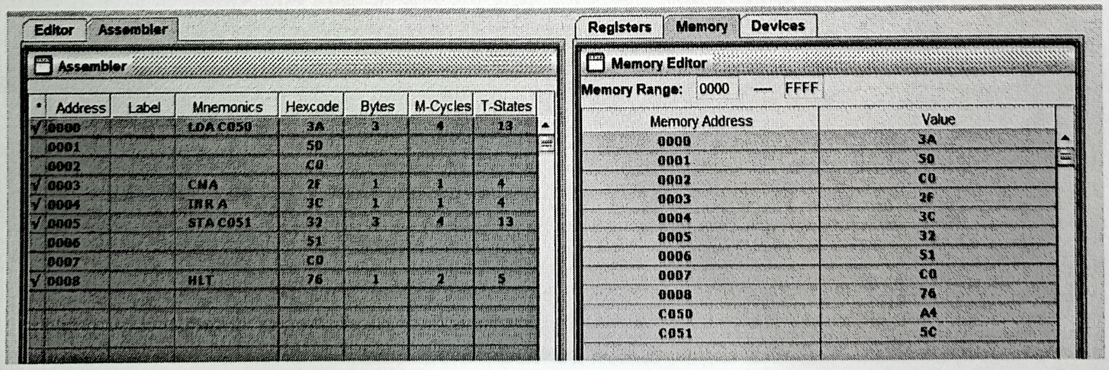
Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

Laboratory Experiment:

2's complement of 8 bit number.







Write an assembly language program that performs 2's complement of 16 bit number.

Steps:

- The 16 bit number is stored in C050, C051.
- The result is stored in C052, C053.

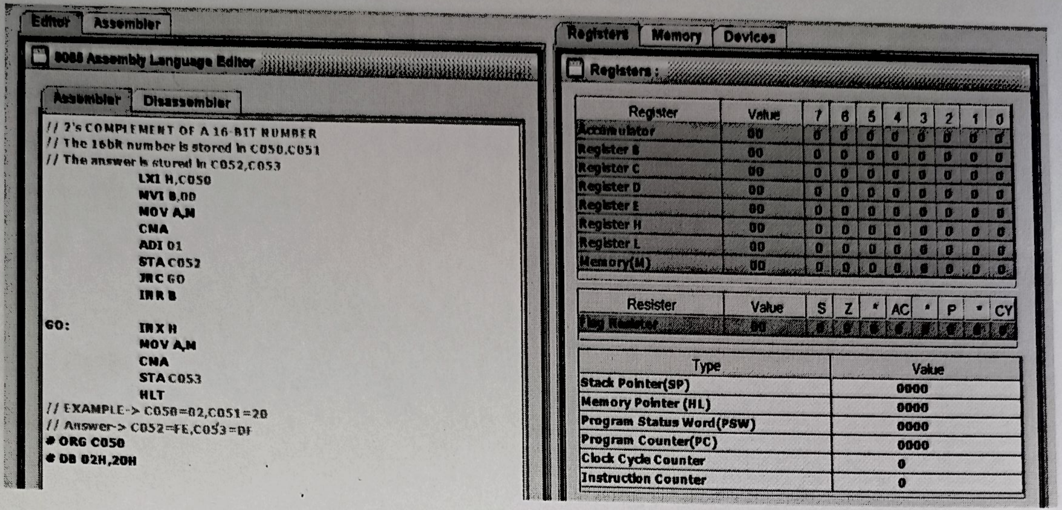
Program:

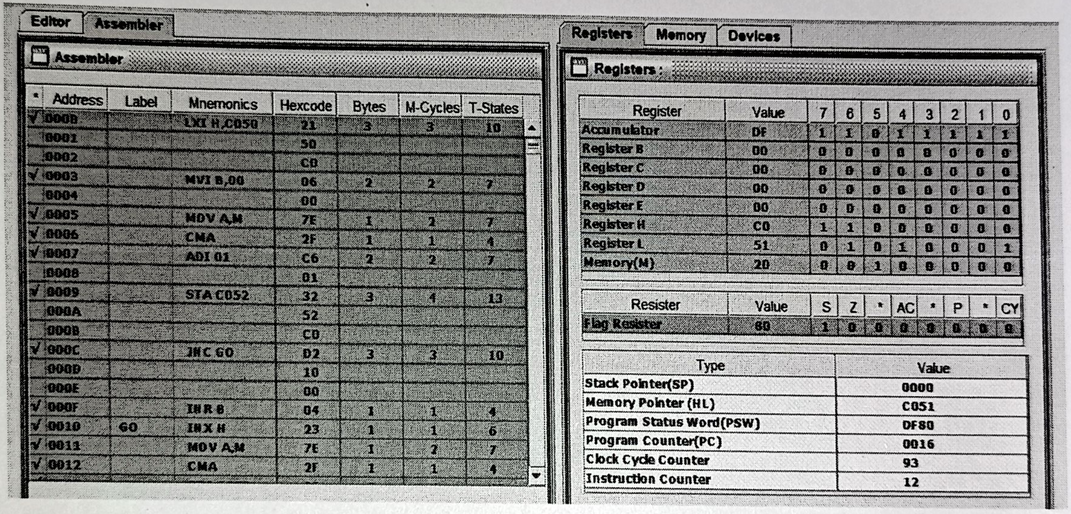
```
        LXI H, C050
        MVI B, 00
        MOV A, M
        CMA
        ADI 01
        STA C052
        JNC GO
        INR B
GO:     INX H
        MOV A, M
        CMA
        STA C053
        HLT
# ORG C050
# DB 02H, 20H
```

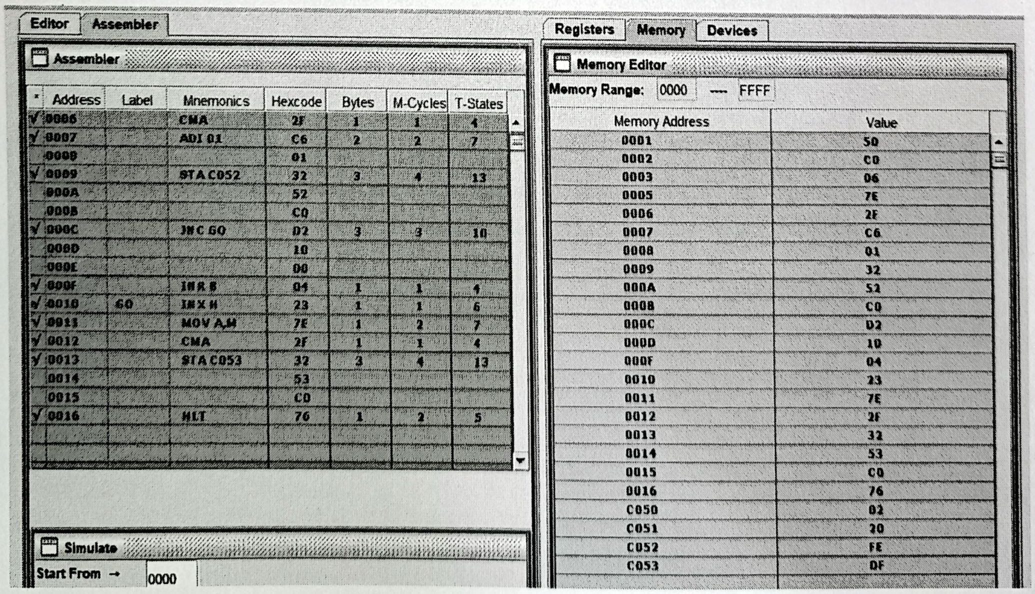
Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

Laboratory Experiment:

2's complement of 16 bit number.







(19)

write an assembly language program that mask off the least significant 4 bits of an 8 bit number.

Steps:

- The number to be masked is stored in C050.
- The result is stored in C051.

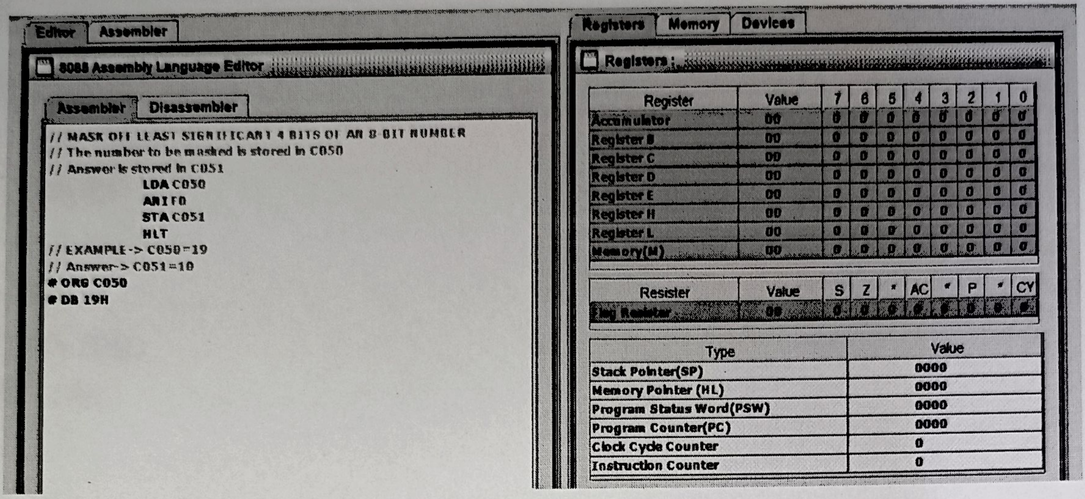
Program:

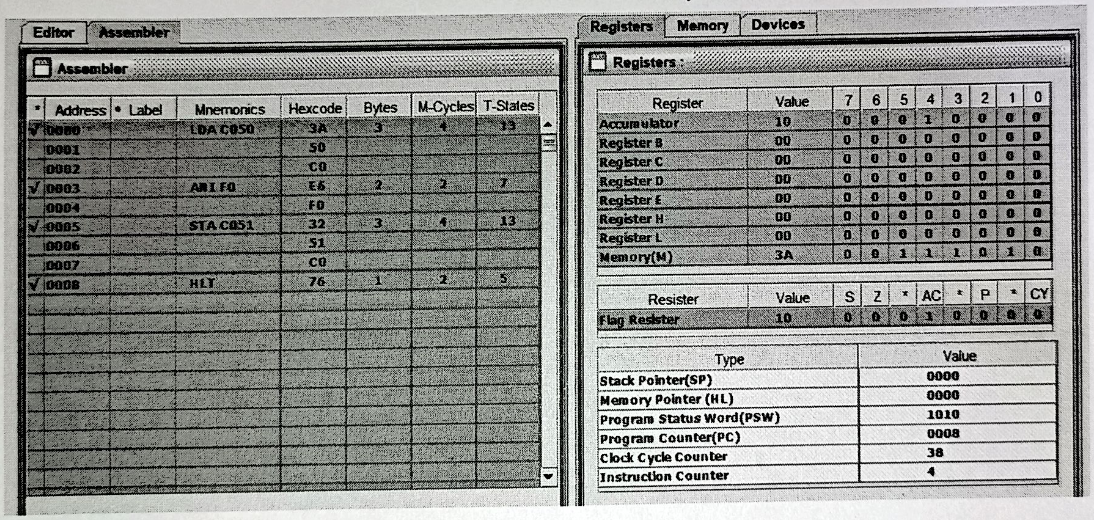
```
LDA C050
ANI F0
STA C051
HLT
#ORG C050
#DB 19H
```

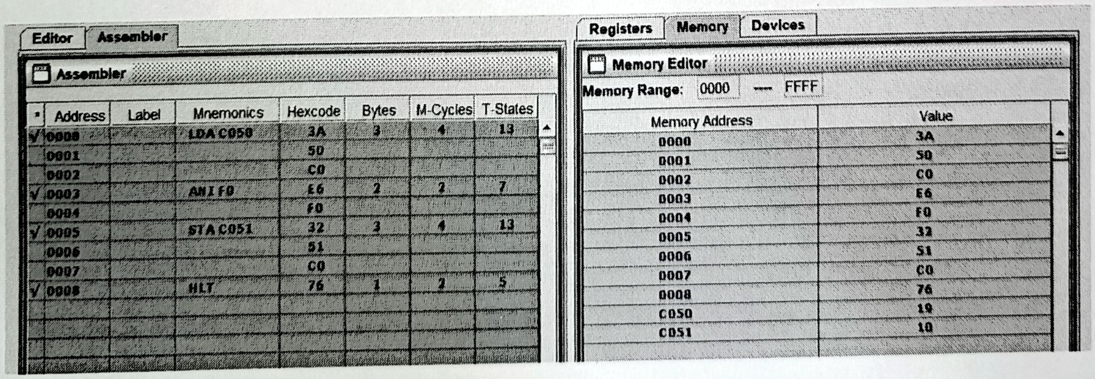
Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

Laboratory Experiment:

Mask off the least significant 4 bits of an 8 bit number.







Write an assembly language program that mask off the most significant 4 bits of an 8 bit number.

Steps:

- The number to be masked is stored in C050.
- Result is stored in C051.

Program:

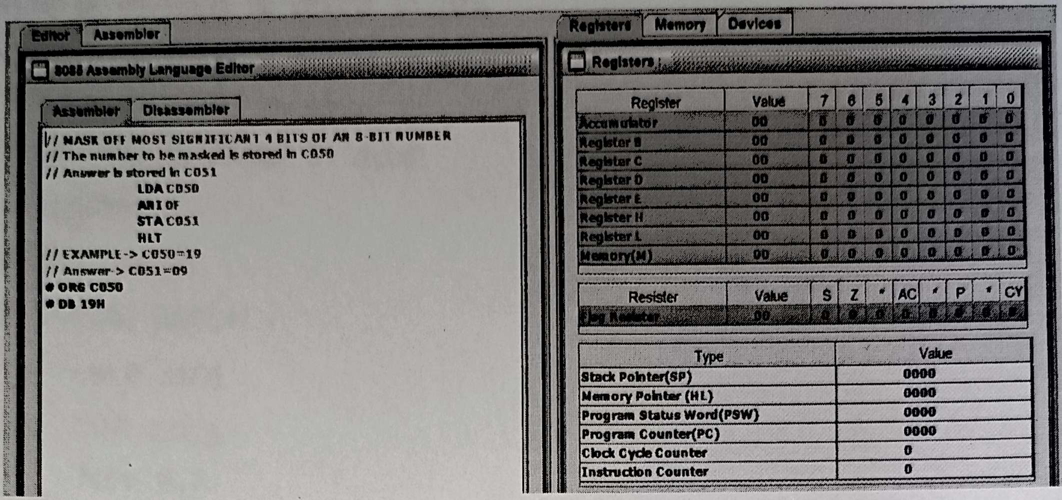
```
   LDA C050
   ANI 0F
   STA C051
   HLT
#ORG C050
#DB 19H
```

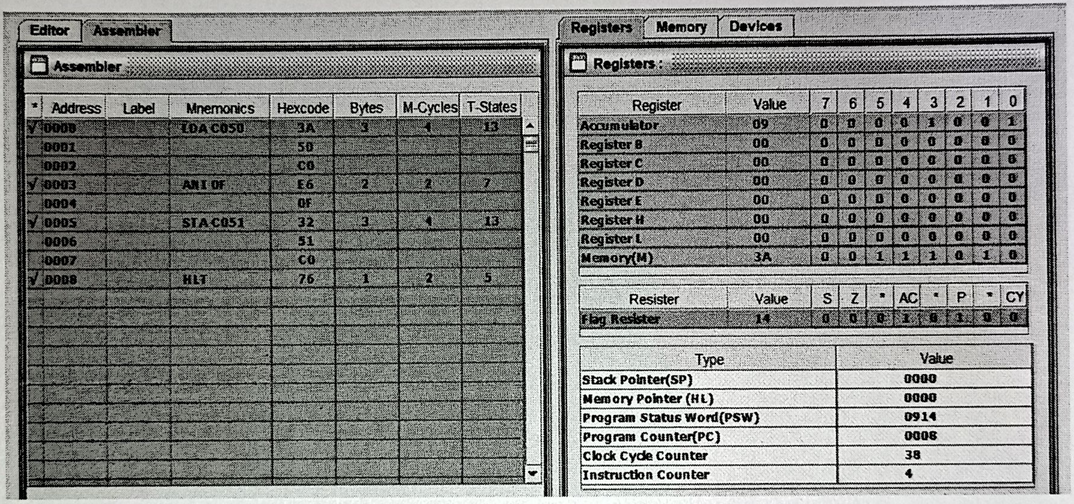
Name: ASHESH NEUPANE

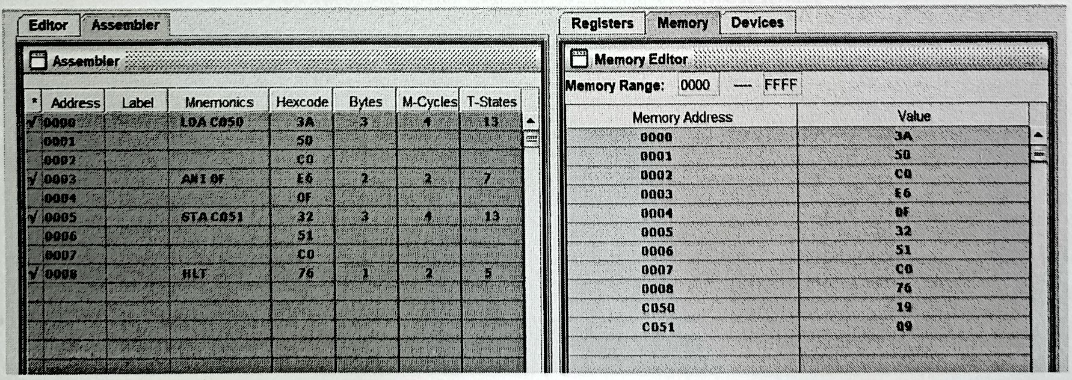
TU Exam Roll No: 16295/23

Laboratory Experiment:

Mask off the most significant 4 bits of an 8 bit number.







write an assembly language program that perform the 8-bit number division.

Steps:

- LSB of dividend is stored in 2501.
- MSB of dividend is stored in 2502.
- Divisor is stored in 2504.
- Remainder is stored in 2505.

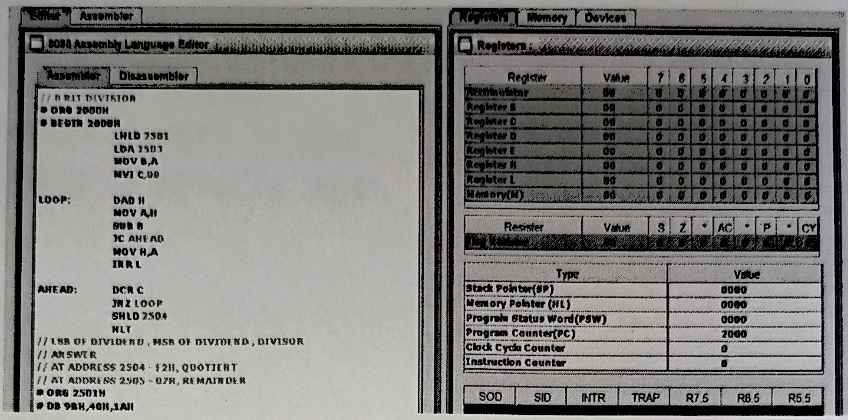
Program:

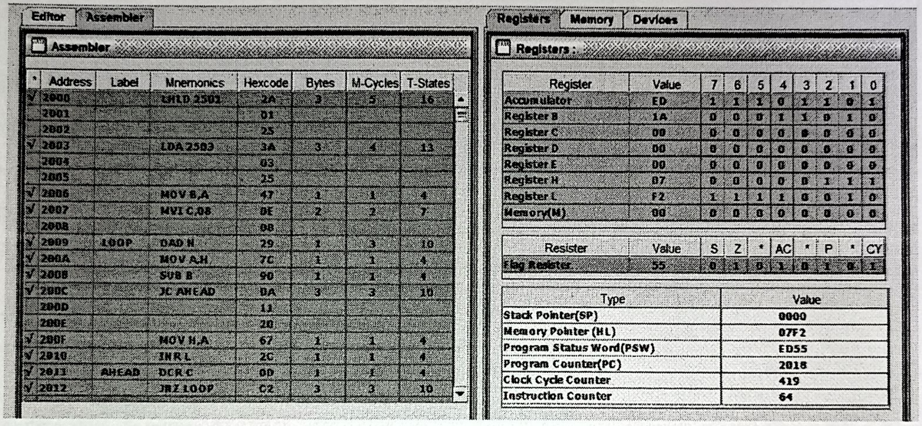
```
#ORG 2000H
# BEGIN 2000H
        LHLD 2501
        LDA 2503
        MOV B,A
        MVI C, 08
LOOP:   DAD H
        MOV A, H
        SUB B
        JC AHEAD
        MOV H, A
        INR L
AHEAD:  DCR C
        JNZ LOOP
        SHLD 2504
        HLT
# ORG 2501H
# DB 9BH, 48H, 1AH
```

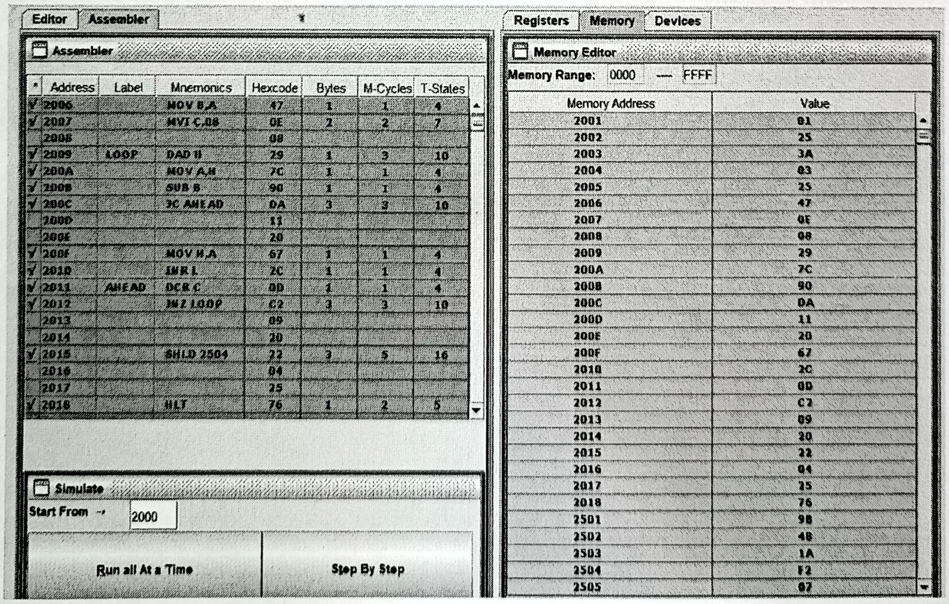
Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

Laboratory Experiment:

8 bit number division







Write an assembly language program that perform 8 bit number multiplication having 16 bit product.

Steps:

- LSB of multiplicand is stored in 2501.
- MSB of multiplicand is stored in 2502.
- Multiplier is stored in 2503.
- LSB of product is stored in 2504.
- MSB of product is stored in 2503.

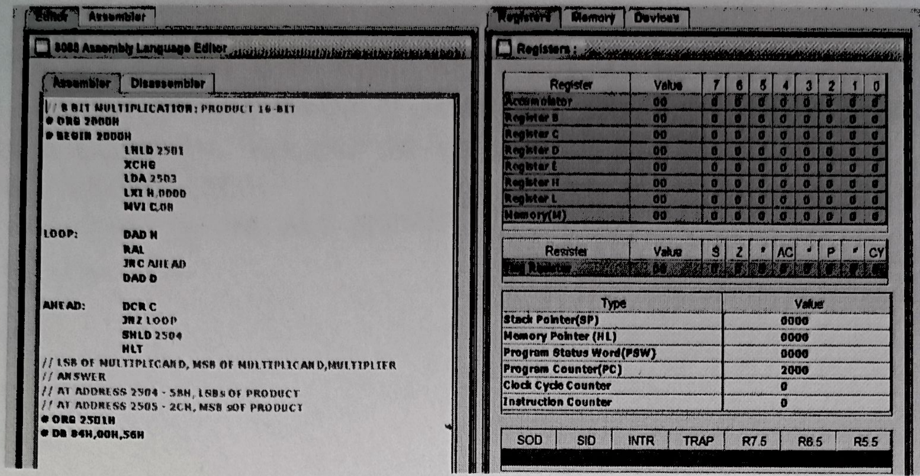
Program:

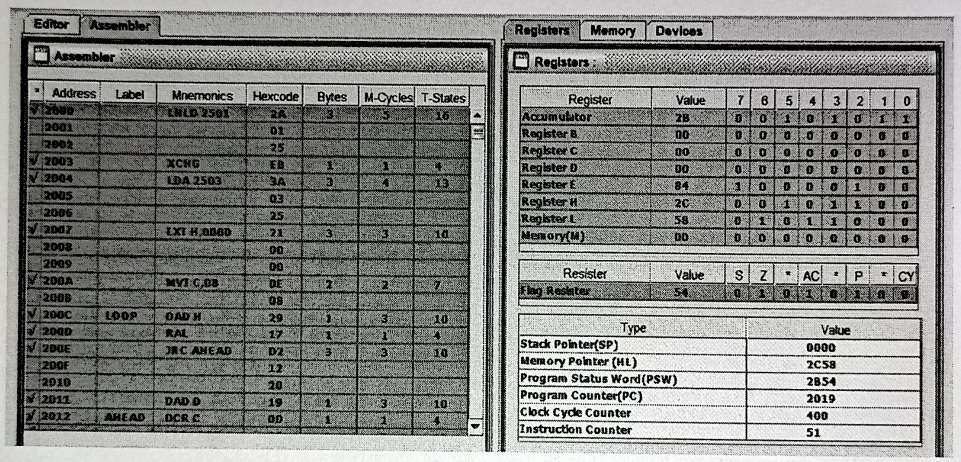
```
#ORG 2000H
#BEGIN 2000H
      LHLD 2501
      XCHG
      LDA 2503
      LXI H,0000
      MVI C,08
LOOP: DAD H
      RAL
      JNC AHEAD
      DAD D
AHEAD: DCR C
      JNZ LOOP
      SHLD 2504
      HLT
#ORG 2501H
#DB 84H,00H,56H
```

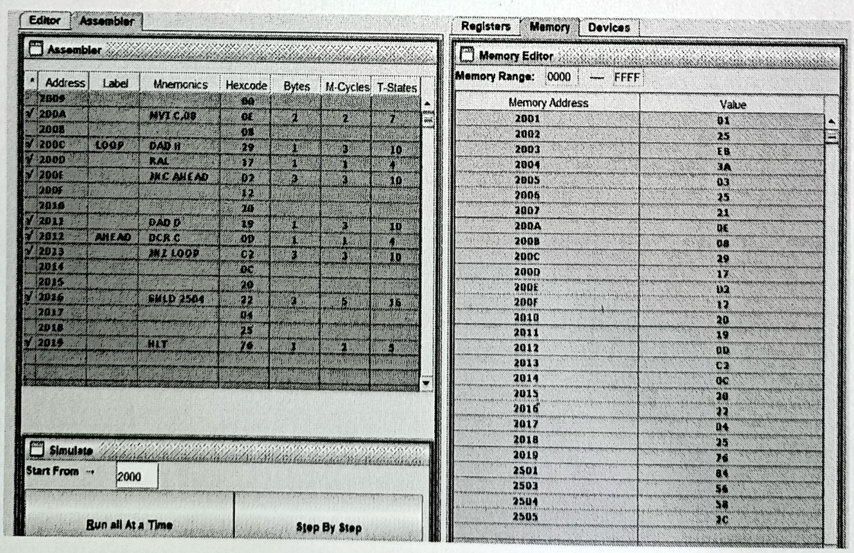
Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

Laboratory Experiment:

8 bit number multiplication having 16 bit product







write an assembly language to perform the bitwise AND operation of two 8-bit numbers. (Logical AND Operation)

Steps:

- Store the first 8 bit number at memory location 2050H.
- Store the second 8 bit number at memory location 2051H.
- Load the first number from 2050H into the accumulator A.
- Increment the memory pointer to access the second number at 2051H.
- Perform a bitwise AND between the value in A and the value in memory location 2051H.
- Store the result of the AND operation in memory location 2052H.
- Halt the execution.

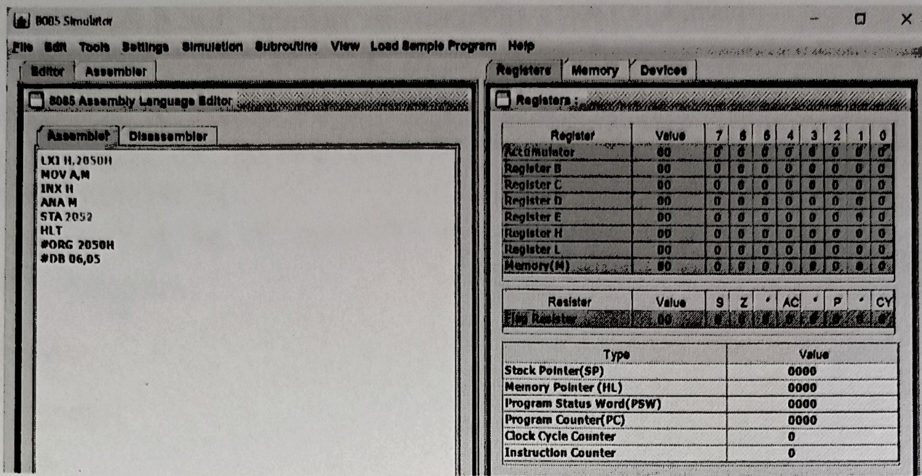
Program:

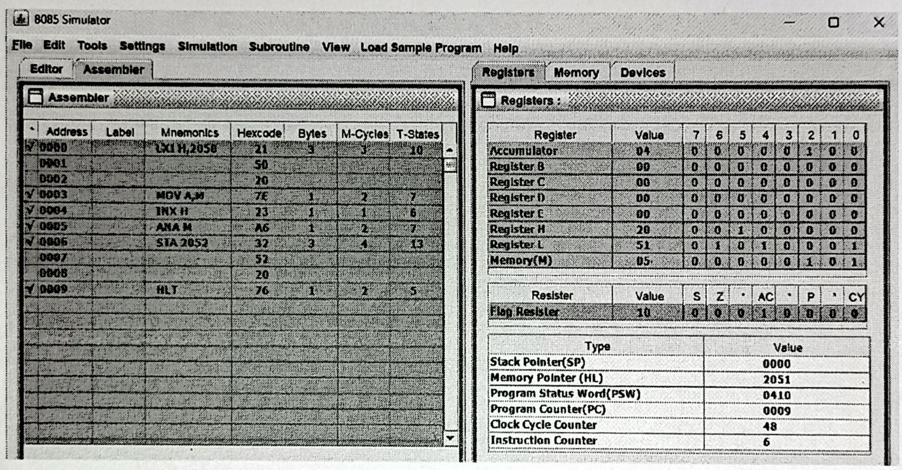
```
LXI H, 2050H
MOV A,M
INX H
ANA M
STA 2052
HLT
#ORG 2050H
# DB 06,05
```

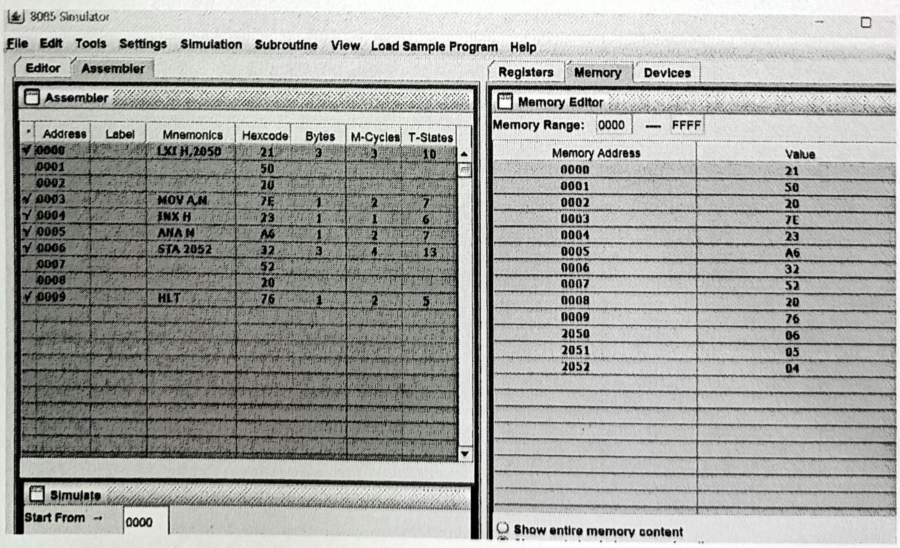
Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

Laboratory Experiment:

Logical AND Operation







write an assembly language program to perform the bitwise OR operation on two 8-bit numbers. (Logical OR operation)

Steps:

- Store the first 8-bit number at memory location 2050H.
- Store the second 8-bit number at memory location 2051H.
- Load the first number into the accumulator A.
- Increment the HL pair to point to the next memory location 2051H.
- Perform a bitwise OR operation between the contents of the accumulator A and the second number M.
- Store the result of the OR operation into memory location 2052H.
- Halt the program.

Program:

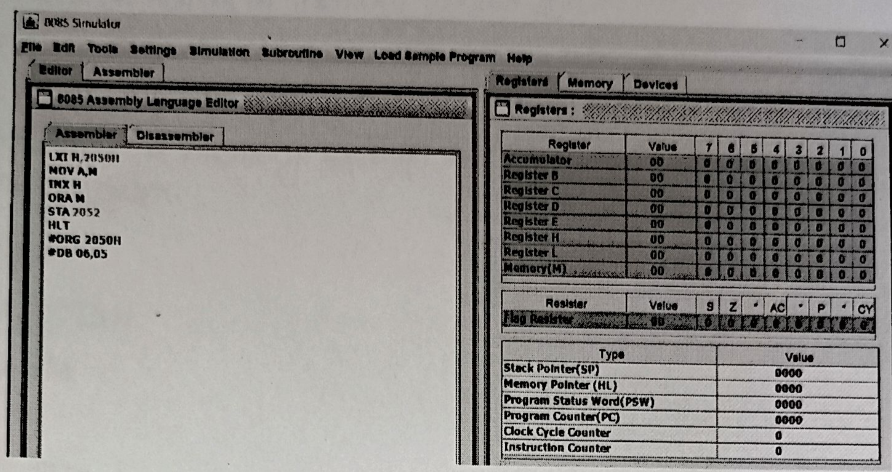
```
LXI H, 2050H
MOV A, M
INX H
ORA M
STA 2052
HLT
#ORG 2050H
#DB 06, 05
```

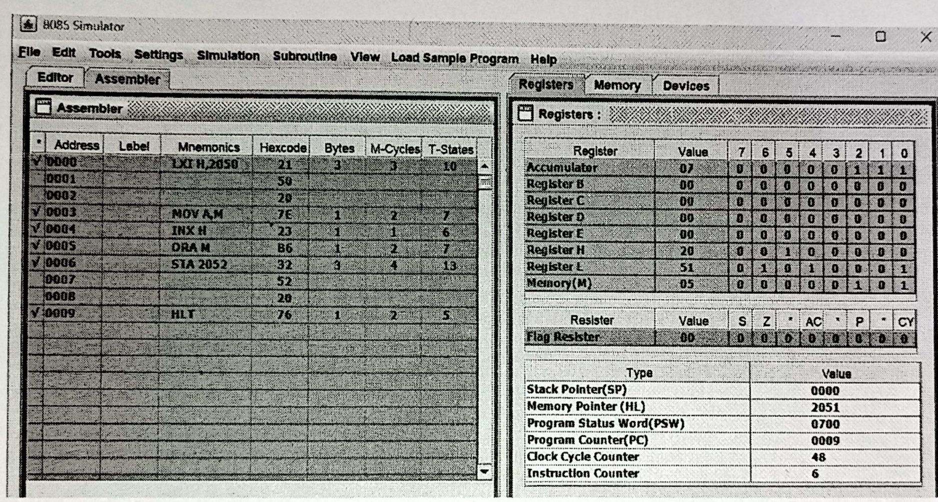
Name: ASHESH NEUPANE

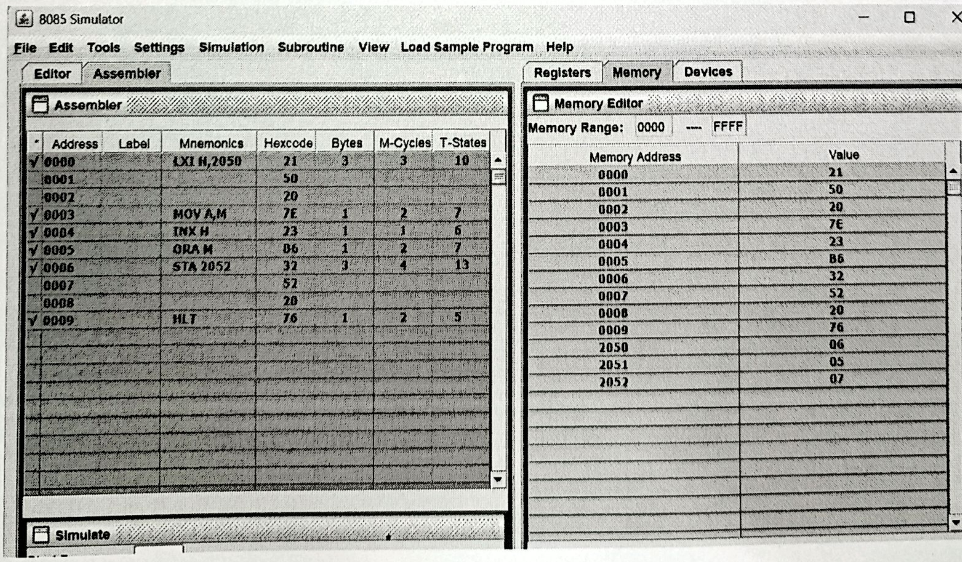
TU Exam Roll No: 16295/23

Laboratory Experiment:

Logical OR Operation







Write an assembly language program to perform the bitwise NOT operation on an 8-bit number. (Logical NOT Operation)

Steps:

- Store the 8-bit number at memory location 2050H.
- Load the number from 2050H into the accumulator A.
- Perform the bitwise NOT operation on the accumulator.
- Store the result at memory location 2051H.
- Halt the program.

Program:

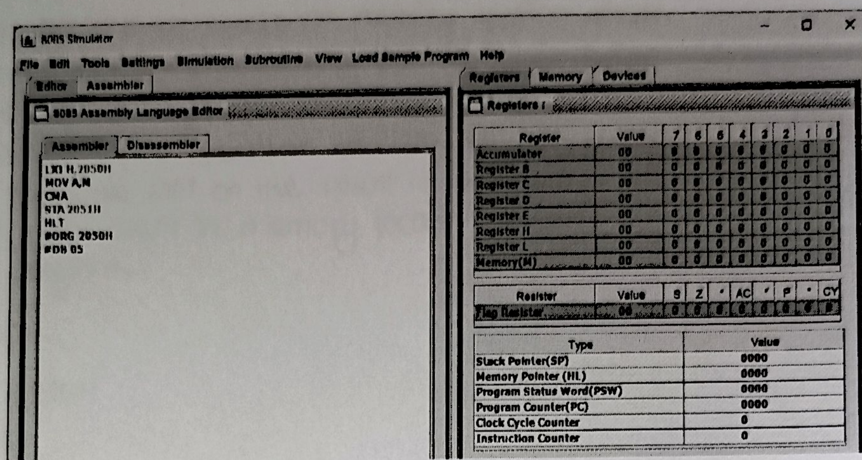
```
LXI H, 2050H
MOV A, M
CMA
STA 2051H
HLT
#ORG 2050H
#DB 05
```

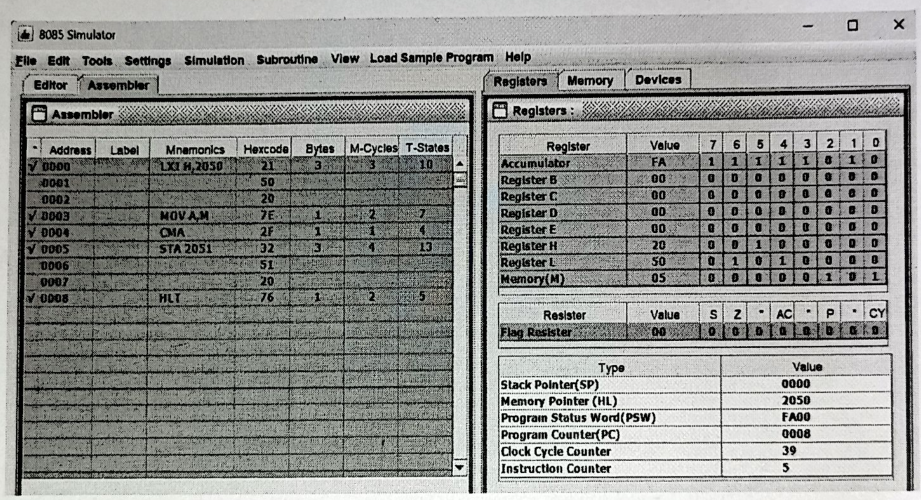
Name: ASHESH NEUPANE

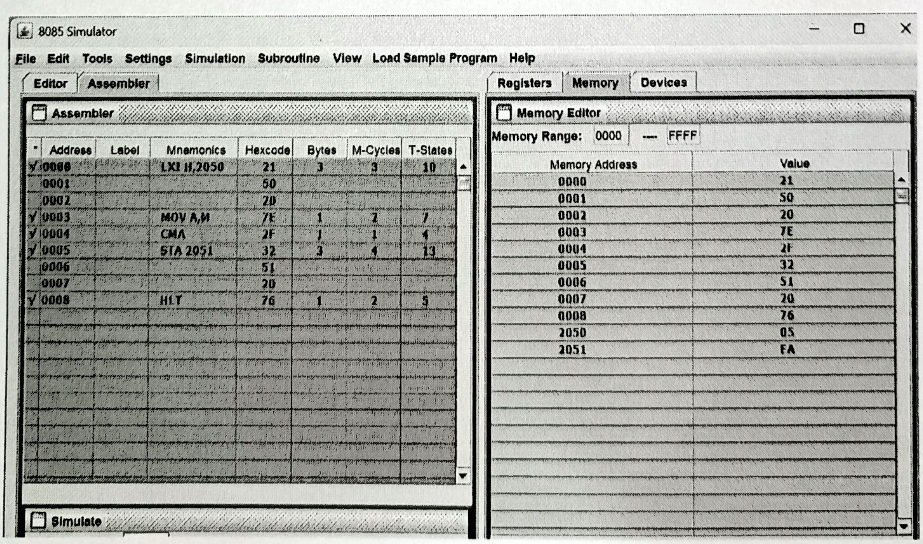
TU Exam Roll No: 16295/23

Laboratory Experiment:

Logical NOT Operation







Write an assembly language program to perform a bitwise NAND operation on two 8-bit numbers. (Logical NAND)

Steps:

- Store the first 8-bit number at memory location 2050H.
- Store the second 8-bit number at memory location 2051H.
- Load the first number from memory into the accumulator.
- Increment the memory pointer to point to the second number.
- Perform a bitwise AND operation between the two numbers.
- Perform a bitwise NOT on the result of the AND.
- Store the final result in memory location 2052H.
- Halt the program

Program:

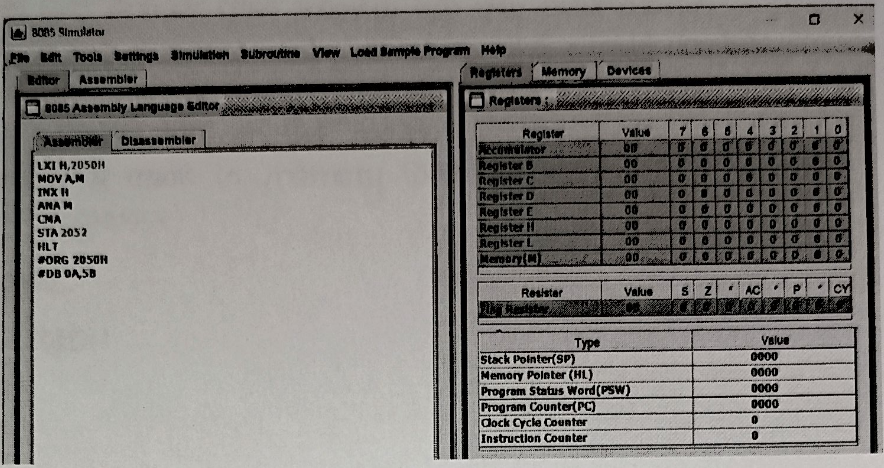
```
LXI H, 2050H
MOV A, M
INX H
ANA M
CMA
STA 2052
HLT
#ORG 2050H
#DB 0A, 5B
```

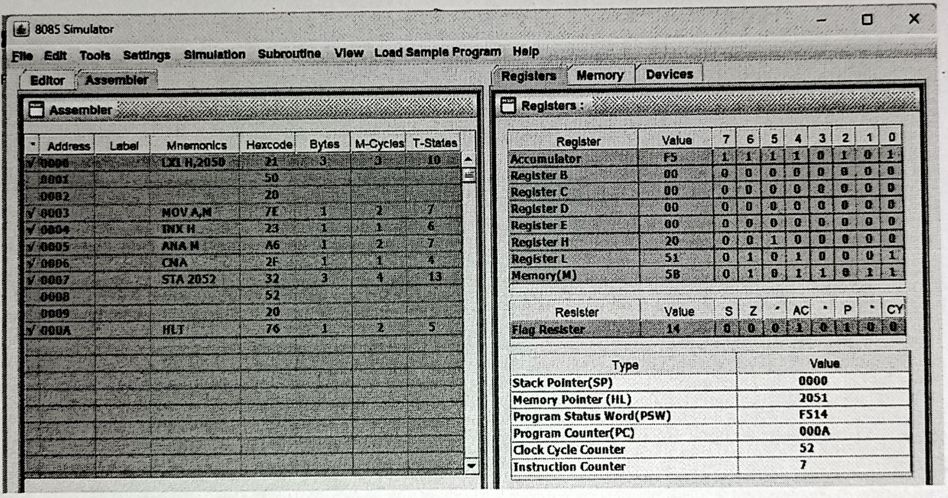
Name: ASHESH NEUPANE

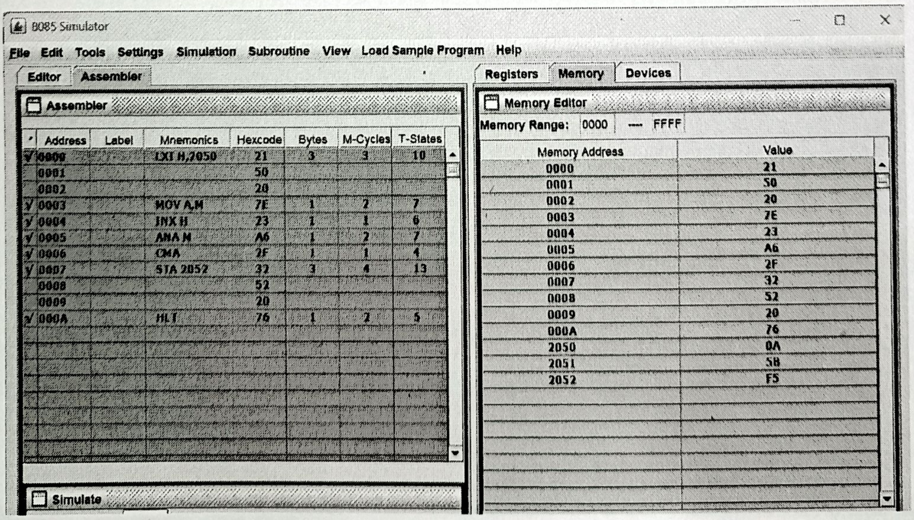
TU Exam Roll No: 16295/23

Laboratory Experiment:

Logical NAND Operation







Write an assembly language program to perform a bitwise NOR operation on two 8-bit numbers. (Logical NOR operation)

Steps:

- Store the first 8-bit number at memory location 2050H.
- Store the second 8-bit number at memory location 2051H.
- Load the first number from memory into the accumulator.
- Increment the memory pointer to access the second number.
- Perform a bitwise OR operation between the two numbers.
- Perform a bitwise NOT on the result.
- Store the final result in memory location 2052H.
- Halt the program.

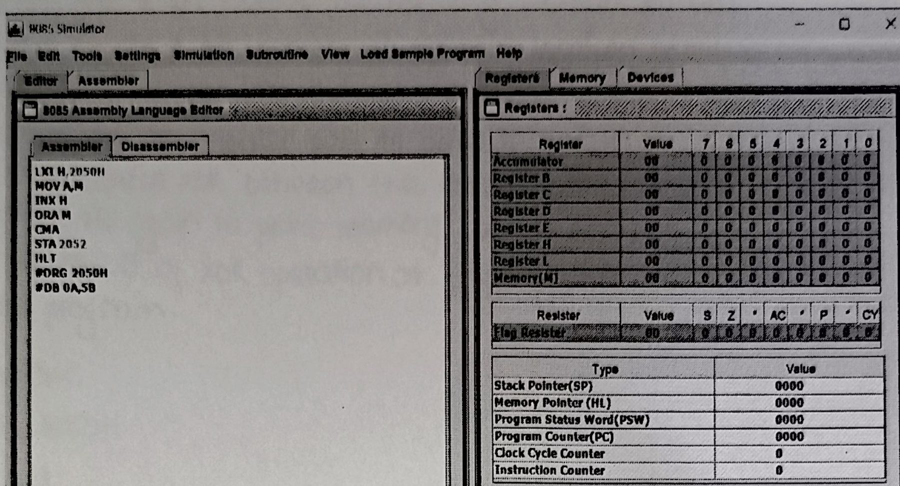
Program:

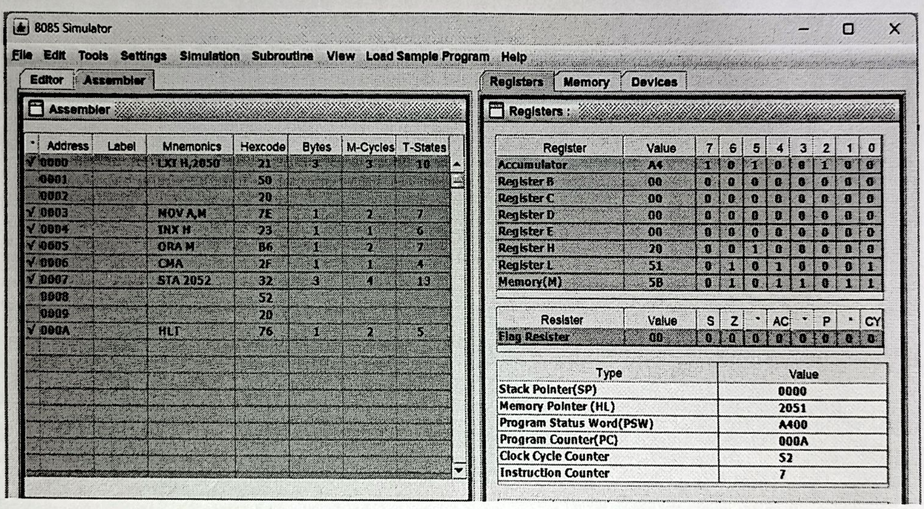
```
LXI H, 2050H
MOV A, M
INX H
ORA M
CMA
STA 2052
HLT
# ORG 2050H
# DB 0A, 5B
```

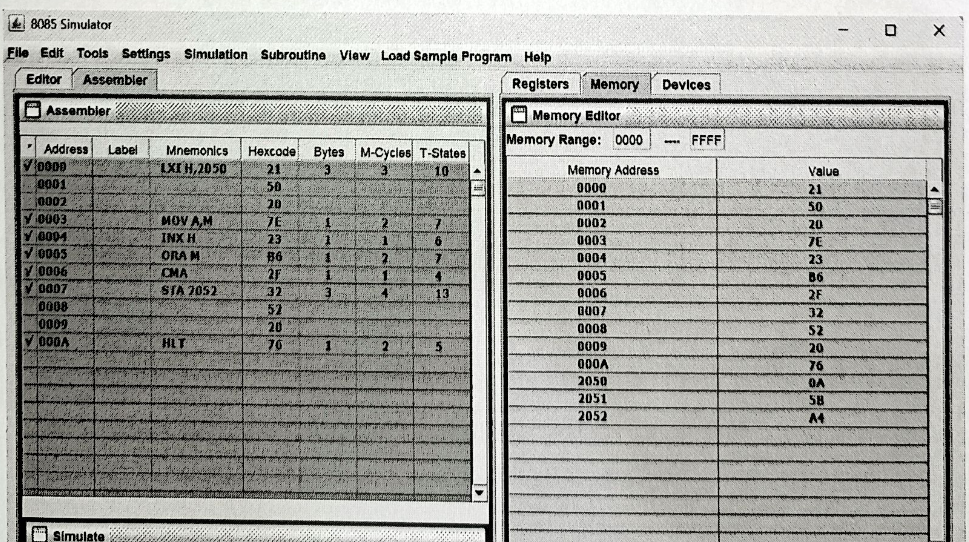
Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

Laboratory Experiment:

Logical NOR Operation







Write an assembly language program to perform a bitwise XOR operation on two 8-bit numbers. (Logical XOR operation)

Steps:

- Store the first 8-bit number at memory location 2050H.
- Store the second 8-bit number at memory location 2051H.
- Load the first number into the accumulator.
- Increment the HL register pair to point to 2051H.
- perform a bitwise XOR between the accumulator and the second number.
- Increment HL again to point memory location 2052H.
- Store the result of XOR operation at memory location 2052H.
- Halt the program.

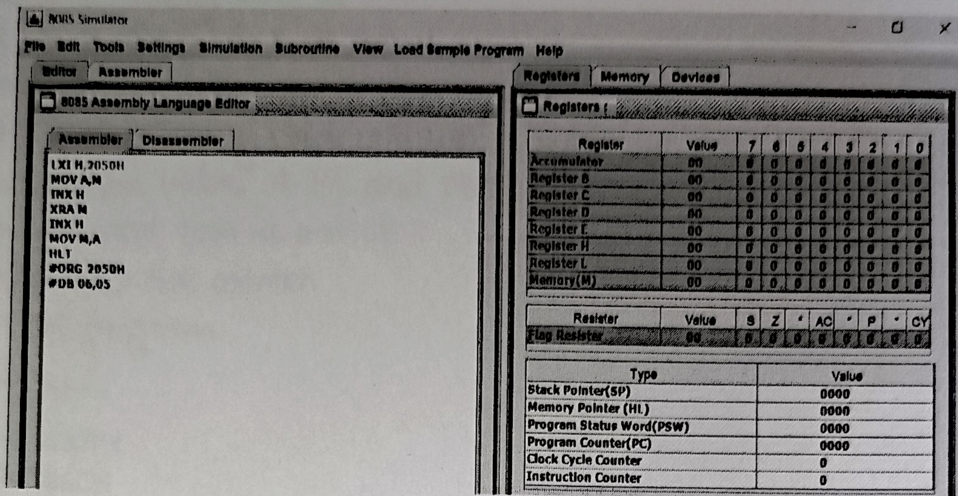
Program:

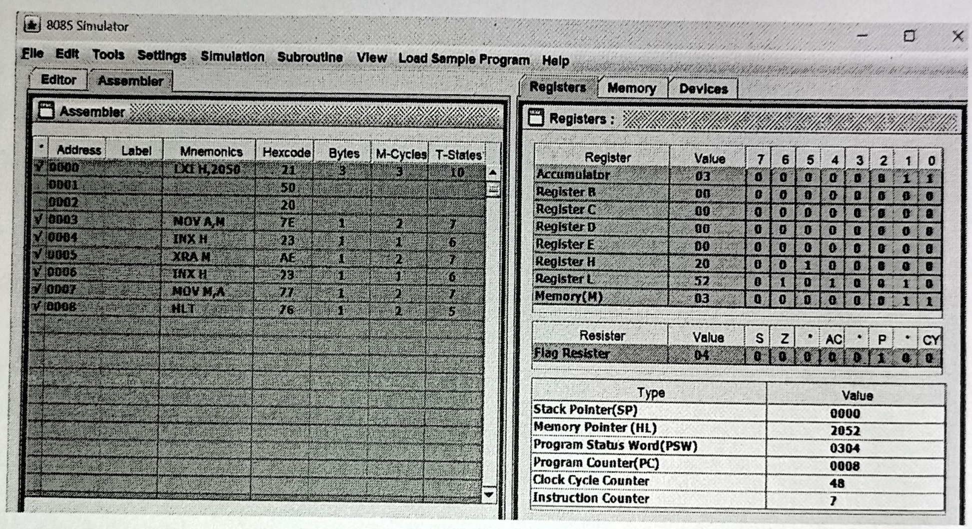
```
LXI H, 2050H
MOV A,M
INX H
XRA M
INX H
MOV M,A
HLT
#ORG 2050 H
# DB 06,05
```

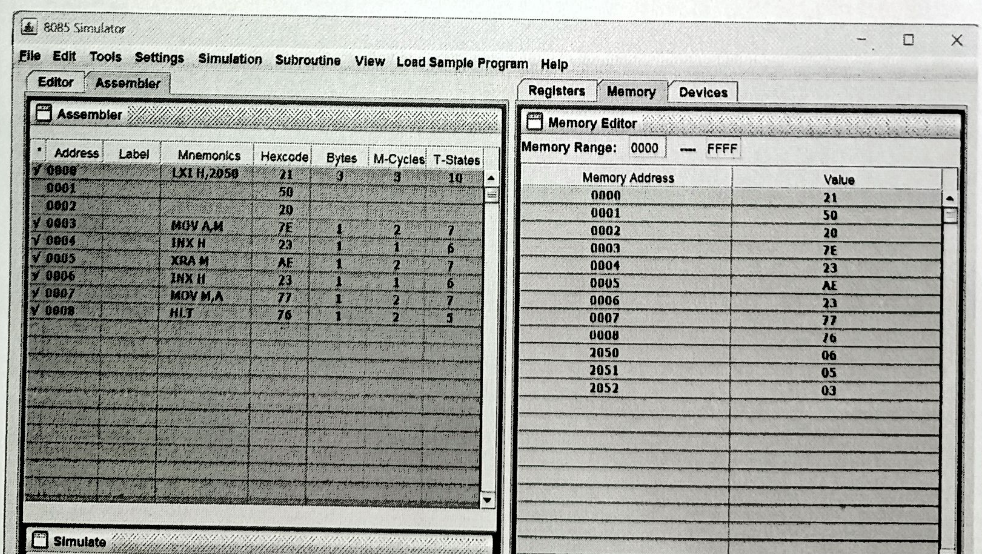
Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

Laboratory Experiment:

Logical XOR Operation







write an assembly language program to exchange (swap) contents of two memory blocks of size 10 bytes each. (2050H to 2060H)

Steps:

- Point HL to memory block at 2050H.
- Point DE to second block at 2060H.
- Set counter C to 10 (number of bytes to swap)
- Repeat the following steps until all 10 bytes are swapped:
    - ↳ Swap the values at HL and DE.
    - ↳ Increment both HL and DE
    - ↳ Decrease the counter.
- Stop the program.

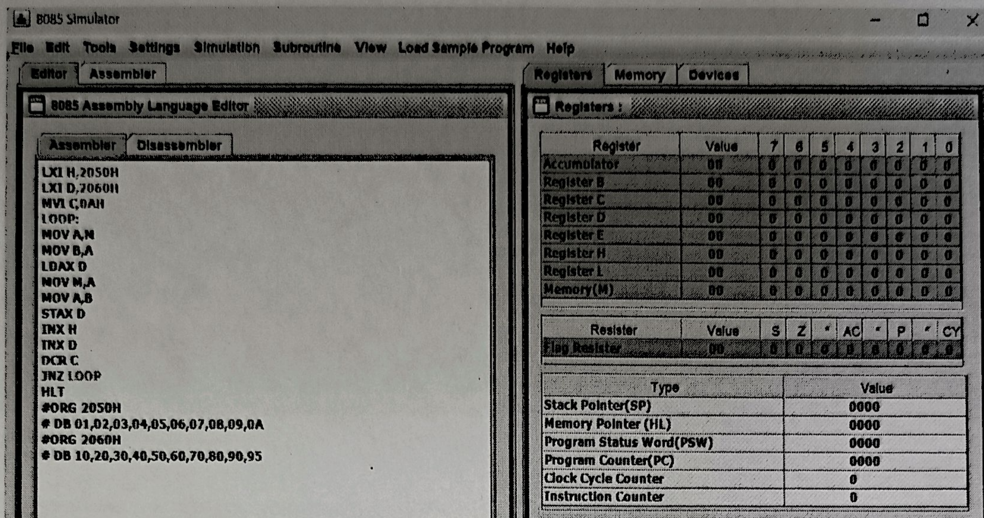
Program:

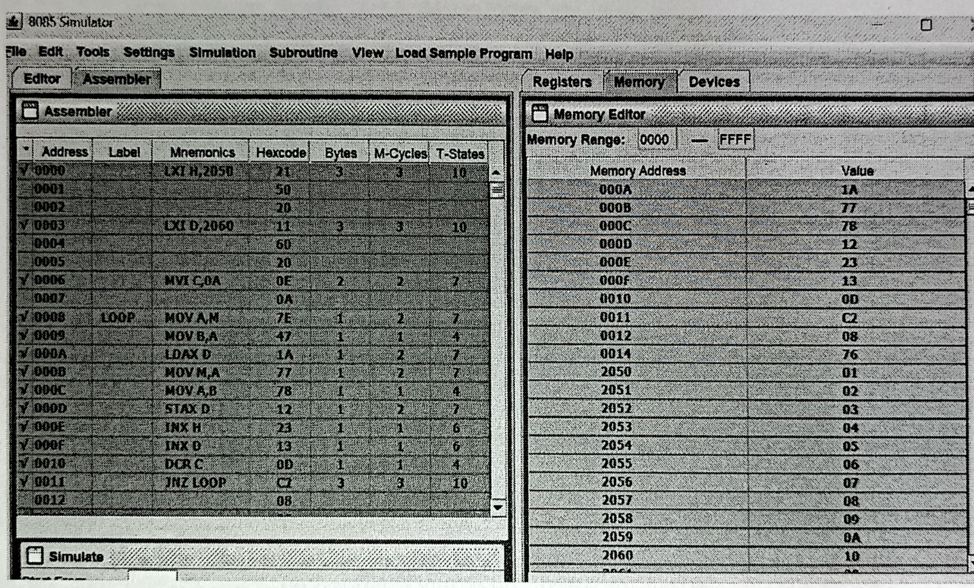
```
LXI H, 2050H
LXI D, 2060H
MVI C, 0AH
LOOP:
 MOV A, M
 MOV B, A
 LDAX D
 MOV M, A
 MOV A, B
 STAX D
 INX H
 INX D
 DCR C
 JNZ LOOP
 HLT
# ORG 2050H
# DB 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A
# ORG 2060H
# DB 10, 20, 30, 40, 50, 60, 70, 80, 90, 95
```

Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

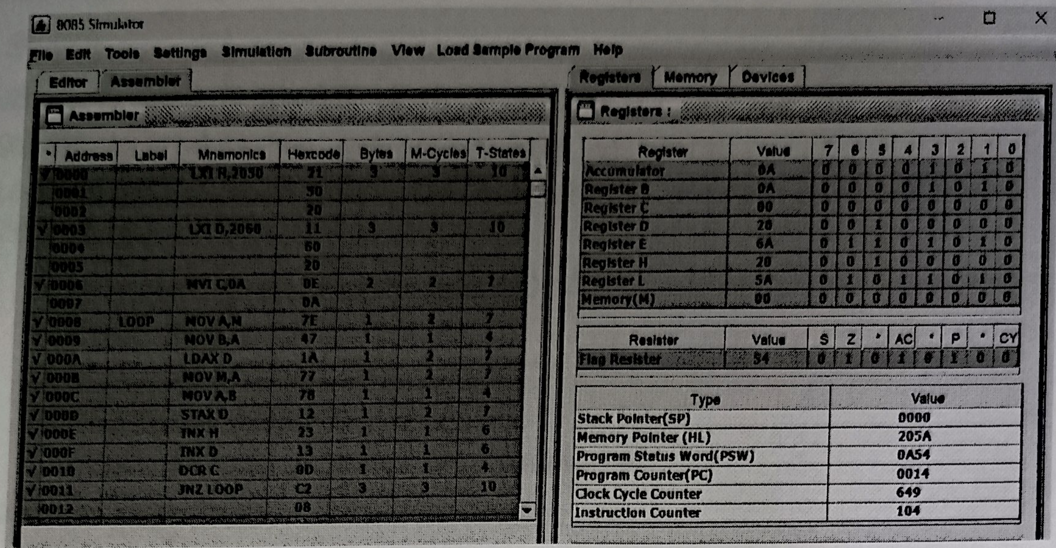
Laboratory Experiment:

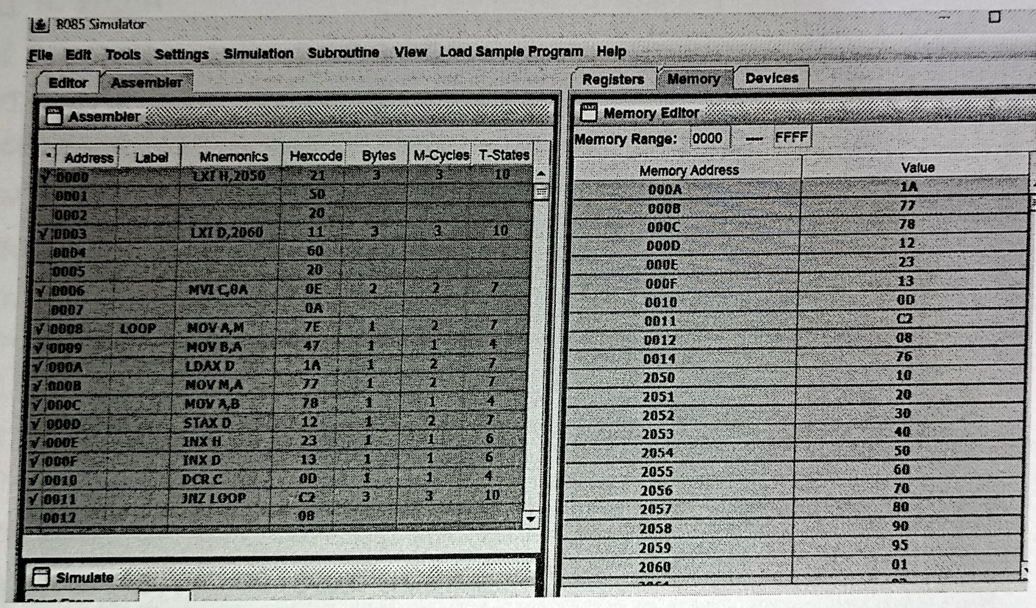
Exchange 10 Bytes of Data from 2050H to 2060H.





Name: ASHESH NEUPANE TU Exam Roll No: 16295/23





Exchanged Data from 2050H to 2060H

Write an assembly language program to transfer 10 bytes of data from one memory block to another. (2050H to 2060H)

Steps:

- Set HL to point to source block at 2050H.
- Set DE to point to the destination block at 2060H.
- Load register C with value 0AH (10 bytes to transfer)
- Start loop:
  - ↳ Load byte from HL into accumulator.
  - ↳ Store byte to DE
  - ↳ Increment HL and DE
  - ↳ Decrement counter C
  - ↳ Repeat until all 10 bytes are transferred.
- Halt the program.

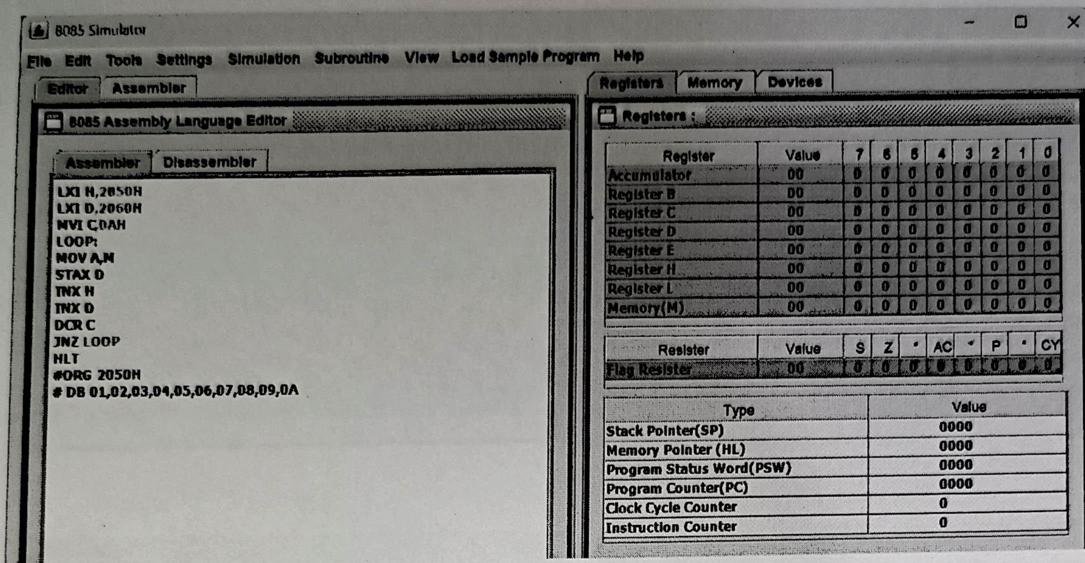
Program:

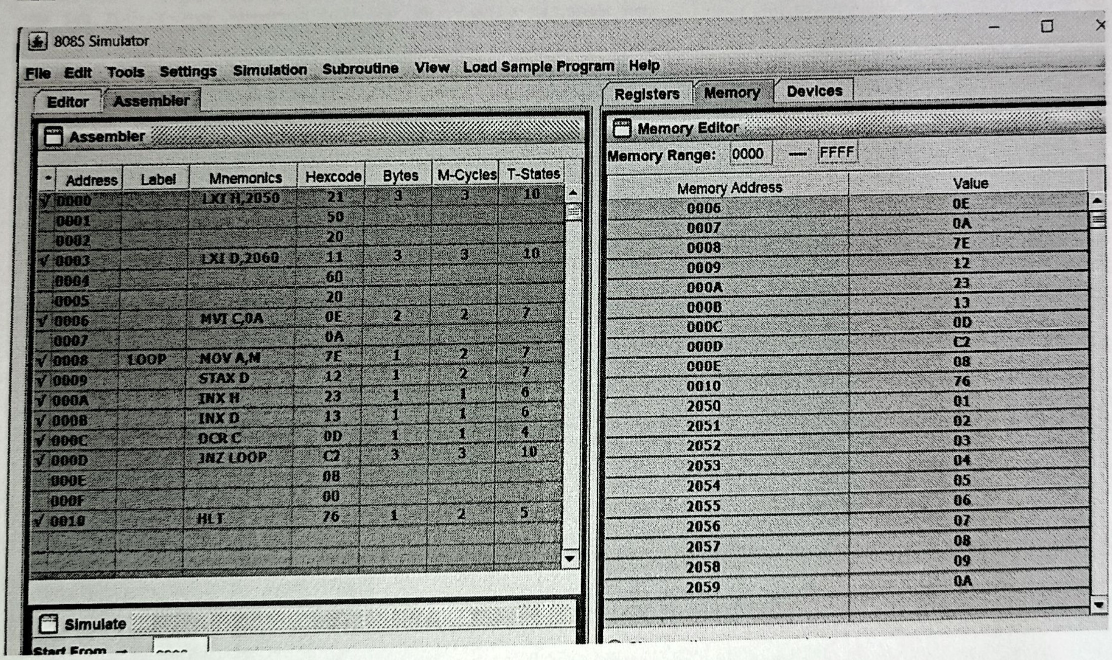
```
LXI H, 2050H
LXI D, 2060H
MVI C, 0AH
LOOP:
MOV A, M
STAX D
INX H
INX D
DCR C
JNZ LOOP
HLT
#ORG 2050H
# DB 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A
```

Name: ASHESH NEUPANE TU Exam Roll No: 16295/23

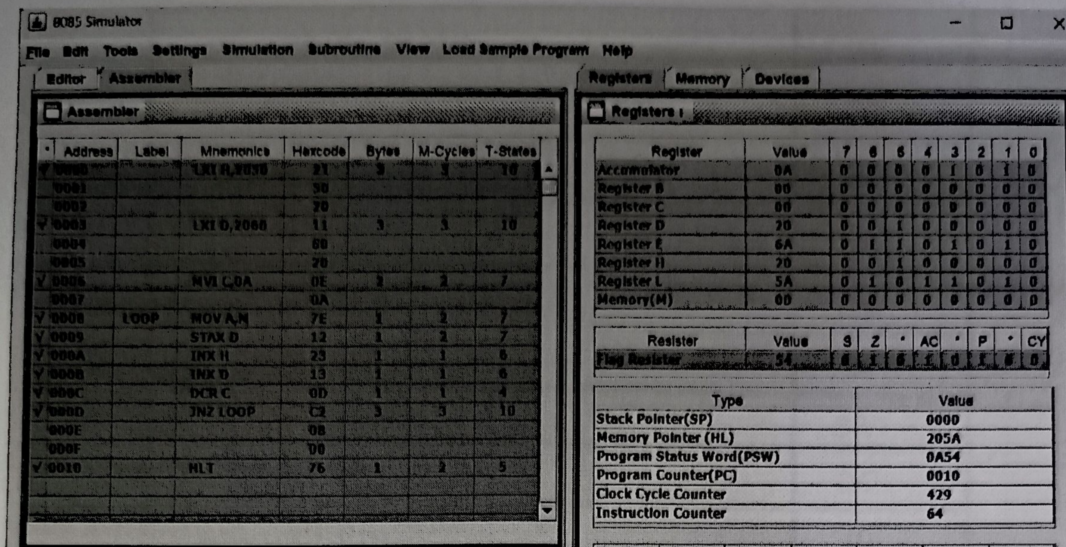
Laboratory Experiment:

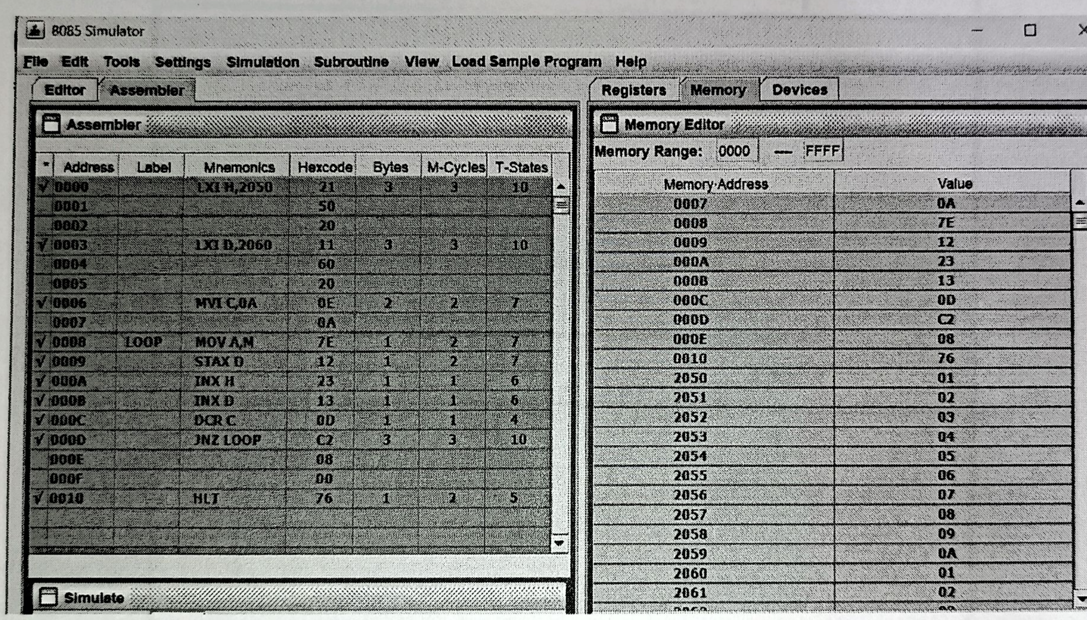
Transfer 10 Bytes of Data from 2050H to 2060H.





Name: ASHESH NEUPANE　　　　　　　　　　　　TU Exam Roll No: 16295/23





Transferred Data from 2050H to 2060H

Microprocessor Training Kit Lab

Address	Memory	Hex Code
→ 8000	LXI H, 8200H	21
8001		00
8002		82
8003	MOV A, M	7E
8004	INX H	23
8005	ADD M	86
8006	STA 9000H	32
8007		00
8008		90
8009	HLT	76

Program

Address	Memory
8200	0A
8201	05

Data

Address	Memory
9000H	0F

Result

Steps:

1) Load program and data in a microprocessor kit.

   Exam memory: Starting address of program (8000)

   Input → Hexcode of program

   • Reset

   Exam memory: Starting address of data (input)

   • Reset

2) Execute program

   Go → Program Start Address (8000 H)

   Execute

   • Reset

3) Check Result

   Exam memory 9000 H

   Next → 0F

Name: ASHESH NEUPANE TU Exam Roll No: 16295/23