INDIAN INSTITUTE OF INFORMATION TECHNOLOGY ,AGARTALA

computer architecture and organization
Lab 2021

Submission By

NAME : MD WASIF

Enrollment no: 19UICS002

Reg : 1910910

Sem & Year: 5TH sem & 2021

submitted to:-

Prof. Kishore Kumar Dhar

Prof. Ardhendu Gupta

ACKNOWLEDGEMENT

I would like to express my special thanks to my faculty members Kishore Kumar Dhar(Assistant Professor,UG Associate Coordinator) & Ardhendu Gupta(LAB/Technical Assistant) ,who gave me the golden opportunity to do this wonderful CAO lab which also helped me in doing a lot of Research and I came to know about so many new things and valuable guidance and feedback has helped me in completing and learning these a lot.

I would again like to thank my teacher for putting efforts even during this pandemic and found out a way to let us learn new things. This could not be a beneficial time without the guidelines and help.

MD WASIF

INDEX

Experiment number	Name of experiment	Date
1	Addition using registers/memory	25 august
2	Addition & subtraction of two 8 bit numbers	8 september
3	Subtraction & 2'complement of 8-bit numbers	15 september
4	Multiplication of 2 & 3 8-bit numbers	22 september
5	Division of two numbers & largest/smallest number in an array	27 october
6	Searching for a number & sum of series in an array.	3 November
7	To store array elements in descending & ascending order.	10 November
8	Sum of all even and odd numbers in an array	17 November

EXPERIMENT-01

Write a program for Addition of two 8-bit numbers.

AIM:- To perform addition of two 8 bit numbers using 8085.

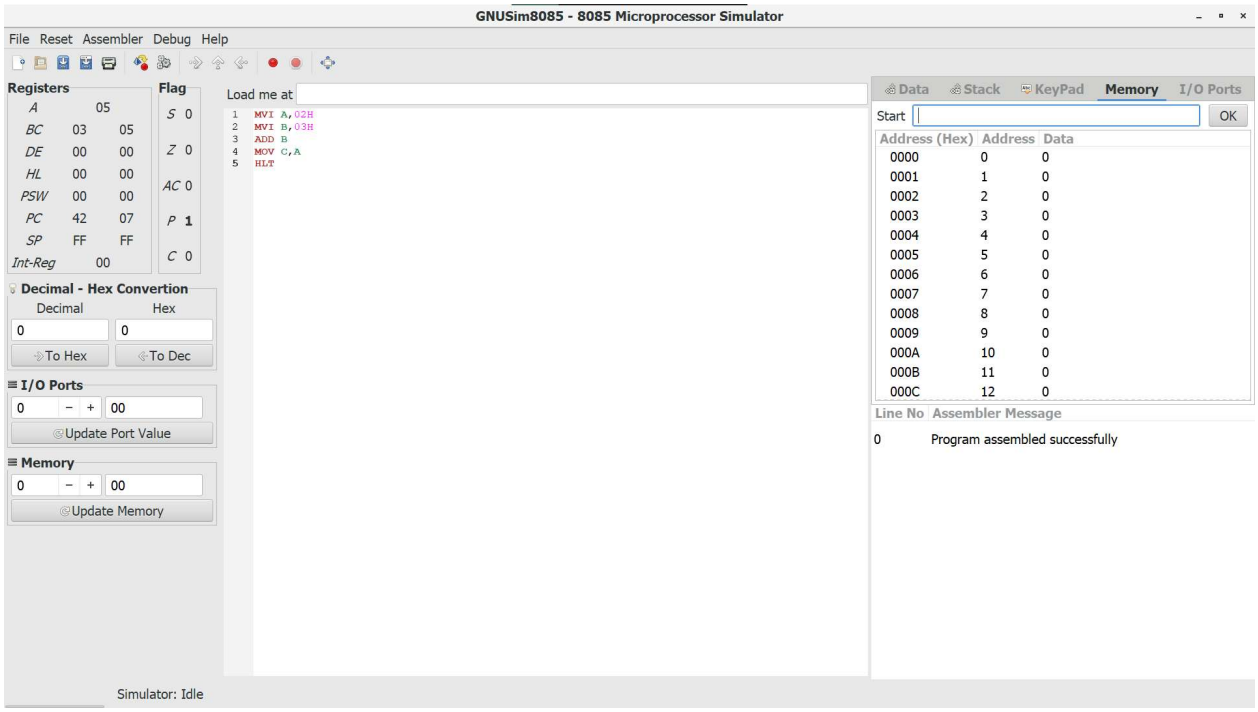
(A) Addition using register and storing result in register.

Algorithm :-

1. Initialize the first register.
2. Initialize the second register.
3. Add the value of the second register to the first.
4. Store the value in the third register.
5. Terminate the program.

Program :-

Memory Address	Op code	Operand	Hex code	comments
C000	MVI	A,02H	3E	Initialize register A by 2.
C001			02	
C002	MVI	B,03H	06	Initialize register B by 3.
C003			01	
C004	ADD	B	80	Add value of B to A
C005	MOV	C,A	4F	Store result in C
C006	HLT		76	terminate



Observation:-

	A	B	C
Input:	02H	03H	output : 05H

Result:

Thus, the program for addition of two 8-bit numbers is executed.

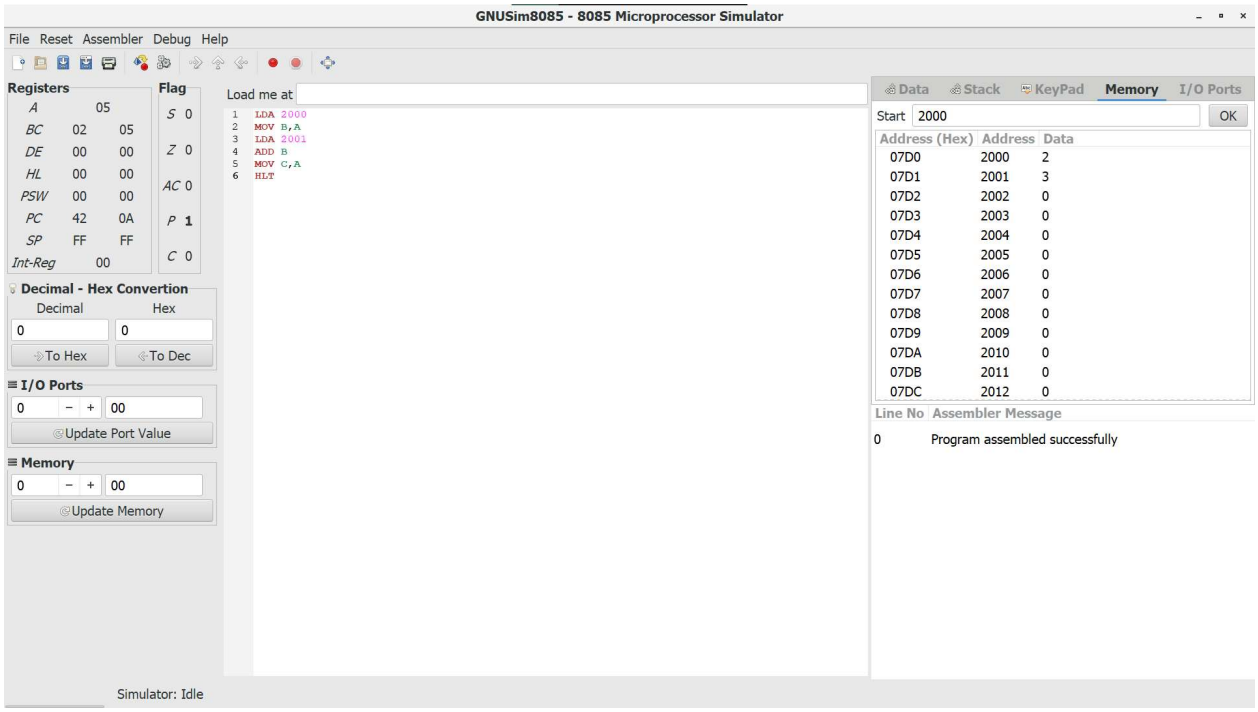
(B) Addition of number stored in memory and storing result in register.

Algorithm :-

1. **Load the first number to the accumulator through memory address 2000.**
2. **Move the content of the accumulator to register B.**
3. **Load the second number to the accumulator through memory address 2001.**
4. **Add the content of the accumulator and register B and the result will be stored at the accumulator.**
5. **Store the result from the accumulator to the register C.**
6. **Terminate the program.**

Program :-

Memory Address	Op code	Operand	comments
2000	LDA	2000	Accessing value from memory location 2000.
2003	MOV	B,A	Move value to B from accumulator
2004	LDA	2001	Accessing value from memory location 2001.
2007	ADD	B	Add value of B to Accumulator
	MOV	C,A	Store result in C
	HLT		Terminate



Observation :-

Memory Addresses:-

Input 2000 : 02

2001 : 03

Registers :-

Output	A	B	C
	5	2	5

Result :-

Thus the program to add 2 8-bit numbers is executed when we have to take input from memory and store answers to registers.

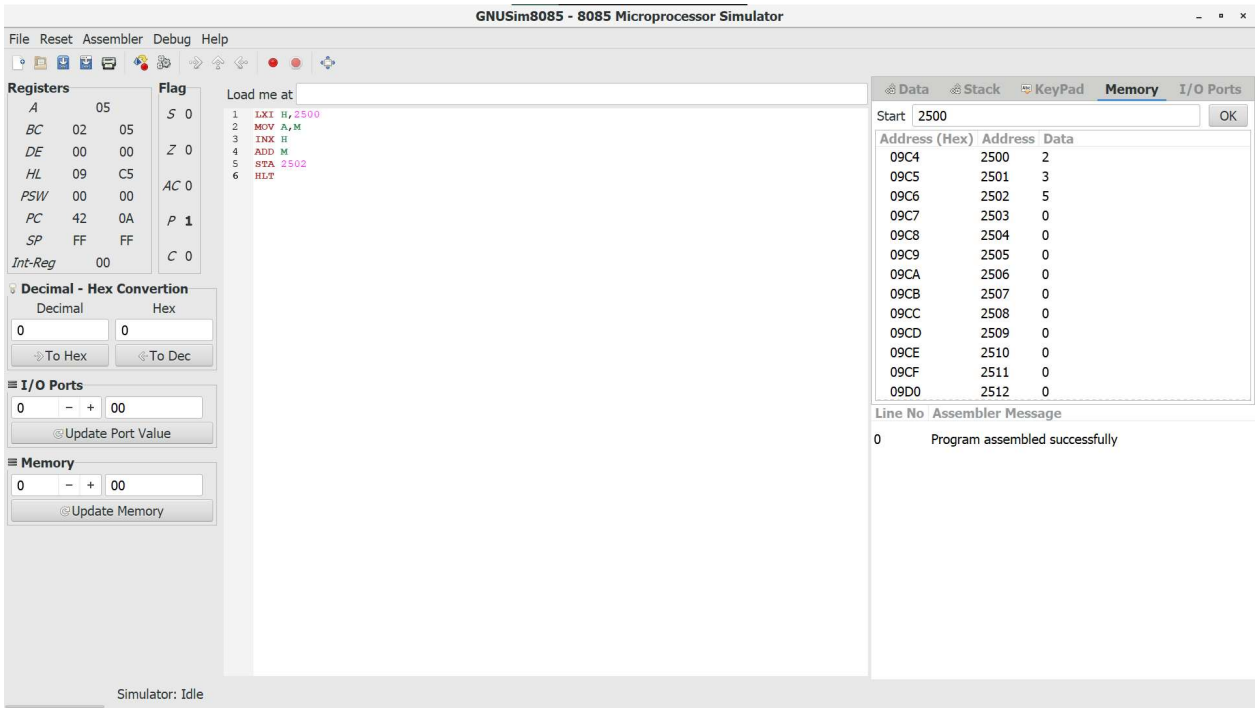
(C) Addition of number stored in memory and storing result in register.

Algorithm :-

1.Load the first number to the accumulator through memory address 2500.
2. Move the content of memory to the accumulator.
3. Increment H-L pair.
4. Add the content of the accumulator and next memory content.
5. Store the result in memory address 2502.
6. Terminate the program.

Program :-

Memory Address	Op code	Operand	comments
2000	LXI	H,2500	Accessing value from memory location 2500.
2003	MOV	A,M	Move value to accumulator from memory.
2004	INX	H	Increase H-L pair.
2007	ADD	M	(A)←(A)+M
2008	STA	2502	Store result in memory location 2502
	HLT		Terminate



Observation :-

Memory Addresses:-

Input 2500 : 02
2501 : 03
Output 2502 : 05

Result :-

Thus the program to add 2 8-bit numbers is executed when we have to take input from memory and store answers to memory.

EXPERIMENT-02

AIM:- Addition of two 8 bit numbers(With Carry).

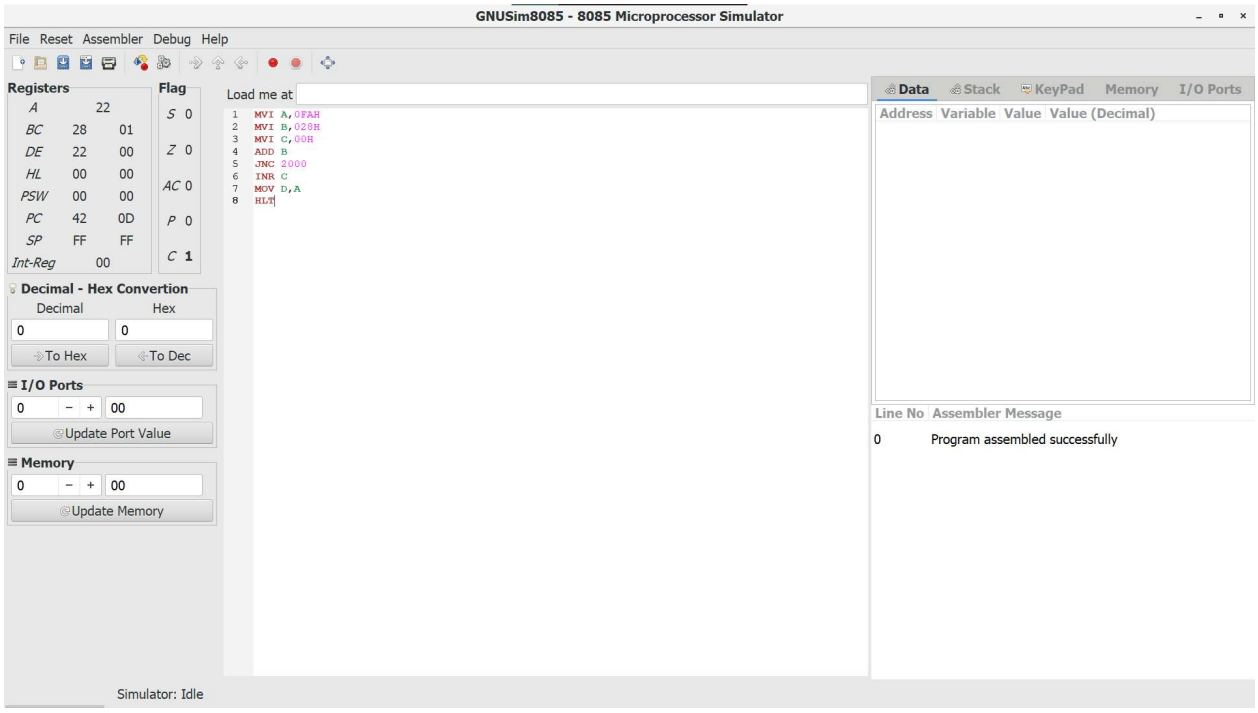
(A) Addition using register and storing result in register.

Algorithm :-

1. Initialize the first register.
2. Initialize the second register.
3. Initialize carry register with 0;
4. Add the value of the second register to the first.
5. Jump to location 2000 if carry not found else
6. Increment the carry register value.
7. Move the value to the fourth register from the accumulator.
8. Terminate the program.

Program :-

Memory Address	Op code	Operand	comments
2000	MVI	A,0FAH	Initialize register A by FA.
2001			
2002	MVI	B,028H	Initialize register B by 28.
2003			
2004	MVI	C	Initialize carry register by 0;
2006	ADD	B	Add B with accumulator value.
2007	JNC	2008	Jump to location 2008 if carry not found.
2009	INR	C	Increment Carry by 1
200A	MOV	D,A	Store value of accumulator to register D.
200B	HLT		Terminate the program.




```
MVI A,0FAH
MVI B,028H
MVI C,00H
ADD B
JNC 2000
INR C
MOV D,A
HLT
```

Observation:-

	A	B		C	D
Input :	0FAH	028H	output :	01H	22H

Result: -

Thus, the program for addition of two 8-bit numbers with carry is executed when input is taken from register and result and carry is stored in register.

(B) Addition of number with carry stored in memory and storing result in register.

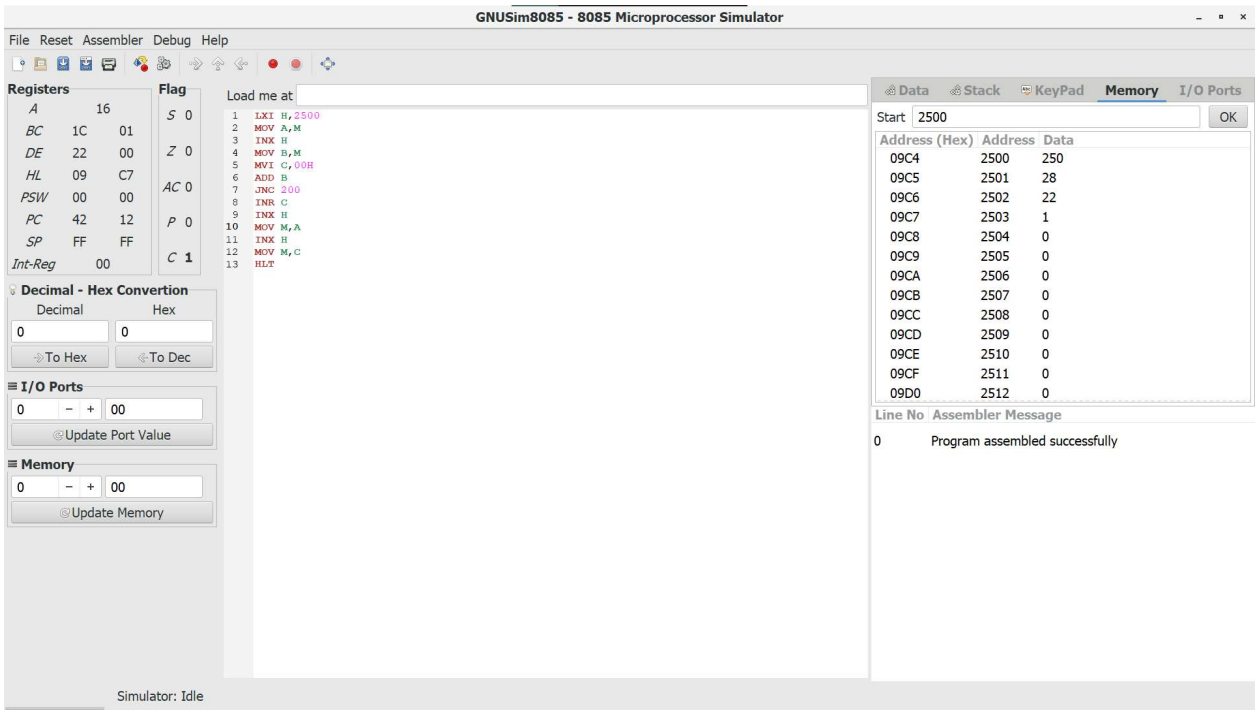
Algorithm :-

1. Load the first number to the accumulator through memory address 2500.
2. Move the content of memory to the accumulator.
3. Increment H-L pair.
4. Move memory content to register B.
5. Initialize carry register with 00H.
6. Add the content of the accumulator and register B content.
7. Jump to location 2000 if carry is not found.
8. If carry found, increase carry register by 1.
9. Increase H-L pair.
10. Store value of accumulator (sum) to memory.
11. Increment H-L pair.
12. Store value of carry to memory.
13. Terminate the program.

Program :-

Memory Address	OP Code	Operand	Comment
2000 2001 2002	LXI	H,2000	Load H-L pair with Address 2500
2003	MOV	A,M	Move memory value to reg A
2004	INX	H	Increment H-L pair
2005	MOV	B,M	Move memory value to reg B
2006 2007	MVI	C,00H	Initialize reg C with 0.
2008	ADD	B	Add B with A

2009 200A 200B	JNC	200DH	Jump to address 200D if there is no carry
200C	INR	C	Increment Reg C
200D	INX	H	Increase H-L pair
200E	MOV	M,A	Store value to memory(2502) from reg A
200F	INX	H	Increase H-L pair
2010	MOV	M,C	Store value to memory (2503)from reg C
2011	HLT		Terminate the program



Observation :- Memory Addresses:-

Input 2500 : 250 **Output 2502 : 22**
2501 : 28 **2503 : 01**

Result :- Thus the program to add 2 8-bit numbers with carry is executed when we have to take input from memory and store answers to memory.

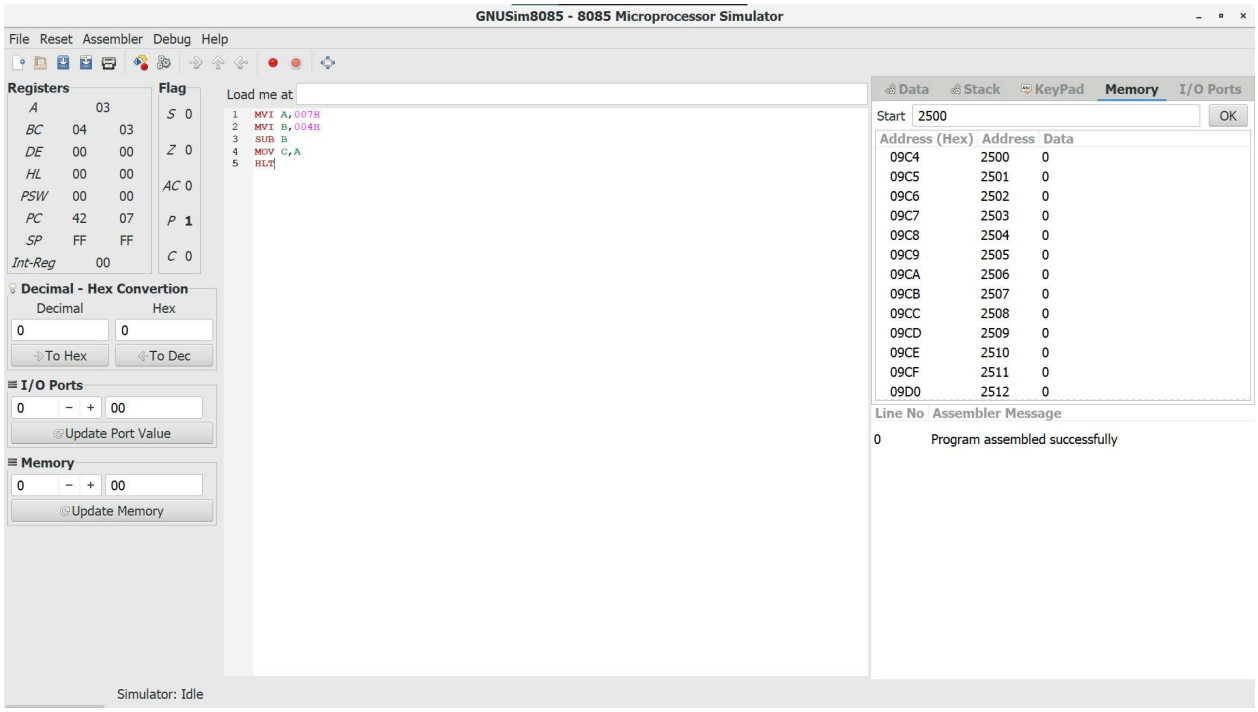
(C) Write a program to subtract two 8-bit numbers using Register.

Algorithm :-

(i) Initialize first register with first operand.
(ii) Initialize Second register with second operand.
(iii) Subtract second register from first.
(iv) Move the result to third register.
(v) Terminate the program.

Program :-

Memory Address	OP Code	Operand	Comment
2000 2001	MVI	A,007H	Initialize register A with 07H
2002 2003	MVI	B,004H	Initialize register B with 04H
2004	SUB	B	Subtract B from A
2005	MOV	C,A	Store result in register C
2006	HLT		Terminate the program.



Observation :-

Input :

A	B
007H	004H

Output :

C
003H

Result :- **Thus, the program for subtraction of two 8-bit numbers without borrow is executed when input is taken from register and result stored in register.**

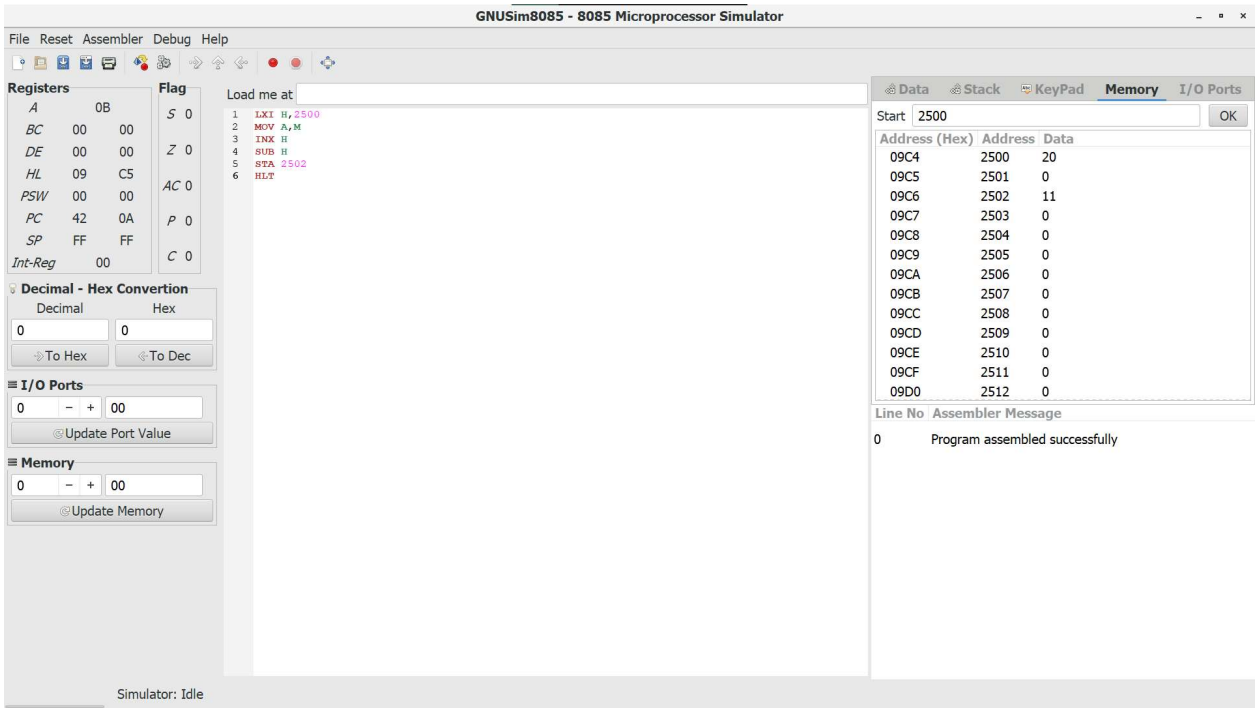
(D) Write a program to subtract two 8-bit numbers using Memory location.

Algorithm :-

(i) Load H-L pair with memory address 2500.
(ii) Move memory value to register A.
(iii) Increase H-L pair.
(iv) Subtract memory value from register A.
(v) Store the accumulator value to memory address 2502.
(vi) Terminate the program.

Program :-

Memory Address	OP Code	Operand	Comment
2000 2001 2002	LXI	H,2500	Load H-L pair with memory address 2500
2003	MOV	A,M	Move memory value to register A
2004	INX	H	Increment H-L pair.
2005	SUB	M	Subtract memory value from reg.A
2006	STA	2502	Store accumulator value to memory address 2502.
2007	HLT		Terminate the program



Observation :-

Input :

Memory Address : 2500 2501

20 9

Output : 2503

11

Result :- Thus, the program for subtraction of two 8-bit numbers without borrowing is executed when input is taken from memory and result is stored in memory .

EXPERIMENT-03

1. Subtraction of two 8 bit numbers (using borrow)

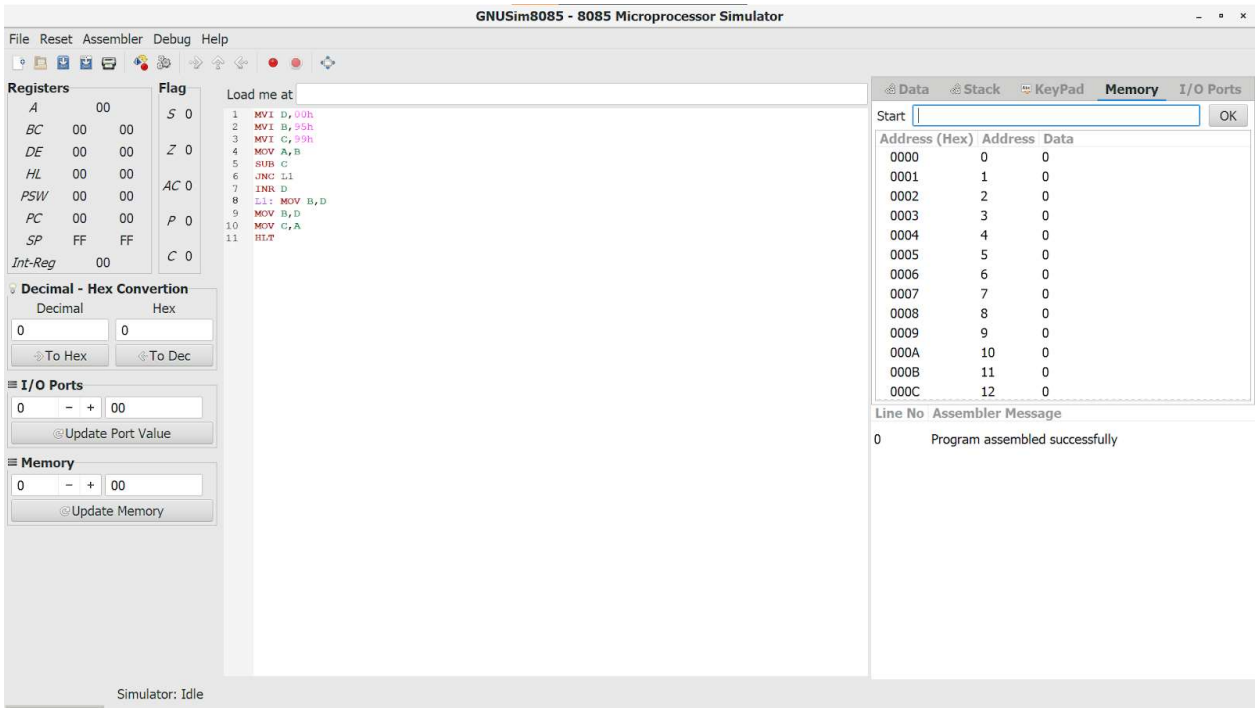
Aim : Subtract two 8-bit numbers with borrow

Algorithm:

1) **Initialize a register(say D) to 0 to store borrow**
2) **Load the register on memory location with data**
3) **Move the first data to the accumulator**
4) **Check the borrowings using JNC . if the borrow is generated then increment the value of D register.**
5) **Move the result to desired register/location**
6) **Terminate**

PROGRAM:-

```
Address    Mnemonics    Operands
2000         LXI          H,3000H
2003         MOV          A, M
2004         INX          H
2005         MOV          B, M
2006         MVI          C, 00H
2008         SUB          B
2009         JNC          200D
200C         INR          c
200D         INX          H
200E         MOV          M, A
200F         INX          H
2010         MOV          M, C
2011         HLT
```



Observation :

Input : 95 B **Output : 01 B -> Carry**
99 C **FC C -> Result**
00D **01 D**

Result:
The result of subtraction using borrow can be seen in the BC register pair. Thus, the program executed correctly.

2. Write a program to find 2'complement of a 8-bit numbers

Aim: find 2'complement of a 8-bit numbers

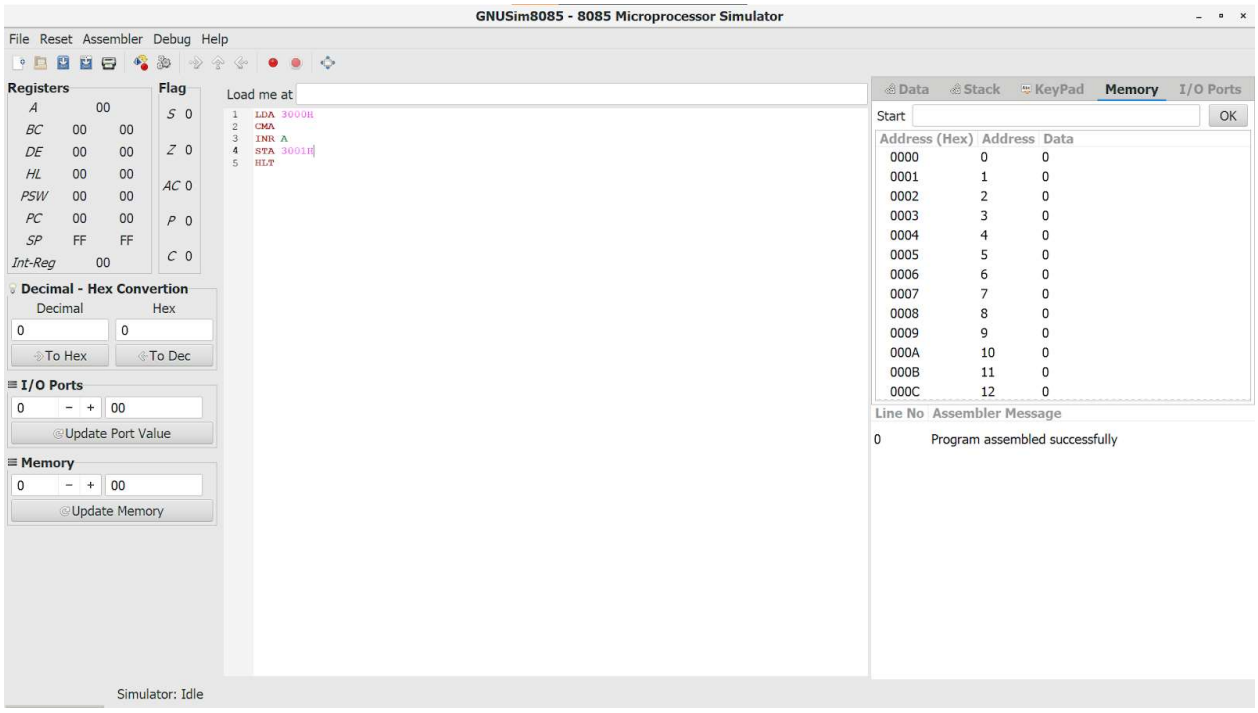
Algorithm:

1) **Load the data from memory 3000 into A (accumulator)**
2) **Complement content of accumulator**
3) **Store content of accumulator in memory 3001 (1's complement**
4) **Add 01 to Accumulator content**

5) **Store content of accumulator in memory 3002 (2's complement)**
6) **Terminate**

Program:-

Memory	Mnemonics	Operands	
2000	LDA	[3000]	[A] <- [3000]
2003	CMA		[A] <- [A^]
2004	STA	[3001]	1's complement
2007	ADI	01	[A] <- [A] + 01
2009	STA	[3002]	2's complement
200C	HLT		Stop



Observation:

Input :	output :
3000H : 85H	3001H : 7BH

Result:
Thus, the program to find the 2's complement of a 8-bit number was successfully executed and the correct result was given.

3. Subtraction of two 8-bit numbers using 2'complement method using 8085

Aim: Subtraction of two 8-bit numbers using 2'complement

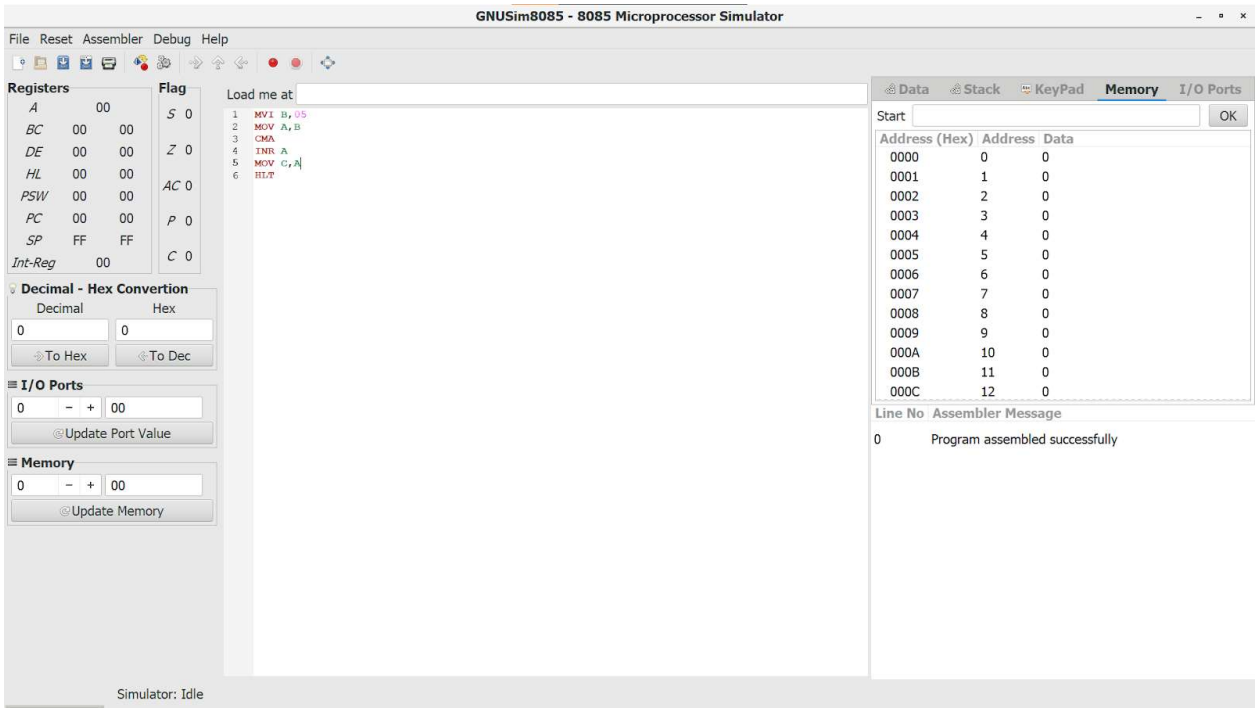
Algorithm:

1) **Load the first data into a register .**
2) **Load the second data value into the accumulator.**
3) **Compliment the value and then add 1 in the result**
4) **Add the first data value to the accumulator**
5) **Move the result to the desired memory location**
6) **Terminate**

Program:-

Address	HEX Codes	Labels	Mnemonics	Comments
F000	0E,00		MVIC,00H	Clear C register
F002	21,00, 80		LXIH,8000H	Load initial address to get operand
F005	7E		MOVA, M	Load Acc with the memory element

F006	23		INX H	Point to next location
F007	46		MOVB, M	Load B with the second operand
F008	90		SUB B	Subtract B from A
F009	D2,0D, F0		JNC STORE	When CY = 0, go to STORE
F00C	0C		INR C	Increase C by 1
F00D	21,50, 80	STORE	LXIH,8050H	Load the destination address
F010	77		MOVM, A	Store the result
F011	23		INX H	Point to next location
F012	71		MOVM, C	Store the borrow
F013	76		HLT	Terminate the program



Observation:

Input:	**output:**
95 B	**95 B**
99 C	**99 C**
XX D	**FC D**

Result:

Thus , the program to find Subtraction of two 8-bit numbers using 2'complement was successfully executed correctly.

4. Write a program to multiply 8-bit numbers using 8085

Aim: to find multiple of two 8-bit numbers

Algorithm:

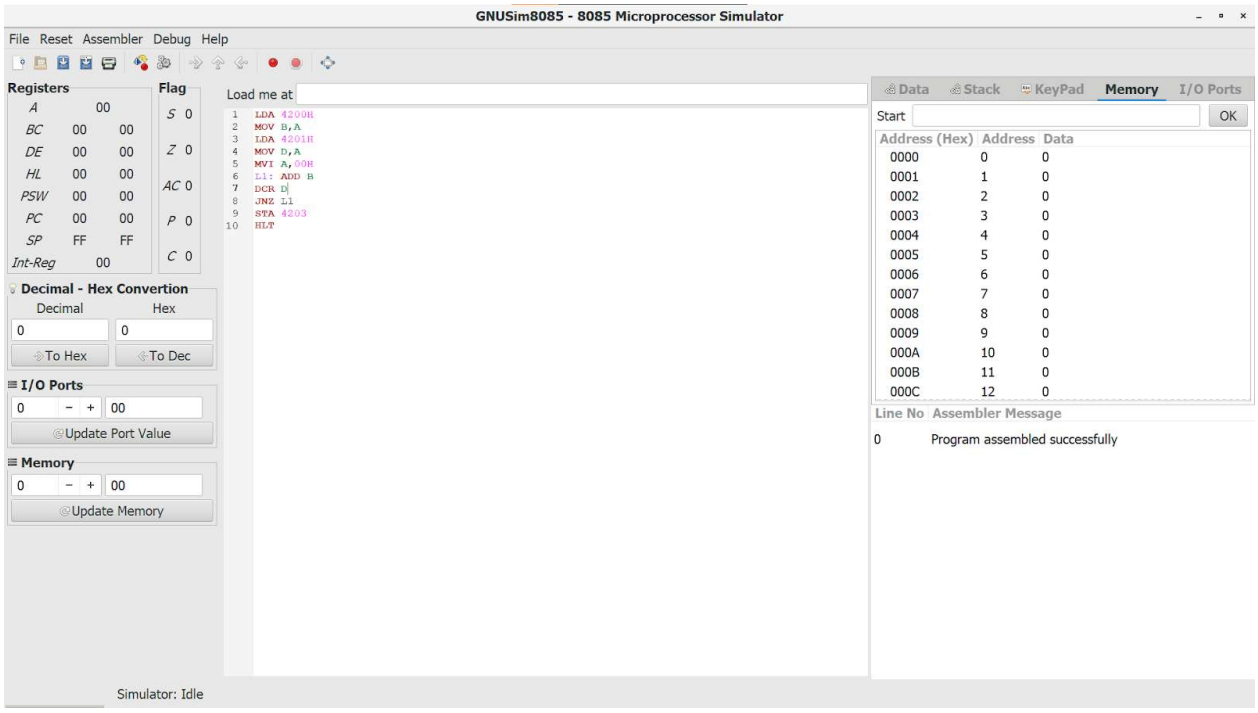
1) **Load the two numbers in separate locations.**
2) **Initialize contents of the accumulator to 0.**
3) **Add either of the number in accumulator the same time, decrement the other the other number**
4) **Keep repeating step 3 until the decremented number becomes 0.**
5) **Move the result from the accumulator to the desired location.**

6) Terminate the program.

Program:-

Address	HEX Codes	Labels	Mnemonics	Comments
F000	21, 00, 80		LXI H,8000H	Load first operand address
F003	46		MOV B, M	Store first operand to B
F004	23		INX H	Increase HL pair
F005	AF		XRA A	Clear accumulator
F006	4F		MOV C, A	Store 00H at register C
F007	86	LOOP	ADD M	Add memory element with Acc
F008	D2, 0C, F0		JNC SKIP	When Carry flag is 0, skip next task
F00B	0C		INR C	Increase C when carry is 1
F00C	05	SKIP	DCR B	Decrease B register
F00D	C2, 07, F0		JNZ LOOP	Jump to loop when Z flag is not 1
F010	21, 50, 80		LXI H,8050H	Load Destination address

F013	71		MOV M, C	Store C register content into memory
F014	23		INX H	Increase HL Pair
F015	77		MOV M, A	Store Acc content to memory
F016	76		HLT	Terminate the program



Observation:

Input:

A	B	C
OO	O2	O3
	D	
	XX	

Output:

A	B	C
06	02	03
	D	
	06	

Result:
Thus , the program to find multiple of two 8-bit numbers was successfully executed correctly.

EXPERIMENT-04

1. Write a program in 8085 for Multiplication of two 8-bit numbers (with carry)

Aim : Multiplication of two 8-bit numbers(with carry)

Algorithm:

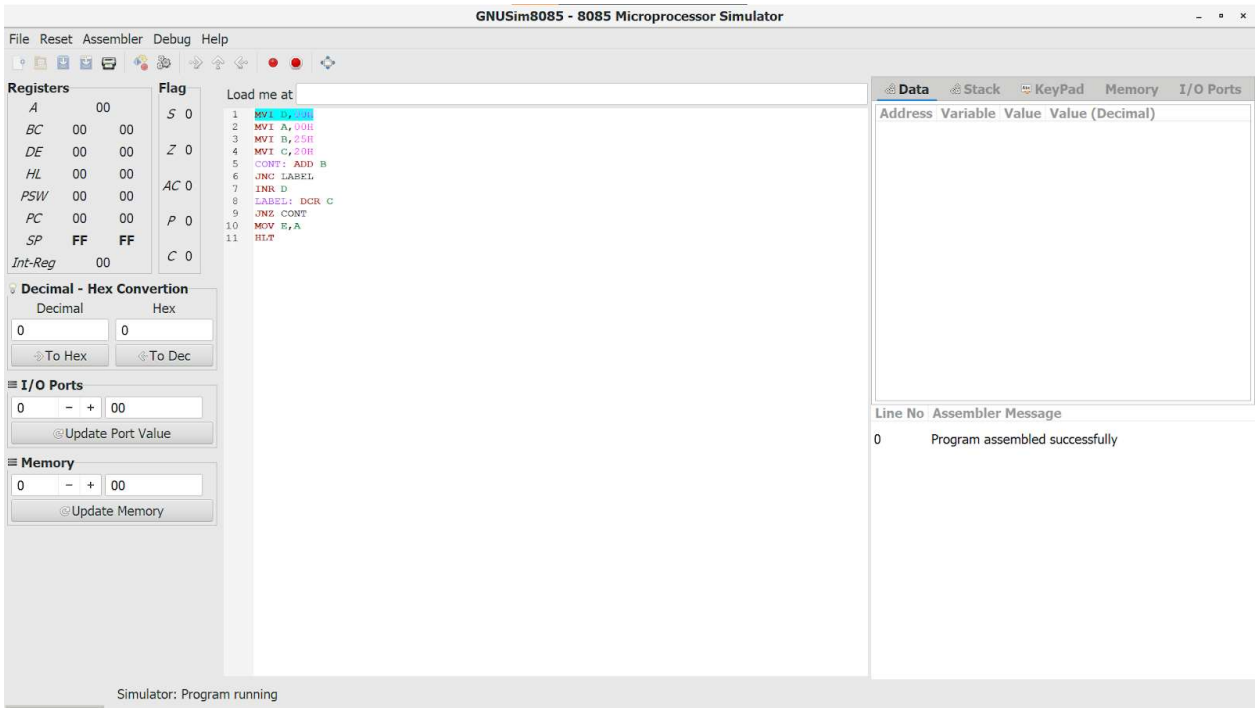
1) **Initialize the accumulator with 0 and register with 0**
2) **Load the numbers to be multiplied in separate registers.**
3) **Add either of the numbers in the accumulator.**
4) **Check for any carry, if carry is generated , add/increment content of register D.**
5) **If no carry, then decrement the other number.**
6) **Keep repeating step 3-5 unless the number being decremented becomes zero.**
7) **Move the result to desired register/location**
8) **Terminate**

Program:-

Address	HEX Codes	Labels	Mnemonics	Comments
F000	21, 00, 80		LXI H,8000H	Load first operand address

F003	46		MOV B, M	Store first operand to B
F004	23		INX H	Increase HL pair
F005	AF		XRA A	Clear accumulator
F006	4F		MOV C, A	Store 00H at register C
F007	86	LOOP	ADD M	Add memory element with Acc
F008	D2, 0C, F0		JNC SKIP	When Carry flag is 0, skip next task
F00B	0C		INR C	Increase C when carry is 1
F00C	05	SKIP	DCR B	Decrease B register
F00D	C2, 07, F0		JNZ LOOP	Jump to loop when Z flag is not 1
F010	21, 50, 80		LXI H,8050H	Load Destination address
F013	71		MOV M, C	Store C register content into memory
F014	23		INX H	Increase HL Pair
F015	77		MOV M, A	Store Acc content to memory

F016	76		HLT	Terminate the program



Observation :

FF (4150)

Input:

FF (4151)

01 (4152)

Output:

FE (4153)

Result:

Thus, the multiplication of two 8-bit numbers without carry was executed.

2. Write a program in 8085 for Multiplication of 8-bit numbers (without carry)

Aim: Multiply three 8-bits numbers(without carry)

Algorithm:

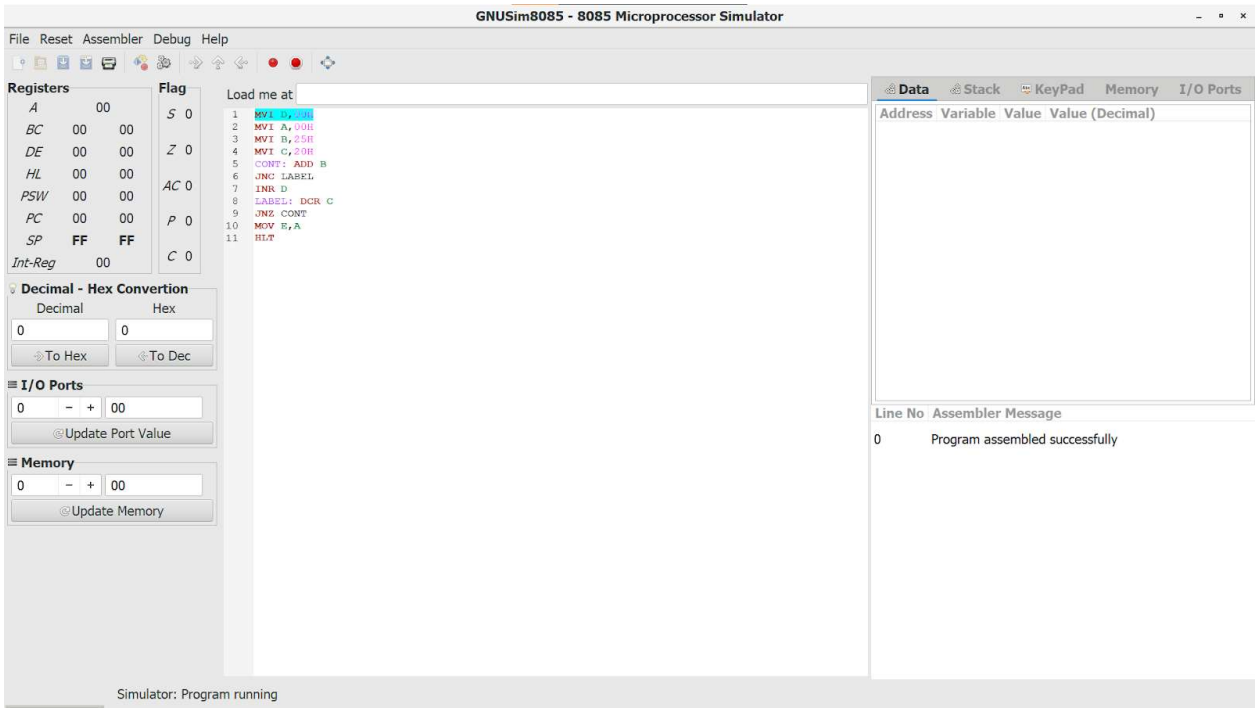
1) **Initialize accumulator with 0.**
2) **Multiply the first two data values like the first case.**

3) **Load the third data value in another register.**
4) **Decrement the value of the third data at the same time keep adding the contents of the accumulator with itself (A)<-(A)+(A).**
5) **Repeat step 4 until the third data value becomes zero.**
6) **Move the result to the desired location.**
7) **Terminate the program.**

Program:-

Address	HEX Codes	Labels	Mnemonics	Comments
F000	21, 00, 80		LXI H,8000H	Load first operand address
F003	46		MOV B, M	Store first operand to B
F004	23		INX H	Increase HL pair
F005	AF		XRA A	Clear accumulator
F006	4F		MOV C, A	Store 00H at register C
F007	86	LOOP	ADD M	Add memory element with Acc
F008	D2, 0C, F0		JNC SKIP	When Carry flag is 0, skip next task
F00B	0C		INR C	Increase C when carry is 1
F00C	05	SKIP	DCR B	Decrease B register

F00D	C2, 07, F0		JNZ LOOP	Jump to loop when Z flag is not 1
F010	21, 50, 80		LXI H,8050H	Load Destination address
F013	71		MOV M, C	Store C register content into memory
F014	23		INX H	Increase HL Pair
F015	77		MOV M, A	Store Acc content to memory
F016	76		HLT	Terminate the program



Observation: -

Input :	B	C	D	output :	A	B	C	D	E
	02	02	02		08	00	02	00	08

Result:
Thus, multiplication of three 8 bit numbers was executed.

3. Write a program in 8085 for Division of two numbers using registers.

Aim: Division of 2 numbers using registers.

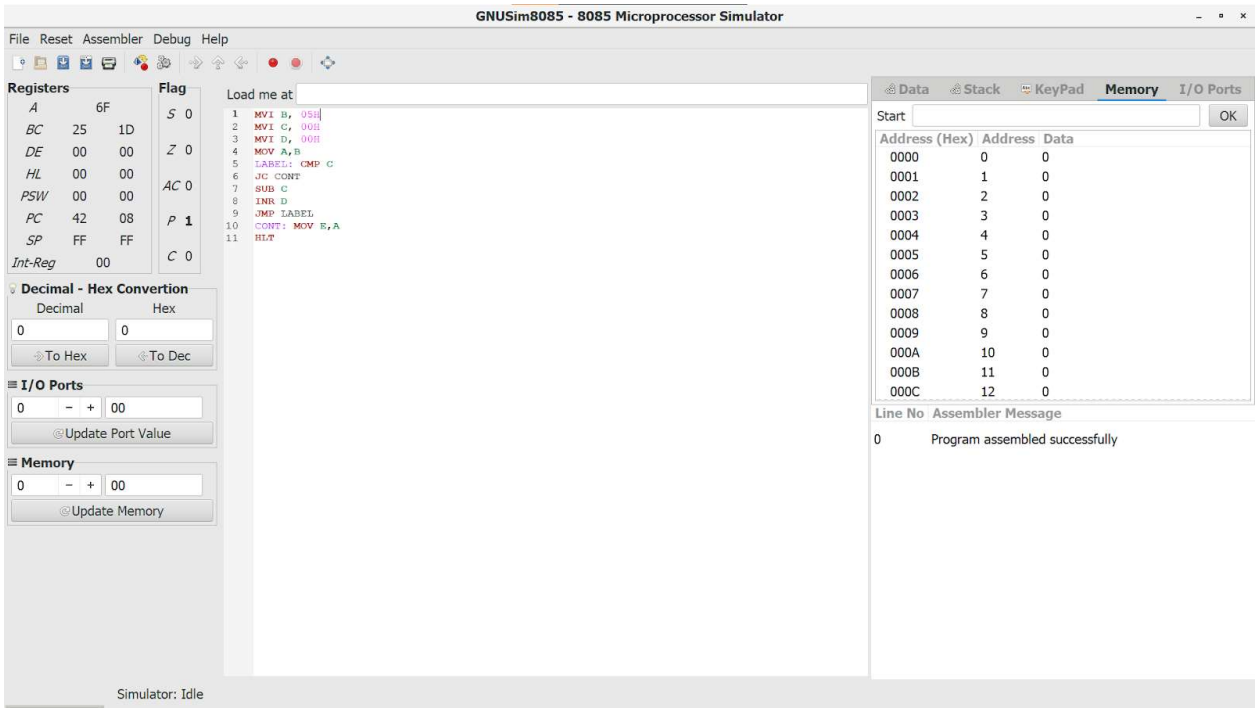
Algorithm:

1) **Load the two numbers in separate registers.**
2) **Initialize another register to 0, which will store the result.**
3) **Move the first data to the accumulator.**
4) **Compare it with the other data.**
5) **If no carry is generated, subtract the second data value and at the same time increment register that will store the result.**
6) **Repeat the step 5 until a carry/borrow is generated.**
7) **The result is generated.**
8) **Terminate the program.**

Program:-

```
LXI     H, 3050
MOV     B, M
MVI     D, 00
INX     H
MOV     A, M
*** CMP B
JC **
SUB     B
INR     C
JMP ***
** STA 3052
MOV     A, C
```

```
    STA     3053
    HLT
```



Observation:

Input:

F (4150)

FF (4251)

Output:

01 (4152) ---- Remainder

FE (4153) ---- Quotient

Result:

Thus , the division of two 8 -bit numbers was executed successfully.

EXPERIMENT-05

Q.No 1. : - Write a program to divide 8-bit numbers using memory location in 8085 simulator.

AIM:- To divide 8-bit numbers using memory location.

.

Algorithm :-

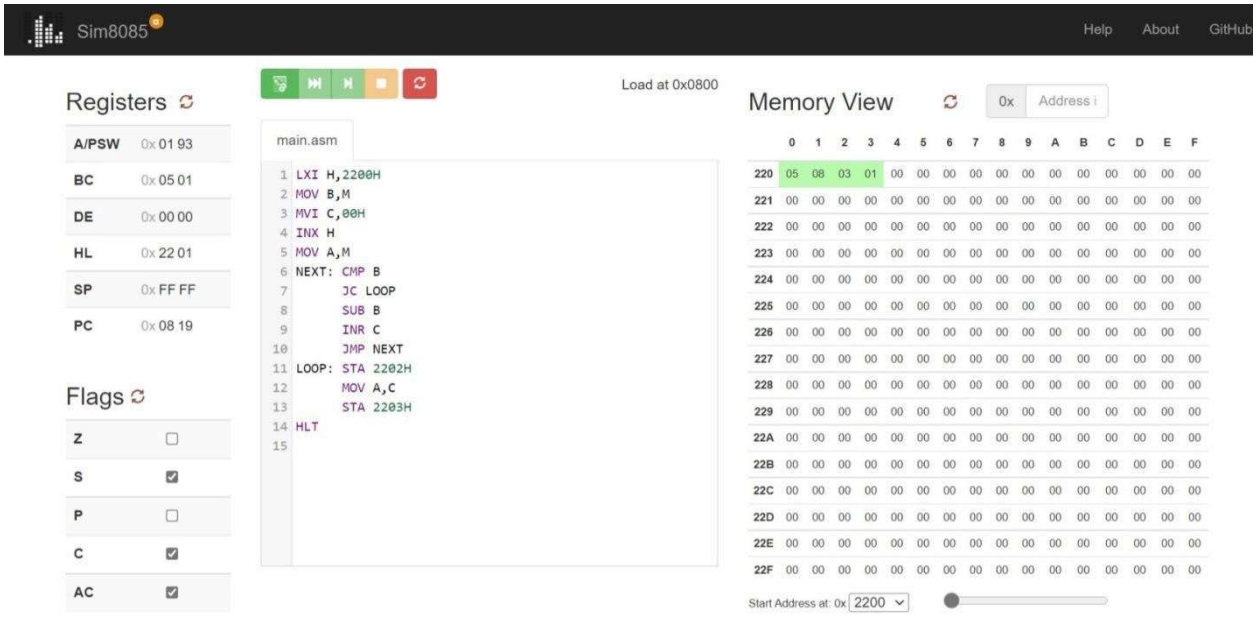
1. Start the program by loading HL register pair with address of memory location.
2. Move the data to a B register.
3. Get the second data and load into Accumulator.
4. Compare the two numbers to check for carry.
5. Subtract the two numbers.
6. Increment the value of carry .
7. Check whether repeated subtraction is over and store the value of product and carry in memory location.
8. Terminate the program.

Program :-

Memory Address	Opcode	Operand	Comments
C000 C001 C002	LXI	H,2200H	Load HL register pair with address of memory location 2200.
C003	MOV	B,M	Move the data to a B register.
C004 C005	MVI	C,00H	Initialize register c with 0,
C006	INX	H	Increment H-L pair.
C007	MOV	A,M	Get the second data and load into Accumulator.
C008 C009 C001	NEXT: CMP JC LOOP	B	Compare the two numbers to check for carry. If carry not found jump to loop.
C00B	SUB B	B	Subtract the two numbers
C00C	INR	C	Increment the value of carry .
C00D	JMP NEXT		Check whether repeated subtraction is over.
C010	STA	2202H	Store the value of product in memory location.
C011	MOV	A,C	Move value of carry to accumulator.
C012	STA	2203H	Store the value of product in memory location.

C013	HLT		Terminate the program.

Screenshot :-



Observation :-

Memory Locations:-

Input:- 2200 5H

2201 8H

Output :- 2202 03H

2200 01H

Result: -

Thus, the program for division of two 8-bit numbers is executed when input is taken from memory location and result is stored in memory locations.

Q.No 2 : - Write a program to find the largest number in an array in 8085.

AIM:- To find the largest number in an array .

Algorithm :-

1. First store array elements to memory locations.

2. Initially, the counter is initialized with the size of an array.

3. Then, two numbers are moved to registers A and B and compared.

4. After comparison, the largest of two must be in the accumulator. If it is accumulator, already then in its the fine, otherwise it is moved to the accumulator.

5.Counter is decremented and checked whether it has reached zero. If it has, the loop terminates otherwise, the next number is moved to register and compared.

6. Let us assume that the memory location counter. 2200H The stores next locations store the array to the memory.

7. Initially, H-L pair is loaded with the address of the counter and is moved to register C.

8. Then, H-L pair is incremented to point to the first number in the array.

5. The first number is moved from memory to accumulator and counter is decremented by one.

10. H-L pair is again incremented and the second number is moved to register B.

11. The two numbers are compared.

12. After comparison, if A>B, then CF = 0, and if A< B, then CF = 1.

13. Carry flag is checked for carry. If there is a carry, it means B is greater than A and it is moved to the accumulator.

14. The counter is decremented and checked whether it has become zero.

15. If it hasn't become zero, it means there are numbers left in the array. In this case, the control jumps back to increment the H-L pair and moves the next number to register B.

16. This process continues until counter becomes zero, i.e. all the numbers in the array are compared.

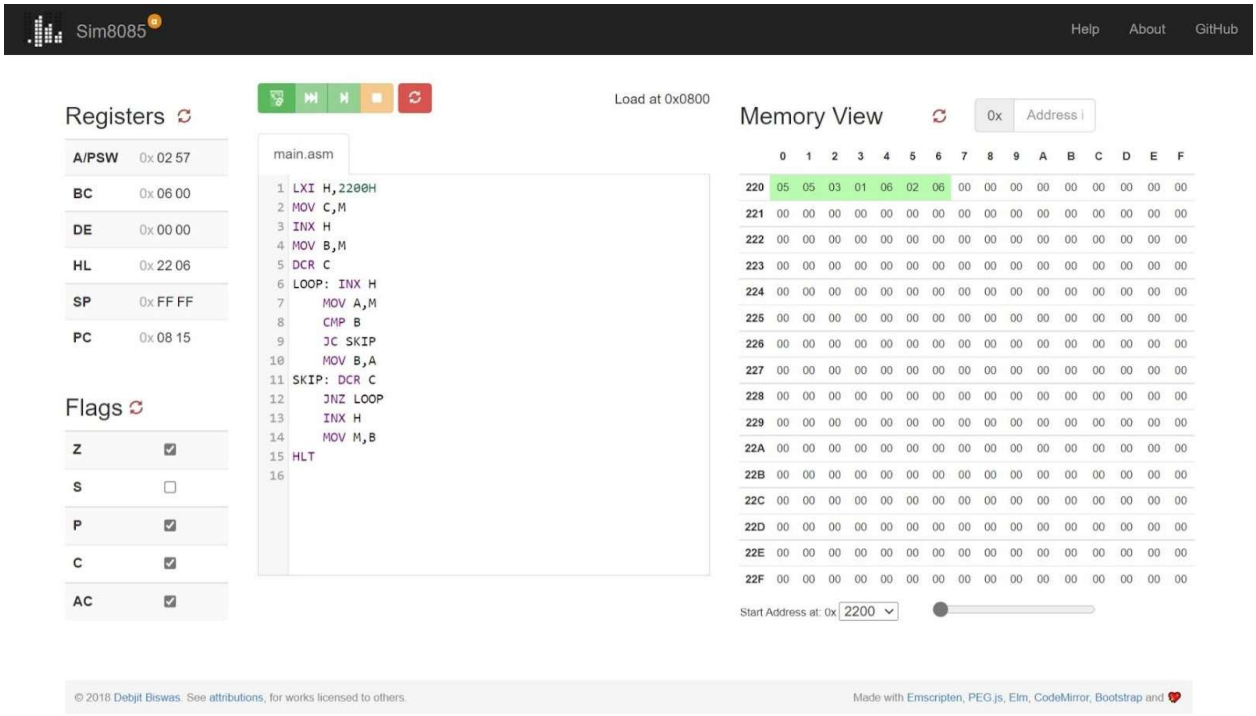
17. At last, H-L pair is incremented and the largest number is moved from accumulator to memory.

18. Terminate the program.

Program :-

Address	HEX Codes	Labels	Mnemonics	Comments
F000	21, 00, 22		LXI H,2200H	Point to get array size
F003	4E		MOV C, M	Get the size of array
F004	23		INX H	Point to actual array
F005	46		MOV B, M	Load the first number into B
F006	0D		DCR C	Decrease C
F007	23	LOOP	INX H	Point to next location
F008	7E		MOV A, M	Get the next number from memory to accumulator.
F009	B8		CMP B	Compare accumulator and B
F00A	DA, 0E, F0		JC SKIP	if B > A,then skip
F00D	47		MOV B, A	If CY is 0, update B
F00E	0D	SKIP	DCR C	Decrease C
F00F	C2, 07, F0		JNZ LOOP	When count is not 0, go to LOOP
F012	21, 06, 22		LXI H,2206H	Point to destination address
F015	70		MOV M, B	Store the minimum number
F016	76		HLT	Terminate the program

Screenshot :-



Observation :-

	Memory Location	Value
Inputs :-	2200H	05H(Counter)
	2201H	05H
	2202H	03H
	2203H	01H
	2204H	06H
	2205H	02H
Output:-	**2206H**	**06H(Largest element)**

Result :- Thus the program to find the largest number in an array of data was executed.

Q.No 3 : - Write a program to find the smallest number in an array in 8085.

AIM:- To find the smallest number in an array .

Algorithm :-

1. First store array elements to memory locations.

2. Initially, the counter is initialized with the size of an array.

3. Then, two numbers are moved to registers A and B and compared.

4. After comparison, the smallest of two must be in the accumulator. If it is accumulator, already then in its the fine, otherwise it is moved to the accumulator.

5.Counter is decremented and checked whether it has reached zero. If it has, the loop terminates otherwise, the next number is moved to register and compared.

6. Let us assume that the memory location counter. 2200H The stores next locations store the array to the memory.

7. Initially, H-L pair is loaded with the address of the counter and is moved to register C.

8. Then, H-L pair is incremented to point to the first number in the array.

6. The first number is moved from memory to accumulator and counter is decremented by one.

10. H-L pair is again incremented and the second number is moved to register B.

11. The two numbers are compared.

12. After comparison, if A>B, then CF = 0, and if A< B, then CF = 1.

13. Carry flag is checked for carry. If there is a carry, it means B is smaller than A and it is moved to the accumulator.

14. The counter is decremented and checked whether it has become zero.

15. If it hasn't become zero, it means there are numbers left in the array. In this case, the control jumps back to increment the H-L pair and moves the next number to register B.

16. This process continues until counter becomes zero, i.e. all the numbers in the array are compared.

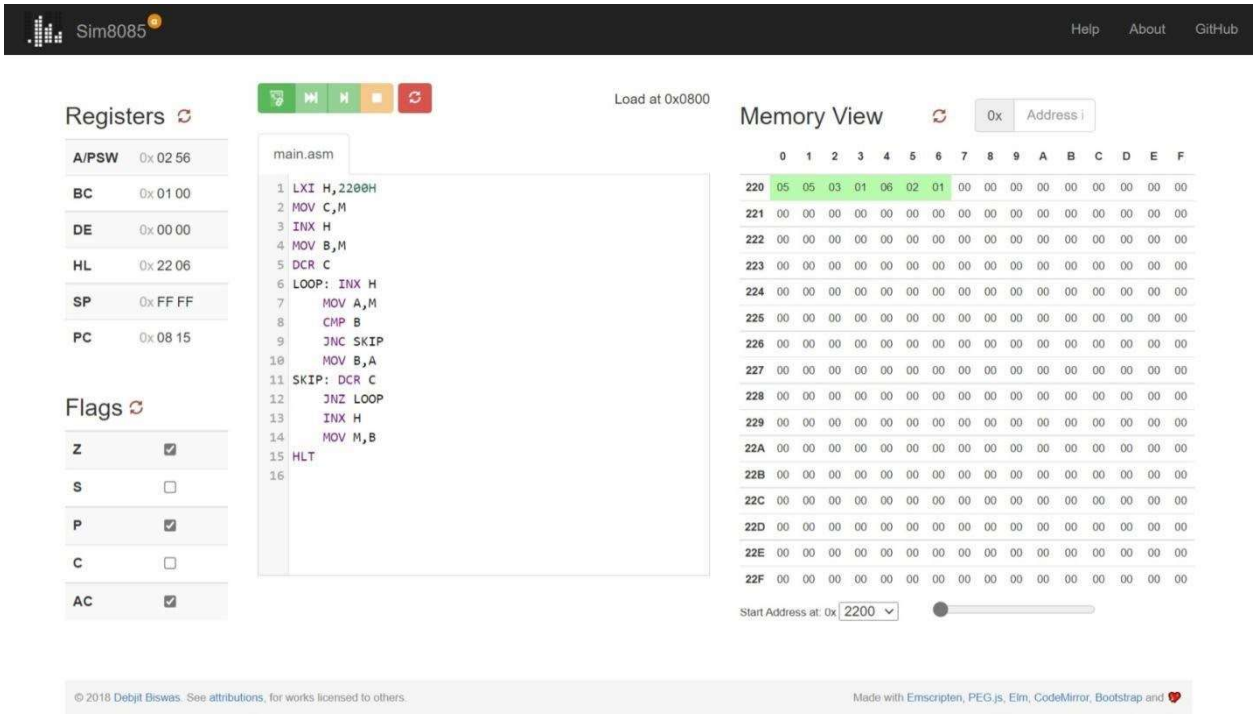
17. At last, H-L pair is incremented and the smallest number is moved from accumulator to memory.

18. Terminate the program.

Program :-

Address	HEX Codes	Labels	Mnemonics	Comments
F000	21, 00, 22		LXI H,2200H	Point to get array size
F003	4E		MOV C, M	Get the size of array
F004	23		INX H	Point to actual array
F005	46		MOV B, M	Load the first number into B
F006	0D		DCR C	Decrease C
F007	23	LOOP	INX H	Point to next location
F008	7E		MOV A, M	Get the next number from memory to accumulator.
F009	B8		CMP B	Compare accumulator and B
F00A	DA, 0E, F0		JNC SKIP	if A>=B,then skip
F00D	47		MOV B, A	If CY is 0, update B
F00E	0D	SKIP	DCR C	Decrease C
F00F	C2, 07, F0		JNZ LOOP	When count is not 0, go to LOOP
F012	21, 06, 22		LXI H,2206H	Point to destination address
F015	70		MOV M, B	Store the minimum number
F016	76		HLT	Terminate the program

Screenshot :-



Observation :-

	Memory Location	Value
Inputs :-	2200H	05H(Counter)
	2201H	05H
	2202H	03H
	2203H	01H
	2204H	06H
	2205H	02H
Output:-	**2206H**	**01H(Smallest element)**

Result :- Thus the program to find the smallest number in an array of data was executed.

EXPERIMENT-06

1 . Write a program to search an 8-bit number in an array.

Aim : To search a number in array of elements

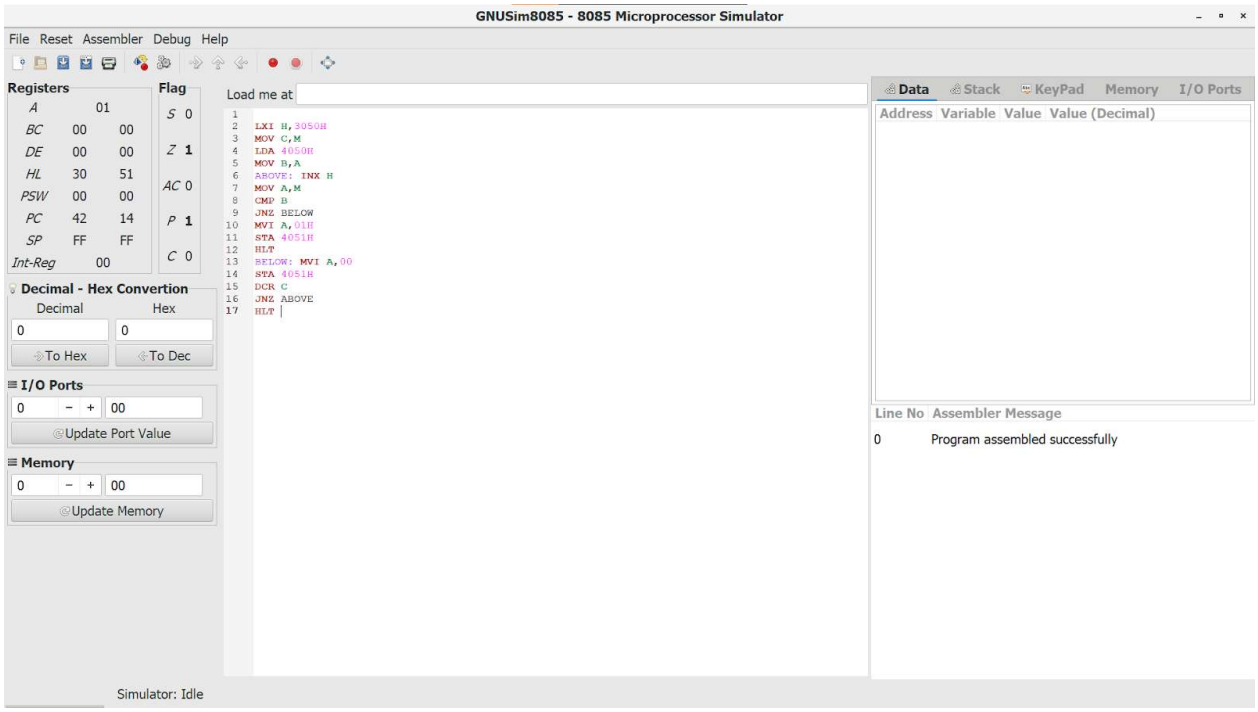
Algorithm:

1) **Load H-L pair to the memory address holding the size of the array**
2) **Move it to a register that will act as a counter .**
3) **Load the element to be searched to another register.**
4) **Increment H-L par to subsequent locations containing the array elements and move it to A.**
5) **Compare each of these elements with the element to be searched ,using CMP.**
6) **If the element is the same it will generate a zero flag else not. So, we check for zero flags.**
7) **If no zero flag is generated , we store value 01 in a desired location, which indicates that our element is found and we terminate the program.**

Program:-

Address	HEX Codes	Labels	Mnemonics	Comments
F000	21, 00, 80		LXI H,8000H	Point to get array size
F003	4E		MOV C, M	Get the size of array
F004	23		INX H	Point to next location
F005	46		MOV B, M	Get the key value to search
F006	78		MOV A, B	Take the key into acc
F007	23	LOOP	INX H	Point to next location

F008	BE		CMP M	Compare memory element with Acc
F009	CA, 19, F0		JZ FOUND	When Z flag is set, go to FOUND
F00C	0D		DCR C	Decrease C by1
F00D	C2, 07, F0		JNZ LOOP	When Count is not 0, jump to LOOP
F010	21, FF, FF		LXI H, FFFF	Load FFFFHinto HL pair
F013	22, 00, 90		SHLD 9000H	Store at9000H
F016	C3, 1C, F0		JMP DONE	Jump to DONE
F019	22, 00, 90	FOUND	SHLD 9000H	Store at9000H
F01C	76	DONE	HLT	Terminate the program



OBSERVATION:

Input:

0

02 3050

XX 3051

5	205
1	205 1
A6	205 2
B3	205 3
2	205 4
EF	205 5

0

Output:

02	3050
01	3051

RESULT:

Thus, the program for searching an element was done and executed.

Conclusion:-

Thus, the program for searching an element was done and executed successfully.

2. Write a program to find the sum of series in an array using 8085.

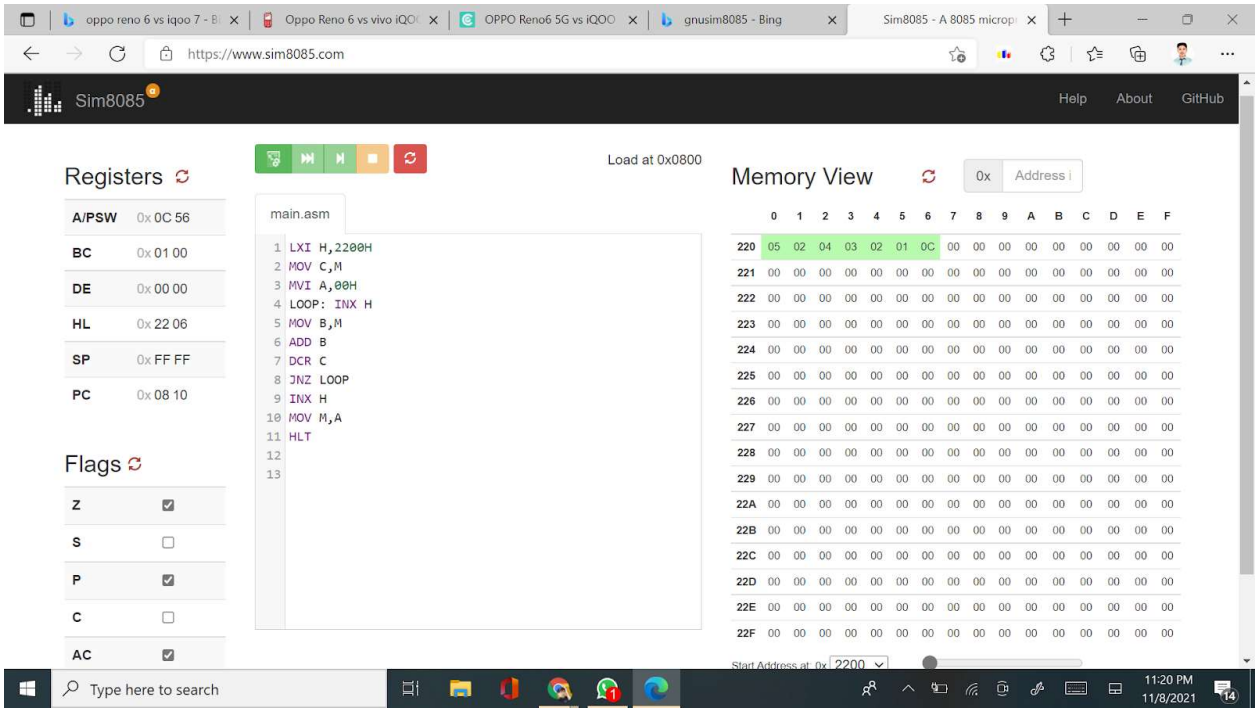
Aim: program to find sum of series in an array.

Algorithm :-

1. **Load H-L pair with address.**
2. **Move the counter from memory to register.**
3. **Initialize accumulator with 00H.**
4. **Add the first element with the content of the accumulator.**
5. **Increment H-L pair.**
6. **If carry occurs, increment C, else simply decrement B.**
7. **Repeat steps 5 to 7 until B becomes zero.**
8. **Move the result to the desired location.**
9. **Terminate the program.**

Memory	Mnemonics	Operands	Comment
2000	MVI	B, 00	[B] <- 00
2002	LXI	H, [3000]	[H-L] <- [3000]
2005	MOV	C, M	[C] <- [M]
2006	DCR	C	[C] <- [C] – 1
2007	INX	H	[H-L] <- [H-L] + 1
2008	MOV	A, M	[A] <- [M]
2009	INX	H	[H-L] <- [H-L] + 1
200A	ADD	M	[A] <- [A] + [M]
200B	JNC	200F	jump if no carry
200E	INR	B	[B] <- [B] + 1
200F	DCR	C	[C] <- [C] – 1
2010	JNZ	2009	jump if not zero
2013	STA	[4000]	result
2016	MOV	A, B	[A] <- [B]

2017	STA	[4001]	carry
201A	HLT		Stop



Observation:-

Input:

05	2050
06	2051
03	2052
04	2053
01	2054
02	2055

Output:

0F	3050
00	3051

Conclusion :-

Thus ,the program to find the sum of elements in an array was executed.

EXPERIMENT-07

1. Write a program to store array elements in ascending order in 8085

Aim : Store array elements in ascending order

Algorithm:

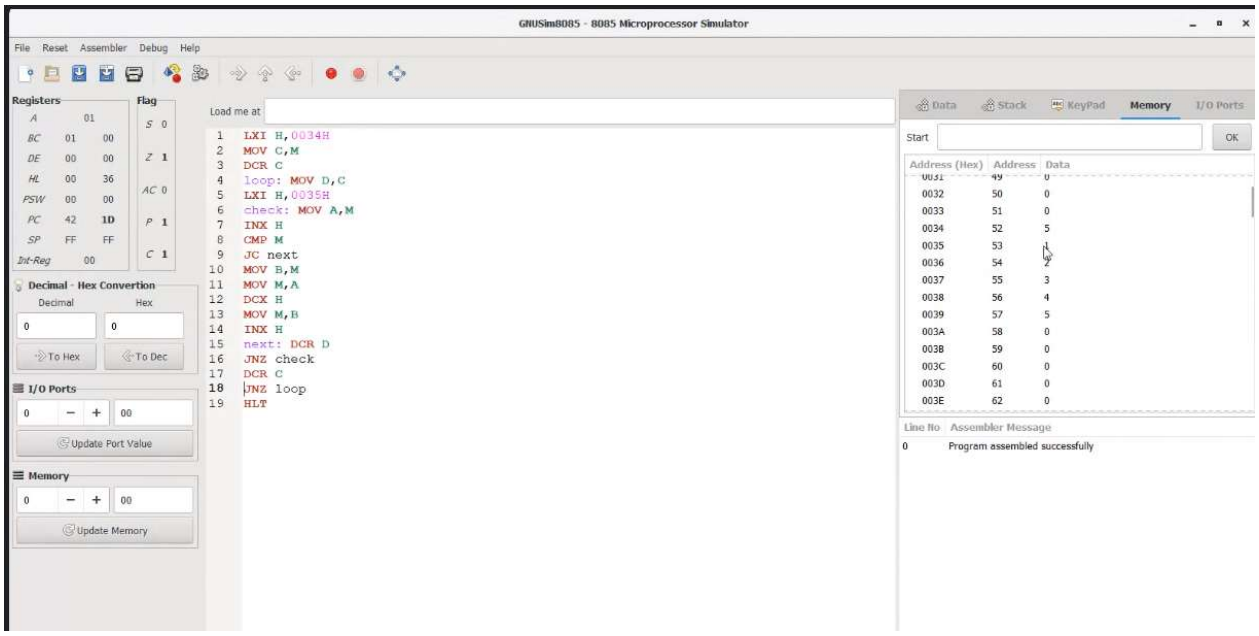
1. **Load H-L pair with the size of array.**
2. **Move the value of register C and move to another register D. These two registers will be counters**
3. **Increment H-L pair to the point next location containing first array data values.**
4. **Move the first array element to the accumulator.**
5. **Compare the subsequent elements with the value in A, if smaller move to accumulator. Swap the two values and store the other value in register B. Also decrement D.**
6. **Continue this step until D reached zero. Then decrement register C store value of (C-10) to D**
7. **Repeat steps 5-6 until C becomes zero.**
8. **Terminate the program.**

Program:-

Address	HEX Codes	Labels	Mnemonics	Comments
8000	21, 40, 80	START	LXI H, 8040H	Pointer to the IN-BUFFER
8003	16, 00		MVI D, 00H	The D register is used as a flag register
8005	4E		MOV C, M	Initialize reg. C with data count
8006	0D		DCR C	Set Reg. C for comparison count

8007	23		INX H	Pointing to the next location
8008	7E	CHECK	MOV A, M	Get the number
8009	23		INX H	Go to next location
800A	BE		CMP M	Compare the contents of the current memory location with the contents of the accumulator
800B	DA, 14, 80		JC NEXTBYT	If (A) < second byte, do not exchange
800E	46		MOV B, M	Get second byte for exchange
800F	77		MOV M, A	Store first byte in second location
8010	2B		DCX H	Point to first location
8011	70		MOV M, B	Store second byte in first location
8012	23		INX H	Get ready for next comparison
8013	16, 01		MVI D, 01H	Load 1 in D as a remainder for exchange

8015	0D	NEXTBYT	DCR C	Decrement comparison count
8016	C2, 08, 80		JNZ CHECK	If comparison count not 0, go back
8019	7A		MOV A, D	Get flag bit in A
801A	0F		RRC	Place flag bit D0in carry
801B	DA, 00, 80		JC START	If flag is 1, exchange occurred
801E	76		HLT	Terminate the program



Observation :

Input:

05	3000
02	3001
03	3002
04	3003
01	3004
06	3005

Output:

05	3000
01	3001
02	3002
03	3003
04	3004
06	3005

Result:
Thus, the array elements have been sorted in ascending order.

2. Write a program to store array elements in descending order in 8085.

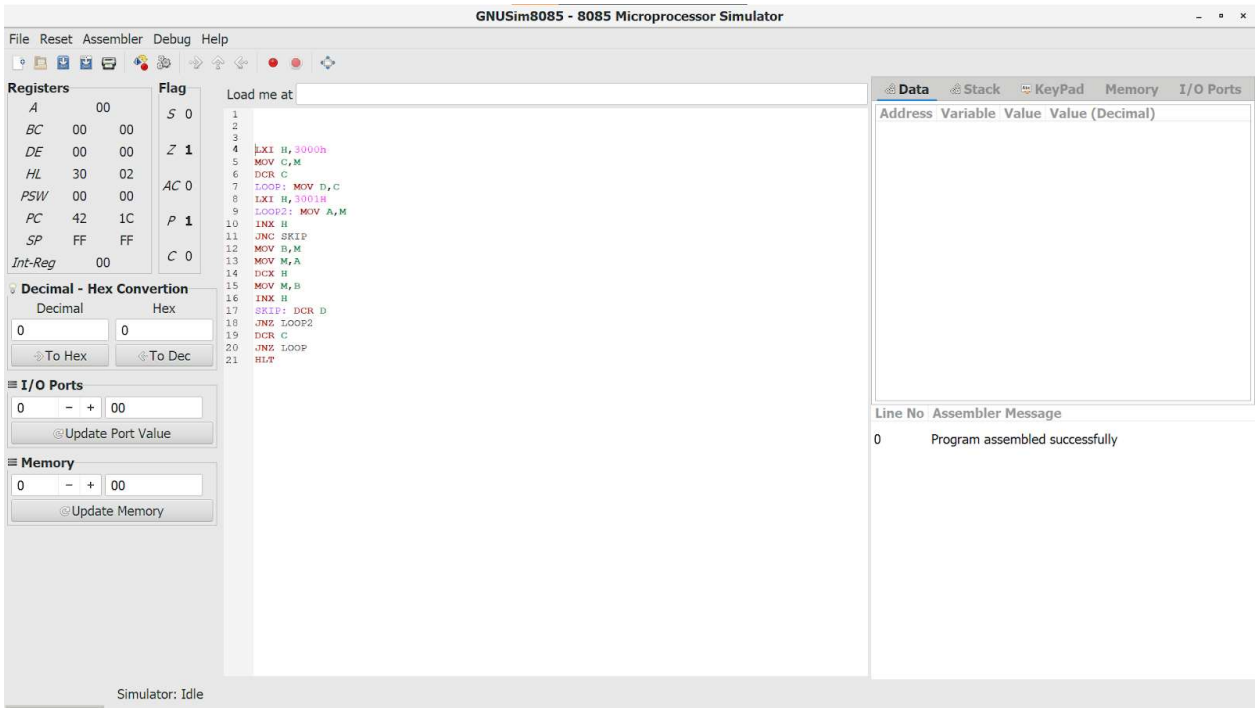
Aim: Store array elements in descending order.

Algorithm:

1. **Load H-L pair with the size of array.**
2. **Move the value of register C and move to another register D. These two registers will be counters**
3. **Increment H-L pair to the point next location containing first array data values.**
4. **Move the first array element to the accumulator.**
5. **Compare the subsequent elements with the value in A, if greater, move to accumulator. Swap the two values and store the other value in register B. Also decrement D.**
6. **Continue this step until D reached zero. Then decrement register C store value of (C-10) to D**
7. **Repeat steps 5-6 until C becomes zero.**
8. **Terminate the program.**

Program :-

```
        LXI H,5000          ;Set pointer for array
        MOV C,M             ;Load the Count
        DCR C               ;Decrement Count
REPEAT: MOV D,C
        LXI H,5001          ;Load starting address of data array
LOOP:   MOV A,M             ;copy content of memory location to Accumulator
        INX H
        CMP M
        JNC SKIP            ;Jump to skip if carry not generated
        MOV B,M             ;copy content of memory location to B - Register
        MOV M,A             ;copy content of A - Register to memory location
        DCX H               ;Decrement content of HL pair of registers
        MOV M,B
        INX H               ;Increment content of HL pair of registers
SKIP:   DCR D
        JNZ LOOP            ;Jump to LOOP if not Zero
        DCR C               ;Decrement Count
        JNZ REPEAT          ;Jump to REPEAT if not Zero
        HLT                 ;Terminate Program
```



Observation:
Input :-
3000h – 5 (no of elements in array)
Array (3001h – 3005h) – 2 3 4 1 6
Output :-
Array (3001h – 3005h) – 6 4 3 2 1 (Descending order)

Result:
Thus, the array elements have been sorted in descending order.

EXPERIMENT-08

1. Write a program in 8085 to find sum of all odd numbers in an array

Aim : find sum of all odd numbers in an array

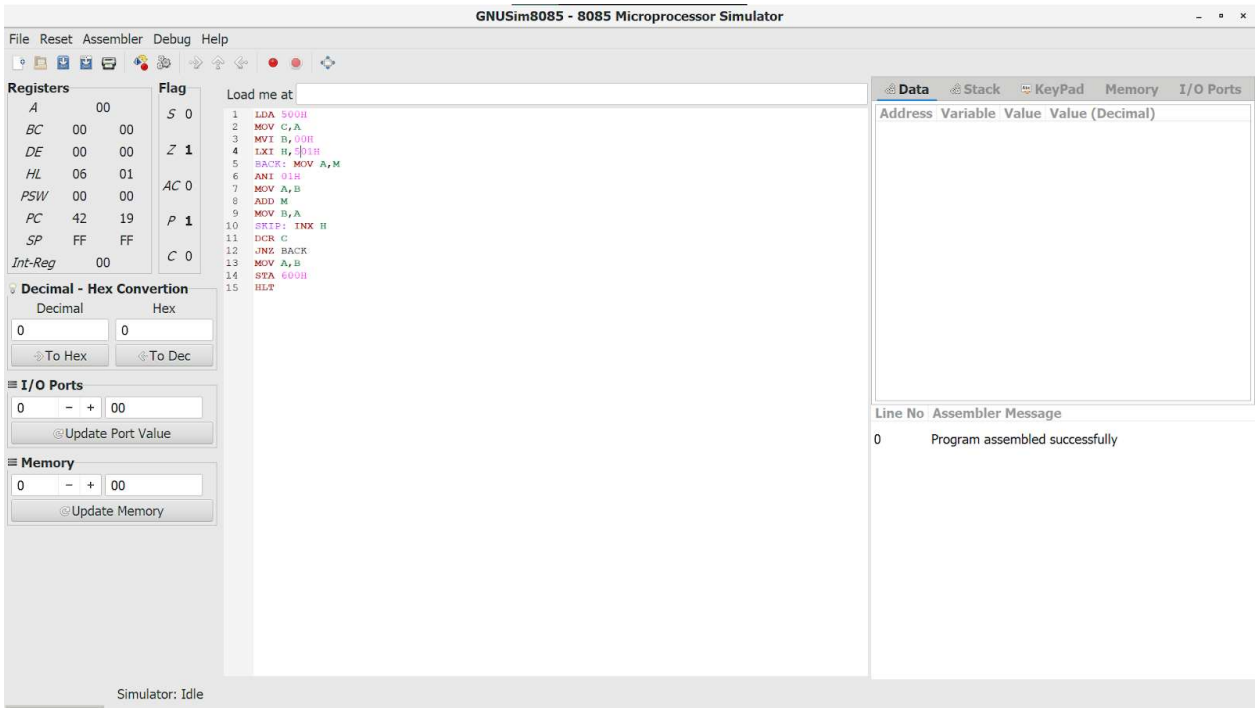
Algorithm:

1. Load data from offset 500 to register CL.
2. Increment the value of offset.
3. Load 00H into CH register.
4. Load 00H into AL register.
5. Load data from offset to register BL.
6. Use TEST instruction to check whether data in BL is even or odd, if zero flag is set means data is even then go to step 7 otherwise data is odd then go to step 8.
7. Jump to memory location 413H.
8. Add the data of AL and BL registers and store the result in AL register.

9. Increment the value of offset.
10. Jump to memory location 40AH if content of CX is not equal to zero otherwise go to step 11.
11. Load the data from AL register to memory location 600
12. End

Program :-

400	MOV SI, 500	SI<-500
403	MOV CL, [SI]	CL<-[SI]
405	INC SI	SI<-SI+1
406	MOV CH, 00	CH<-00
408	MOV AL, 00	AL<-00
40A	MOV BL, [SI]	BL<-[SI]
40C	TEST BL, 01	BL.01
40F	JZ 413	Jump to 413 memory location if zero flag is set
411	ADD AL, BL	AL<-AL+BL
413	INC SI	SI<-SI+1
414	LOOP 40A	jump to 40A memory location if content of CX is not equal to zero
416	MOV [600], AL	[600], AL
41A	HLT	end



Observation :

INPUT:-

500 H = 04
501 H = 15
502 H = 28
503 H = 07
504 H = 08

Result = 600 = 1C

Result:-

Thus , a program to find the sum of odd numbers in a given array was successfully executed.

---

2. Write a program in 8085 to find sum of all even numbers in an array

Aim : find sum of all even numbers in an array

Algorithm:-

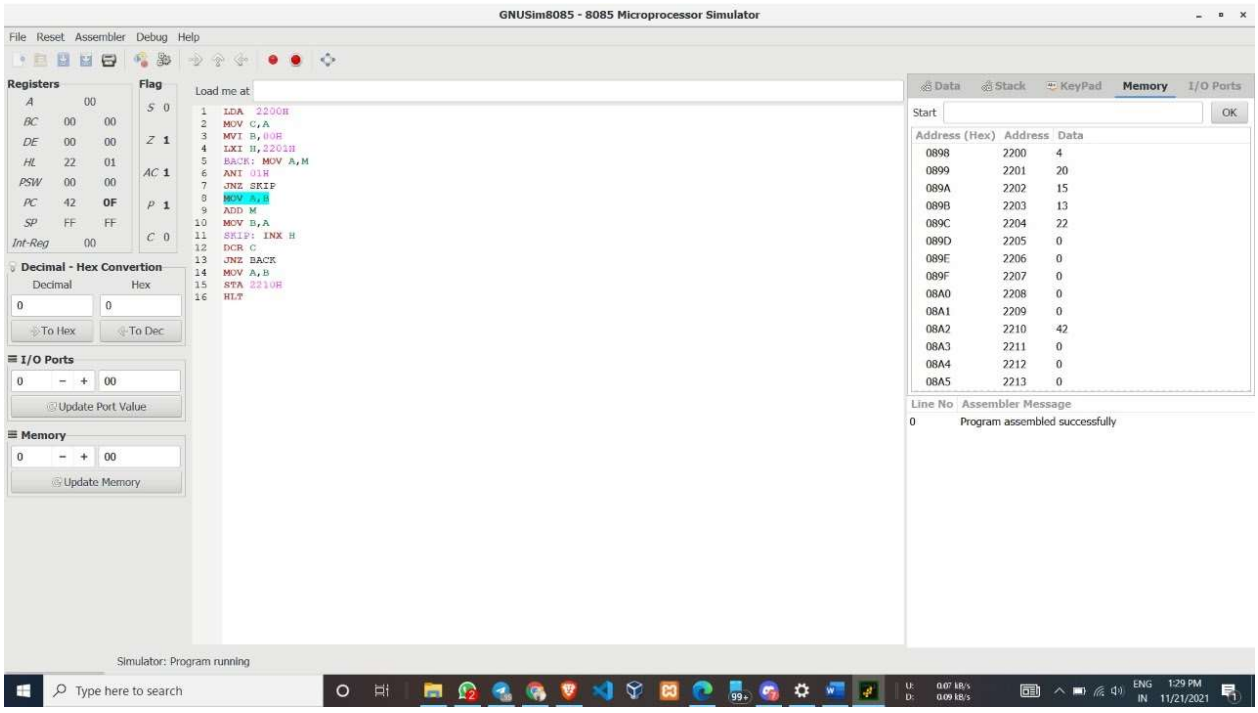
1. **Assign 2200 to SI**
2. **Load data from offset SI to register CL (count) and assign 00 to register CH inc. SI by 1**
3. **Load data from offset SI and apply TEST with 01, if result is non zero jump to step 5**

4. **Add the offset data with register AL**
5. **Increase offset by 1**
6. **LOOP to step 3**
7. **Store the result (content of register AL) to offset 600**
8. **Stop**

Program:-

Address	HEX Codes	Labels	Mnemonics	Comments
F000	3A, 00, F1		LDA F100	Load number of elements into A
F003	4F		MOV C,A	Load counter
F004	06, 00		MVI B,00	Clear B to store Sum
F006	21, 01, F1		LXI H,F101	Initialize pointer to take numbers
F009	7E	LOOP	MOV A,M	Take number from memory
F00A	E6, 01		ANI 01	Mask bits to check even or not
F00C	C2, 12, F0		JNZ SKIP	If it is not zero, skip it
F00F	78		MOV A,B	Load B to A
F010	86		ADD M	Add A and memory element

F011	47		MOV B,A	Store A to B
F012	23	SKIP	INX H	Point to next location
F013	0D		DCR C	Decrease Counter
F014	C2, 09, F0		JNZ LOOP	Until Z = 1, jump to LOOP
F017	32, 50, F1		STA F150	Store result
F020	76		HLT	Terminate program



Observation:

INPUT:-

2500 H = 4H
2501 H = 20H
2502 H = 15H
2503 H = 13H
2504 H = 22H

Result = 2505 H = 20+22= 42H

Result:

Thus , a program to find the sum of even numbers in a given array was successfully executed.

Conclusion: -

Thus, all the experiments were implemented and executed successfully.

***