

# EXPERIMENT NO : 5

NAME : MD WASIF || ENRIL : 19UICS002  
SUBJ : CAO LAB || EXPERIMENT NO : 5

## 1. Write a program in 8085 for Division of two numbers using memory location

**AIM-** To perform the division of two numbers with memory location

### **ALGORITHM-**

- 1) Load the H-L pair with data values from specific memory location.
- 2) Move the data into accumulator.
- 3) Load the H-L pair with next data.
- 4) Move the data to another register.
- 5) Compare the two data values.
- 6) If no carry is generated, subtract the smaller data and increment the register(say C) which will hold the quotient.
- 7) Repeat step 6 until borrow is generated.
- 8) The result is generated and remainder is the contents of accumulator.
- 9) Terminate the program.

## PROGRAM:

PROGRAM:				
Memory Address	Opcode	Operand	Hexcode	Comments
C000	LXI	H,D000	21	Load H-L pair with content of location D000
C001			0	
C002			D0	
C003	MOV	B,M	46	Moving data to register B
C004	MVI	C,00	0E	Initialize register C to 0
C005			0	
C006	INX	H	23	Increment H-L pair
C007	MOV	A,M	7E	Moving data to accumulator
C008	CMP	B	88	compare with register B
C009	JC	C011	DA	Jump on carry to location C011
C00A			11	
C00B			C0	
C00C	SUB	B	90	Subtract B from A
C00D	INR	C	0C	Increment value of C
C00E	JMP	C008	C3	Jump back to looping condition
C00F			8	
C010			C0	
C011	STA	D002	32	Store result in location D001
C012			2	
C013			D0	
C014	MOV	A,C	79	Moving quotient to accumulator
C015	STA	D003	32	Storing to location D002
C016			32	
C017			D0	
C018	HLT		76	Terminate

## OVSERVATIONS:

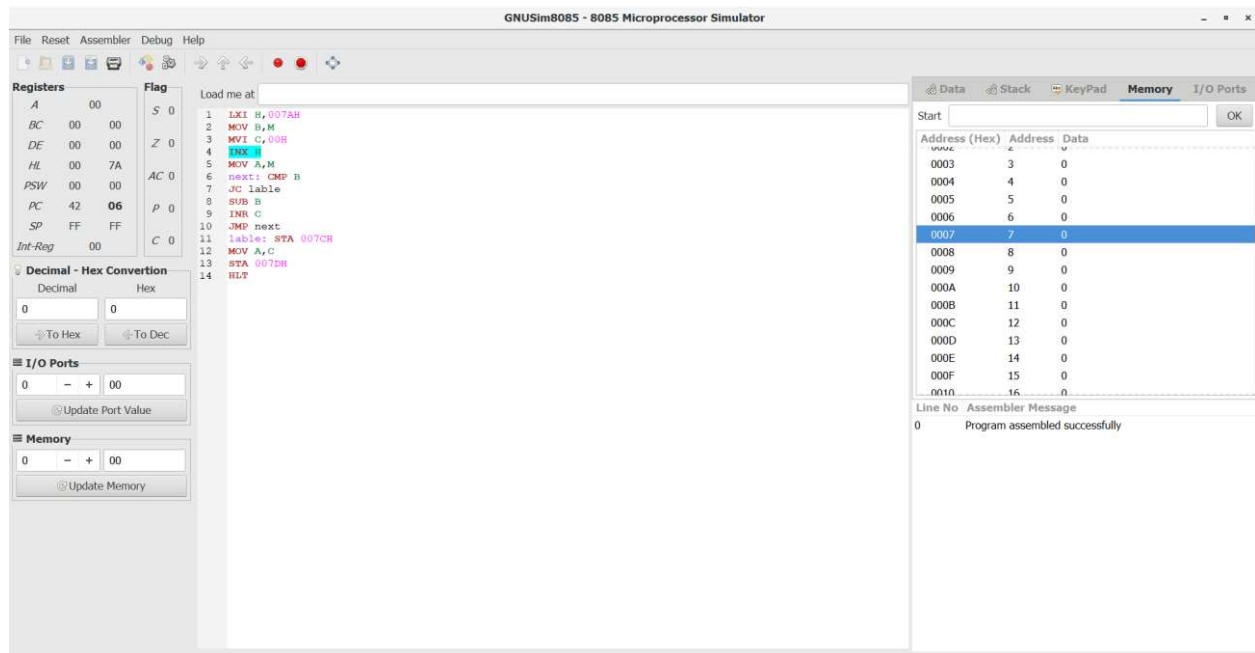
Input: 03 D000      Output: 03 D000

05 D001                      05 D001

XX D002                      02 D002

XX D003                      01 D003

Result: Thus, the division of two 8-bit numbers using memory location executed.



## 2. Write a program to find the largest number in an array in 8085.

**AIM:** To perform the largest number in an array using 8085.

### ALGORITHM-

- 1) Load the H-L pair with the size of array which is stored in a particular memory location.
- 2) Increase H-L pair to point to next locations, containing the data values of the array.
- 3) Move the first data into the accumulator.
- 4) Compare the subsequent data with the first data in accumulator.
- 5) Decrement counter(size of array).
- 6) If carry is generated , that is , data is larger than the pervious data, move it to the accumulator.
- 7) The loop runs unit the counter becomes zero.
- 8) Move the result to desired location.
- 9) Terminate the program.

## PROGRAM:

### Program

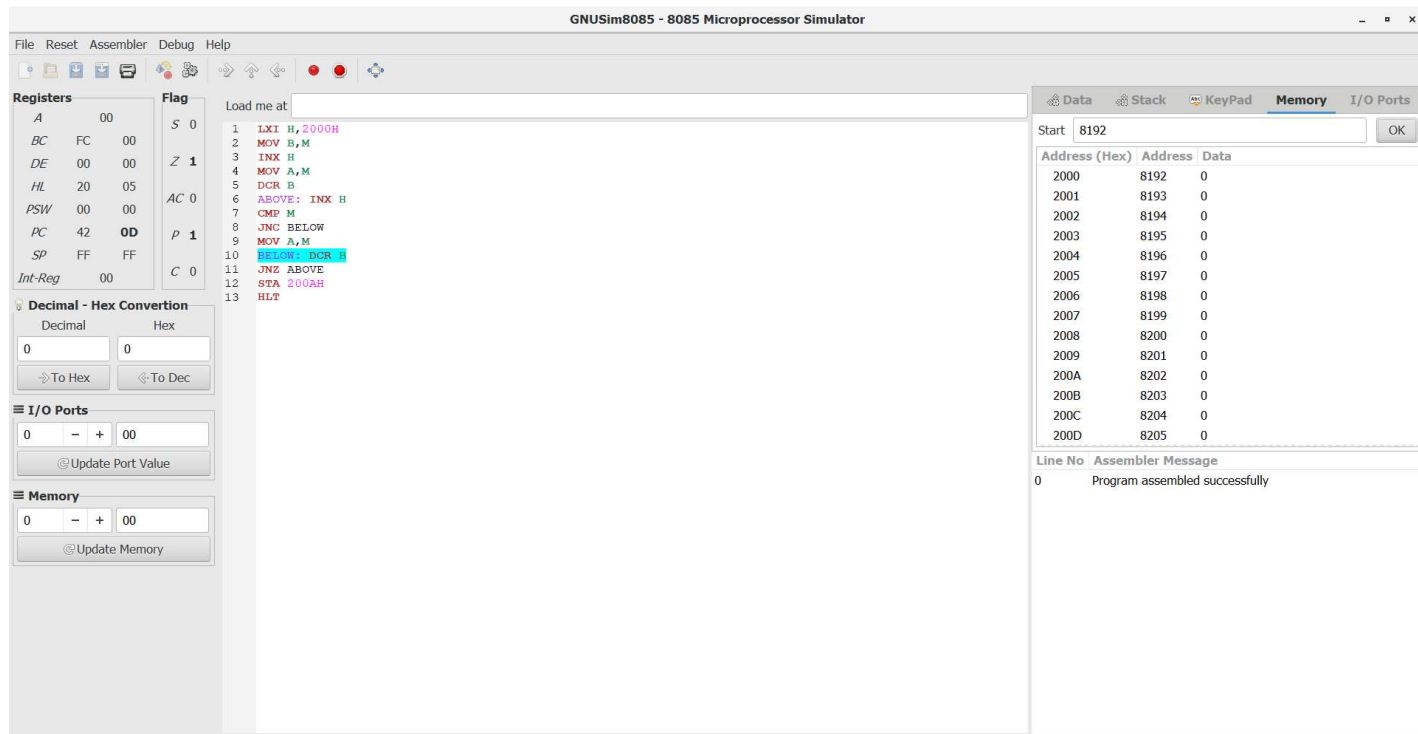
Address	Mnemonics	Operand	Opcode	Comments
2000	LXI	H, 3000H	21	Load H-L pair with address 3000H.
2001			00	
2002			30	
2003	MOV	C, M	4E	Move the counter from memory to reg. C.
2004	INX	H	23	Increment H-L pair.
2005	MOV	A, M	7E	Move the 1 <sup>st</sup> number from memory to reg. A.
2006	DCR	C	0D	Decrement counter.
2007	INX	H	23	Increment H-L pair.
2008	MOV	B, M	46	Move the next number from memory to reg. B.
2009	CMP	B	B8	Compare B with A.
200A	JNC	200EH	D2	Jump to address 200EH if there is no carry.
200B			0E	
200C			20	
200D	MOV	A, B	78	Move largest from reg. B to reg. A.
200E	DCR	C	0D	Decrement counter.
200F	JNZ	2007H	C2	Jump to address 2007H if the counter is not zero.
2010			07	
2011			20	
2012	INX	H	23	Increment H-L pair.
2013	MOV	M, A	77	Move the result from reg. A to memory.
2014	HLT		76	Halt

## OBSERVATIONS:-

3000H: 05H (Counter)  
3001H: 15H  
3002H: 01H  
3003H: 65H  
3004H: E2H  
3005H: 83H

*After Execution:*

3006H: E2H



### 3. Write a program to find the smallest number in an array in 8085

**AIM:-** To perform the smallest number in an array using 8085

#### **ALGORITHM:**

- 1) Load the address of the first element of the array in HL pair
- 2) Move the count to B – reg.
- 3) Increment the pointer
- 4) Get the first data in A – reg.
- 5) Decrement the count.
- 6) Increment the pointer
- 7) Compare the content of memory addressed by HL pair with that of A - reg.
- 8) If carry = 1, go to step 10 or if Carry = 0 go to step 9
- 9) Move the content of memory addressed by HL to A – reg.
- 10) Decrement the count
- 11) Check for Zero of the count. If ZF = 0, go to step 6, or if ZF = 1 go to next step.
- 12) Store the smallest data in memory.
- 13) Terminate the program.

## PROGRAM:

Program

Address	Mnemonics	Operand	Opcode	Comments
2000	LXI	H, 3000H	21	Load H-L pair with address 3000H.
2001			00	
2002			30	
2003	MOV	C, M	4E	Move the counter from memory to reg. C.
2004	INX	H	23	Increment H-L pair.
2005	MOV	A, M	7E	Move the 1 <sup>st</sup> number from memory to reg. A.
2006	DCR	C	0D	Decrement counter.
2007	INX	H	23	Increment H-L pair.
2008	MOV	B, M	46	Move the next number from memory to reg. B.
2009	CMP	B	B8	Compare B with A.
200A	JC	200EH	DA	Jump to address 200EH if there is no carry.
200B			0E	
200C			20	
200D	MOV	A, B	78	Move smallest from reg. B to reg. A.
200E	DCR	C	0D	Decrement counter.
200F	JNZ	2007H	C2	Jump to address 2007H if the counter is not zero.
2010			07	
2011			20	
2012	INX	H	23	Increment H-L pair.
2013	MOV	M, A	77	Move the result from reg. A to memory.
2014	HLT		76	Halt

## OVSERVATIONS:

*Before Execution:*

3000H: 05H (Counter)

3001H: 15H

3002H: 01H

3003H: 65H

3004H: E2H

3005H: 83H

*After Execution:*

3006H: 01H

