Computer Organization And Architecture

Lab Manual

Subject Code: CSE181402
Class: IV Semester(CSE)

Prepared By Mr. Biswajit Sarma
Assistant Professor

Department of Computer Science & Engineering
JORHAT ENGINEERING COLLEGE
JORHAT : 785007, ASSAM

Vision of the Department

To become a prominent department of Computer Science and Engineering for producing quality human resources to meet the needs of the industry and society

Mission of the Department

1: To impart quality education through well-designed curriculum and academic facilities to meet the computing needs of the industry and society
2: To inculcate the spirit of creativity, team work, innovation, entrepreneurship and professional ethics among the students
3: To facilitate effective interactions to foster networking with alumni, industries, institutions of learning and research and other stake-holders
4: To promote research and continuous learning in the field of Computer Science and Engineering

OBJECTIVE:This lab complements the Computer Organization and Architecture course. Students will gain practical experience with designing and implementing concepts of micro processor systems using 8085 micro processor such as simple arithmetic operations, loop, pointer, counter, interrupt, interfacing.

Program Outcomes

PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal,health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

PSO1	Gain ability to employ modern computer languages, environments and platforms in creating innovative career paths
PSO2	Achieve an ability to implement, test and maintain computer based system that fulfils the desired needs

COMPUTER ORGANIZATION AND ARCHITECTURE LAB SYLLABUS
(Practical Hours: 04, Credits: 00)

Implement the following programs using 8085 Simulator(gnusim8085 and 8085Compiler.jar) and 8085 Kits.

Experiment No:1

AIM: Write a program using 8085Microprocessor for Decimal, Hexadecimal addition and subtraction of two Numbers.

a) Write a program to add two hexadecimal & decimal numbers.

DESCRIPTION/ALGORITHM:-

Hexadecimal Addition : The program takes the content of 2009, adds it to 200B & stores the result back at 200C.

Steps:

1. Initialize HL Reg. pair with address where the first number is lying.
2. Store the number in accumulator.
3. Get the second number.
4. Add the two numbers and store the result in 200B.
5. Go back to Monitor

Let: (2009 H) = 80 H

(200B H) = 15 H

Result = 80 H + 15 H = 95 H

(2009 H) → A

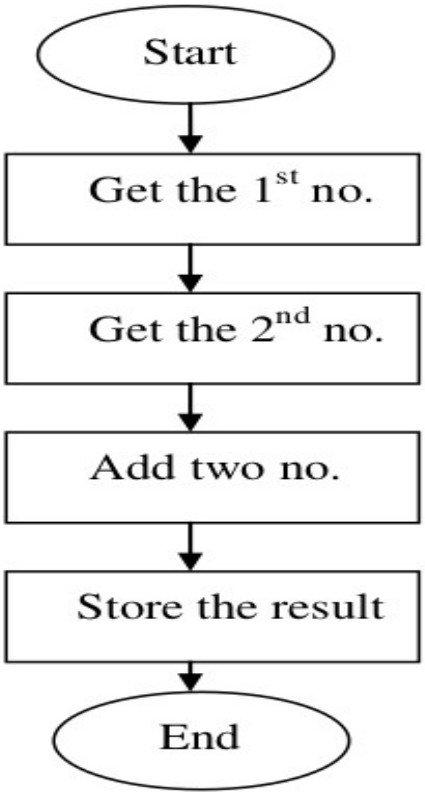
A → B

(200B H) → A

A + B → A

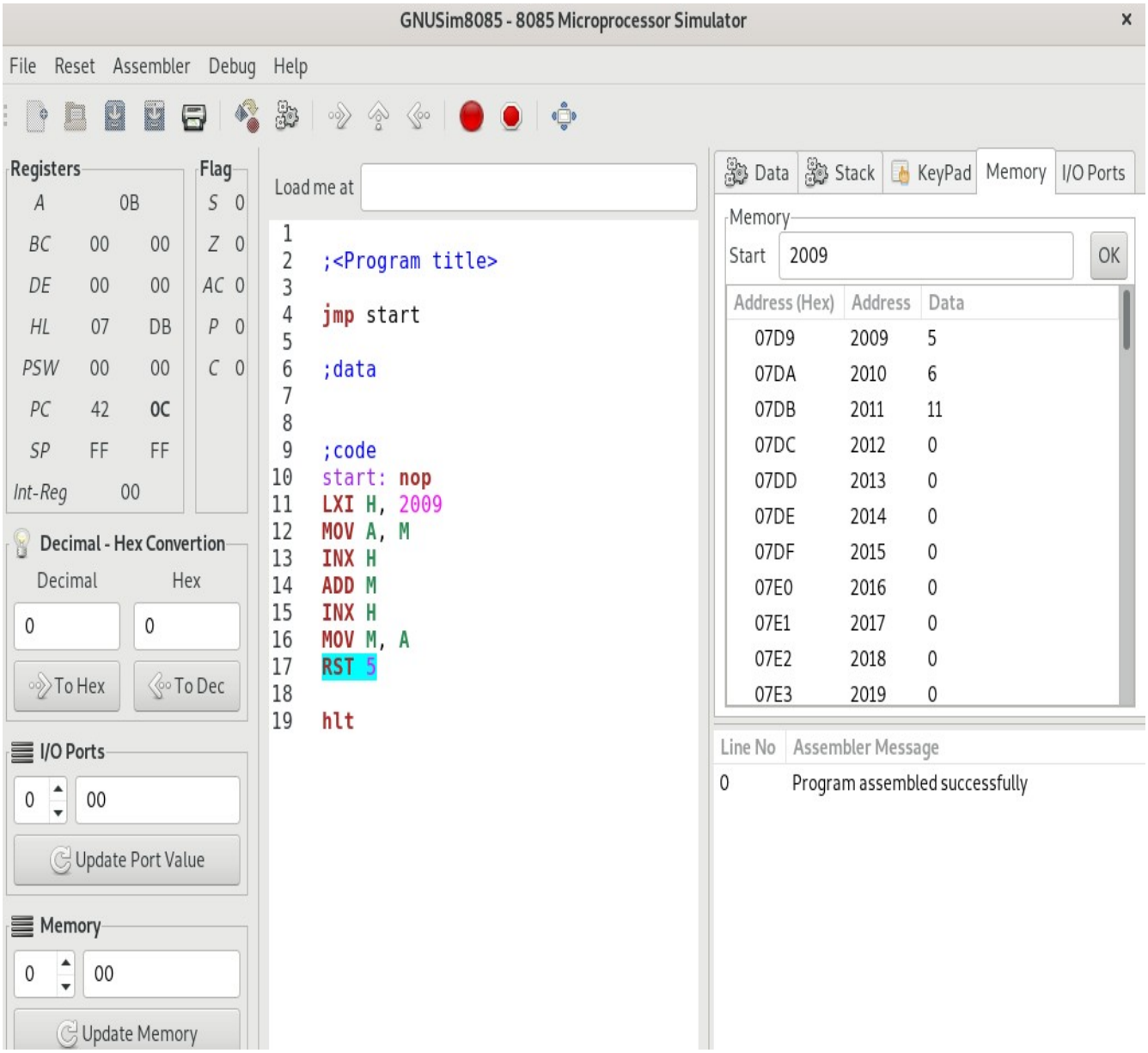
A → (200C H)

FLOWCHART : -



PROGRAM:-

```
LXI H, 2009;  Point 1 st no.
MOV A, M;     Load the acc.
INX H;        Adv Pointer
ADD M;        ADD 2 nd NO.
INX H;        Adv Pointer
MOV M, A;     Store Result
RST 5;
```

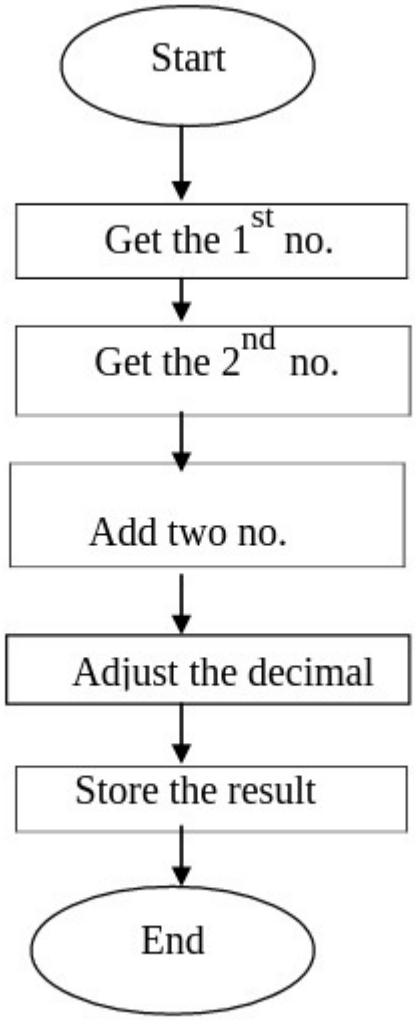

GNUSim8085 - 8085 Microprocessor Simulator
File Reset Assembler Debug Help
Registers
A 0B
BC 00 00
DE 00 00
HL 07 DB
PSW 00 00
PC 42 0C
SP FF FF
Int-Reg 00
Flag
S 0
Z 0
AC 0
P 0
C 0
Decimal - Hex Convertion
Decimal
Hex
0
0
To Hex
To Dec
I/O Ports
0
00
Update Port Value
Memory
0
00
Update Memory
Load me at
1 ;<Program title>
2 jmp start
3 ;data
4 ;code
5 start: nop
6 LXI H, 2009
7 MOV A, M
8 INX H
9 ADD M
10 INX H
11 MOV M, A
12 RST 5
13 hlt
Data Stack KeyPad Memory I/O Ports
Memory
Start 2009
OK
Address (Hex) Address Data
07D9 2009 5
07DA 2010 6
07DB 2011 11
07DC 2012 0
07DD 2013 0
07DE 2014 0
07DF 2015 0
07E0 2016 0
07E1 2017 0
07E2 2018 0
07E3 2019 0
Line No Assembler Message
0 Program assembled successfully

Decimal Addition:

Steps:

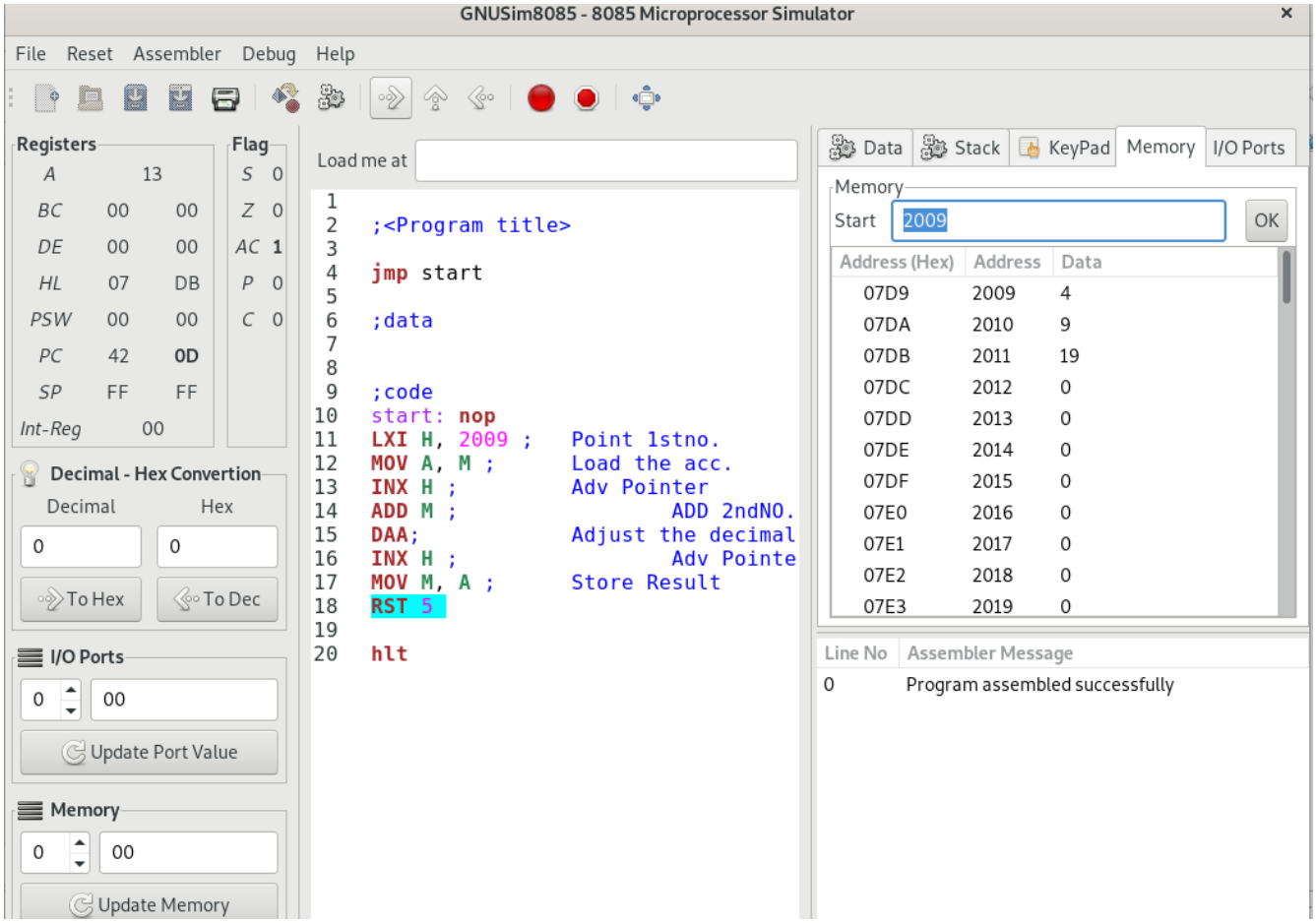
1. Initialize HL Reg. pair with address where the first number is lying.
2. Store the number in accumulator.
3. Get the second number.
4. Add the two numbers and store the result in 200B.
5. Go back to Monitor

FLOWCHART:-



PROGRAM:-

```
LXI H, 2009 ;        Point 1stno.
MOV A, M ;   Load the acc.
INX H ;              Adv Pointer
ADD M ;              ADD 2ndNO.
DAA;         Adjust the decimal
INX H ;              Adv Pointer
MOV M, A ;   Store Result
RST 5
```



Question: Write a program using 8085Microprocessor for Decimal, Hexadecimal subtraction of two Numbers.

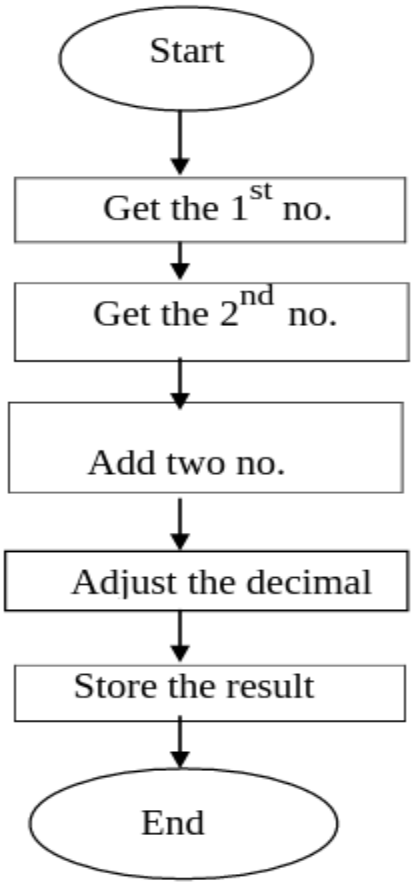
Experiment No:2

AIM: Write a program using 8085 Microprocessor for addition and subtraction of two BCD numbers.

DESCRIPTION/ALGORITHM:-

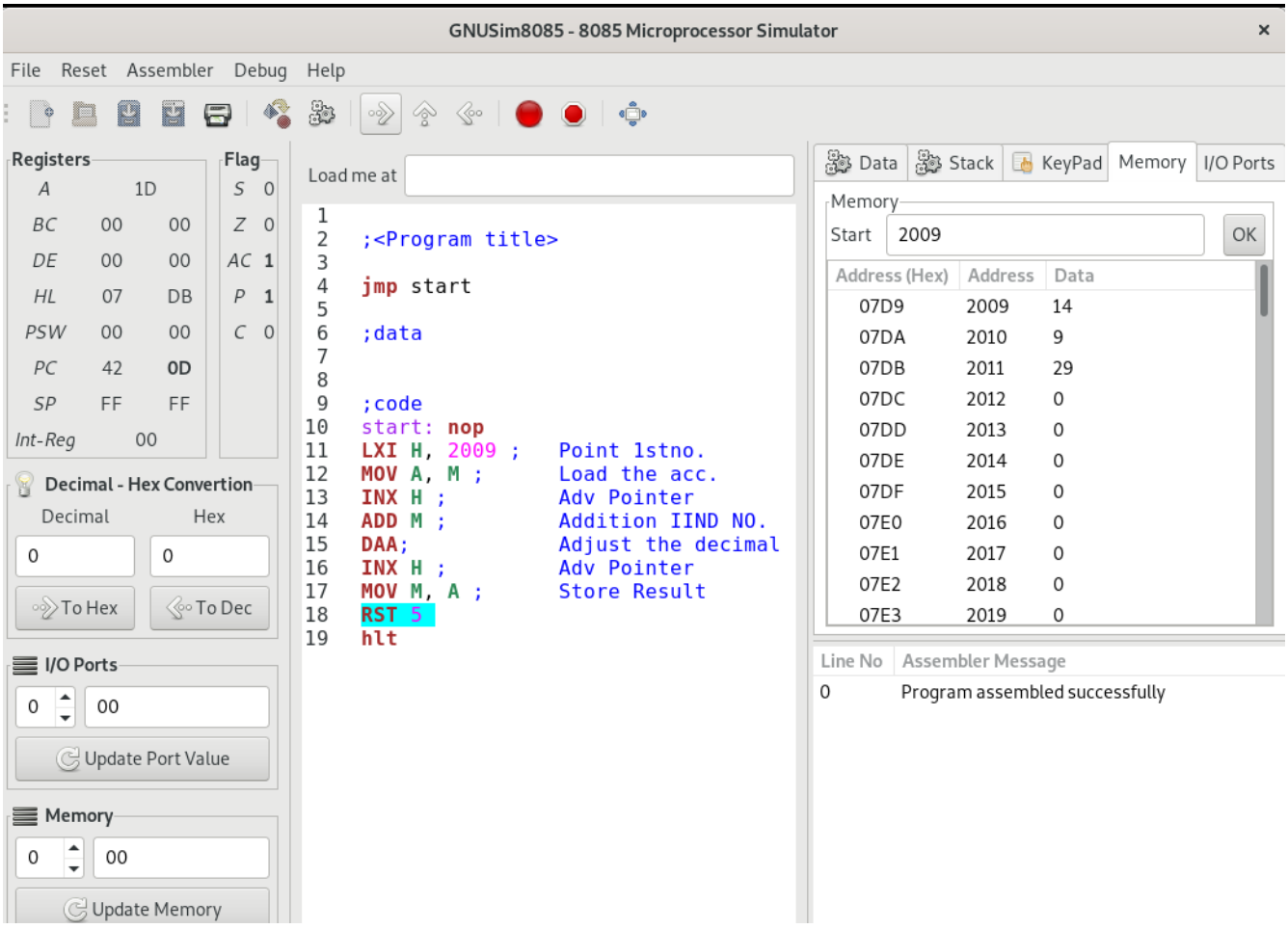
Steps:1. Initialize HL Reg. pair with address where the firstnumber is lying.
2. Store the number in accumulator.
3. Get the second number.
4. Add the two numbers and store the result in 200B.
5. Go back to Monitor

FLOWCHART:-



PROGRAM:-

```
LXI H, 2009 ;      Point 1stno.
MOV A, M ;         Load the acc.
INX H ;            Adv Pointer
ADD M ;            Addition IIND NO.
DAA;               Adjust the decimal
INX H ;            Adv Pointer
MOV M, A ;         Store Result
RST 5
```



Question: Write a program using 8085 Microprocessor for subtraction of two BCD numbers.

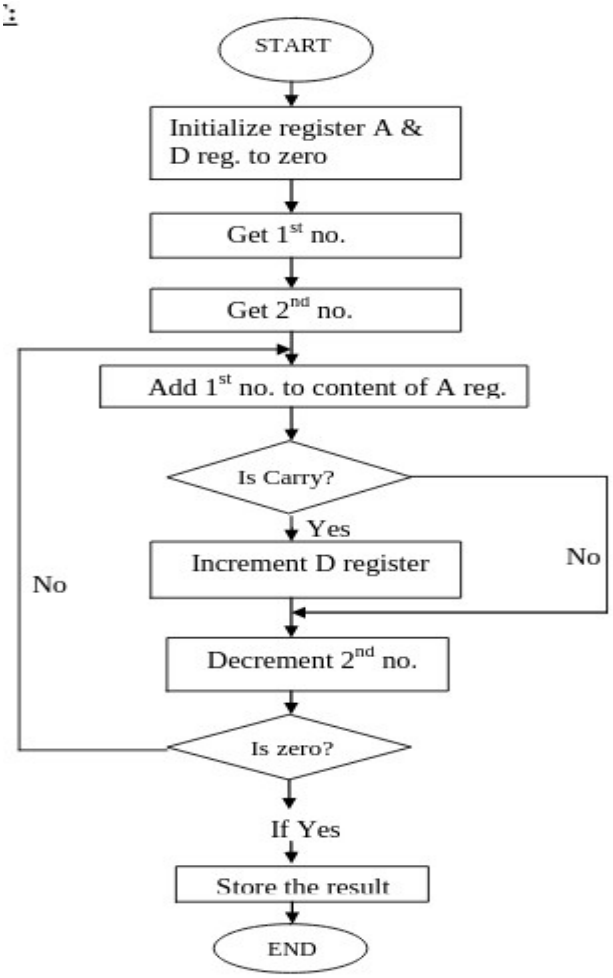
Experiment No:3

AIM: To perform multiplication and division of two 8 bit numbers using 8085.

DESCRIPTION/ALGORITHM:-

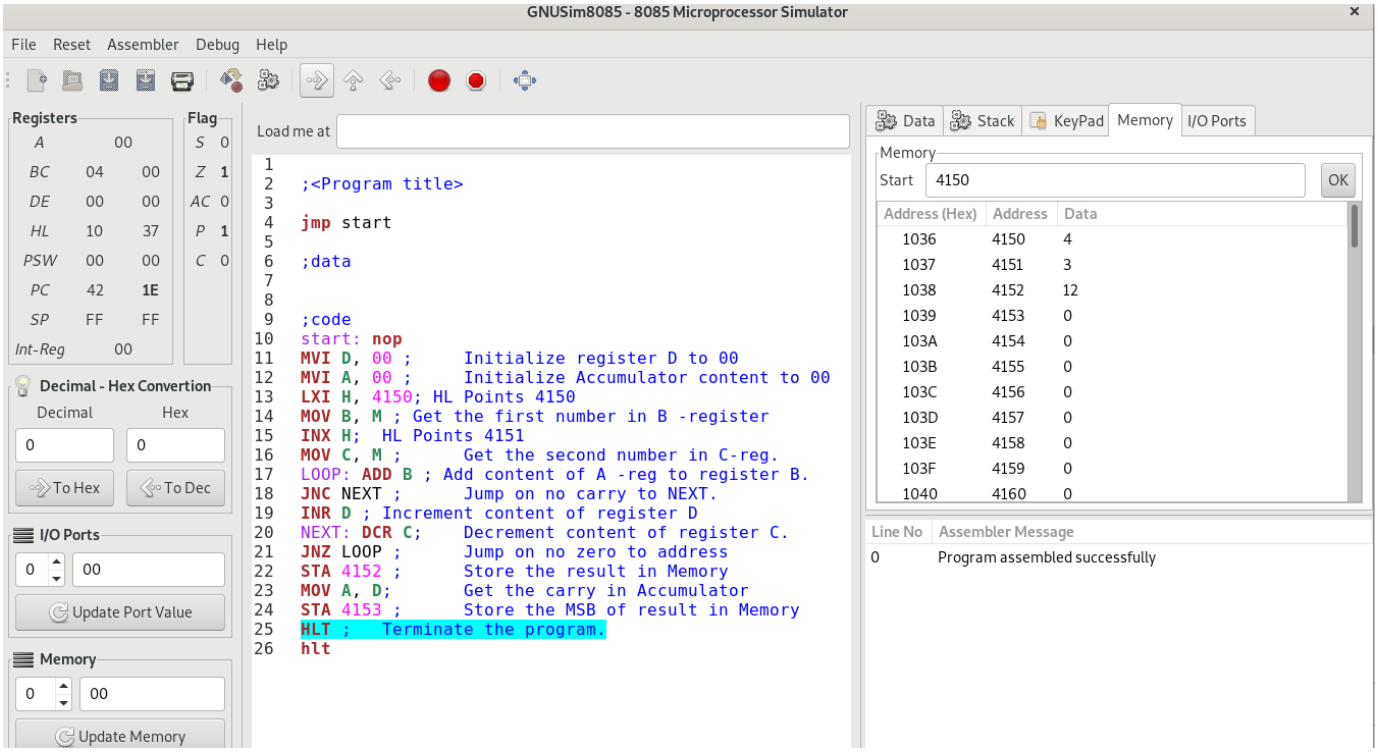
1) Start the program by loading HL register pair with address of memory location.
2) Move the data to a register (B register).
3) Get the second data and load into Accumulator.
4) Add the two register contents.
5) Check for carry.
6) Increment the value of carry.
7) Check whether repeated addition is over and store the value of product and carryin memory location.

FLOWCHART:



PROGRAM:

```
MVI D, 00 ;    Initialize register D to 00
MVI A, 00 ;    Initialize Accumulator content to 00
LXI H, 4150; HL Points 4150
MOV B, M ; Get the first number in B -register
INX H;         HL Points 4151
MOV C, M ;     Get the second number in C-reg.
LOOP: ADD B ; Add content of A -reg to register B.
JNC NEXT ;     Jump on no carry to NEXT.
INR D ;        Increment content of register D
NEXT: DCR C;       Decrement content of register C.
JNZ LOOP ;     Jump on no zero to address
STA 4152 ;     Store the result in Memory
MOV A, D;      Get the carry in Accumulator
STA 4153 ;     Store the MSB of result in Memory
HLT ;  Terminate the program.
```



OBJECTIVE: -Write a program to perform multiplication of two 8 bit numbers using bit rotation method.

DESCRIPTION/ALGORITHM:-

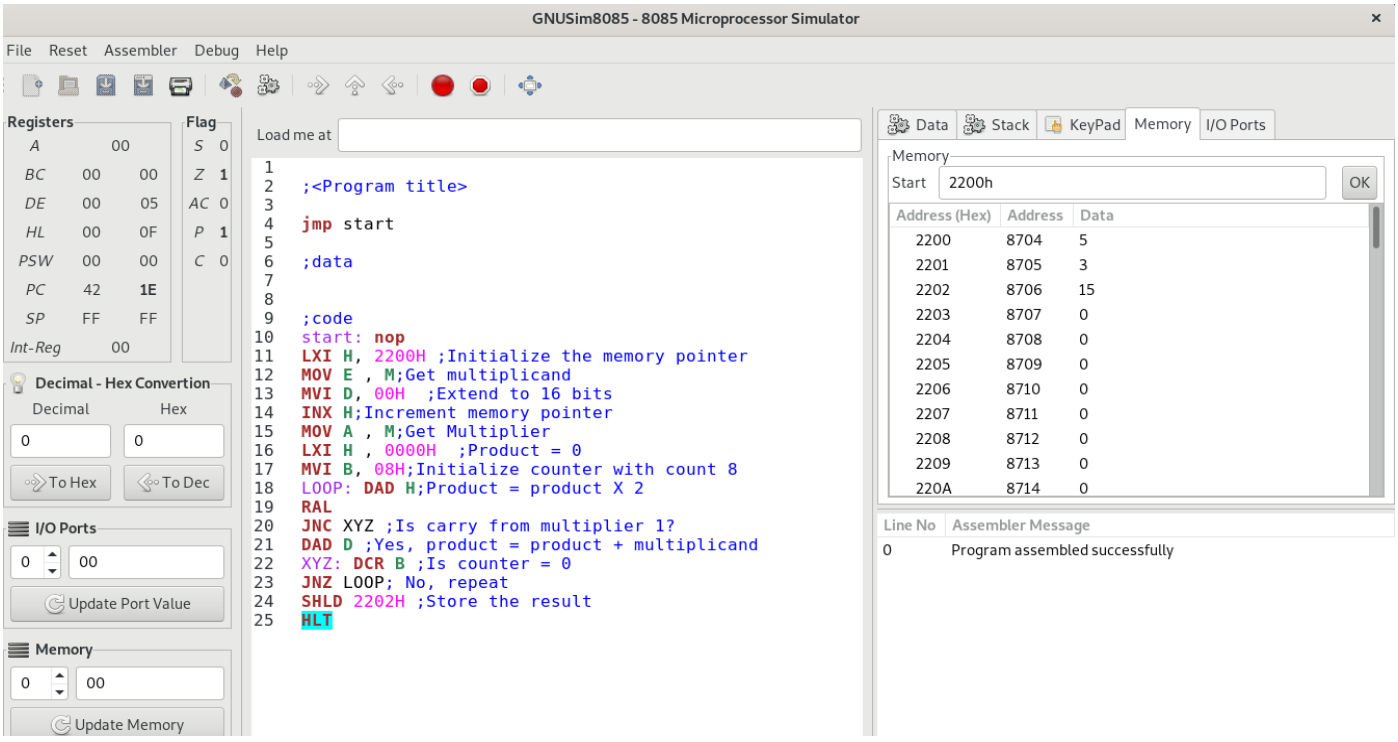
1) Start the program by loading HL register pair with address of memory location.
2) Move the data to a register (Eregister).
3) Get the second data and load into Accumulator.
4) Add the two register contents.
5) Check for carry.
6) Increment the value of carry.
7) Check whether repeated addition is over and store the value of product and carryin memory location.
8) Terminate the program.

EXAMPLE :

Steps	Product								Multiplier					Comments
	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0	CY	B_3	B_2	B_1	B_0	
	0	0	0	0	0	0	0	0	0	0	1	0	1	Initial Stage
Step 1	0	0	0	0	0	0	0	0	0	1	0	1	0	Shift left by 1
	0	0	0	0	0	0	0	0	0	1	0	1	0	Don't add since CY= 0
Step 2	0	0	0	0	0	0	0	0	1	0	1	0	0	Shift
	0	0	0	0	1	1	0	0	1	0	1	0	0	Add multiplicand;CY=1
Step 3	0	0	0	1	1	0	0	0	0	1	0	0	0	Shift left by 1
	0	0	0	1	1	0	0	0	0	1	0	0	0	Don't add since CY= 0
Step 4	0	0	1	1	0	0	0	0	1	0	0	0	0	Add multiplicand;CY=1

PROGRAM:

```
LXI H, 2200H ;Initialize the memory pointer
MOV E , M;Get multiplicand
MVI D, 00H  ;Extend to 16 bits
INX H;Increment memory pointer
MOV A , M;Get Multiplier
LXI H , 0000H  ;Product = 0
MVI B, 08H;Initialize counter with count 8
LOOP: DAD H;Product = product X 2
RAL
JNC XYZ ;Is carry from multiplier 1?
DAD D ;Yes, product = product + multiplicand
XYZ: DCR B ;Is counter = 0
JNZ LOOP; No, repeat
SHLD 2202H ;Store the result
HLT
```



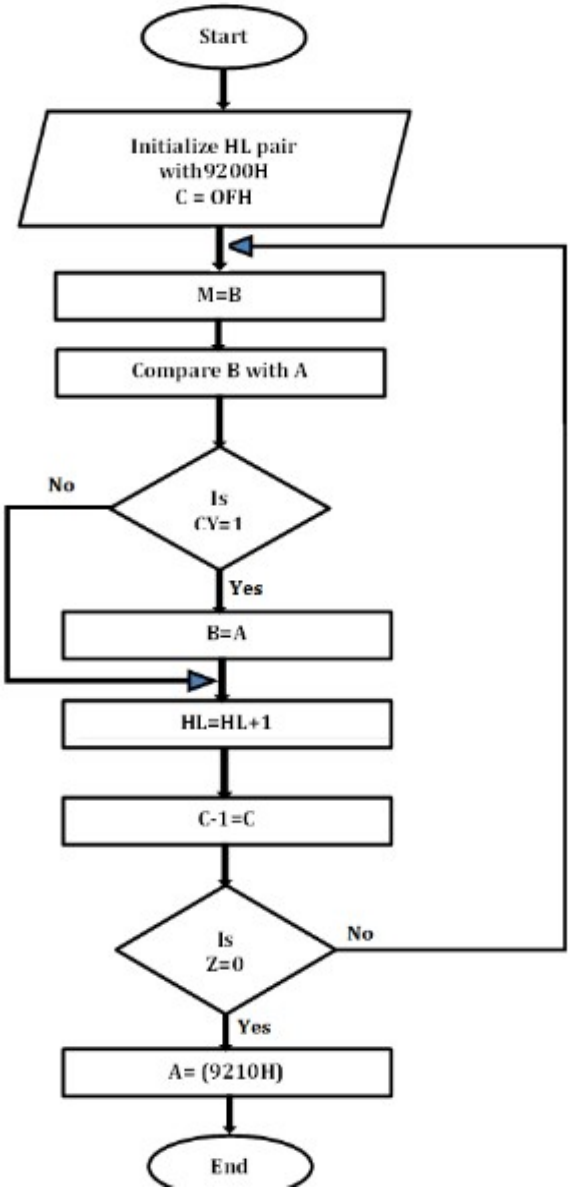
Question: Perform division of two 8 bit numbers using 8085.

Experiment No:4

AIM: To find the largest and smallest number in an array of data using 8085 instruction set.

DESCRIPTION/ALGORITHM:-Write a program to find the largest number in a given array of 6 elements. The array is stored in memory from 9200H onwards. Store the result at the end of the array.

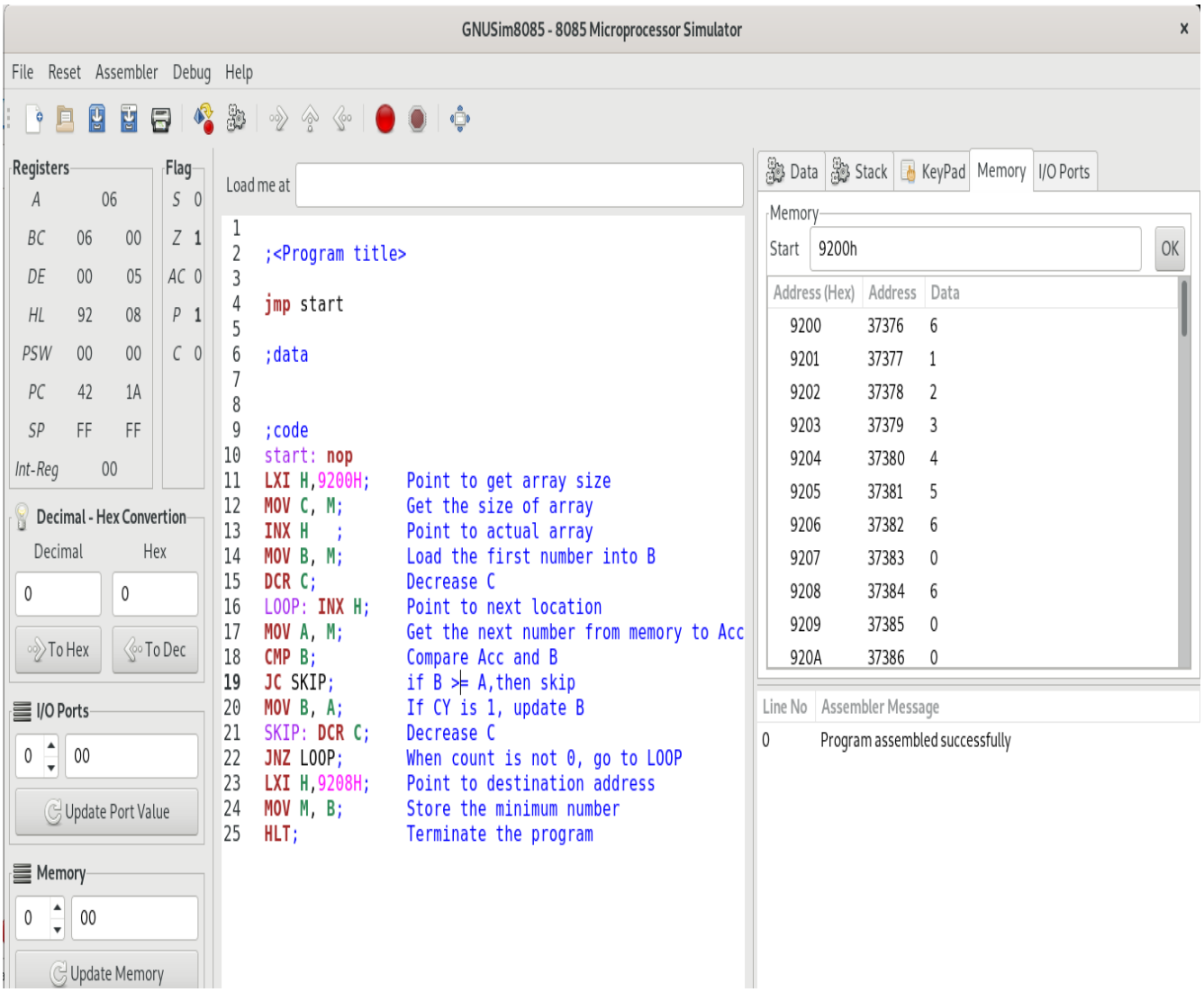
FLOWCHART:-



PROCEDURE:-To find largest of given no. of a given string we compare all given no. one by one. Suppose given no. is 2, 4, 3, 1, 0 1st we compare 2 & 4 (2 is in register A & 4 is in RegisterB). A < B so put B into (A) & Compare with next number i.e. 3 Here A > B so directly compare 4 with 1 then 0.

Program:

```
LXI H,9200H;      Point to get array size
MOV C, M;         Get the size of array
INX H ;           Point to actual array
MOV B, M;         Load the first number into B
DCR C;            Decrease C
LOOP: INX H;      Point to next location
MOV A, M;         Get the next number from memory to Acc
CMP B;            Compare Acc and B
JC SKIP;          if B >= A,then skip
MOV B, A;         If CY is 1, update B
SKIP: DCR C;      Decrease C
JNZ LOOP;         When count is not 0, go to LOOP
LXI H,9208H;      Point to destination address
MOV M, B;         Store the minimum number
HLT;              Terminate the program
```



Question: Find the smallest number in an array of data using 8085 instruction set.

Experiment No:5

AIM: To write a program to arrange an array of data in ascending and descending order.

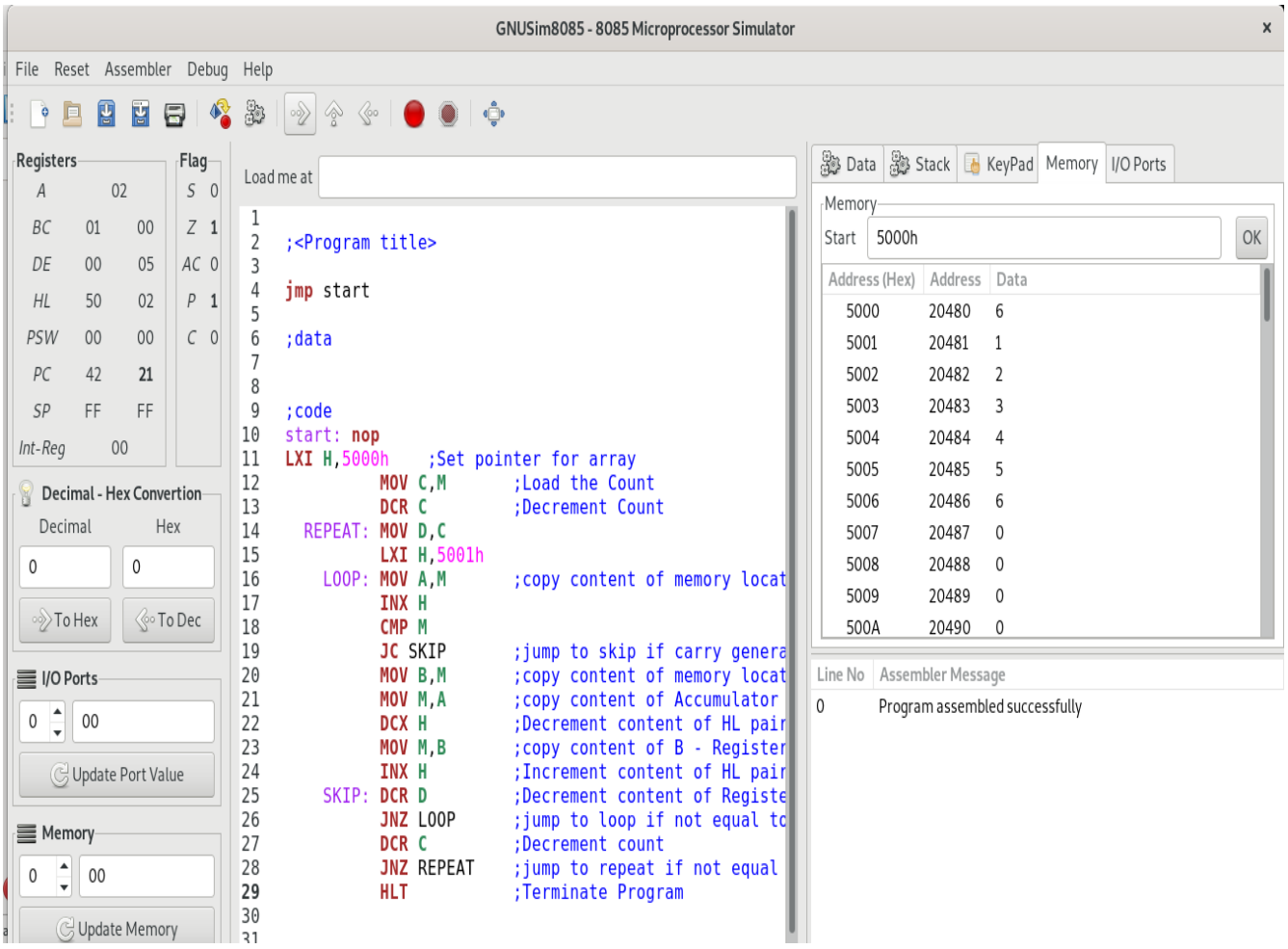
Algorithm

1. Initialize HL pair as memory pointer
2. Get the count at 5000 into C - register
3. Copy it in D - register (for bubble sort (N-1) times required)
4. Get the first value in A - register
5. Compare it with the value at next location
6. If they are out of order, exchange the contents of A - register and Memory
7. Decrement D - register content by 1
8. Repeat steps 5 and 7 till the value in D- register become zero
9. Decrement C - register content by 1
10. Repeat steps 3 to 9 till the value in C - register becomes zero

Program:

```
        LXI H,5000h      ;Set pointer for array
        MOV C,M          ;Load the Count
        DCR C            ;Decrement Count
REPEAT: MOV D,C
        LXI H,5001h
 LOOP:  MOV A,M          ;copy content of memory location to Accumulator
        INX H
        CMP M
        JC SKIP          ;jump to skip if carry generated
        MOV B,M          ;copy content of memory location to B - Register
        MOV M,A          ;copy content of Accumulator to memory location
        DCX H            ;Decrement content of HL pair of registers
        MOV M,B          ;copy content of B - Register to memory location
        INX H            ;Increment content of HL pair of registers
```

```
SKIP: DCR D        ;Decrement content of Register - D
      JNZ LOOP     ;jump to loop if not equal to zero
      DCR C        ;Decrement count
      JNZ REPEAT   ;jump to repeat if not equal to zero
      HLT          ;Terminate Program
```



Question: write a program to arrange an array of data in descending order.

Experiment No:6

AIM: Write an simple Interrupt service routine to understand interrupt.

For this Experiment Please download 8085Compiler.jar file (A new simulator) and run

$ java -jar 8085Compiler.jar

Software interrupts and their vector addresses

Instruction	Machine hex code	Interrupt Vector Address
RST 0	C7	0000H
RST 1	CF	0008H
RST 2	D7	0010H
RST 3	DF	0018H
RST 4	E7	0020H
RST 5	EF	0028H
RST 6	F7	0030H
RST 7	FF	0032H

Hardware interrupts of 8085

Interrupt	Interrupt vector address	Maskable or non-maskable	Edge or level triggered	priority
TRAP	0024H	Non-makable	Level	1
RST 7.5	003CH	Maskable	Rising edge	2
RST 6.5	0034H	Maskable	Level	3
RST 5.5	002CH	Maskable	Level	4
INTR	Decided by hardware	Maskable	Level	5

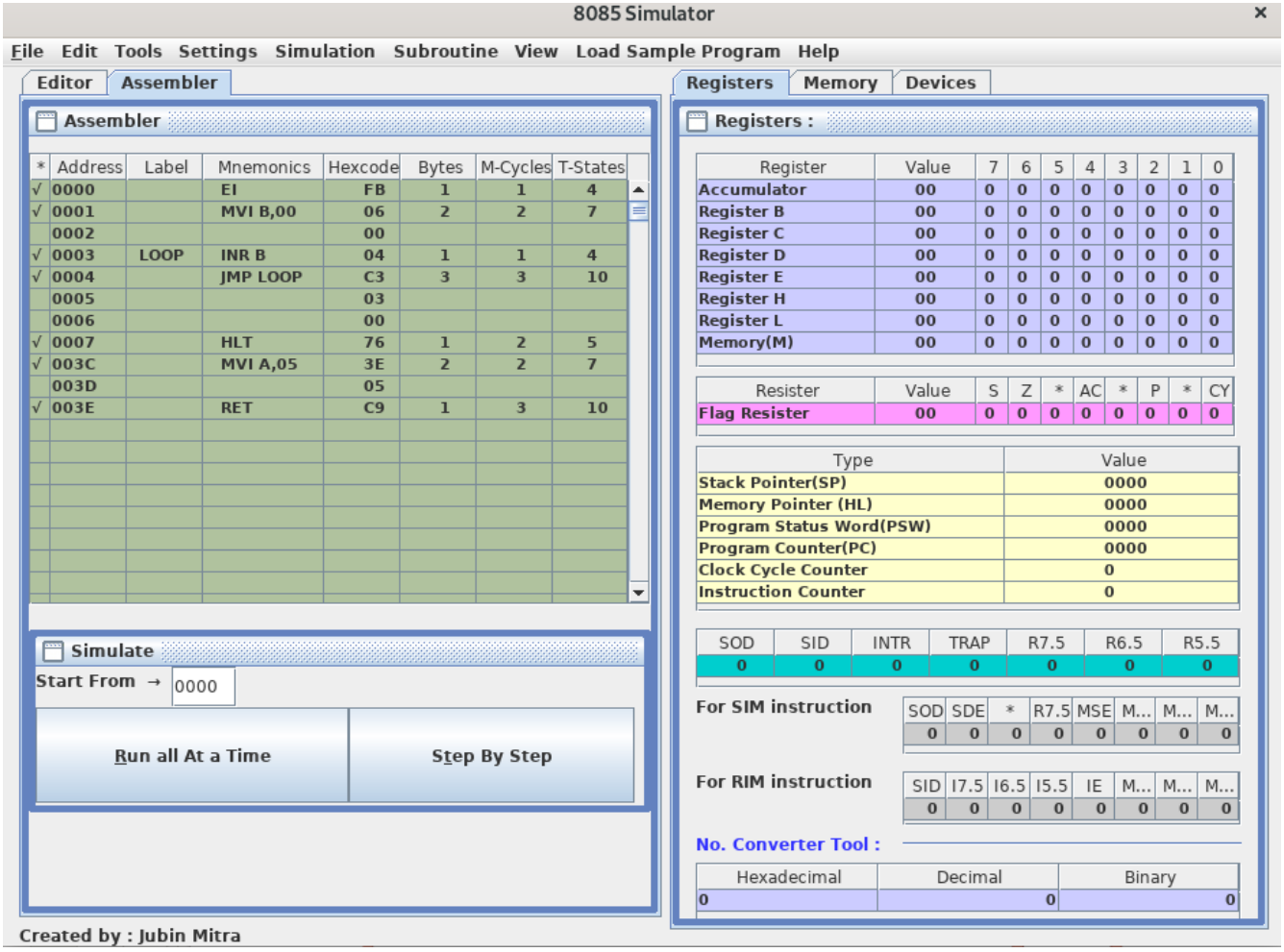
Procesure: As mention in the above tables all the interrrupts have their own interrrupt address. Interrupt means we are doing something (Normal code) suddenly one special thing happens (Interrupt happens) and we need to perform the special task immideatly and then return back to the original work. Lets take one example:

Normal code: Register B is initialized to 0 and it is incremnting indefinately.

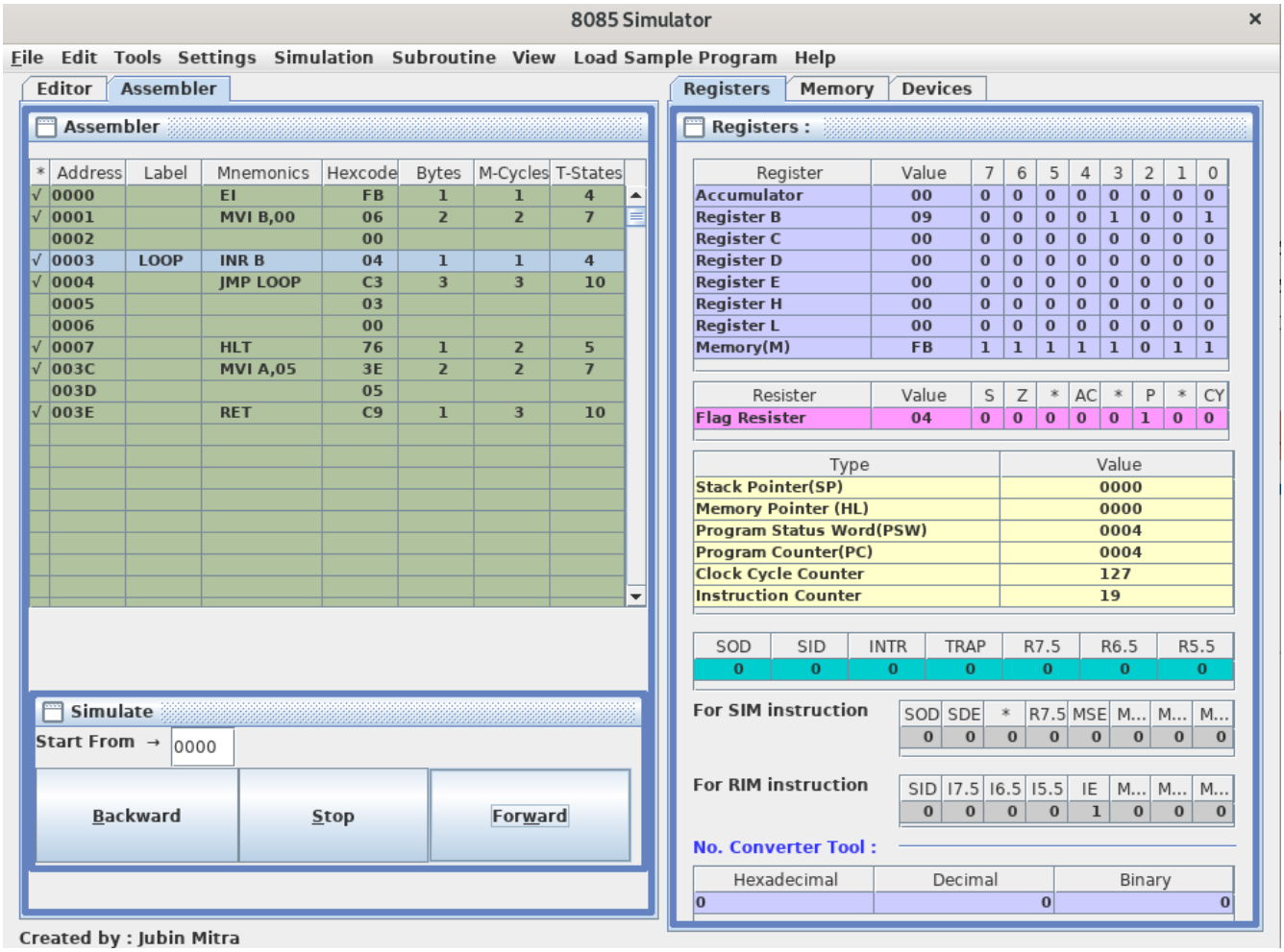
Interrupt code: When interrupt occurs Register A is assigned a value 05.

program:

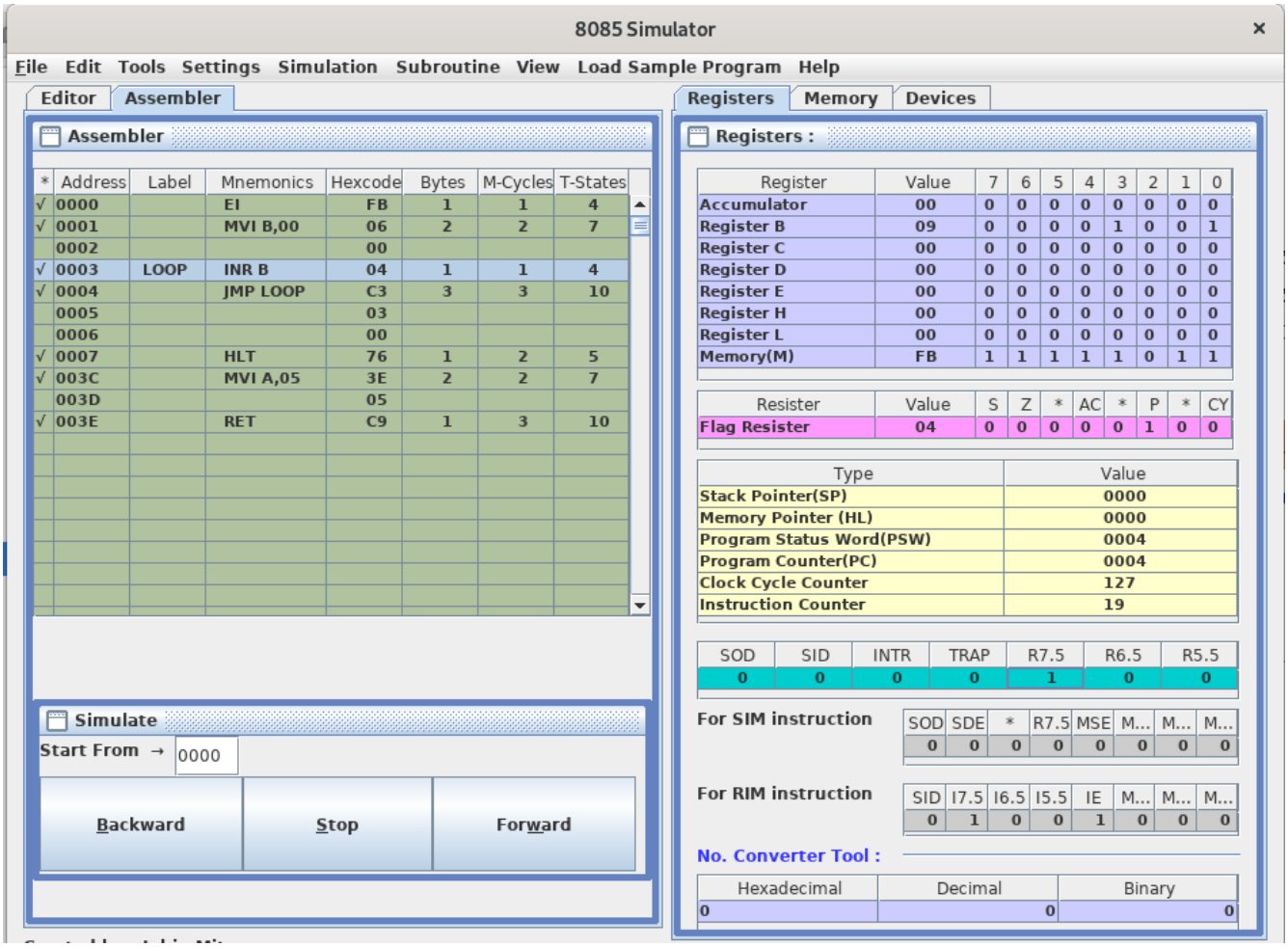
```
EI;             Enabling interrupt
mvi b,00;       Register B is initialized to 0
loop: inr b;    incremening register B
jmp loop;       for ever loop
hlt;            stop the program
#org 003ch ;    RST 7.5 interrupt address, we need to write the interrupt service routine in the address 003ch
mvi a,05;       Register A is set as 05
ret
```



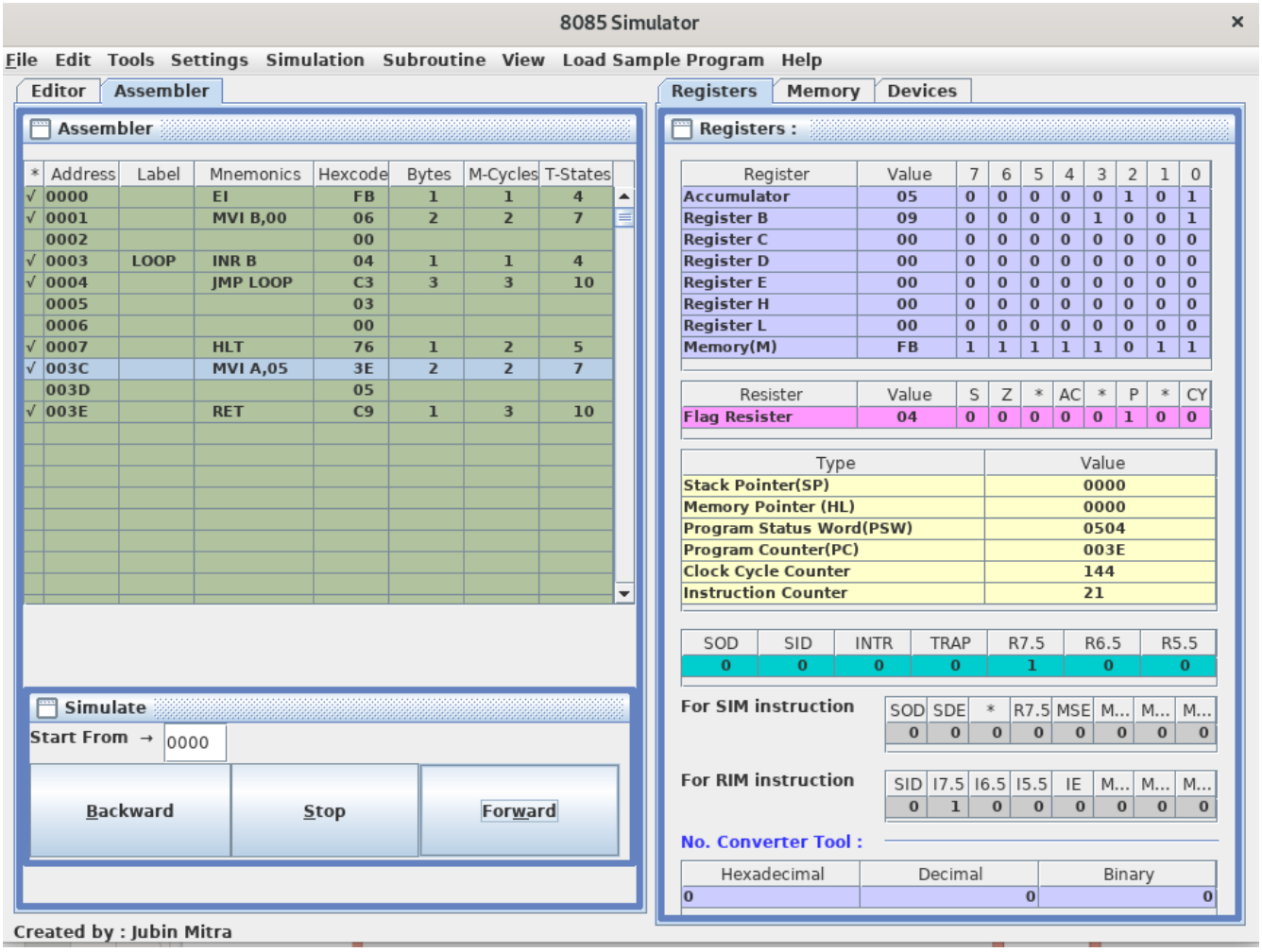
Normal code is loaded in the address 0000 and Inerrupt code loaded in 003c



When we start the program Register B is incremented and reaches 09 value. Still R7.5 is set to 0.



Now we set the value of R7.5 to 1, enabling RST7.5.



When we set the value of R7.5 automatically control jumps to location 003ch and register A value set to as 05.

Experiment No:7

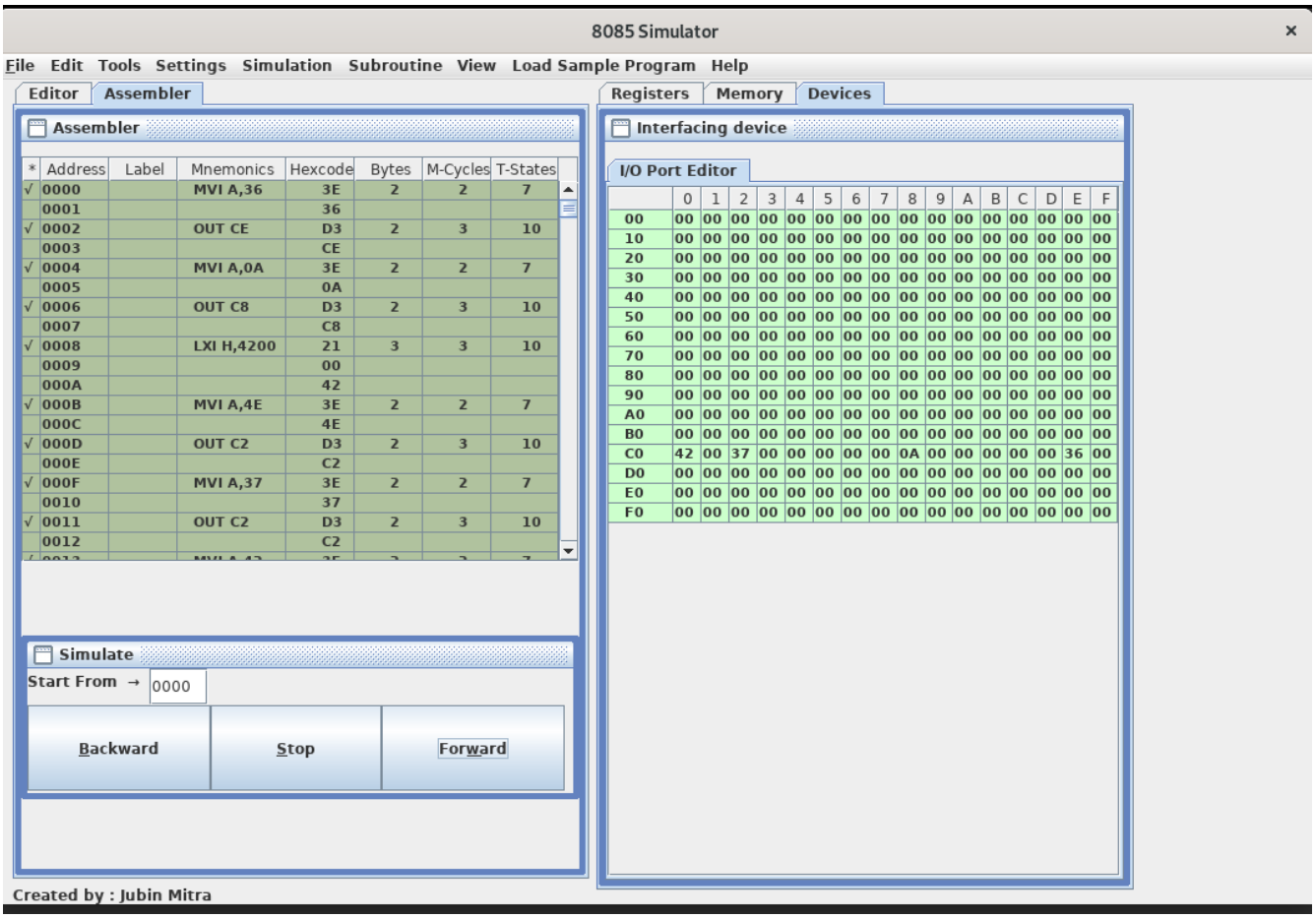
AIM: Interfacing a program to initiate 8251 and to check transmissionand reception of character

DESCRIPTION/ALGORITHM:-

Steps:1.Intitialize timer IC
2.Move the mode command word to A
3.Output it to port address C2
4.Moce the command instruction word to A reg.
5.Output it to port address C2
6.Move the data to be transferred to A
7.Output it to port address C0
8.Reset the system
9.Get data from input port C0
10.Store the value in memory
11.Reset the system

PROGRAM:

```
MVI A,36H
Out CEH
MVI A,0AH
Out C8
HLXI H,4200H
MVI A,4EH
Out C2H
MVI A, 37H
Out C2H
MVI A, 42H
Out C0H
RST 1
#ORG 4200H
In C0H
STA 4500H
RST 1
```



*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0000		MVI A,36	3E	2	2	7
	0001			36			
√	0002		OUT CE	D3	2	3	10
	0003			CE			
√	0004		MVI A,0A	3E	2	2	7
	0005			0A			
√	0006		OUT C8	D3	2	3	10
	0007			C8			
√	0008		LXI H,4200	21	3	3	10
	0009			00			
	000A			42			
√	000B		MVI A,4E	3E	2	2	7
	000C			4E			
√	000D		OUT C2	D3	2	3	10
	000E			C2			
√	000F		MVI A,37	3E	2	2	7
	0010			37			
√	0011		OUT C2	D3	2	3	10
	0012			C2			

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C0	42	00	37	00	00	00	00	00	0A	00	00	00	00	00	36	00
D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Have a look at the I/O port addresses C000h,C002H,C008H,C00EH.