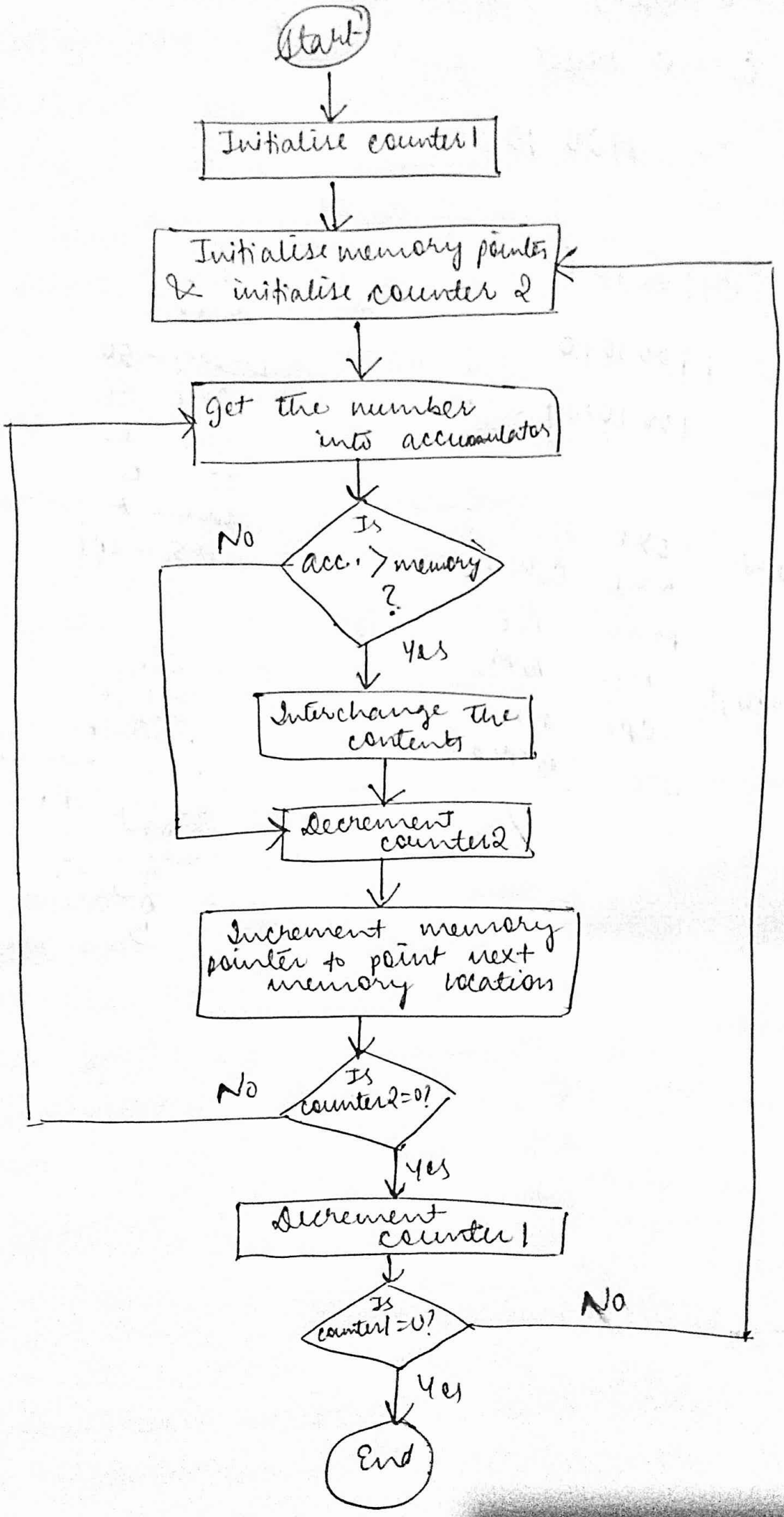
Experiment - 6

AIM - Write a program to arrange a number in ascending order using 8085 & verify

ALGORITHM/THEORY -

1. Counter 1 and counter 2 loaded with counters and memory points initialised
2. The number loaded into accumulator
3. Compare accumulator and memory pointer
4. If carry is set, then go to step 7
5. The contents are interchanged.
6. Memory pointer incremented to point next memory location
7. Counter 2 decremented
8. If counter 2 is not zero, go to step 2
9. Counter 1 decremented
10. If counter 1 is not zero, go to step 1.
11. Terminate.

FLOWCHART :


Start
Initialise counter1
Initialise memory pointer & initialise counter 2
Get the number into accumulator
Is acc. > memory ?
No
Yes
Interchange the contents
Decrement counter2
Increment memory pointer to point next memory location
Is counter2=0?
No
Yes
Decrement counter1
Is counter1=0?
No
Yes
End

Code:

```
#ORG 2000H
LDA F100          // Load count from F100 to acc.
DCR A             //Decrement A by 1
MOV C,A           // A => C
MOV B,C           // B => C
LXI H,F200        // HL <= F200
up: MOV A,M       // [HL] => A
INX H             //HL+1 => HL
CMP M             // compare reg M to A
JC down           // If A<M jump condition is true
MOV D,M           // M => D
MOV M,A           // A => M
DCX H             // HL-1 => HL
MOV M,D           // D <= M
INX H             //HL+1 => HL
down: DCR B       //decrement B by 1
JNZ up            // ~~decre~~ Jump until B=00
DCR C             //decrement C by 1
JNZ 2005          // Jump until C=00
RST 1             // Terminate

#ORG F100H        // Store counter at the address
# DB 04           // Store counter
# ORG F200H       //Store nos. at the address
# DB DD,CC,BB,AA  //Store nos.
```

Input: F100-04H, F200-DDH, F201-CCH, F202-BBH, F203-AAH

Output: F200-AAH, F201-BBH, F202-CCH, F203-DDH

Assembler

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	2000		LDA F100	3A	3	4	13
	2001			00			
	2002			F1			
√	2003		DCR A	3D	1	1	4
√	2004		MOV C,A	4F	1	1	4
√	2005		MOV B,C	41	1	1	4
√	2006		LXI H,F200	21	3	3	10
	2007			00			
	2008			F2			
√	2009	UP	MOV A,M	7E	1	2	7
√	200A		INX H	23	1	1	6
√	200B		CMP M	BE	1	2	7
√	200C		JC DOWN	DA	3	3	10
	200D			14			
	200E			20			
√	200F		MOV D,M	56	1	2	7
√	2010		MOV M,A	77	1	2	7
√	2011		DCX H	2B	1	1	6
√	2012		MOV M,D	72	1	2	7

Simulate

Start From → 0000

Run all At a Time | Step By Step

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
200F	56
2010	77
2011	2B
2012	72
2013	23
2014	05
2015	C2
2016	09
2017	20
2018	0D
2019	C2
201A	05
201B	20
201C	CF
F100	04
F200	DD
F201	CC
F202	BB
F203	AA

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

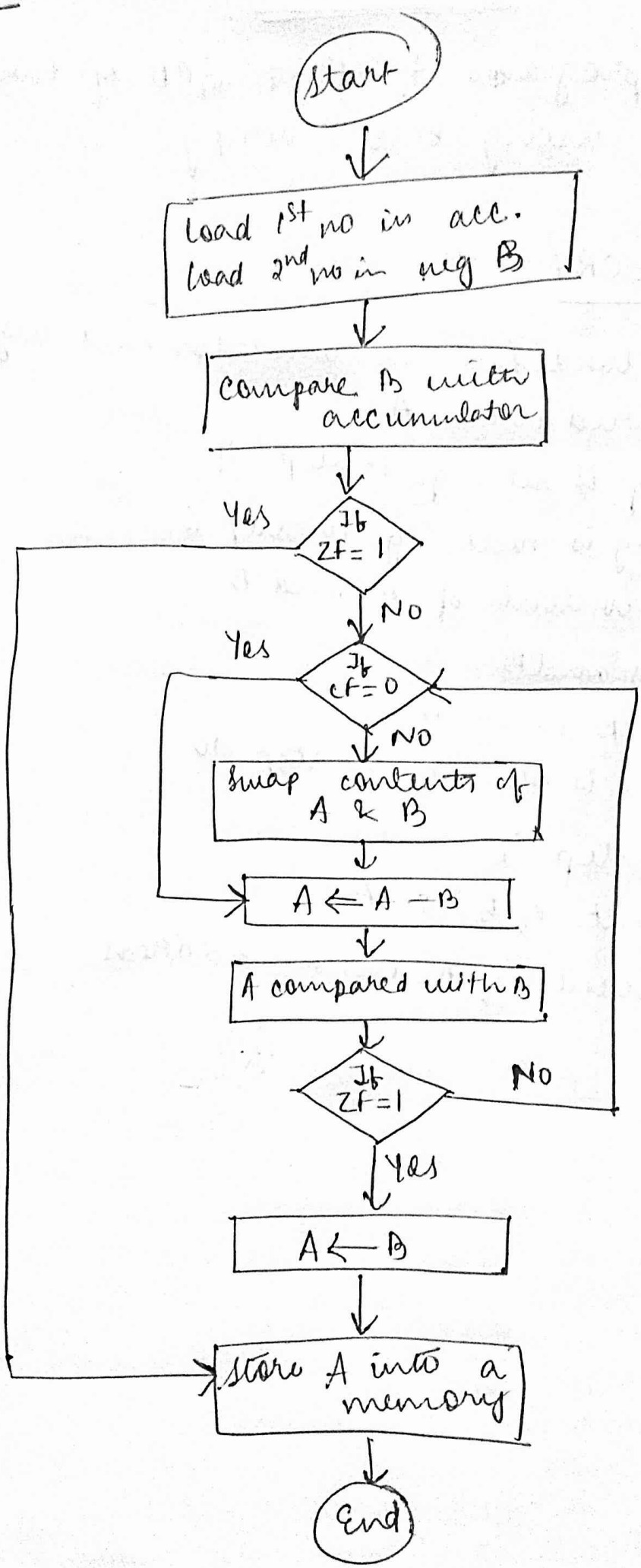
Experiment - 8

AIM: Write a program to find GCD of two numbers using 8085 & verify.

ALGORITHM/ THEORY:

1. The numbers loaded in accumulator and regB
2. Reg B compared with A.
3. If zero flag is set, go to step 11
4. If carry flag is reset, go to step 6
5. Swap the contents of A and B
6. Subtract A and B
7. compared B with A
8. If zero flag is set, go to step 10
9. Jump to step 4
10. Move content of B into A
11. Store content of A in an address
12. Stop

FLOWCHART :


Start
Load 1st no in acc.
load 2nd no in reg B
Compare B with accumulator
Yes
If ZF = 1
No
Yes
If CF = 0
No
Swap contents of A & B
A ⇐ A − B
A compared with B
If ZF = 1
No
Yes
A ⇐ B
Store A into a memory
End

Code:

```
#ORG 2000H

MVI A,09        // Load first no in regA
MVI B,07        // Load second no in regB
CMP B           //Compare B to A
JZ down         // True if A = B
JNC shift       // True if A > B
MOV C,A         // A ⟹ C
MOV A,B         // A ⟸ B
MOV B,C         // C ⟸ B

shift: SUB B    // A-B ⟹ A
CMP B           // Compare B to A
JZ move         // True if A=B
JMP 2008        // Jump until A=B.

move: MOV A,B   // B ⟹ A
down: STA F200  //A ⟹ [address]
RST 1           //Terminate
```

Input: A-09H, B-07H

Output: A-01H, F200-01H.

Assembler

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	2000		MVI A,09	3E	2	2	7
	2001			09			
√	2002		MVI B,07	06	2	2	7
	2003			07			
√	2004		CMP B	B8	1	1	4
√	2005		JZ DOWN	CA	3	3	10
	2006			17			
	2007			20			
√	2008		JNC SHIFT	D2	3	3	10
	2009			0E			
	200A			20			
√	200B		MOV C,A	4F	1	1	4
√	200C		MOV A,B	78	1	1	4
√	200D		MOV B,C	41	1	1	4
√	200E	SHIFT	SUB B	90	1	1	4
√	200F		CMP B	B8	1	1	4
√	2010		JZ MOVE	CA	3	3	10
	2011			16			
	2012			20			

Terminate

Simulate

Start From → 2000

Run all At a Time | Step By Step

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	01	0	0	0	0	0	0	0	1
Register B	01	0	0	0	0	0	0	0	1
Register C	01	0	0	0	0	0	0	0	1
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	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	54	0	1	0	1	0	1	0	0

Type	Value
Stack Pointer(SP)	FFFE
Memory Pointer (HL)	0000
Program Status Word(PSW)	0154
Program Counter(PC)	0009
Clock Cycle Counter	244
Instruction Counter	38

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

For SIM instruction

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

For RIM instruction

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

No. Converter Tool :

Hexadecimal	Decimal	Binary
0	0	0