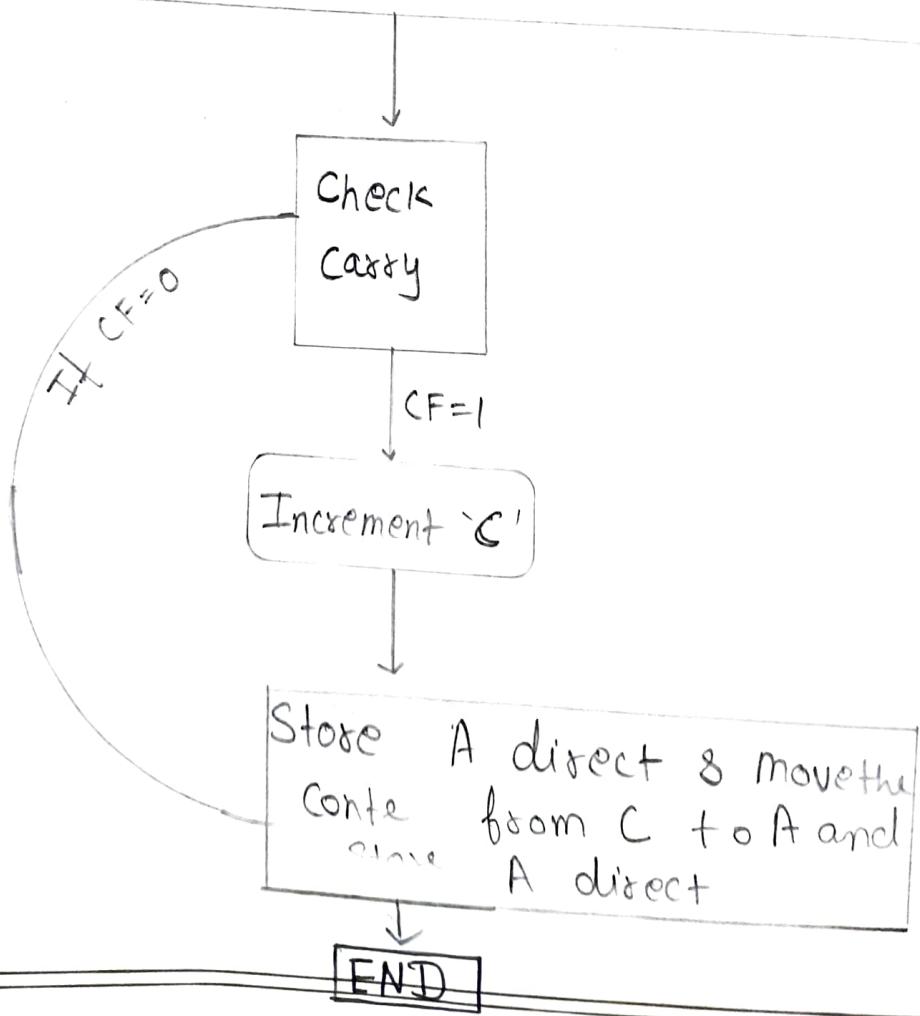
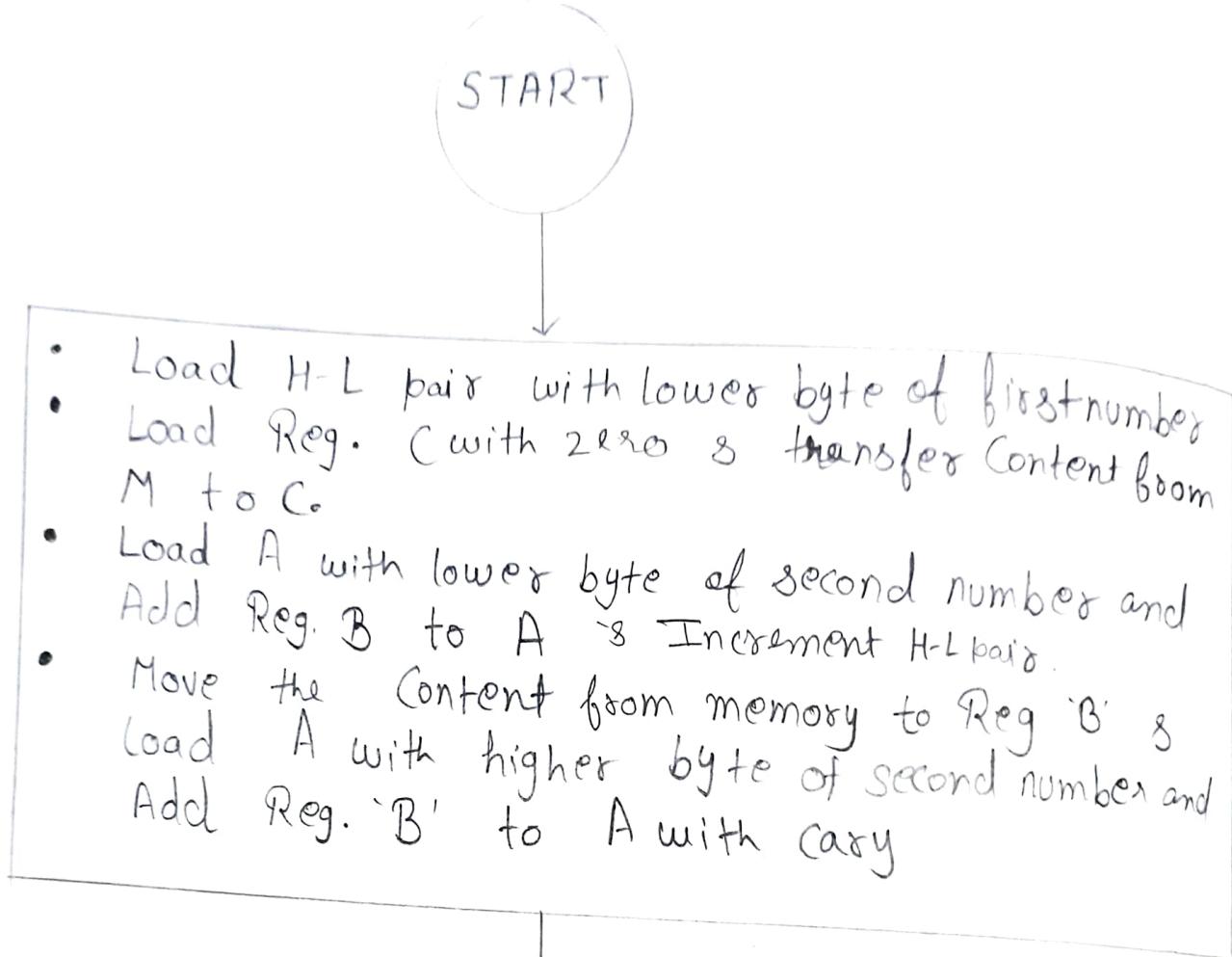


Flow-Chart of Hangoan :-



Experiment No. :- 1

- Aim :- Write an ALP for addition of two multibyte Numbers.
- Apparatus used :- 8085-microprocessor kit.
- Code of the program

Mnemonics

Comments

LXI H, 4000H	; Load H-L pair with the lower byte of 1st number
MVI C, 00H	; Clear 'C' register
MOV B, M	; Load memory content into 'B' Register
LDA 4003H	; Load A with lower byte of 2nd number
ADD B	; Add 'b' to A.
STA 4007H	; Store accumulator direct to 4007H.
INX H	; Increment H-L pair
MOV B, M	; Load memory content into 'b' register
LDA 4004H	; Load accumulator with higher byte of 2nd number
ADC B	; Add 'b' with 'A' with carry
STA 4008H	; Store accumulator direct to 4008H.
INX H	; Increment H-L pair
MOV B, M	; Load memory content into 'b' register
LDA 4005H	; Load accumulator with higher byte of 2nd no.
ADC B	; Add 'b' to 'A' with carry
JNC Skip	; Jump to 'skip' if carry is non-zero.
INR C	; Increment register 'C'.
skip: STA 4009H	; Store accumulator direct to 4009H
MOV A, C	; Load content of register 'C' into accumulator



Teacher's Signature _____

STA 400AH ; Store Content of register direct to 200AH.
 HLT ; END

Op-Code

Address

Op-code

2000

21, 00, 40

2003

0E, 00

2005

46

2006

3A, 03, 40

2009

80

200A

32, 07, 40

200D

24

200E

46

200F

3A, 04, 40

2012

88

2013

32, 08, 40

2016

24

2017

46

2018

3A, 05, 40

201B

88

201C

D2, 20, 20

201F

0C

2020

32, 09, 40

2023

79

2024

32, 0A, 40

2026

76

• Input :-

Address (H)

4000

4001

4002

4003

4004

4005

4006

4007

4008

4009

400A

Data (H)

10

20

30

0A

0B

0C

00

00

00

00

• Output :-

Address (H)

4000

4001

4002

4003

4004

4005

4006

4007

4008

4009

Data (H)

10

20

30

0A

0B

0C

00

1A

1B

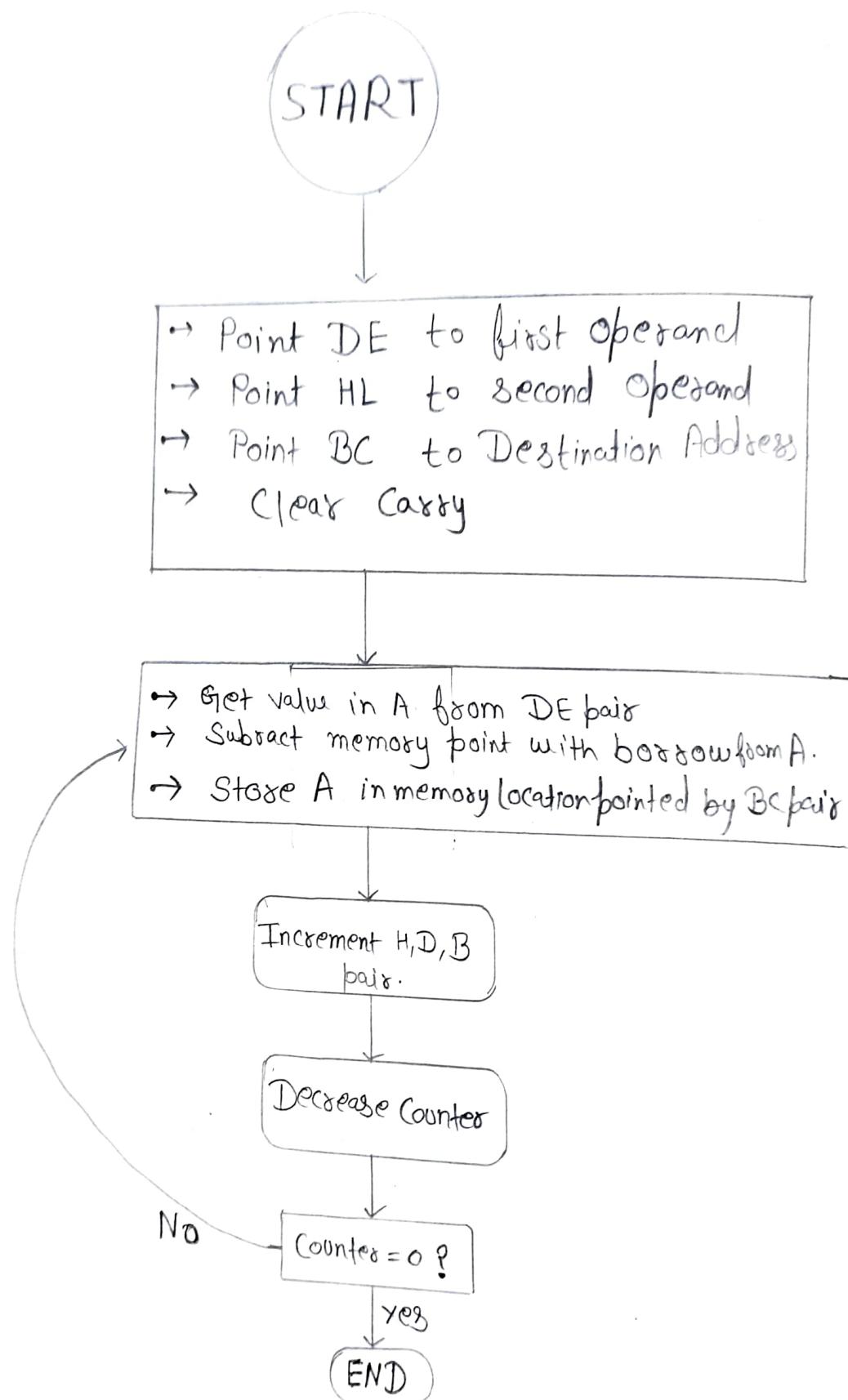
1C

400A

00

- Result :- Addition of two 3-byte Numbers stored at location 4000, 4001, 4003, for first numbers 4004, 4005, 4006 for second Number is done using the 8085 ALP and the Sum is stored at from memory location 4007 to 4009 and carry is stored at 400A.

Flow-Chart & Program



Experiment :- 2

- Aim :- Write an ALP for Subtraction of two multi-byte numbers.
- Code :-

Mnemonics	Comments
LXI D , 8001H	; Point to first operand
LXI H , 8004H	; Point to second operand
LXI B , 8050H	; Point destination address
STC	; Set the Carry
CMC	; Complement carry
Loop: LDA A X D	; Load A from memory pointed by DE
SBB M	; Subtract memory element and borrow
STA X B	; Store the result into memory pointed by BC pair
INX H	; Increment H-L pair
INX D	; Increment D-E pair
INX B	; Increment B-C pair
LDA 8000H	; Load the size into A.
DCR A	; Decrease A by 1
STA 8000H	; Store updated size in memory
JN2 Loop	; If Z=0, jump to 'Loop'
HLT	; END

- Op-codes :-

Address	Op-code
2000	11, 01, 80
2003	21, 04, 80
2006	01, 50, 80
2007	37
2008	3F
2009	1A
200A	9E
200B	02
200C	23
200D	13
200E	03
200F	3A, 00, 80
2012	3D
2013	32, 09, 20
2016	76
2019	

• Discussion :- We are using 3-byte numbers. The numbers are stored into memory locations 8001 and 8004H. Counter is being stored at 8000H. The result is stored at 8050H.

• Input :-

Address (H)

8000

8001

8002

8003

8004

8005

8006

8007

8008

8009

Data (H)

03

10

20

30

01

02

03

00

00

00

• Output :-

Address (H)

8050

8051

8052

8053

8054

Data (H)

0F

1E

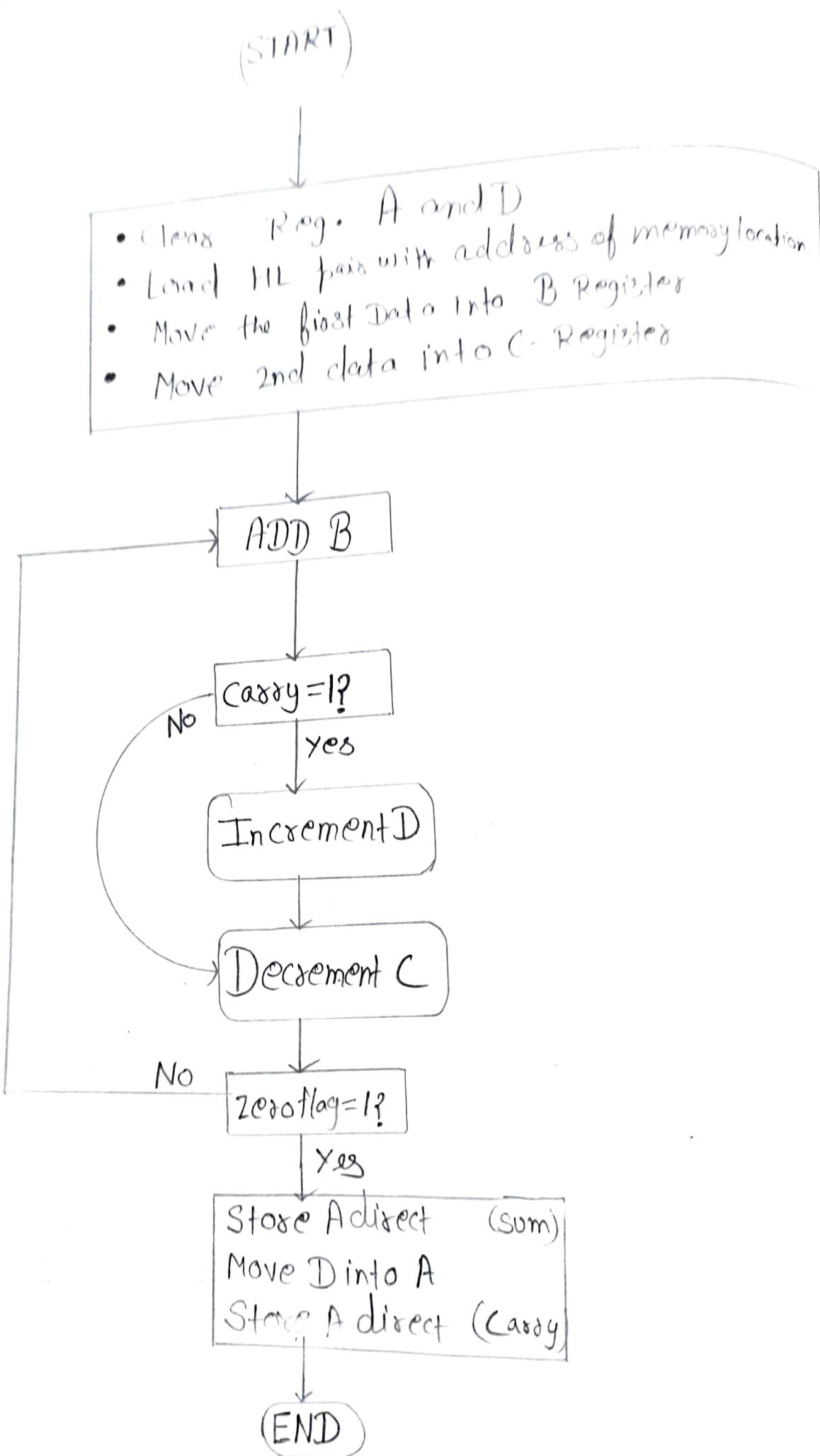
2D

00

00

• Result :- Subtraction of two multi-byte numbers
are successfully evaluated using 8085 ALP
and also successfully verified.

Flow-Chart of Program :-



Experiment :- 3

- Aim :- To perform the multiplication of two 8-bit numbers.
- Apparatus Used :- 8085 microprocessor kit
- Code :-

Mnemonics	Comments
SUB A	; Clear A
MOV D,A	; Clear D by transfer of A into D
LXI H, 2050	; Point to first operand
MOV B,M	; Get first number into B
INX H	; Increment H-L pair
MOV C,M	; Get second number into C
Loop: ADD B	; Add B to A.
JNC Skip	; Jump to 'Skip' if CY=0
INRD	; Increase 'D'
Skip: DCR C	; Decrease the Counter
JNZ Loop	; Jump to 'Loop' if zero flag ≠ 0
STA 2052	; Store A direct
MOV A,D	; Move data of D into A.
STA 2053	; Store A direct
HLT	; END the program.

• Op-Code :-

Address	Op-Code
2000	97
2001	57
2002	21, 50, 20
2005	46
2006	23
2007	4E
2008	80
2009	C2, 0D, 20
200C	14
200D	0D
200E	C2, 08, 20
2011	32, 52, 20
2014	7A
2015	32, 53, 20
2018	76

- Discussion :- Two 8-bit numbers are stored at memory location 2050 and 2051 and the result is stored at 2052.

- Input

Address (H)	Data (H)
2050	FF
2051	FF

a) Output :-

Address

2052

2193

Bytes

•1

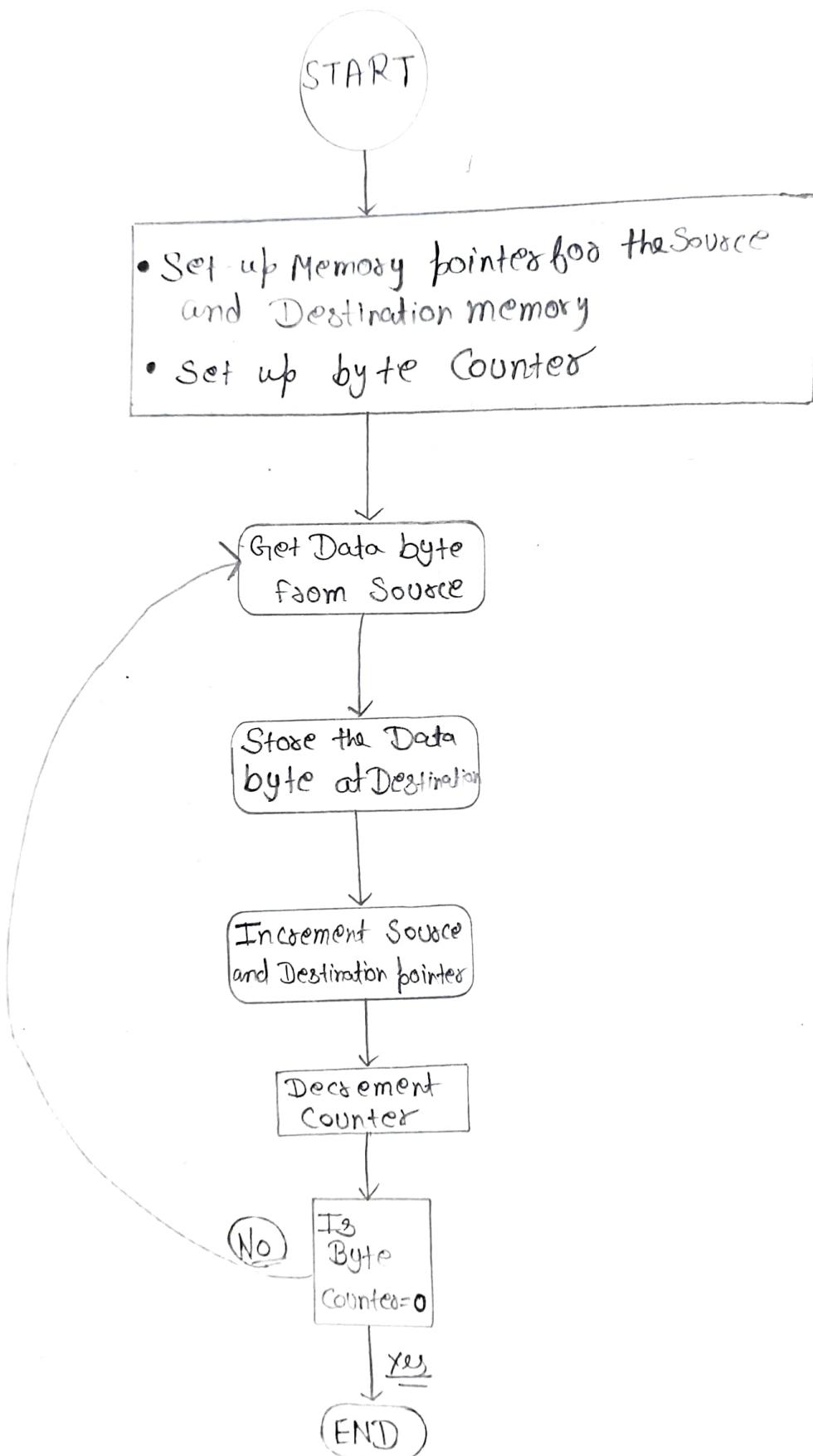
FF

b) Result :- Thus the program to multiply two 3-bit numbers was executed and result was verified.

(H) MOOTAM

Teacher's Signature _____

• Flowchart of program :-



Experiment :- 4.

- Aim :- Write an ALP to perform transfer of a block of Data.
- Apparatus Used :- 8085 - microprocessor kit
- Code :-

Mnemonics	Comments	Hex-codes Hex-codes
MVI C, 10	; Store counter in reg. C	; 0E, 10
LXI H, 2050	; Get from Memory pointer of Source; 21, 50, 20	
LXI D, 3050	; Get Destination pointer ; 11, 50, 30	
Loop: MOV A, M	; Get first byte into A	; 7E
STA D	; Store A into destination	; 12
INX H	; Increment H-L pair	; 23
INX D	; Increment D-E pair	; 13
DCR C	; Decrement Reg. C	; 0D
JN2 Loop	; Jump if C ≠ 0	; C2, 08, 20
HLT	; END	; 76

Op-codes

Address

Op-Code

2000

OE, 10

2002

21, 50, 20

2005

11, 50, 30

2008

7E



Teacher's Signature _____

2009	12
200A	23
200B	13
200C	0D
200D	C2, 08, 20
2010	76

• Discussion :- 10 numbers stored at memory location
 Starting at 2050 are transferred to memory
 location 3050 and onwards.

• Input

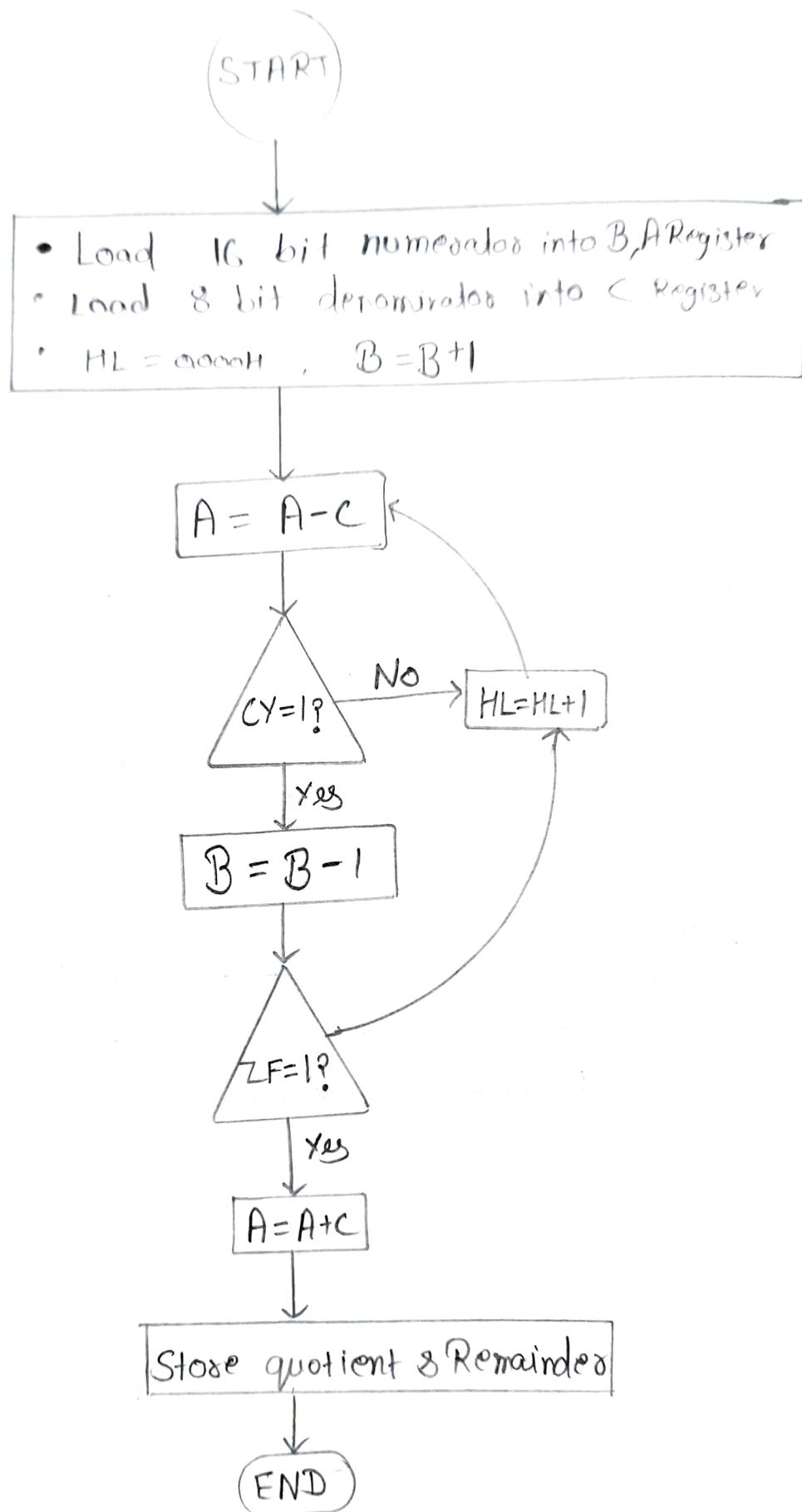
Address(H)	Data(H)	Address(H)	Data(H)
2050	1A	3050	FF
2051	00	3051	FF
2052	FF	3052	FF
2053	CC	3053	FF
2054	10	3054	FF
2055	11	3055	FF
2056	2S	3056	FF
2057	1A	3057	FF
2058	B1	3058	FF
2059	0C	3059	FF

• Output :-

Address(W)	Data(W)
3050	1A
3051	00
3052	FF
3053	CC
3054	10
3055	11
3056	25
3057	1A
3058	B1
3059	0C

• Result :- Program for transfer of block of data was executed and 10 distinct data from 10 different memory address to another 10 different memory location.

• Flow-Chart of Program :-



Experiment :- 5

Aim :- Write an ALP to divide 16-bit number by an 8-bit number.

Apparatus Used :- 8085 microprocessor kit

Code

Mnemonics

Comments

LXI H, 4000H

; Point 4000H address

MOV A, M

; Store the lower order byte

INX H

; Increase the H-L pair to point next

MOV B, M

; Store the higher order byte

INX H

; Increment H-L pair

MOV C, M

; Load the denominator

INR B

; Increase B register

LXI H, 0000H

; Store 0000H into HL pair

Loop: SUB C

; Subtract C from accumulator

JC Skip

; Jump to 'Skip' when CY=1

INCR: INX H

; Increase quotient part

JMP Loop

; Jump to Loop

Skip: DCR B

; Decrease B

J2 ~~Result~~

; Jump to store when Z=1

JMP INCR

; Jump to 'INCR'

Result: ADDC

; Add 'C' into 'A'

XCHG

; Swap DE and HL pair Content

LXI H, 4000H

; Load the destination address

MOV M,E	; Store the lower order quotient
INX H	; Increase H-L pair
MOV M,D	; Store the higher order quotient
INX H	; Increase H-L pair
MOV M,A	; Store the remainder
HLT	; Terminate the program.

Op- Codes:-

Address

Op-code

2000

21, 00, 40

2003

7E

2004

23

2005

46

2006

23

2007

4E

2008

34

2009

21, 00, 00

200C

91

200D

DA, 14, 20

2010

23

2011

C3, 0C, 20

2014

05

2015

CA, 1B, 20

2018

C3, 10, 20

201B

89

201C

EB

201D

21, 03, 40

2020

73

2021

23

2022

72

2023

23

2024

77

2025

76

• Discussion :- In this program we are taking the 16-bit numbers from 4000H and 4001H. The 4000H is holding the lower order byte and 4001H is holding the higher order byte. The 4002 is holding the 8-bit number. After Dividing the 16-bit, quotient is stored at location 4005H and 4006H. The remainder is stored at 4007H.

• Input :- We taken 16 bit number = 94S (03B1H), 8bitnumber =
 Result should be Quotient = 18 (12H)
 Remainder = 45 (2D)

Address

Data(H)

Data(Dec)

4000

B1

945

4001

03

4002

32

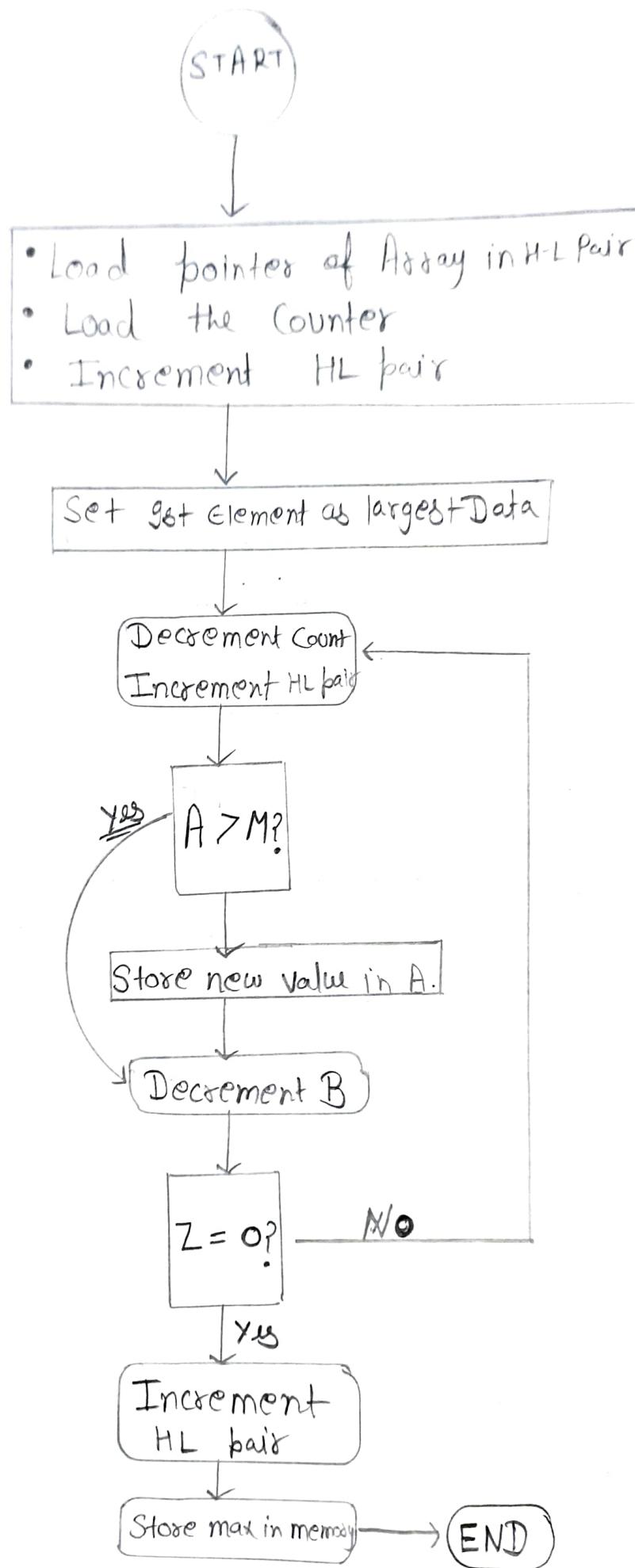
50

• Output :-

Address(H)	Data(Hex)	Data(Dec)
4005	12	18
4006	00	0
4007	2D	45
4008	00	0

• Result :- ALP to divide a 16-bit number by a 8-bit number have been executed and verified.

Flow-Chart of Program : (To find Maximum number).



Experiment : 6

- Aim :- Write an ALP to find Minimum and maximum among N numbers.

- Apparatus Used :- 8085 microprocessor

- To find Maximum among N-number

- Code

Mnemonics	Comments
LXI H, 4000H	; Set pointer of array
MOV B, M	; load the Count
INX H	; increment the memory
MOV A, M	; Set 1st element as largest data
DCR B	; Decrement the count
loop: INX H	; Increment HL pair
CMP M	; Compare A and M
JNC Skip	; if $A > M$ go to 'skip'
MOV A, M	; Set the new value of largest
skip: DCR B	; decrement count
JN2 Loop	; go to 'Loop' if count $\neq 0$
INX H	; Increment HL
MOV M, A	; Store the largest no. in memory
HLT	; terminate the program

Teacher's Signature _____

- Op codes

Address

Op-Code

2000

21, 00, 40

2003

46

2004

23

2005

7E

2006

05

2007

23

2008

BE

2009

D2, 0D, 20

200C

7E

200D

05

200E

C2, 07, 20

2011

23

2012

7F

2013

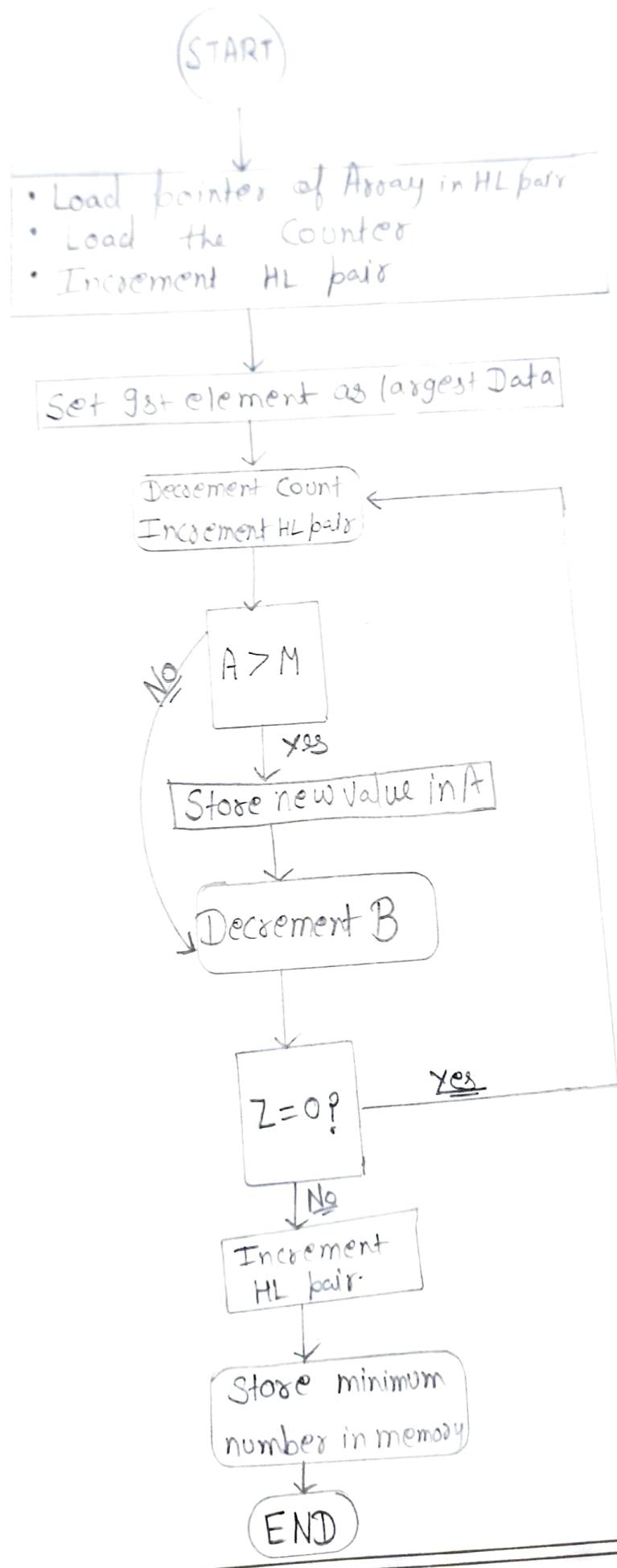
76

- Input

<u>Address(H)</u>	<u>Data(H)</u>	<u>Address(H)</u>	<u>Data(H)</u>
4000	0A	4006	11
4001	10	4007	13
4002	EF	4008	25
4003	EC	4009	2F
4004	EA	4009	9F
4005	EB		

Flow-chart of program: (To find minimum number).

Expt. No.



- Output :-

Address(H)

400B

Data(H)

EF

- Discussion :- In this program we are storing Counter at location 4000H which will find the maximum number b/w the Next locations as according to Counter and result is stored just after the input numbers.
In this case we have taken Counter = 0A (10D). So the maximum number of location 4001 to 400A is stored at 400B.

2). To find minimum among N numbers

- Code :-

Mnemonics

Comments

LXI H, 4000H

; Set the pointer of array

MOV B, M

; Load the Count

INX H

; Increment the memory

MOV A, M

; Set 1st element as largest

DCR B

; Decrement the Count

Loop: INX H

; Increment HL pair

CMP M

; Compare A and M

JL Skip

; if A < M go to 'Skip'

MOV A, M

; Set the new value

Skip: DCR B

; Decrement Count

JNZ Loop

; Go to 'Loop' if Count = 0

INX H
Mov M,A
HLT

; Increment HL pair
; Store the largest no. In memory
; to terminate the program.

• Op-codes :-

Address

2000

2003

2004

2005

2006

2007

2008

2009

200C

200D

200E

2011

2012

2013

Op-code

21, 00, 40

46

23

7E

05

23

BE

DA, OD, 20

7E

05

C2, 07, 20

23

77

76

• Input :-

Address (H)	Data (H)	Address (H)	Data (H)
4000H	09	4005H	03
4001H	05	4006H	09
4002H	EF	4007H	EE
4003H	FF	4008H	CA
4004H	10	4009H	AC

• Output :-

Address(H)

400A

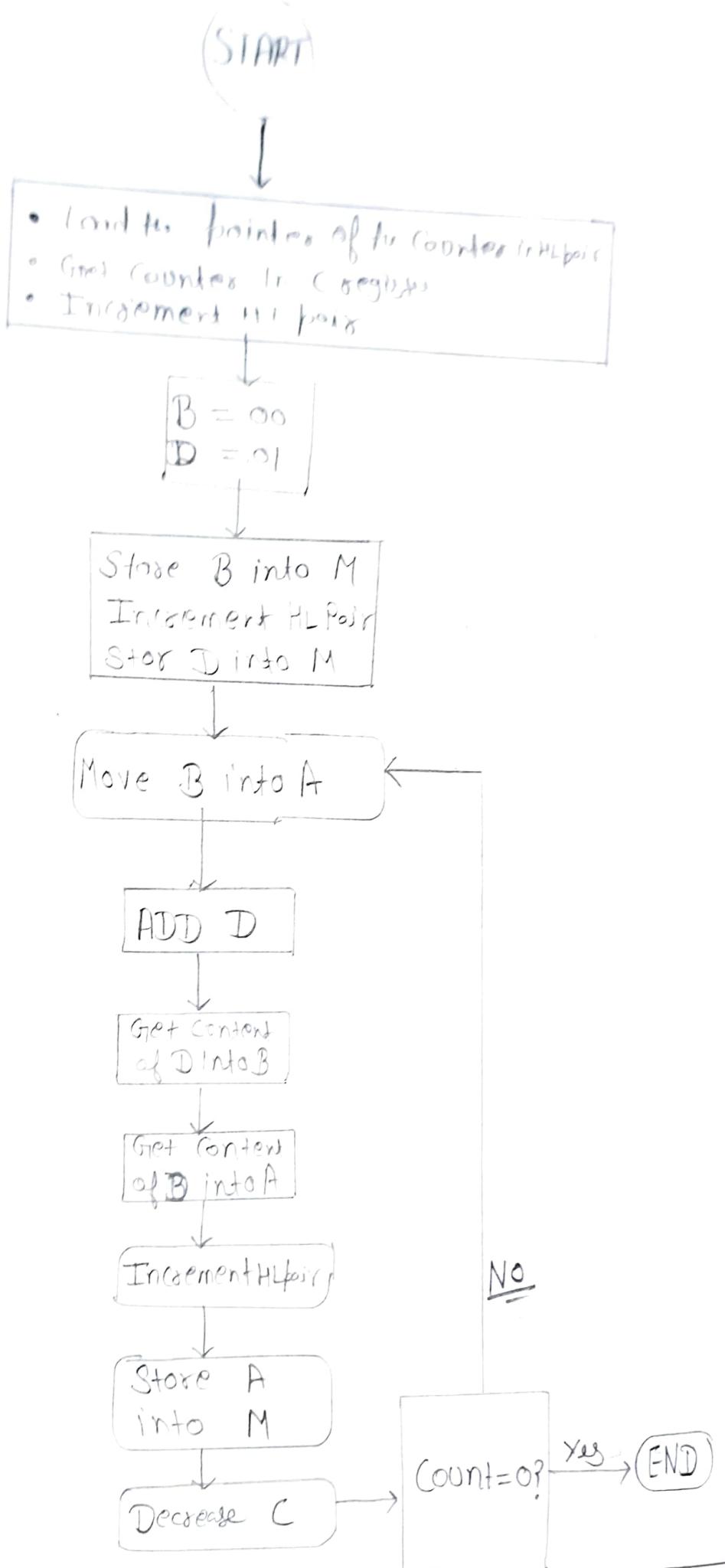
Data(H)

03

- Discussion :- In this program we are storing Counter at location 4000H which will find the minimum number b/w the next locations as according to counter and result is stored just after the input numbers. In this case we have taken Counter = 9. So the minimum number of location 4001 and 4009 is stored at 400A.

- Result :- ALP for finding the Maximum and minimum numbers among N-number is executed and Verified.

Flowchart of the program:



Experiment :- 7

- Aim :- To write an ALP to display terms of Fibonacci series.
- Apparatus Used :- 8085 - microprocessor kit
- Code

Mnemonics	Comments
LXI H, 2050	; Pointer to Counter location
MOV C, M	; Get Counter in C
INX H	; Increment HL pair
MVI B, 00	; Clear B register
MVI D, 01	; Move 01 into D register
MOV M, B	; Store first term into M
INX H	; Increment HL pair
MOV M, D	; Store Second term into M
Loop: MOV A, B	; Get first content of B into A
ADD D	; ADD A with Content of D.
MOV B, D	; Get Content of D into B
MOV D, A	; Get Content of A into D
INX H	; Increment HL pair
MOV M, A	; Store A into M
DCR C	; Decrease Counter
JNZ Loop	; go to 'Loop' if count ≠ 0
HLT	; END

- Op-codes :-

Address	Op-code
2000	21, 50, 20
2003	4E
2004	23
2005	06, 00
2007	16, 01
2008	70
2009	23
200A	72
200B	78
200C	82
200D	42
200E	57
200F	23
2010	77
2011	0D
2012	C2, 0C, 20
2015	76

- Discussion :- This program will generate the terms of Fibonacci Series. The counter is stored at memory location 2050. If OA is stored at location then 10 terms then 10 fibonacci terms will be stored at 2051 and onwards.

- Input :-

Address (H)

2050

Data (H)

0A

- Output :-

Address (H)

2050

Data (H)

0A

2051

00

2052

01

2053

01

2054

02

2055

03

2056

05

2057

08

2058

0D

2059

45

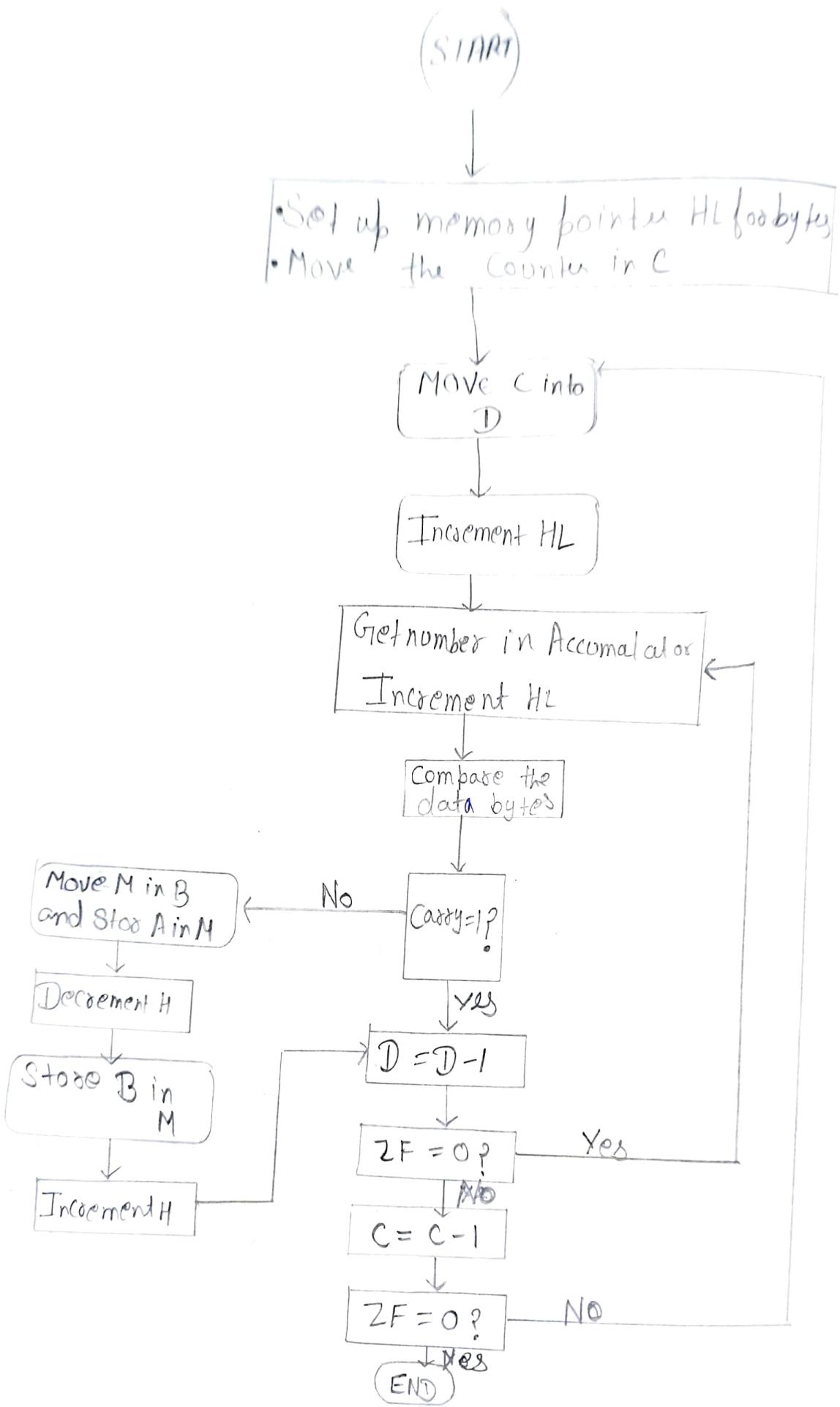
205A

22

- Result :- ALP for generating term of Fibonacci series is executed and verified.

Teacher's Signature

Flow chart of program



Experiment :- 8

- Aim :- Write an ALP for sorting of number in Ascending & Descending Order.

- Apparatus used :- 8085- microprocessor kit

- Ascending - order program :-

- Code :-

Mnemonics

Comments

LXI H, 2050

; Set up memory pointer for bytes

MOV C, M

; Get Counter into C register

DCR C

; Decrement Counter

Loop1: MOV D, C

; Get content of C into D

INX H

; Increment HL pair

Loop2: MOV A, M

; Get first byte

INX H

; Increment HL

CMP M

; Compare A with M

JC 2012

; If A < M Jump to '2012'

MOV B, M

; Get another number in B

MOV M, A

; Store A at M

2012: DCR D

; Decrement D ~~at M~~

JNZ Loop2

; If D ≠ 0, go to 'Loop2'

DCR C

; Decrement C

JNZ Loop1

; If C ≠ 0, go to 'Loop1'

RST

; Terminate the program

- Discussion :- This ALP will sort N numbers in Ascending order from When N is stored in location 2050 the next N numbers onwards 2051 will be sorted in Ascending order. Here we have stored OS in location 2050 and next 5 numbers in location 2051 to 2055 are sorted in Ascending order.

- Input :-

Address (H)	Data (H)
2050	05
2051	9A
2052	F1
2053	DS
2054	16
2055	FF

- Output :-

Address (H)	Data (H)
2050	05
2051	16
2052	9A
2053	DS
2054	F1
2055	FF

Teacher's Signature

Descending - order program :-

Code

Mnemonics

LXI H 2050

MOV C, M

DCR C

Loop1: MOV D, C

INX H

Loop2: MOV A, M

INX H

~~DCR D~~ CMP M

JNZ 20 JNC 2012

MOV B, M

MOV M, A

DCX H

MOV M, B

INX H

2012: DCR D

JNZ Loop2

DCR C

JNZ Loop1

RST

Comments

; Set-up memory pointer for bytes

; Get counter into C register

; Decrement Counter

; Get content of C into D

; Increment HL pair

; Get first byte

; Increment HL pair

; Compare A with M

; If A > M, Jump to 2012

; Get ~~next~~ next number in Register

; Store A at M

; Decrement HL pair

; Store B into M

; Increment HL pair

; Decrement D

; If D ≠ 0 go to 'Loop2'

; Decrement C

; If C ≠ 0 go to 'Loop1'

; Terminate the program

Op-Codes :-

Address

2000

2003

2004

2005

2006

2007

2008

2009

200A

200D

200E

200F

2010

2011

2012

2013

2016

2017

201A

Op-code

21, 05, 20

4E

0D

51

23

7E

23

BE

D2, 12, 20

46

77

2B

70

23

15

C2, 07, 20

0D

C2, 05, 20

EF

Input :-Address(H)

2050

2051

2052

2053

Data(H)

06

FE

9A

F1



NOOTAN

Teacher's Signature _____

2054
2055
2056

15
11
16

Output:

Address(H)

Data(H)

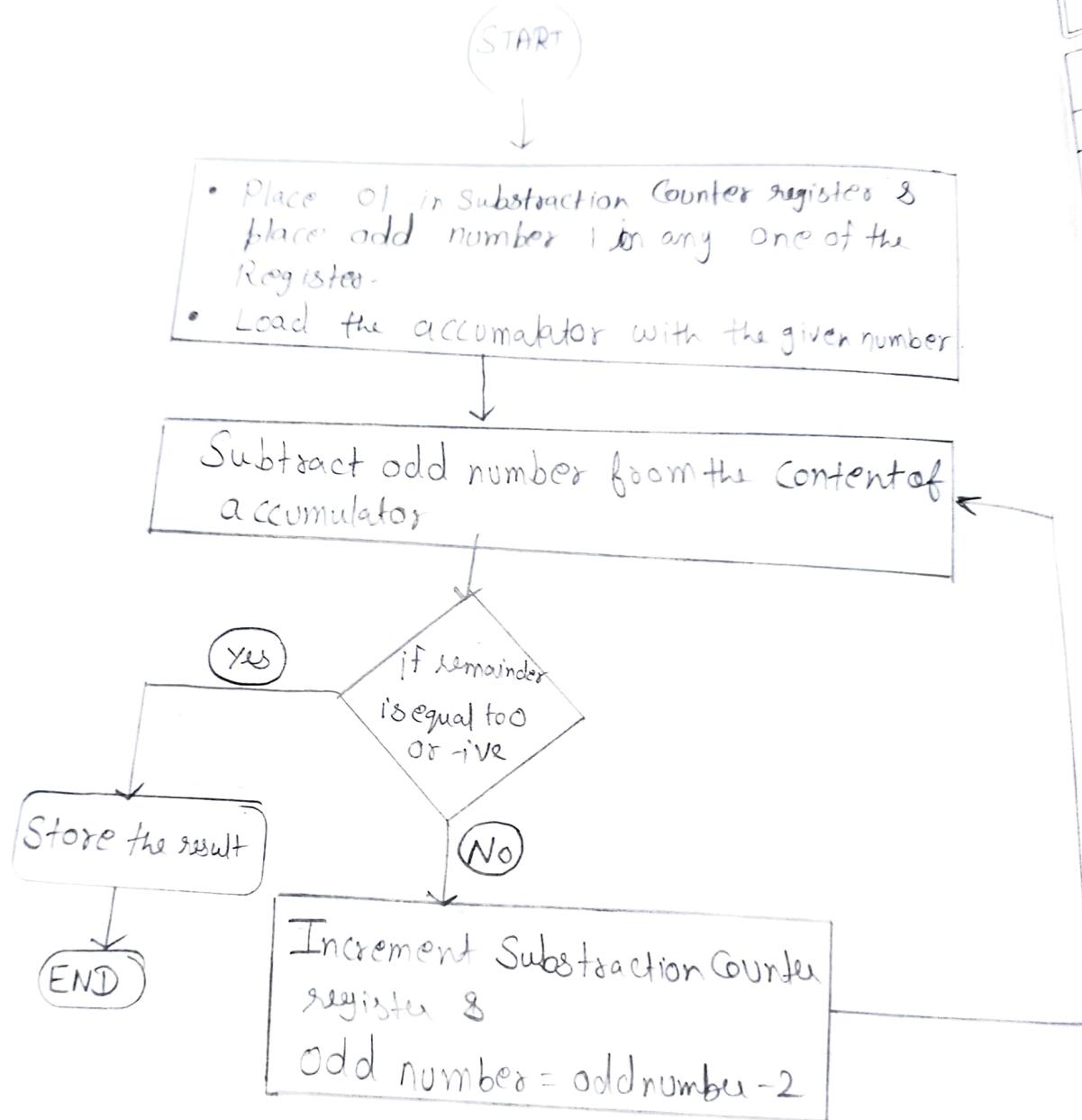
2050
2051
2052
2053
2054
2055
2056

06
FE
F1
D5
9A
16
11

Result :- ALP for sorting numbers in Ascending and Descending order is done and verified.

Teacher's Signature _____

Flow chart of Program:



9

Experiment : 9

Write an ALP to find square root of a number.

Aim :- Write an ALP to find square root of a number.

Apparatus used :- 8085 Microprocessor

Code :-

Nomenclature

MVI C, 01

MVI B, 01

MVI A, 31

UP : SUBB

JZ Down

INR C

INR B

INR B

JMP UP

DOWN : MOV A, C

STA 2050H

HLT

Comments

; Place 01 in reg. C

; Place odd number 1 in reg. B

; Load accumulator with the given number

; Subtract odd numbers from the accumulator

; If A content is zero, go ahead to Down

; Increment reg. C

; Increment reg. B

; Increment reg. B

; Repeat Subtraction

; Move the contents of C to A

; Store the result in the memory location 2050H

; END.

Op-codes :-

Address

2000

2002

2004

Op-codes

0E, 01

06, 01

3E, 31

9

2006
2007
200A
200B
200C
200D
2010
2011
2014

90
CA, 10, 20
0C
04
04
(3, 06, 20
79
32, 50, 20
76

Input :-

Address	Data (Dec)	Data (Hex)
Load the accumulator with the given no.	25	19
	36	24
	49	31
	100	64

Output :-

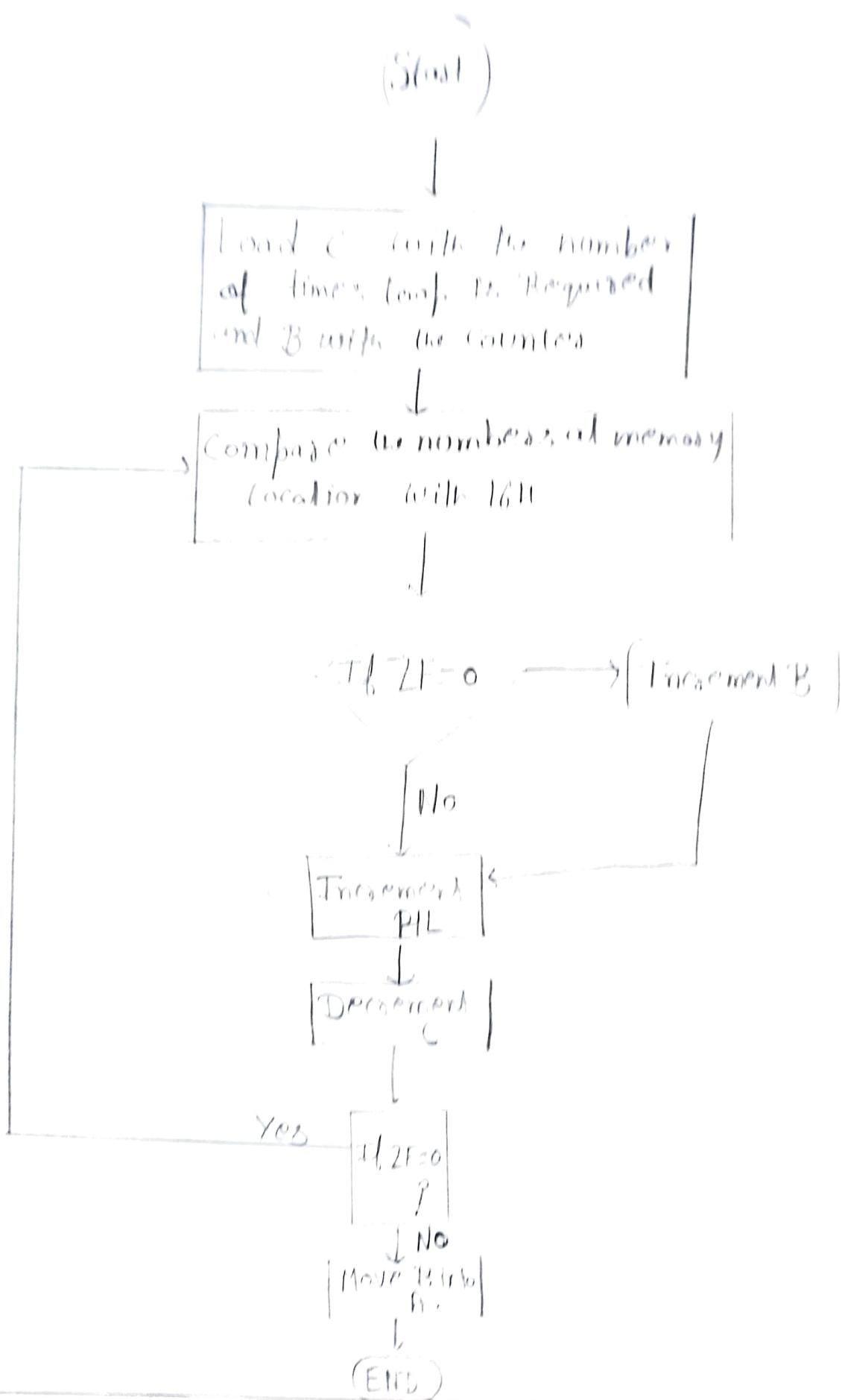
Address	Data (Dec)	Data (Hex)
2060H	5	5
	6	6
	7	7
	10	A

Result :- ALP to find square root of a number 18 done and verified.

Teacher's Signature _____

Theory:

Flowchart of program :

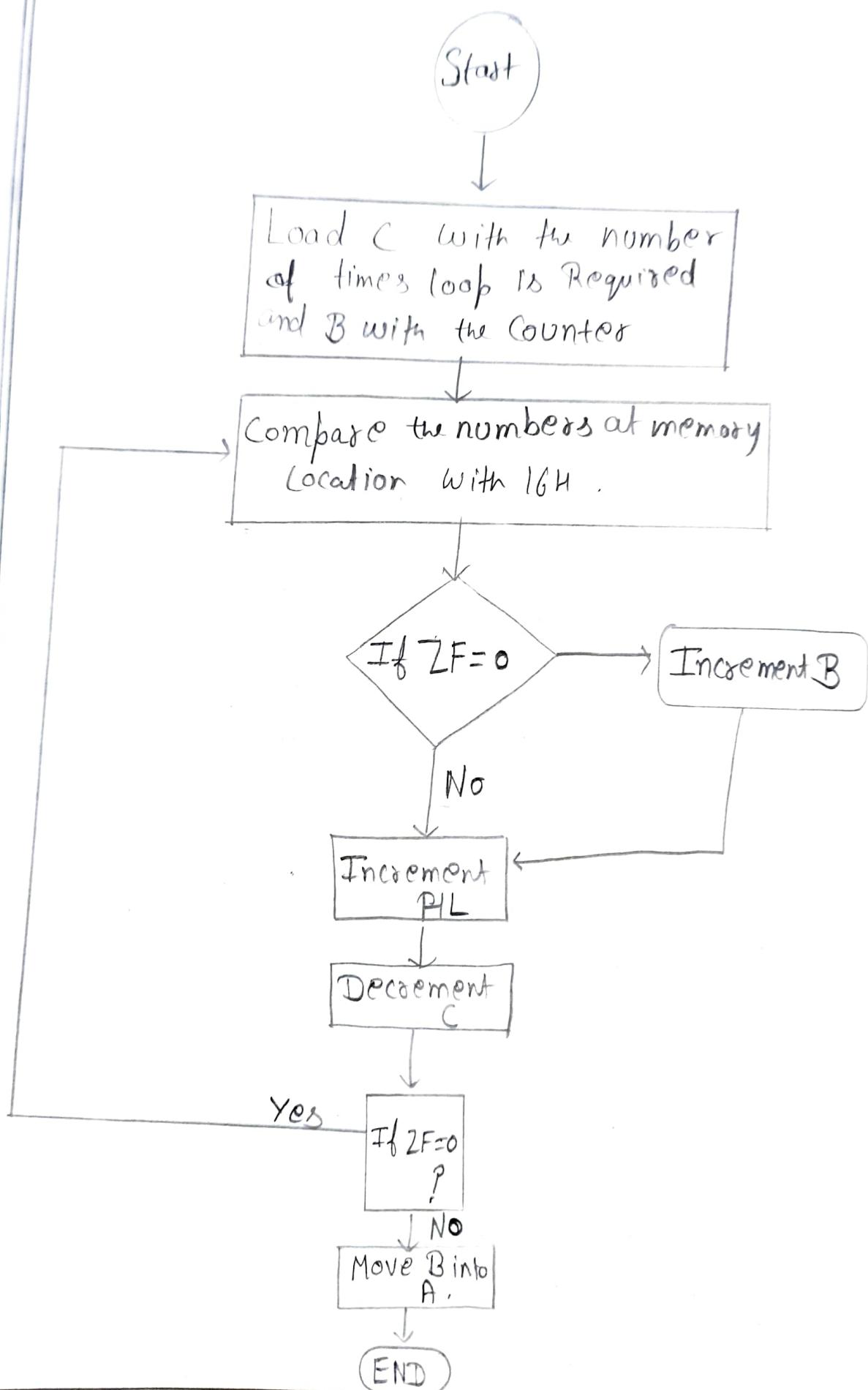


2003	OE
2004	OA
2005	O 6
2006	OO
2007	7E
2008	FE
2009	16
200A	C2
200B	OE
200C	20
200D	04
200E	23
200F	00
2010	C2
2011	27
2012	20
2013	78
2014	76

- Result :- Thus, searching for a number and counting its occurring time is also stored in A is done successfully.

Theory :-

Flowchart of program :-



Experiment :- 10

Aim :- Program to search number '16' which is stored b/w locations 2050H to 2059H.

Apparatus Used :- 8085 - microprocessor kit

Code :-

Mnemonics

Mnemonics	Comments
LXI H, 2050H	; Load the HL pair with address
MVI C, 0AH	; Load the register 'C'
MVI B, 00H	; Load the register 'B'.
NEXT: MOV A, M	; Move the data from mem. to reg. A.
CPI 16H	; Compare 16H with content of A.
JNZ skip	; Jump to 'skip' if not zero
INRB	; Increment B.
skip: INX H	; Increment HL pair
DCRC	; Increment C
JNZ Next	; Jump to 'Next' if not zero.
MOV A, B	; Move the contents of B to reg. A
HLT	; Terminate the program

Op-codes :-

Address (H)

2000

2001

2002

Data (H)

21

50

20

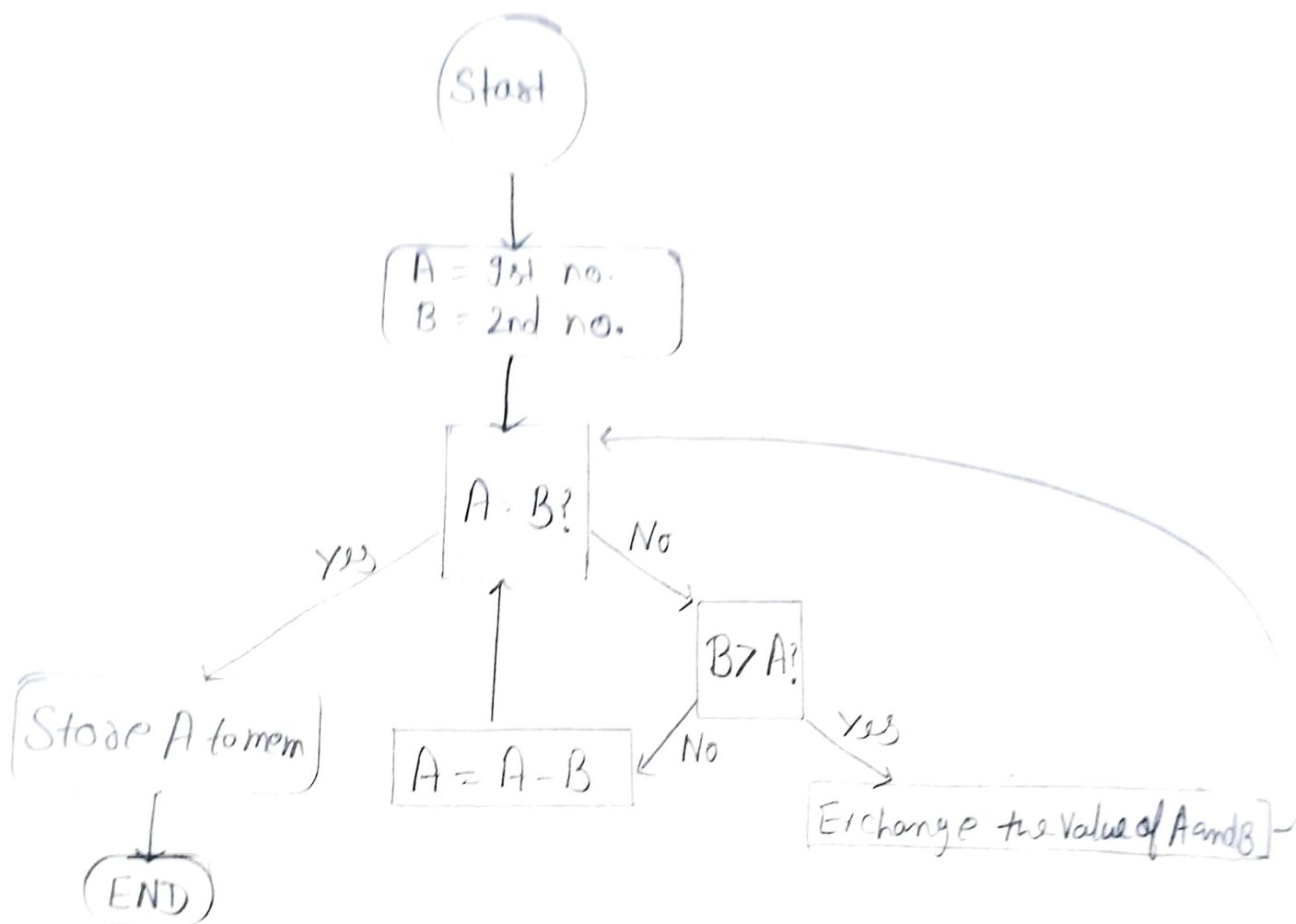


Teacher's Signature _____

2003	OE
2004	OA
2005	O 6
2006	OO
2007	7E
2008	FE
2009	16
200A	C2
200B	OE
200C	90
200D	04
200E	23
200F	00
2010	C2
2011	07
2012	20
2013	78
2014	76

Result :- Thus , searching for a number and counting its occurring time is also stored in A . is done successfully.

Flowchart of the program :-



HCF of two numbers

Experiment :- II

Aim :- Write an ALP to find GCD (Greatest Common divisor) of two numbers

Apparatus required :- 8085 - microprocessor kit

Code

Mnemonics	Comment
LXI H, 2050H	; Point to the first number
MOV A, M	; Load the first no. into A
INX H	; Point to next location
MOV B, M	; Load the second number
Loop: CMP B	; Compare B with A
JZ STORE	; Jump to STORE when zero flag is set
JC EXG	; If B > A go to 'EXG'
SUB B	; Subtract B from A
JMP Loop	; Jump to Loop
EXG MOV C,B	; Load C with B
MOV B,A	; MOV A to B
MOV A,C	; MOV C to A.
JMP Loop	; Jump to loop
STORE SIA 2054H	; Store the value into memory
HLT	; END

1) Input

Address

2050 H

2051 H

Data (Hex)

87 1 H

69 H

2) Output

2054 H

0 F (H)

(H.C.F)

2) Input

2050 H

6C (H)

2051 H

40 (H)

Output

2054 H

04 (H)

(H.C.F)

Op codes

Address

2000

2003

2004

2005

2006

2007

200A

200D

200E

2011

2012

2013

2014

2017

201A

Data(Hex)

21, 50, 20

7E

23

46

B8

CA, 17, 20

DA, 11, 20

90

C3, 06, 20

48

47

79

C3, 06, 20

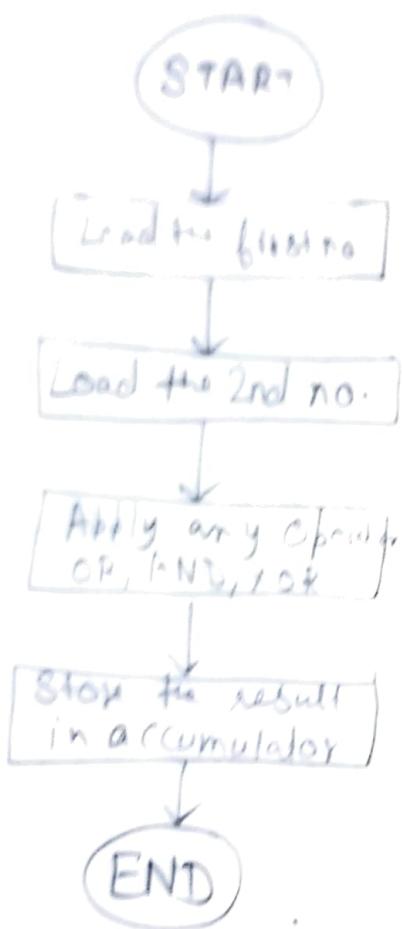
32, 54, 20

76

.

Result :- Thus we have found the GCD of two numbers using 8085 successfully and verified it.

Flowchart -



Truth table for AND operation

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Truth table for XOR operation

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

Experiment :- 12

Aim: Write a ALP to verify the truth tables of the logic gates.

Apparatus used :- 8085 microprocessor kit

Codes

for AND logic gate.

Mnemonics

LXI H, 2050H

MOV A, M

INX H

MOV B, M

ANA B

INX H

MOV M, A

HLT

Comments

; Load the HL pair with locations

; Move the content of Memory to A

; Increment the HL pair

; Move the content of memory to B

; AND operation of A with B.

; Increment ^{HL pair} Move the Content of memory to A.

; ↴

; END the program

Op-Codes

Address (H)

2000

2003

2004

2005

2006

2007

2008

Data (H)

21, 50, 20

7E

23

46

A0

23

77

1) For AND gate

Input

Address	Data(H)
20SO(H)	01 H
20SI (H)	01 H

Output

Accumulator	Data(H)
	01 H

2) For XOR gate

Input

Address	Data(H)
Accumulator	5A(H)
Reg. B	1F H

Output

Address	Data
20SOH	49H

Truth table for OR operation

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Truth-table for NOR operation

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

2009

EF

2) For XOR logic gate

Mnemonics

MVI A, SAH

; Load A with 1st no.

MVI B, IFH

; Load B with 2nd no.

XRA B

; XOR op. on A and B

STA 2050H

; Store Accumulator to destination

HLT

; Terminate the programme.

Comments

Op-Code

Address

2000

Data (H)

3E, 5A

2002

06, 1F

2004

A8

2005

32, 50, 20

2008

76

3) For 'OR' logic operations:

Mnemonics

MVI A, SAH

; Load the accumulator with SAH

MVI B, IFH

; Load the B with IFH

ORAB

; OR operation of A and B

STA 2030H

; Store A at the destination Address

RST

; Terminate the programme.

Comments

Teacher's Signature _____

3) For OR gate:-

Input

Address	Data(H)
Accumulator	5A(H)
Reg. B	1F(H)

Output :-

Address	Data(H)
2030H	5FH

4) For NOR gate :-

Input :-

Address	Data(H)
Accumulator	43H
Register B	6AH

Output

Address	Data(H)
2020	94H

Truth-table for NAND operation

A	B	Output
0	0	
0	1	1
1	0	1
1	1	0

Truth-table for XNOR gate

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1

Opcodes

Address

2000

2002

2004

2005

2008

Data (H)

BF, SA

06, 1F

BO

32, 30, 20

EF

4) For NOR logic operation:-

Mnemonics

MVI A, 43H

MVI B, 6AH

ORA B

CMA

STA 2020H

HLT

Comments

; Load the accum. with 43H

; Load the B with 6AH

; OR operation of A with B

; Compliment copy contents of A

; Store A at destination add.

; Terminate the program.

Opcodes

Address

2000

2002

2004

2005

2006

2009

Data (H)

3E, 43

06, 6A

BO

2F

32, 20, 20

76

Teacher's Signature _____



5) For NAND gate:

Input:

Address
Accumulator
Register B

Data (H)
74H
B9H

Output:-

2040H

CFH

6) For XNOR gate

Input:

Address
2050H
2051H

Data(H)
2D
41

Output

2080H

90

5) For NAND Logic operation:-

Mnemonics

MVI A, 74H

MVI B, B9H

ANA B

CMA

STA, 2040H

HLT

Comments

; Load the accumulator with 74H

; Load the Register B with B9

; AND operation on A with B

; Complement the Contents of A

; Store the A at Destination Add.

; Terminate the program.

Hex Code :Address

2000

2002

2004

2005

2006

2009

Data (H)

3E, 74

06, B9

A0

2F

32, 40, 20

F6

6). For XNOR Logic - operation :-

Mnemonics

LHLD, 2050H

MOV A, L

XRA A

CMA

STA 2080H

Comments:

; Load the HL pair with locations

; Move the Contents of L to Acc.

; XOR operation on acc. with HL

; Complement the Contents of Acc.

; Store the destination Address

Teacher's Signature _____

RST

i) Terminate the program

Ob-codes

Address (H)

2000

2003

2004

2005

2006

2009

Data (H)

2A, 50, 20

7D

AC

2F

32, 80, 20

EF

Result :-

Hence we verified the truth tables of all logic operations successfully.