

NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR  
CACHAR, ASSAM

LABORATORY EXERCISE BOOK

B.TECH. III<sup>RD</sup> SEM.

NAME: SUBHOJIT GHIMIRE

SCH. ID.: 1912160

BRANCH: C.S.E. - B

SUBJECT: MICROPROCESSOR LAB

CODE : EE224

AIM: WRITE A PROGRAM AND TEST FOR TYPICAL DATA:

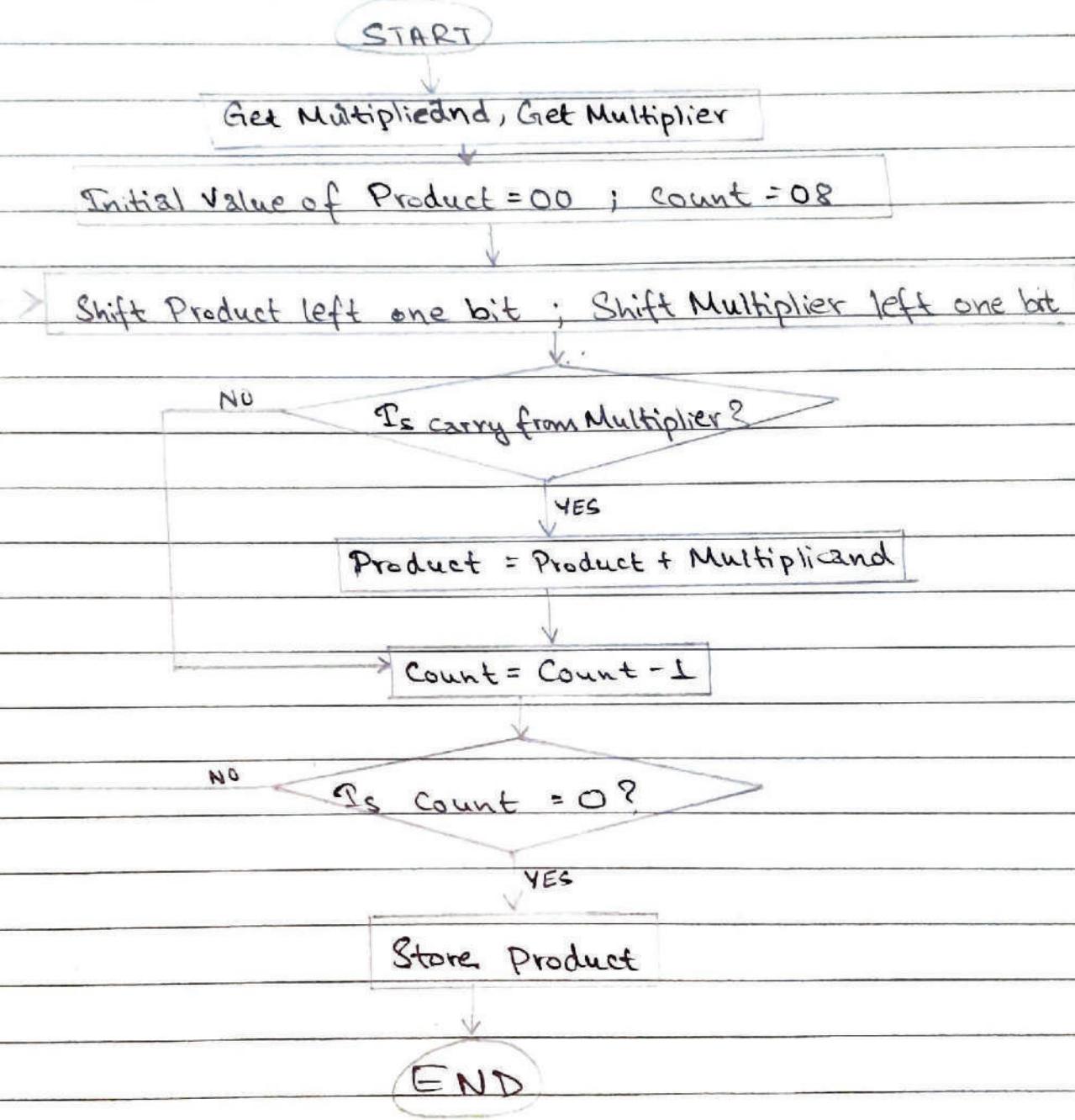
(A) MULTIPLICATION OF TWO 8-BIT NUMBERS BY  
ROTATION METHOD

(B) DIVISION OF TWO 8-BIT NUMBERS BY  
REPEATED SUBTRACTION METHOD.

THEORY:

1. LXI H loads 16 bit data in register pair designated by the operand.
2. XCHG exchanges HL with DE pair
3. LDA copies address accum content to accumulator.
4. LDH address. loads 16 bit data from specified address to designated in register pair
5. MOV A,M copies memory address content of H-L pair into accumulator.
6. MVI moves immediate value to specified register.
7. DAD instruction adds specified register pair content to HL pair content and stores into HL pair.
8. SUB subtracts register pair content to accumulator and stores in accumulator.
9. CMP compares the register content to accumulator.
  - if  $(A) < (\text{Reg/Mem})$ ; carry flag = 0 and zero flag = Reset
  - if  $(A) = (\text{Reg/Mem})$ ; carry flag = Reset and zero flag = Set
  - if  $(A) > (\text{Reg/Mem})$ , Both carry and zero flag = Reset
10. INC Addr. jumps to specified address if carry flag is reset
11. DEC decrements specified register content by 1.
12. STA copies content of accumulator to specified memory address
13. SHLD stores HL pair content to specified address.
14. JNC jumps to specified address if carry flag is reset.

## FLOWCHART (for AIM <AS>)



# PROGRAM (for AIM <P>)

Address	Label	Mnemonics	Opcodes	Comments
		#ORG 2000H		
2000		LHLD 7501	2A	Get multiplicand in HL pair
2001			0L	
2002			20	
2003		XCHG	EB	Exchange HL with DE pair
2004		LDA 7503	3A	Get 2nd number in acc.
2005			03	
2006			20	
2007		LXI H, 0000	21	Initial product in HL = 00
2008			00	
2009			40	
200A		MVI C, 08	0E	Count = 8 in register C
200B			08	
200C	UP	DAD H	29	Shift partial product left 1 bit
200D		RAL	17	Rotate multi by 1 bit
200E		INC <sup>DOWN</sup> 7502	D9	If multiplier != 0, goto Down
200F			12	
2010			00	
2011		DAD D	19	Product = Product * Multiplier
2012	DOWN	DCR C	0D	Decrement Count
2013		TJNZ UP	C2	Jump until C = 0
2014			0C	
2015			20	
2016		SHLD 7504	22	Store result.
2017			04	
2018			20	
2019		HLT	76	Terminate
		#ORG 7501H		Store input in address
		# DB 25, 00, 05		Get nos from addresses

# RESULT:

Input: 7501 - 25H ; 7502 - C0H ; 7503 - 05H

Output: 7504 - B9H ; 7505 - 00H

## 8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

8085 Assembly Language Editor

Assembler Disassembler

```
# ORG 2000H
    LHLD 7501
    XCHG
    LDA 7503
    LXI H,0000
    MVI C,08

UP:   DAD H
      RAL
      JNC DOWN
      DAD D

DOWN:  DCR C
      JNZ UP
      SHLD 7504
      HLT

# ORG 7501H
# DB 25, 00, 05
```

Autocorrect Assemble

Created by : Jubin Mitra

Registers Memory Devices

### Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	02	0	0	0	0	0	0	1	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	25	0	0	1	0	0	1	0	1
Register H	00	0	0	0	0	0	0	0	0
Register L	B9	1	0	1	1	1	0	0	1
Memory(M)	00	0	0	0	0	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	55	0	1	0	1	0	1	0	1

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	00B9
Program Status Word(PSW)	0255
Program Counter(PC)	2019
Clock Cycle Counter	386
Instruction Counter	49

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 Current Address

## 8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓	2000		LHLD 7501	2A	3	5	16
	2001			01			
	2002			75			
✓	2003		XCHG	EB	1	1	4
✓	2004		LDA 7503	3A	3	4	13
	2005			03			
	2006			75			
✓	2007		LXI H,0000	21	3	3	10
	2008			00			
	2009			00			
✓	200A		MVI C,08	0E	2	2	7
	200B			08			
✓	200C	UP	DAD H	29	1	3	10
✓	200D		RAL	17	1	1	4
✓	200E		JNC DOWN	D2	3	3	10
	200F			12			
	2010			20			
✓	2011		DAD D	19	1	3	10
✓	2012	DOWN	DCR C	0D	1	1	4

Registers Memory Devices

### Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
200B	08
200C	29
200D	17
200E	D2
200F	12
2010	20
2011	19
2012	0D
2013	C2
2014	0C
2015	20
2016	22
2017	04
2018	75
2019	76
7501	25
7503	05
7504	B9

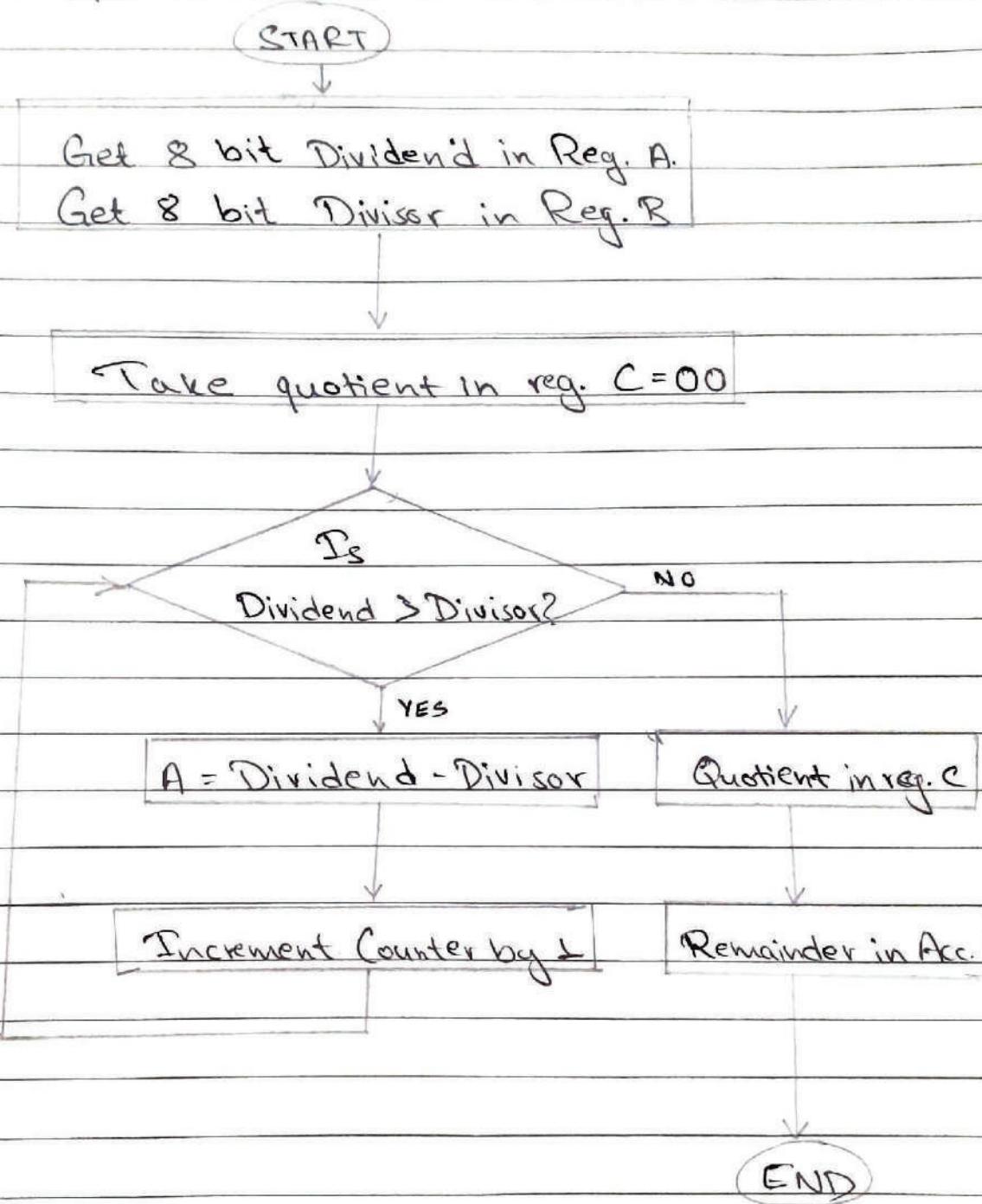
- Show entire memory content
- Show only loaded memory location
- Store directly to specified memory location

Simulate

Start From → 2000

Run all At a Time Step By Step

FLOWCHART (for AIM < B >)



# PROGRAM ( for AIM < B>)

Address	Label	Mnemonics	Opcode	Comment
		#ORG 0000H		
0000		LDA 7501	3A	[ <sup>7501</sup> <sub>0000</sub> ] $\Rightarrow$ A (Divisor)
0001			01	
0002			02	
0003		MOV B,A	47	Take divisor in Reg. B
0004		LDA 7502	3A	Take dividend in reg. A
0005			02	
0006			20	
0007		MVI C,00	0E	Quotient = 00
0008			00	
0009		CMP B	B8	Compare A to B
000A		JC DOWN	DA	Jump if carry
000B			13	
000C			00	
000D	UP	SUB B	90	Dividend - Divisor $\Rightarrow$ A
000E		INR C	0C	C = C + 1
000F		CMP B	B8	is dividend < divisor?
0010		JNC UP	D2	If not, go to UP
0011			0D	
0012			00	
0013	DOWN	STA 7503	32	Stores Remainder
0014			04	
0015			20	
0016		MOV A,C	79	moves C content to A
0017		STA 7504	32	Stores Quotient
0018			04	
0019			20	
001A		HLT	76	Terminate
		#ORG 7501H		Store input in address

# RESULT:

Input : 7501-06H ; 7502-26H

Output : 7503-02H ; 7504-06H

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

Registers Memory Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	06	0	0	0	0	0	1	1	0
Register B	06	0	0	0	0	0	1	1	0
Register C	06	0	0	0	0	0	1	1	0
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)	3A	0	0	1	1	1	0	1	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	85	1	0	0	0	0	1	0	1

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0685
Program Counter(PC)	001A
Clock Cycle Counter	212
Instruction Counter	34

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

Autocorrect Assemble

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

Registers Memory Devices

Assembler :

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LDA 7501	3A	3	4	13
0001			01			
0002			75			
✓ 0003		MOV B,A	47	1	1	4
✓ 0004		LDA 7502	3A	3	4	13
0005			02			
0006			75			
✓ 0007		MVI C,00	0E	2	2	7
0008			00			
✓ 0009		CMP B	B8	1	1	4
✓ 000A	UP	JC DOWN	DA	3	3	10
000B			13			
000C			00			
✓ 000D	UP	SUB B	90	1	1	4
✓ 000E		INR C	0C	1	1	4
✓ 000F		CMP B	B8	1	1	4
✓ 0010		JNC UP	D2	3	3	10
0011			0D			
0012			00			

Simulate

Start From → 0000

Run all At a Time Step By Step

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
000A	DA
000B	13
000D	90
000E	0C
000F	B8
0010	D2
0011	0D
0013	32
0014	03
0015	75
0016	79
0017	32
0018	04
0019	75
001A	76
7501	06
7502	26
7503	02
7504	06

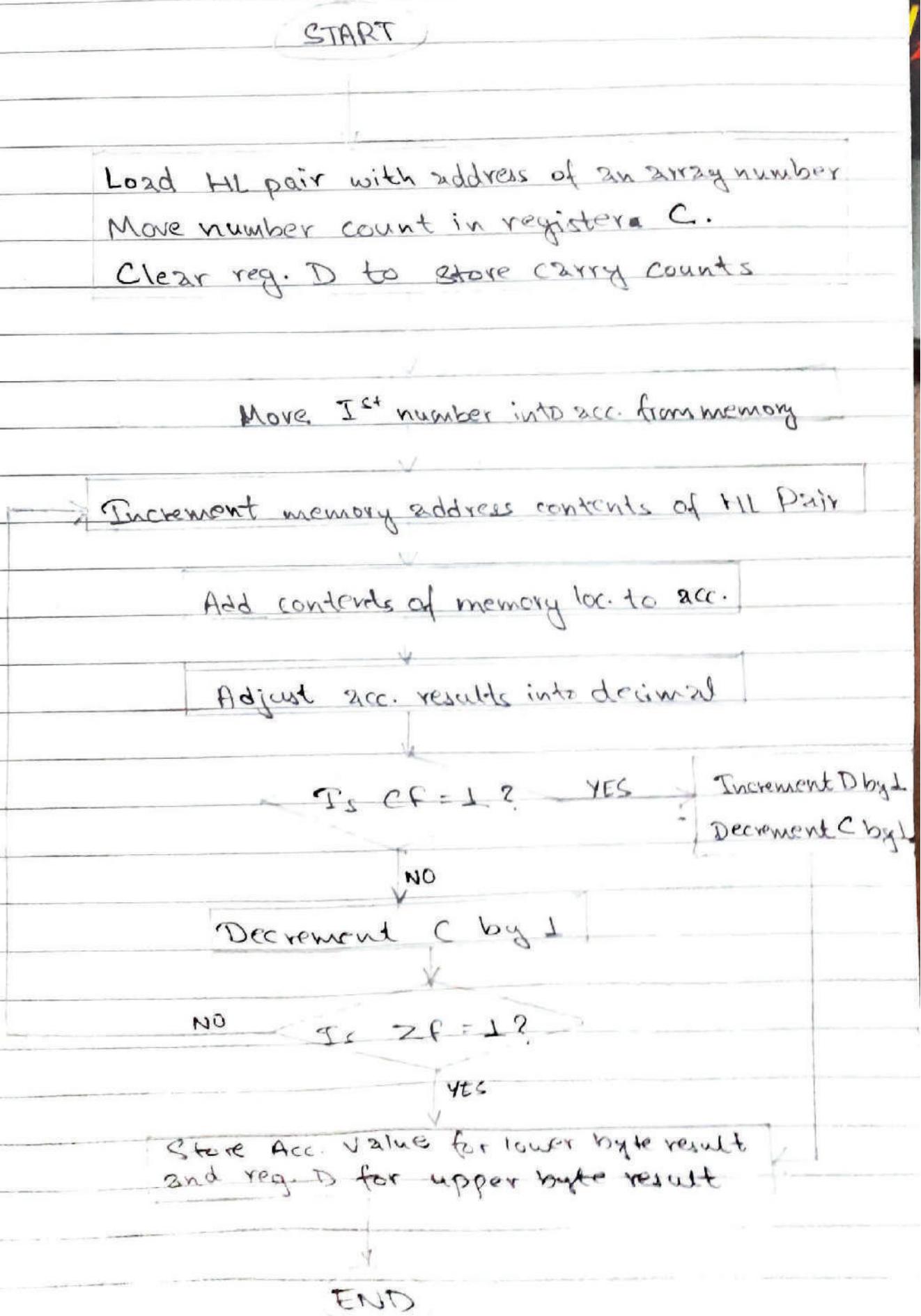
Show entire memory content  
 Show only loaded memory location  
 Store directly to specified memory location

AIM: WRITE A PROGRAM TO ADD 'N' TWO DIGIT BCD NUMBERS USING 8085 AND VERIFY

### THEORY:

1. LXI H loads 16 bit data in register pair HL
2. INX H increments contents of register pair by 1.
3. MOV A,M copies content of memory location pointed by HL pair into accumulator.
4. MVI moves immediate value to specified register.
5. DCR Reg decrements register content by 1.
6. DAA changes the content of accumulator from 2 binary value to two 4-bit BCD digits.
7. JNZ Addr. jumps the execution to specified address if zero flag is reset.
8. ADD M adds memory address content specified in HL register to accumulator and result stored in accumulator.
9. SUB Reg. subtracts register content by accumulator and result is stored into accumulator.
10. JMP jumps execution to specified address.
11. STA copies content of accumulator to specified address.
12. RST 1 finishes the execution of the program.

## FLOWCHART:



## PROGRAM :

Address	Label	Mnemonics	Opcode	Comments
		#ORG 2000H		
2000		LXI H, F100	2L	HL pair loads F100
2001			00	
2002			F1	
2003		MOV C,M	3E	Moves memory content to C
2004		MVI D,00	16	Sets reg. D as 00H
2005			00	
2006		INX H	23	Increases HL content by 1
2007		DCR C	0D	Decreases C content by 1
2008		MOV A,M	7E	Moves memory content to acc.
2009	UP	INX H	23	Increases HL content by 1
200A		ADD M	86	Adds acc. and memory
200B		DAA	27	Decimal Adjust after Addition
200C		JNC 2010	D2	Jumps to 2010 address
200D			10	
200E			20	
200F		INR D	14	Increases D content by 1
200G	DOWN	DCR C	0D	Decreases C content by 1
2011		INZ 2009	C2	
2012			09	
2013			20	
2014		STA F200	32	Stores acc. content to F200
2015			00	
2016			F2	
2017		HLT	76	Terminates
		#ORG F100H		Store inputs at address
		#DB 04,43,77,55,55		Get nos from addresses.

# RESULT:

Input: F100 - 04 H ; F101 - 43 H ; F102 - 77 H  
 F103 - 55 H ; F104 - 55 H

Output: D - 02 H ; A - 30 H

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

8085 Assembly Language Editor

Assembler Disassembler

```
# ORG 2000H
LXI H,F100
MOV C,M
MVI D,00
INX H
DCR C
MOV A,M

UP: INX H
ADD M
DAA
JNC 2010
INR D

DOWN: DCR C
JNZ 2009
STA F200
HLT

# ORG F100H
# DB 04,43,77,55,55
```

Autocorrect Assemble

Registers Memory Devices

Registers:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	30	0	0	1	1	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	02	0	0	0	0	0	0	1	0
Register E	00	0	0	0	0	0	0	0	0
Register H	F1	1	1	1	1	0	0	0	1
Register L	04	0	0	0	0	0	1	0	0
Memory(M)	55	0	1	0	1	0	1	0	1

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	55	0	1	0	1	0	1	0	1

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	F104
Program Status Word(PSW)	3055
Program Counter(PC)	2017
Clock Cycle Counter	181
Instruction Counter	28

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:

Created by : Jubin Mitra

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 2000		LXI H,F100	21	3	3	10
2001			00			
2002			F1			
✓ 2003		MOV C,M	4E	1	2	7
✓ 2004		MVI D,00	16	2	2	7
2005			00			
✓ 2006		INX H	23	1	1	6
✓ 2007		DCR C	0D	1	1	4
✓ 2008		MOV A,M	7E	1	2	7
✓ 2009	UP	INX H	23	1	1	6
✓ 200A		ADD M	86	1	2	7
✓ 200B		DAA	27	1	1	4
✓ 200C		JNC 2010	D2	3	3	10
200D			10			
200E			20			
✓ 200F		INR D	14	1	1	4
✓ 2010	DOWN	DCR C	0D	1	1	4
✓ 2011		JNZ 2009	C2	3	3	10
2012			09			
2013			00			

Simulate

Start From → 2000

Registers Memory Devices

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
200B	27
200C	D2
200D	10
200E	20
200F	14
2010	0D
2011	C2
2012	09
2013	20
2014	32
2016	F2
2017	76
F100	04
F101	43
F102	77
F103	55
F104	55
F200	30

Show entire memory content  
 Show only loaded memory location  
 Store directly to specified memory location