

NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR
CACHAR, ASSAM

LABORATORY EXERCISE BOOK

B.Tech. IIIRD SEM.

NAME: SUBHOJIT GHIMIRE

SCH.ID.: 1912160

BRANCH: C.S.E. - B

SUBJECT: MICROPROCESSOR LAB

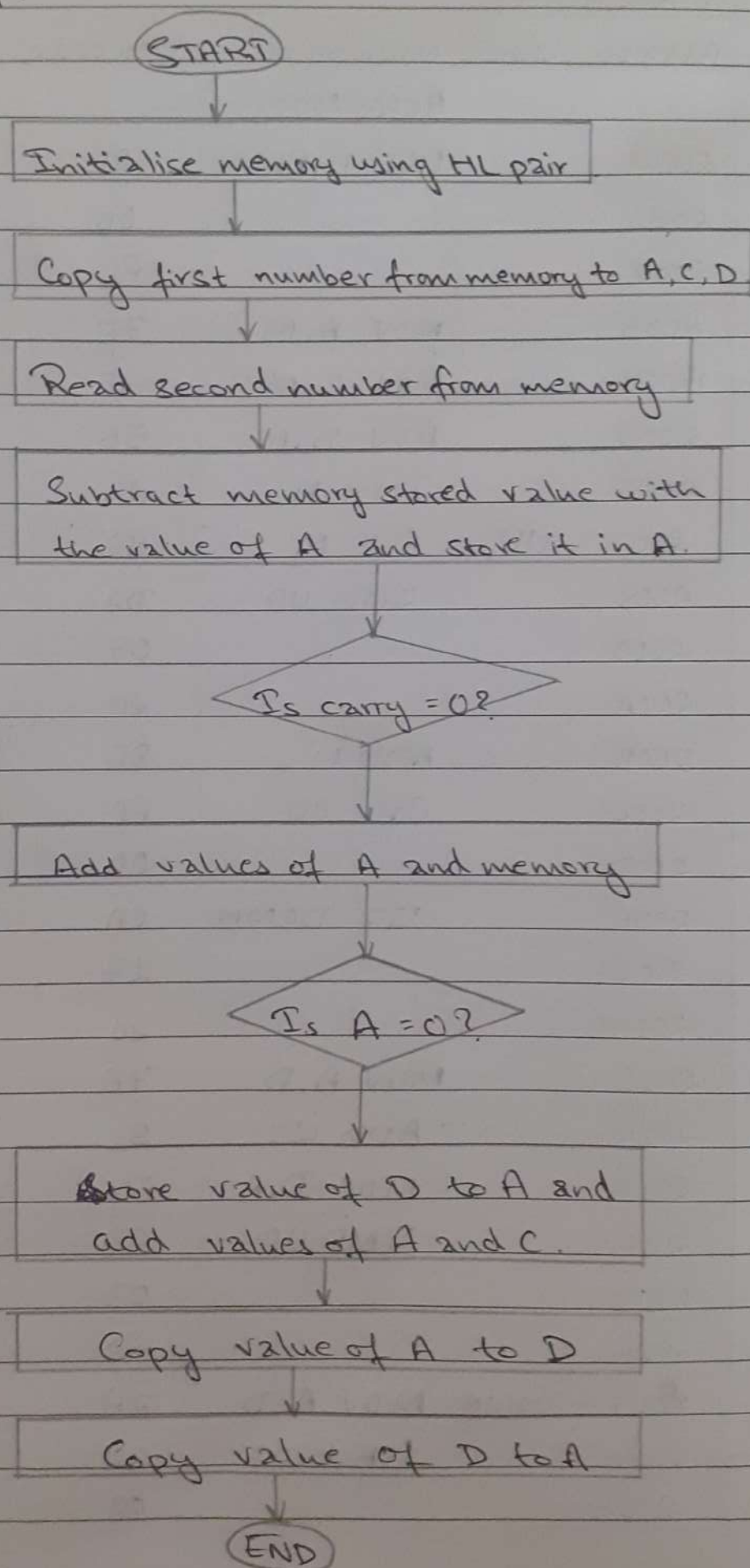
CODE : EE224

AIM: Write a PROGRAM To FIND LCM OF TWO NUMBERS
USING 8085 AND VERIFY:

THEORY:

1. ORG Address Directive reserves the starting address for Program Code or data in specified memory array.
2. LXI H loads 16 bit data in register pair designated by operand.
3. INX H increments the contents of the register pair by one.
4. MOV A, M copies data in memory address of HL pair to the accumulator.
5. SUB M subtracts contents of memory from acc.
6. MVI moves immediate value to specified register.
7. CPI compares immediate value with accumulator.
If, $(A) < \text{Register}$, carry flag is set and zero flag is reset.
If, $(A) = \text{Register}$, carry flag is reset and zero flag is set.
If, $(A) > \text{Register}$, both carry and zero flags are reset.
8. JZ addr. jumps execution to the specified address if zero flag is set.
9. JNC Addr. jumps execution to the specified address if the carry flag is reset.
10. ADD Reg. adds register and accumulator values.
11. JMP jumps execution to specified address.
12. STA copies content of accumulator to specified memory address.
13. HLT terminates further execution.
14. DB derivative stores values in specified address.

FLOWCHART:



PROGRAM:

Address	Label	Mnemonics	Hexcode	Comment
		#ORG 2000H		
2000		LXI H, F100	21	HL ← F100
2001			00	
2002			F1	
2003		MOV A, M	7E	A ← [HL]
2004		MOV C, M	4E	C ← [HL]
2005		MOV D, M	56	D ← [HL]
2006		INX H	23	HL+1 → HL
2007	UP	SUB M	96	A-M → M
2008		JNC UP	D2	If A > M, jump to UP
2009			07	
200A			20	
200B		ADD M	86	M+A → A
200C		CPI 00	FE	Compare A with 00H
200D			00	
200E		JZ DOWN	CA	If A=00H, Jump Down
200F			17	
2010			20	
2011		MOV A, D	7A	D → A
2012		ADD C	81	A+C → A
2013		MOV D, A	57	A → D
2014		JMP UP	C3	Jump to UP
2015			07	
2016			20	
2017	DOWN	MOV A, D	7A	D → A
2018		STA F200	32	A → [F200]
2019			00	

201A

201B

HLT

76

Execution Terminate

#ORG F100H

Store inputs at address

#DB 05H, 03H

Store bytes in successive locati

RESULT:

Input: F100-05H ; F101-03H

Output: A-0FH ; F200-0FH

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 A,M
MOV C,M
MOV D,M
INX H

UP:

SUB M
JNC UP
ADD M
CPI 00
JZ DOWN
MOV A,D
ADD C
MOV D,A
JMP UP

DOWN:

MOV A,D
STA F200
HLT

#ORG F100H

#DB 05H, 03H

Autocorrect

Assemble

Registers Memory Devices

Registers:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	0F	0	0	0	0	1	1	1	1
Register B	00	0	0	0	0	0	0	0	0
Register C	05	0	0	0	0	0	1	0	1
Register D	0F	0	0	0	0	1	1	1	1
Register E	00	0	0	0	0	0	0	0	0
Register H	F1	1	1	1	1	0	0	0	1
Register L	01	0	0	0	0	0	0	0	1
Memory(M)	03	0	0	0	0	0	0	1	1

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	45	0	1	0	0	0	1	0	1

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	F101
Program Status Word(PSW)	0F45
Program Counter(PC)	201B
Clock Cycle Counter	364
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. Converter Tool :

Hexadecimal	Decimal	Binary
0		0

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 A,M	7E	1	2	7
✓ 2004		MOV C,M	4E	1	2	7
✓ 2005		MOV D,M	56	1	2	7
✓ 2006		INX H	23	1	1	6
✓ 2007	UP	SUB M	96	1	2	7
✓ 2008		JNC UP	D2	3	3	10
2009			07			
200A			20			
✓ 200B		ADD M	86	1	2	7
✓ 200C		CPI 00	FE	2	2	7
200D			00			
✓ 200E		JZ DOWN	CA	3	3	10
200F			17			
2010			20			
✓ 2011		MOV A,D	7A	1	1	4
✓ 2012		ADD C	81	1	1	4

Simulate

Start From →

2000

Run all At a Time

Step By Step

Registers Memory Devices

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
2008	D2
2009	07
200A	20
200B	86
200C	FE
200E	CA
200F	17
2010	20
2011	7A
2012	81
2013	57
2014	C3
2015	07
2016	20
2017	7A
2018	32
201A	F2
201B	76
F100	05
F101	03
F200	0F

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