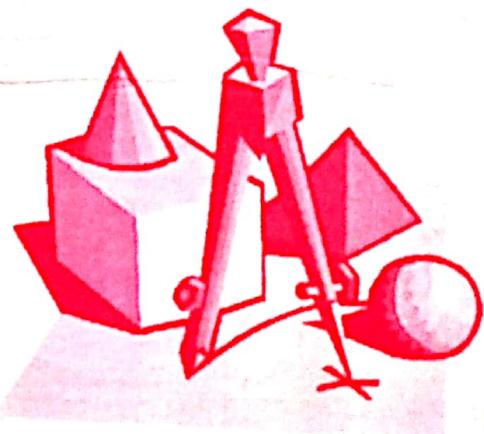


Physics

Practical Note Book



Name Santosh Bhandari

Class BCA Sec. B

Subject Microprocessor & Computer Architecture Roll No. 58

House _____

Subject Teacher's name Suri Sharma

S.N.	Name of the Experiment	Date	Page	Remarks
1.	Add 2 8-bit Numbers		1-2	
2.	Subtract 2 8-bit Numbers		3-4	
3.	Multiply 2 8-bit Numbers		5-6	
4.	Divide 2 8-bit Numbers		7-8	
5.	Find greatest among 2 8-bit numbers		9-10	
6.	Find smallest among 2 8-bit numbers.		11-12	
7.	Find 1's complement		13-14	
8.	Find 2's complement		15-16	
9.	Add 16-bit numbers		17-18	
10.	Subtract 16-bit numbers		19-20	
11.	Generate fibonacci series upto 10th term		21-22	
12.	Find a number is odd or Even		23-26	
13.	Display number from 1 to 10.		27-28	
14.	Find sum of numbers from 1 to 10.		29-30	
15.	Display odd numbers from 1 to 10.		31-32	
16.	Display even numbers from 1 to 10		33-34	
17.	Find smallest number in array.		35-36	
18.	Right-shift 8-bit no.		37-38	

Aman

1. TITLE :- To Add two 8-bit Numbers

2. OBJECTIVES :- To add the data 03H and 06H and store result in memory location 3000H

3. REQUIREMENTS :-

- 8085 Microprocessor Kit
- +5V Power Supply

4. ALGORITHM :-

Step 1: Start

Step 2: Input numbers from memory location 3050H and 3051H and store in register A and B.

Step 3: Add content of accumulator and B register

Step 4: Store result in memory location 3000H

Step 5: Terminate the program

5. PROCESS:-

Memory Address	Instruction	Hexcode
2000	LDA 3050 H	3A
2001		50
2002		30
2003	MOV B,A	47
2004	LDA 3051 H	3A
2005		51
2006		30
2007	ADD B	80
2008	STA 3000 H	32
2009		00
200A		30
200B	HLT	76

6. RESULT:-

INPUT

Memory Address	3050H	3051H
Data	03H	06H

OUTPUT

Memory Address	3000H
Data	09H

1. TITLE:- To Subtract two 8-bit Numbers

2. OBJECTIVES:- To subtract the data 06H and 03H and store result in memory location 3000H.

3. REQUIREMENTS:-

- 8085 Microprocessor Kit

- +5V Power Supply

4. ALGORITHM:-

Step 1: Start

Step 2: Input numbers from memory location 2000H and 2001H and store in register A and B.

Step 3: Subtract content of accumulator and B register.

Step 4: Store result in memory location 3000H.

Step 5: Terminate the program

5. PROCESS:-

Memory Address	Instruction	Hexcode
2000	LDA 2001H	3A
2001		00
2002		20
2003	MOV B,A	47
2004	LDA 2001H	3A
2005		01
2006		20
2007	SUB B	90
2008	STA 3000H	32
2009		00
200A		30
200B	HLT	76

6. RESULT:-

INPUT

Memory Address	2000H	2001H
Data	03H	06H

OUTPUT

Memory Address	3000H
Data	08H

1. TITLE:- Multiply 2 8-bit Numbers
2. OBJECTIVES:- To multiply the data 02H and 06H and store result in memory location 3055H.

3. REQUIREMENTS:-

- 8085 microprocessor kit
- +5 v power supply

4. ALGORITHM:-

Step 1: Start

Step 2: Input the numbers from memory location 3040H and 3041H and store in register B and C.

Step 3: Move immediate 00H data to Accumulator.

Step 4: Add the contents of accumulator with register B.

Step 5: Decrement the contents of C register

Step 6: Jump if zero flag is not set.

Step 7: Store the result in 3055H

Step 8: Terminate the program.

5. PROCESS:-

Memory Address	Instruction	Hex code
3000	LDA 3040H	3A
3001		40
3002		30
3003	MOV B,A	47
3004	LDA 3041H	3A
3005		41
3006		30
3007	MOV C,A	4F
3008	MUL A,60H	3E
3009		00H
300A	TOP : ADD B	80
300B	DCR C	0D
300C	JNZ TOP	C2
300D		3A
300E		30
300F	STA 3055	32
3010		55
3011		30
3012	HLT	76

6. RESULT:-

INPUT

Memory Address	3040H	3041H
Data	02H	06H

OUTPUT

memory address	3055H
Data	02H

1. TITLE:- Divide 2 8-bit Numbers
2. OBJECTIVES:- To divide the data 06H and 03H and Store quotient in 3000H memory location and remainder in memory location 3001H.
3. REQUIREMENTS:-
- 8085 microprocessor kit
 - +5V power supply
4. ALGORITHM
- Step 1: Start
- Step 2: Input the numbers from memory location 3003H and 3004H and store in register B and Accumulator
- Step 3: Move immediate 00H data to C register.
- Step 4: Compare the content of Accumulator with register B.
- Step 5: Jump if carry flag is set.
- Step 6: Subtract the content of accumulator with B register.
- Step 7: Increment the content of C register.
- Step 8: Jump to compare the content of Accumulator with register B if carry flag is not set.
- Step 9: Store the remainder in memory location 3001H.
- Step 10: Move the content of C register in Accumulator.
- Step 11: Store the quotient in memory location 3000H
- Step 12: Terminate the program

5. PROCESS :-

Memory Address	Instruction	Hex code
2000	LDA 3004H	3A
2001		04
2002		30
2003	MOV B,A	47
2004	LDA 3003H	3A
2005		03
2006		30
2007		30
2008	MVI C,00H	0E
2009		00
200A	TOP: CMP B	B8
200B	JC SKIP	DA
200C		12
200D		20
200E	SUB B	90
200F	INR C	0C
2010	JMP TOP	C3
2011		09
2012	SKIP: STA 3001	32
2013		01
2014		30
2015	MOV A,C	79
2016	STA 3000H	3A
2017		00
2018		30
2019	HLT	76

6. RESULT :-

INPUT

Memory Address	3003H	3004H
Data	06H	03H

OUTPUT

Memory Address	3000H	3001H
Data	02H	00H

1. TITLE:- Find greatest among 2 8-bit numbers

2. OBJECTIVES:- To find the greatest number among the data 06H and 07H and store the result in memory location 3055H.

3. REQUIREMENTS:-

- 8085 microprocessor k91

- +5V power supply

4. ALGORITHM:-

Step 1:- Start

Step 2:- Input the number from memory location 2050H and 2051H and store in accumulator and B register.

Step 3:- Compare the content of Accumulator with Register B

Step 4:- Jump if carry flag is not set.

Step 5:- Move the content of B register to accumulator.

Step 6:- Store the result in memory location 3055H

Step 7:- Terminate the program

5. PROCESS:-

Memory Address	Instruction	Hexcode
8000	LDA 2051H	3A
8001		51
8002		20
8003	MOV B,A	47
8004	LDA 2050H	3A
8005		50
8006		20
8007	CMP B	B8
8008	JNC X	D2
8009		0C
800A		80
800B	Mov A,B	38
800C	X: STA 3055H	32
800D		55
800E		30
800F	HLT	76

6. RESULT:-

INPUT		OUTPUT	
Memory Address	Data	Memory Address	Data
2050H	06H	2051H	07H

1. TITLE:- Find smallest among 2 8-bit numbers

2. OBJECTIVES:- To find smallest number among the 2 data OFH and DAH and store the result in memory location F055H.

3. REQUIREMENTS:-

- 8085 microprocessor kit

- +5V power supply

4. ALGORITHM:-

Step 1 : Start

Step 2 : Load the data from memory location 2051H and 2050H and store in accumulator and X register.

Step 3 : Compare the content of accumulator with register C.

Step 4 : Jump if carry flag is set.

Step 5 : Move the content of register C to accumulator.

Step 6 : Store the content of accumulator in F055H memory location.

Step 7 : Terminate the program

5. PROCESS:-

Memory address	Instruction	Hexcode
A000	LDA 2051H	3A
A001		51
A002		20
A003	MOV B,A	47
A004	LDA 2050H	3A
A005		50
A006		20
A007	CMP B	B8
A008	JC Y	DA
A009		OC
A00A		AO
A00B	MOV A,B	78
A00C	X: STA F055H	32
A00D		055
A00E		F0
A00F	HLT	76

6. RESULT:-

INPUT

Memory address	2050H	2051H
Data	OFH	DAH

OUTPUT

Memory address	F055H
Data	0AH

1. TITLE:- Find 1's complement

2. OBJECTIVES:- To find the 1's complement of 03H and store the result in AAAAH memory location.

3. REQUIREMENTS:-

- 8085 microprocessor 128t
- +5V power supply

4. ALGORITHM:-

Step 1: Start

Step 2: Input the data from memory location 8000H and store it in accumulator.

Step 3: Complement the content of accumulator

Step 4: Store the content of accumulator in AAAAH memory location.

Step 5: Terminate the program

5. PROCESS :-

Memory Address	Instruction	Hexcode
B000	LDA 8000H	3A
B001		00
B002		80
B003	CMA	2F
B004	STA AAAAH	32
B005		AA
B006		AA
B007	HLT	76

6. RESULT :-

INPUT

Memory Address	8000H
Data	03H

OUTPUT

Memory Address	AAAHH
Data	00H

1. TITLE :- Find 2's complement

2. OBJECTIVES :- To find the 2's complement of 03H and store the result in memory location 3333H.

3. REQUIREMENTS :-

- 8085 Microprocessor 129t
- +5V power supply

4. ALGORITHM :-

Step 1: Start

Step 2: Input the data from memory location 8888H and store it in accumulator.

Step 3: Complement the content of accumulator and add 02H with content of accumulator.

Step 4 :- Store the content of accumulator in 3333H memory location.

Step 5: Terminate the program

5o PROCESS:-

Memory Address	Instruction	Hexcode
F000	LDA 8888H	3A
F001		88
F002		88
F003	CMA	2F
F004	ADI 01H	C6
F005		01
F006	STA 3333H	32
F007		33
F008		33
F009	HLT	76

6o RESULT:-

INPUT		OUTPUT	
Memory Address	Data	Memory Address	Data
8888H	03	3333H	01H

1. TITLE: Add two 16-bit Numbers

2. OBJECTIVES: - To add 16-bit data 4533H and 3424H and store result in memory location 2055H and 2056H.

3. REQUIREMENTS:

- 8085 microprocessor or kit
- +5V power supply

4. ALGORITHM:

Step 1: Start

Step 2: Input numbers from memory location 2052H and 2053H and store in pair register HL.

Step 3: Exchange the contents of HL pair with DE pair.

Step 4: Input second number from memory location 2050H and 2051H and store in HL pair.

Step 5: Move content of L register to accumulator and add the content of accumulator with register E.

Step 6: Move content of accumulator to L register and content of H register to accumulator.

Step 7: Add accumulator content with register D and move the content of accumulator to H register.

Step 8: Store the content of HL pair in memory location 2055H and 2056H.

Step 9: Terminate the program

18. 5. PROCESS :-

Memory Address	Instruction	Hexcode
3000	LHLD 2052H	3A
3001		52
3002		20
3003	XCHG	EB
3004	LHLD 2050H	2A
3005		50
3006		30
3007	MOV A,L	7D
3008	ADD E	33
3009	MOVL,A	6F
300A	MOV A,H	7C
300B	ADe D	8A
300C	MOV H,A	67
300D	SHLD 2055H	22
300E		55
300F		20
3010	HLT	76

6. RESULT :-

INPUT

Memory Address	2050H	2051H	2052H	2053H
Data	33H	45H	24H	34H

OUTPUT

Memory Address	2055H	2056H
Data	57H	79H

1. TITLE :- Subtract 16-bit Numbers

2. OBJECTIVES :- To subtract 16-bit data 9233H and 3222H and store result in 3055H and 3056H memory location.

3. REQUIREMENTS :-

- 8085 microprocessor kit

- +5V power supply

4. ALGORITHM :-

Step 1 :- Start

Step 2 :- Input the numbers from memory location 2052H and 2053H and store in HL pair.

Step 3 :- Exchange the content of HL pair with DE pair.

Step 4 :- Input the numbers from memory location 2050H and 2051H and store in HL pair.

Step 5 :- Move content of L register to accumulator

Step 6 :- Subtract the content of accumulator with E register

Step 7 :- Move content of accumulator to register L

Step 8 :- Move content of H register to accumulator

Step 9 :- Subtract the content of accumulator with D register and move the content of accumulator to H register

Step 10 :- Store the content of HL pair in memory location 3055H and 3056H.

Step 11 :- Terminate the program

20

5. PROCESS:-

Memory Address	Instruction	Hexcode
5000	LHLD 2052H	2A
5001		52
5002		20
5003	XCHG	E8
5004	LHLD 2050H	2A
5005		50
2006		20
2007	MOVAL	7D
2008	SUB E	93
2009	MOV L,A	6F
200A	MOV A,H	7C
200B	SBB D	9A
200C	MOV H,A	67
200D	SHLD 3055H	22
200E		35
200F		30
2010	HLT	76

6.

RESULT:-

INPUT

Memory Address	2050H	2051H	2052H	2053H
Data	33H	92H	22H	32H

OUTPUT

Memory Address	3055H	3056H
Data	11H	60H

1. TITLE:- Generate Fibonacci series up to 10th term
2. OBJECTIVES:- To display and store fibonacci series up to 10th term on memory location 2050H onwards.
3. REQUIREMENTS:-
 - 8085 microprocessor kit
 - +5V power supply
4. ALGORITHM:-

Step 1:- Start

Step 2:- Load the memory location 2050H in HL pair

Step 3:- Move immediate 08H, 00H and 01H data to registers C, B and D respectively

Step 4:- Move the content of B register in memory location pointed by HL pair.

Step 5:- Increment the memory location pointed by HL pair

Step 6:- Move the content of D register in memory location pointed by HL pair.

Step 7:- Move the content of B register to accumulator

Step 8:- Add the content of accumulator with D register.

Step 9:- Move the content of D register to B and content of accumulator to D register.

Step 10:- Increment the memory location pointed by HL pair and Move the content of accumulator to memory location pointed by HL pair.

- 22 Step 01:- Decrement the content of register C
Step 02:- Jump if zero flag is not set.
Step 03:- Terminate the program.

So PROCESS:-

Memory Address	Instruction	Hexcode
3000	LXZH, 2058H	21
3001		50
3002	MVZ C, 08H	20
3003		6E
3004		08H
3005	MVI B, 00H	06
3006		00H
3007	MVI D, 01H	16
3008		01H
3009	MOV M, B	70
300A	INX H	23
300B	MOVM, D	72
300C	TOP: MOV A, 8	78
300D	ADD D	82
300E	MOV B, D	42
300F	MOV D, A	57
3010	INX H	23
3011	MOV M, A	77
3012	DCR C	0D
3013	JNZ TOP	C2
3014		0C
3015		30
3016	HLT	76

So RESULT:-

OUTPUT

Memory Address	2058H	2051H	2052H	2053H	2054H	2055H	2056H	2057H	2058H	2059H
Data	0	1	1	2	3	5	8	13	21	34

1. TITLE :- Find a number is odd or Even

2. OBJECTIVES :- To find data $09H$ is odd or Even and Store the result in memory location $3005H$.

3. REQUIREMENTS :-

- 8085 microprocessor kit
- +5V power supply

4. ALGORITHM :-

Step 1 :- Start

Step 2 :- Input number from memory location $3000H$ and store it in Accumulator.

Step 3 :- Perform AND operation with accumulator value.

Step 4 :- Jump if zero flag is set

Step 5 :- Move immediate $01H$ data to accumulator

Step 6 :- Jump to mentioned label.

Step 7 :- Move immediate $00H$ data to accumulator

Step 8 :- Store the result in memory location $3005H$

Step 9 :- Terminate the program

5. PROCESS :-

Memory Address	Instruction	Hexcode
6000	LDA 3000H	3A
6001		00
6002		30
6003	ANI 01H	E6
6004		02
6005	J2 X	CA
6006		0C
6007		60
6008	MVI A, 01H	3E
6009		01
600A	JMP Y	C3
600B		0F
600C	X: MVI A, 00H	3E
600D		00H
600E		
600F	Y: STA 3005H	32
6010		05
6011		30
6012	HLT	76

6. RESULT :-

INPUT

Memory Address	3000H
Data	06H

OUTPUT

Memory Address	3005H
Data	02H

1. TITLE: - Display number from 1 to 20.

2. OBJECTIVES: - To display and store the number 1 to 20 on memory location 4000H onwards.

3. REQUIREMENTS: -

= 8085 microprocessor kit

= +5V power supply

4. ALGORITHM:

Step 1:- Start

Step 2:- Load the memory location 4000H in HL pair.

Step 3:- Move immediate 00H and 0AH data to register B and C.

Step 4:- Increment the content of register B.

Step 6:- Move the content of B register to memory location pointed by HL pair.

Step 7:- Increment the memory location pointed by HL pair.

Step 8:- Decrement the content of C register.

Step 9:- Jump if zero flag is not set.

Step 10:- Terminate the program.

5. PROCESS:-

Memory location	Instruction	Hexcode
2000	LXI H, 4000H	21
2001		00
2002		40
2003	MVI B, 00H	06
2004		00
2005	MVI C, 0A	0E
2006		0A
2007	TOP; INR B	04
2008	MOV M, B	70
2009	INX H	23
200A	DCR C	0D
200B	JNZ TOP	C2
200C		07
200D		20
200E	HLT	76

6. RESULT:-

OUTPUT

Memory Address	4000H	4001H	4002H	4003H	4004H	4005H	4006H	4007H
Data	01H	02H	03H	04H	05H	06H	07H	08H

4008H	4009H
09H	0AH

1. TITLE:- Sum of Numbers from 1 to 10

2. OBJECTIVES:- To find the sum of number 1 to 10 and store the result in 3000H memory location.

3. REQUIREMENTS:-

- 8085 microprocessor kit
- +5V Power Supply

4. ALGORITHM:

Step 1: Start

Step 2: Load the 3000H memory location in HL pair.

Step 3: Move immediate 01H, 0AH and 00H data to registers B, C and A respectively.

Step 4: Add the content of accumulator with B register.

Step 5: Increment the content of B register and decrement the content of C register.

Step 6: Jump if Zero flag is not set

Step 7: Move the content of Accumulator to the memory location pointed by HL pair

Step 8: Terminate the program

5. PROCESS:-

Memory location	Instruction	Hexcode
S000	LXI H, 3000H	21
S001		00
S002		30
S003	MVI B, 01H	06
S004		01
S005	MVZ C, 0AH	0E
S006		0A
S007	MVI A, 00H	3E
S008		00
S009	TOP : ADD B	80
S00A	INR B	04
S00B	DCR C	0C
S00C	JNZ TOP	C2
S00D		09
S00E		50
S00F	MOV M, A	77
S010	HLT	76

6. RESULT :-

OUTPUT	
Memory location	3000H
Data	37H

1. TITLE:- Display Odd Numbers from 1 to 10
2. OBJECTIVES:- TO display and store the odd numbers from 1 to 10 in memory location 3000H onwards
3. REQUIREMENTS:-
 - 8085 Microprocessor kit
 - +5V power supply
4. ALGORITHM:-
 Step 1:- Start
 Step 2:- Load 3000H memory location in HL pair.
 Step 3:- Move immediate data 01H, 0AH to register B and C.
 Step 4:- Move the content of B register pointed by HL pair memory location.
 Step 5:- Increment the content of HL pair
 Step 6:- Increment the content of register B by 2 and decrement the content of C register by 1.
 Step 7:- Jump if zero flag is not set.
 Step 8:- Terminate the program.

5. PROCESS

Memory Address	Instruction	Hexcode
1000	BLT H, 3000H	21
1001		00
1002		30
1003	MVI B, 01H	06
1004		01
1005	MVI C, 0AH	0E
1006		
1007	TOP: MOV M, B	0A
1008	INX H	23
1009	INR B	04
100A	INR B	04
100B	DCR C	0D
100C	DCR C	0D
100D	JNZ TOP	C2
100E		07
100F		10
1010	HLT	76

6. RESULT

OUTPUT

Memory Address	3000H	3001H	3002H	3003H	3004H
Data	01H	03H	05H	07H	09H

1. TITLE:- Display Even numbers from 1 to 20

2. OBJECTIVES :- To display and store even numbers from 1 to 20 in memory location 8000H onwards.

3. REQUIREMENTS:-

- 8085 microprocessor kit
- +5V power supply

4. ALGORITHM:-

Step 1:- Start

Step 2:- Load 8000H memory location in HL pair.

Step 3:- Move immediate 00H and 0A data to register B and C.

Step 4:- Increment the content of B register by 2.

Step 5:- Mov the content of register B to the memory location pointed by HL pair register.

Step 6:- Increment the memory location pointed by HL pair.

Step 7:- Decrement the content of C register by 2.

Step 8:- Jump if zero flag is not set.

Step 9:- Terminate the program

5: PROCESS :-

Memory Address	Instruction	Hexcode
2000	LXI H, 8000H	21
2001		00
2002		80
2003	MVI B, 00H	06
2004		00
2005	MVI C, 0AH	0E
2006		0A
2007	TOP: INRB	04
2008	INRB	04
2009	MOV M, B	70
200A	INX H	23
200B	DCR C	0D
200C	DCRC	0D
200D	JNZ TOP	C2
200E		07
200F		20
2010	HLT	76

6: RESULT :-

OUTPUT

Memory Address	8000H	8001H	8002H	8003H	8004H
Data	02H	04H	06H	08H	10H

1. TITLE:- Find Smallest Number in array

2. OBJECTIVES:- To find smallest number from the array and
Store the result in 2300H memory location.

3. REQUIREMENTS:-

- 8085 microprocessor kit
- +5V Power Supply

4. ALGORITHM:-

Step 1:- Start

Step 2:- load the data from memory location 2200H and store
in register C.

Step 3:- load memory location 2201H in HL pair register.

Step 4:- Move immediate 00H data to accumulator.

Step 5:- Compare the content of Accumulator with the content of
memory location pointed by HL pair.

Step 6:- Jump if carry flag is set.

Step 7:- Move content of memory location pointed by HL pair
to accumulator.

Step 8:- Increment the memory location pointed by HL pair.

Step 9:- Decrement the content of C register.

Step 10:- Jump if zero flag is not set.

Step 11:- Store the content of accumulator in 2300H memory
location.

Step 12:- Terminate the program

5. PROCESS:-

Memory Address	Instruction	Hexcode
3000	LDA 2200H	3A
3001		00
3002		22
3003	MOV C,A	4F
3004	LXI H,2201H	21
3005		01
3006		22
3007	MVI A,00H	3E
3008		00
3009	*: CMP M	BD
300A	JCX	DA
300B		0E
300C		30
300D	MOV A,M	7E
300E	R: ZNX H	23
300F	DCRC	0D
3010	JNZ X	C2
3011		09
3012		30
3013	STA 2300H	32
3014		00
3015		23
3016	HLT	76

6. RESULT

INPUT

Memory Address	2201H	2202H	2203H	2204H	2205H	---
DATA	01H	06H	08H	0AH	16H	.

OUTPUT

Memory Address	2300H
DATA	01H

1. TITLE:- Right-Shift 8-bit number

2. OBJECTIVES:- To perform shift on 8-bit data four bit right and store the result in memory location 2055H.

3. REQUIREMENT:-

- 8085 microprocessor kit

- +5V power supply

4. ALGORITHM:-

Step 1:- Start

Step 2:- load data from memory location 2054H to accumulator.

Step 3:- Perform rotate accumulator 1-bit right by 4 times.

Step 4:- Store the result in memory location 2055H.

Step 5:- Terminate the program

5. PROCESS:-

Memory Address	Instruction	Hexcode
3000	LDA 2051H	3A
3001		51
3002		20
3003	RAR	1F
3004	RAR	1E
3005	RAR	1F
3006	RAR	1E
3007	STA 2055H	32
3008		58
3009		20
300A	HLT	76

6. RESULT:-

INPUT	
Memory Address	2051H
Data	01H

OUTPUT	
Memory Address	2055H
Data	00H

1. TITLE:- Transfer block of data.
2. OBJECTIVES:- To transfer the value of memory location 3000H - 3009H to memory location 4000H - 4009H.
3. REQUIREMENTS:-
 — 8085 microprocessor kit
 — +5V power supply
4. ALGORITHM:-
 Step 1 :- Start.
 Step 2 :- Load memory location 3000H and 4000H in pair BC and DE registers.
 Step 3 :- Move immediate OA H in L register.
 Step 4 :- Load the content to the accumulator from the memory location pointed by BC pair register and store the content of accumulator to the memory location pointed by DE register.
 Step 5 :- Increment the content of pair register BC and DE.
 Step 6 :- Decrement the content of L register.
 Step 7 :- Jump if zero flag is not set.
 Step 8 :- Terminate the program.

5. PROCESS

Memory Address	Instruction	Hexcode
8000	LXI B, 3000H	01
8001		00
8002		30
8003	LXI D, 4000H	11
8004		00
8005		40
8006	MVI L, 0AH	2E
8007		0A
8008	TOP; LDAX B	0A
8009	STAX D	12
800A	INX B	03
800B	INX D	13
800C	DCR L	2D
800D	JNZ TOP	C2
800E		08
800F		80
8010	HLT	76

6. RESULT:-

INPUT

Memory Address	Data
3000H	01H
3001H	03H
3002H	06H
3003H	0AH
3004H	0FH
3005H	FFH
3006H	09H
3007H	89H
3008H	09H
3009H	0AH

OUTPUT

Memory Address	Data
4000H	01H
4001H	03H
4002H	06H
4003H	0AH
4004H	02H
4005H	FFH
4006H	09H
4007H	09H
4008H	06H
4009H	0AH

1. TITLE:- Exchange block of data

2. OBJECTIVES:- To exchange the value or data between memory location 3000H - 3009H and 4000H - 4009H.

3. REQUIREMENTS:-

- 8085 microprocessor kit

- 5V power supply

4. ALGORITHM:-

Step 1:- Start

Step 2:- Load memory location 3000H and 4000H in pair registers BC and DE.

Step 3:- Move immediate data 0AH in L register

Step 4:- Load the content of memory location pointed by BC pair in H register and memory location pointed by DE pair in accumulator.

Step 5:- Store the content of accumulator to memory location pointed by BC pair and move the content of H register to accumulator and store the content of accumulator in memory location pointed by DG pair register.

Step 6:- Increment the content of BC and DE pair registers and decrement the content of L register.

Step 7:- Jump if zero flag is not set

Step 8:- Terminate the program

5. PROCESS:-

Memory Address	Instruction	Hexcode
9000H	LXI B, 3000H	01
9001H		00
9002H		30
9003H	LXI D, 4000H	11
9004H		00
9005H		64
9006H	MVI L, 0AH	2E
9007H		0A
9008H	TOP : LDAX B	0A
9009H	MOV H, A	67
900AH	LDAX D	1A
900BH	STAX B	02
900CH	MOV A, H	7C
900DH	STAX D	12
900EH	INX B	03
900FH	ZNX D	13
9010	DCR L	2D
9011	JNZ TOP	C2
9012		08
9013		90
9014	HLT	76

6. RESULT :-

INPUT

Memory Address	3000H	3001H	3002H	3003H	3004H	3005H	3006H	3007H	3008H	3009H
Data	03H	06H	09H	0AH	0BH	0CH	0DH	0EH	0FH	06H

OUTPUT

Memory Address	3000H	3001H	3002H	3003H	3004H	3005H	3006H	3007H	3008H	3009H
Data	06H	09H	03H	02H	08H	00H	01H	0CH	FFH	A8H

Memory Address	4000H	4001H	4002H	4003H	4004H	4005H	4006H	4007H	4008H	4009H
Data	03H	06H	09H	0AH	0BH	0CH	0DH	0EH	0FH	06H