POORNIMA INSTITUTE OF ENGINEERING & TECHNOLOGY, JAIPUR

Department of Computer Science and Engineering

Lab Manual

Microprocessor & Interface Lab
<4CS4-21>

Branch	**CS**	**Name of Lab**	**Microprocessor and Interface Lab**
Session	**2020-21**	**Subject Code**	**4CS4-21**
Year	**2nd Year**	**Faculty**	**Abhishek Dadhich**
Semester	**4th Semester**	**Lab Assistant**	**Ms. Divya Rastogi**

Document No.	PIET/ECE/CS/2019/III/LAB/MP/ 4CS4-21	Created By	Abhishek Dadhich
Version	4CS4-21/1.2	Verified By	
Authorized By (HOD)	Mr. Deepak Moud		

LAB RULES

DO'S	DON'TS
Be regular to the lab.	Do not come late to the lab.
Follow proper dress code.	Do not throw the connecting wires on the floor.
Maintain Silence.	Do not operate µp/IC trainer kits unnecessarily.
Know the theory behind the experiment before coming to the lab.	Don't bring any external material inside the LAB.
Arrange the chairs/stools and equipment properly before leaving the lab.	Do not panic if you don't get the output.
Avoid unnecessary talking while doing the experiment.	Don't carry any LAB equipment outside the lab.
Keep the Table clean.	Do not try to repair or tamper lab equipment.

INSTRUCTIONS

Before entering in the lab

- All the students are supposed to prepare the theory regarding the next experiment/ Program.
- Students are supposed to bring their lab records as per their lab schedule.
- Previous experiment/program should be written in the lab record.
- If applicable trace paper/graph paper must be pasted in lab record with proper labeling.
- All the students must follow the instructions, failing which he/she may not be allowed in the lab.

While working in the lab

- Adhere to experimental schedule as instructed by the lab in-charge/faculty.
- Get the previously performed experiment/ program signed by the faculty/ lab in charge.
- Get the output of current experiment/program checked by the faculty/ lab in charge in the lab copy.
- Each student should work on his/her assigned computer at each turn of the lab.
- Take responsibility of valuable accessories.

SYLLABUS

Practical Hrs: 2 Hrs/ week Maximum Marks = 50

EXPERIMENT NO	NAME OF EXPERIMENT
1	Add the contents of memory locations XX00 &XX01 & place the result in memory location XX02.
2	Add the 16 bit numbers stored in memory location & store the result in another memory location.
3	Transfer a block of data from memory location XX00 to another memory location XX00 in forward & reverse order.
4	Write a program to Swap two blocks of data stored in memory.
5	Write a program to find the square of a number.
6	Write a main program & a conversion subroutine to convert Binary to its equivalent BCD.
7	Write a program to find largest & smallest number from a given array.
8	Write a program to Sort an array in ascending & descending order.
9	Write a program to multiply two 8 bit numbers whose result is 16 bit.
10	Write a program of division of two 8 bit numbers.
11	Write a program to control the speed of a motor.

EVALUATION SCHEME

(To be verified from RTU syllabus)

Name of Exam	Conducted By	Experiment Marks	Viva Marks
I Mid Term	PIET	15	5
II Mid Term	PIET	15	5
End Term	RTU	15	5

Name of Exam	Conducted By	Performance Marks	Attendance Marks
Sessional	PIET	15	5

Distribution of Lab Record Marks per Experiment

Attendance	Record	Performance	Total
2	3	5	10

INDEX

LAB PLAN

Total number of experiment 12

Total number of turns required 12

Number of turns required for

No.	Experiments	Lab Turn
1	Zero Lab (Lab rules, Introduction to different types of microprocessor and microcontroller. Application of Microprocessor and microcontroller	Turn-01
2	Add the contents of memory locations XX00 &XX01 & place the result in Memory location XX02 and Register pair.(With Carry and Without Carry)	Turn-02
3	Add the 16 bit numbers stored in memory location & store the result in Another memory location. and Register pair.(With Carry and Without Carry)	Turn-03
4	**Introduction of Arduino and implementing the interfacing between LCDs (16*2).**	Turn-04
5	Transfer a block of data from memory location XX00 to another memory location XX00 in forward & reverse order.	Turn-05
6	Write a program to swap two blocks of data stored in memory.	Turn-06
	I MIDTERM EXAMINATION	
7	Write a program to add ten 8 bit numbers stored in memory location XX00 & store the result in another memory location and in Register Pair.	Turn-07
8	Write a program to multiply two 8 bit numbers whose result is 16 bit, & store the result in another memory location and in Register Pair.	Turn-08
9	Write a program to find the square of a number. & store the result in another memory location and in Register.	Turn-09
10	Write a program of division of two 8 bit numbers.(Using DAD Instruction and without DAD Instruction.)	Turn-10
11	Write a program to find largest & smallest number from a given array.	Turn-11
12	**Introduction to Raspberry PI.Implementing the interfacing between RPi and LED**	Turn-12

Distribution of lab hours

Attendance	05 minutes
Explanation of features of language	15 minutes
Explanation of experiment	15 minutes
Performance of experiment	70 minutes
Viva / Quiz / Queries	15 min

Hardware required

Microprocessor kit, 8255 study card, 40 pin FRC cable etc

Lab Objective and Outcome

Lab Objective:

- To expose students to the operation of typical microprocessor (8085) trainer kit.
- To prepare the students to be able to solve different problems by developing different programs.
- To develop the quality of assessing and analyzing the obtained data.
- Digital techniques are useful because it is easier to get an electronic device to switch into one of a number of known states than to accurately reproduce a continuous range of values.

Lab Outcomes:

Upon completion of the subject, students will be able to:

1. Identify relevant information to supplement to the Microprocessor and Microcontroller course.
2. Set up programming strategies and select proper mnemonics and run their program on the training boards.
3. Practice different types of programming keeping in mind technical issues and evaluate possible causes of discrepancy in practical experimental observations in comparison.
4. Develop testing and experimental procedures on Microprocessor and Microcontroller analyze their operation under different cases.
5. Prepare professional quality textual and computational results, incorporating accepted data analysis and synthesis methods, simulation software, and word-processing tools.
6. Primarily via team-based laboratory activities, students will demonstrate the ability to interact effectively on a social and interpersonal level with fellow students, and will demonstrate the ability to divide up and share task responsibilities to complete assignments.

List of Experiments (As per RTU)

EXPERIMENT NO	NAME OF EXPERIMENT
1	Add the contents of memory locations XX00 &XX01 & place the result in memory location XX02.
2	Add the 16 bit numbers stored in memory location & store the result in another memory location.
3	Transfer a block of data from memory location XX00 to another memory location XX00 in forward & reverse order.
4	Write a program to Swap two blocks of data stored in memory.
5	Write a program to find the square of a number.
6	Write a main program & a conversion subroutine to convert Binary to its equivalent BCD.
7	Write a program to find largest & smallest number from a given array.
8	Write a program to Sort an array in ascending & descending order.
9	Write a program to multiply two 8 bit numbers whose result is 16 bit.
10	Write a program of division of two 8 bit numbers.
11	Write a program to control the speed of a motor.

Procedure

- One experiment with its variants will be performed by the Batch in Each Lab.
- Students will perform experiments in the group of four (Self Help Group) for a time slot and so on.
- For first 30 Minutes logic for the Experiments will be explained.
- Students will Develop Coding in the remaining Session and perform the experiments.
- In last 30 minutes viva and lab records will be evaluated.
- Students will be directed to write all the experiments in their Lab Records.

PROCEDURE

1. Enter the Inputs with the help of 1 and then press A to write a program in assembly language then press enter then enter RAM address(initial programming address) in it .
2. The execution of the program is shifted to that particular memory address now start writing your code from that ram address.
3. After finishing the code press reset .
4. Give inputs to specified memory location using **MXXXX**(here XXXX is specified memory adderss in program) and then give input to that location and than press reset.
5. Execute the program using command **GXXXX** (here XXXX is starting RAM address given by you)then press **shift and $** key simultaneously so that executing displays on the LCD and reset the device.
6. Now check the output using command **M** address on a particular memory location.
7. Write down the corresponding address and data of inputs and outputs.

List of Lab Exercises (ROTOR-1)

LAB No.	Experiment No.	Name of Experiment	Assigned Batch
1	1	Zero Lab (Lab rules, Introduction to different types of microprocessor and microcontroller. Application of Microprocessor and microcontroller	
2	2	Add the contents of memory locations XX00 &XX01 & place the result in Memory location XX02 and Register pair.(With Carry and Without Carry)	
3	3	Add the 16 bit numbers stored in memory location & store the result in Another memory location. and Register pair.(With Carry and Without Carry)	
4	4	**Introduction of Arduino and implementing the interfacing between LCDs (16*2).**	
5	5	Transfer a block of data from memory location XX00 to another memory location XX00 in forward & reverse order.	
6	6	Write a program to swap two blocks of data stored in memory.	

List of Lab Exercises (ROTOR-2)

LAB No.	Experiment No.	Name of Experiment	Assigned Batch
7	7	Write a program to add ten 8 bit numbers stored in memory location XX00 & store the result in another memory location and in Register Pair.	
8	8	Write a program to multiply two 8 bit numbers whose result is 16 bit, & store the result in another memory location and in Register Pair.	
9	9	Write a program to find the square of a number. & store the result in another memory location and in Register.	
10	10	Write a program of division of two 8 bit numbers.(Using DAD Instruction and without DAD Instruction.)	
11	11	Write a program to find largest & smallest number from a given array.	
	12	**Introduction to Raspberry PI.Implementing the interfacing between RPi and LED**	

Experiments

Experiment No 1

Aim/Object:

Write a program to add two 8 bit numbers.

Equipment Required:

Microprocessor trainer Kits, Keyboard.

Programming Code:

(Two 8 bit no stored in internal registers)

Address	Label	Mnemonics	Op Code	Comment/Syntax
2000		MVI A,05	3E	Move Immediately 05 to accumulator
2001			05	Next subsequent 8 bit data
2002		MVI B,05	06	Move Immediately 05 to register B
2003			05	Next subsequent 8 bit data
2004		ADD B	80	B+A➔A
2005		STA 3012	32	Store Accumulator value to a address
2006			12	Lower address bits
2007			30	Higher address bits
2008		HLT	76	Terminate the program

(Two 8 bit no stored in memory address)

Address	Label	Mnemonics	Op Code	Comment/Syntax
2000		LXI H,3000	21	Move 3000 into Hl Pair
2001			00	Lower address bits
2002			30	Higher address bits
2003		MOV A,M	7E	Copy data from M to A
2004		INX H	23	Increment data in HL by 1
2005		ADD M	86	M+A➔A
2006		STA 3012	32	Store Accumulator value to a address
2007			12	Lower address bits
2008			30	Higher address bits
2009		HLT		Terminate the program

Procedure:

8. Enter the Inputs with the help of 1 and then press A to write a program in assembly language then press enter then enter RAM address(initial programming address) in it .
9. The execution of the program is shifted to that particular memory address now starts writing your code from that ram address.
10. After finishing the code press reset.
11. Give inputs to specified memory location using **MXXXX**(here XXXX is specified memory adderss in program) and then give input to that location and than press reset.

12. Execute the program using command **GXXXX** (here XXXX is starting RAM address given by you)then press **shift and $** key simultaneously so that executing displays on the LCD and reset the device.
13. Now check the output using command **M** address on a particular memory location.
14. Write down the corresponding address and data of inputs and outputs.

Input \Output (Input \Output should be on the left page)**:**
Input:05H, 02H
Output: 07H
Result:
By the following experiment we have done addition of two 8 bit data.

Experiment No.2

Aim/Object:

Write a program to add two 16 bit numbers.

Equipment Required:

Microprocessor trainer Kits, Keyboard.

Programming Code:

Address	Label	Mnemonics	Op Code	Comment/Syntax
3000		LHLD 2000	2A	Get 1st 16 bit no in HL
3001			00	
3002			20	
3003		XCHG	EB	Same 1st 16 bit no in DE
3004		LHLD 2002H	2A	Get 2nd 16 bit no in HL
3005			02	
3006			20	
3007		MOV A,E	7B	Get lower byte of 1st no
3008		ADD L	85	
3009		MOV L,A	6F	A→L
300A		MOV A,D	7A	D→A
300B		ADC H	8C	A+CY+H→A
300C		MOV H,A	67	A→H
300D		SHLD 2004	22	H-L→2004 & 2005
300E			04	Lower address
300F			20	Higher address
3010		HLT	76	Terminate program

Procedure:

1. Enter the Inputs with the help of 1 and then press A to write a program in assembly language then press enter then enter RAM address (initial programming address) in it .
2. The execution of the program is shifted to that particular memory address now starts writing your code from that ram address.
3. After finishing the code press reset.
4. Give inputs to specified memory location using **MXXXX** (here XXXX is specified memory address in program) and then give input to that location and then press reset.

5. Execute the program using command **GXXXX** (here XXXX is starting RAM address given by you)then press **shift and $** key simultaneously so that executing displays on the LCDand reset the device.
6. Now check the output using command **M** address on a particular memory location.
7. Write down the corresponding address and data of inputs and outputs.

Input \Output (Input \Output should be on the left page)**:**
Input: 0002H, 0003
Output: 0005H
Result:
By the following experiment we have done addition of two 16 bit data.

Experiment No.3

Aim/Object:

Transfer a block of data from one memory location to another memory location in Forward and Reverse Order.

Equipment Required:

Microprocessor trainer Kits, Keyboard.

Programming Code:

(Forward order)

Memory address	Mnemonics	label	Op Code	Comments/syntax
2000	LXI H 2500		21	2500➔H-L
2001			00	Lower Address
2002			25	Higher Address
2003	LXID3500		11	3500➔D-E
2004			00	Lower Address
2005			35	Higher Address
2006	MVIC,05		OE	05➔C
2007			O5	value
2008	MOV A,M	L1	7E	[2500]➔A
2009	STAXD		12	A➔[D-E]
200A	INXH		23	H-L+1➔H-L
200B	INXD		13	D-E+1➔D-E
200C	DCRC		0D	C-1➔C
200D	JNZ 2008		C2	Jump at 2008
200E			08	
200F			20	
2010	HLT		76	Terminate program

(Reverse order)

Memory Address	Label	Mnemonics	Op Code	Comment/syntax
2000		LXIH2500	21	2500➔H-L
2001		-	00	Lower Address
2002		-	25	Higher Address
2003		LXID3504	11	3500➔D-E
2004		-	04	Lower Address
2005		-	35	Higher Address
2006		MVI,C 05	0E	05➔C
2007		-	05	value
2008		MOV A,M	7E	[2500]➔A
2009		STAXD	12	A➔[D-E]
200A		INXH	23	H-L+1➔H-L
200B		DCXD	1B	D-E+1➔D-E
200C		DCRC	0D	C-1➔C

200D		**JNZ 2008**	**C2**	Jump at 2008
200E		**-**	**08**	
2010F		**-**	**20**	
2010		**HLT**	**76**	Terminate program

Procedure:

1. Enter the Inputs with the help of 1 and then press A to write a program in assembly language then press enter then enter RAM address (initial programming address) in it .
2. The execution of the program is shifted to that particular memory address now starts writing your code from that ram address.
3. After finishing the code press reset.
4. Give inputs to specified memory location using **MXXXX**(here XXXX is specified memory address in program) and then give input to that location and than press reset.
5. Execute the program using command **GXXXX** (here XXXX is starting RAM address given by you)then press **shift and $** key simultaneously so that executing displays on the LCD and reset the device.
6. Now check the output using command **M** address on a particular memory location.
7. Write down the corresponding address and data of inputs and outputs.

Input \Output (Input \Output should be on the left page)**:**

Input : 2500:01 3500:00
2501:02 3501:00
2502:03 3502:00
2503:04 3503:00
2504:05 3504:00

Output:

2500:01 3500:01
2501:02 3501:02
2502:03 3502:03
2503:04 3503:04
2504:05 3504:05

Result:

By the following experiment we have transferred one block of data to another block

Experiment No.4

Aim/Object:

Write a program to swap two blocks of data stored in memory locations.

Equipment Required:

Microprocessor trainer Kits, Keyboard.

Programming Code:

S.NO.	Memory Address	Label	Mnemonics	Op Code	Comment/syntax
1	2000		LXIH,2500	21	2500➔H-L
2	2001		-	00	Lower Address
3	2003		-	25	Higher Address
4	2004		LXID,3500	11	3500➔D-E
5	2005		-	00	Lower Address
6	2006		-	35	Higher Address
7	2007	L1	MVI C, 05	0E	05➔C
8	2008		-	05	
9	2009		MOV B,M	46	[2500]➔B
10	200A		LDAXD	1A	[3500]➔A
11	200B		MOVM, A	77	A➔M
12	200C		MOV A,B	78	B➔A
13	200D		STAXD	12	A➔[3500]
14	200E		INXH	23	H-L+1➔H-L
15	201F		INXD	13	D-E+1➔D-E
16	2010		DCRC	0D	C-1➔C
17	2011		JNZ 2007	C2	Jump on 2008
18	2012		-	07	
19	2012		-	20	
20	2013		HLT	76	Terminate Program

Procedure:

1. Enter the Inputs with the help of 1 and then press A to write a program in assembly language then press enter then enter RAM address (initial programming address) in it .
2. The execution of the program is shifted to that particular memory address now starts writing your code from that ram address.
3. After finishing the code press reset.
4. Give inputs to specified memory location using **MXXXX** (here XXXX is specified memory address in program) and then give input to that location and then press reset.

5. Execute the program using command **GXXXX** (here XXXX is starting RAM address given by you)then press **shift and $** key simultaneously so that executing displays on the LCD and reset the device.
6. Now check the output using command **M** address on a particular memory location.
7. Write down the corresponding address and data of inputs and outputs.

Input \Output (Input \Output should be on the left page)**:**

Input : 2500:01 3500:06
2501:02 3501:07
2502:03 3502:08
2503:04 3503:00
2504:05 3504:09

Output:

2500:06 3500:01
2501:07 3501:02
2502:08 3502:03
2503:00 3503:04
2504:09 3504:05

Result:

By the following experiment we have done Swapping of two 8 bit data.

Experiment No.5

Aim/Object:

Write a program to find the square of an 8 bit number.

Equipment Required:

Microprocessor trainer Kits, Keyboard.

Programming Code:

S.NO.	Memory Address	Label	Mnemonics	Op Code	Comment/syntax
1	2000		LXI H, 6200H	21	6200➔H-L
2	2001		-	00	Lower Address
3	2002		-	62	Higher Address
4	2003		LXI D, 6100H	11	6100➔D-E
5	2004		-	00	Lower Address
6	2005		-	61	Higher Address
7	2006		LXI B, 7000H	01	7000➔B-C
8	2007		-	00	Lower Address
9	2008		-	70	Higher Address
10	2009		LDAX,D	1A	[D-E]➔A
11	200A		MOV L,A	6F	A➔L
12	200B		MOV A,M	7E	M➔A
13	200C		STAXB	02	A➔[B-C]
14	200D		INX D	13	D-E+1➔D-E
15	200E		INX B	03	B-C+1➔B-C
16	200F		MOV A,C	79	C➔A
17	2010		CPI 05H	FE	
18	2011		JNZ 200A	C2	Jump on address
19	2012		HLT	76	Terminate Program

Procedure:

1. Enter the Inputs with the help of 1 and then press A to write a program in assembly language then press enter then enter RAM address(initial programming address) in it .
2. The execution of the program is shifted to that particular memory address now start writing your code from that rams address.

3. After finishing the code press reset.
4. Give inputs to specified memory location using **MXXXX**(here XXXX is specified memory adders in program) and then give input to that location and then press reset.
5. Execute the program using command **GXXXX** (here XXXX is starting RAM address given by you)then press **shift and $** key simultaneously so that executing displays on the LCD and reset the device.
6. Now check the output using command **M** address on a particular memory location.
7. Write down the corresponding address and data of inputs and outputs.

Input \Output (Input \Output should be on the left page)**:**
Input: 02H
Output: 04H
Result:
By the following experiment we have Calculated Square of two 8 bit data.

Experiment No.6

Aim/Object:

Write a main program and a conversion subroutine to convert the binary to its equivalent BCD.

Equipment Required:

Microprocessor trainer Kits, Keyboard.

Programming Code:

Memory Address	Mnemonics	Op Code	Comments/syntax
2000	LXI SP 27FF	31	27FF→SP
2001	-	FF	
2002	-	27	
2003	LDA 6000H	3A	[6000]→A
2004	-	00	
2005	-	60	
2006	Call 3000	CD	PC→3000
2007	HLT	76	Terminate t

Subroutine to convert binary number into its equivalent BCD number.

Binary to BCD:-

Memory Address	Label	Mnemonics	Op Code	Comments/syntax
3000		PUSH B	C5	[BC]→SP, SP+1
3001		PUSH D	D5	[DE]→SP, SP+1
3002		MVI B,64H	06	64→B
3003			64	
3004		MVI C,0AH	0E	0A→C
3005			0A	
3006		MVI D,00H	16	00→D
3007			00	
3008		MVI E,00H	1E	00→E
3009			00	
300A	STEP1	CMP B	B8	Compare A and B register
300B		JC STEP 2	DA	Jump when carry is set
300C		SUB B	90	A-B→A
300D		INR E	1C	E+1→A
300E		JMP STEP 1	C3	Jump on address/step 1
300F	STEP2	CMP C	B9	Compare A with C
3010		JC STEP 3	DA	Jump when carry is set
3011		SUB C	91	A-C→A

3012		**INR D**	**14**	**D+1➔D**
3013		**JMP STEP 2**	**C3**	**Jump on step 2**
3014				
3015	**STEP3**	**STA 6100H**	**32**	**A➔[6100]**
3016			**00**	
3017			**61**	
3018		**MOV A, D**	**7A**	**D➔A**
3019		**STA 6101H**	**32 , 01 ,61**	**A➔[6101]**
301A		**MOV A,E**		**E➔A**
301B		**STA 6102H**	**32 ,02 ,61**	**A➔[6102]**
301C		**POP D**	**D1**	**SP,SP+1➔[DE]**
301D		**POP B**	**C1**	**SP,SP+1➔[BC]**
301E		**RET**	**C9**	**Return to the program**

Procedure:

1. Enter the Inputs with the help of 1 and then press A to write a program in assembly language then press enter then enter RAM address(initial programming address) in it .
2. The execution of the program is shifted to that particular memory address now start writing your code from that rams address.
3. After finishing the code press reset.
4. Give inputs to specified memory location using **MXXXX**(here XXXX is specified memory adders in program) and then give input to that location and then press reset.
5. Execute the program using command **GXXXX** (here XXXX is starting RAM address given by you)then press **shift and $** key simultaneously so that executing displays on the LCD and reset the device.
6. Now check the output using command **M** address on a particular memory location.
7. Write down the corresponding address and data of inputs and outputs.

Input \Output (Input \Output should be on the left page)**:**
Input : 0AH
Output: 10
Result:
By the following experiment we converted a binary number into its BCD equivalent.

Experiment No.7

Aim/Object:

Write a program to find largest & smallest number from a given array.

Equipment Required:

Microprocessor trainer Kits, Keyboard.

Programming Code:

(SMALLEST NUMBER)

Memory Address	Label	Mnemonic	Op code	Comments
2000		LXI H, 2500	21H	Load reg^r pair H-L with Address 2500
2001			00H	
2002			25H	
2003		LDA 3000	3AH	Load acc^r with data Content stored at memory Location 3000H
2004			00H	
2005			30H	
2006		MOV B, A	47H	Mov content of A intoB
2007		DCR B	05H	Decrease reg^r B by 1
2008		MOV A,M	7EH	Move content of memory location into acc^r
2009	L2	INX H	23H	Increase reg^r pair H by1
200A		CMP M	BEH	Compare acc^r and present memory value
200B	L1	JNC 2007	D2H	If carry is no generated During comparision jump At memory location 2007
200C			07H	
200D			20H	
200E		DCR B	05H	Decrement reg^r B by 1
200F	L2	JNZ 2009	C2H	Jump to memory location 2007 if no. Zero is generated
2010			09H	
2011			20H	
2012		STA 2500	32H	Store data content of Acc^r Into 3500
2013			00H	
2014			35H	
2015		HLT	76H	End of program

Output: -

Size of array = 3000:05

Input: 2500=4, 2501=6, 2502=1, 2503=3, 2504=8

Output: 3500=01
(LARGEST NUMBER)

Address	Label	Mnemonic	Op code	Comments
2000		LXI H, 2500	21H	Load reg^r pair H-L with Address 2500
2001			00H	
2002			25H	
2003		LDA 3000	3AH	Load acc^r with data Content stored at Memory 3000
2004			00H	
2005			30H	
2006		MOV B, A	47H	Mov content from Acc^r to B
2007	L1	DCR B	05H	Decrement reg^r B by 1
2008		MOV A,M	7EH	Move data from memory to acc^r
2009	L2	INX H	23H	Increase in H-L pair by 1
200A		CMP M	BEH	Compare acc^r and H-L pair data
200B	L1	JC 2007	DAH	Jump If carry generated During comparision at 2007
200C			07H	
200D			20H	
200E		DCR B	05H	Decrement in value of B.
200F	L2	JNZ 2009	02H	
2010			09H	
2011			20H	
2012		STA 3500	32H	Store data of Acc^r at
2013			60H	address 3500
2014			35H	
2015		HLT	76H	End of program

Procedure:

1. Enter the Inputs with the help of 1 and then press A to write a program in assembly language then press enter then enter RAM address (initial programming address) in it.
2. The execution of the program is shifted to that particular memory address now start writing your code from that rams address.
3. After finishing the code press reset.
4. Give inputs to specified memory location using **MXXXX** (here XXXX is specified memory adderss in program) and then give input to that location and then press reset.

5. Execute the program using command **GXXXX** (here XXXX is starting RAM address given by you)then press **shift and $** key simultaneously so that executing displays on the LCD and reset the device.
6. Now check the output using command **M** address on a particular memory location.
7. Write down the corresponding address and data of inputs and outputs.

Output: -
Size of array = 3000:05
Input: 2500=4, 2501=3, 2502=9, 2503=6, 2504=8
Output: 3500=09
Result:
By the following experiment we have done searching of Smallest & Largest No. from a given array.

Experiment No.8

Aim/Object:

Write a program to Sort an array in ascending & descending order.

Equipment Required:

Microprocessor trainer Kits, Keyboard.

Programming Code:

(Ascending order)

Address	Label	Mnemonic	Op code	Comments
2000		MVI C,04H		Load Acc^r Cwith 04H
2001				
2002		DCR C		Decrement in content of Reg^r C by 1
2003		LXI H,3001H		Load H-L pair From 3001H
2004				
2005				
2006		MOV D,C		Move Content of C in D Reg^r
2007		MOV A,M		Move Content of M in A Reg^r.
2008		INX H		Increase in H-L pair by 1
2009		MOV B,M		Move Content of M in B Reg^r
200A		DCX H		Decrease in H-L pair by 1
200B		CMP B		Compare Content of B Reg^r With A Reg^r.
200C		JC 2013		
200D				
200E				
200F		MOV M,B		Move Content of B in Memory
2010		INX H		
2011		MOV M, A		Move Content from Acc^r to M
2012		DCX H		Decrease in H-L pair by 1
2013		INX H		Increase in H-L pair by 1
2014		DCR D		Decrease in D Reg^r by 1
2015		JNZ 2007		
2016				
2017				

2018		**DCR C**		**Decrease in C Reg^{r} by 1**
2019		**JNZ 2003**		
201A				
201B				
201C		**HLT**	**76 H**	**End of program**

(For descending order)

Memory address	Label	Mnemonics	Op Code	Comments/syntax
2000		**LXI H,2500**		**Initialize HL as the memory pointer**
2001			**00**	
2002			**25**	
2003		**MOVC ,M**		**Get count in the C register**
2004		**DCR C**		
2005	**REPEAT**	**MOV D,C**		**Copy it in D register**
2006		**LXI H,2501**		**Read first value**
2007			**01**	
2008			**25**	
2009	**LOOP**	**MOV A,M**		**Get first value in the A**
200A		**INX H**		**Increment pointer**
200B		**CMP M**		**Compare A and M**
200C		**JNC 2012/SKIP**		**Jump when carry is reset**
200D			**12**	
200E			**20**	
200F		**MOV B,M**		**Exchange data**
2010		**MOV M,A**		
2011		**DCX H**		**Decrease the pointer**
2012		**MOV M,B**		
2012		**INX H**		**Increment pointer**
2014	**SKIP**	**DCR D**		**Decrease counter**
2015		**JNZ 2008/LOOP**		**Jump when the zero is reset**
2016				
2017				

2018		**DCR C**		**Decrement C**
2019		**JNZ 2005/REPEAT**		**Repeat the loop until count is equal to zero**
201A			**05**	
201B			**20**	
201C		**HLT**	**76**	**Terminate program**

Procedure:

1. Enter the Inputs with the help of 1 and then press A to write a program in assembly language then press enter then enter RAM address (initial programming address) in it .
2. The execution of the program is shifted to that particular memory address now start writing your code from that rams address.
3. After finishing the code press reset.
4. Give inputs to specified memory location using **MXXXX**(here XXXX is specified memory adderss in program) and then give input to that location and than press reset.
5. Execute the program using command **GXXXX** (here XXXX is starting RAM address given by you)then press **shift and $** key simultaneously so that executing displays on the LCD and reset the device.
6. Now check the output using command **M** address on a particular memory location.
7. Write down the corresponding address and data of inputs and outputs.

Input \Output (Input \Output should be on the left page):

Inputs:Outputs :(ascending order)	Outputs: (descending)	
2500:06	2500:02	2500:15
2501:02	2501:03	2501:09
2502:03	2502:05	2502:06
2503:09	2503:06	2503:05
2504:15	2504:09	2504:03
2505:05	2505:15	2505:02

Result:

By the following experiment we have obtained values of in ascending and descending order

Experiment No.9

Aim/Object:

Write a program to multiply two 8 bit numbers whose result is 16 bit.

Equipment Required:

Microprocessor trainer Kits, Keyboard.

Memory Address	Mnemonics	Label	Op Code	Comment/syntax
2000	MVI A,00		3E	00→A
2001	-		00	
2002	MVI B,F2		06	F2→B
2003	-		F2	
2004	MVI C,64		0E	64→C
2005	-		64	
2006	MVI D,00		16	00→D
2007	-		00	
2008	ADD.B		80	A+B→A
2009	JNC 200D		D2	Jump Not Carry on 200D
200A	-		0D	
200B	-		20	
200C	INR D		14	D-1→D
200D	DCR C		0D	C-1→C
200E	JNZ 2008		C2	Jump on 2008
200F	-		08	
2010	-		20	
2011	MOV E,A		5F	A→E

Procedure:

1. Enter the Inputs with the help of 1 and then press A to write a program in assembly language then press enter then enter RAM address (initial programming address) in it .
2. The execution of the program is shifted to that particular memory address now start writing your code from that rams address.
3. After finishing the code press reset.
4. Give inputs to specified memory location using **MXXXX** (here XXXX is specified memory address in program) and then give input to that location and than press reset.
5. Execute the program using command **GXXXX** (here XXXX is starting RAM address given by you)then press **shift and $** key simultaneously so that executing displays on the LCDand reset the device.
6. Now check the output using command **M** address on a particular memory location.
7. Write down the corresponding address and data of inputs and outputs.

Input \Output (Input \Output should be on the left page):
Input : 06H, 03H
Output: 11H
Result:
By the following experiment we have done Multiplication of two 8 bit data.

Experiment No.10

Aim/Object:

Write a program of division of two 8 bit numbers.

Equipment Required:

Microprocessor trainer Kits

Keyboard.

Programming Code:

(Division of 8 bit number by 8 bit number)

Memory address	Label	Mnemonics	Op Code	Comments/syntax
2000		LXI H,2500		Get dividend into B register
2001				
2002				
2003		MOV B,M		M➔B
2004		MVI C,00		Clear C for quotient
2005				
2006		INX H		Increment pointer
2007		MOV A,M		Get divisor in A
2008	NEXT	CMP B		Compare A and B
2009		JC 2010/LOOP		Jump when the carry is set
200A			10	
200B			20	
200C		SUB B		A-B➔A
200D		INR C		Increment of C
200E		JMP 2007/NEXT		Jump on NEXT
200F			07	
2010			20	
2011	LOOP	STA 2600		Store remainder into 2600
2012				
2012				
2014		MOV A,C		C➔A
2015		STA 2700		Store quotient into 2600
2016			00	
2017			27	
2018		HLT	76	Terminate program

(Division of 16 bit number by 8 bit number)

Memory address	Label	Mnemonics	Op Code	Comments/syntax
2000		LXI H,2500		Get dividend into B register
2001				
2002				
2003		MOV B,M		M➔B
2004		MVI C,00		Clear C for quotient
2005				
2006		INX H		Increment pointer
2007		MOV A,M		Get divisor in A
2008	NEXT	CMP B		Compare A and B
2009		JC 2010/LOOP		Jump when the carry is set
200A			10	
200B			20	
200C		SUB B		A-B➔A
200D		INR C		Increment of C
200E		JMP 2007/NEXT		Jump on NEXT
200F			07	
2010			20	
2011	LOOP	STA 2600		Store remainder into 2600
2012				
2012				
2014		MOV A,C		C➔A
2015		STA 2700		Store quotient into 2600
2016			00	
2017			27	
2018		HLT	76	Terminate program

Procedure:

1. Enter the Inputs with the help of 1 and then press A to write a program in assembly language then press enter then enter RAM address(initial programming address) in it .
2. The execution of the program is shifted to that particular memory address now start writing your code from that ram address.
3. After finishing the code press reset .

4. Give inputs to specified memory location using **MXXXX**(here XXXX is specified memory adderss in program) and then give input to that location and than press reset.
5. Execute the program using command **GXXXX** (here XXXX is starting RAM address given by you)then press **shift and $** key simultaneously so that executing displays on the LCD and reset the device.
6. Now check the output using command **M** address on a particular memory location.
7. Write down the corresponding address and data of inputs and outputs.

Input \Output (Input \Output should be on the left page):
Input:
2500:05
2501:09
Output:
2600:5
2700:100
Result:
By the following experiment we have observed the division process in 8085 microprocessor

Experiment No.11

Aim/Object:

Write a program to control the stepper motor.

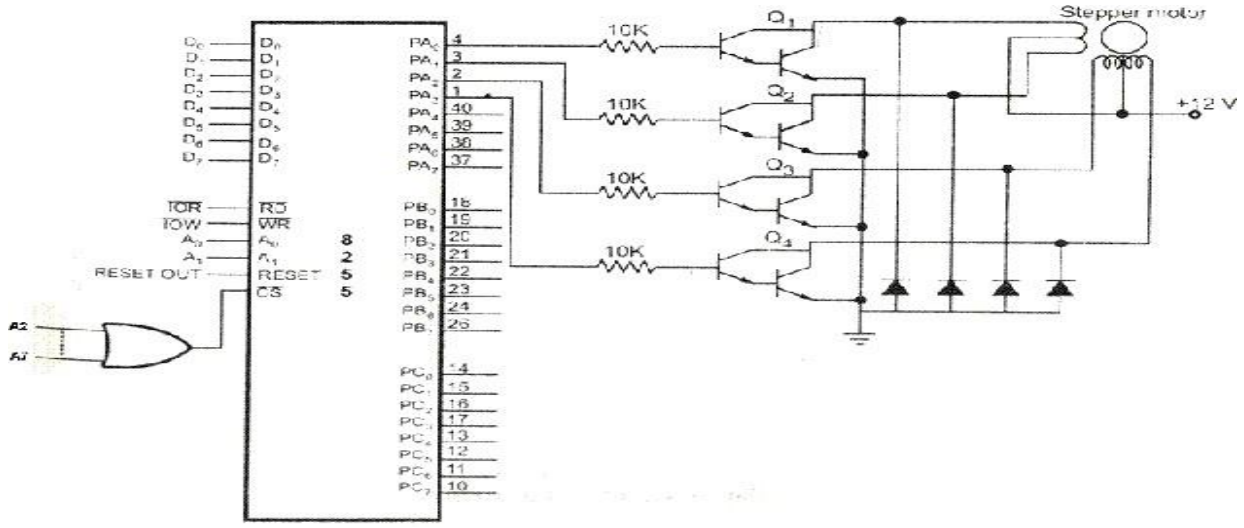
Equipment Required:

Microprocessor trainer Kit, stepper motor interface card and stepper motor
Keyboard and connecting wires.

Theory:

A stepper motor is a digital motor. It can be driven by digital signal. Fig. shows the typical 2 phase motor rated 12V /0.67 A/ph interfaced with the 8085 microprocessor system using 8255. Motor shown in the circuit has two phases, with center-tap winding. The center taps of these windings are connected to the 12V supply. Due to this, motor can be excited by grounding four terminals of the two windings. Motor can be rotated in steps by giving proper excitation sequence to these windings. The lower nibble of port A of the 8255 is used to generate excitation signals in the proper sequence. These excitation signals are buffered using driver transistors. The transistors are selected such that they can source rated current for the windings. Motor is rotated by 1.80 per excitation.

Fig. shows the interfacing diagram to control 12 electric bulbs. Port A is used to control lights on N-S road and Port B is used to control lights on W-E road. Actual pin connections are listed in Table 1 below.



Stepper Motor Control Program:

6000H Excite code DB 03H, 06H, 09H, and OCH: This is the code sequence for clockwise rotation. Subroutine to rotate a stepper motor clockwise by 360° - Set the counts:

Programming Code:

Memory address	Label	Mnemonics	Op Code	Comments/syntax
2000		MVI C,32		Set repetition count to 5010
2001			32	
2002		MCI B,04		Count excitation sequence
2003			04	
2004		LXI H,6000		Initialize pointer
2005			00	
2006			60	
2007	BACK	MOV A,M		Get excitation code
2008		OUT PORT A		Send excite code
2009		CALL 3000/DELAY		wait
200A			00	
200B			30	
200C		INX H		Increment pointer
200D		DCR B		Repeat 4 times
200E		JNZ 2007/BACK		Jump back
200F				
2010				

Delay Subroutine:

Memory address	Label	Mnemonics	Op Code	Comments/syntax
3000		LXI D,COUNT		Initialize counter
3001				
3002				
3003	BACK	DCX D		Decrease counter
3004		MOV A,D		
3005		ORA E		Check the value of the accumulator
3006		JNZ 3003/BACK		Jump back
3007			03	
3008			30	
3009		RET		Return to main program

Procedure:

1. Enter the Inputs with the help of 1 and then press A to write a program in assembly language then press enter then enter RAM address(initial programming address) in it .
2. The execution of the program is shifted to that particular memory address now start writing your code from that ram address.
3. After finishing the code press reset .
4. Give inputs to specified memory location using **MXXXX**(here XXXX is specified memory adderss in program) and then give input to that location and than press reset.
5. Execute the program using command **GXXXX** (here XXXX is starting RAM address given by you)then press **shift and $** key simultaneously so that executing displays on the LCD and reset the device.
6. Now check the output using command **M** address on a particular memory location.
7. Write down the corresponding address and data of inputs and outputs.
8. Now check the output on the stepper motor in forms of various steps which can be varied through the delay time.

Input \Output (Input \Output should be on the left page):
Output: stepper motor steps
Result:
By the following experiment we have observed the various steps to control stepper motor.