UMA CHARAN PATTNAIK ENGINEERING SCHOOL BRAHMAPUR

MICROPROCESSOR AND MICROCONTROLLER LABORATORY MANUAL Pr.3

For 4th SEMESTER, DIPLOMA E&TC Students

PREPARED BY POONAM PANDA, E&TC DEPT

Name of the Student_____

Roll No of the student_____

Laboratory Section_____

[This laboratory manual will guide you to perform the necessary experiments to meet the fulfillment of SCTE&VT syllabus for **MICROPROCESSOR AND MICROCONTROLLER** Lab for 4th semester students. This manual also consists of the supplementary materials like data sheets for some useful components to be used throughout the course.]

TABLE OF CONTENTS

RULES & REGULATIONS

1. **Don't be late** to the lab.
2. Put your signature on attendance sheet first, submit the record and then go to your place correspondence to experiment kit.
3. **Without fair record and lab manual** you are **not allowed** to the lab.
4. Handlings of other things, which are not related to the experiment, are strictly prohibited inside the lab.
5. You may be charged with full cost and additional penalty for destroying the equipments.
6. You should put on your own identity card.
7. Till the time you are inside the lab you should stick to your place and don't move here & there.
8. Before leaving the lab, be sure that power supply is switched off and you have returned the components proper manner otherwise all the group members will be punished.
9. Your duly filled lab manual along with stressing sheet must be verified by any of the instructors present there else your record will not be considered.
10. You yourself will be responsible for exchange or missing of any precious materials.
11. You can demand for marks on the basis of performance ONLY, but can't beg.
12. **Extra labs may be allowed** on your request if your application is forwarded through proper channel **within the time limit**. Beyond the time limit it is never accepted at any situation.

FORMAT FOR RECORD WRITING

1. Maintain the contain page with date of performing the experiment, experiment name, page number.
2. Put the page numbers on the top of the individual page, Date & experiment no.
3. Write the aim of the experiment, apparatus required short theoretical description with required circuit/block diagram, observation & conclusion.
4. Attached the tracing sheet to the left side of the paper.
5. Put your full signature at bottom right corner of the last page of that experiment.
6. Record should be very neat and clean and the entire circuits/block diagram should be drawn with **pencil only.**
7. **Bad hand** writing will **affect** the record mark.
8. **Incomplete record** will lead towards **negative** marking
9. **Data on lab manual** should match with fair record.
10. Arrange the pages properly according to the experiment list (ascending order).
11. Use the record and internal pages **without rolling** (**SUPPLIED format ONLY**).
12. Don't submit loose sheets without record at any condition.

FORMAT

1. Aim of the Experiment
2. Requirements
3. Theory (Methodology algorithm)
4. Coding/programming
5. Exercise
6. Conclusion (On the basis of experiment result)

MARK DISTRIBUTION

Sl. No.	Parts	Marks
1	Attendance	20
2	Record	20+10
3	Experiment	30
4	Via-voice/ Quiz	20
	TOTAL	**100**

I hereby Mr./Ms……..with roll no have understood the above rules and regulations of the **Microprocessor and Microcontroller** Lab and **will accept any kind of punishment given by the instructor for violating the rule** mentioned above.

Date: Signature of the student

LIST OF EXPERIMENTS

1. 1's and 2's Complements

2.a. Addition of two 8 bit numbers resulting 8/16 bit numbers.

2.b. Subtraction of two 8 bit numbers resulting 8/16 bit numbers.

2.c. Multiplication of two 8 bit numbers resulting 8/16 bit numbers.

2.d. Division of two 8 bit numbers resulting 8/16 bit numbers.

3. Compare between two numbers

4. Smallest /Largest number among n numbers in a given data array.

5.a. Addition of 16 bit nos.

5.b. Subtraction of 16 bit nos.

5.c. Multiplication of 16 bit nos.

5.d. Division of 16 bit nos.

6. Sorting an array of numbers in ascending/descending order.

7. Study of stepper Motor and its operations (Clockwise, anticlockwise, angular movement, rotate in various speeds).

8. Study about the operation at 8255 using 8085 and 8051 microcontroller

9. Study about the operation at 8259 programmable interrupt controller.

10. Study of 8279 (keyboard & display interface).

11. Initialize data to register and memory using immediate, register, direct, indirect addressing modes.

***

EXPERIMENT-1

1'S AND 2'S COMPLIMENT OF 8 BITS NUMBER

AIM OF THE EXPERIMENT:-

To find the 1's and 2's compliment of an 8-bit data at specified memory location and store the result in another memory location.

APPARATUS REQUIRED:-

1. Microprocessor Trainer Kit OMEGA OEJ 85A
2. Power Supply

THEORY:-

In 2's compliment of binary number find out the 1's compliment of that binary number then add 1 to the result. Using 8085 only 8 bits operation is possible. Accumulator loads data from memory as well as stores the result.
For example: 2's compliment of 55 h can be found in the following ways
55 h = 01010101 b
Then 1's compliment of 55 h = 10101010 b = AA h
Now add 1 to this, so we get the 2's compliment which is 10101011 b = AB h

PROGRAM:-

```
LDA 9550 H
CMA
INR A
STA 9551 H
HLT
```

PROCEDURE:-

- Key in the opcodes into memory from 8330/9330.
- Enter the data to be complemented at 8252/9551.
- Execute the program and check for result.
- Try changing the data and execute the program each time and check results.

OBJECT CODES FOR 1'S COMPLEMENT

Memory location	Opcodes	Label	Mnemonics	Operand	Description
8330	3A		LDA	8251	Load the content of specified memory location to accumulator content
8331	51				
8332	82				
8333	2F		CMA		It compliment acc. content
8334	32		STA	8252	Store the acc. data in the specific memory location
8335	52				
8336	82				
8337	76		HLT		It stop the program

OBJECT CODES FOR 2'S COMPLEMENT

Memory location	Opcodes	Label	Mnemonics	Operand	Description
9330	3A		LDA	9550	Load the content of specified memory location to accumulator content
9331	50				
9332	95				
9333	2F		CMA		It compliment acc. content
9334	3C		INR	A	Increment of acc. content
9335	32		STA	9551	Store the acc. data in the specific memory location
9336	51				
9337	95				
9338	76		HLT		It stop the program

EXERCISE OF 1'S COMPLEMENT:-

1. Data (9550): 55h Result (9551): (AA)h
2. Data (9550): 62h Result (9551): (9D)h
3. Data (9550): 58h Result (9551): (A7)h

EXERCISE OF 2'S COMPLEMENT:-

1. Data (9550):55h Result (9551): (AB))h
2. Data (9550):62h Result (9551)(9E)h
3. Data (9550):58h Result (9551):((A8)h

CONCLUSION:-

From the above experiment we have found out 1's and 2's compliment of 8-bit data and verified for different examples.

VIVA QUESTIONS:-

1. Define microprocessor.
2. What are the GPR'S &SPR'S in 8085?
3. What do you mean by address and dada bus?
4. Define memory & its types.
5. Study of trainer kit used in lab.

EXPERIMENT: 2(A)

ADDITION OF TWO 8 BIT NO. IN 8085 MICROPROCESSOR

AIM OF THE EXPERIMENT:-

Write a program to add two 8 bit no. and store the result in a memory location.

APPARATUS REQUIERD:-

- Microprocessor Trainer Kit omega OEJ -85 A
- Power Supply

THEORY: -

The 1st data is brought to accumulator A and the 2nd one the other register say B. The addition is done using ADD instruction. The result is then stored at 8127 the ADD instruction affects depending on result.

EXAMPLE:

The two data added area at 8125, 8126 and the result are stored at 8127.
DATA (8125) =(23)H
DATA (8126) =(35)H
RESULT (8127) =(58)H

PROGRAM:-

```
LDA 8125
MOV B, A
LDA 8126
ADD B
STA 8127
HLT
```

PROCEDURE:-

- Specified key in the opcode from the address.
- Enter data at 8125 & 8126 as specified in the example.
- Execute the program and check for the result at 8127.
- Change data at 8125 & 8126 and execute each time and check for result.

OBJECT CODE:-

Memory Location	Opcode	Mnemonics	Operand	Description
8085	3A	LDA	8125	Load the content of memory locatio 8125 into the accumulator (D_1)
8006	25			
8007	81			
8008	47	MOV	B,A	Move the data acc. Now 1st data R=
8009	3A	LDA	8126	Load the content of memory location 8126 into the memory
800A	26			
800B	81			
800C	80	ADD	B	Add the data of register B with accumulator
800D	32	STA	8127	Store the result from acc. into the memory location 8027
800E	27			
800F	81			
8010	76	HLT		Stop the program

EXERCISE:-

- DATA(8150)=(36)h DATA(8151)=(23)h
  RESULT(8152)=(59)h
- DATA(8150)=(22)h DATA(9151)=(21)h
  RESULT(9152)=(43)h
- DATA(9153)=(47)h DATA(9154)=(21)h
  RESULT(9155)=(68) h

CONCLUSION:-

From the above experiment we have find out the addition of two 8 bit numbers

VIVA QUESTIONS:-

1. What is the function of instructions used in this program?
2. How to perform binary, decimal & hexadecimal addition?
3. Define flag register.
4. How many flags are affected after addition? Explain with example.

EXPERIMENT: 2(B)

SUBTRACTION OF TWO 8 BIT NO. IN 8085 MICROPROCESSOR

AIM OF THE EXPERIMENT:-

Write a program to subtract two 8 bit no. and store the result in a memory location.

APPARATUS REQUIERD:-

1. Microprocessor Trainer Kit omega OEJ -85 A
2. Power Supply

THEORY: -

In the experiment the HL register pair is 1st initialized to the store address of memory at which the data is present. The data is brought to accumulator A the other from M is subtracted then result store in accumulator. Store 8 bits results into memory from accumulator. SUB instruction sets and clear flag according to result.

EXAMPLE:

The two data subtracted area at 8150, 8151 and the result are stored at 8152.
DATA (8150) =(47)h
DATA (8151) =(25)h
RESULT (8152) =(22)h

PROGRAM:-

```
LXI H, 8150
MOV A, M
INX H
SUB M
INX H
MOV M, A
HLT
```

PROCEDURE:-

- load the opcode in the specified address.
- Enter data that needs for execution of program at 8150.
- Execute the program and check for the result at 8152.
- Change data at 8150 & 8151 and execute each time and check for result.

OBJECT CODE:-

Memory Location	Opcode	Mnemonics	Operand	Description
9000	21	LXI	H, 8150	Move immediate 8 bit data HL pair.
9001	50			
9002	81			
9003	7E	MOV	A, M	Move the memory content to acc.
9004	23	INX	H	Increment the memory adds.
9005	96	SUB	M	Subtract memory content from accumulator.
9006	23	INX	H	Increments the memory adds.
9007	77	MOV	M, A	Move the content of the memory location to accumulator .
9008	76	HLT		Stop the program

EXERCISE:-

- DATA(8150)= (59)h DATA(8151)=(23)h
  RESULT(8152)=(36)h
- DATA(8150)= (43)h DATA(9151)=(21)h
  RESULT(9152)=(22)h
- DATA(9153)= (68)h DATA(9154)=(21)h
  RESULT(9155)=(47) h

CONCLUSION:-

From the above experiment we find out the subtraction of two 8 bit numbers using 8085IC.

VIVA QUESTIONS:-

1. What are the steps used in this program?
2. How to perform binary, decimal & hexadecimal subtraction.
3. Define addressing modes of 8085.
4. How many flags are affected after subtraction? Explain with example.

EXPERIMENT: 2(C)

MULTIPLICATION OF TWO 8 BIT NO. IN 8085 MICROPROCESSOR

AIM OF THE EXPERIMENT:-

Write a program to multiply two 8 bit no. and store the result in a memory location.

APPARATUS REQUIERD:-

1. Microprocessor Trainer Kit omega OEJ -85 A
2. Power Supply

THEORY: -

In the experiment the HL register pair is 1st initialized to the store address of memory at which the data is present. The data is brought to accumulator A the other from M is added then result store in accumulator. Result again stored into memory from accumulator. ADD instruction sets and clear flag according to result.

EXAMPLE: -

The two data multiplied are stored at 8150, 8151 and the result is stored at 8152.

DATA (8250) =(03)h

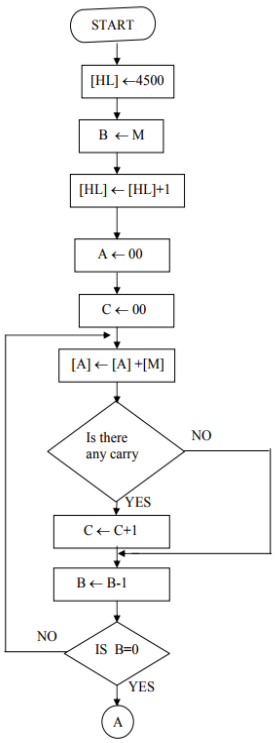
DATA (8251) =(08)h

RESULT (8252) =(18)h

PROCEDURE:-

- Load the opcode in the specified address.
- Enter data that needs for execution of program at specified memory address.
- Execute the program and check for the result at given address.
- Change data and execute program each time and check for result.

FLOW CHART:



OBJECT CODE:-

Memory Address	Opcode	Mnemonics	Operand	Description
8000	21	LXI	H,8250	Load 8 bit data from memory location 8250
8001	50			
8002	82			
8003	4E	MOV	C,M	Move 8 bit data from memory to register C
8004	3E	MVI	A,00	Initialize accumulator to 00
8005	00			
8006	23	INX	H	Increment address of HL pair by 1
8007	86	ADD	M	Add content of accumulator with data stored in given memory location
8008	0D	DCR	C	Decrement counter by 1
8009	C2	JNZ	8007	Jump to given memory location and add the content again
800A	07			
800B	80			
800C	32	STA	9250	Store the result from accumulator to memory location 9250
800D	50			
800E	92			
800F	76	HLT		Stop the program

EXERCISE:-

- DATA(8150)= (59)h DATA(8151)=(23)h
  RESULT(8152)=(36)h
- DATA(8150)= (43)h DATA(9151)=(21)h
  RESULT(9152)=(22)h
- DATA(9153)= (68)h DATA(9154)=(21)h
  RESULT(9155)=(47) h

CONCLUSION: -

From the above experiment we have find out the multiplication of two 8 bit numbers.

VIVA QUESTIONS:-

1. What are the steps used in this program?
2. Define JUMP instruction
3. How many flags are affected after subtraction? Explain with example.

EXPERIMENT – 2(D)

DIVISION OF 8-BIT NUMBER IN MEMORY

AIM OF THE EXPERIMENT :-

Write a program to division two 8-bit numbers in memory and store the result also in memory.

APPARATUS REQUIRED: –

1. Microprocessor trainer kit
2. Power supply

THEORY: –

In this example HL pair register is used to initialize address to memory at which data is stored. Two memory locations are used for storing dividend and divisor. In division dividend is subtracted by divisor. The result is store in memory location and remainder is stored in other memory location.

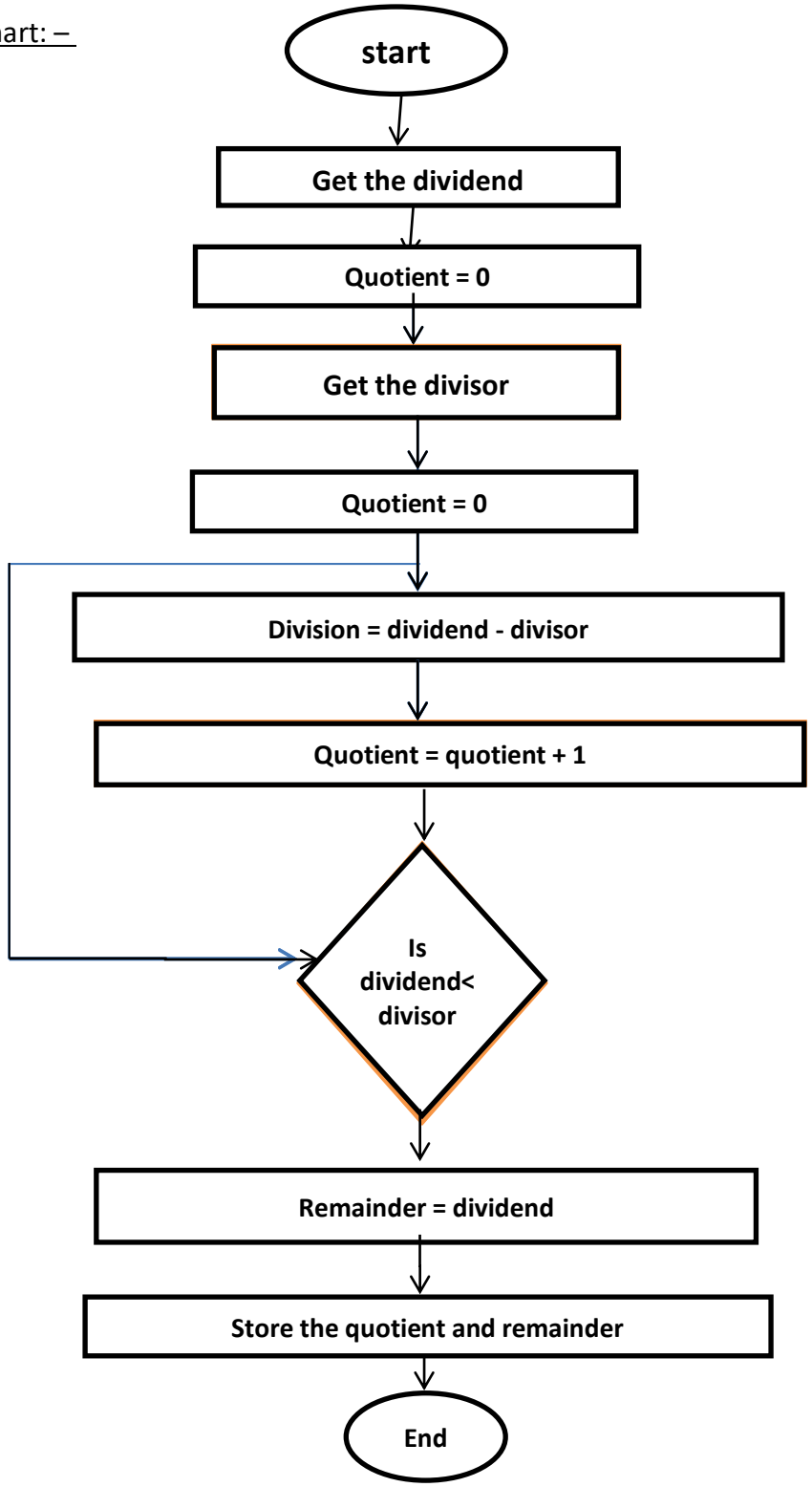
PROGRAM: –

```
LXI H 8050
MOV B,M
MVI C,00
CMP B
JC 9011
SUB B
INR C
JMP 9108
STA 9050
MOV A,C
STA 9051
HLT
```

PROCEDURE: –

- Start the program by loading the HL pair reg. with address of memory location.
- Move the data to B reg.
- Load the second data into accumulator.
- Compare the two no's to check carry.
- Subtract two no's.
- Increment the value of carry.
- Check whether the repeated subtraction is over.
- Then store the result in given memory location.
- Terminal the program.

Flow chart: –


start
Get the dividend
Quotient = 0
Get the divisor
Quotient = 0
Division = dividend - divisor
Quotient = quotient + 1
Is dividend< divisor
Remainder = dividend
Store the quotient and remainder
End

EXERCISE: –

Data – (7A)h,(07)h
Result – (11)h, (03)h

Data – (25)h,(05)h
Result – (05)h, (00)h

Data – (FE)h,(AE)h
Result – (01)h, (06)h

Memory	Opcode	Label	Mnemonics	Operand	Description
9100	21		LXI	H,8050	Load the HL pair reg. with the address 8050 of memory location.
9101	50				
9102	80				
9103	46		MOV	B,M	Copies the content of memory into reg. B.
9104	0E		MVI	C,00	Assigned 00 to c
9105	00				
9106	23		INX	H	Increment reg. pair HL
9107	7E		MOV	A,M	Copies the content of memory into acc.
9108	B8	NEXT	CMP	B	Compare the content of acc. and reg. B
9109	DA		JC	9111	Jump to address 9111 if carry flag is set.
910A	11				
910B	91				
910C	98		SUB	B	Sub the content of acc. With reg. B and store result in acc.
910D	0C		INR	C	Increment the reg. c
910E	C3		JMP	9108	Control with shift to memory address 9108
910F	08				
9110	91				
9111	32	LOOP	STA	9050	Store the remainder at memory address 9050
9112	50				
9113	90				
9114	79		MOV	A,C	Copy the content of reg. into acc.
9115	32		STA	9051	Store the remainder at memory location 9051
9116	51				
9117	90				
9118	76		HLT		Stop execution the program and halts any further execution.

CONCLUSION: –

From the above experiment we have studied about the division operation using 8085 microprocessor.

VIVA QUESTIONS:-

1. Define INX & INR instruction.
2. What do you mean by conditional and unconditional jump?
3. What is status of flag register after division? Explain with example.

EXPERIMENT-3

SMALLEST & LARGEST NO. BETWEEN TWO 8 BIT NUMBERS

AIM OF THE EXPERIMENT:-

To study & find out the smallest & largest no. between two 8 bit numbers and store the result in given memory location.

APPARATUS REQUIRED:-

1. Microprocessor trainer kit omega OEJ-851
2. Power Supply

THEORY:-

The first number brought to the accumulator & the 2nd number of other register named as B. the largest & smallest no can be found by comparing the 2nd no, which is done by executing the CMP instruction.
For largest no during execution of CMP instruction if a carry will generated then the accumulator has largest number. So move the largest number to accumulator by using MOV in instruction on & the store it.
For smallest number during execution of CMP instruction it a carry will generate then accumulator has smallest number if carry will not generate then register having smallest number. So move the smallest number to accumulator by using MOV instruction & then store the result.

PROGRAM:-

Largest no:-

```
LDA   8250H
MOV   B , A
LDA   8251H
CMP   B
JNC   GO
MOV   A , B
GO STA 8252H
HLT
```

Smallest no:-

```
LDA   9250H
MOV   B , A
LDA   9251H
CMP   B
JC    GO
MOV   A , B
GO STA 9252H
HLT
```

PROCEDURE:-

1. Enter the opcode in specified memory location.
2. Enter the data in memory location as specify in example.
3. Execute the program & check the result in given memory location.
4. Change the data in specified memory location & execute the program each time & check result.

Largest number:-

Memory location	Opcode	Label	Mnemonics	Operand	Description
8000	3A		LDA	8250H	Load the content of memory location 8250 into acc.
8001	50				
8002	82				
8003	47		MOV	B , A	Move the copy of content of acc into register B.
8004	3A		LDA	8251H	Load the content of memory with to content register B.
8005	51				
8006	82				
8007	B8		CMP	B	Compare the content of acc with the content.
8008	D2		JNC	800CH	Jump the MC 800c if there is no carry jump the MC 800c if there is no carry.
8009	0C				
800A	80				
800B	78		MOV	A , B	Move the copy of content of acc register B to acc.
800C	32		STA	8252H	Store the content of acc into Memory location 8252.
800D	52				
800E	82				
800F	76		HLT		Stop the program.

Smallest number:-

Memory location	Opcode	Label	Mnemonics	Operand	Description
9000	3A		LDA	9250H	Load the content of memory location 9250 into acc.
9001	50				
9002	92				
9003	47		MOV	B , A	Move the copy of content of acc into register B.
9004	3A		LDA	9251H	Load the content of memory location 9251 into acc.
9005	51				
9006	92				
9007	D8		CMP	B	Compare the content of acc with the content.
9008	D2		JNC	900CH	Jump the Memory location 900c if there is carry.
9009	0C				
900A	90				
900B	78		MOV	A , B	Move the copy of content of register in to acc.
900C	32		STA	9252H	Store the content of acc into Memory location 9252.
900D	52				
900E	92				
900F	76		HLT		Stop the program.

EXERCISE:-

Largest no:-

1. Data –(58)h , (08)h (8250 , 8251)
Result – (58)h (8252)
2. Data – (98)h , (78)h (8350 , 8351)
Result – (98)h (8352)
3. Data – (1A)h , (2C)h (8450 , 8451)
Result – (2C)h (8452)

Smallest no:-

1. Data – (46)h , (57)h (9250 , 9251)
Result – (46)h (9252)
2. Data – (02)h , (09)h (9350, 9351)
Result – (02)h (9352)
3. Data –(A8)h , (2F) h(9450 , 9451)
Result –(A8)h (9452)

CONCLUSION:-

Form the above experiment we s find out the largest & smallest no. between two 8-bit numbers.

VIVA QUESTIONS:-

1. Define JNC & JC Instruction.
2. What are different branch instructions in 8085?
3. What are different logical instructions in 8085

EXPERIMENT-4

LARGEST AND SMALLEST NUMBER IN AN 8-BIT DATA ARRAY

AIM OF THE EXPERIMENT:-

To study and find the largest and smallest number in an 8-bit data array and store the result in the accumulator.

APPARATUS REQUIRED:-

1. **Microprocessor trainer kit omega OEJ-85(A)**
2. **Power supply**

THEORY:-

SMALLEST: A set of numbers given in the consecutive memory location is called as the data array. The first number indicate the counter i.e. among how may number the smallest number can be find out. This can be done by CMP instruction. The first value move the resister name as C. The second data brought to the accumulator, third data brought to another resister, then the CMP instruction is executed other and when the counter become zero then store the result is given memory location unless the loop instruction will be executed again and again.

LARGEST: A set of number given in the consecutive memory location is called as the data array. The first number indicates the counter i.e. among how many number the largest number can be find out. This can be done by the CMP instruction. The 1st data value move to resister named as C, The 2nd data brought to the accumulator, 3rd data brought to other memory counter become zero then store the result. The loop instruction will be executed again and again.

PROCEDURE:-

1. Enter the opcode in the specified memory location.
2. Load counter value at 8501 and number of data in consecutive memory location.
3. Decrement the counter and compare the accumulator data will other.
4. After comparing result stored in the given memory location.
5. Change the data at the given memory location and executed the program each time.

SMALLEST NUMBER:

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC	OPERAND	DESCRIPTION
BED0	21		LXI	H,8501	Move the data of 8501 to HL pair.
BED1	01				
BED2	85				
BED3	4E		MOV	C,M	Move counter in reg C
BED4	23		INX	H	Address of 1st no. in HL pair.
BED5	7E		MOV	A,M	Move the 1st no. in to accumulator.
BED6	0D		DCR	C	Decrement of counter.
BED7	23	LOOP	INX	H	Address of next no. in HL pair.
BED8	BE		CMP	M	Compare next no. with previous smallest no.
BED9	DA		JC	GO	No. i.e. if smallest no. in acc. then jump to next step without moving memory content to acc.
BEDA	DD				
BEDB	BE				
BEDC	7E		MOV	A,M	Move the smallest no to acc.
BEDD	0D	GO	DCR	C	Decrement of counter.
BEDE	C2		JNZ	LOOP	Jump to loop if zero flag is not zero.
BEDF	D7				
BEE0	BE				
BEE1	32		STA	8763 H	Store the smallest no. at the memory 8763H.
BEE2	63				
BEE3	87				
BEE4	76		HLT		It stop the program.

LARGEST NUMBER:

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC	OPERAND	DESCRIPTION
BED0	21		LXI	H,8501	Move the data of 8501 to HL pair.
BED1	01				
BED2	85				
BED3	4E		MOV	C,M	Move counter in reg C
BED4	23		INX	H	Address of 1st no. in HL pair.
BED5	7E		MOV	A,M	Move the 1st no. in to accumulator.
BED6	0D		DCR	C	Decrement of counter.
BED7	23	LOOP	INX	H	Address of next no. in HL pair.
BED8	BE		CMP	M	Compare next no. with previous larger no.
BED9	D2		JNC	GO	No. i.e. if larger no. in acc. Then jump to next step without moving memory content to acc.
BEDA	DD				
BEDB	BE				
BEDC	7E		MOV	A,M	Move the larger no to acc.
BEDD	0D	GO	DCR	C	Decrement of counter.
BEDE	CA		JZ	LOOP	Jump to loop if zero flag is not zero.
BEDF	D7				
BEE0	BE				
BEE1	32		STA	8763 H	Store the larger no. at the memory 8763H.
BEE2	63				
BEE3	87				
BEE4	76		HLT		It stops the program.

EXERCISE:-

Smallest no.

1. Data -07,68,57,48,52,A7,AE,DA
Result-(48)h
2.Data-04,92,49,AD,EB
Result-(49)h
3.Data-03,43,62,94
Result-(43)h

Largest no.

1.Data -07,68,57,48,52,A7,AE,DA
Result-(DA)h
2.Data-04,92,49,AD,EB
Result-(EB)h
3.Data-03,43,62,94
Result-(94)h

CONCLUSION:-

From the above experiment we have find the smallest no and largest no among the given data array.

VIVA QUESTIONS:-

1. Define CMP, DCR instruction.
2. What is value of flag register after execution of above program?
3. Define instruction & types of instructions used in 8085.

EXPERIMENT-5 (A)

ADDITION OF TWO 16- BIT NUMBER USING 8086 MICROPROCESSOR

AIM OF THE EXPERIMENT:-

To add two 16-bit hexadecimal numbers residing in memory and store the result in memory.

APPARATUS REQUIRED:-

1. Microprocessor Trainer Kit (Omega OEJ 85-A)
2. Power Supply

THEORY:- The ADD instruction requires either the addend or the augend to be in a register unless the source operand is immediate since the addressing modes permitted for the source and destination are register to register, memory to register, register to memory and finally memory of two numbers in memory. Hence, one of the operands is initially moved to AX. Then using the ADD instruction, 16-bit addition is performed.

PROGRAM:-

```
MOV AX, [1100]      ; addend in A
ADD AX, [1102]      ; add
MOV [1200], AX      ; result
HLT
```

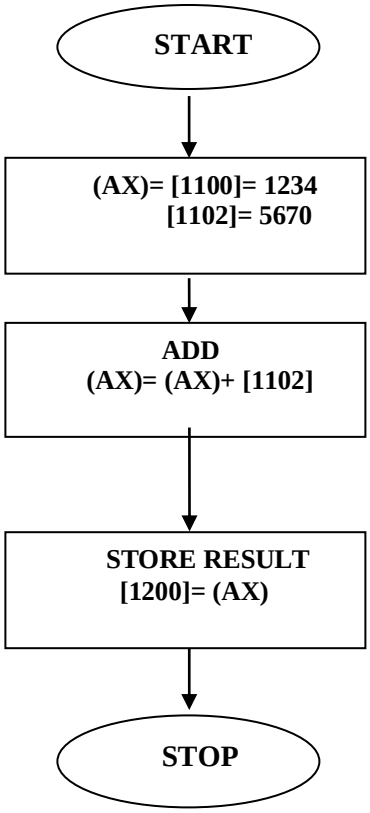
PROCEDURE:-

- Key in the opcodes in to memory.
- Enter the data to be complement at
- Execute the program and check for result.
- Try changing the data and execute the program each time and check the result.

OBJECT CODE:-

Memory Address	Object Codes	Mnemonics	Description
1000	A1	MOV AX, [1100]	Move data stored in memory location 1100 in to AX register.
1001	00		
1002	11		
1003	03	ADD AX, [1102]	ADD the content of register with memory location data [1002].
1004	06		
1005	02		
1006	11		
1007	A3	MOV [1200], AX	Move data from AX register in to memory location in [1200].
1008	00		
1009	12		
100A	F4	HLT	It stops the program.

Flowchart:-



EXERCISE:-

1. Data: (A852)h, (28D9)h
   Result:(D12B)h
2. Data: (D8BC)h, (E45B)h
   Result:(BD17)h
3. Data: (DCB2)h, (1C52)h
   Result:((F904)h

CONCLUSION:-

From the above experiment we have performed the addition of two 8-bit numbers.

VIVA QUESTIONS:-

1. What are differences between 8086 & 8085?
2. Define GPR & SPR in 8086.
3. Explain pin diagram of 8086.

EXPERIMENT: 5(B)

SUBSTRACTION OF TWO 16 BIT NUMBER USING 8086 MICROPROCESSOR

AIM OF THE EXPERIMENT:-

To subtract two words in memory and places the difference in a memory location.

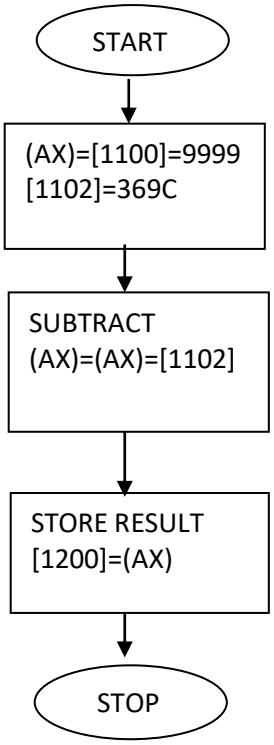
APPARATUS REQUIRED:-

1. Microprocessor trainer kit-OMEGA OEJ 85A
2. Power supply:

THEORY:-

The arithmetic subtraction as discussed in ADD it permits the same modes of addressing. Hence moving the minuend to a register pair is necessary. Then the result is moved to a location in memory.

Flow chart:-



PROGRAM:-

```
MOV   AX,[1100]
SUB    AX,[1102]
MOV   [1200], AX
HLT
```

PROCEDURE:-

1. Key in the epodes from the address specified.
2. Enter data that is needed for execution of 16 bit subtraction.
3. Execution the program and check for result.
4. Try changing data check for result.

OBJECT CODIE:-

MEMORY ADDRESS	OBJECT CODES	MNEMONICS	DESCRIPTION
1000	A1	MOVAX ,[1100]	Move data stored in memory location 1100 in to AX reg.
1001	00		
1102	11		
1003	2B	SUB AX,[1102]	SUB the content of AX reg with memory location data 1102.
1004	06		
1005	02		
1006	11		
1007	A3	MOV [1200],AX	Move data from AX reg in to memory location1200.
1008	00		
1009	12		
100A	F4	HLT	It stops the program.

EXERCISE:-

[1] [1100] = (9999)h
[1102] = (3698)h
[1200] = (3601)h (RESULT)
[2] [1100] = (9BCF)h
[1102] = (8ABD)h
[1200] = (1112)h (RESULT)
[3] [1100] = (7FCB)h
[1102] = (1FAB)h
[1200] =(6020) h(RESULT)

CONCLUSION:-

From the above experiment we have performed 16 bits subtraction operation using 8086 microprocessor.

VIVA QUESTIONS:-

1. Define pipe lining in 8086.
2. What is size of flag register?
3. What do you mean by addressing modes?
4. Define arithmetic instruction of 8086.

EXPERIMENT- 5 (C)

MULTIPLICATION OF 16 BIT NUMBER USING 8086 MICROPROCESSOR

AIM OF THE EXPERIMENT:-

To study & find the multiplication of 16 bit number using 8086 microprocessor.

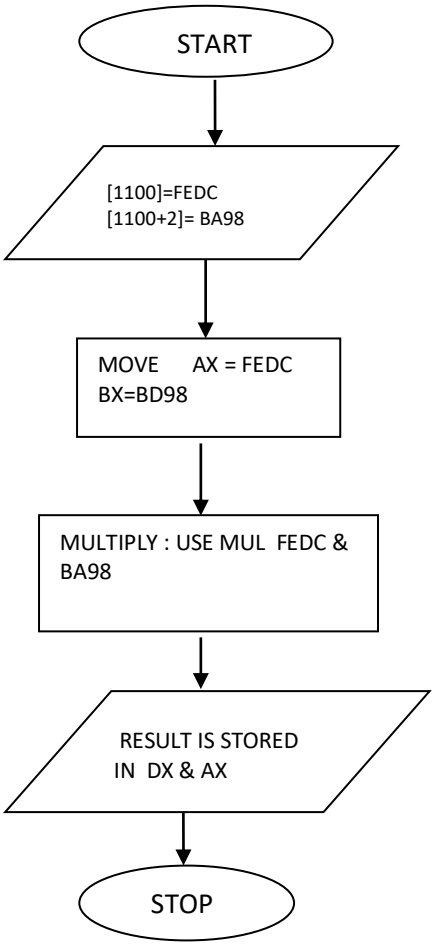
APPARATUS REQUIRED:-

1 Microprocessor trainer kit omega OEJ-86(A)
2 Power supply

THEROY:-

Unlike most of the 8 bit processor which do not have an arithmetic multiply instruction , 16 bit processors from 8086 upward provide both signed & unsigned multiply in their instruction sets to overcome the loss of efficiency in performing repeated addition .
The MUL instruction can have both 16 & 8 bit operands & the multiplicand is AX or AL , accordingly the result for a byte multiply is a 16 bit number in AX while that for a word multiply is a 32 bit number , the lower word of which is in AX & the higher word in DX.

Flow chart
WORD MULTIPICATION



PROGRAM:-

```
MOV AX, 1100
MUL 1102
MOV 1200,DX
MOV 1202,AX
HLT
```

PROCEDURE:-

1. Enter the opcode in specified memory location
2. Enter the data at 1100,1102
3. Set the accumulator with immediate data OOH
4. Now multiplication is done by MUL instruction
5. Now store the data from AX to memory location 1202 if it is a 16 bit data else store the another 16 bit data in 1200 memory location from DX

OBJECT CODE;-

Memory location	Opcode	Mnemonics	Description
1000	A1	MOV AX,1100	MOVE CONTENT OF MEMORY LOCATION 1100 INTO AX
1001	00		
1002	11		
1003	F7	MUL 1102	MULTIPLY CONTAINS OF AX WITH CONTENT STORED IN MEMORY LOCATION 1102
1004	26		
1005	02		
1006	11		
1007	87	MOV 1200,DX	MOVE CONTENT OF DX INTO 1200 MEMORY LOCATION
1008	16		
1009	00		
100A	12		
100B	A3	MOV 1202,AX	MOVE CONTENT OF AX TO 1202 MEMORY LOCATION
100C	02		
100D	12		
100E	F4	HLT	STOP THE PROGRAM

EXERCISE:-

Data:	Data:	Data:
[1100] – (AAAA)h	[1100] –(DCBA)h	[1100] – (45AB)h
[1102] – (3333)h	[1102] –(FFFF)h	[1102] –(123E)h
Result:	Result:	Result:
[1200] – (2221)h	[1200] –(DCB9)h	[1200] –(4F6E)h
[1202]— (DDDE)h	[1202] –(2346)h	[1202] –(056A)h

CONCLUSION:-

From the above experiment we study & find out the multiplication of two 16-bit data.

VIVA QUESTIONS:-

1. Define memory & its types.
2. Explain MUL instruction with example

EXPERIMENT-5(D)

DIVISION OF TWO 16 BIT NUMBER USING 8086 MICROPROCESSOR

AIM OF THE EXPERIMENT:-

Write a program divide two 16 bit memory and to store the result also in memory.

APPARATUS REQUIRED:-

1. Microprocessor tanner kit (omega OEJ-86A)
2. Power supply

THEORY:-

The 16 bit microprocessor from 8086 upward provide both signed and unsigned drive in their instruction sets to overcome the loss of efficiency in performing repeated subtraction. The DIV instruction can have 16 bit operands and the divide is AX register, accordingly. The result of 16 bit number is stored in AX while 16 bits remainder is stored in DX register.

Example:-

Divide the 16 bit number-
FEDC and 2222 at 1100 and 1102
Input: [1100]=(FEDC)h
[1102]=(2222)h
Result: [1200]=(0007)h
Remainder: [1202]=(0FEE)h

PROGRAM:-

```
MOV   AX,[1100]
 DIV    [1102]
 MOV  [1200], DX
 MOV  [1202], AX
 HLT
```

PROCEDURE:-

1. Enter the opcode in the specified memory location.
2. Enter the 16 bit data at 1100, 1102.
3. Move the contents of AX register from memory 1100.
4. Load divisor from memory location 1102.
5. Divide content of accumulator with 16 bit data stored in 1102.
6. Result is stored in accumulator.
7. Store the result in memory location 1200 and remainder in 1202.
8. Stop the program.

OBJECT CODE:-

Memory location	Opcode	Mnemonics	Operand	Description
1000	A1	MOV	AX,[1100]	Move the data stored in memory location 1100 to AX register.
1001	00			
1002	11			
1003	F7	DIV	[1102]	Divide content AX with 1102.
1004	26			
1005	02			
1006	11			
1007	87	MOV	[1200],DX	Move content of DX into 1200 memory location.
1008	16			
1009	00			
100A	12			
100B	A3	MOV	[1202],AX	Move content of AX to 1202 memory location.
100C	02			
100D	12			
100E	F4	HLT		Stops the program.

EXERCISE :-

1. [1100]=(FEDC) H Result [1200]=(0007)H
   [1102]=(2222)H Remainder [1202]=(0FEE)H
2. [1100]=(BFCE)H Result [1200]=(0005)H
   [1102]=(2352)H Remainder [1202]=(OF2C)H
3. [1100]=(DFEA)H Result [1200]=(0004)H
   [1102]=(3333)H Remainder [1202]=(131E)H

CONCLUSION:-

From the above experiment we have performed division operation of 16 bits no using 8086

VIVA QUESTIONS:-

1. Define DIV instruction.
2. How many memory locations are used for storing input & output data?
3. Define Segment.
4. What are differences between minimum & maximum mode.

EXPERIMENT-6

STUDY OF DESCENDING ORDER OF AN UNSORTED ARRAY OF AN 8086 MICROPROCESSOR.

AIM OF THE EXPERIMENT:-

To study the descending order of an unsorted array of an 8086 microprocessor.

APPARATUS REQUIRED:-

1. Microprocessor trainer kit omega OEJ-86(A)
2. Power supply

THEORY :-

In this program 16 bits data array are stored in consecutive memory locations. For arranging data in descending order first data is loaded in accumulator.CMP instruction is used to compare accumulator data with data if next memory location. Counter stores size of data array which says last address of memory location of input data.

PROCEDURE:-

1. Enter the opcode in the specified memory location at 5001, 5003,5005,5007,5009.
2. Move the counter with 04.
3. Set the B resister with immediate data 00h.
4. Add 2 with B resister and compare with acc.
5. Largest data store in acc of the comparison.
6. All the 4-, 16bits number will arrange in descending order.
7. Store the result in memory location 6001 to 6009.
8. Stop the program.

Example

Input data ⇨	04	8232	9005	9614	9050
Memory address ⇨	5001	5003	5005	5007	5009

OBJECT CODE:-

Memory Location	Opcode	Mnemonics	Operands	Description
1000	8E	MOV	SI,0500	Load value of memory location in register SI
1001	00			
1002	05			
1003	BO	MOV	CL,[SI]	Move 8 bit data present in memory location SI to CL
1004	06			
1005	FE	DEC	CL	Decrememt value of counter by 1 present in register CL
1006	01			
1007	8E	MOV	SI,0500	Load value of memory location in register SI
1008	00			
1009	05			
100A	B1	MOV	CH,[SI]	Move 8 bit data present in memory location SI to CH
100B	06			
100C	FE	DEC	CH	Decrememt value of counter by 1 present in register CH
100D	05			
100E	FF	INC	SI	Increment SI by one
100F	A1	MOV	AL,[SI]	Mov data from memory address of SI to AL register
1010	00			
1011	FF	INC	SI	Increment SI by one
1012	05	CMP	AL,[SI]	Compare content of memory SI with data present in AL
1013	10			
1014	73	JNC	101C	Jump from current memory location to 101c location if CY=0
1015	1C			
1016	10			
1017	EB	XCHG	AL,[SI]	Exchange data from SI to AL register
1018	4E	DEC	SI	Decrement value of counter by 1 present in register SI
1019	06			
101A	EB	XCHG	AL,[SI]	Exchange data from SI to AL register
101B	FF	INC	SI	Increment SI by one
101C	FE	DEC	CH	Decrement value of counter by 1 present in register CH
101D	05	JNZ	100F	Jump from current memory location to 100F location if ZF=0
101E	74			
101F	0F			
1020	10	DEC	CL	Decrement value of counter by 1 present in register CL
1021	FE	JNZ	1007	Jump from current memory location to 1007 location if ZF=0
1022	74			
1023	07			
1024	10	HLT		Stop the program

Output ⇨	04	9614	9050	9005	8232
Memory address ⇨	5001	5003	5005	5007	5009

CONCLUSION:-

From the above experiment we have studied and verified descending order of an unsorted array of an 8086 M/P.

VIVA QUESTIONS:-

1. Define XCHG instruction.
2. WAP for ascending order.

EXPERIMENT-7

STUDY OF STEPPER MOTOR AND ITS OPERATION

AIM OF THE EXPERIMENT:-

To study the stepper motor and its operation.

APPARATUS REQUIRED:-

1. Stepper motor trainer kit
2. Connecting wire and power supply.

THEORY:-

Stepper motor is an electric mechanical device, which actuates a train of step angular or liner moment in response train of input pulse one to one basic one step actuation of each pulse input.
Step motor often referred as stepper motor are different from all other types of electrical device in the sense that they operate an discrete steps in the other hand ordinary electrical ac and dc are analog in nature and rotate continuously depending on magnitude and polarity of the control signal received .The discrete nature of operation of a stepper motor makes it suitable for interfacing with a microprocessor and direct microprocessor controller. These motors are widely employed in industrial control. Especially for CNC machines are open loop control in discrete steps are acceptable.
These motors can also adopted for continuous rotation in these lesson we would discuss about construction and principles of operation of different types of step motors and elaborate on the drive schemes used.
Step motors are normally of three types:-

- Permanent magnet stepper motor.
- Variable reluctance stepper.
- Hybrid synchronous stepper.

In a step motor the exaction of voltage to the coils is dc and the number phase indicates the number of winding in both the two cases exaction binding are in the stator .in a permanent magnet with a number of poles on the other hand rotor of a variable relocation type motor is in of a cylindrical structure with a number of project tooth.

Permanent magnet stepper motor:-

Permanent magnets motors yes a permanent magnet in the rotor and operate on the attraction or repulsion between the rotor PM and the stator electromagnets.
Variable reluctance motor have a plain rotor and iron rotor and operator based on they principles of that minimum reluctance occurs with the minimum gape hence the rotor pointy are attracted toward the sector magnet poles.

Hybrid synchronous stepper motor:-

Hybrid stepper motor are named because the use a combination of permanent magnet and variable reluctance to achieve maximum power in a small package size.

Operation of stepper motor:-

Stepper motor operate differently from dc brush motors which rotate he voltage is applied to their terminals stepper motors among the other hand effectively have multiple tooth end electromagnets arranged around a central gear shaped piece of iron the electromagnets are energized by an external control circuit for example microcontroller.

Use of stepper motor:-

1. As the stepper motor are digitally controlled using an input pulse they are suitable for use with computer control system.
2. They are used in numeric control of machines tools.
3. Used in tape drives, floppy disc drives printer and electrical watches.

Advantage of stepper motor:-

1. The percentage of step error does not accumulate as the motor rotates.
2. It is able to run at wide range of speeds including very slow speeds without reducing gear.
3. Stepper motor provide excellent during start stop and reverse mode.
4. It is highly reliable since no brushes or commentator are used .its life time depend on life time of bearing.
5. Stepper motor control circuit is simple and low coast it is mainly used for low power application.

CONCLUSION:-

From the above we have studied operation of stepper motor and its application.

VIVA QUESTIONS:-

1. What are the parts of stepper motor?
2. Explain operation of stepper motor.
3. How motor rotates in clockwise and anti clock wise direction

EXPERIMENT-8

TO STUDY ABOUT THE OPERATION AT 8255 USING 8085 AND 8051 MICROCONTROLLER.

AIM OF THE EXPERIMENT: –

To study about the operation at 8255 using 8085 and 8051 microcontroller.

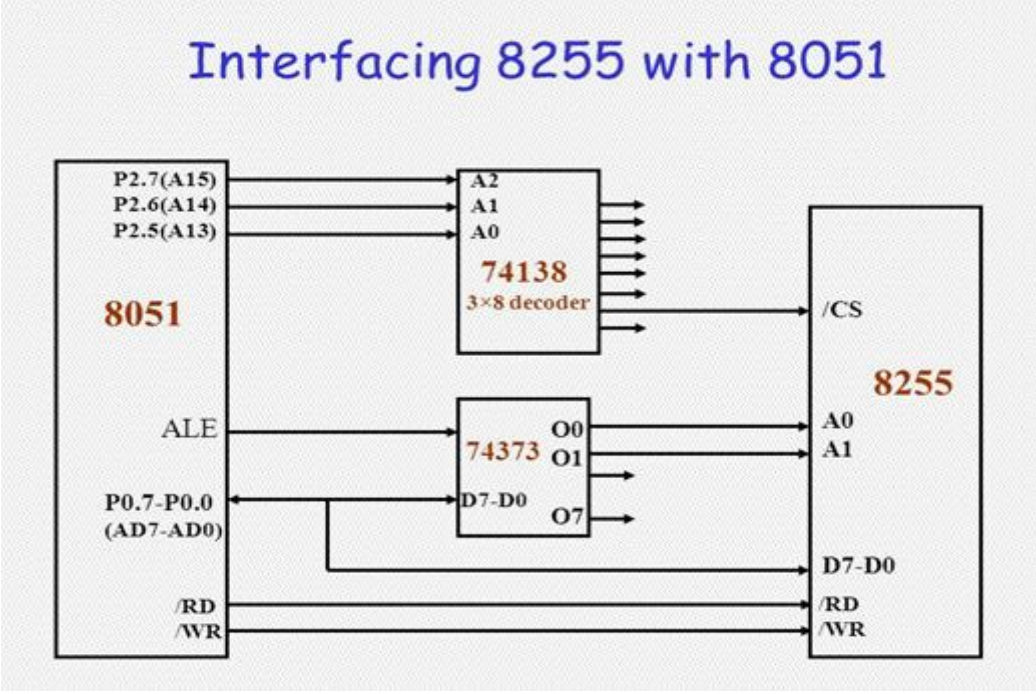
APPARATUS REQUIRED: –

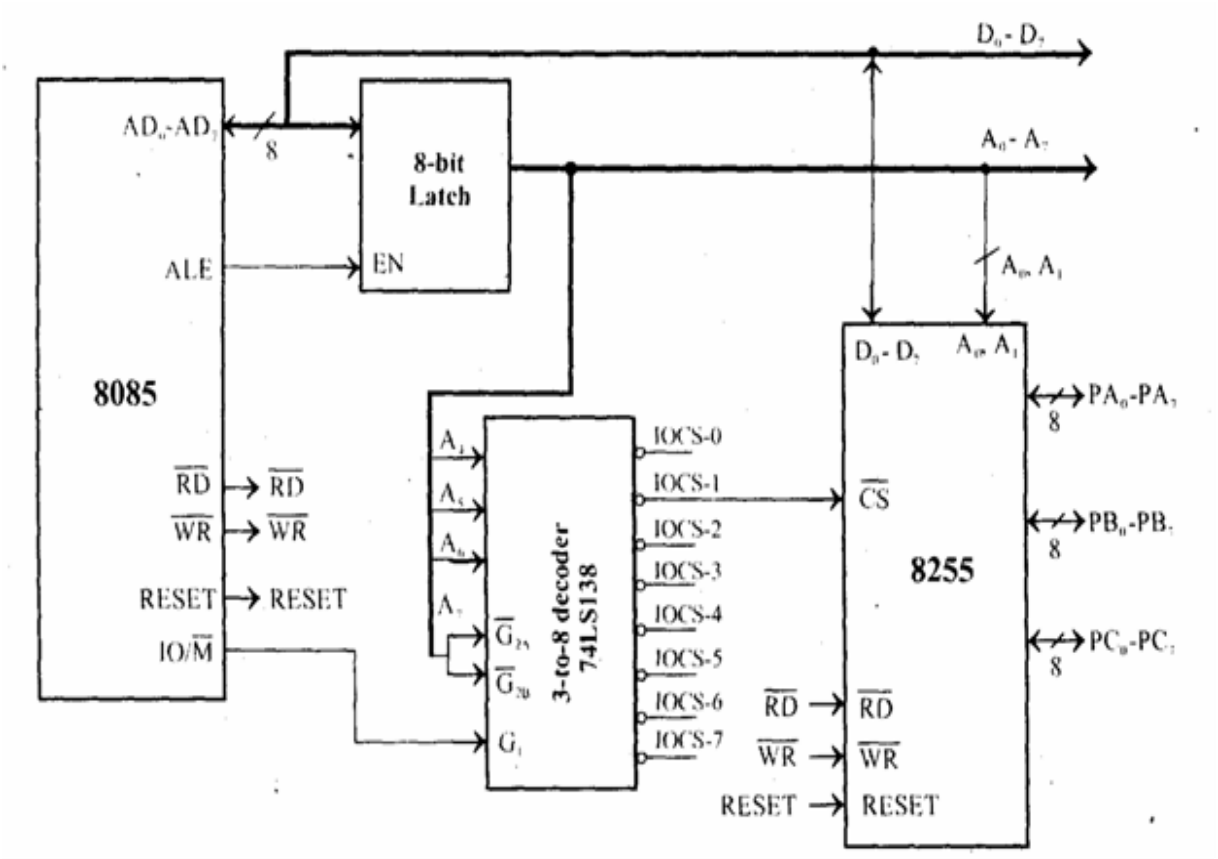
1.8086 microprocessor trainer kit
2. 8255 IC
3. Power Supply, Connecting Wires.

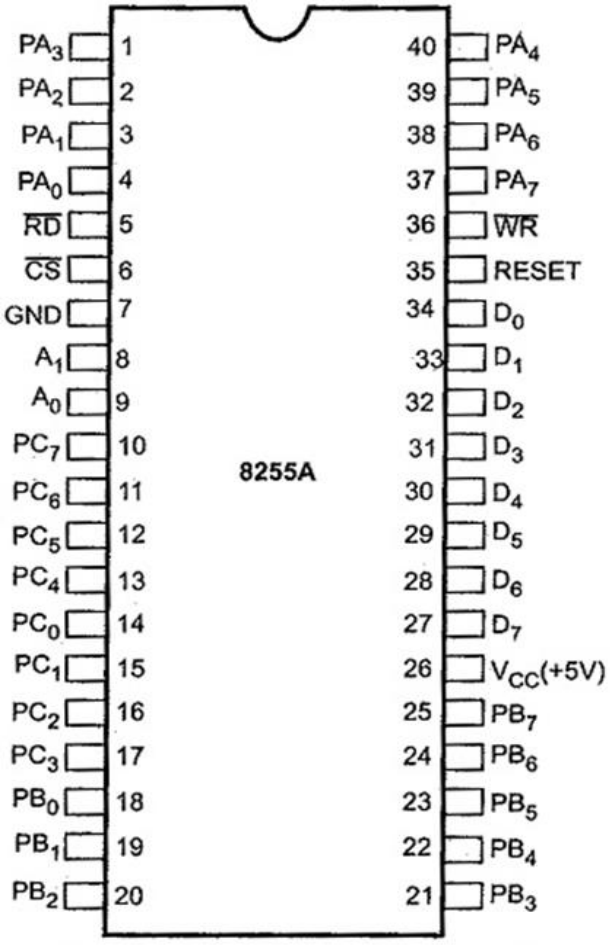
THEORY:-

The parallel input-output port chips of 8255 is also called as programmable peripherals input-output port. The INTEL 8255 is designed for use with INTEL's 8-bit and 16-bit and higher capability microprocessor. It has 24 I/O lines which may be individually programmed in groups of twelve lines each, or three groups of eight lines. The two groups of I/O pins are named as Group A and Group B. Each of these two group contain as Group A and subgroup of four I/O lines or a 4-bit port. Thus Group A contains an 8-bit Port A along with a 4-bit Port C $_{upper}$. The Port A lines are identified by symbols PA_0 – PA_7 while Port C are identified as PC_4 – PC_7. Similarly Port B contains 8-bit port from PB_0- PB_7 and 4-bit Port C with lower bits PC_0 – PC_3.

The Port C $_{upper}$ and the Port C $_{lower}$ can be used in combination as 8-bit Port C. Both the ports are assigned the same address. Thus one may have either three 8-bit I/O ports or two 8-bit and two 4-bit I/O ports from 8255. All of these ports can be independently used either as input or as output port. This can be achieved by programming the bits of an Internal Register of 8255 called as Control-Word-Register (CWR). The 8–bit data bus buffer is controlled by the R/W control logic manages all of the internal and external transfers of both data and control words.







Pin diagram of 8255A

(i). (PA_7 to PA_0) – These are eight ports A lines that acts as either latched output or buffered input lines depending upon the control word loaded into the Control-Word-Register (CWR).

(ii). (PC_7 – PC_4) – Upper nibble of Port C lines. They may act as either output latches or input buffer lines. This port can also be used for generation of handshake lives in mode 1 or mode 2.

(iii). (PC_3 to PC_0) – These are the lower port C lines, other details are same as (PC_7 –PC_4) lines.

(iv). (PB_0 to PB_7) – These are the Port B lines which are used as latched output lines or buffered input lines in the same way as port A.

(v). RD' – This is the input line driven by the microprocessor and should be low to indicate read operation to 8255.
(vi). WR' – this is an input line driven by the microprocessor A low (0) on this line indicates write operation.

(vii). CS' – This is a chip select line. If this line goes low, it enables in 8255 to respond the RD' and WR' signal, otherwise RD' and CS' signals are neglected.

viii). (A_1 – A_0) – These are the address input lines and are driven by the microprocessor. These lines (A1 – A0) with RD, WR and CS from the following operation of 8255. These addressing lines are used for addressing any one out of four register.

(ix). (D_0 – D_7) – These are the data bus that carry data or control word to/from the microprocessor.

(x). RESET – A logic high on this pin cleans the Control Word Register (CWR), of 8255 microprocessor using 8085 and 8051 microcontroller.

CONCLUSION: –

From the above experiment we have studied about 8255 microprocessor using 8085 and 8051 microprocessor.

VIVA QUESTIONS:-

1. What are the features of 8255 IC?
2What are the different registers used in this IC?
3. Explain different pins of this IC.
4. What is use of 8255ic

EXPERIMENT-9

TO STUDY ABOUT THE OPERATION AT 8259 PROGRAMMABLE INTERRUPT CONTROLLER.

AIM OF THE EXPERIMENT: –

To study about the operation at 8259 programmable interrupt controller.

APPARATUS REQUIRED: –

1.8086 microprocessor trainer kit,
2. 8259 IC, Power Supply, Connecting Wires.

THEORY:-

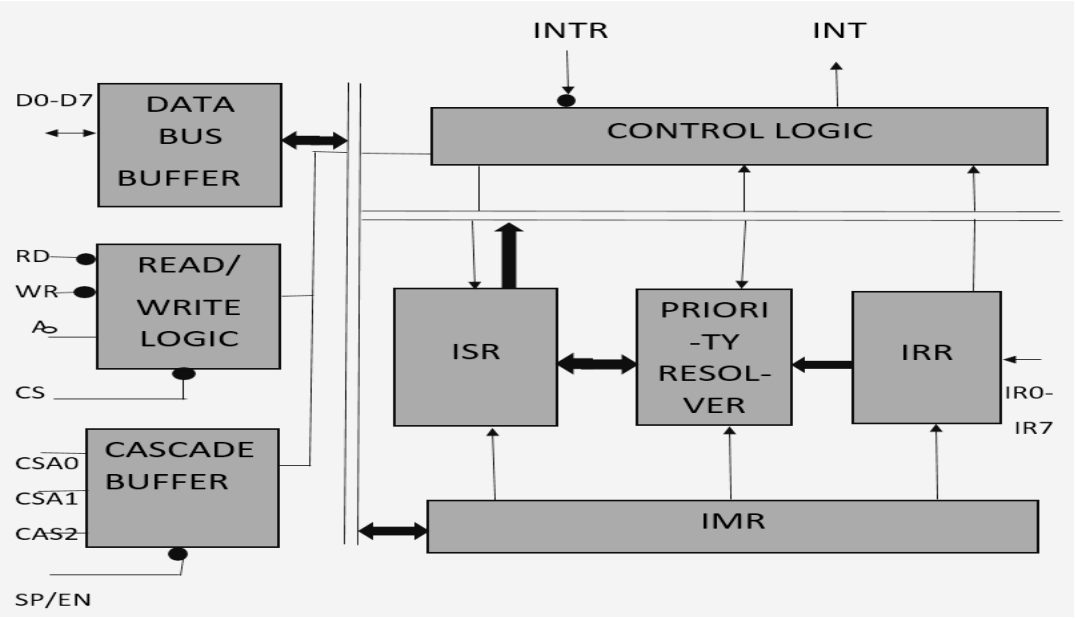
8259 microprocessor is defined as **Programmable Interrupt Controller (PIC)** microprocessor. There are 5 hardware interrupts and 2 hardware interrupts in 8085 and 8086 respectively. But by connecting 8259 with CPU, we can increase the interrupt handling capability. 8259 combines the multi interrupt input sources into a single interrupt output. Interfacing of single PIC provides 8 interrupts inputs from IR0-IR7.

For example, interfacing of 8085 and 8259 increases the interrupt handling capability of 8085 microprocessor from 5 to 8 interrupt levels.

FEATURES OF 8259 PIC MICROPROCESSOR: –

1. Intel 8259 is designed for Intel 8085 and Intel 8086 microprocessor.
2. It can be programmed either in level triggered or in edge triggered interrupt level.
3. We can mask individual bits of interrupt request register.
4. We can increase interrupt handling capability up to 64 interrupt level by cascading further 8259 PIC.
5. Clock cycle is not required.

Block Diagram of 8259 –



The Block Diagram consists of 8 blocks which are – Data Bus Buffer, Read/Write Logic, Cascade Buffer Comparator, Control Logic, Priority Resolver and 3 registers- ISR, IRR, IMR.

1. **Data bus buffer –**
   This Block is used as a mediator between 8259 and 8085/8086 microprocessor by acting as a buffer. It takes the control word from the 8085 (let say) microprocessor and transfer it to the control logic of 8259 microprocessor. Also, after selection of Interrupt by 8259 microprocessor, it transfer the opcode of the selected Interrupt and address of the Interrupt service sub routine to the other connected microprocessor. The data bus buffer consists of 8 bits represented as D0-D7 in the block diagram. Thus, shows that a maximum of 8 bits data can be transferred at a time.

2. **Read/Write logic** –
   This block works only when the value of pin CS is low (as this pin is active low). This block is responsible for the flow of data depending upon the inputs of RD and WR. These two pins are active low pins used for read and write operations.

3. **Control logic –**
   It is the centre of the microprocessor and controls the functioning of every block. It has pin INTR which is connected with other microprocessor for taking interrupt request and pin INT for giving the output. If 8259 is enabled, and the other microprocessor Interrupt flag is high then this causes the value of the output INT pin high and in this way 8259 responds to the request made by other microprocessor.

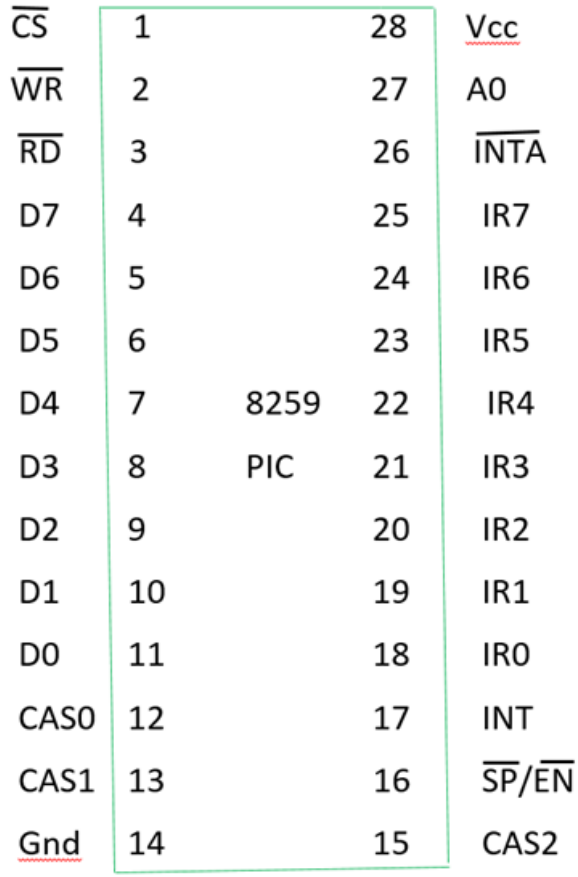
4. **Interrupt request register (IRR) –**
   It stores all the interrupt level which are requesting for Interrupt services.

5. Interrupt service register (ISR) –
   It stores the interrupt level which are currently being executed.

6. **Interrupt mask register (IMR) –**
   It stores the interrupt level which have to be masked by storing the masking bits of the interrupt level.

7. **Priority resolver –**
   It examines all the three registers and set the priority of interrupts and according to the priority of the interrupts, interrupt with highest priority is set in ISR register. Also, it reset the interrupt level which is already been serviced in IRR.

8. **Cascade buffer –**
   To increase the Interrupt handling capability, we can further cascade more number of pins by using cascade buffer. So, during increment of interrupt capability, CSA lines are used to control multiple interrupt structure.



We can see through above diagram that there are total 28 pins in 8259 PIC microprocessor where Vcc :5V Power supply and Gnd: ground. SP/EN (Slave program/Enable buffer) pin is when set to high, works in master mode else in slave mode. In Non Buffered mode, SP/EN pin is used to specify whether 8259 work as master or slave and in Buffered mode, SP/EN pin is used as an output to enable data bus.

CONCLUSION: –

From the above experiment we have studied about 8255 microprocessor using 8085 and 8051 microprocessor.

VIVA QUESTIONS:-

1. What are the features of 8259 IC?
2. What are the different registers used in this IC?
3. Explain different pins of this IC.

EXPERIMENT -10

STUDY OF 8279 (KEYBOARD & DISPLAY INTERFACE)

AIM OF THE EXPERIMENT:-

Study of 8279 (Keyboard & Display interface)

APPARATUS REQUIRED:-

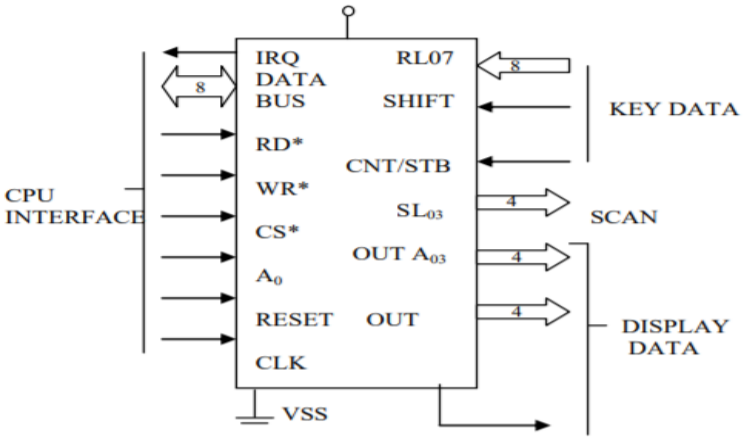
1. IC 8279 & MICROPROCESSOR IC
2. Seven segment display
3. Keyboard
4. Connecting wire & power supply

THEORY:-

The Intel 8279 is a general-purpose programmable keyboard and display I/O interface device designed for use with Intel* microprocessors. The keyboard portion can provide a scanned interface to a 64-contact key matrix.

The keyboard portion will also interface to an array of sensors interface keyboard, such as the Hall Effect and ferrite variety. Key depressions can be 2-key lockout or N-key rollover. Keyboard entries are denounced and strobe in an 8-character FIFO. If more than 8 characters are entered, overrun status is set.

Key entries set the interrupt output line to the CPU. The display portion provides a scanned display interface for LED, incandescent, and other popular display technologies. Both numeric and alphanumeric segment displays may be used as well as simple indicators. The 8279 have 16x8 displays RAM, which can be organized into dual 16x4. The RAM can be loaded or interrogated by the CPU. Right entry, calculator and left entry typewriter display formats are possible. Both read and write of the display RAM can be done with auto-increment of the display RAM address.



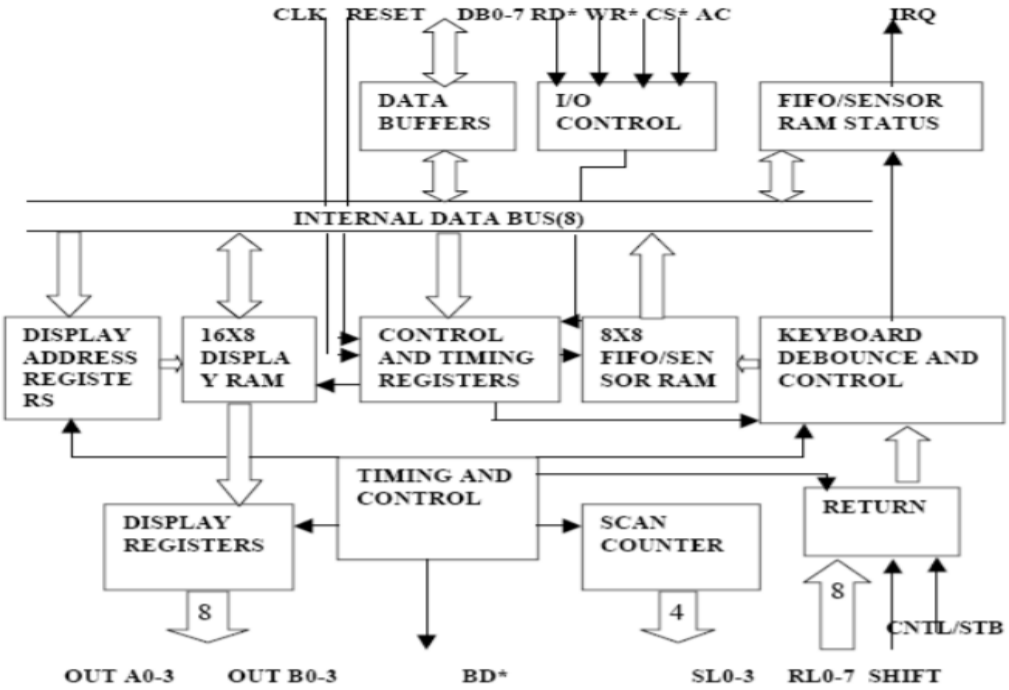
FUNCTIONAL DESCRIPTION:

Since data input and display are an integral part of many microprocessor designs, the system designer needs an interface that can control these functions without placing a large load on the CPU. The 8279 provide this function for 8-bit microprocessors. The 8279 have two sections: keyboard and display. The keyboard section can interface to regular typewriter style keyboards or random toggle or

thumb switches. The display section drives alphanumeric displays or a bank of indicator lights. Thus the CPU is relieved from scanning the keyboard or refreshing the display.
The 8279 is designed to directly connect to the microprocessor bus. The CPU can program all operating modes for the 8279.

PRINCIPLES OF OPERATION:

The following is a description of the major elements of the 8279 Programmable keyboard/Display interface device. Refer to the block diagram in Figure.



INTERNAL BLOCK DIAGRAM OF 8279

RD* with CS* low and A high. The status logic also provides an IRQ signal when the FIFO is not empty. In Scanned Sensor Matrix mode, the memory is a Sensor RAM. Each row of the Sensor RAM is loaded with the status of the corresponding row of sensor in the sensor matrix. In this mode, IRQ is high if a change in a sensor is detected.

Display Address Registers and Display RAM:

The Display Address Registers hold the address of the word currently being displayed. The read/write address is programmed by CPU command. They also can be set to auto increment after each read or write. The Display RAM can be directly read by the CPU after the correct mode and address is set. The addresses for the A and B nibbles are automatically updated by the 8279 to match data entry by the CPU. The A and B nibbles can be entered independently or as one word according to the mode that is set by the CPU Data entry to the display can be set to either left or right entry see Interface Considerations for details.

CONCLUSION:-

From the above experiment we have studied 8279 programmable keyboard/display interface device.

VIVA QUESTIONS:-

1. What are the features of 8279 IC?
2. What are the different registers used in this IC?
3. Explain different pins of this IC.

EXPERIMENT-11

INITIALIZE DATA TO REGISTER AND MEMORY USING IMMEDIATE, REGISTER, DIRECT, INDIRECT ADDRESSING MODES.

AIM OF THE EXPERIMENT: –

Initialize data to register and memory using immediate, register, direct, indirect addressing modes..

APPARATUS REQUIRED: –

1.8051 microcontroller trainer kit
2. Keyboard
3. Power Supply, Connecting Wires.

THEORY:-

ADDRESSING MODES OF 8051

In 8051 there are six types of addressing modes.

- Immediate Addressing Mode.
- Register Addressing Mode.
- Direct Addressing Mode.
- Register Indirect Addressing Mode.
- Indexed Addressing Mode.
- Implied Addressing Mode.

Immediate addressing mode:

In this addressing mode, the data is directly specified in the instruction itself. The source is the immediate data and the destination of this instruction could be any register. "#" symbol specifies that the operand is an immediate data on which operation is to be performed and the result will be stored in the destination register.
Ex: **MOV A, #65H** ; The digit 65H is copied to the Accumulator (A). MOV DPTR, #1615H ; A 16 bit digit 1615H is copied to Data pointer which is 16 bits.

Register addressing mode:

In register addressing mode, the source and the destination both are registers, and must be of same size as indifference is size will give errors. The data is specified in the registers for various operations as per the given instructions. Only the Accumulator (A) and R0 to R7 registers of each memory bank are allowed in this mode to transfer the data. The data transfer can take place between Rn(R0 to R7) registers and the Accumulator (A) only and cannot be done between Rn registers.
Ex: **MOV A, R3** ; The contents of register R3 are copied to Accumulator (A). ADD A, R4 ; The contents of register R4 are added with the contents of the Accumulator (A) and the result is stored in the Accumulator (A). MOV DPTR, A ; This instruction will end up giving error as there is size indifference i.e. DPTR = 16 bits and A= 8 bits.

Direct addressing mode:

In this addressing mode, source and destination could be a register or a RAM location, but both cannot be the same; either the source has to be a register followed by RAM location as destination and vice versa. The address of the operand is specified in the instruction itself. Internal RAM addresses starting from location 00H to 7FH and SFR addresses starting from 80H to FFH are only allowed in direct addressing mode.
Ex**: MOV R3, 70H;** The contents of RAM location 70H are copied to the register R3. MOV 32H, A ; The contents of Accumulator (A) are copied to RAM location 32H.

REGISTER indirect addressing mode:

In this mode, the address of the operand is specified in the in a register. Only the registers R0 and R1 are data pointers i.e. the data is stored on the RAM location whose address is held by either R0 or R1. It can only use addresses from 00H to 7FH. @ sign is used in the instruction and is placed before the registers R0 and R1 to make the two registers as pointers.
Ex: **MOV A, @R0** ; The contents of the memory location held by [R0] is copied to Accumulator (A). **MOV @R0, A** ; The contents of Accumulator (A) are copied to the memory location held by [R0]. **MOV** X **A, @DPTR** ; The contents of the external RAM location held by [DPTR] (16-bit) are copied to the Accumulator (A). MOV **X** A, @R0 ; The contents of external RAM location held by register [R0] (8-bit) are copied to the Accumulator (A).

Indexed addressing mode:

In this addressing mode, address is indirectly specified in Accumulator (A), data pointer (DPTR) and program counter. It is usually the sum of the addresses stored at [A+DPTR] or [A+PC].
This addressing mode is very useful because ROM contains permanent data which is stored in the form of Look-Up tables. To access the Look-Up, the addresses are given as SUM of contents of two registers, where one acts as the base and other acts as the index within the table. Also this mode is used to access data from the code memory and is denoted by **"C"** in the instruction.

Ex**: MOV**C **A, @A+DPTR** ; Contents of the ROM location pointed by the sum of addresses A and DPTR (16-bits) are copied to Accumulator A. In this case the data pointer holds the base address and the accumulator (8-bits) holds the index or displacement.

CONCLUSION:-

From the above experiment we have studied addressing modes of 8051 microcontroller.

VIVA QUESTIONS:-

1. What are the features of 8051 IC?
2. What are the different registers used in this IC?
3. Explain different pins of this IC.4
4. What are difference between microcontroller & microprocessor?
5. Define flag register of 8051 IC