Exp No.1: Programs for 16 bit arithmetic operations for 8086 (usingVarious Addressing Modes).

a) Addition:

i)16 bit addition:

AIM: - To write an assembly language program for Addition of two 16-bit numbers.

APPARATUS:	1.8086 microprocessor kit/MASM	----1
	2.RPS (+5V)	----1

PROGRAM:

i) By using MASM:

```
                Assume cs: code

                Code segment

        Start:  MOV AX, 4343

                MOV BX, 1111

                ADD AX, BX

                INT 3

                Code ends

                End start
```

ii) By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV AX,4343
			MOV BX,1111
			ADD AX,BX
			INT 3

OUTPUT:

Input		**output**	
Register	**Data**	**Register**	**Data**
AX	4343	**AX**	**5454**
BX	1111		

ii) Multi byte addition

AIM: - Program to perform multi byte addition

APPARATUS: 1. 8086 microprocessor kit/MASM ----1

2. RPS (+5V) ----1

PROGRAM:

j) By using MASM:

```
          Assume cs: code
          Code segment
Start:    MOV AX, 0000
          MOV SI, 2000
          MOV DI, 3000
          MOV BX, 2008
```

```
                 MOV CL, 04
        UP :     MOV AL, [SI]
                 ADD AL,[BX]
                 MOV [DI], AL
                 INC SI
                 INC BX
                 INC DI
                 DEC CL
                 JNZ UP
                  INT 3
        CODE   ENDS
               END START
```

ii) By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV AX,0000
			MOV SI, 2000
			MOV DI, 3000
			MOV BX, 2008
			MOV CL, 04
		UP	MOV AL, [SI]
			ADD AL, [BX]
			MOV [DI], AL
			INC SI
			INC BX
			INC DI
			DEC CL
			JNZ UP
			INT 3

OUTPUT:

Input				output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**	**MEMORY LOCATION**	**Data**
2000	01	2008	23	3000	24
2001	02	2009	27	3001	29
2002	07	200A	10	3002	17
2003	08	200B	14	3003	1C
2004	X				
2005	X				
2006	X				
2007	X				

b) Subtraction:

i) 16 bit subtraction:

AIM: - To write an assembly language program for subtraction of two 16-bit numbers.

APPARATUS: 1. 8086 microprocessor kit/MASM ----1 2.
RPS (+5V) ----1

PROGRAM:

k) By using MASM:

```
                Assume cs: code

                Code segment

      Start:    MOV AX, 4343

                MOV BX, 1111

                SUB AX, BX

                INT 3

                Code ends

                End start
```

iii) By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION
4000			MOV AX,4343 MOV BX,1111 SUB AX,BX INT 3

OUTPUT:

Input		**output**	
Register	**Data**	**Register**	**Data**
AX	4343	AX	3232
BX	1111		

ii) Multi byte subtraction

AIM: - Program to perform multi byte subtraction.

APPARATUS: 1. 8086 microprocessor kit/MASM ----1

2. RPS (+5V) ----1

PROGRAM:

1) **By using MASM:**

```
                Assume cs: code
                Code segment
Start:          MOV AX, 0000
                MOV SI, 2000
                MOV DI, 3000
                MOV BX, 2008
                MOV CL, 04
UP :            MOV AL, [SI]
                SUB AL, [BX]
                MOV [DI], AL
                INC SI
                INC BX
                INC DI
                DEC CL
                JNZ UP
                 INT 3
        CODE   ENDS
               END START
```

2) By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV AX,0000
			MOV SI, 2000
			MOV DI, 3000
			MOV BX, 2008
			MOV CL, 04
		UP	MOV AL, [SI]
			SUB AL, [BX]
			MOV [DI], AL
			INC SI
			INC BX
			INC DI
			DEC CL
			JNZ UP
			INT 3

OUTPUT:

Input				output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**	**MEMORY LOCATION**	**Data**
2000	23	2008	02	3000	21
2001	27	2009	04	3001	23
2002	44	200A	01	3002	43
2003	43	200B	03	3003	40
2004	X				
2005	X				
2006	X				
2007	X				

c) Multiplication:

i) 16 bit multiplication:

AIM: - To write an assembly language program for multiplication of two 16-bit numbers.

APPARATUS: 1.8086 microprocessor kit/MASM ----1

2.RPS (+5V) ----1

PROGRAM:

A) By using MASM:

```
                Assume cs: code

                Code segment

      Start:    MOV AX, 4343

                MOV BX, 1111

                MUL BX

                INT 3

                Code ends

                End start
```

B) By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV AX,4343 MOV BX,1111 MUL BX INT 3

OUTPUT:

Input		Output	
Register	**Data**	**Register**	**Data**
AX	4343	AX	EA73
BX	1111	DX	047B

ii) 16 bit multiplication (signed numbers)

AIM: - To write an assembly language program for multiplication of two 16-bit signed numbers.

APPARATUS:

1.8086 microprocessor kit/MASM	----1
2.RPS(+5V)	----1

PROGRAM:

A) By using MASM:

```
                Assume cs: code

                Code segment

Start:          MOV SI, 2000

                MOV DI, 3000

                MOV AX, [SI]

                ADD SI, 02

                MOV BX, [SI]
```

```
IMUL BX

MOV [DI], AX

ADD DI, 02

MOV [DI], DX

INT 3

Code ends

End start
```

B) By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV SI,2000
			MOV DI,3000
			MOV AX,[SI]
			ADD SI,02
			MOV BX,[SI]
			IMUL BX
			MOV [DI],AX
			ADD DI,02
			MOV [DI],DX
			INT 3

OUTPUT:

Input		**Output**	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000	E4(-28)	3000	8C
2001	E4(-28)	3001	4C
2002	3B(+59)	3002	F5
2003	3B(+59)	3003	34

d) Division:

I) 16 bit division:

AIM: - To write an assembly language program for multiplication of two 16-bit numbers.

APPARATUS: 1.8086 microprocessor kit/MASM ----1

2.RPS (+5V) ----1

PROGRAM:

A) By using MASM:

```
                Assume cs: code

                Code segment

     Start:     MOV AX,4343
```

```
MOV BX,1111

MUL BX

INT 3

Code ends

End start
```

B) By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV AX,4343 MOV BX,1111 MUL BX INT 3

OUTPUT:

Input		output	
Register	**Data**	**Register**	**Data**
AX	4343	AX	EA73
BX	1111	DX	047B

RESULT: 16 bit arithmetical operations are performed by using different addressing modes.

Viva:

1) How many bit 8086 microprocessor is?

2) What is the size of data bus of 8086?

3) What is the size of address bus of 8086?

4) What is the max memory addressing capacity of 8086?

5) Which are the basic parts of 8086?

EXERCISE:

1. Write an alp program for addition and subtraction of two 16bit numbers?
   1) A278
   2) B634
2. Write an alp program for multiplication and division of two 16bit numbers?
   1) 0012
   2) 0006

EXP NO.2: Program for sorting an array for 8086.

i) ASCENDING ORDER

AIM:-Program to sort the given numbers in ascending order

APPARATUS: 1. 8086 microprocessor kit/MASM ----1

2. RPS (+5V) ----1

PROGRAM:

A) By using MASM:

```
                ASSUME CS: CODE
                CODE SEGMENT
    START:      MOV AX, 0000H
                MOV CH, 0004H
                DEC CH
    UP1 :        MOV CL, CH
                MOV SI, 2000
    UP:         MOV AL, [SI]
                INC SI
                CMP AL, [SI]
                JC DOWN
                XCHG AL, [SI]
                DEC SI
                MOV [SI], AL
                INC SI
    DOWN:       DEC CL
                JNZ UP

                DEC CH

                JNZ UP1
                INT 3
```

```
CODE   ENDS

            END START
```

B) By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV AX, 0000H
			MOV CH, 0004H
			DEC CH
		UP1:	MOV CL, CH
			MOV SI,2000
		UP:	MOV AL,[SI]
			INC SI
			CMP AL,[SI]
			JC DOWN
			XCHG AL,[SI]
			DEC SI
			MOV [SI],AL
			INC SI
		DOWN:	DEC CL
			JNZ UP
			DEC CH
			JNZ UP1
			INT 3

OUTPUT:

Input		**Output**	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000	03	2000	03
2001	06	2001	04
2002	07	2002	06
2003	04	2003	07

ii) DESCENDING ORDER

AIM:-Program to sort the given numbers in descending order

APPARATUS:	1. 8086 microprocessor kit/MASM	----1
	2. RPS (+5V)	----1

PROGRAM:

A) By using MASM:

```
                ASSUME CS: CODE
                CODE SEGMENT
    START:      MOV AX, 0000H
                MOV CH, 0004H
                DEC CH
    UP1 :        MOV CL, CH
                MOV SI, 2000
    UP:         MOV AL, [SI]
```

```
                INC SI
                CMP AL, [SI]
                JNC DOWN
                XCHG AL, [SI]
                DEC SI
                MOV [SI], AL
                INC SI
        DOWN:   DEC CL
                JNZ UP

                DEC CH

                JNZ UP1

                INT 3

CODE   ENDS

          END START
```

B) By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV AX, 0000H
			MOV CH, 0004H
			DEC CH
		UP1:	MOV CL, CH
			MOV SI,2000
		UP:	MOV AL,[SI]
			INC SI
			CMP AL,[SI]
			JNC DOWN

		DOWN:	XCHG AL,[SI] DEC SI MOV [SI],AL INC SI DEC CL JNZ UP DEC CH JNZ UP1 INT 3

OUTPUT:

Input		**Output**	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000	03	2000	07
2001	06	2001	06
2002	07	2002	04
2003	04	2003	03

RESULT: Program for sorting an array performed by using masm software and trainer kit.

1) What are the functions of EU?

2) How many pin IC 8086 is?

3) What IC8086 is?

4) What is the size of instruction queue in 8086?

EXERCISE:

1. Write an alp program to sort the given numbers in ascending order?

    1) 14

    2) A2

    3) 85

    4) 54

2. Write an alp program for to sort the given number in descending order?

    1) 1E

    2) 2A

    3) 56

    4) 98

EXP NO:3 Program for searching for a number or character in a string for 8086.

AIM: Write an alp program for to search a number or character from a string.

APPARATUS: 1. 8086 microprocessor kit/MASM ----1

2. RPS (+5V) ----1

PROGRAM:

A) By using MASM:

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT

LIST DW 53H, 15H, 19H, 02H
DEST EQU 3000H

COUNT EQU 05H
DATA ENDS



START:       MOV AX, DATA

             MOV DS, AX

             MOV AX, 15H

             MOV SI, OFFSET LIST

             MOV DI, DEST

             MOV CX, COUNT

             MOV AX, 00
```

```
                CLD

                 REP

                SCASW

                JZ LOOP

                MOV AX, 01

LOOP            MOV [DI], AX

                MOV AH, 4CH

                INT 3H

CODE ENDS

END START
```

B) By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV AX, 15H
			MOV SI, 2000
			MOV DI, 3000
			MOV CX, 0004
			MOV AX, 00
			CLD
			REP
			SCASW
			JZ LOOP
			MOV AX, 01

		LOOP:	MOV [DI], AX MOV AH, 4CH INT 3H

OUTPUT:

Input		**output**	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000	53	3000	01
2001	15		
2002	19		
2003	02		

RESULT: Program for search a number or character from a string performed by using masm software and trainer kit.

Viva:

1) What is the size of instruction queue in 8086?

2) Which are the registers present in 8086?

3) What do you mean by pipelining in 8086?

4) How many 16 bit registers are available in 8086?

5) Specify addressing modes for any instruction?

EXERCISE:

1. Write an alp program to search a number 05 from a given array?

    1) 02

    2) 06

    3) 05

    4) 08

2. Write an alp program to search a number 45 from a given array?

    1) 09

    2) 45

    3) 22

    4) A2

EXP NO.4: Program for string manipulations for 8086.

1) Moving Block Of Data From One Memory Location To Another Memory Location

AIM: To write an alp for transfer block of data from one memory location to another memory location.

APPARATUS:	1. 8086 microprocessor kit/MASM	----1
	2. RPS (+5V)	----1

PROGRAM:

A) By using MASM:

```
            ASSUME CS: CODE

            CODE SEGMENT

    START:          MOV SI, 2000

                    MOV DI, 2008

                    MOV CX, 0008

                    REP

                    MOVSB

                    INT 03

            CODE ENDS

            END START
```

B) By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV SI,2000 MOV DI,2008 MOV CX,0008 REP MOVSB INT 3

OUTPUT:

Input		output	
MEMORY LOCATION	Data	MEMORY LOCATION	Data
2000	01	2008	01
2001	04	2009	04
2002	03	200A	03
2003	02	200B	02
2004	01	200C	01
2005	04	200D	04
2006	03	200E	03
2007	02	200F	02

2) Reverse of a data:

AIM: To write a alp for reverse of a given string

APPARATUS: 1. 8086 microprocessor kit/MASM ----1

2. RPS (+5V) ----1

PROGRAM:

A) By using MASM:

```
            ASSUME CS: CODE



START:          MOV SI, 2000

                MOV DI, 2008

                MOV CX, 0008

                ADD SI, 07

UP:             MOV AL, [SI]

                MOV [DI], AL

                DEC SI

                INC DI



                JNZ UP

                INT 3
```

CODE ENDS

END START

B) By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV SI,2000
			MOV DI,2008
			MOV CX,0008
		UP	ADD SI,07
			MOV AL,[SI]
			MOV [DI],AL
			DEC SI
			INC DI
			DEC CX
			JNZ UP
			INT 3

OUTPUT:

Input		output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000	01	2008	08
2001	02	2009	07
2002	03	200A	06
2003	04	200B	05
2004	05	200C	04
2005	06	200D	03
2006	07	200E	02
2007	08	200F	01

3) INSERT A BYTE IN A GIVEN STRING

AIM: Write an alp for insert a new byte in a given string.

APPARATUS: 1. 8086 microprocessor kit/MASM ----1

2. RPS (+5V) ----1

PROGRAM:

A) By using MASM:

```
        ASSUME CS: CODE

        CODE SEGMENT

START:      MOV SI, 2000

            MOV DI, 3000


            MOV BX, 5000


            MOV CX, 0005


            CLD


        L1: MOV AL, [SI]


            CMP AL, [BX]


            JZ L2


            MOVSB


            JMP L3


        L2: MOVSB


            MOV BX, 7000


            MOV AL, [BX]
```

```
          MOV [DI], AL

          DEC CX

          INC DI

          REP MOVSB

L3:       INT 3

CODE ENDS

END START
```

OUTPUT:

Input		output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000	02	3000	02
2001	04	3001	04
2002	43	3002	43
2003	76	3003	08
2004	01	3004	76
5000	43	3005	01
7000	08		

By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV SI,2000
			MOV DI,3000
			MOV BX,5000
			MOV CX,0005
			CLD
		L1	MOV AL,[SI]
			CMP AL,[BX]
			JZ L2
			MOVSB
			JMP L3
		L2	MOVSB
			MOV BX,7000
			MOV AL,[BX]
			MOV [DI],AL
			DEC CX
			INC DI
			REP
			MOVSB
		L3	INT 3

OUTPUT:

Input		output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000	02	3000	02
2001	04	3001	04
2002	43	3002	43
2003	76	3003	08
2004	01	3004	76
5000	43	3005	01
7000	08		

4)DELETE A BYTE IN A GIVEN STRING

AIM: To write a alp for delete a byte in a given string

APPARATUS: 1. 8086 microprocessor kit/MASM with pc ----1

2. RPS (+5V) ---1

PROGRAM:

B)By using MASM:

```
ASSUME CS: CODE
```

```
        CODE SEGMENT
START:      MOV SI, 2000
            MOV DI, 3000
            MOV BX, 5000
            MOV CX, 0005
            CLD
        L1: MOV AL, [SI]
            CMP AL, [BX]
            JZ L2
            MOVSB
            LOOP L1
            JMP L3
        L2: INC SI
            DEC CX
            REP MOVSB
        L3: INT 3
        CODE ENDS
        END START
```

C) By using 8086 kit:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV SI,2000
			MOV DI,3000
			MOV BX,5000
			MOV CX,0005
			CLD
		L1	MOV AL,[SI]
			CMP AL,[BX]
			JZ L2
			MOVSB
			LOOP L1
			JMP L3
		L2	INC SI
			DEC CX
			REP
			MOVSB
		L3	INT 3

OUTPUT:

Input		output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000	01	3000	01
2001	02	3001	02
2002	03	3002	03
2003	04	3003	05
2004	05		
5000	04		

RESULT: Program for string manipulation performed by using masm software and trainer kit.

Viva:

1) What do you mean by assembler directives?

2) What .model small stands for?

3) What is the supply requirement of 8086?

4) What is the relation between 8086 processor frequency & crystal Frequency?

5) Functions of Accumulator or AX register?

EXERCISE:

1. Write an alp for insert or delete a byte in a given string with SI memory location is 4000 and DI location is 6000?
2. Write an alp for moving or reversing the given string with the length of the string is 12?

EXP.NO.5: Program for digital clock design using 8086.

AIM: To write an ALP program for displaying the system clock.

APPARATUS: 1.MASM

2. PC

PROGRAM:

```
                ASSUME CS: CODE

                CODE SEGMENT

EXTERN GET_TIME: NEAR

.MODEL      SMALL

.STACK      100H

.DATA

TIME_BUF    DB      “00:00:00$“

CODE

MAIN PROC

                                MOV AX,@DATA

                                MOV DS, AX

                                LEA BX, TIME_BUF

                                CALL GET_TIME
```

```
                    LEA DX, TIME_BUF

                    MOV AH, 09H

                    INT 21H

                    MOV AH, 4CH

                    INT 21H

   MAIN ENDP

   END   MAIN
```

RESULT: Program for displaying the system clock performed using masm software.

EXP.NO.6: Interfacing ADC and DAC to 8086.

AIM:

1. To write a program for conversion of analog data to digital output.

2. To write a program for conversion of digital data to analog output. The analog output will be in the form of triangular wave, saw tooth wave, square wave/rectangular wave.

1. 8086 Trainer.

2. Power supply for trainer and interface module.

3. A/D, D/A interface module.

4. Power mate connector.

5. FRC connector.

6. Cathode ray oscilloscope.

PROCEDURE:-

1. Connect the 26 core FRC connector to the 8086 trainer at connector no CN4 and the interface module.

2. Connect the power mate connector to the interface module and the other side of the connector to the power supply. The connections to the power supply are given below.

   Connections: (power supply)

3. 5- Way power mate is wired to the motor. This power mate is to be inserted into the male socket provided on the interface. Care should be taken such that, below given code for the

particular colored wire coincides with the code on the interface.

4. After the completion of the program and connections enter the program as given in the listing below.

   G0< STARTING ADDRESS< ENTER (on the key board of trainer).

PROGRAM TO GENERATE SQUARE WAVE:

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC
			MOV AL,80
			MOV DX,0FFC6
			OUT DX
			MOV DX,0FFC2
		A0	MOV AL,00
			OUT DX
			CALL DELAY 1
			MOV AL,0FF
			OUT DX
			CALL DELAY2
			JMP A0

DELAY PROGRAM1

ADDRESS	OPCODE	LABEL	MNEMONIC
			MOV CX,0020
		A1	LOOP A1
			RET

DELAY PROGRAM2

ADDRESS	OPCODE	LABEL	MNEMONIC
			MOV CX,0020
		A2	
			LOOP A2
			RET

A/D CONVERTER

PROGRAM:

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC
			MOV AL,90
			MOV DX,0FFC6
			OUT DX
			MOV AL,07
			MOV DX,FFC4
			OUT DX
			MOV AL,0F
			MOV DX,0FFC6
			OUT DX
			MOV CX,3FFF
		D1	LOOP **D1**
			MOV AL,0E
			MOV DX,0FFC6
			OUT DX
			MOV AL,0C
			MOV DX,0FFC6
			OUT DX
			MOV DX,0FFC0
			IN DX
			AND AL,80
			CMP AL,80
			JNZ **D2**
			MOV AL,0D
		D2	MOV DX,0FFC6
			OUT DX
			MOV DX,0FFC0
			IN DX
			MOV DX,0FFC2
			OUT DX
			JMP **D3**

OUTPUT:

INPUT:	OUTPUT:
POWER SUPPLY: 05V	DISPLAY SHOWS: FF
CHANNEL NO:“00 T0	

RESULT: Program for interfacing ADC and DAC to 8086 performed.

Viva:

1) Which is by default pointer for CS/ES?

2) How many segments present in it?

3) What is the size of each segment?

4) Basic difference between 8085 and 8086?

5) Which operations are not available in 8085?

EXERCISE:

1. Using the program generate a waveform and identify that

ADDRESS	OPCODE	LABEL	MNEMONIC
			MOV AL,80
			MOV DX,0FFC6
			OUT, DX
			MOV DX,0FFC2
			MOV AL,00
		L2	
		L1	OUT DX
			INC AL
			CMP AL,0FF
			JB L1
			OUT DX
			JMP L2

2. Using the program generate a waveform and identify that

PROGRAM TO GENERATE TRAINGULARWAVE

ADDRESS	OPCODE	LABEL	MNEMONIC
			MOV AL,80
			MOV DX,0FFC6
			OUT DX
		A0	MOV AL,00
			MOV DX,0FFC2
		A2	OUT DX
			INC AL
			CMP AL,0FF
			JC A2
			MOV DX,0FFC2
			OUT DX
			DEC AL

		A1	CMP AL,00
			JNBE A1
			JMP A0

EXP.NO.7: Parallel communication between two microprocessors using 8255.

AIM: To write an alp for parallel communication between two microprocessors by using 8255.

APPARATUS: 8086 Trainer kit-2, 8255, Power Supply and connectors.

PROCEDURE:-

1. Connect the 26 core FRC connector to the 8086 trainer at connector no CN4 and the interface module.
2. Connect the power mate connector to the interface module and the other side of the

   connector to the power supply. The connections to the power supply are given below.

Connections: (power supply)

Black & Red: Gnd.

Blue & Green: +5V

3. 5- Way power mate is wired to the motor. This power mate is to be inserted into the male socket provided on the interface. Care should be taken such that, below given code for the particular colored wire coincides with the code on the interface.

A- GREEN

C- RED & WHITE

B- GREEN & WHITE

D- RED

V_{DD}- BLACK & WHITE.

4. After the completion of the program and connections enter the program as given in the listing below.

G0< STARTING ADDRESS< ENTER (on the key board of trainer).

PROGRAM:

MEMORY LOCATION	OPCODE	LABEL	MNEMONICS
4000		LOOP1	MOV AL,90
			MOV DX,3006
			OUT DX
			MOV DX,3000
			IN AL DX
			NOT AL
			MOV DX,3002
			OUT DX
			MOV AL,02
			MOV DX,3006
			OUT DX
			CALL DELAY
			MOV AL 03
			MOV DX,3006
			OUT DX
			CALL DELAY
			MOV AL,OA
			MOV DX,3006
			OUT DX
			CALL DELAY
			MOV AL,OB
			MOV DX,3006
			OUT DX
			CALL DELAY
			MOV AL,OE
			MOV DX,3006
			OUT DX
			CALL DELAY
			MOV AL,OF
			MOV DX,3006
			OUT DX
			CALL DELAY
			JMP LOOP1

DELAY PROGRAM

MEMORY LOCATION	OPCODE	LABEL	MNEMONICS
			MOV CX,7FFF
4500		**NEXT**	LOOP **NEXT**
			RET

RESULT: Program for parallel communication between two microprocessors by using 8255 performed.

Viva:

1) What is the difference between min mode and max mode of 8086?

2) What is the difference between near and far procedure?

3) What is the difference between Macro and procedure?

4) What is the difference between instructions RET & IRET?

5) What is the difference between instructions MUL & IMUL?

EXERCISE:

1. 16-Bit Addition in Location mode using 8086 Microprocessor Kit.
2. 16-Bit subtraction in Location mode using 8086 Microprocessor Kit.

EXP.NO.8: Serial communication between two microprocessor kits using 8251

AIM: Interface the 8251 USART to the two 8086 microprocessor kits.

APPARATUS:

1. 8086 Trainer kit - 2no‟s
2. 8251 USART
3. Power Supply
4. Connectors.

PROCEDURE:-

1. Connect the 26 core FRC connector to the 8086 trainer at connector no CN4 and the interface module.
2. Connect the power mate connector to the interface module and the other side of the

   connector to the power supply. The connections to the power supply are given below.

Connections: (power supply)

3. 5- Way power mate is wired to the motor. This power mate is to be inserted into the male socket provided on the interface. Care should be taken such that, below given code for the particular colored wire coincides with the code on the interface.

A- GREEN

C- RED & WHITE

B- GREEN & WHITE

D- RED

4. After the completion of the program and connections enter the program as given in the listing below.

G0< STARTING ADDRESS< ENTER (on the key board of trainer)

PROGRAM:

MEMORY LOCATION	OPCODE	LABEL	MNEMONICS
4000			MOV AL,36
			MOV DX,0086H
			OUT DX,AL
			MOV DX,0080H
			MOV AL,0A
			OUT DX,AL
			MOV AL,00
			OUT DX,AL
			MOV SP,3000
			MOV DX,0092
			OUT DX,AL
			OUT DX,AL
			OUT DX,AL
			OUT DX,AL
			CALL DELAY
			MOV AL,40
			OUT DX,AL
			CALL DELAY
			MOV AL,CE
			OUT DX,AL
			CALL DELAY
			MOV AL,27
			OUT DX,AL
			CALL DELAY
			MOV SI,2100
		L1	MOV DX,0092
			IN AL,DX

			CMP AL,1B
			JE L1
			MOV DX,0090
			IN AL,DX
			AND AL,81
			CMP BL,AL
			JE L3
		L2	MOV DX,0092
			IN AL,DX
			AND AL,81
			CMP AL,81
			JNE L2
			MOV AL,BL
			MOV DX,0090
			OUT DX,AL
			OUT DX,AL
			MOV [SI],AL
			INC SI
			JMP L1
			OUT DX,AL
			INC SI
			JMP L2
		L3	INT 03

DELAY PROGRAM:

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC
4500			MOV CX,0002
		A3	LOOP A3
			RET

RESULT: Program for serial communication between two microprocessors by using 8251 Performed

Viva:

1) What is the difference between instructions DIV & IDIV?

2) What is difference between shifts and rotate instructions?

3) Which are strings related instructions?

4) Which are addressing modes and their examples in 8086?

5) What does u mean by directives?

EXERCISE:

1. Write an alp program to find the smallest number in an array using masm software.
2. Write an alp program to find the largest number in an array using masm software.

EXP.NO.9: Interfacing to 8086 and programming to control stepper motor.

AIM: Write an Assembly Language Program to rotate the Stepper Motor in clockwise as well as anti-clockwise direction.

APPARATUS: 8086 Trainer kit, Stepper,
Motor Interface Card,
Stepper Motor,
Power supply.

PROCEDURE:-

1. Connect the 26 core FRC connector to the 8086 trainer at connector no CN4 and the interface module.

2. Connect the power mate connector to the interface module and the other side of the

   connector to the power supply. The connections to the power supply are given below.

Connections: (power supply)

3. 5- Way power mate is wired to the motor. This power mate is to be inserted into the male socket provided on the interface. Care should be taken such that, below given code for the particular colored wire coincides with the code on the interface.

A- GREEN

C- RED & WHITE

B- GREEN & WHITE

D- RED

4. After the completion of the program and connections enter the program as given in the listing below.

   G0< STARTING ADDRESS< ENTER (on the key board of trainer).

Program to rotate in clockwise direction

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC
4000			MOV AL,80
			MOV DX,0FFC6
			OUT DX
			MOV BX,02
		A0	MOV CX,00FF
		A1	MOV AL,77
		A2	MOV DX,0FFC4
			OUT DX
			CALL DELAY
			MOV AL,0BB
			MOV DX,0FFC4
			OUT DX
			CALL DELAY
			MOV AL,0DD
			MOV DX,0FFC4
			OUT DX
			CALL DELAY
			MOV AL,0EE
			MOV DX, 0FFC4
			OUT DX
			CALL DELAY
			LOOP A1
			DEC BX
			JNZ A2

Program to rotate in Anti clockwise direction

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC
4000			MOV AL,80
			MOV DX,OFFC6
			OUT DX
		A0	MOV BX,0002
			MOV CX,00FF
		A1	
			MOV AL,0EE
		A2	
			MOV DX,0FFC4
			OUT DX
			CALL DELAY
			MOV AL,0DD
			MOV DX,0FFC4
			OUT DX
			CALL DELAY
			MOV AL,0BB
			MOV DX,0FFC4
			OUT DX
			CALL DELAY
			MOV AL,077
			MOV DX,0FFC4
			OUT DX
			CALL DELAY
			LOOP A1
			DEC BX
			JNZ A2

DELAY PROGRAM:

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC
4500			MOV AX,0500
		A3	NOP
			NOP
			DEC AX
			JNZ A3
			RET

RESULT: An operation to interface Stepper Motor with 8086 Microprocessor performed.

Viva:

1) What does u mean by Prefix?

2) What .model small means?

3) Difference between small, medium, tiny, huge?

4) What is dd, dw, db?

5) Interrupts in 8086 and there function.

EXERCISE:

1. Write an alp program to find the unpacked BCD to the given BCD number 56 using 8086 trainer kit?

2. Write an alp program to find the ASCII number to the given BCD number 56 using 8086 trainer kit?

EXECUTION PROCEDURE FOR 8051

Writing a alp program into 8051:

Switch on kit

Press reset

Press A (give starting address)

Press enter

Enter 1st mnemonic

Press enter

Enter 2nd mnemonic

Press enter

Enter nth mnemonic

Press enter

Press enter

Press enter

Display: A D G M T S

To compile:

Press G (give starting address)

Press enter

Display: program executed

Result:

Press R

A, B, R0, -------, Rn

Exp.No.10: Programming using arithmetic, logical and bit manipulation instructions of 8051.

I) Arithmetical operations:

i) 8 bit addition

AIM: To perform 8 bit addition by using 8051.

APPARATUS: 8051 with keyboard

PROGRAM:

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC
8000			MOV A,#02 MOV B,#02 ADD A,B LCALL 03

OUTPUT:

Input		output	
REGISTER	Data	REGISTER	Data
A	02	A	04
B	02		

ii) 8 bit subtraction

AIM: To perform 8 bit subtraction by using 8051.

APPARATUS: 8051 with keyboard **PROGRAM:**

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC
8000			MOV A,#04 MOV B,#02 SUBB A,B LCALL 03

OUTPUT:

Input		output	
REGISTER	Data	REGISTER	Data
A	04	A	02
B	02		

iii) 8 bit multiplication:

AIM: To perform 8 bit multiplication by using 8051.

APPARATUS: 8051 with keyboard **PROGRAM:**

Memory location	Opcode	Label	Mnemonic
8000			MOV DPTR,#9000 MOVX A,@DPTR MOV F0,A

			INC DPTR MOVX A,@DPTR MUL AB LCALL 03

OUTPUT:

Input		output	
MEMORY LOCATION	**Data**	**REGISTER**	**Data**
9000	03	A	06
9001	02		

iv) 8 bit division:

AIM: To perform 8 bit division by using 8051.

APPARATUS: 8051 with keyboard

PROGRAM:

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC
8000			MOV DPTR,#9000 MOVX A,@DPTR MOV R0,A

			INC DPTR MOVX A,@DPTR MOV F0,A MOV A,R0 DIV AB LCALL 03

OUTPUT:

Input		output	
MEMORY LOCATION	**Data**	**REGISTER**	**Data**
9000	03	A	06
9001	02		

v) 16 bit addition:

AIM: To perform 16 bit addition by using 8051.

APPARATUS: 8051 with keyboard

PROGRAM:

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC
8000			MOV DPTR,#9500 MOVX A,@DPTR MOV R0,A MOV R2,#00 INC DPTR MOVX A,@DPTR

			MOV R1,A
			INC DPTR
			MOVX A,@DPTR
			ADD A,R0
			MOV R6,A
			INC DPTR
			MOVX A,@DPTR
			ADDC A,R1
			JNC LOOP1
			INC R2
		LOOP1	INC DPTR
			MOVX @DPTR,A
			INC DPTR
			MOV A,R6
			MOVX @DPTR,A
			INC DPTR
			MOV A,R2
			MOVX @DPTR,A
		LOOP2:	SJMP LOOP2
			LCALL 03

OUTPUT:

Input		**output**	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
9500	BC	9504	80
9501	19	9505	34
9502	88	9506	01

9503	99		

II) logical operations:

i) AND operation

AIM: To perform AND operation by using 8051.

APPARATUS: 8051 with keyboard **PROGRAM:**

MEMORY LOCATION	OPCODE	LABEL	MEMONIC
8000			MOV R0,#DATA 1 MOV A,#DATA 2 ANL A,R0 MOV R1,A LCALL 03

OUTPUT:

Input		output	
REGISTER	**Data**	**REGISTER**	**Data**
R0	14 (DATA 1)	R1	10
A	12 DATA 2)		

ii) XOR operation

AIM: To perform AND operation by using 8051.

APPARATUS: 8051 with keyboard

PROGRAM:

MEMORY LOCATION	OPCODE	LABEL	MEMONIC
8000			MOV R0,#DATA 1 MOV A,#DATA 2 XRL A,R0 MOV R1,A LCALL 03

OUTPUT:

Input		output	
REGISTER	**Data**	**REGISTER**	**Data**
R0	23 (DATA 1)	R1	17
A	34 DATA 2)		

BIT AND BYTE OPERATIONS BY USING 8051

AIM: To write an assembly language program to perform the BIT and BYTE operations like set, reset and swap by using 8051 microcontroller.

APPARATUS: **1.**8051 Micro Controller kit.

2.Key Board.

3.Adapter.

PROGRAM:

BIT OPERATIONS:

SET A BIT:

```
            MOV         DPTR, #STARTING ADDRESS
            MOVX        A, @DPTR
            SETB        0E5
            INC         DPTR
            MOVX        @DPTR, A
L4:   SJMP        L4 (OFFSET ADDRESS)
      LCALL       03
```

RESET A BIT:

```
  MOV         DPTR, #STARTING ADDRESS
 MOVX          A, @DPTR
 CLR          0E5
 INC          DPTR
 MOVX          @DPTR, A
L4:  SJMP     L4 (OFFSET ADDRESS)
 LCALL        03
```

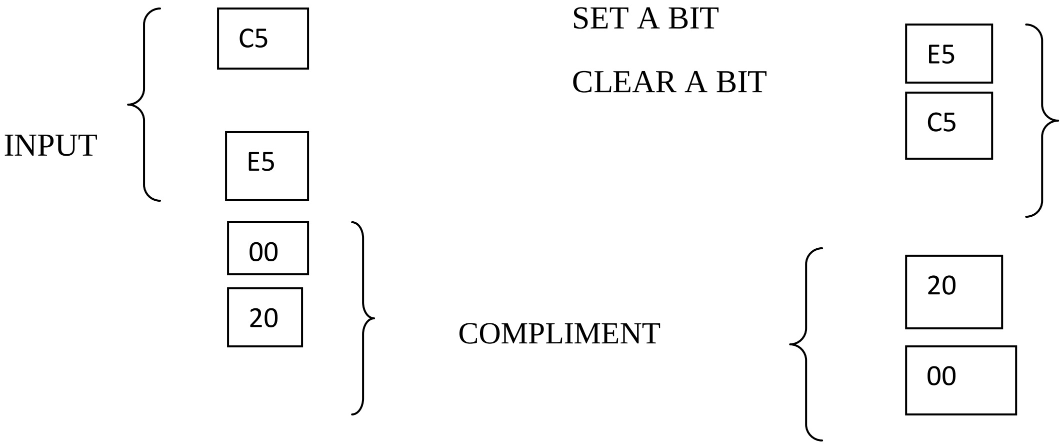
COMPLIMENT A BIT:

```
 MOV          DPTR, #STARTING ADDRESS
```

```
    MOVX                               A, @DPTR
    CPL                               0E5
    INC                               DPTR
    MOVX                               @DPTR, A


  L4: SJMP                            L4 (OFFSET ADDRESS)
    LCALL                             03
```

OUTPUT:



PROGRAM:

BYTE OPERATONS:

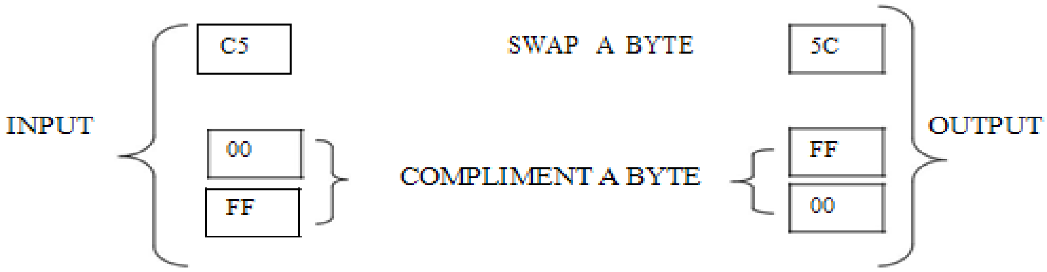
SWAP A BYTE:

```
        MOV          DPTR, #STARTING ADDRESS
        MOVX         A, @DPTR
        SETB         A
        INC          DPTR
        MOVX         @DPTR, A
    L4: SJMP         L4 (OFFSET ADDRESS)
        LCALL        03
```

COMPLIMENT A BYTE:

```
        MOV      DPTR, #STARTING ADDRESS
        MOVX      A, @DPTR
        CPL      A
        INC      DPTR
        MOVX      @DPTR, A
L4:     SJMP     L4 (OFFSET ADDRESS)
        LCALL    03
```

OUTPUT:



RESULT: Programs for arithmetic, logical and bit manipulation instructions of 8051 perfor

Viva:

1) What is the function of 01h of Int 21h?

2) What is the function of 02h of Int 21h?

3) What is the function of 09h of Int 21h?

4) What is the function of 0Ah of Int 21h?

5) What is the function of 4ch of Int 21h?

EXERCISE:

1. Write an alp program to perform OR operation using 8051 microcontroller trainer Kit?

2. Write an alp program to perform addition and subtraction operation using 8051 microcontroller trainer Kit

   a) 56

   b) 12

EXP.NO.11: PROGRAM AND VERIFY TIMER/COUNTER IN 8051

AIM: To Perform Timer 0 and Timer 1 in Counter Mode and Gated Mode Operation.

APPARATUS: 8051kit with keyboard, timer module kit, frc cables & power supply.

PROCEDURE:

1. Make the power supply connections from 4-way power mate connector on the ALS-NIFC-09 board.
    - +5V ……...blue wire
    - Ground …..black wire
2. Connect 26-pin flat cable from interface module to P1 of the trainer kit.
3. Enter the program in the RAM location in 9000 and execute the program

   GO<STARTING ADDRESS><EXEC>

PROGRAM TO VERIFY TIMER '0'- COUNTER MODE:

ADDRESS	OPCODE	LABEL	MNEMONICS
9200			MOV A,TMOD (TMOD=89)
			ORL A,#05H
			MOV TMOD,A
			SETB TRO (TRO=8C)
			LCALL 68EAH
		LOOP	MOV DPTR,#0194H

			MOV A,TLO (TLO=8A)
			MOVX @DPTR,A
			INC DPTR
			MOV A,THO (THO=8C)
			MOVX @DPTR,A
			LCALL 6748H
			SJMP LOOP

Execution:1) short jp1 of 1&2 pins and press sw1 for manual increment

2) Short jp1 of 2&3 pins for auto increment

PROGRAM TO VERIFY TIMER-1 COUNTER MODE:

ADDRESS	OPCODE	LABEL	MNEMONICS
9100			MOV A, TMOD (TMOD=89)
			ORL A,#50H
			MOV TMOD,A
			SETB TR1 (TR1=8E)
			LCALL 68EAH
		LOOP	MOV DPTR,#0194H
			MOV A,TL1 (TL1=8B)
			MOVX @DPTR,A
			INC DPTR
			MOV A,TH1 (TH1=8D)
			MOVX @DPTR,A
			LCALL 6748H
			SJMP LOOP

Execution: 1) short jp1 of 5&6 pins and press sw2 for manual increment 2) Short jp2 of 4&5 pins for auto increment

RESULT: Programs for Timer 0 and Timer 1 in Counter Mode and Gated Mode Operations

performed**.**

Viva:

1) What is the reset address of 8086?

2) What is the size of flag register in 8086? Explain all.

3) What is the difference between 08H and 01H functions of INT 21H?

4) Which is faster- Reading word size data whose starting address is at even or at odd address of memory in 8086?

5) Which is the default segment base: offset pairs?

EXERCISE:

1. write an ALP program to study timer-1 gated mode

EXP.NO.12: PROGRAM AND VERIFY INTERRUPT HANDLING IN 8051.

AIM: Write ALP in 8051 to allow the external interrupt 1.

APPARATUS: 8051 with keyboard interrupt kit module.

PROCEDURE:

1. Make the power supply connections from 4-way power mate connector on the ALS-NIFC-09 board.
   - +5V ……...blue wire
   - Ground …..black wire
2. Connect 26-pin flat cable from interface module to P1 of the trainer kit.
3. Enter the program in the RAM location in 9000 and execute the program

   GO<STARTING ADDRESS><EXEC>

PROGRAM:

MEMORY LOCATION	OPCODE	LABEL	MEMONIC
			ORG 0000
		AGAIN	LJMP AGAIN
			ORG 0013
			SETB P1.3
			MOV R3,#255
		BACK	DJNZ R3,BACK
			CLR P1.3
			ORG 30H
		MAIN	MOV IE,#10000100B
		HERE	SJMP HERE
			LCALL 03

OUTPUT:

1. When key is pressed, LED ON.
2. When key is opened, LED OFF

RESULT: program for interrupt handling in 8051 verified.

Viva:

1) Can we use SP as offset address holder with CS?

2) Which is the base registers in 8086?

3) Which is the index registers in 8086?

4) What do you mean by segment override prefix?

5) Whether micro reduces memory requirements?

EXERCISE:

1. Write an alp program to find the length of the given array using masm software.
2. Write an alp program to find the sum of „n" numbers using masm software.

EXP. NO.13: UART OPERATION IN 8051

AIM: Write ALP Of UART operation in 8051.

APPARATUS: 8051 with keyboard UART module.

PROCEDURE:

1. Make the power supply connections from 4-way power mate connector on the ALS-NIFC-09 board.
   - +5V ……...blue wire
   - Ground …..black wire
2. Connect 26-pin flat cable from interface module to P1 of the trainer kit.
3. Enter the program in the RAM location in 9000 and execute the program

   GO<STARTING ADDRESS><EXEC>

PROGRAM:

SEND CHAR:

MEMORY LOCATION	OPCODE	LABEL	MEMONIC
			MOV SBUF,A
		B0	JNB TI,B0
			CLR TI
			RET
		N1	JNB RI,N1
			MOV A,SBUF
			CLR RI
			RET
		INITSMOD	MOV SCON,#52
			MOV TMOD,#20
			MOV TH1,#0F4
			SETB TRI
			RET

MODE 0:

MEMORY LOCATION	OPCODE	LABEL	MEMONIC
			MOV SCON,#0D
			CLR TI
		LOOP	MOV SBUF,#0AA
		HERE	JNB TI,HERE
			CLR TI
			SJMP LOOP
			LCALL 03

MODE 1:

MEMORY LOCATION	OPCODE	LABEL	MEMONIC
			MOV SCON,#40
			MOV TMOD,#20
			MOV TH,#0D0
			MOV TCON,#40
			CLR TI
		LOOP	MOV SBUF,#0AA
		W1	JNB TI,W1
			CLR TI
			JMP LOOP

INTERRUPT DRIVEN MODE:

MEMORY LOCATION	OPCODE	LABEL	MEMONIC
			JMP SERIAL_INT
		MAIN	MOV SCON,#50
			MOV TMOD,#20
			MOV TMI,#0DD
			MOV PCON,#80
			MOV IE,#90
			CLR RI
		LOOP	JMP LOOP
		SERIAL_INT	CLR RI
			MOV PI,SBUF
			RETI
			LCALL 03

RESULT: Program for URAT operation in 8051 performed.

Viva:

1) What do you mean by macro?

2) What is diff between macro and procedure?

3) Types of procedure?

4) What TASM is?

5) What TLINK is?

EXERCISE:

1. Write an alp program to perform an operation to find the sum of squares of a given array using masm software.

2. Write an alp program to perform an operation to find the cubes of squares of a given array using masm software

EXP.NO 14: COMMUNICATION BETWEEN 8051 KIT AND PC.

AIM: Interface an 8051 microcontroller trainer kit to pc and establish a communication between them through RS 232.

APPARATUS:

1. ESA 8051 Trainer kit
2. 8251 USART,
3. PC
4. Power Supply
5. Connectors.

PROCEDURE:

1. Make the power supply connections from 4-way power mate connector on the ALS-NIFC-09 board.
   - +5V ……...blue wire
   - Ground …..black wire
2. Connect 26-pin flat cable from interface module to P1 of the trainer kit.
3. Enter the program in the RAM location in 9000 and execute the program

PROGRAM:

ADDRESS	OPCODE	LABEL	MNEMONICS
			MOV A,#36 MOV DPTR,#2043 MOVX @DPTR,A MOV DPTR,#2040 MOV A,#0A MOVX @DPTR,A

			MOV A,#00
			MOVX @DPTR,A
			MOV R1,#3000

			MOV DPTR,#0092
			MOVX @DPTR,A
			MOVX @DPTR,A
			MOVX @DPTR,A
			MOVX @DPTR,A
			CALL DELAY
			MOV A,#40
			MOVX @DPTR,A
			CALL DELAY
			MOV A,#CE
			MOVX @DPTR,A
			CALL DELAY
			MOV A,#27
			MOVX @DPTR,A
			CALL DELAY
			MOV DPTR,9000
			MOV DPTR,#0092
		UP	MOVX @DPTR,A
			CMP A,1B
			JE UP
			MOV DPTR,#0090
			MOVX @DPTR,A
			ANL A,81
			CJNE B,A.DOWN
			MOV DPTR,#0092

		UP1	MOVX @DPTR,A
			ANL A,81
			CJNE AL,81.UP1
			MOV A,B
			MOV DPTR,#0090
			MOVX @DPTR,A

			MOVX @DPTR,A
			MOV R3,9700
			MOV R3,A
			INC R3
			JMP UP
			MOVX @DPTR,A
			INC R3
			JMP UP
			INT 03
		DOWN	MOV CX,0002
		DELAY	LOOP HERE
		HERE	RET

.

RESULT: Thus, the 8251 USART can be used to establish communication between two processors by receiving the characters from the USART and displaying these characters on the console.

Viva:

1) What TD is?

2) What do u mean by assembler?

3) What do u mean by linker?

4) What do u mean by loader?

5) What do u mean by compiler?

EXERCISE:

1. Write an alp program to perform an operation to find the squares of a given number using masm software.
2. Write an alp program to perform an operation to find the squares of a given number using MP trainer kit

EXP.NO.15: Interfacing LCD to 8051.

AIM: Interface an LCD with 8051 microcontroller.

APPARATUS:

1. 8051 Trainer kit
2. LCD module
3. FRC cables
4. Power Supply.

PROCEDURE:

1. Make the power supply connections from 4-way power mate connector on the ALS-NIFC-09 board.
   - +5V ……...blue wire
   - Ground …..black wire
2. Connect 26-pin flat cable from interface module to P1 of the trainer kit.
3. Enter the program in the RAM location in 9000 and execute the program

   GO<STARTING ADDRESS><EXEC>

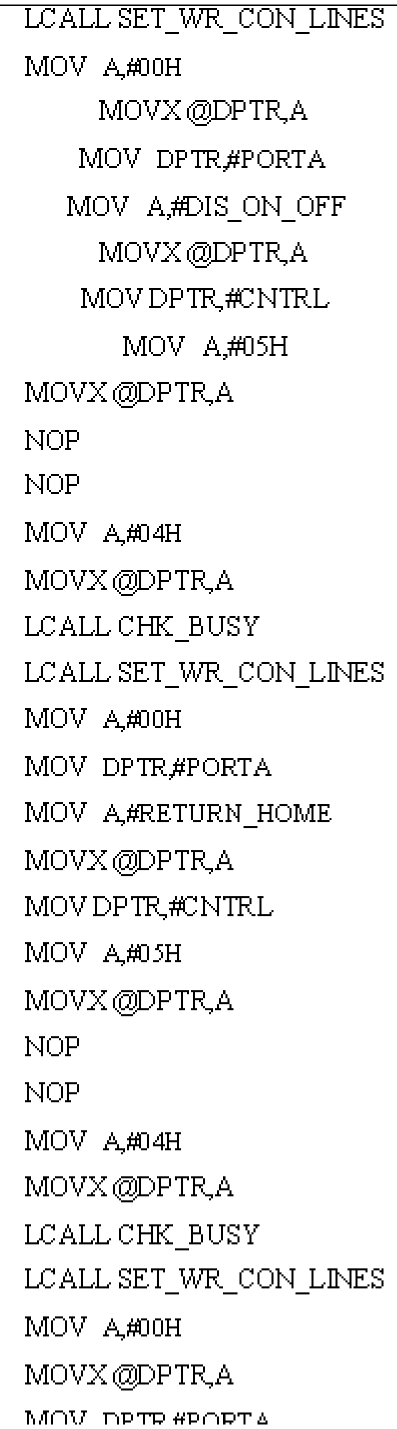
PROGRAM:

```
CNTRL          EQU    2043H                ; 8255 control port address
PORTC                 EQU    2042H         ; 8255 port C address
PORTB                 EQU    2041H         ; 8255 port B address
PORTA                 EQU    2040H         ; 8255 port A address
FUNCTION_SET          EQU    38H           ; display commands
DIS_ON_OFF EQU        0EH
RETURN_HOME           EQU    02H
MODE_SET   EQU        06H
CLEAR_DIS  EQU        01H
DDRAM_ADD             EQU    80H
CNT  EQU 40H

CNT1 EQU 41H
CNT2 EQU 42
```

ADDRESS	OP CODE	LABEL	MNEMONICS
			MOV SP,#50H
			MOV PSW,#00H
			MOV CNT2,#10H
			MOV R0,#14H
			MOV R1,#FFH
			LCALL DELAY
			MOV DPTR,#CNTRL
			MOV A,#80H
			MOVX @DPTR,A
			LCALL SET_CON_LINES
		BACK	MOV R2,#03H
			LCALL SET_WR_CON_LINES
			MOV A,#00H
			MOVX @DPTR,A
			MOV DPTR,#PORTA
			MOV A,#FUNCTION_SET
			MOVX @DPTR,AMOV
			DPTR,#CNTRL
			MOV A,#05H
			MOVX @DPTR,A
			NOP
			NOP
			MOV A,#04H
			MOVX @DPTR,A
			MOV R0,#06H
			MOV R1,#E4H
			LCALL DELAY
			DJNZ R2,BACK

			LCALL CHK_BUSY

			 LCALL SET_WR_CON_LINES MOV A,#00H MOVX @DPTR,A MOV DPTR,#PORTA MOV A,#DIS_ON_OFF MOVX @DPTR,A MOV DPTR,#CNTRL MOV A,#05H MOVX @DPTR,A NOP NOP MOV A,#04H MOVX @DPTR,A LCALL CHK_BUSY LCALL SET_WR_CON_LINES MOV A,#00H MOV DPTR,#PORTA MOV A,#RETURN_HOME MOVX @DPTR,A MOV DPTR,#CNTRL MOV A,#05H MOVX @DPTR,A NOP NOP MOV A,#04H MOVX @DPTR,A LCALL CHK_BUSY LCALL SET_WR_CON_LINES MOV A,#00H MOVX @DPTR,A MOV DPTR,#PORTA

			MOV A,#MODE_SET MOVX @DPTR,A MOV DPTR,#CNTRL MOV A,#05H MOVX @DPTR,A NOP NOP MOV A,#04H MOVX @DPTR,A LCALL CHK_BUSY LCALL SET_WR_CON_LINES MOV A,#00H MOVX @DPTR,A MOV DPTR,#PORTA MOV A,#CLEAR_DIS MOVX @DPTR,A MOV DPTR,#CNTRL MOV A,#05H MOVX @DPTR,A NOP NOP MOV A,#04H MOVX @DPTR,A MOV CNT1,#02H MOV CNT,#08H MOV R0,#DDRAM_ADD LCALL CHK_BUSY LCALL SET_WR_CON_LINES MOV A,#00H MOV DPTR,#PORTA MOV A,R0

			MOVX @DPTR,A
			MOV DPTR,#CNTRL
			MOV A,#05H
			MOVX @DPTR,A
			NOP
			NOP
			MOV A,#04H
			MOVX @DPTR,A
			CLR A
			MOV DPTR,#MSG
			MOVX A,@DPTR
		BACK3	MOV R1,A
			INC DPTR
			PUSH DPH
			PUSH DPL
			LCALL CHK_BUSY
			LCALL SET_WR_CON_LINES
			MOV A,#01H
			MOVX @DPTR,A
			MOV DPTR,#PORTA
			MOV A,R1
			MOVX @DPTR,A
			MOV DPTR,#CNTRL
			MOV A,#05H
			MOVX @DPTR,A
			NOP
			NOP
			MOV A,#04H
			MOVX @DPTR,A

			POP DPL POP DPH

			CLR A
			PUSH R0
			PUSH R1
			MOV R0,#7FH
			MOV R1,#FFH
			LCALL DELAY
			POP R1
			POP R0
			DJNZ CNT,BACK3
			DJNZ CNT1,F1
			DJNZ CNT2,FORW1
			LJMP FORW
			MOV CNT,#08H
		F1	PUSH DPH
			PUSH DPL
			LCALL CHK_BUSY
			LCALL SET_WR_CON_LINES
			MOV A,#00H
			MOVX @DPTR,A
			MOV DPTR,#PORTA
			MOV A,#C0H
			MOVX @DPTR,A
			MOV DPTR,#CNTRL
			MOV A,#05H
			MOVX @DPTR,A
			NOP

				NOP
				MOV A,#04H
				MOVX @DPTR,A
				POP DPL
				POP DPH

				CLR A
				LJMP BACK3
				PUSH DPH
			FORW1	PUSH DPL
				MOV R0,#DDRAM_ADD
				LCALL CHK_BUSY
				LCALL SET_WR_CON_LINES
				MOV A,#00H
				MOVX @DPTR,A
				MOV DPTR,#PORTA
				MOV A,R0
				MOVX @DPTR,A
				MOV DPTR,#CNTRL
				MOV A,#05H
				MOVX @DPTR,A
				NOP
				NOP
				MOV A,#04H
				MOVX @DPTR,A
				MOV CNT,#08H
				MOV CNT1,#02H
				POP DPL
				POP DPH

			CLR A
			LJMP BACK3
		FORW	:LCALL 0003H
		SET_CON_LINES:	MOV DPTR,#CNTRL
			MOV A,#01H
			MOVX @DPTR,A
			MOV A,#03H
			MOVX @DPTR,A
			MOV A,#04H
			MOVX @DPTR,A
			RET
		CHK_BUSY:	MOV DPTR,#CNTRL
			MOV A,#90H
			MOVX @DPTR,A
			MOV A,#04H
			MOVX @DPTR,A
			MOV A,#00H
			MOVX @DPTR,A
			MOV A,#03H
			MOVX @DPTR,A
		BACK2	MOV A,#05H
			MOVX @DPTR,A
			MOV DPTR,#PORTA
			MOVX A,@DPTR
			MOV B,A
			MOV DPTR,#CNTRL
			MOV A,#04H
			MOVX @DPTR,A

			MOV A,B
			JNB A.7,F2
			LJMP BACK2
		F2	MOV DPTR,#CNTRL
			MOV A,#80H
			MOVX @DPTR,A
			RET
		SET_WR_CON_LINES:	MOV DPTR,#CNTRL
			MOV A,#04H
			MOVX @DPTR,A
			MOV A,#02H

			MOVX @DPTR,A
		DELAY:	RET
		LOOP1:	PUSH R1
		LOOP:	NOP
			DJNZ R1,LOOP
			POP R1
			DJNZ R0,LOOP1
			RET

RESULT: program for interfacing an LCD with 8051 microcontroller performed.

Viva:

1) What do you mean by emulator?

2) Stack related instruction?

3) .stack 100 means?

4) What do you mean by 20 dup (0)?

5) Which flags of 8086 are not present in 8085?

EXERCISE:

1. Write an alp program to perform an operation to find the cubes of a given number using masm software

2. Write an alp program to perform an operation to find the cubes of a given numbers using MP trainer kit

EXP.NO.16: Interfacing Matrix/Keyboard to 8051.

AIM: Interface a Keyboard to 8051 microcontroller.

APPARATUS: 8051 Trainer kit, keyboard module, FRC cables, & Power Supply.

PROCEDURE:

1. Make the power supply connections from 4-way power mate connector on the ALS-NIFC-09 board.
   - +5V ……...blue wire
   - Ground …..black wire
2. Connect 26-pin flat cable from interface module to P1 of the trainer kit.
3. Enter the program in the RAM location in 9000 and execute the program

   GO<STARTING ADDRESS><EXEC>

PROGRAM:

```
CNTRL      EQU    2043H         ;CONTROL PORT ADDRESS OF 8255
PORTA      EQU    2040H         ;PORTA ADDRESS OF 8255
PORTB      EQU    2041H         ;PORTB ADDRESS OF 8255
PORTC      EQU    2042H         ;PORTC ADDRESS OF 8255
```

ADDRESS	**OP CODE**	**LABEL**	**MNEMONICS**
			MOV A,#90H
			MOV DPTR,#CNTRL
			MOVX @DPTR,A
			MOV B,#20H
		BLINK2	MOV DPTR,#PORTB
			MOV A,#FFH

			MOVX @DPTR,A
			MOV DPTR,#PORTC
			MOV A,#00H
			MOVX @DPTR,A
			MOV A,#F0H
			MOVX @DPTR,A

			DJNZ B,BLNK2
		BACK	MOV A,#FEH
			MOV B,#21H
		BLINK1	MOV DPTR,#PORTB
			MOVX @DPTR,A
			MOV DPTR,#PORTC
			MOV A,#00H
			MOVX @DPTR,A
			MOV A,#F0H
			MOVX @DPTR,A
			LCALL DELAY
			RL A
			DJNZ B,BLNK1
			SJMP BACK
		DELAY:	MOV R0,#F7H
		OLOOP:	MOV R1,#FFH
		ILOOP:	DJNZ R1,ILOOP
			DJNZ R0,OLOOP
			RET

RESULT: program for interfacing a keyboard to 8051 microcontroller performed.

Viva:

1) What is the size of flag register?

2) Can you perform 32 bit operation with 8086? How?

3) Whether 8086 is compatible with Pentium processor?

4) What is 8087? How it is different from 8086?

5) While accepting no. from user why u need to subtract 30 from that?

EXERCISE:

1. Write an alp program for addition of multi byte numbers.
2. Write an alp program for multiplication of given number in location mode
   a) 0060
   b) 0002

EXP.NO.17: Data Transfer from Peripheral to Memory through DMA controller 8237/8257.

AIM: Write a alp program for data transfer from peripheral to memory through DMA controller 8257.

APPARATUS:

1. 8086 Trainer kit
2. 8257 module
3. FRC cables
4. Power Supply.

PROGRAM:

DMAL = 00

DMAH = 50

TCL = FF

TCH = 47

ADDRESS	OP CODE	LABEL	MNEMONICS
			MOV AX,DMAL
			OUT 80,AX
			MOV AX,DMAH
			OUT 80,AX
			MOV AL,TCL

			OUT 81,AL MOV AL,TCH OUT 81,AL MOV AL,41

RESULT: Program for transfer from Peripheral to Memory through DMA controller 8257 performed.

Viva:

1) While displaying no. from user why u need to add 30 to that?

2) What are ASCII codes for nos. 0 to F?

3) How does U differentiate between positive and negative numbers?

4) What is range for these numbers?

5) Which no. representation system you have used?

EXERCISE:

1. Write an alp program to divide 32 bit by the 16 bit.

2. Write an alp program for median of an array.

These interview questions test the knowledge of x86 Intel architecture and 8086 microprocessor specifically.

1. What is a Microprocessor? - Microprocessor is a program-controlled device, which fetches the instructions from memory, decodes and executes the instructions. Most Micro Processor are single-chip devices.

2. Give examples for 8 / 16 / 32 bit Microprocessor? - 8-bit Processor - 8085 / Z80 / 6800; 16-bit Processor - 8086 / 68000 / Z8000; 32-bit Processor - 80386 / 80486.

3. Why 8085 processor is called an 8 bit processor? - Because 8085 processor has 8 bit ALU (Arithmetic Logic Review). Similarly 8086 processor has 16 bit ALU.

4. What is 1st / 2nd / 3rd / 4th generation processor? - The processor made of PMOS / NMOS / HMOS / HCMOS technology is called 1st / 2nd / 3rd / 4th generation processor, and it is made up of 4 / 8 / 16 / 32 bits.

5. Define HCMOS? - High-density n- type Complimentary Metal Oxide Silicon field effect transistor.

6. What does microprocessor speed depend on? - The processing speed depends on DATA BUS WIDTH.

7. Is the address bus unidirectional? - The address bus is unidirectional because the address information is always given by the Micro Processor to address a memory location of an input / output devices.

8. Is the data bus is Bi-directional? - The data bus is Bi-directional because the same bus is used for transfer of data between Micro Processor and memory or input / output devices in both the direction.

9. What is the disadvantage of microprocessor? - It has limitations on the size of data. Most Microprocessor does not support floating-point operations.

10. What is the difference between microprocessor and microcontroller? - In Microprocessor more op-codes, few bit handling instructions. But in Microcontroller: fewer op-codes, more bit handling Instructions, and also it is defined as a device that includes micro processor, memory, & input / output signal lines on a single chip.

11. What is meant by LATCH? - Latch is a D- type flip-flop used as a temporary storage device controlled by a timing signal, which can store 0 or 1. The primary function of a Latch is data storage. It is used in output devices such as LED, to hold the data for display.

12. Why does microprocessor contain ROM chips? - Microprocessor contain ROM chip because it contain instructions to execute data.

13. What is the difference between primary & secondary storage device? - In primary storage device the storage capacity is limited. It has a volatile memory. In secondary storage device the storage capacity is larger. It is a nonvolatile memory. Primary devices are: RAM / ROM. Secondary devices are: Floppy disc / Hard disk.

14. Difference between static and dynamic RAM? - Static RAM: No refreshing, 6 to 8 MOS transistors are required to form one memory cell, Information stored as voltage level in a flip flop. Dynamic RAM: Refreshed periodically, 3 to 4 transistors are required to form one memory cell; Information is stored as a charge in the gate to substrate capacitance.

15. What is interrupt? - Interrupt is a signal send by external device to the processor so as to request the processor to perform a particular work.

16. What is cache memory? - Cache memory is a small high-speed memory. It is used for temporary storage of data & information between the main memory and the CPU (center processing unit). The cache memory is only in RAM.

17. What is called .Scratch pad of computer? - Cache Memory is scratch pad of computer.

18. Which transistor is used in each cell of EPROM? - Floating .gate Avalanche Injection MOS (FAMOS) transistor is used in each cell of EPROM.

19. Differentiate between RAM and ROM? - RAM: Read / Write memory, High Speed, Volatile

Memory. ROM: Read only memory, Low Speed, Non Voliate Memory.

20. What is a compiler? - Compiler is used to translate the high-level language program into machine code at a time. It doesn‟t require special instruction to store in a memory, it stores automatically. The Execution time is less compared to Interpreter.

21. Which processor structure is pipelined? - All x86 processors have pipelined structure.

22. What is flag? - Flag is a flip-flop used to store the information about the status of a processor and the status of the instruction executed most recently

23. What is stack? - Stack is a portion of RAM used for saving the content of Program Counter and general purpose registers.

24. Can ROM be used as stack? - ROM cannot be used as stack because it is not possible to write to ROM.

25. What is NV-RAM? - Nonvolatile Read Write Memory also called Flash memory. It is also known as shadow RAM.