



# SIDDHARTHA

## INSTITUTE OF TECHNOLOGY & SCIENCES

Korremula Road, Narapally, Ghatkesar Mandal, Ranga Reddy District.



**LAB MANUAL**  
*of*  
**MICROPROCESSOR AND MICRO CONTROLLER**  
**(III B.Tech ECE- I Semester- R18 AY: 2022 2023)**

**Department of**

*Electronics & communication Engineering*

**2022-2023**

**[www.siddhartha.co.in](http://www.siddhartha.co.in)**

## **LAB INSTRUCTIONS**

### **GUIDELINES TO WRITE YOUR OBSERVATION BOOK**

1. Assembly Language Programs (ALP's), Algorithm, Theoretical Result and Practical Result should be on right side.
2. Flow chart should be left side
3. Result should always be in the ending.
4. You all are advised to leave sufficient no of pages between ALP's for theoretical or model calculations purpose.

### **DO'S AND DON'TS IN THE LAB:**

#### **DO'S:**

1. Proper dress has to be maintained while entering in the Lab. (Boys Tuck in and shoes and girls should be neatly dressed)
2. Students should carry observation notes and record completed in all aspects.
3. ALP and its theoretical result should be there in the observation before coming to the next lab.
4. Student should be aware of next ALPs.
5. Students should be at their concerned desktop, unnecessary moment is restricted.
6. Student should follow the procedure to start executing the ALP they have to get signed by the Lab instructor for theoretical result then with the permission of Lab instructor they need to switch on the desktop and after completing the same they need to switch off and keep the chairs properly.
7. After completing the ALP Students should verify the ALP by the Lab Instructor.
8. The Practical Result should be noted down into their observations and result must be shown to the Lecturer In-Charge for verification.
9. Students must ensure that all switches are in the OFF position, desktop is shut down properly.

#### **DON'TS:**

1. Don't come late to the Lab.
2. Don't leave the Lab without making proper shut down of desktop and keeping the chairs properly.
3. Don't leave the Lab without verification by Lab instructor.
4. Don't leave the lab without the permission of the Lecturer In-Charge

### **List of Experiments**

The following programs/experiments are written for assembler and execute the same with 8086 and 8051 kits.

#### **Cycle-1: Using 8086 Processor Kits and /or Assembler.**

##### **Assembly Language Programs to 8086 to perform.**

1. Arithmetic, Logical Operations, Rotate, Shift Swap and Branch Operations.
2. Bit Level Logical Operations, Rotate, Shift, Swap, Branch Operations.

#### **Cycle-2: Using 8051 Microcontroller kit.**

1. Assembly language program to perform Arithmetic (both signed and unsigned) 16 bit data operations, Logical operations (Byte and bit level Operations, Rotate, shift, Swap, Branch Instructions).
2. Time Delay Generation using Timers of 8051
3. Serial Communication from/to 8051 to /from I/O devices.
4. Generating a Square wave using 8051.

#### **Cycle-3 Interfacing I/O Devices to 8051**

1. 7 Segment Display.
2. Keyboard.
3. Sequence Generator
4. 8-bit ADC
5. Triangular Wave Generation through back interfaces.

### **EXECUTION PROCEDURE FOR 8086 (for registers)**

- i) Writing a alp program into processor:

Switch On Power Supply

Press A 2times

SG 0 press enter

DA starting address

Press enter

N

Then display shows @ here you have to write 1<sup>st</sup> mnemonic

Press enter

N (display displays address opcode mnemonic)

N

Then display shows @ here you have to write 2nd mnemonic

Press enter

N

N

-

-

Then display shows @ up to last mnemonic

Press enter

N

N

! Press enter

EX press enter

- ii) Execution of program (for registers):

G enter starting address

Press enter

- iii) Verify the result (for registers): press

R Press E.

**EXECUTION PROCEDURE FOR 8086 (for memory locations)**

- i) Writing a program into processor:

Switch On Power Supply

Press A

A

SG 0

Press enter

DA starting address

Press enter

N

Then display shows @ here you have to write 1<sup>st</sup> mnemonic

Press enter

N(display displays address opcode mnemonic)

N

Then display shows @ here you have to write 2nd mnemonic

Press enter

N

N

-

-

-

-

Then display shows @ up to last mnemonic

Press enter

N

N

! Press enter

EX press enter

E (exam byte)

Here you have to type SI address, give 1<sup>st</sup> data, 2nd data,----- , nth data, Press enter

ii) Execution of program (for memory locations):

G enter starting address

Press enter

iii) Verify the result (for memory locations):

E

Then give DI address press, then display shows the  
result of 1<sup>st</sup> 8 bit data

For 2<sup>nd</sup> 8 bit data again press,----- , nth data

**Features of the ALS-SDA-86 8086 MEL Microprocessor Trainer**

- 8086 CPU operating at 5 MHz MAX mode.
- Provision for on-board 8087 coprocessor.
- Provision for 256 KB of EPROM & 256 KB of RAM onboard
- Battery backup facility for RAM.
- 48 programmable I/O lines using two 8255's
- Three 16 bit timers using 8253A
- Priority Interrupt Controller (PIC) for eight input using 8259A
- Computer compatible Keyboard.
- Display is 16 x 2 line LCD.
- Designed & engineered to integrate user's application specific interface conveniently at a minimum cost.
- Powerful & user-friendly keyboard / serial monitor, support in development of application programs.
- Software support for development of programs on Computer, the RS-232C interface cable connecting to computer from the kit facilitates transfer of files between the trainer kit & computer for development & debugging purposes.
- High quality reliable PCB with solder mask on both sides & clear legend prints with maximum details provided for the user.

## **SPECIFICATIONS:**

- ❖ **CPU** : Intel 8086 operating at 5Mhz in MAX mode.
- ❖ **MEMORY** : Total 256KB of memory is in the Kit provided.
  - **EPROM** : 2 JEDEC compatible sockets for EPROM.
  - **RAM** : 2 JEDEC compatible sockets for RAM.
- ❖ **PARALLEL I/O** : 48 I/O lines using two 8255"s.
- ❖ **SERIAL I/O** : One RS-232C compatible interface Using USART 8251A .
- ❖ **TIMER** : Three 16 bit counter / timers 8253A  
Counter 1 is used for serial I/O  
Baud rate generation.
- ❖ **PIC** : Programmable Interrupt controller  
Using 8253A provides interrupt  
Vectors for 8 jumpers selectable  
Internal /External sources.
- ❖ **KEYBOARD / DISPLAY**
  - **Keyboard** : Computer keyboard can be hooked on to the trainer.
  - **Display** : LCD 2x16 display.



❖ **INTERRUPTS**

- NIM : Provision for connecting NMI to a key switch
- INTR : Programmable Interrupt controller using 8259A provides Interrupt vectors for 8 jumper selectable Internal / External Sources.

❖ **INTERFACE BUS SIGNALS**

- CPU BUS : All address, data & control lines are TTL compatible & are terminated in berg strip header.
- PARALLEL I/O: All signals are TTL compatible & Terminated in berg strip header For PPI expansion.
- SERIAL I/O: Serial port signals are terminated in Standard 9-pin „D“ type connector.

❖ **MONITOR SOFTWARE**

128KB of serial / Keyboard monitor with Powerful commands to enter, verify and Debug user programs, including onboard Assembler and disassemble commands.

❖ **COMPUTER INTERFACE**

This can be interfaced to host computer System through the main serial port, also Facilitates uploading, downloading of Intel Hex files between computer and the trainer.

❖ **POWER REQUIREMENTS**

+5V DC with 2.5 Amps current rating (Max).

## 1.2 OPERATING CONFIGURATION

Two different modes of operation trainer are possible. They are

- (i) Serial operation
- (ii) Keyboard operation

The first configuration requires a computer system with an RS-232C port, can be used as the controlling device. When a computer system is interfaced to trainer, the driver program must be resident in the computer system.

The second mode of operation is achieved through Onboard KEYBOARD / DISPLAY. In this mode, the trainer kit interacts with the user through a computer keyboard and 16x2 LCD Display. This configuration eliminates the need for a computer and offers a convenient way for using the trainer as a stand – alone system.

## Exp No.1: Programs for 16 bit arithmetic operations for 8086 (using Various 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            ----1  
                                  2.RPS (+5V)                                ----1

**PROGRAM:**

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

.

#### ii) Multi byte addition

[illegible]

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

**PROGRAM:**

**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



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

#### **PROGRAM:**

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

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

**PROGRAM:**

**A) 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

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

**PROGRAM:**

**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

**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

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

### PROGRAM:

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

ii) **DESCENDING ORDER**

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

**APPARATUS:** 1. 8086 microprocessor kit----- 1

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

**PROGRAM:**

**B) By using 8086 kit:**

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC
4000			MOV AX, 0000H
			MOV CH, 0004H
		UP1:	DEC CH
			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

**OUTPUT:**

Input		output	
MEMORY LOCATION	Data	MEMORY LOCATION	Data

**RESULT:** Program for sorting an array performed by using trainer kit.

**Viva:**

- 1) What are the functions of BIU?
- 2) What are the functions of EU?
- 3) How many pin IC 8086 is?
- 4) What IC8086 is?
- 5) 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

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

### PROGRAM:

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
		LOOP:	MOV AX, 01
			MOV [DI], AX
			MOV AH, 4CH
			INT 3H

### OUTPUT:

Input		output	
MEMORY LOCATION	Data	MEMORY LOCATION	Data

**RESULT:** Program for search a number or character from a string performed by using masn 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



[illegible]

## **2) Reverse of a data:**

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

**APPARATUS:**

1. 8086 microprocessor kit-----	1
2. RPS (+5V)-----	1

**PROGRAM:**

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

**OUTPUT:**

[illegible]

### 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----- 1

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

**PROGRAM:**

**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 LOCATIO N	Data

[illegible]

### **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?

## Cycle-2



## EXECUTION PROCEDURE FOR 8051

Writing a alp program into 8051:

Switch on kit

Press reset

Press A (give starting address)

Press enter

Enter 1<sup>st</sup> 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

**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

**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

**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

**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

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
		<b>LOOP1</b>	INC DPTR
			MOVX @DPTR,A
			INC DPTR
			MOV A,R6
			MOVX @DPTR,A
			INC DPTR
			MOV A,R2
		<b>LOOP2:</b>	MOVX @DPTR,A
			SJMP LOOP2
			LCALL 03

**OUTPUT:**

Input		output	
MEMORY LOCATION	Data	MEMORY LOCATION	Data

**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

**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

**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
MOVBX    A, @DPTR
SETB     0E5
INC      DPTR
MOVBX    @DPTR, A

```

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

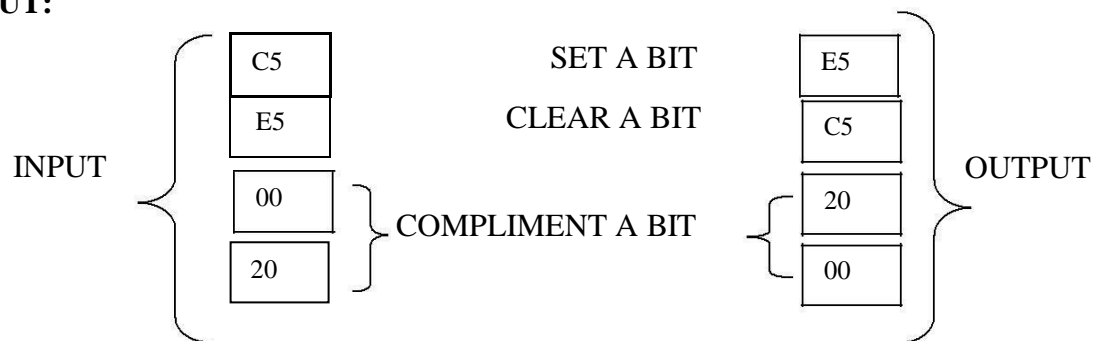
#### RESET A BIT:

```
      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 OPERATIONS:

##### SWAP A BYTE:

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



```

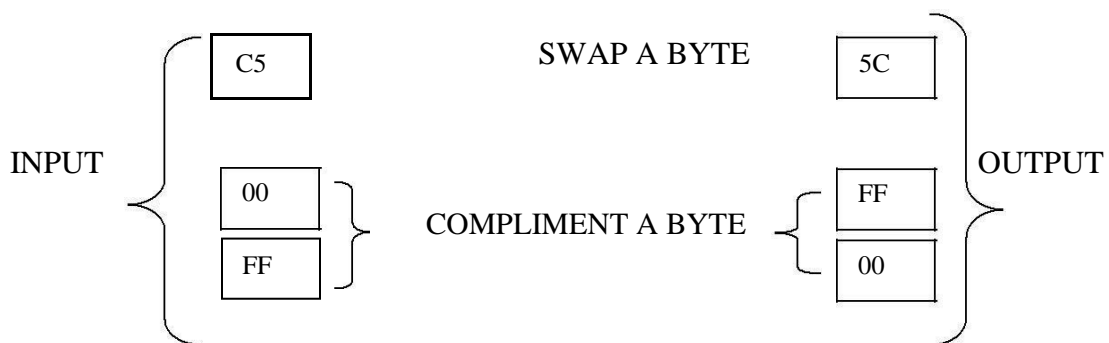
MOVX    A, @DPTR
SETB    A
INC      DPTR
MOVX    @DPTR, A
L4:     SJMP    L4 (OFFSET ADDRESS)
        LCALL    03
    
```

**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:**

**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

## 2: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		<b>LOOP</b>	MOV A,TMOD (TMOD=89) ORL A,#05H MOV TMOD,A SETB TRO (TRO=8C) LCALL 68EAH 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		<b>LOOP</b>	MOV A, TMOD (TMOD=89) ORL A,#50H MOV TMOD,A SETB TR1 (TR1=8E) LCALL 68EAH 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

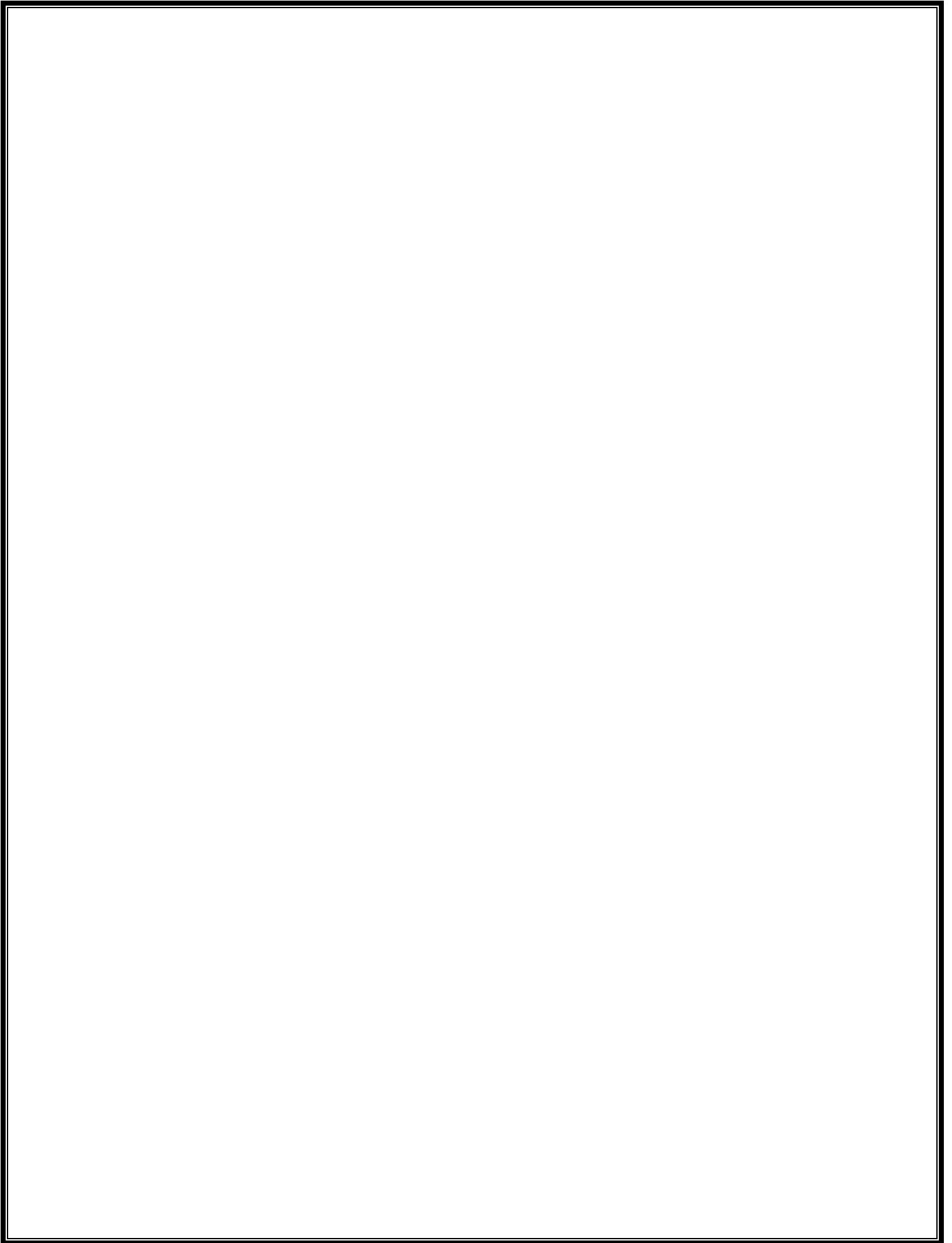
Execution: 1) short jp1 of 5&6 pins and press sw2 for manual increment  
 2) Short jp2 of 4&5 pins for auto increment

**RESULT:****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



## Cycle-3

## 2: 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
		<b>BLINK2</b>	MOV A,#90H MOV DPTR,#CNTRL MOVX @DPTR,A MOV B,#20H 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
		<b>BACK</b>	MOV A,#FEH

			<b>BLINK1</b>  MOV B,#21H 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  <b>DELAY:</b> MOV R0,#F7H <b>OLOOP:</b> MOV R1, #FFH <b>ILOOP:</b> DJNZ R1,ILOOP DJNZ R0,OLOOP RET	
--	--	--	---	--

**RESULT:**

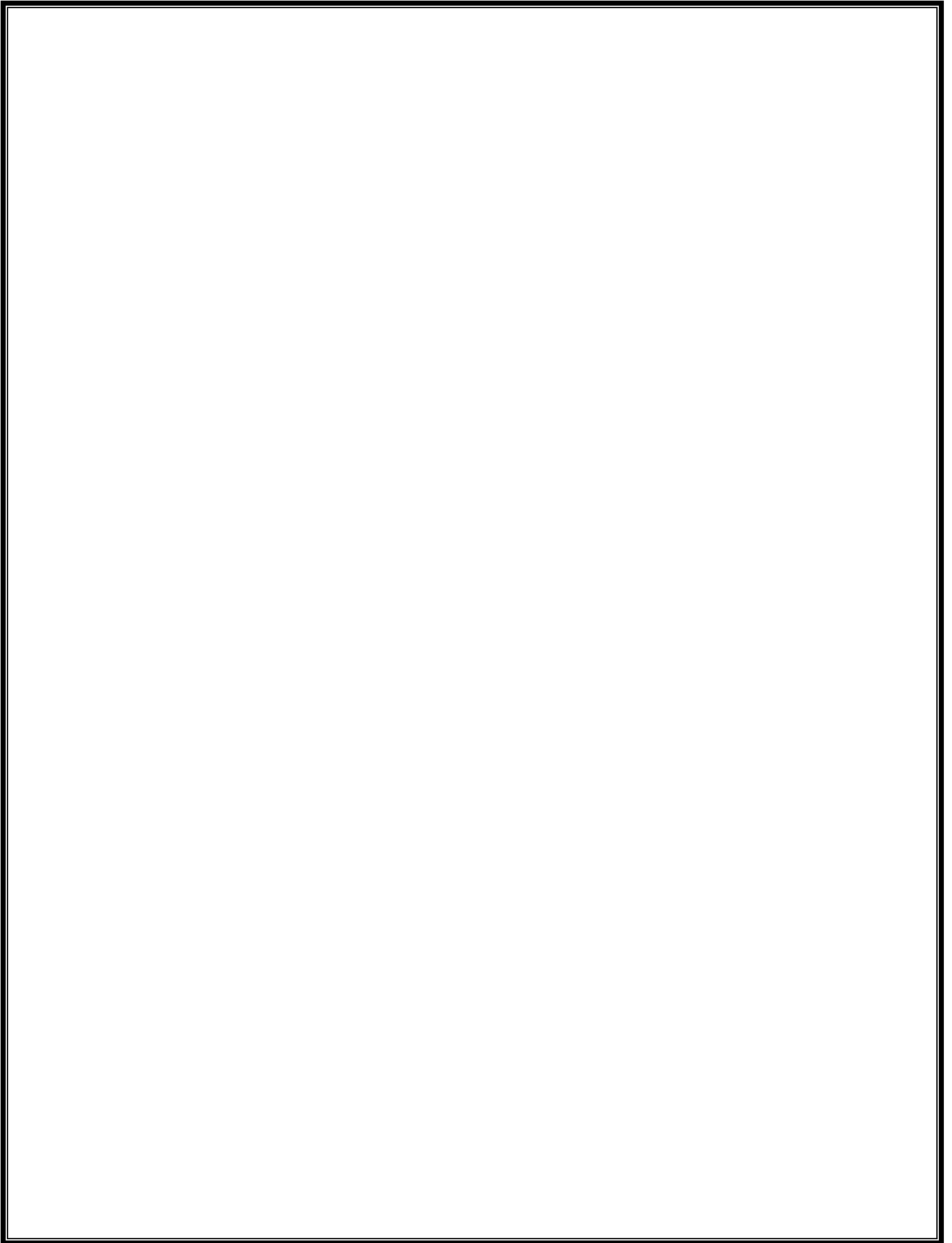
**Viva:**

- 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



**ADDITIONAL EXPERIMENT:**

**PROGRAM 5**

**TONE GENERATOR INTERFACING TO 8086**

**OBJECTIVE:** Write a program in ALP to interface stepper motor to 8086 and rotate it in clockwise direction/anti-clockwise direction.

**APPARATUS:**

1. ADS-SDA-86-STA kit
2. Tone generator interface card
3. 1 Amp Power Supply.
4. Adapter, Keyboard, Cables, Connecting Wires Etc . . .

**PROCEDURE:**

1. Connect 8086 kit PC using RS232 cable.
2. Connect Power supply to 8086 kit
3. Connect 1Amp Power Supply to tone generator
4. Connect clock1 to JP1 of 25 pin and clock 2 to 3rd pin of JP2.
5. Connect FRC cable from connector 3 to study card.
6. Keep the DIP switch in 1 & 7 on (8086kit), open TALK, and go to options select target device as 8086 and Connect.
7. Change dip switch into 1 & 5on, once reset 8086 kit.
8. Go to file →Download hex file-> NIFC 19
9. Keep the DIP switch in 1 & 7 on (8086kit)
10. G-7000(on kit keyboard), now the LED'S will glow.

**PROGRAM**

output 2500AD

;program to test NIFC19( TONE GENERATOR )interface using nifc19.

DSEG                SEGMENT

                    ORG 0000:4000H

DSEG                ENDS

code    segment at 0:4500H

         assume cs:code,ds:DSEG

porta	equ	ffc0h	;porta address of 8255
portb	equ	ffc2h	;portb address of 8255
portc	equ	ffc4h	;portc address of 8255
cntrl	equ	ffc6h	;control port address
of 8255			

```

timer2 equ    ffc dh
timer_cntrl equ ffc fh
timer2_mode equ                                b6h
SA0_COUNT    equ                                1DDCh
RI0_COUNT    equ                                1A5Ah
GA0_COUNT    equ                                1764h
MA0_COUNT    equ                                165Eh
PA0_COUNT    equ                                13EEh
DA0_COUNT    equ                                11C1h
NI0_COUNT    equ                                0FD1h
SA1_COUNT    equ                                0EF0H
RI1_COUNT    equ                                3407D
GA1_COUNT    equ                                3034D
MA1_COUNT    equ                                2865D
PA1_COUNT    equ                                2551D
DA1_COUNT    equ                                2275D
NI1_COUNT    equ                                2024D
SA2_COUNT    equ                                1912D
org 0:5000    mov     al,80h    ;all 8255 ports o/p ports
               mov     dx,cntrl
               out     dx,al
               mov     al,timer2_mode
               mov     dx,timer_cntrl
               out     dx,al
               first:
               mov     al,00
               mov     dx,portc
               out     dx,al
               mov     ax,SA0_COUNT
               mov     dx,timer2
               out     dx,al

```

mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
mov	ax,RI0_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
mov	ax,GA0_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al

mov	ax,MA0_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
mov	ax,PA0_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
mov	ax,DA0_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay

mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
mov	ax,NI0_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
mov	ax,SA1_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	ax,SA1_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al

mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
mov	ax,NI0_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
mov	ax,DA0_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al



mov	ax,PA0_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
mov	ax,MA0_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
mov	ax,GA0_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay

mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
mov	ax,RI0_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
mov	ax,SA0_COUNT
mov	dx,timer2
out	dx,al
mov	al,ah
out	dx,al
mov	al,80h ;enable AND gate
mov	dx,portc
out	dx,al
call	delay
mov	al,00h ;disable AND gate
mov	dx,portc
out	dx,al
call	delay

```

    jmp firstdelay: mov             bl,05
lp0:             mov             cx,ffffh
lp:             loop             lp
                dec             bl
                jnz             lp0
                ret
                code             ends

```

end

### **VIVA QUESTIONS:**

1. What is 8253 ?
2. Define the modes of 8253?
3. Explain the necessity of each counter.
4. What is the purpose of Timer/counter.
5. Which timer is used to set the baud rate for serial communication in 8051?

### **Outcome:**

Upon completion of this experiment the student will be able to:

- 1.Demonstrate and Tone generator , how the 8086 microprocessor is interfaced.
- 2.Design 8086 microprocessor system by using the peripheral devices
3. Interpret the data transfer from 8086 microprocessor to the peripheral device and vice versa

**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.

## **Department of Electronics and communication**

**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.