MICROPROCESSORS & MICROCONTROLLERS

LAB MANUAL

Subject Code : A60494
Regulations : R15 – JNTUH
Class : III Year II Semester (ECE)
IV Year I Semester (EEE)

Prepared by

Mr.N Papa Rao
Assistant Professor

Mrs.C.Deepthi
Assistant Professor

Mr.S.Lakshmana Chari
Assistant Professor

Mr.M.Lakshmi Raviteja
Assistant Professor

Department of Electronics & Communication Engineering
INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal – 500 043, Hyderabad

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad

Electronics & Communication Engineering

Vision
To produce professionally competent Electronics and Communication Engineers capable of effectively and efficiently addressing the technical challenges with social responsibility.

Mission
The mission of the Department is to provide an academic environment that will ensure high quality education, training and research by keeping the students abreast of latest developments in the field of Electronics and Communication Engineering aimed at promoting employability, leadership qualities with humanity, ethics, research aptitude and team spirit.

Quality Policy
Our policy is to nurture and build diligent and dedicated community of engineers providing a professional and unprejudiced environment, thus justifying the purpose of teaching and satisfying the stake holders. A team of well qualified and experienced professionals ensure quality education with its practical application in all areas of the Institute.

Philosophy
The essence of learning lies in pursuing the truth that liberates one from the darkness of ignorance and Institute of Aeronautical Engineering firmly believes that education is for liberation. Contained therein is the notion that engineering education includes all fields of science that plays a pivotal role in the development of world-wide community contributing to the progress of civilization. This institute, adhering to the above understanding, is committed to the development of science and technology in congruence with the natural environs. It lays great emphasis on intensive research and education that blends professional skills and high moral standards with a sense of individuality and humanity. We thus promote ties with local communities and encourage transnational interactions in order to be socially accountable. This accelerates the process of transfiguring the students into complete human beings making the learning process relevant to life, instilling in them a sense of courtesy and responsibility.

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

ELECTRONICS AND COMMUNICATION ENGINEERING

Program Outcomes	
PO1	**Engineering knowledge**: An ability to apply knowledge of basic sciences, mathematical skills, engineering and technology to solve complex electronics and communication engineering problems (**Fundamental Engineering Analysis Skills**).
PO2	**Problem analysis**: An ability to identify, formulate and analyze engineering problems using knowledge of Basic Mathematics and Engineering Sciences. (**Engineering Problem Solving Skills**).
PO3	**Design/development of solutions**: An ability to provide solution and to design Electronics and Communication Systems as per social needs**(Social Awareness)**
PO4	**Conduct investigations of complex problems**: An ability to investigate the problems in Electronics and Communication field and develop suitable solutions (**Creative Skills**).
PO5	**Modern tool usage** An ability to use latest hardware and software tools to solve complex engineering problems (**Software and Hardware Interface**).
PO6	**The engineer and society**: An ability to apply knowledge of contemporary issues like health, Safety and legal which influences engineering design (**Social Awareness**).
PO7	**Environment and sustainability** An ability to have awareness on society and environment for sustainable solutions to Electronics & Communication Engineering problems(**Social awareness**).
PO8	**Ethics**: An ability to demonstrate understanding of professional and ethical responsibilities(**Engineering impact assessment skills**).
PO9	**Individual and team work**: An ability to work efficiently as an individual and in multidisciplinary teams(**Team Work**).
PO10	**Communication**: An ability to communicate effectively and efficiently both in verbal and written form(**Communication Skills**).
PO11	**Project management and finance**: An ability to develop confidence to pursue higher education and for life-long learning(**Continuing education awareness**).
PO12	**Life-long learning**: An ability to design, implement and manage the electronic projects for real world applications with optimum financial resources**(Practical engineering analysis skills**).
Program Specific Outcomes	
PSO1	**Professional Skills:** An ability to understand the basic concepts in Electronics & Communication Engineering and to apply them to various areas, like Electronics, Communications, Signal processing, VLSI, Embedded systems etc., in the design and implementation of complex systems.
PSO2	**Problem-solving skills:** An ability to solve complex Electronics and communication Engineering problems, using latest hardware and software tools, along with analytical skills to arrive cost effective and appropriate solutions.
PSO3	**Successful career and Entrepreneurship:** An understanding of social-awareness & environmental-wisdom along with ethical responsibility to have a successful career and to sustain passion and zeal for real-world applications using optimal resources as an Entrepreneur.

ATTAINMENT OF PROGRAM OUTCOMES & PROGRAM SPECIFIC OUTCOMES			
S.No.	**Experiment**	**Program Outcomes Attained**	**Program Specific Outcomes Attained**
1	Write an ALP program using 8086 & MASM and verify for : a) Addition of two 16-bit numbers and multibyte addition b) Subtraction of two 16 bit numbers and multibyte subtraction c) Multiplication of two 16 bit numbers d) Division of two 16 bit numbers	PO1, PO2	PSO1
2	Write an ALP program using 8086 to sort the given numbers a) program to sort the given numbers in ascending order b) program to sort the given numbers in descending order	PO1, PO2	PSO1
3	Write an ALP program using 8086 & MASM program for searching for a number or character in a string a) To search a number or character from a string.	PO1, PO2	PSO1, PSO2
4	Write an ALP program using 8086 & MASM program for string manipulations a) Program for transfer block of data from one memory location to another memory location. b) Program for reverse of a given string c) Program for insert a new byte in a given string d) Program for delete a byte in a given string	PO1, PO2	PSO1
5	Write an ALP program for code conversions using 8086 a) To write a program for conversion of analog data to digital output b) 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	PO1, PO2, PO5	PSO1, PSO2
6	Write an ALP program to interface stepper motor with 8086 and rotate in clock wise and as well as anti clock wise direction	PO1, PO2, PO5	PSO1
7	Write an ALP program using 8051 and MASM & perform arithmetic, logical and bit manipulation instructions a) To perform 8 bit arithmetical operations by using 8051. b) Logical operations	PO1, PO2	PSO1
8	Write an ALP program using 8051and MASM & verify timer/counter in 8051	PO1, PO2	PSO1
9	Write an ALP program using 8051and MASM & verify interrupt handling	PO1, PO2, PO5	PSO1
10	Write an ALP program using 8051and MASM to perform UART operation a) Program for mode-0-transmitter b) Program for mode-0-reciever:	PO1, PO2, PO5	PSO1, PSO2
11	Write an ALP program to interface LCD with 8051	PO1, PO2, PO4, PO5	PSO2
12	Write an ALP program to interface MATRIX/Keyboard with 8051	PO1, PO2, PO5	PSO2

ATTAINMENT OF PROGRAM OUTCOMES & PROGRAM SPECIFIC OUTCOMES			
S.No.	**Experiment**	**Program Outcomes Attained**	**Program Specific Outcomes Attained**
Content Beyond Syllabi			
1	To write an assembly language program to find LCM of a given number using 8086	PO1, PO2	PSO1
2	To write an assembly language program to find square and cube of a number using 8086	PO1, PO2	PSO1
3	To write an alp for parallel communication between two microprocessors by using 8255	PO1, PO2	PSO1, PSO2
4	To write an alp for serial communication between two microprocessors by using 8255	PO1,PO2	PSO1,PSO2
5	To Write an ALP program to Interface an 8051 microcontroller trainer kit to pc and establish a communication between them through RS 232	PO1,PO5	PSO1,PSO2

MICROPROCESSORS & MICROCONTROLLERS LABORATORY

Course Overview:

This lab provides an in-depth understanding of the operation of microprocessors and microcontrollers, machine language programming & interfacing techniques. This course is designed to impart professional training to the students engineering, to interface and build microprocessors and Microcontroller based applications involving interfacing of 8255 with 8086 and serial communication. The objective of this course is to teach students design and interfacing of microcontroller-based embedded systems

Course Out Comes:

1. **Design** and analyze the the assembly level programming
2. **Identity** the assembly level programming in given problem.
3. **Understand** the applications of Microprocessors and Microcontrollers.
4. **Choose** the appropriate programming level for a specified application.
5. **Understand** the basic programming knowledge on processor and controller
6. **Understand** and develop assembly language programming with various applications
7. **Compare** different implementations and designing with interfacing circuits
8. **Write** complex applications using Assembly language programming methods.

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad

MICROPROCESSORS & MICROCONTROLLERS LAB SYLLABUS

S. No.	List of Experiments	Page No.	Date	Remarks
1	Study the architecture of 8086 & 8051 familiarization with its hardware, commands & operation of microprocessor,	25		
2	Programs for 16 bit arithmetic operations for 8086 (using Various Addressing Modes).	34		
3	Program for sorting an array for 8086.	39		
4	Program for searching for a number or character in a string for 8086	42		
5	Program for string manipulations for 8086	49		
6	Interfacing ADC and DAC to 8086	56		
7	Interfacing to 8086 and programming to control stepper motor	60		
8	Programming using arithmetic, logical and bit manipulation instructions of 8051.	67		
9	Program and verify Timer/Counter in 8051	71		
10	Program and verify Interrupt handling in 8051	76		
11	UART Operation in 8051	80		
12	Interfacing LCD to 8051	91		
13	Interfacing matrix/keyboard to 8051	95		
Content Beyond Syllabi				
1	Program to find LCM of a given number using 8086	98		
2	Arithmetic programs to find square and cube using 8086	100		
3	Program for Parallel communication between two microprocessors using 8255	103		
4	Program for Serial communication between two microprocessors using 8255	107		
5	Program for communication between 8051 kit and PC			

*****Content beyond the university prescribed syllabi

I. INTRODUCTION:

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.

1 SPECIFICATIONS:

CPU: Intel 8086 operating at 5 MHz 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

SERIAL I/O: One RS-232C compatible interface Using USART 8251A

TIMER: Three 16 bit counter / timers 8253ACounter 1 is used for serial I/O Baud rate generation.

PIC: Programmable Interrupt controller using 8253A provides interrupts Vectors for 8 jumpers' selectable Internal /External sources.

2 KEYBOARD / DISPLAY:

Keyboard: Computer keyboard can be hocked on to the trainer.

Display: LCD 2x16 display.

3 INTERRUPTS:

NIM: Provision for connecting NMI to a key switch

INTR: Programmable Interrupt controller using 8259A provides Interrupt vectors for 8 jumpers selectable Internal/ External Sources.

4 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.

5 MONITOR SOFTWARE:

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

6 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.

7 I/O decoding:

IC U30 is used for on card I/O decoding. The following table gives the list of on card I/O devices and their address map.

I/O device	I/O address	I/O register	usage
8255 I (U14)	FFCO	PORT A	AVAILABLE TO USER
	FFC2	PORT B	
	FFC4	PORT C	
	FFC6	CONTROL PORT	
8255 II (U15)	FFC1	PORT A	AVAILABLE TO USER
	FFC3	PORT B	
	FFC5	PORT C	
	FFC7	CONTROL PORT	
8253 A(U28)	FFC9	TIMER 0	AVAILABLE TO USER
	FFCB	TIMER 1	USED FOR BAUD RATE
	FFCD	TIMER 2	AVAILABLE TO USER
	FFCF	CONTROL	AVAILABLE TO USER
8251A (U13)	FFD0	DATA COMMAND PORT STATUS	
	FFD2		
INPUT PORT TO DIP SWITCH (SW1)		USED AS I/P PORT TO READ SW1 AND CONFIGURE 86ME	
8259A (U12)	FFD8 TO FFDE	PRIORITY INTERRUPT CONTROLLER	

8 POWER REQUIREMENTS:

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

9 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.

2.0 EXECUTION PROCEDURE FOR 8086 (for registers):

i) Writing a alp PROGRAM into processor:

Switch On Power Supply

Press A

A

SG 0(Zero)

Press enter

DA starting address

Press enter

Press N

Then display shows @ here you have to write 1st MNEMONIC OPERAND

Press enter

N (display displays address op- code MNEMONIC OPERAND)

N

Then display shows @ here you have to write 2nd MNEMONIC OPERAND

Press enter

Press N

Press N

-

-

-

Then display shows @ up to last MNEMONIC OPERAND

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):

R

E then it shows AX register then press, symbol it shows BX register then press, symbol it shows CX register.

2.1 EXECUTION PROCEDURE FOR 8086 (for memory locations):

i) Writing a alp 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 1stMNEMONIC OPERAND

Press enter

N (display displays address op-code MNEMONIC OPERAND)

N

Then display shows @ here you have to write 2nd MNEMONIC OPERAND

Press enter

N

N

-

Then display shows @ up to last MNEMONIC OPERAND

Press enter

N

N

! Press enter

EX press enter

E (exam byte)

Here you have to type SI address, give 1st 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 1st 8 bit data For 2nd 8 bit data again press, -------, nth data

2.2 Introduction to MASM:

MASM: (Microsoft assembler)

Run command prompt and go to Masm directory

i.e. C:\masm

Type the program by opening an editor using Edit command

i.e. C:\masm\edit filename.asm

After typing the program assemble the program using masm command.

i.e. C:\masm\masm filename.asm;

After assembling, link the file using link command

i.e. C:\masm\link filename.obj;

Finally use debug command to execute the program.

C:\masm\debug filename .exe

-t; for single step execution

-g; for at a time execution

-I; for restarting the program execution

-d; to see the data segment

-q; to quit the execution

C:\masm\afdebug filename .exe

F1; for single step execution

g; for at a time execution

L filename .exe; to reload the program

Quit; to come out of the execute screen

3.0 INTRODUCTION OF ALS SDA 51-MEL:

The Intel's family of 8bit single chip microcontroller has become very popular because of their unique and powerful instruction set, architecture and over all philosophy. The 8051 family has three members: 8031,8051 and 8751.the 8031 have no on-chip program memory execution is from external program memory. The 8051 has 4k bytes of factory masked ROM and has the 8751 has 4k bytes of EPROM.

The SDA 51-MEL is a System Design Aid for learning the operation of these Microcontroller devices. It uses 8031/51 as the controller. It is designed to assist students and engineers in learning about the architecture and programming of 8031/51 and designing around this Microcontroller.

The address and data bus controllers separate the 8051 microcontroller multiplexed address/data bus, creating a 16 bit address bus and 8bit data bus.

The monitor program for the SDA 51-MEL is contained in 32kbytes EPROM. The monitor interacts with the user through a CRT terminal host computer system connected through serial I/O interface or through the PC Keyboard (AT) and 16X2 LCD display.

3.1 SPECIFICTIONS

CPU: 8051 operating at 11.0592MHZ

MEMORY: **EPROM1**-one JEDEC compatible 28 pin socket to provide up to 32Kbytememory using 27256 with monitor software.

EPROM2-optional-canbe used as program memory, if ram is configured as data only.

RAM1-one JEDEC compatible 28 pin socket to provide up to 32Kbytes of Data memory using 62256.

RAM2-one JEDEC compatible 28 pin socket to provide up to 32Kbytes Program/data or data memory.

I/O PARALLEL: 48 I/O lines using two 8255, terminated in two 26 headers.

I/O SERAL: One RS232 compatible interface, using one chip UART lines. The lines Are terminated in a 9-pin D-type female connector.onchip UART lines are also terminated in a 10 pin FRC connector.

TIMER: Three 16 bit counter/timer using 8253programmable timers terminated in a 20pin berg stick.

KEYBOARD: EXTERNAL PC –AT keyboard

DISPLAY: Alpha numeric LCD module (2linex 16 CHARS)

BUS SIGNALS: All address data and control signals are terminated in a 50 pin header Connector for user expansion. Controller specific lines like port lines T0,T1, INT1 etc are terminated in this connector.

MONITOR SOFTWARE: 32Kbytes of user of user friendly monitor software (27256) that allows Program enter, verification, debugging and execution from the system keyboard or a CRT Terminal or a PC functioning as a terminal. File uploading/downloading option is in serial mode

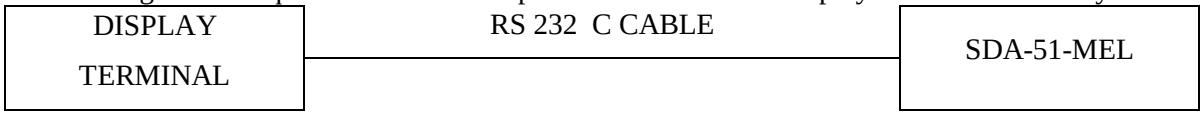
THE EXTERNAL PC: AT keyboard allows users to directly assemble /disassemble mnemonics/instructions for 8051 using the alphanumeric LCD display

OPERATING CONFIGURATION

Two different modes of Operation SDA -51MEL are possible. They are

- **serial operation**

This configuration requires an RS232 compatible terminal as the display and command entry device.



A computer system is interfaced to SDA51-MEL, a driver program must be resident in the computer system. Driver program (b30drv for DOS,TALK setup for windows) for interfacing SDA-51 MEL to a PC has been developed by ALS and is available to the user as an optional accessory.

Keyboard Operation

This mode of operation is achieved through on board KEYBOARD/DISPLAY. In this mode,SDA-51 MEL interacts with the user through an PC/AT Keyboard and a 16x2 alphanumeric LCD display. This eliminates the need for a terminal / host computer and offers a convenient way for using the SDA-51 MEL as a "STAND –ALONE" system.

3.2 SERIAL MODE:

SERIAL COMMUNICATION AND SERIAL UTILITIES

OPERATING INSTRUCTIONS

To invoke this mode press < RES> and then the < ESLR> key on the KEYBOARD to transfer control to the CRT terminal/HOST computer the prompt "SERIAL MODE" will be appears on LCD DISPLAY > ALS 8051/31 MONITER V1.0 is displayed on the terminal to indicate that the system interrogation mode and ready to accept the command. All command that be entered through interrogation modes.

SYSTEM MONITER

The SDA-51-MEL operation is controlled by monitor program stored in 32kbytes of EPROM (U5, 27256), located at SDA-51 MEL memory map (0000-7FFF).The system executes the monitor program when ever power is turn ON or when RESET is pressed.

In serial mode, the monitor program allows the user to perform following operations,

- Communicate with the SDA-51-MEL through the CRT terminal/HOST computer, using the on board serial I/O interface.
- Executes user programmers in real time or single step.
- SET break points on program,
- Examine and modify memory locations, registers and bits in SDA-51-MEL on board program/data memory and in the 8051's on chip data and register memories.
- Upload and download programmers from host computer system like PC/XT/AT(in INTEL HEX FORMAT only)

COMMANDS AVAILABLE

HELP

Syntax: H

Gives the details of the commands used in serial mode of communication**.**

DISPLAY COMMAND

The command is used to display the contents of register, bit memory, internal memory, program memory and external data memory

Syntax: D

On entering this command at the monitor command prompt, the following options are displayed.

DISPLAY(R, B, M, P, D)

The options are,

R for Registers,

B for bit memory

M for Internal memory

P for program memory

D for data memory

Press 'enter' to terminate the command.

EDIT COMMANDS

This command is used to edit the contents of register, bit memory, internal memory, program memory and the external data memory.

Syntax: E

On entering the command letter at the monitor command the following options are displayed.

EDIT (R, B, M, P and D).

The options are,

R for registers

P for program memory

B for bit memory

M for internal memory

D for data memory

During editing, the following keys can be used.

P to display the previous location **N** or space bar to display the next location **CR** to update and display the same location.

All other keys expect 0 to 9 and A to F can be used to abort the command.

3.3 PROGRAM EXECUTION COMMANDS:

The following commands are used to control the execution of user programs. The B and C commands set and clear breakpoint address. The GO and Step commands cause the system to enter execution mode from interrogation mode.

G command:

The **G** command initiates program execution at real time (12MHZ crystal, 1micro-second cycle). The real-time execution mode allows the user to run the user code stored in program memory. Execution begins when the user enters a go command in interrogation mode. Real-time execution can be controlled by breakpoints set by the user. If program halts after executing the instruction that contained the breakpoints address, then it returns to the interrogation mode .if the breakpoints are not used, the program runs until the user terminates execution with a call to the address 0003H.

The different formants of this command and their functions are described below.

8051>G

Enter start address: 8000

This command begins real time execution of the user program beginning with the instruction currently addressed by the program counter. During program execution, the following message is displayed on the screen:

3.4 PROGRAM EXECUTION:

Execution continues until one of the following occurs:

A break point is encountered (applies only when breakpoints are enabled)

The program attempts to execute across location 0003H.this location is reserved for system operation.

After execution if break point were not specified, then all the register contents will be displayed and the monitor comes back to interrogation mode with the prompt '8051>' meaning that the it is ready to a accept the next command

Note:

- The system uses the current program counter address as the start address.
- It program breakpoint or data breakpoint have been enabled then the program will be executed the command is terminated without execution of the program.

SINGLE STEL COMMAND:

This command executes one instruction at the address in the program counter

8051>S

8051>enter star address=8000<CR>

After each instruction, the system displays the values of the updated program counter, accumulator, data, pointer register, and stack pointer. To terminate this command press ESC or SPACE BAR. The actual format & the output of each of the instruction is given in the section serial communication demo

BREAK COMMAND

SET BREAK COMMAND:

SYNTAX8051<B>

Set breakpoint: up to eight breaks can be set in the user program. After giving the command 'B' at the prompted with the break number, enter the break no between 1 to 8.press <CR> after the break no. And enter the break address and press <CR>to go to conform the address and press another<CR>to go to the next break address selection or <SP><SP>to terminate the command.

CLEAR BREAKPOINTS

SYNTAX :> C

This command prompts the user for the break no, which has to be cleared. To clear all break points, enter the break number has to be cleared.

FILE UPLOAD FROM SDA-MEL TO PC

This option allows the user to save any program in memory as file in Intel hex format. On entering the command 'F10' and select option 4 on following this, the driver program prompts for the name of the file in which the data is to be stored and enter the START & END address and press,<CR>.the program assumes a default extension of HEX for the file. This system then receives the data and stores it in the specified file and on completion the main menu will be displayed.

Ex: F10

Select option 4

Enter the file name in which the data is to be stored.

Enter START address = 8000 <CR>

ENTER END address = 805F <CR>

FILE DOWNLOAD FROM PC TO SDA-EL-MEL

This option allows the user to transfer an Intel hex file on a floppy diskette to program/data memory. On processing ':' key, the following message is displayed.

Go to the main menu by pressing F10 and select option 3

On following this, the driver program prompts for the name of the file to be downloading. Enter the file name and press <CR>. While the transfer operation in progress, the system displays the number record be transferred.

At the end of the transfer the main menu is displayed. Go to terminal mode press <CR>, the following message

File received O.K. will be displayed

Ex:':'

Go to the main menu by pressing F10 and select option 3

KEY BOARD MODE OF OPERATION

At the power on the monitor automatically goes into keyboard mode, at power on the sign on message SDA 51/31/-STA<E> HELP appears on the LCD display.

THE FUNCTION OF SOME SPECIAL KEYS ON THE PC/AT KEYBOARD ARE LISTED BELOW

KEY LABEL	DESCRIPTION
RESET	Transfers control to the monitor at location 0000H
NXT	The monitor interrupts this key as a delimiter. Different commands are explained later .
ENTER	The monitor command terminator
BMOVE	Selects the monitor block move command
GO	Selects the monitor go command (program execution)
PREV	A monitor delimiter key, and in the next coming section its usage's are explained
STEP	Selects the monitor single step function
EREG	Selects the monitor examine / modify cpu register function
EDM	In combination with substitute memory command this key allows the using to examine and modify external data memory
IDM	In combination with substitute memory command this key allows the using to examine and modify internal data memory
IBM	In combination with substitute memory command this key allows the using to examine and modify internal bit memory
EPM	In combination with substitute memory command this key allows the using to examine external program memory
EPGM	Used to program EPROM's using EPROM programmer I/F(NIFC 03)
EPRD	Used to read the EPROm contents using EPROM programmer I/F(NIFC 03)
ESRL	Key to invoke serial mode
ASM	Key to invoke assembler mode
DSM	Key to invoke di assembler mode
BS	Provides back facility in assembler mode

SUBSTITUTE MEMORY COMMAND

This command is used to examine/modify the memory functions. This command wills support examine/modification of following memories.

- External data memory (EDM)
- External program memory(EPM)
- internal data memory(IDM)
- internal bit memory(IBM)

This command is invoked using 'SMEM' key in the ASCII key board the message "SUBSITUTE MEMORY" appears on the display.

Then user can select any one of above mention four memories, and enter the location address to be Examine/modify and press <NXT> to display the data present in that memory location, now user can modify that data byte if required then again he has to press <NXT>, now PC is incremented to show the contents of the next memory location. If the user wants to see the content of previous location i.e. if 9005H is the current PC content &he wants to see the 9004 location content then he has to press<PREV>key.

EXTERNAL DATA MEMORY

<SMEM><EDM><address of memory location><NXT><new byte if required><NXT>…..<ENTER>.

This command is used to enter the data in data memory (0300H to 1FFFH,4000H to 7FFFH) or data/code into data/program memory (8000H to FFFFH).

INTERNAL DATA MEMORY

<SMEM><IDM><Address of the memory location><NXT><new byte if required><NXT>…...<ENTER>

Internal data memory ranges from 00H to 7FH(128bytes)

INTERNAL BIT MEMORY

<SMEM><IBM><Address of the memory location><NXT><new byte if required><NXT>…...<ENTER>

Internal bit memory ranges from 00 to 7F(128bits) values entered must be 1 or 0 only.

EXTERNAL PROGRAM MEMORY

<SMEM><EPM><Address of the memory location><NXT<NXT>…<ENTER>

If the user attempts to edit data in this region an ERROR message will be displayed.

EXAMINE/MODIFY CPU REGISTERS COMMAND

The examine/modify register command allows the user to examine/modify the contents of CPU registers. This command is invoked using EREG key in the ASCII keyboard, the message "which register?"appears on the first line of LCD display then the user can select the CPU register which he wants to examine/modify through a key designator (for the key designators see the table given below) then if <NXT> pressed the register name in the registers sequence and its content will be displayed, the registers display sequence if A,B,R0,R1,R2,R3,R4,R5,R6,R7,PCL,PCH,PSW,SP,DPH,DPL.

DESIGNATOR(KEY)	CPU REGISTERS	DESIGNATIOR(KEY)	CPU REGSITERS
0	RO	8	PCL
1	R1	9	PCH
2	R2	A	A
3	R3	B	B
4	R4	C	SP
5	R5	D	DPH
6	R6	E	DPL
7	R7	F	PSW

EXECUTE USER PROGRAM COMMAND

The execute user program command allows user to execute a program that he has entered/downloaded. To invoke this execute user program command press <GO> now the current PC and its data are displayed on the LCD display and then the command is completed when the user press<ENTER>the message "PROGRAM EXECUTED" will be displayed on the LCD display.

SYNTAX: Go<Program starting address><ENTER>

EX: To execute a program which is having the starting address at 8000H<GO>8000<ENTER>

SINGLE STEP COMMAND

The single step command allows the user to 'instruction step' through his program, this command is invoked through <STEP> key when the user press<STEP> the current PC content and data of that location are displayed on the LCD module. The user can now change the address, if required and then press <ENTER >,the instruction at that address is executed and its contents are displayed, now by pressing <NXT>key the display updates to next logical address and its contents. To examine register or memory contents at this stages press<ENTER>then <EREG>/<SMEM> or any command provided to user in keyboard mode and again to enter single step press <ENTER>and to continue the stepping process press<NXT><NXT>….

In this single step mode, we use INT0 with its priority bit set. A such the other interrupts are not functional.

SYNTAX:

<STEP><Starting address of user program><ENTER><NXT><NXT>…..

EX:To single step a program with starting address 9000H,and in the third step exam register command has to be invoked to see the content of registers A,B,R0, then again come back for single stepping.

<STEP>8000<ENTER><NXT><NXT><NXT><ENTER>

<EREG><A><NXT><NXT><NXT><ENTER><STEP><ENTER><NXT>

<NXT>…….

TALK software Procedure:

First identify Location of TALK software. If it is in D drive then choose run prompt and select CMD then follow below procedure.

D:\>

ENTER

D:\>cd comm_pack86

ENTER

D:\cd comm_pack86 >cd comm_pack86

Enter

D:\cd comm_pack86 >cd comm_pack86>cd x8086

Enter

D:\cd comm_pack86 >cd comm_pack86>cd x8086>edit file name

Enter

Enter the program

Go to file and save & go to file exit

Press x8086

Enter

Listing destination :d

Enter

Generate cross reference (Y/N): y

Enter

Input filename: GIVE INPUT FILE NAME.ASM

Enter

Output filename:

Enter

Link assembled: Assembled Errors:

Enter

Input filename: GIVE filename.obj

Enter

Enter offset for ‘cseg’: 0(Zero)

Enter

Input file name:

Enter

Output file name :

Options<D,S,A,M,X,H,E,T,1,2,3,<CR>=Default>:h

Enter

Exit

Enter

Next selected go to talk

Going to options in settings

Comport-com1

Bit per seconds -9600

Data bit -8

Parity –none

Stop bit -1

Flow control-none

Transfer mode-ASCII key

NEXT PRESS OK

Selected options in that selected target board 8086 kit

Press ok

1st selected in m.p kit as keep 1 and 5 pins ON

Then go to options disconnected and connected, press reset button in kit

Display –als-86 monitor

Go file selected download Intel hex. File<comm._pack86>,<openx8086>,<filename>open

Enter

Display #

Next selected in kit 1&7 pins keeps ON and press reset button in kit

Selected in G

Give the address and press enter

EXPERIMENT No 1

16 BIT ARITHMETIC OPERATIONS FOR 8086

(USING VARIOUS ADDRESSING MODES)

1.1 AIM: -

To write an assembly language PROGRAM for Addition of two 16-bit numbers.

1.2 COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/MASM with PC		1
2	Keyboard		1
3	RPS	+5v	1 1

1.3 PROGRAM:

i) Software

```
Assume cs: code
Code segment
        Start:
                MOV AX, 4343H
                MOV BX, 1111H
                ADD AX, BX
                INT 03H
Code ends
End start
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND

			MOV AX,4343
			MOV BX,1111
			ADD AX,BX
			INT 3

Observation Table

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

ii) MULTIBYTE ADDITION

AIM: -

Program to perform multi byte addition

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/MASM with PC		1
2	Keyboard		1
3	RPS	+5v	1 1

PROGRAM:

i) Software

```
Assume cs: code
Code segment
        Start:  MOV AX,0000H
                MOV  SI,2000H
                MOV DI,3000H
                MOV BX,2008H
                MOV CL,04H
        UP :    MOV AL,[SI]
                ADD AL,[BX]
                MOV [DI],AL
                INC SI
                INC BX
                INC DI
                DEC CL
                JNZ  UP
                INT 03H
Code   ends
End start
```

ii) Hardware:

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND

			MOV AX,0000
			MOV SI, 2000
			MOV DI, 3000
			MOV BX, 2008
			MOV CL, 04
			MOV AL, [SI]
			UP ADD AL, [BX]
			MOV [DI], AL
			INC SI
			INC BX
			INC DI
			DEC CL
			JNZ UP
			INT 3

Observation Table:

Input				Output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**	**MEMORY LOCATION**	**Data**
2000		2008		3000	
2001		2009		3001	
2002		200A		3002	
2003		200B		3003	
2004					
2005					
2006					
2007					

SUBTRACTION:

i) 16 bit subtraction

AIM: -

To write an assembly language PROGRAM for subtraction of two 16-bit numbers.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/MASM with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM:

i) Software

```
Assume cs: code
Code segment
        Start:  MOV AX, 4343H
                MOV BX, 1111H
                SUB AX, BX
                INT 03H
Code ends
End start
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND

			MOV AX,4343
			MOV BX,1111
			SUB AX,BX
			INT 03

Observation Table

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

ii) MULTIBYTE SUBTRACTION

AIM: - PROGRAM to perform multi byte subtraction.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/MASM with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM:

i) Software

```
Assume cs: code
Code segment
Start:  MOV AX,0000H
        MOV  SI,2000H
        MOV DI,3000H
        MOV BX,2008H
        MOV CL,04H
UP :    MOV AL,[SI]
        SUB AL,[BX]
        MOV [DI],AL
        INC SI
        INC BX
        INC DI
        DEC CL
        JNZ  UP
        INT 03H
 Code  ends
 End start
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND

			MOV AX,0000
			MOV SI, 2000
			MOV DI, 3000
			MOV BX, 2008
			MOV CL, 04
			MOV AL, [SI]
			SUB AL, [BX]
			MOV [DI], AL
			INC SI
			INC BX
			INC DI
			DEC CL
			JNZ UP
			INT 03

		UP	

Observation Table:

Input				Output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**	**MEMORY LOCATION**	**Data**
2000		2008		3000	
2001		2009		3001	
2002		200A		3002	
2003		200B		3003	
2004					
2005					
2006					
2007					

C) MULTIPLICATION

i) 16 bit multiplication

AIM: -

To write an assembly language PROGRAM for multiplication of two 16-bit numbers.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/MASM with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM:

i) Software

```
Assume cs: code
Code segment
        Start:  MOV AX, 4343H
                MOV BX, 1111H
                MUL BX
                INT 03H
Code ends
End start
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND

			MOV AX,4343
			MOV BX,1111
			MUL BX
			INT 3

Observation Table

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.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/MASM with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM

i) Software

```
Assume cs: code
Code segment
    Start:  MOV SI,2000H
            MOV DI,3000H
            MOV AX,[SI]
            ADD SI,02H
            MOV BX,[SI]
            IMUL BX
            MOV [DI],AX
            ADD DI,02H
            MOV [DI],DX
            INT 03H
    Code ends
    End start
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND

			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

Observation Table

Input		Output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000		3000	
2001		3001	
2002		3002	
2003		3003	

D) DIVISION
i) 16 bit division

AIM:-

To write an assembly language PROGRAM for multiplication of two 16-bit numbers.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/MASM with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM

i) Software

```
Assume cs: code
Code segment
        Start:  MOV AX, 4343H
                MOV BX, 1111H
                DIV BX
                INT 03H
Code ends
End start
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND

			MOV AX,4343
			MOV BX,1111
			DIV BX
			INT 3

Observation Table

Input		Output	
Register	**Data**	**Register**	**Data**
AX	4343	AX	0003
BX	1111	DX	03F2

1.4 RESULT:

1.5 PRE LAB QUESTIONS:

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?

1.6 LAB ASSIGNMENT:

1. Write an alp program for addition and subtraction of two 16bit numbers?

   1) A 2 7 8

   2) B 6 3 4

2. Write an alp program for multiplication and division of two 16bit numbers?

   1) 0012

   2) 0006

1.7 POST LAB QUESTIONS:

1. How to move data from one register to other

2. To swapping the data what type register used

3. What are the advantages of maximum mode

EXPERIMENT No 2

2.1. ASCENDING ORDER

AIM:-

Program to sort the given numbers in ascending order

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/MASM with PC		1
2	Keyboard		1
3	RPS	+5v	1 1

PROGRAM:

i) Software

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

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND

			MOV AX, 0000
			MOV CH, 0004
			UP1: DEC CH
			MOV CL, CH
			UP: MOV SI,2000
			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 03

Observation Table:

Input		Output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000		2000	
2001		2001	
2002		2002	
2003		2003	

2.2 DESCENDING ORDER

AIM:-

Program to sort the given numbers in descending order

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
2	8086 microprocessor kit/MASM with PC		1
	Keyboard		1
3	RPS	+5v	1 1

PROGRAM:

i) Software

```
ASSUME CS: CODE
CODE SEGMENT
    START:      MOV AX,0000H
                MOV CH,0004H
                DEC CH
     UP1 :      MOV CL, CH
                MOV SI,2000H
    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
          I     NT 3
    CODE  ENDS
    END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND
			MOV AX, 0000
			MOV CH, 0004
		UP1:	DEC CH
			MOV CL,CH
		UP:	MOV SI,2000
			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

Observation Table

Input		Output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000		2000	
2001		2001	
2002		2002	
2003		2003	

RESULT:

PRE LAB QUESTIONS:

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?

LAB ASSIGNMENT:

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

POST LAB QUESTIONS:

1. How clock signal is generated in 8086
2. What is the maximum internal clock frequency of 8086?
3. What is the need for Port

EXPERIMENT No 3
PROGRAM FOR SEARCHING FOR A NUMBER OR CHARACTER IN A STRING FOR 8086

AIM:-

To Write an ALP program to search a number or character from a string.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
2	8086 microprocessor kit/MASM with PC		1
	Keyboard		1
3	RPS	+5v	1 1

i) Software

```
ASSUME CS: CODE, DS: DATA
CODE SEGMENT
        START:MOV CX, 0004H
              MOV AX,0000H
              MOV SI,2000H
              MOV BX,3000H
        UP:   MOV AL,[SI]
              CMP AL,[BX]
               JZ DOWN
              INC SI
              DEC CL
              JNZ UP
              MOV AH,00H
              JMP L3
 DOWN:        DEC CL
              MOV AH,01H
              MOV [DI], AH
        L3:   INT 3H
CODE ENDS
END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND

			MOV CX, 0004
			MOV AX,0000
			MOV SI,2000
			MOV BX,3000

		UP:	MOV AL,[SI]
			CMP AL,[BX]
			JZ DOWN
			INC SI
			DEC CL
			JNZ UP
			MOV AH,00
			JMP L3

		DOWN:	DEC CL
			MOV AH,01
			MOV [DI], AH

		L3:	INT 03

OBSERVATION TABLE:

Input		Output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000		3000	
2001			
2002			
2003			

RESULT:

PRE LAB QUESTIONS:

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?

LAB ASSIGNMENT:

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

POST LAB QUESTIONS:

1. Why crystal is a preferred clock source
2. What is Tri-state logic?
3. What happens when HLT instruction is executed in processor?

EXPERIMENT No 4

PROGRAM FOR STRING MANIPULATIONS FOR 8086

AIM:-

To write a alp for transfer block of data from one memory location to another memory location.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
2	8086 microprocessor kit/MASM with PC		1
	Keyboard		1
3	RPS	+5v	1 1

PROGRAM:

i) Software

```
ASSUME CS: CODE
CODE SEGMENT
        START:      MOV SI, 2000H
                    MOV DI, 2008H
                    MOV CX, 0008H
                    REP MOVSB
                    INT 03H
CODE ENDS
END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND

			MOV SI,2000
			MOV DI,2008
			MOV CX,0008
			REP
			MOVSB
			INT3

Observation Table

Input		Output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000		2008	
2001		2009	
2002		200A	
2003		200B	
2004		200C	
2005		200D	
2006		200E	
2007		200F	

1) REVERSE OF A DATA

AIM:-

To write a ALP for reverse of a given string

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
2	8086 microprocessor kit/MASM with PC		1
	Keyboard		1
3	RPS	+5v	1 1

PROGRAM:

i) Software

```
ASSUME CS: CODE
CODE SEGMENT
START:    MOV SI, 2000H
          MOV DI, 2008H
          MOV CX, 0008H
          ADD SI, 07H
 UP:      MOV AL,[SI]
          MOV [DI], AL
          DEC SI
          INC DI
          DEC CX
          JNZ UP
          INT 03H
CODE ENDS
END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND

			MOV SI,2000
			MOV DI,2008
			MOV CX,0008
			ADD SI,07
			UP: MOV AL,[SI]
			MOV [DI],AL
			DEC SI
			INC DI
			DEC CX
			JNZ UP
			INT 3

Observation Table

Input		Output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000		2008	
2001		2009	
2002		200A	
2003		200B	
2004		200C	
2005		200D	
2006		200E	
2007		200F	

2) INSERT A BYTE IN A GIVEN STRING

AIM:-

To write a ALP for insert a new byte in a given string

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/MASM with PC		1
2	Keyboard		1
3	RPS	+5v	1 1

PROGRAM:

i) Software

```
ASSUME CS: CODE
CODE SEGMENT
START:  MOV SI,2000H
        MOV DI,3000H
        MOV BX,5000H
        MOV CX,0005H
        CLD
L1:     MOV AL,[SI]
        CMP AL,[BX]
        JZ L2
        MOVSB
        LOOP L1
        JMP L3
L2:     MOVSB
        MOV BX,7000H
        MOV AL,[BX]
        MOV [DI],AL
        DEC CX
        INC DI
        REP MOVSB
L3:     INT 3
CODE ENDS
END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND
			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:	MOVSB
			MOV BX,7000
			MOV AL,[BX]
			MOV [DI],AL
			DEC CX
			INC DI
			REP
			MOVSB
		L3:	INT 3

Observation Table

Input		Output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000		3000	
2001		3001	
2002		3002	
2003		3003	
2004		3004	
5000		3005	
7000			

3) DELETE A BYTE IN A GIVEN STRING

AIM:-

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

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
2	8086 microprocessor kit/MASM with PC		1
	Keyboard		1
3	RPS	+5v	1 1

PROGRAM:

i) Software

```
ASSUME CS:CODE
CODE SEGMENT
START:  MOV SI,2000H
        MOV DI,3000H
        MOV BX,5000H
        MOV CX,0005H
        CLD
L1:     MOV AL,[SI]
        CMP AL,[BX]
        JZ L2
        MOVSB
        LOOP L1
        JMP L3
L2:     INC SI
        DEC CX
        REP MOVSB
L3:     INT 03H
CODE ENDS
END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND

			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

Observation Table:

Input		output	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000		3000	
2001		3001	
2002		3002	
2003		3003	
2004			
5000			

RESULT:

PRE LAB QUESTIONS:

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?

LAB ASSIGNMENT:

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?

POST LAB QUESTIONS:

1. Which interrupts are generally used for critical events?
2. Which Stack is used in 8086?
3. What is SIM and RIM instructions

EXPERIMENT NO 5
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.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/TALK with PC		1
2	Keyboard		1
3	RPS	+5v	1 1
4	A/D, D/A Interfacing modules		1
5	Power mate connector		1
6	FRC Connector,RS-232 cable		1
7	CRO		1

THEORY:

The A/D converter is treated as an input device by the microprocessor that sends an initializing signal to the ADC to start the analog to digital data conversion process. The start of conversion signal is a pulse of specific duration. After the conversion is over , ADC sends end of conversion signal to inform the microprocessor about it and result is ready at the output buffer of the ADC.

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: Gnd
Blue: +5V
Red: +12V
Green: -12V

3. After the completion of the PROGRAM and connections enter the PROGRAM as given in the listing below.
4. G0< STARTING ADDRESS< ENTER (on the key board of trainer).

D/A CONVERTER:
PROGRAM TO GENERATE SQUARE WAVE:

i) Software

```
.output 2500ad
cseg segment
        org 0000:4000h
        Assume cs: cseg
Start:
        MOV AL,80H
        MOV DX,0FFC6H
        OUT DX,AL
        MOV DX,0FFC2H
        A0:MOV AL,00H
        OUT DX,AL
        CALL DELAY 1
        MOV AL,0FFH
        OUT DX,AL
        CALL DELAY2
        JMP A0
        Delay1:MOV CX,0020H
        A1:LOOP A1
        RET
        Delay2: MOV CX,0020H
        A1:LOOP A1
        RET
Cseg ends
end
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND

			MOV AL,80
			MOV DX,0FFC6
			OUT DX
			MOV DX,0FFC2
			MOV AL,00
			OUT DX
			CALL DELAY 1
			MOV AL,0FF
			OUT DX
			CALL DELAY2
			JMP A0

DELAY PROGRAM1

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

DELAY PROGRAM2

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

PROGRAM TO GENERATE SAWTOOTH WAVE:

i) Software

```
.output 2500ad
cseg segment
                org 0000:4000h
                Assume cs: cseg
Start:
                MOV AL,80H
                MOV DX,0FFC6H
                OUT, DX
                MOV DX,0FFC2H
                L2:MOV AL,00H
                L1:OUT DX
                INC AL
                CMP AL,0FFH
                JB L1
                OUT DX
                JMP L2
Cseg ends
end
```

ii) Hardware

ADDRESS	OPCODE	LABEL	MNEMONIC OPERAND

		L2:
		L1:
			MOV AL,80
			MOV DX,0FFC6
			OUT, DX
			MOV DX,0FFC2
			MOV AL,00
			OUT DX
			INC AL
			CMP AL,0FF
			JB L1
			OUT DX
			JMP L2

PROGRAM TO GENERATE TRAINGULARWAVE:

i) Software

```
.output 2500ad
cseg segment
                org 0000:4000h
                Assume cs: cseg
Start:
                MOV AL,80H
                MOV DX,0FFC6H
                OUT DX,AL
                MOV AL,00H
                L3:MOV DX,0FFC2H
                L1:OUT DX,AL
                INC AL
                CMP AL,0FFH
                JC A2
                L2:OUT DX,AL
                DEC AL
                CMP AL,00H
                JNBE  A1
                JMP A0
Cseg ends
End
```

ii) Hardware

ADDRESS	OPCODE	LABEL	MNEMONIC OPERAND
			MOV AL,80
			MOV DX,0FFC6
			OUT DX
			MOV AL,00
		L3:	MOV DX,0FFC2
		L1:	OUT DX
			INC AL
			CMP AL,0FF
			JC L1
			OUT DX
		L2	DEC AL
			CMP AL,00
			JNBE L2
			JMP L3

A/D CONVERTER

PROGRAM:

i) Software

```
.output 2500ad
cseg segment
        org 0000:4000h
        Assume cs: cseg
Start:
        MOV AL,90H
        MOV DX,0FFC6H
        OUT DX,AL
        A0:MOV AL,07H
        MOV DX,0FFC4H
        OUT DX,AL
        MOV AL,0FH
        MOV DX,0FFC6H
        OUT DX,AL
        MOV CX,3FFFH
        D1:LOOP D1
        MOV AL,0EH
        MOV DX,0FFC6H
        OUT DX,AL
        MOV AL,0CH
        MOV DX,0FFC6H
        OUT DX,AL
        MOV DX,0FFC0H
        A1:INAL,DX
        AND AL,80H
        CMP AL,80H
        JNZ  A1
        MOV AL,0DH
        MOV DX,0FFC6H
        OUT DX,AL
        MOV DX,0FFC0H
        IN AL,DX
        MOV DX,0FFC2H
        OUT DX

        JMP  A0
Cseg ends
end
```

ii) Hardware

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

Observation Table:

INPUT	OUTPUT

RESULT:

PRE LAB QUESTIONS

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?

LAB ASSIGNMENT:

1. Using the program generate a waveform and identify that

2. Using the program generate a waveform and identify that

POST LAB QUESTIONS:

1. Which is the tool used to connect the user and the computer
2. What is the position of the Stack Pointer after the PUSH instruction
3. Logic calculations are done in which type of registers

EXPERIMENT NO 6

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.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/TALK with PC		1
2	Keyboard		1
3	RPS	+5v	1
4	Stepper motor interfacing card,RS-232		1
5	Stepper motor		1
6	FRC Connector,RS-232 cable		1

THEORY:

A stepper motor is a device used to obtain an accurate position control of rotating shafts. It employs rotation of its shaft in terms of steps, rather than continuous rotation as in case of AC or DC motors. To rotate the shaft of the stepper motor, a sequence of pulses is needed to be applied to the windings of the stepper motor, in a proper sequence. The number of pulses required for one complete rotation of the shaft of the stepper motor are equal to its number of internal teeth on its rotor.

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
VDD- 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 TO ROTATE IN CLOCKWISE DIRECTION :

i) Software

```
.output 2500ad
cseg segment
                org 0000:4000h
                Assume cs: cseg
Start:          MOV AL,80H
                MOV DX,0FFC6H
                OUT DX,AL
                MOV BX,02H
           A2:MOV CX,00FFH
           A1:MOV AL,77H
                MOV DX,0FFC4H
                OUT DX,AL
                CALL DELAY
                MOV AL,0BBH
                MOV DX,0FFC4H
                OUT DX,AL
                CALL DELAY
                MOV AL,0DDH
                MOV DX,0FFC4H
                OUT DX,AL
                CALL DELAY
                MOV AL,0EEH
                MOV DX, 0FFC4H
                OUT DX,AL
                CALL DELAY
                LOOP  A1
                DEC BX
                JNZ  A2
         Delay: MOV AX,0500H
             A3:NOP
                NOP
                DEC AX
                JNZ A3
                RET

Cseg ends
End
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
			MOV AL,80
			MOV DX,0FFC6
			OUT DX
			MOV BX,02
		A2:	MOV CX,00FF
		A1:	MOV AL,77
			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

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
			JNZ A2

DELAY PROGRAM:

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

RESULT:

PRE LAB QUESTIONS:

1. Functions of BX register?
2. Functions of CX register?
3. Functions of DX register?
4. How Physical address is generated?
5. Which are pointers present in this 8086?

LAB ASSIGNMENT:

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?

POST LAB QUESTIONS:

1. In string operations which is by default string source pointer
2. What is the size of flag register
3. Can you perform 32 bit operation with 8086? How

EXPERIMENT NO 7

PROGRAMMING USING ARITHMETIC, LOGICAL AND BIT MANIPULATION INSTRUCTIONS OF 8051

AIM:-

Write an ALP program to perform 8 bit arithmetical operations by using 8051.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8051 trainer kit with keyboard		1
2	Talk with PC		1
3	RPS	+5v	1

PROGRAM FOR ADDITION:

i) Software

```
Org 9000h
MOV A,#02
MOV B,#02
ADD A,B
LCALL 03
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND

			MOV A,#02
			MOV B,#02
			ADD A,B
			LCALL 03

OBSERVATION TABLE:

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

PROGRAM FOR SUBTRACTION:

i) Software

```
Org 9000h
MOV A,#02
MOV B,#02
SUBB A,B
LCALL 03
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND

8000			MOV A,#04
			MOV B,#02
			SUBB A,B
			LCALL 03

OBSERVATION TABLE

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

PROGRAM FOR MULTIPLICATION:

i) Software

```
Org 9000h
MOV DPTR,#9000H
MOVX A,@DPTR
MOV F0,A
INC DPTR
MOVX A,@DPTR
MUL AB
LCALL 03
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND

8000			MOV DPTR,#9000
			MOVX A,@DPTR
			MOV F0,A
			INC DPTR
			MOVX A,@DPTR
			MUL AB
			LCALL 03

Observation Table

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

PROGRAM FOR DIVISION:

i) Software

```
Org 9000h
MOV DPTR,#9000H
MOVX A,@DPTR
MOV R0,A
INC DPTR
MOVX A,@DPTR
MOV F0,A
MOV A,R0
DIV AB
INC DPTR
MOV @DPTR,A
LCALL 03
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND

			MOV DPTR,#9000
			MOVX A,@DPTR
			MOV R0,A
			INC DPTR
			MOVX A,@DPTR
			MOV F0,A
			MOV A,R0
			DIV AB
			INC DPTR
			MOV @DPTR,A
			LCALL 03

OBSERVATION TABLE:

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

LOGICAL OPERATIONS:

AIM:-

To perform logical operations by using 8051.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8051 trainer kit with keyboard		1
2	Talk with PC		1
3	RPS	+5v	1
4	RS - 232		1

PROGRAM FOR AND OPERATION:

i) Software

```
Org 9000h
MOV R0,#DATA 1
MOV A,#DATA 2
ANL A,R0
MOV R1,A
LCALL 03
```

i) Hardware

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

OBSERVATION TABLE:

Input		Output	
Register	**Data**	**Register**	**Data**
R0		R1	
A			

PROGRAM FOR OR OPERATION:

i) Software

```
Org 9000h
MOV R0,#DATA 1
MOV A,#DATA 2
ORL A,R0
MOV R1,A
LCALL 03
```

ii) Hardware

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

OBSERVATION TABLE:

Input		Output	
REGISTER	**Data**	**REGISTER**	**Data**
R0		R1	
A			

PROGRAM FOR XOR OPERATION:

i) Software

```
Org 9000h
MOV R0,#DATA 1
MOV A,#DATA 2
XRL A,R0
MOV R1,A
LCALL 03
```

ii) Hardware

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

OBSERVATION TABLE:

Input		output	
REGISTER	**Data**	**REGISTER**	**Data**
R0		R1	
A			

RESULT:

PRE LAB QUESTIONS:

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?

LAB ASSIGNMENT:

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

POST LAB QUESTIONS:

1. What do u mean by emulator
2. What is the size of flag register
3. What are ASCII codes for nos. 0 to F
4. Which no. representation system you have used

EXPERIMENT NO 8

PROGRAM AND VERIFY TIMER/COUNTER IN 8051

AIM:-

Write an ALP program to Perform Timer 0 and Timer 1 in Counter Mode and Gated Mode operation.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8051 trainer kit with keyboard		1
2	Talk with PC		1
3	RPS	+5v	1
4	RS - 232		1
5	FRC cables		1

THEORY:

The 8051 has two 16 bit timer/ counters. They can be used either as a timer or event count. Each 16 bit timer accessed as two separate registers as TL0, TL1 and TH0, TH1 bytes.

PROGRAM TO VERIFY TIMER '0'- COUNTER MODE:

i) Software

```
Org 8000h
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
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERANDS
			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:

i) Software

```
Org 8000h
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
```

ii) Hardware

Memory location	OPCODE	LABEL	MNEMONIC OPERANDS
			MOVA, TMOD (TMOD=89)
			ORL A,#50H
			MOV TMOD,A
			SETB TR1 (TR1=8E)
			LCALL 68EAH
			MOV DPTR,#0194H
			MOV A,TL1 (TL1=8B)
		LOOP:	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:

PRE LAB QUESTIONS:

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?

LAB ASSIGNMENT:

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

POST LAB QUESTIONS:

1. Why we indicate FF as 0FF in program?
2. What is a type of queue in 8086
3. While accepting no. from user why u need to subtract 30 from that?

EXPERIMENT No 9
PROGRAM AND VERIFY INTERRUPT HANDLING IN 8051

AIM:-

Write ALP program to allow the external interrupt 1 using 8051

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8051 trainer kit with keyboard		1
2	Talk with PC		1
3	RPS	+5v	1
4	RS - 232		1
5	FRC cables		1
6	Timer module kit		

THEORY:

8051 has five interrupts which are available to user. They are two from the timer/counter overflow flags, one for timer 0and one for timer 1, two interrupts from external interrupts and one from serial port. To enable or disable the corresponding interrupt it uses IE register.

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:

i) Software

```
ORG     E000H
 LCALL 68EAH
            MOV    A,TMOD
            ORL    TMOD,#10H
            MOV     TH1,#00H
```

```
        MOV   TL1,#FFH
        SETB  IT1
        SETB  ET1
        SETB  EX1
        SETB  TR1
        SETB  PX1
        MOV    R6,#26H
        SETB  EA
HERE:   AJMP  HERE
        ORG   8028H
        LJMP  8500H
        ORG   8500H
        MOV    A,R6
        SWAP  A
        MOV   R6,A
        LCALL  677DH
        MOV R0,#FFH
        MOV R1,#FFH
        LCALL  6850H
        MOV R0,#FFH
        MOV R1,#FFH
        LCALL  6850H
        MOV   TH1,#00H
        MOV   TL1,#FFH
        SETB  TR1
        RETI
        ORG   8010H
        SETB   RS0
        MOV   R5,#3H
        LJMP  8600H
        ORG   8600H
        MOV   R6,#0
        MOV   R7,#0
LOOP_B: LCALL  675FH
        LCALL  6850H
        DJNZ   R5,INCRE
          MOV R6,#0
        MOV   R7,#0
        LCALL  675FH
         CLR    RS0
        RETI
INCRE:          INC     R6
        SJMP   LOOP_B
        END
     END
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
			ORG E000H
			LCALL 68EAH
			MOV A,TMOD
			ORL TMOD,#10H
			MOV TH1,#00H
			MOV TL1,#FFH
			SETB IT1
			SETB ET1
			SETB EX1
			SETB TR1
			SETB PX1
			MOV R6,#26H
			SETB EA
		HERE	AJMP HERE
			ORG 8028H
			LJMP 8500H
			ORG 8500H
			MOV A,R6
			ORG 8028H
			LJMP 8500H
			ORG 8500H

			MOV A,R6
			SWAP A
			MOV R6,A
			LCALL 677DH
			MOV R0,#FFH
			MOV R1,#FFH
			LCALL 6850H
			MOV R0,#FFH
			MOV R1,#FFH
			LCALL 6850H
			MOV TH1,#00H
			MOV TL1,#FFH
			SETB TR1
			RETI
			ORG 8010H
			SETB RS0
			MOV R5,#3H
			LJMP 8600H
			ORG 8600H
			MOV R6,#0
			MOV R7,#0
			LOOP_B
			LCALL 675FH
			LCALL 6850H
			DJNZ R5,INCRE

		INCRE	MOV R6,#0
			MOV R7,#0
			LCALL 675FH
			CLR RS0
			RETI
			INC R6
			SJMP LOOP_B

RESULT:

PRE LAB QUESTIONS:

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?

LAB ASSIGNMENT:

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

POST LAB QUESTIONS:

1. What is the difference between direct and indirect IO interfacing?
2. What do you mean by segment override prefix?
3. Whether micro reduces memory requirements?

EXPERIMENT No 10

UART OPERATION IN 8051

AIM:-

Write an ALP program Of UART operation in 8051.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8051 trainer kit with keyboard		1
2	Talk with PC		1
3	RPS	+5v	1
4	RS – 232		1
5	FRC cables		1
6	UART Module		1

PROGRAM FOR MODE-0-TRANSMITTER:-

i) Software

```
Org 9000h
MOV SCON,#00H (SCOON=98)
UP1:MOV R7,#8H
MOV A,#80H     (SBUF=99)
Up:CLRTi       (Ti=99)
MOV SBUF,A
XX:JNBTi,XX
CLR P1.0
SETB P1.0
LCALL DELAY
RR A
DJNZ R7,UP
JMP UP1
Delay: MOV R0,#0FFH
Up3:MOV R1,#0FFH
Up2:DJNZ R1,UP2
DJNZ R0,UP3
RET
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
			MOV SCON,#00H (SCOON=98)
		UP1:	MOV R7,#8H
			MOV A,#80H (SBUF=99)
		UP:	CLR Ti (Ti=99)
			MOV SBUF,A
		XX:	JNB Ti,XX
			CLR P1.0
			SETB P1.0
			LCALL DELAY
			RR A
			DJNZ R7,UP
			JMP UP1

DELAY PROGRAM:

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
			MOV R0,#0FFH
		UP3:	MOV R1,#0FFH
		UP2:	DJNZ R1,UP2
			DJNZ R0,UP3
			RET

PROGRAM FOR MODE-0-RECIEVER:

i) Software

```
ORG 9000h
MOV SCON,#11H (SCOON=98)
Up1:CLR P1.0        (P1.0=90)
CLR P3.1        (P3.1=B1)
SETB ri     (Ri=99)
SETB P1.0
CLR Ri              (SBUF=99)
XX: JNB Ri,XX
MOV A,SBUF
MOV R6,A
LCALL DELAY
SJMP UP
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
			ORG
			MOV SCON,#11H (SCOON=98)
		UP1:	CLR P1.0 (P1.0=90)
			CLR P3.1 (P3.1=B1)
			SETB ri (Ri=99)
			SETB P1.0
			CLR Ri (SBUF=99)
		XX:	JNB Ri,XX
			MOV A,SUBF
			MOV R6,A
			LCALL DELAY
			SJMP UP

RESULT:

PRE LAB QUESTIONS:

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?

LAB ASSIGNMENT:

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

POST LAB QUESTIONS:

1. In string operations which is by default string source pointer
2. In string operations which is by default string destination pointer
3. When divide overflow error occurs

EXPERIMENT No 11

INTERFACING LCD TO 8051

AIM:-

Write an ALP program to Interface an LCD with 8051 microcontroller.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8051 trainer kit with keyboard		1
2	LCD Module		1
3	RPS	+5v	1
4	FRC cables		1

THEORY:

LCD display is an inevitable part in almost all embedded projects and this article is about interfacing 16×2 LCD with 8051 microcontroller. Microprocessors use different types of displays to display the data for user. For large amount of data CRT is used but for small amount of data LCDs (Liquid Crystal Display) or 7 segments LED displays will be used. Alphanumeric liquid Crystal displays (LCDs) allow a better user interface, with text messages to enter the instructions and get the response in the form of text and know in a better manner what the machine is doing, including its diagnostics information. The main advantages of LCD displays are there low power consumption and the speed with which the displayed information is updated.

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 42H
```

Memory Location	OPCODE	LABEL	MNEMONIC OPERANDS

			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
			LCALLSET_CON_LINES
			MOV R2,#03H

		BACK	LCALLSET_WR_CON_LINES
			MOV A,#00H
			MOVX @DPTR,A
			MOV DPTR,#PORTA
			MOV A,#FUNCTION_SET
			MOVX @DPTR,A
			MOV 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
			LCALLCHK_BUSY
			LCALLSET_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 @DPTR,A
			MOV R1,A
			INC DPTR
			PUSH DPH
			PUSH DPL
			LCALL CHK_BUSY

		BACK3:	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
			PUSH DPH
			PUSH DPL
			LCALL CHK_BUSY
			LCALL SET_WR_CON_LINES
			MOV A,#00H
			MOVX @DPTR,A
			MOV DPTR,#PORTA
		F1:	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
			PUSH DPL
			MOV R0,#DDRAM_ADD
			LCALL CHK_BUSY
			LCALLSET_WR_CON_LINES
			MOV A,#00H
			MOVX @DPTR,A
			MOV DPTR,#PORTA
			MOV A,R0

		FORW1:	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
			:LCALL 0003H
			MOV DPTR,#CNTRL
			MOV A,#01H
			MOVX @DPTR,A
			MOV A,#03H
			MOVX @DPTR,A
			MOV A,#04H
			MOVX @DPTR,A
			RET
			MOV DPTR,#CNTRL

		FORW:
		
		SET_CON_LINES:
			MOV A,#90H
			MOVX @DPTR,A
			MOV A,#04H

			MOVX @DPTR,A
			MOV A,#00H
			MOVX @DPTR,A
			MOV A,#03H
			MOVX @DPTR,A
			MOV A,#05H
			MOVX @DPTR,A

		CHK_BUSY:	MOV DPTR,#PORTA
			MOVX @DPTR,A
			MOV B,A
			MOV DPTR,#CNTRL
			MOV A,#04H
			MOVX @DPTR,A
			MOV A,B
			JNB A.7,F2
			LJMP BACK2

		BACK2:	MOV DPTR,#CNTRL
			MOV A,#80H
			MOVX @DPTR,A
			RET
			MOV DPTR,#CNTRL
			MOV A,#04H
			MOVX @DPTR,A
			MOV A,#02H

		F2: SET_WR _CON_L I:NES: DELAY: LOOP1: LOOP:	MOVX @DPTR,A RET PUSH R1 NOP DJNZ R1,LOOP POP R1 DJNZ R0,LOOP1 RET

RESULT:

PRE LAB QUESTIONS:

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?

LAB ASSIGNMENT:

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

POST LAB QUESTIONS:

1. Explain the logic of finding out negative nos. from an array of signed nos.
2. Explain the logic of non overlap and overlap block transfer program
3. Explain the logic of string related programs.

EXPERIMENT No 12

INTERFACING MATRIX/KEYBOARD TO 8051

AIM:-

Interface a Keyboardto8051 microcontroller.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8051 trainer kit with keyboard		1
2	Key board module		1
3	RPS	+5v	1
4	FRC cables		1
5	RS-232 cable		

THEORY:

8255 is a general purpose Programmable peripheral interface device. It can be used to interface keyboard with 8051 microcontroller. All the I/O devices require up to 3 I/O ports (Port A, Port B and Port C) which is provided by 8255. Interface circuit also will be simple.

Port A is configures as an input port to receive the row-column code.

Port B is configures as an output port to display the key(s) pressed.

Port C is configures as an output port to output zeros to the rows to detect a key.

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
```

i) Software

```
            Org 9000h
            MOV A,#90H
            MOV DPTR,#CNTRL
            MOVX @DPTR,A
            MOV   B,#20H
Blink 2:  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
```

ii) **Hardware**

Memory Location	**OPCODE**	**LABEL**	**MNEMONIC OPERANDS**
			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
			MOV A,#FEH
		BACK:	MOV B,#21H
			MOV DPTR,#PORTB

		BLINK1:
		
		DELAY:
		OLOOP:
		ILOOP:
			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
			MOV R0,#F7H
			MOV R1,#FFH
			DJNZ R1,ILOOP
			DJNZ R0,OLOOP
			RET

RESULT:

PRE LAB QUESTIONS:

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?

LAB ASSIGNMENT:

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

POST LAB QUESTIONS:

1. Compare memory interfacing and IO interfacing
2. how the even odd address are assigned through 8086
3. how the cs:ip is working during interrupt

CONTENT BEYOND SYLLABI

EXPERIMENT No 1
PROGRAM TO FIND LCM OF A GIVEN NUMBER

AIM:-

To write an assembly language program to find LCM of a given number using 8086

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 trainer kit with keyboard		1
2	RPS	+5v	1
3	RS-232 cable		

PROGRAM:

i) Software

```
Assume cs: code
Code segment
        Start:
                mov ax,data
                mov ds,ax
                mov dx,0h
                mov ax,num
                mov bx,num+2
                up:  push ax
                push dx
                div bx
                cmp dx,0
                je exit
                pop dx
                pop ax
                add ax,num
                jnc down
                inc dx
                down:  jmp up
                exit:  pop lcm+2
                pop lcm
                mov ah,4ch
                int 03h


  Code ends
End start
```

ii) Hardware:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4500		START: MOV	Load the Data to AX.
		AX,DATA	Move the Data AX to DS.
		MOV DS,AX	Initialize the DX.
		MOV DX,0H	Move the first number to AX.
		MOV AX,NUM	Move the second number to BX.
		MOV BX,NUM+2	Store the quotient/first number in AX.
		UP: PUSH AX	Store the remainder value in DX.
		PUSH DX	Divide the first number by second number.
		DIV BX	Compare the remainder.
		CMP DX,0	If remainder is zero, go to EXIT label.
		JE EXIT	If remainder is non-zero,
		POP DX	Retrieve the remainder.
		POP AX	Retrieve the quotient.
		ADD AX,NUM	Add first number with AX.
		JNC DOWN	If no carry jump to DOWN label.
		INC DX	Increment DX.
		DOWN: JMP UP	Jump to Up label.
		EXIT: POP LCM+2	If remainder is zero, store the value at
		POP LCM	LCM+2.
		MOV AH,4CH	
		INT 21H	
		CODE ENDS	
		END START	

Observation Table:

Input	Output
0A, 04	02

Result:

Thus the program to find LCM of a given number using 8086 successfully

PRE LAB QUESTIONS:

1. Types of procedure
2. What does mean by linker
3. What is diff between macro and procedure
4. Which flags of 8086 are not present in 8085?
5. What is LEA?

LAB ASSIGNMENT:

1. Write an ALP program to convert unpacked bcd to packed bcd using 8051
2. Write an ALP program to convert unpacked bcd to Ascii by using 8051

POST LAB QUESTIONS:

1. What is @data indicates in instruction- MOV ax, @data?
2. Explain the logic of string related programs.
3. Which assembler directives are used with far procedure?

EXPERIMENT No 2

PROGRAM TO FIND SQUARE AND CUBE OF A NUMBER

AIM:-

To write an assembly language program to find square and cube of a number using 8086

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 trainer kit with keyboard		1
2	RPS	+5v	1
3	RS-232 cable		

PROGRAM:

iii) Software

```
Assume cs: code
Code segment
        Start:

                mov ax,data
                mov ds,ax
                mov ax,x
                mov bx,x
                mul bx
                mov square,ax
                mul bx
                mov cube,ax
                mov ah,4ch
                                        Int 03h
 Code ends
End start
```

iv) Hardware :

ADDRESS	OPCODE	MNEMONICS	COMMENTS

4500		MOV AX,DATA
		MOV DS,AX
		MOV AX,X
		MOV BX,X
		MUL BX
		MOV SQUARE,AX
		MUL BX
		MOV CUBE,AX
		MOV AH,4CH
		INT 21H
			Load the Data to AX.
			Move the Data AX to DS.
			Move the X number Data to AX.
			Move the X number Data to BX.
			Perform the multiplication by BX.
			Store value in SQUARE.
			Perform the multiplication by BX.
			Store value in CUBE.

Observation Table:

	Input	Output
Square	4h	10h
Cube	4h	40h

Result:

Thus the program to find square and cube of a given number using 8086 successfully

PRE LAB QUESTIONS:

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?

LAB ASSIGNMENT:

1. Write an alp program to divide 32 bit by the 16 bit.
2. Write an alp program for median of an array.

POST LAB QUESTIONS:

1. 80386 is how many bit processors?
2. How many pin IC 80836 is?
3. Mention the priority of interrupts in8086

EXPERIMENT No 3
PARALLEL COMMUNICATION BETWEEN TWO MICROPROCESSORS USING 8255

AIM:-

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

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 trainer kit with keyboard		1
2	RPS	+5v	1
3	8255		1

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 LOCATIO	OPCODE	LABEL	MNEMONICS
4000			MOV AL,90 MOV DX,3006
		LOOP1	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,0A MOV DX,3006 OUT DX CALL **DELAY** MOV AL,0B MOV DX,3006 OUT DX CALL **DELAY** MOV AL,0E MOV DX,3006 OUT DX CALL DELAY MOV AL,0F MOV DX,3006 OUT DX CALL DELAY JMP **LOOP1**

Delay Program:

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

Result:

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

PRE LAB QUESTIONS:

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?

LAB ASSIGNMENT:

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

POST LAB QUESTIONS:

1. What is the size of IVT
2. Which steps 8086 follows to handle any interrupt?
3. role of pointers

EXPERIMENT No 4
SERIAL COMMUNICATION BETWEEN TWO MICROPROCESSOR KITS USING 8251

AIM:-

To write an alp for serial communication between two microprocessors by using 8255

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 trainer kit with keyboard		1
2	RPS	+5v	1
3	8255		1

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			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

			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

PRE LAB QUESTIONS:

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?

LAB ASSIGNMENT:

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.

POST LAB QUESTIONS:

1. How an interrupt is acknowledged?
2. How the even odd addresses are assigned through 8086?
3. How 16 bit processor generates 20 bit addresses

EXPERIMENT No 5
COMMUNICATION BETWEEN 8051 KIT AND PC

AIM:-

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

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8051 trainer kit with keyboard		1
2	RPS	+5v	1
3	8251 UART		1
4	PC		1

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:

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.

PRE LAB QUESTIONS

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?

LAB ASSIGNMENT:

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

POST LAB QUESTIONS

1. Specify addressing modes for any instruction
2. What is the relation between 8086 processor frequency & crystal frequency?
3. How Physical address is generated?