LAB MANUAL

MICROPROCESSORS AND INTERFACING

LAB MANUAL

Subject Code	:	AEC115
Regulations	:	IARE-R16
Class	:	VI Semester (IT)

Prepared By

Ms. B.Lakshmi Prasanna
Assistant Professor, ECE

Department of Computer Science and Engineering
INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal , Hyderabad-500 043

INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal, Hyderabad - 500 043

MICROPROCESSORS AND INTERFACING

LAB WORK BOOK

Name of the Student			
Roll No.			
Branch			
Class		**Section**	

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad

Vision

To bring forth professionally competent and socially sensitive engineers, capable of working across cultures meeting the global standards ethically.

Mission

To provide students with an extensive and exceptional education that prepares them to excel in their profession, guided by dynamic intellectual community and be able to face the technically complex world with creative leadership qualities.

Further, be instrumental in emanating new knowledge through innovative research that emboldens entrepreneurship and economic development for the benefit of wide spread community.

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 - 500 043, Hyderabad

Certificate

This is to Certify that it is a bonafied record of Practical work done by

Sri/Kum. ______________________________ *bearing the Roll No.*

_______________________ *of* _____ *Class*

____ *Branch in the*

___________________________*laboratory during the Academic year*

__________*under our supervision.*

Head of the Department **Lecture In-Charge**

External Examiner **Internal Examiner**

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad

COMPUTER SCIENCE AND ENGINEERING

COURSE OBJECTIVE:

I. Developing of assembly level programs and provide the basics of the microprocessors.

II. Provide solid foundation on interfacing the external devices to the processor according to the user requirements to create novel products and solutions for the real time problems.

III. Understand various interfacing circuits necessary for various applications.

COURSE OUTCOMES:

Upon the completion of Microprocessor & Microcontrollers practical course, the student will be able to:

CO1. Familiarize with the assembly level programming using 8086microprocessor.
CO2. Design circuits for various applications using microprocessor.
CO3. An in-depth knowledge of applying the concepts on real- timeapplications
CO4. Design and apply interfacing circuits for different applications
CO5. Understand the basic concepts of 8086 microprocessors with their application

COURSE LEARNING OUTCOMES (CLOs):

The students should enable to:

1. Design and develop an Assembly language program using 8086microprocessor.
2. Understand the 16 Bit arithmetic and logical operations using WIN862 software.
3. Understand the program to perform multi byte addition, subtraction and 3*3 matrixmultiplications.
4. Understand the to perform ascending and descending order using 8086
5. Understand the program to perform LCM & HCF, square and cube of a given numbers.
6. Understand the programming concepts on strings using 8086
7. Understand the programming for Code converters.
8. Design and interacting stepper motor to 8086.
9. Analyze and interfacing to convert analog todigital.
10. Analyze and interfacing to convert digital toanalog.
11. Develop and design a ALP program to interface stepper motor to 8086.
12. Develop and design a ALP program for serial and parallel communication between two microprocessors.
13. Develop and design an Interface traffic light controller and tone generator using 8086.

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad

INFORMATION TECHNOLOGY

INSTRUCTIONS TO THE STUDENTS

1. Students are required to attend all labs.

2. Students should work individually in the hardware and software laboratories.

3. Students have to bring the lab manual cum observation book, record etc along with them whenever they come for lab work.

4. Should take only the lab manual, calculator (if needed) and a pen or pencil to the work area.

5. Should learn the prelab questions. Read through the lab experiment to familiarize themselves with the components and assembly sequence.

6. Should utilize 3 hour‟s time properly to perform the experiment and to record the readings. Do the calculations, draw the graphs and take signature from the instructor.

7. If the experiment is not completed in the stipulated time, the pending work has to be carried out in the leisure hours or extended hours.

8. Should submit the completed record book according to the deadlines set up by the instructor.

9. For practical subjects there shall be a continuous evaluation during the semester for 25 sessional marks and 50 end examination marks.

10. Out of 25 internal marks, 15 marks shall be awarded for day-to-day work and 10 marks to be awarded by conducting an internal laboratory test.

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

Electronics & Communication Engineering

Program Outcomes	
PO1	**Engineering Knowledge :** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems
PO2	**Problem Analysis :** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
PO3	**Design/Development of Solutions :** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
PO4	**Conduct Investigations of Complex Problems** : Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide validconclusions
PO5	**Modern Tool Usage** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
PO6	**The Engineer And Society** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
PO7	**Environment and sustainability** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
PO8	**Ethics** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
PO9	**Individual and Team Work** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
PO10	**Communication** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions
PO11	**Project management and finance** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one‟s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments
PO12	**Life-long learning** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

Program Specific Outcomes	
PSO1	**Professional Skills:** The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexity.
PSO2	**Problem-Solving Skills:** The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.
PSO3	**Successful Career and Entrepreneurship:** The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.

MICROPROCESSORS and INTERFACING

LAB SYLLABUS

Recommended Systems/Software /Hardware Requirements:

Pentium based desktop PC with minimum of 166 MHZ or faster processor with at least 64 MB RAM and 100MB free disk space. WIN862 software

The Following experiments are to be written for assembler and execute the same with 8086 hardware kits.

S. No.	List of Experiments	Page No.
1	Design and develop an Assembly language program using 8086 microprocessor and to show the following aspects. a. Programming b. Execution c. Debugging To Demonstrate the Tool Chain for MASM and Hardware for 8086 Microprocessor	12
2	a. Write an ALP program to perform 8 Bit arithmetic operations using MASM software and 8086. b. Write an ALP program to perform 16 Bit arithmetic operations using MASM software and 8086.	30
3	a. Write an ALP program to perform multi byte addition and subtraction. b. Write an ALP program to perform 3*3 matrix multiplicationand addition	39
4	a. Write an ALP program to perform ascending order using 8086 b. Write an ALP program to perform descending order using 8086	48
5	a. Write an ALP program to find the LCM & HCF of givennumbers. b. Write an ALP program to find square and cube of a given numbers.	53
6	a. Write an ALP program to insert or delete a byte in the givenstring. b. Write an ALP program to search a number/character in a given string.	58
7	a. Write an ALP program to move a block of data from one memory location to the other. b. Write an ALP program for reverse of a given string.	65
8	a. Write an ALP program to find the number of even and odd numbers in the given string. b. Write an ALP program to generate a Fibonacci series.	69
9	a. Write an ALP program to convert packed BCD to Unpacked BCD. b. Write an ALP program to convert packed BCD to ASCII. c. Write an ALP program to convert hexadecimal to ASCII.	73
10	a. Write an ALP program to convert analog to digital using8086. b. Write an ALP program to convert digital to analog using 8086.	79

11	Write an ALP program to generate Saw tooth and staircase wave forms.	84
12	a. Write an ALP program to rotate stepper motor in clockwise direction. b. Write an ALP program to rotate stepper motor in anti clockwise direction.	87
13	a. Parallel communication between two microprocessors using8255. b. Serial communication between two microprocessor kitsusing 8251.	93
14	a. Write a program to interface traffic light controller. b. Write an ALP program to interface tone generator.	103

ATTAINMENT OF PROGRAM OUTCOMES & PROGRAM SPECIFIC OUTCOMES

Exp. No.	Experiment	Program Outcomes Attained	Program Specific Outcomes Attained
1	Design and develop an Assembly language program using 8086 microprocessor and to show the following aspects. a. Programming b. Execution c. Debugging To Demonstrate the Tool Chain for MASM and Hardware for 8086 Microprocessor	PO1, PO2	PSO1
2	a. Write an ALP program to perform 8 Bit arithmetic operations using MASM software and 8086. b. Write an ALP program to perform 16 Bit arithmetic operations using MASM software and 8086.	PO1, PO2	PSO1
3	a. Write an ALP program to perform multi byte addition andsubtraction. b. Write an ALP program to perform 3*3 matrix multiplication and addition	PO1, PO2	PSO1, PSO2
4	a. Write an ALP program to perform ascending order using 8086 b. Write an ALP program to perform descending order using 8086	PO1, PO2	PSO1
5	a. Write an ALP program to find the LCM & HCF of given numbers. b. Write an ALP program to find square and cube of a given numbers.	PO1, PO2, PO5	PSO1, PSO2
6	a. Write an ALP program to insert or delete a byte in the given string. b. Write an ALP program to search a number/character in a given string.	PO1, PO2, PO5	PSO1

ATTAINMENT OF PROGRAM OUTCOMES & PROGRAM SPECIFIC OUTCOMES

Exp. No.	Experiment	Program Outcomes Attained	Program Specific Outcomes Attained
7	a. Write an ALP program to move a block of data from one memory location to the other. b. Write an ALP program for reverse of a given string.	PO1, PO2	PSO1
8	a. Write an ALP program to find the number of even and odd numbers in the given string. b. Write an ALP program to generate a Fibonacci series.	PO1, PO2	PSO1
9	a. Write an ALP program to convert packed BCD to Unpacked BCD. b. Write an ALP program to convert packed BCD to ASCII. c. Write an ALP program to convert hexadecimal to ASCII.	PO1, PO2, PO5	PSO1
10	a. Write an ALP program to convert analog to digital using 8086. b. Write an ALP program to convert digital to analog using 8086.	PO1, PO2, PO5	PSO1, PSO2
11	Write an ALP program to generate Saw tooth and staircase wave forms.	PO1, PO2, PO4, PO5	PSO2
12	a. Write an ALP program to rotate stepper motor in clockwise direction. b. Write an ALP program to rotate stepper motor in anti clockwise direction.	PO1, PO2, PO5	PSO2
13	a. Parallel communication between two microprocessors using 8255. b. Serial communication between two microprocessor kits using 8251.	PO1, PO2	PSO1
14	a. Write a program to interface traffic light controller. b. Write an ALP program to interface tone generator.	PO1, PO2	PSO1
15	Write an ALP program to interfacing Elevator to 8086	PO1, PO2	PSO1

EXPERIMENT NO: 1

DESIGN A PROGRAM USING WIN862

INTRODUCTION:

Features of the ESA -86/88 Microprocessor Trainer

- 8086 CPU operating at 8 MHz MAX mode.
- Provision for on-board 8087 (NDP) coprocessor.
- Provision for 256 KB of EPROM & 128 KB of RAM onboard
- Battery backup facility for RAM.
- 48 programmable I/O lines using two 8255"s
- Timer1 & Timer2 signals are brought out to header pins
- Priority Interrupt Controller (PIC) for eight input using 8259A
- In standalone mode using on board keypad or with PC compatible system through its RS-232 interface
- Display is 8 seven segment LED
- Designed & engineered to integrate user"s application specific interface conveniently at a minimum cost.
- Powerful & user-friendly keyboard / serial monitor, support in development of application programs.
- Software support for development of programs on Computer, the RS-232C interface cable connecting to computer from the kit facilitates transfer of files between the trainer kit & computer for development & debugging purposes.
- High quality reliable PCB with solder mask on both sides & clear legend prints with maximum details provided for the user.

SPECIFICATIONS:

CPU: Intel 8086 operating at 8 MHz in MAX mode.

MEMORY: Total 1MB of memory is in the Kit provided.

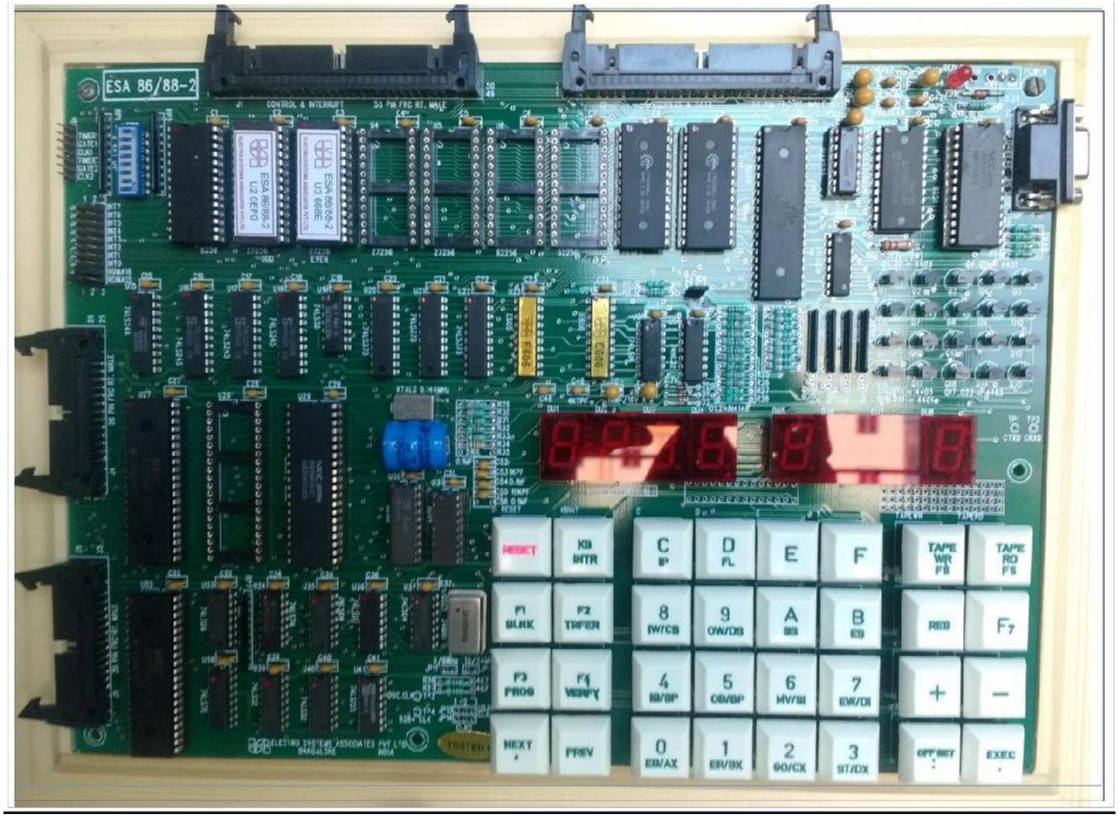
EPROM: 4 JEDEC compatible sockets for EPROM

RAM: 4 JEDEC compatible sockets for RAM

PARALLEL I/O: 48 I/O lines using two 8255

SERIAL I/O: One RS-232C compatible interface Using UART 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.

KEYBOARD / DISPLAY:

Keyboard: keyboard on to the trainer.

Display: 8 seven segment displays

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.

INTERFACE BUS SIGNALS:

CPU BUS: All address, data & control lines are TTL compatible & are terminated in berg strip header.

PARALLEL I/O: All signals are TTL compatible & Terminated in berg strip header For PPI expansion.

SERIAL I/O: Serial port signals are terminated in Standard 9-pin „D type connector.

MONITOR SOFTWARE:

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

COMPUTER INTERFACE:

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

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	

POWER REQUIREMENTS:

+5V DC with 1300 mA current rating (Max).

OPERATING CONFIGURATION:

Two different modes of operation trainer are possible. They are

(i) Serial operation

(ii) Keypad 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.

EXECUTION PROCEDURE FOR 8086 (for registers):

i) Writing a alp PROGRAM intoprocessor:

Switch On Power Supply

Check if DIP switches board is in serial or keyboard mode (Serial mode = 1 on, Board mode = 4 On)

Press Reset

Press „EB‟(Examine Byte)

Enter Starting Memory location (Ex: 2000)

Press next button, Enter OP-Code value

Then press next button Enter 2nd memory location and op code

.

.

.

Enter up to nth values

Execution:

Press Exec. Button

Press Go enter starting memory location

Press Exec.

Press ER (Examine Register)

Press AX (Now see the result in Ax)

EXECUTION PROCEDURE FOR 8086 (for memory locations):

ii) Writing a alp PROGRAM into processor:

Switch On Power Supply

Check if DIP switches board is in serial or keyboard mode (Serial mode = 1 on, Board mode = 4

On)

Press Reset

Press „EB‟ (Examine Byte)

Enter Starting Memory location (Ex: 2000)

Press next button, Enter OP-Code value

Then press next button Enter 2nd memory location and op code

.

.

.

Enter up to nth values

Execution:

Press Exec. Button

Press Go enter starting memorylocation

Press Exec.

Press EB give input memory location and input values

Press Exec.

Press Go Give starting memorylocation

Press Exec.

Press Go Now observe the results in memory location

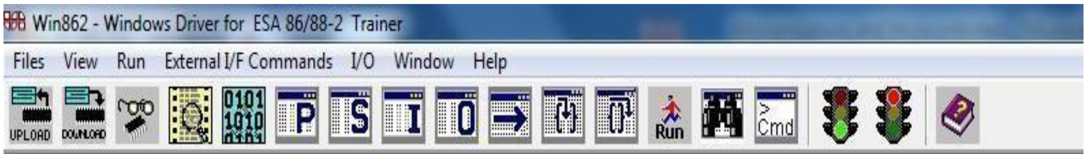
WIN862 Software procedure:

Registers:

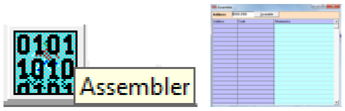
Step 1: Open Win862 icon on desktop (see Fig.1) and opened Window see fig.2

1.

2.



Step 2: Click on Assembler and give starting address (Like 0000:4000) then press Enter button.



Step 3: Then write 1st Instruction then press enter button.

Step 4: Then write 2nd Instruction then press enter button.

Step 5: Then write up to nth Instruction then press enter button and close the Assembler window.

Step 6: Now click on Dis Assembler and give starting address (Like 0000:4000) then press enter button.

Step 7: Click on Set PC then give starting address then press Enter button.

Step 8: Click on Run (check whether program is executed or not)

Step 9: Click on view registers and observe the results in registers.

<u>Memory locations:</u>

Step 1: Open Win862 icon on desktop.

Step 2: Click on Assembler and give starting address (Like 0000:4000) then press Enter button.

Step 3: Then write 1st Instruction then press enter button.

Step 4: Then write 2nd Instruction then press enter button.

Step 5: Then write up to nth Instruction then press enter button and close the Assembler window.

Step 6: Now click on Dis Assembler and give starting address (Like 0000:4000) then press enter button.

Step 7: Click on Set PC then give starting address then press Enter button.

Step 8: Click on Run (check whether program is executed or not)

Step 9: Click on view memory

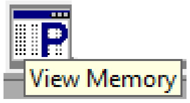
Step 10: Now enter input address

Step 11: Click on Modify and Give desired input values

Step 12: Click on Set PC. Enter initial address and press Dis-Assembler

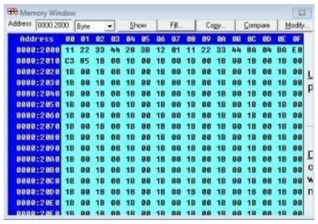
Step 13: Click on Run (check whether program is executed or not)

Step 14: Now observe the result in view memory.



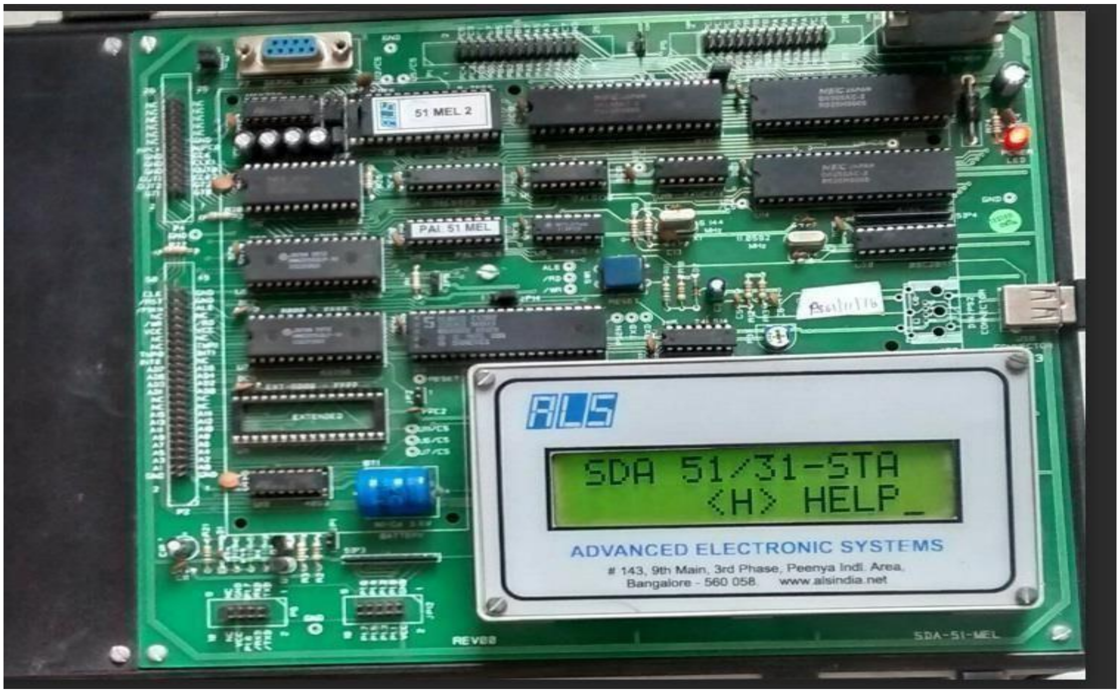
Step 15: Click on view memory and enter destination address then press enter button

Step 16: Now observe the result.



```
Address    00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000:2000  11 22 33 44 28 3B 12 01 11 22 33 44 8A 04 8A E0
0000:2010  C3 85 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:2020  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:2030  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:2040  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:2050  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:2060  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:2070  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:2080  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:2090  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:20A0  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:20B0  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:20C0  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:20D0  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:20E0  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
0000:20F0  1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00 1B 00
```

INTRODUCTION OF ALS SDA 8051-MEL:


51 MEL 2
PAL 51 MEL
ALS
SDA 51/31-STA
<H> HELP_
ADVANCED ELECTRONIC SYSTEMS
# 143, 9th Main, 3rd Phase, Peenya Indl. Area,
Bangalore - 560 058. www.alsindia.net
REV02
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.

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 friendlymonitor software (27256) that allows
Program enter, verification, debugging and execution from the system keyboard or aCRT
Terminal or a PC functioning as a terminal. File uploading/downloading option is in serial mode

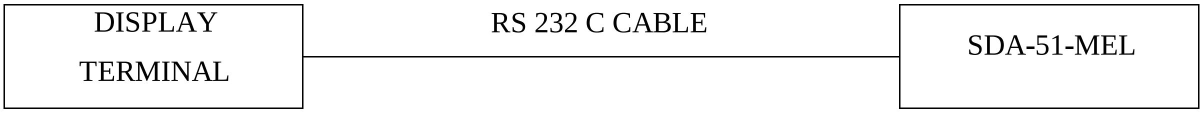
THE EXTERNAL PC: AT keyboard allows users to directly assemble /disassemble mnemonics/instructions for 8051 using the alphanumeric LCD display

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 optionalaccessory.

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.

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.

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:

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 startaddress.
- It program breakpoint or data breakpoint have been enabled then the program will beexecuted 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 bestored.

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 >cdcomm_pack86

Enter

D:\cd comm_pack86 >cd comm_pack86>cd x8086

Enter

D:\cd comm_pack86 >cd comm_pack86>cd x8086>edit filename

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: 2

8 AND 16 BITARITHMETIC OPERATIONS

a) **Write an ALP program to perform 8 Bit arithmetic operations using MASM software and 8086.**

AIM: -

To write an assembly language program for Addition of two 8-bit numbers.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/Win862 with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM FOR TWO 8-BIT NUMBERS:

A) ADDITION

i) Software

```
MOV AL, 43
MOV BL, 11
ADD AL, BL
INT 03
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABLE	MNEMONIC OPERAND	COMMENTS
			MOV AL,43 MOV BL,11 ADD AL,BL INT 3	

Observation Table

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

B) SUBTRACTION

8 Bit Subtraction

AIM: -

To write an assembly language program for subtraction of two 8-bit numbers.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/Win862 with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM:

i) Software

```
MOV AL, 43
MOV BL, 11
SUB AL, BL
INT 03
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABLE	MNEMONIC OPERAND	COMMENTS
			MOV AL,43 MOV BL,11 SUB AL,BL INT 03	

Observation Table

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

C) MULTIPLICATION

8 Bit Multiplication

AIM: -

To write an assembly language program for multiplication of two 8-bit numbers.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/Win862with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM:

i) Software

```
MOV AL, 43
MOV BL, 11
MUL BL
INT 03
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND	COMMENTS
			MOV AL,43 MOV BL,11 MUL BL INT 3	

Observation Table

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

D) DIVISION

i) 8 bit division

AIM:-

To write an assembly language program for division of two 8-bit numbers.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/Win862 with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM

i) Software

```
MOV AL, 10
MOV BL, 02
DIV BL
INT 03
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND	COMMENTS
			MOV AL,10 MOV BL,02 DIV BL INT 3	

Observation Table

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

RESULT:

b) **Write an ALP program to perform 16 Bit arithmetic operations using MASM software and 8086.**

AIM: -

To write an assembly language program for addition of two 16-bit numbers.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/Win862 with PC		1
2	Keyboard		1
3	RPS	+5v	1

ADDITION:

i) **Software**

```
MOV AX, 4343
MOV BX, 1111
ADD AX, BX
INT 03
```

ii) **Hardware**

MEMORY LOCATION	OP-CODE	LABLE	MNEMONIC OPERAND	COMMENTS
			MOV AX,4343 MOV BX,1111 ADD AX,BX INT 3	

Observation Table

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

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/Win862 with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM:

i) Software

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

ii) Hardware

MEMORY LOCATION	OP-CODE	MNEMONIC OPERAND	COMMENTS
		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		

C) 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/Win862with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM:

i) Software

```
MOV AX, 4343
MOV BX, 1111
MUL BX
INT 03
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND	COMMENTS
			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

D) DIVISION

AIM:-

To write an assembly language program for division of two 16-bit numbers.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/Win862 with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM

i) Software

```
MOV AX, 0080
        MOV BX, 0008
        DIV BX
        INT 03
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND	COMMENTS
			MOV AX,0080 MOV BX,0008 DIV BX INT 3	

Observation Table

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

RESULT:

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?

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?

3. 1) 0012

4. 2) 0006

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: 3

PALINDROME, ABSTRACT CLASS

a) Write an ALP program to perform multi byte addition and subtraction

i) MULTI BYTE ADDITION

AIM: -

Write an ALP program to perform multi byte addition

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/Win862 with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM:

i) Software

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

ii) Hardware:

MEMORY LOCATION	OP-CODE	LABLE	MNEMONIC OPERAND	COMMENTS
			MOV AX,0000 MOV SI, 2000 MOV DI, 3000 MOV BX, 2008	

			MOV CL, 04	
		UP:	MOV AL, [SI]	
			ADD AL, [BX]	
			MOV [DI], AL	
			INC SI	
			INC BX	
			INC DI	
			DEC CL	
			JNZ UP	
			INT 3	

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					

RESULT:

ii) MULTI BYTE SUBTRACTION

AIM: - Write an ALP program to perform multi byte subtraction.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/Win862 with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM:

i) Software

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

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND	COMMENTS
			MOV AX,0000	
			MOV SI, 2000	
			MOV DI, 3000	
			MOV BX, 2008	
			MOV CL, 04	
		UP	MOV AL, [SI]	
			SUB AL, [BX]	

			MOV [DI], AL INC SI INC BX INC DI DEC CL JNZ UP INT 03	

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					

RESULT:

b) Write an ALP program to perform 3*3 matrix multiplication andaddition

AIM: - Write an ALP program to perform 3*3 matrix multiplication andaddition.

COMPONENTS & EQUIPMENT REQUIRED: -

S.No	Device	Range / Rating	Quantity (in No's)
1	8086 microprocessor kit/Win862 with PC		1
2	Keyboard		1
3	RPS	+5v	1

MULTIPLICATION:

PROGRAM:

i) Software

```
ASSUME CS: CODE
CODE SEGMENT
          MOV  SI,1000
          MOV  BP,1020
          MOV DI,1050
      L2:  MOV  CX,00
      L1: MOV AL,[SI]
          MOV BL,[BP]
          MUL BL
          ADD CX,AX
          ADD BP,03
          INC SI
          CMP BP,1029
          JB L1
          SUB SI,03
          SUB BP,08
          ADD DI,02
          CMP BP,1023
          JB L2
         ADD SI,03
         SUB BP,03
         CMP DI,1051
         JB L2
          HLT
CODE ENDS
END START
```

ii) Hardware

MEMORY LOCATION	**OP-CODE**	**LABEL**	**MNEMONIC OPERAND**	**COMMENTS**
			MOV SI,1000	
			MOV BP,1020	
			MOV DI,1050	
		L2 :	MOV CX,00	
		L1 :	MOV AL,[SI]	
			MOV BL,[BP]	
			MUL BL	
			ADD CX,AX	
			ADD BP,03	
			INC SI	
			CMP BP,1029	
			JB L1	
			SUB SI,03	
			SUB BP,08	
			ADD DI,02	
			CMP BP,1023	
			JB L2	
			ADD SI,03	
			SUB BP,03	
			CMP DI,1051	
			JB L2	
			INT 03	

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				3004	
2005				3005	
2006				3006	
2007				3007	

RESULT:

ADDITION:

PROGRAM:

i) Software

```
ASSUME CS: CODE
CODE SEGMENT
        START: MOV AX, DATA
               MOV DS, AX
               MOV BX,OFFSET MATRIX1
               MOV BP,OFFSET RESULT
               SUB BX,N
               DEC BX
               PUSH BX
       NEW_ROW:MOV DI,OFFSET MATRIX2
               DEC DI
               POP BX
               ADD BX,N
               INC BX
               DEC BP
               MOV CX,N
               MOV COUNTER,CX
     EACH_ROW: INC BP
               PUSH BX
               INC DI
               DEC COUNTER
               JE NEW_ROW
          COL: MOV AH,[BX]
               MOV AL,[DI]
               MUL AH
               ADD [BP],AL
               INC BX
               ADD DI,N
               DEC COLUMN
               JNZ COL
               MOV CX,N
               MOV COLUMN,CX
               POP BX
               DEC ROW
               JNZ EACH_ROW
               MOV AX, 4C00H
               INT 21H
        CODE ENDS
        END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND	COMMENTS
			MOV BX,OFFSET MATRIX1	
			MOV BP,OFFSET RESULT	
			SUB BX,N	
			DEC BX	
			PUSH BX	
		NEW_ROW:	MOV DI,OFFSET MATRIX2	
			DEC DI	
			POP BX	
			ADD BX,N	
			INC BX	
			DEC BP	
			MOV CX,N	
			MOV COUNTER,CX	
			INC BP	
			PUSH BX	
		EACH_ROW:	INC DI	
			DEC COUNTER	
			JE NEW_ROW	
			MOV AH,[BX]	
			MOV AL,[DI]	
		COL:	MUL AH	
			ADD [BP],AL	
			INC BX	
			ADD DI,N	
			DEC COLUMN	
			JNZ COL	
			MOV CX,N	
			MOV COLUMN,CX	
			POP BX	
			DEC ROW	
			JNZ EACH_ROW	
			MOV AX, 4C00H	
			INT 03H	

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				3004	
2005				3005	
2006				3006	
2007				3007	

RESULT:

EXPERIMENT NO: 4

PROGRAMS TO SORT NUMBERS

a) Write an ALP program to perform ascending order using 8086

AIM:-

Write an assembly language 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/win 862 with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM:

i) Software

```
                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
```

ii) Hardware

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

RESULT:

b) Write an ALP program to perform descending order using 8086

AIM:-

Write an assembly language 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/Win862 with PC		1
	Keyboard		1
3	RPS	+5v	1

PROGRAM:

i) Software

```
                MOV  AX,0000
                MOV  CH,0004
                DEC CH
   UP1 :        MOV CL, CH
                MOV SI, 2000
  UP:           MOV  AL,[SI]
                INC SI
                CMP AL,[SI]
                JNC DOWN
                XCHG AL,[SI]
                DEC SI
                MOV [SI],AL
                INC SI
  DOWN:         DEC CL
               JNZ UP
               DEC CH
               JNZ UP1
               INT 3
```

ii) Hardware

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

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 ascendingorder?

   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 5

PROGRAM TO FIND LCM OF A GIVEN NUMBER

a) Write an assembly language Program to find the LCM and HCF of 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 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 21H CODE ENDS END START	Load the Data to AX. Move the Data AX to DS. Initialize the DX. Move the first number to AX. Move the second number to BX. Store the quotient/first number in AX. Store the remainder value in DX. Divide the first number by second Compare the remainder. If remainder is zero, go to EXIT label. If remainder is non-zero, Retrieve the remainder. Retrieve the quotient. Add first number with AX. If no carry jump to DOWN label. Increment DX. Jump to Up label. If remainder is zero, store the value at LCM+2.

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 using8051
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?

b) Write an assembly language program to find square and cube of a number using 8086

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:

i) 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
```

ii) 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 negativenumbers?
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 16bit.
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: 6

PROGRAM FOR STRING MANIPULATIONS OPERATIONS

a) Write an ALP program to insert or delete a byte in the givenstring.

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	COMMENTS
			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	
			MOVSB	
		L2:	MOV BX,7000	
			MOV AL,[BX]	
			MOV [DI],AL	
			DEC CX	
			INC DI	
			REP	
			MOVSB	
			INT 3	
		L3:		

Observation Table

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

RESULT:

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)
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:     INC SI
                DEC CX
                REP MOVSB
        L3:     INT 03H
CODE ENDS
END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND	COMMENTS
			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	

			INC SI	
		L2:	DEC CX	
			REP	
			MOVSB	
			INT 3	
		L3:		

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

b) Write an ALP program to search a number/character in a givenstring.

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)
1	8086 microprocessor kit/Win862 with PC		1
2	Keyboard		1
3	RPS	+5v	1

PROGRAM:

i) Software

```
                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 3
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND	COMMENTS
			MOV CX, 0004	
			MOV AX,0000	
			MOV SI,2000	
		UP:	MOV BX,3000	
			MOV AL,[SI]	
			CMP AL,[BX]	
			JZ DOWN	
			INC SI	
		DOWN:	DEC CL	
			JNZ UP	
		L3:	MOV AH,00	
			JMP L3	
			DEC CL	
			MOV AH,01	
			MOV [DI], AH	
			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 givenarray?
   1) 02
   2) 06
   3) 05
   4) 08
2. Write an alp program to search a number 45 from a givenarray?
   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: 7

PROGRAM FOR STRING MANIPULATIONS OPERATIONS

a) Write an ALP program to move a block of data from one memory location to the other.

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)
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, 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	

RESULT:

b) Write an ALP program for reverse of a given string.

AIM:-

To write a ALP for reverse of 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

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	COMMENTS
			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	

RESULT:

EXPERIMENT NO: 8

PROGRAMS FOR STRING MANIPULATIONS OPERATIONS

a) Write an ALP program to find the number of even and odd numbers in the given string.

AIM:-

To write a ALP program to find the number of even and odd numbers in the 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

PROGRAM:

i) Software

```
ASSUME CS: CODE
CODE SEGMENT
START:       MOV CH, 05H
              MOV SI, 2000H
              MOV BX, 0000H
              MOV DX, 0000H
        L3:   MOV AL,[SI]
              MOV CL, 01H
              ROR AL,CL
              JC L1
              INC DX
              JMP L2
        L1:   INC BX
        L2:   INC SI
              DEC CH
              JNZ L3
              INT 03H
CODE ENDS
END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND	COMMENTS
			MOVCH, 05H	
			MOV SI, 2000H	
			MOV BX, 0000H	
			MOV DX, 0000H	
		L3:	MOV AL,[SI]	
			MOV CL, 01H	
			ROR AL,CL	
			JC L1	
			INC DX	
			JMP L2	
		L1:	INC BX	
		L2:	INC SI	
			DEC CH	
			JNZ L3	
			INT 03H	

Observation Table

Input		Output	
MEMORY LOCATION	**Data**	**REGISTER**	**Data**
2000		BX	
2001		DX	
2002			
2003			
2004			

RESULT:

b) Write an ALP program to generate a Fibonacci series.

AIM:-

To write a ALP program to generate a Fibonacci series.

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 CX, 05H
        UP:     MOV AL,[SI]
               INC SI
                MOV BL,[SI]
                ADD AL,BL
                INC SI
                MOV [SI],AL
                DEC SI
                DEC CX
                JNZ UP
                INT 03H
CODE ENDS
END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND	COMMENTS
			MOV SI, 2000H	
			MOV CX, 05H	
		UP:	MOV AL,[SI]	
			INC SI	
			MOV BL,[SI]	
			ADD AL,BL	
			INC SI	
			MOV [SI],AL	
			DEC SI	

			DEC CX JNZ UP INT 03H	

Observation Table

Input		**Output**	
MEMORY LOCATION	**Data**	**MEMORY LOCATION**	**Data**
2000		2000	
2001		2001	
		2002	
		2003	
		2004	

RESULT:

EXPERIMENT NO: 9

CODE CONVERTIONS

a) Write an ALP program to convert packed BCD to Unpacked BCD.

AIM:-

To write a ALP program to convert packed BCD to Unpacked BCD.

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 AL, 72H
            MOV AH, AL
            AND AL, 0FH
            MOV CL, 04H
            SHR AH, CL
            INT 03H
CODE ENDS
END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND	COMMENTS
			MOV AX, 0000H MOV AL, 72H MOV AH, AL AND AL, 0FH MOV CL, 04H SHR AH, CL INT 03H	

Observation Table

Input		Output	
REGISTER	**Data**	**REGISTER**	**Data**
AL		AX	

RESULT:

b) Write an ALP program to convert packed BCD to ASCII.

AIM:-

To write a ALP program to convert packed BCD to ASCII.

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 AL, 56H
                MOV AH, AL
                AND AL, 0FH
                MOV CL, 04H
                SHR AH, CL
                OR AX, 3030H
                INT 03H
CODE ENDS
END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND	COMMENTS
			MOV AL, 56H MOV AH, AL AND AL, 0FH MOV CL, 04H SHR AH, CL OR AX, 3030 INT 03H	

Observation Table

Input		Output	
REGISTER	**Data**	**REGISTER**	**Data**
AL		AX	

RESULT:

c) Write an ALP program to convert hexadecimal to ASCII.

AIM:-

To write a ALP program to convert hexadecimal to ASCII.

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 CX, 0005H
        UP: MOV AL, [SI]
                CMP AL, 0AH
                JC FWD
                ADD AL, 07H
        FWD: OR AL, 30H
                MOV [DI], AL
                INC SI
                INC DI
                DEC CX
                JNZ UP
                INT 03H
CODE ENDS
END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND
			MOV SI, 2000H
			MOV DI, 3000H
			MOV CX, 0003H
		UP:	MOV AL, [SI]
			CMP AL, 0AH
			JC FWD

		FWD:	ADD AL, 07H OR AL, 30H MOV [DI], AL INC SI INC DI DEC CX JNZ UP INT 03H

Observation Table

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

RESULT:

EXPERIMENT NO: 10

INTERFACING ADC & DAC DEVICES

a) Write an ALP program to convert analog to digital using 8086.

AIM:-

To write an Assembly Language Program for Interfacing ADC in 8086Microprocessor kit.

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 Interfacing modules		1
5	Power mate connector		1
6	FRC Connector, RS-232 cable		1

PROGRAM:

i) Software

```
ASSUME CS: CODE
CODE SEGMENT
START:      MOV AL, 98H
            MOV DX, 0FFE6
            OUT DX,AL
            MOV AL, 01H
            OUT DX,AL
            MOV AL, 00H
            OUT DX,AL
            MOV AL, 02H
            MOV DX, 0FFE2H
            OUT DX,AL
            MOV DX, 0FFE4H
      BACK: IN AL,DX
            ROR AL, 1H
            JNC BACK
            MOV DX, 0FFE0H
            IN AL, DX
            MOV DI, 2000H
            MOV [DI], AL
```

```
                INT 03H
        CODE ENDS
        END START
```

ii) Hardware

MEMORY LOCATION	OP-CODE	LABEL	MNEMONIC OPERAND	COMMENTS
			MOV AL, 98H	
			MOV DX, 0FFE6	
			OUT DX,AL	
			MOV AL, 01H	
			OUT DX,AL	
			MOV AL, 00H	
			OUT DX,AL	
			MOV AL, 02H	
			MOV DX, 0FFE2H	
			OUT DX,AL	
			MOV DX, 0FFE4H	
		BACK:	IN AL,DX	
			ROR AL, 1H	
			JNC BACK	
			MOV DX, 0FFE0H	
			IN AL,DX	
			MOV DI, 2000H	
			MOV [DI], AL	
			INT 03H	

Observation Table

Input		Output	
REGISTER	Data	MEMORY LOCATION	Data
AL		2000	

RESULT:

b) Write an ALP program to convert digital to analog using 8086.

AIM:-

To write a program for conversion of digital data to analog output. The analog output will be in the form of square wave and triangular 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	D/A Interfacing modules		1
5	Power mate connector		1
6	FRC Connector, RS-232 cable		1
7	CRO		1

D/A CONVERTER:

PROGRAM TO GENERATE SQUARE WAVE:

i) Software

```
ASSUME CS: CODE
CODE SEGMENT
START:    MOV AL, 80H
          MOV DX, 0FFE6H
          OUT DX, AL
          MOV DX, 0FFE0H
 BACK:     MOV AL, 00H
          OUT DX, AL
          MOV CX, 0147H
 SELF1:   LOOP SELF1
          MOV AL, 0FFH
          OUT DX, AL
          MOV CX, 0147H
  SELF2:  LOOP SELF2
          JMP BACK

   CODE ENDS
   END START
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
			MOV AL, 80H
			MOV DX, 0FFE6H
			OUT DX, AL
			MOV DX, 0FFE0H
		BACK:	MOV AL, 00H
			OUT DX, AL
			MOV CX, 0147H
		SELF1:	LOOP SELF1
			MOV AL, 0FFH
			OUT DX, AL
			MOV CX, 0147H
		SELF2:	LOOP SELF2
			JMP BACK

PROGRAM TO GENERATE TRAINGULARWAVE:

i) Software

```
            ASSUME CS: CODE
            CODE SEGMENT
  START:    MOV AL,80H
            MOV DX,0FFE6H
            OUT DX,AL
            MOV AL,00H
     L3:    MOV DX,0FFE2H
     L1:    OUT DX,AL
            INC AL
            CMP AL,0FFH
            JB L1
     L2:    OUT DX,AL
            DEC AL
            CMP AL,00H
            JNBE L2
            JMP L3
        CODE ENDS
        END START
```

ii) Hardware

ADDRESS	OPCODE	LABEL	MNEMONIC OPERAND
			MOV AL,80H
			MOV DX,0FFE6H
			OUT DX,AL
			MOV AL,00H
		L3:	MOV DX,0FFE2H
		L1:	OUT DX,AL
			INC AL
			CMP AL,0FFH
			JB L1
		L2:	OUT DX,AL
			DEC AL
			CMP AL,00H
			JNBE L2
			JMP L3

RESULT:

EXPERIMENT NO: 11

GENARATE SQUARE, SINE & TRIANGLE WAVES

Write an ALP program to generate Saw tooth and staircase wave forms.

AIM:-

To write a program for conversion of digital data to analog output. The analog output will be in the form of Saw tooth and staircase wave forms.

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	D/A Interfacing modules		1
5	Power mate connector		1
6	FRC Connector, RS-232 cable		1
7	CRO		1

PROGRAM:

A) TO GENERATE SAW TOOTH WAVE FORM:

i) Software

```
              ASSUME CS: CODE
              CODE SEGMENT
      START:  MOV AL, 80H
              MOV DX,0FFE6H
              OUT DX,AL
         L2:  MOV AL,00H
              MOV DX,0FFE2H
         L1:  OUT DX,AL
              INC AL
              CMP AL,0FFH
              JB L1
              OUT DX,AL
              JMP L2
         CODE ENDS
         END START
```

ii) Hardware

ADDRESS	OPCODE	LABEL	MNEMONIC OPERAND
			MOV AL,80H
			MOV DX,0FFE6H
			OUT DX,AL
		L2:	MOV AL,00H
			MOV DX,0FFE2H
		L1:	OUT DX,AL
			INC AL
			CMP AL,0FFH
			JB L1
			OUT DX,AL
			JMP L2

RESULT:

B) TO GENERATE STAIRCASE WAVE FORM:

i) Software

```
                 ASSUME CS: CODE
                 CODE SEGMENT
        START:   MOV AL,80H
                 MOV DX,0FFE6H
                 OUT DX,AL
                 MOV DX,0FFE0H
           UP:   MOV BL, 06H
         BACK:   MOV AL,00H
                 OUT DX,AL
                 MOV CX, 05BE
         SELF:   LOOP SELF
                 ADD AL, 33H
                 DEC BL
                 JNZ BACK
                 JMP UP
         CODE ENDS
         END START
```

ii) Hardware

ADDRESS	OPCODE	LABEL	MNEMONIC OPERAND
			MOV AL,80H
			MOV DX,0FFE6H
			OUT DX,AL
			MOV DX,0FFE2H
		UP:	MOV BL, 06
		BACK:	MOV AL,00H
			OUT DX,AL
			MOV CX, 05BE
		SELF:	LOOP SELF
			ADD AL, 33H
			DEC BL
			JNZ BACK
			JMP UP

RESULT:

EXPERIMENT NO: 12

INTERFACING STEPPER MOTOR

a) Write an ALP program to rotate stepper motor in clockwisedirection.

AIM:-

Write an Assembly Language PROGRAM to rotate the Stepper Motor in 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 numbers 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 givenbelow.

   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 theinterface.

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

```
                        ASSUME CS: CODE
                        CODE SEGMENT
        Start:          MOV AL,80H
                        MOV DX,0FFE6H
                        OUT DX,AL
                        MOV BX,02H
                        MOV AL,33H
                        MOV DX,0FFE0H
                BACK: OUT DX,AL
                        MOV CX, 2DF7H
                SELF: LOOP SELF
                        ROR AL, 1
                        DEC BX
                        JNZ BACK
                        INT 03H
        CSEG ENDS
        END START
```

ii) Hardware

MEMORY LOCATION	**OPCODE**	**LABEL**	**MNEMONIC OPERAND**
			MOV AL,80H MOV DX,0FFE6H OUT DX,AL MOV BX,02H

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
		BACK: SELF:	MOV AL,33H MOV DX,0FFE0H OUT DX,AL MOV CX, 2DF7H LOOP SELF ROR AL, 1 DEC BX JNZ BACK INT 03H

RESULT:

b) Write an ALP program to rotate stepper motor in anti clockwise direction.

AIM:-

Write an Assembly Language PROGRAM to rotate the Stepper Motor in 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 numbers 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:

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

6. 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.

7. 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 theinterface.

   A- GREEN
   C- RED & WHITE
   B- GREEN & WHITE

D- RED
VDD- BLACK & WHITE.

8. 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 ANTI CLOCKWISE DIRECTION :

iii) Software

```
          ASSUME CS: CODE
          CODE SEGMENT
Start:          MOV AL,80H
                MOV DX,0FFE6H
                OUT DX,AL
                MOV BX,02H
                MOV AL,33H
                MOV DX,0FFE0H
          BACK: OUT DX,AL
                MOV CX, 2DF7H
          SELF: LOOP SELF
                ROL AL, 1
                DEC BX
                JNZ BACK
                INT 03H
          CODE ENDS
          END START
```

iv) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
			MOV AL,80H
			MOV DX,0FFE6H
			OUT DX,AL
			MOV BX,02H
			MOV AL,33H
			MOV DX,0FFE0H
		BACK:	OUT DX,AL
			MOV CX, 2DF7H
		SELF:	LOOP SELF
			ROL AL, 1
			DEC BX

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
			JNZ BACK INT 03H

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: 13

PARALLEL AND SERIAL COMMUNICATION

a) 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

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:

i) Software

```
                ASSUME CS: CODE
                CODE SEGMENT
        START:      MOV AL, 90
                    MOV DX, 3006
                    OUT DX
          BACK:     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 BACK
        CSEG ENDS
        END START
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
			MOV AL,90
			MOV DX,3006
			OUT DX
		BACK:	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

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
			MOV DX,3006 OUT DX CALL DELAY MOV AL,0F MOV DX,3006 OUT DX CALL DELAY JMP BACK

Delay Program:

MEMORY LOCATION	OPCODE	LABEL	MNEMONICS
4500			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

b) 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

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:

i) Software

```
                ASSUME CS: CODE
                CODE SEGMENT
       START:   MOV AL, 90
                MOV DX, 3006
                OUT DX
        BACK:   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 BACK
CSEG ENDS
END START
```

ii) Hardware

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

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

Delay Program:

MEMORY	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 masmsoftware.
2. Write an alp program to find the largest number in an array using masmsoftware.

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: 14

PARALLEL AND SERIAL COMMUNICATION

a) Write a program to interface traffic light controller.

AIM:-

To write an ALP for Interfacing TRAFFIC LIGHT CONTROLLER with 8086 Microprocessor trainer kit 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
4	Traffic Light Controller Interfacing Kit		1

PROGRAM:

i) Software

```
        ASSUME CS: CODE
        CODE SEGMENT
START:  MOV AL,80H
        MOV DX,0FFE6H
        OUT    DX,AL
AGAIN:  MOV SI,2038H
NEXTST: MOV AL,[SI]
        MOV DX,0FFE0H
        OUT DX,AL
        INC SI
        ADD DX,2
        MOV AL,[SI]
        OUT DX,AL
        INC SI
        ADD DX,2
        MOV AL,[SI]
        OUT DX,AL
        INC SI
        CALL   DELAY
        CMP   SI,2056H
        JNZ NEXTST
        JMP SHORT AGAIN
CODE ENDS
END START
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
		START:	MOV AL, 80 H
			MOV DX, 0FFE 6 H
			OUT DX, AL
		AGAIN:	MOV SI,2038H
		NEXTST:	MOV AL,[SI]
			MOV DX,0FFE0H
			OUT DX,AL
			INC SI
			ADD DX,2
			MOV AL,[SI]
			OUT DX,AL
			INC SI
			CALL DELAY
			CMP SI,2056H
			JNZ NEXTST
			JMP SHORT AGAIN

Delay Program:

MEMORY LOCATION	OPCODE	LABEL	MNEMONICS
		DELAY:	MOV CX,0FFH
		DLY5:	PUSH CX
			MOV CX,03FFH
		DLY10:	NOP
			LOOP DLY10
			POP CX
			LOOP DLY5
			RET

RESULT:

Program for interfacing traffic light controller with 8086 Microprocessor trainer kit by using 8255 is verified and executed.

b) Write an ALP program to interface tone generator.

AIM:-

To write an ALP to Interface tone generator with 8086 Microprocessor trainer kit 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
4	Tone generator Interfacing Kit		1

PROGRAM:

i) Software

```
                ASSUME CS: CODE
                CODE SEGMENT
        START:  MOV AL,80H
                MOV DX,0FFE6H
                OUT    DX,AL
      GETKEY:   MOV SI,2100H
                CALL FAR 0FF00:0B1CH
                CMP AL,0FH
                JG GETKEY
                MOV BH,00H
                MOV BL,AL
                MOV CL,4FH
                MOV DX,0FFE4H
         FREQ:  MOV AL,00H
                OUT DX,AL
                MOV  CH,[BX][SI]
       NXTPL:   NOP
                NOP
                NOP
                NOP
                DEC CH
```

```
                 JNZ NXTPL
                 MOV AL,0FFH
                 OUT DX,AL
                 MOV CH,[BX][SI]
          NXTPH: NOP
                 NOP
                 NOP
                 NOP
                 DEC CH
                 JNZ NXTPH
                 DEC CL
                 JNZ FREQ
                 JMP SHORT GETKEY
             CODE ENDS
             END START
```

ii) Hardware

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
		START:	MOV AL,80H
			MOV DX,0FFE6H
			OUT DX,AL
		GETKEY:	MOV SI,2100H
			CALL FAR 0FF00:0B1CH
			CMP AL,0FH
			JG GETKEY
			MOV BH,00H
			MOV BL,AL
			MOV CL,4FH
			MOV DX,0FFE4H
		FREQ:	MOV AL,00H
			OUT DX,AL
			MOV CH,[BX][SI]
		NXTPL:	NOP
			NOP

MEMORY LOCATION	OPCODE	LABEL	MNEMONIC OPERAND
		NXTPH:	NOP NOP DEC CH JNZ NXTPL MOV AL,0FFH OUT DX,AL MOV CH,[BX][SI] NOP NOP NOP NOP DEC CH JNZ NXTPH DEC CL JNZ FREQ JMP SHORT GETKEY

RESULT:

Program to Interface tone generator with 8086 Microprocessor trainer kit by using 8255 is verified and executed.