DEPARTMENT OF COMPUTER SCIENCES & ENGINEERING

VISION:

Our vision is to emerge as a world class Computer Science and Engineering department through excellent teaching and a strong research environment that responds swiftly to the challenges of changing computer science technology and addresses technological needs of the stakeholders.

MISSION:

To enable our students to master the fundamental principles of computing and to develop in them the skills needed to solve practical problems using contemporary computer-based technologies and practices to cultivate a community of professionals who will serve the public as resources on state-of- the-art computing science and information technology.

COURSE OUTCOMES:

By the end of the course, the student will be able to:

1. Possessed a better command over the instruction of set of 8085 and 8086 microprocessor for programmatically deployment.
2. Demonstrate the interfacing of 8085 microprocessor with external I/O devices through 8255 PPI.
3. Analyze the internal communication of microprocessor with the external devices through the interrupts and working with various types of vector interrupts
4. Students will possess the knowledge to design and develop a working prototype with various simulators and emulators that they have used throughout the lab sessions.

PROGRAM OUTCOMES (POs):

Graduate Attribute1:	Engineering Knowledge
PO-1	Apply the knowledge of basic engineering sciences, humanities core engineering and computing concept in modelling and designing computer based systems.
Graduate Attribute2:	Problem Analysis
PO-2	Identify, analyze the problems in different domains and define the requirements appropriate to the solution.
Graduate	

Attribute3:	Design/Development of Solution
PO-3	Design, implement & test a computer based system, component or process that meet functional constraints such as public health and safety, cultural, societal and environmental considerations.
Graduate Attribute4:	Conduct Investigations of Complex Problems
PO-4	Apply computing knowledge to conduct experiments and solve complex problems, to analyze and interpret the results obtained within specified timeframe and financial constraints consistently.
Graduate Attribute5:	Modern Tool Usage
PO-5	Apply or create modern techniques and tools to solve engineering problems that demonstrate cognition of limitations involved in design choices.
Graduate Attribute6:	The Engineer and Society
PO-6	Apply contextual reason and assess the local and global impact of professional engineering practices on individuals, organizations and society.
Graduate Attribute7:	Environment and Sustainability
PO-7	Assess the impact of engineering practices on societal and environmental sustainability.
Graduate Attribute8:	Ethics
PO-8	Apply professional ethical practices and transform into good responsible citizens with social concern.
Graduate Attribute9:	Individual and Team Work
PO-9	Acquire capacity to understand and solve problems pertaining to various fields of engineering and be able to function effectively as an individual and as a member or leader in a team.
Graduate Attribute10:	Communication
PO-10	Communicate effectively with range of audiences in both oral and written forms through technical papers, seminars, presentations assignments, project reports etc.
Graduate Attribute11:	Project Management and Finance
PO-11	Apply the knowledge of engineering, management and financial principles to develop and critically assess projects and their outcomes in multidisciplinary areas.

Graduate Attribute12:	Life-long Learning
PO-12	Recognize the need and prepare oneself for lifelong self learning to be abreast with rapidly changing technology.

PROGRAM SPECIFIC OUTCOMES (PSOs):

1	Programming and software Development skills: Ability to acquire programming efficiency to analyze, design and develop optimal solutions apply standard practices in software project development to deliver quality software product.
2	Computer Science Specific Skills: Ability to formulate, simulate and use knowledge in various domains like data engineering, image processing and information and network security, artificial intelligence etc., and provide solutions to new ideas and innovations

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

A LABORATORY MANUAL

FOR

MICROPROCESSOR AND INTERFACING LAB

(CSE 227)

SEMESTER – II

Prepared by

1. G Gowripushpa (Asst. Professor)

2. G Santoshi. (Asst. Professor)

3. CH K Rupesh Kumar (Asst. professor).

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CYCLE	LIST OF EXPERIMENTS		CO's
I	8085 programs		1&4
II	8085 Interfacing with trainer kit		2 &3
	1	8255 study card scenarios	
	2	8255 Mode 0 and mode 1 scenarios using hard ware interrupt	
	3	Traffic lights(All Directions)	
	5	Hex keypad	
III	8085 Interfacing with PC		2 &3
	1	Steeper motor rotation (clock wise and anti clockwise)	
	2	Traffic lights(All Directions)	
	3	Logic controller (data transfer, 1's and 2's complement)	
IV	8086 programs		1&4

LIST OF INDUSTRY RELEVANT SKILLS:

1. To provide high quality academic programmers, training activities, research facilities and opportunities supported by continuous industry institute interaction.
2. Aimed at employability, entrepreneurship, leadership and research aptitude among students.

LABORATORY RULES

General Rules of Conduct in Laboratories:

1. You are expected to arrive on time and not depart before the end of a laboratory.
2. You must not enter a lab unless you have permission from a technician or lecturer.
3. You are expected to comply with instructions, written or oral, that the laboratory Instructor gives you during the laboratory session.
4. You should behave in an orderly fashion always in the lab.
5. You must not stand on the stools or benches in the laboratory.
6. Keep the workbench tidy and do not place coats and bags on the benches.
7. You must ensure that at the end of the laboratory session all equipment used is stored away where you found it.
8. You must put all rubbish such as paper outside in the corridor bins. Broken components should be returned to the lab technician for safe disposal.
9. You must not remove test equipment, test leads or power cables from any lab without permission.
10. Eating, smoking and drinking in the laboratories are forbidden.
11. The use of mobile phones during laboratory sessions is forbidden.
12. The use of email or messaging software for personal communications during laboratory sessions is forbidden.
13. Playing computer games in laboratories is forbidden.

SPECIFIC SAFETY RULES FOR LABORATORIES:

1. You must not damage or tamper with the equipment or leads.
2. You should inspect laboratory equipment for visible damage before using it. If there is a problem with a piece of equipment, report it to the technician or lecturer. DONOT return equipment to a storage area
3. You should not work on circuits where the supply voltage exceeds 40 volts without very specific approval from your lab supervisor. If you need to work on such circuits, you should contact your supervisor for approval and instruction on how to do this safely before commencing the work.
4. Always use an appropriate stand for holding your soldering iron.
5. Turn off your soldering iron if it is unlikely to be used for more than 10minutes.
6. Never leave a hot soldering iron unattended.
7. Never touch a soldering iron element or bit unless the iron has been disconnected from the mains and has had adequate time to cool down.
8. Never strip insulation from a wire with your teeth or a knife, always use an appropriate wire stripping tool.
9. Shield wire with your hands when cutting it with a pliers to prevent bits of wire flying about the bench.

GUIDELINES TO TEACHERS

1. Faculty must verify the observations and records before assigning the system.
2. Faculty must verify Students Id cards before entering into Laboratory
3. Faculty must take the attendance at the starting and ending of the lab time period.

SESSIONAL MARKS : 50 MARKS

1. Daily Evaluation (Includes Record, Observation & regular performance) – 25 marks
2. Attendance – 5 marks
3. Internal Exam – 20 marks

DAILY EVALUATION (25 MARKS)

1. Every Student must execute a minimum set of sample programs to secure 60% of marks in Daily Evaluation i.e. 15 Marks and to appear in external examination.
2. In addition to that if a student finishes the minimum set and 5 programs from an additional set of programs would secure 80% of marks in Daily Evaluation i.e. 20 Marks.
3. If a student finishes all the programs in both the set s will secure 100% of marks in Daily Evaluation

INTERNAL EXAM (20 MARKS)

1. Every student is given 2 questions in the internal exam out of which the difficulty level of 1 questions is easy / medium and 1 questions of difficulty level is high
2. Each easy / medium level question carries 20% of marks and difficulty level question carries 30% of marks

EXTERNAL EXAM (50 MARKS)

1. Viva voce – 10 marks
2. Write up + Execution – 40 marks

WRITE UP + EXECUTION (40 MARKS)

1. Every student is given 2 questions in the external exam out of which the difficulty level of 1 questions is easy / medium and 1 questions of difficulty level is high
2. Each easy / medium level question carries 30% of marks and difficulty level question carries 20% of marks.

INSTRUCTIONS TO STUDENTS:

1. Students should use computer related components smoothly
2. Students should not carry other items into lab.
3. Students must wear the dress code and ID cards.
4. Every student is given 4 questions in the external exam out of which the difficulty level of 2 questions is easy / medium and 2 questions of difficulty level is high

5. Each easy / medium level question carries 30% of marks and difficulty level question carries 20% of marks.

GUIDELINES TO LAB PROGRAMMERS:

1. Lab Programmers must verify All the Systems and trainer kits whether they are working properly or not.
2. Lab Programmers must verify all the other equipments.

LAB RUBRICS

NAME OF THE PROGRAMME: II/IV B.TECH

SESSION DURATION: 3 HOURS

SEMESTER: II

TITLE && COURSE CODE: MICROPROCESSOR AND INTERFACING LAB (CSE 227)

Key Performance Criteria(KPC) (25 pts)	**4-Very Good**	**3-Good**	**2-Fair**	**1-Need to improve**
Aim & apparatus / equipment details (3)	The student covers all major aspects of carryin out the experiment and the results.(3)	The student covers most of the major aspects of the experiment, some minor details are missing.(3)	The student misses one or more major aspects of the objective.(2)	Several major aspects of the experiment are missing; student displays a lacl of understandin about the objective.(1)
Experimental Procedure/ algorithm/ flow chart/ analysis (5)	The program logic is explained clearly and th details are covered(5)	The program logic is explained, some minor details are missing in the observation.(4	Missing some logic details both in explaining the procedure and ir observation. (3)	Missing several important experimental details. (2)
Experimental Results(3)	All calculations, opcodes address values are written and contain titles/captions. (3)	All calculations, opcodes, address values are written bu some have minor problems or could still be improved.(3)	Most of calculations, opcodes, address values are documented still missing some important or required features. (2)	calculations, opcodes, address value contain error or are poorly constructed, incorrect calculations, units missing or incorrect, etc.(1)
Test cases verification(2)	All important conclusions have been clearly made, student shows good understanding.(2)	All important conclusions have been presented, could be bette stated(2)	Conclusions regarding major points is presented, but many are misstated, indicating a lack	Conclusions missing or missing the important points(1)

			of understanding(1	
Courtesy , safety and ethics (based on physical observation) (2)	While conducting the procedure, the student is in uniform, always respectful of others, mindful of safety, and leaves the area clean. (2)	While conducting the procedure, the student is in uniform, many times respectful of others, many times mindful of safety, and leaves the area clean only after being reminded.(2)	While conducting the procedure, the student is in partial uniform, sometimes respectful of others, sometimes mindful of safety and leaves the area clean only after being reminded.(1)	While conducting the procedure the student is not in uniform, not respectful of others, not mindful of safety, and leaves the area messy even after being reminded.(1)
Presentation of record/ documentation (5)	Submitted on time, presented / communicated opcodes, memory address description, outputs etc. are excellently presented. (5)	Submitted on time, presented / communicated neatly the, memory address, opcodes, description, outputs, equations etc.(4)	Submitted on time, presented / communicated memory address opcodes, description, outputs etc. not so neatly but in a readable manner(3)	Submitted on time, presented / communicate memory address, opcodes, description, outputs etc. with ambiguity/no in a readable manner.(2)
Oral Presentation (Viva)(5)	Their knowledge related to the experiment is excellent.(5)	Their knowledge related to the experiment is good.(4)	Their knowledge related to the experiment is average.(3)	Their knowledge related to the experiment is poor.(2)

CYCLE I: 8085 PROGRAMS

EXPERIMENT - 1: 1's Complement and 2's Complement

To write an assembly language program to find one's compliment and two's compliment of 8bit or 16 bit data

Practical significance:

The one's compliment of a given operand can be obtained by inverting 1's to 0's and vice versa. The CMA instruction complements (inverts) the contents of an operand register or a memory location bit by bit. The two's compliment is obtained by adding 1 to one's compliment result. This can be done by using ADD instruction or just by using INR instruction.

Relevant Program Outcomes:

PO1,PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

1. To apply the concepts in the design of microprocessor/microcontroller based systems in real time applications
2. Develop code, debug, test and execute various assembly language programs using 8085 instruction set.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required:

1. 8085 Microprocessor Kit
2. +5V Power supply

Precautions

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections
3. Handle the Trainer kit carefully.

Algorithm/circuit/Diagram/Description:

1. Load memory location of data 8050H in H-L registers pair.
2. Move data into accumulator
3. Complement accumulator
4. Store the result in memory location 8051H

Test cases:

1's Complement

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8500	**F0H**	**8003**	**0FH**

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**55H**	**8003**	**AAH(LSB)**
8001	**60H**	8004	9FH(MSB)

2's Complement

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8500	**F0H**	**8003**	**10H**

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**55H**	**8003**	**ABH(LSB)**
8001	**60H**	8004	A0H(MSB)

Sample output:

Input Data		**Result**	
Memory Location	**Data**	**Memory Locatior**	**Data**
8000	**04H**	**8003**	**0BH(1's complement)**
		8004	0CH(2's complement)

Practical Related Questions:

1. Draw the flow chart for obtaining one's and two's compliment for a given operand?
2. Do the contents of flag register will change after compliment operation?
3. What is the function of XCHG instruction?
4. How you can load 16-bit data in 8500H and 8501H memory locations?
5. What is the difference between LSLD and SHLD instructions?
6. What is difference between byte and word?
7. What is a Register? What is the maximum size of any register in 8085?
8. What is an addressing mode? What are different types of Addressing modes supported
9. Define Opcode and Operand?
10. What is Physical Address of Instruction?
11. What is the purpose of the following instructions:
    a.MOV b. LXI H c. INR

Exercise Questions:

1. The number is stored in memory location 8050H and the one's complement of number will be stored in location 8051H. Assume the program memory starts from 8000H.
2. Formulate an Assembly Language Program to find one's compliment and two's compliment of a given 8-bit operand.
3. Write an Assembly Language Program to find one's compliment and two's compliment of a given 16-bit operand

EXPERIMENT - 2: ADDITION & SUBTRACTION

To add two 8 bit or 16-bit numbers residing in memory and store the result in memory

Practical significance:

We use ADI and ADD instruction for addition . ADD instruction adds an immediate data or contents of a memory location specified in the instruction or a register (source) to the contents of another register (destination) or memory location. The result is in the destination operand. However, both the source and destination operands cannot be memory operands. That means memory to memory addition is not possible. Also the contents of the segment registers cannot be added using this instruction. All the condition code flags are affected depending upon the result

Relevant Program Outcomes :

PO1,PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

1. Gain Knowledge in working with assembly language programming, finding the Opcodes for the assembly language program.
2. Identification of address locations.
3. The steps to execute an 8 bit addition using indirect addressing.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required:

1. 8085 Microprocessor Kit
2. +5V Power supply

Precautions

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections
3. Handle the Trainer kit carefully.

Algorithm/circuit/Diagram/Description:

1. Initialise the memory location of first number in HL register pair.
2. Move first number/data into accumulator
3. Increment the content of HL register pair to initialise the memory location of second data
4. Add the second data with accumulator
5. Store the result in memory location 8003H

Test cases:

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**02H**	**8003**	**07H(SUM)**
8001	**05H**	8004	00H(Carry)

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**F3H**	**8003**	**E5H(SUM)**
8001	**F2H**	8004	01H(Carry)

Memory Location	**Data**	**Memory Location**	**Data**
8000	**05H**	**8003**	**03H(SUB)**
8001	**02H**	8004	00H(borrow)

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**F3H**	**8003**	**01H(SUB)**
8001	**F2H**	8004	00H

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**03H**	**8003**	**FEH(SUB)**
8001	**05H**	8004	01H(borrow)

Sample output:

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**26H**	**8003**	**88H(SUM)**

8001	**62H**		

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**62H**	**8003**	**3CH(SUB)**
8001	**26H**	8004	01H(borrow)

Practical Related Questions:

1. What is the purpose of the following instructions:

    a. ADD
    b. ADC
    c. ADI
    d. e.INX,
    e. f.SUB
    f. SBB
    g. h.SBI

2. How you can read a data from a memory location?
3. What are flags available in 8085 ?
4. What is the function of RESET key of a 8085 microprocessor kit
5. What is the function of DAD D instruction?

Exercise Questions:

1. Consider the first number 26H is stored in memory location 8000H and the second number 62His stored in memory location 8001H. The result after addition of two numbers is to be stored in the memory location 8002 H. Assume program starts from memory location 8500H.
2. The first number F2H is stored in memory location 8501H and the second number 2FH is storedin memory location 8502H. The result after addition will be stored in the memory location 8503 H and 8504H. Consider program is written from memory location 8000H.
3. The first 16-bit number is stored in 8501 and 8502 memory locations. The second 16-bit numberis stored in 8503 and 8504 memory locations. After addition result will be stored from 8505 and 8506 memory locations. Assume program starts from memory location 8000H.
4. The first number F2H is stored in memory location 8501H and the second number 2FH is storedin memory location 8502H. The result after subtract will be stored in the memory location 8503 H and 8504H. Consider program is written from memory location 8000H.
5. The first 16-bit number is stored in 8501 and 8502 memory locations. The second 16-bit numberis stored in 8503 and 8504 memory locations. After subsract the result will be stored from 8505 and 8506 memory locations. Assume program starts from memory location 8000H.
6. Write an assembly language program to add two 8-bit numbers using Q3. (USing indirect addressing and direct addressing modes).
7. Addition of two 8-bit operands stored at memory locations 8500 & 8501.

8. Write an assembly language program to add two 8-bit numbers using Q3. (USing indirect addressing and direct addressing modes).
9. Addition of two 8-bit operands stored at memory locations 8500 & 8501 and 8052& 8053.

EXPERIMENT - 3: MULTIPLICATION& DIVISION

To perform multiplication and division arithmetic operations over two 8 bit or 16-bit numbers.

Practical significance:

To solve a multiplication problem through repeated addition, we repetitively group and add the same number again and again to find the answer.
Repeated Subtraction is a method that **subtracts the equal number of items from** a group, also known as division. Using this method, the same number is subtracted repeatedly from another larger number until the remainder is zero, or smaller than the number being subtracted.

Relevant Program Outcomes :

PO1,PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills :

1. Gain Knowledge in working with assembly language programming , finding the Opcodes for the assembly language program.
2. Identification of address locations

Prerequisites :

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required :

1. 8085 Microprocessor Kit
2. +5V Power supply

Precautions

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections
3. Handle the Trainer kit carefully.

Algorithm/circuit/Diagram/Description:

1. Start the microprocessor
2. Get the 1st 8 bit numbers

3. Move the 1st 8it number to register ‘B’
4. Get the 2nd 8 bit number
5. Move the 2nd 8 bit number to register ‘C’
6. Initialise the accumulator as zero
7. Initialise the carry as zero
8. Add both register ‘B’ value as accumulator
9. Jump on if no carry
10. Increment carry by 1 if there is carry
11. Decrement the 2nd value and repeat from step 8, till the 2nd value becomes zero.
12. Store the multiplied value in accumulator
13. Move the carry value to accumulator
14. Store the carry value in accumulator

Test cases:

Multiplication

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**05H**	**8003**	**0AH(LSB)**
8001	**02H**	8004	00H(MSB)

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**55H**	**8003**	**E4H(LSB)**
8001	**60H**	8004	0CH(MSB)

DIVISION

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**03H(divisor**	**8003**	**01H(Quotient)**
8001	**05H(dividend)**	8004	02H(Rem)

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**05H(divisor**	**8003**	**02H(Quotient)**
8001	**0AH(dividend)**	8004	00H(Rem)

Sample output:

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**62H**	**8003**	**4CH(LSB)**
8001	**26H**	8004	06H(MSB)

Practical Related Questions:

1. What is the function of CMP instruction? And how the flags are affected.
2. What is the function of JNC instruction?
3. What is the difference between conditional and unconditional jump instruction?
4. What is the function of STA 2500$ instruction?

Exercise Questions:

1. Write an assembly language program to multiply and division two 8-bit numbers using the (Use immediate and direct addressing modes)
2. Multiplication and Division of two 8-bit operands stored at memory locations 8100 & 8101.

EXPERIMENT - 4: BINARY CODED DECIMAL OPERATIONS

To perform BCD addition of two 8 bit or 16-bit numbers.

Practical significance :

A decimal number contains 10 digits (0-9). Now the equivalent binary numbers can be found out of these 10 decimal numbers. In case of **BCD** the binary number formed by four binary digits, will be the equivalent code for the given decimal digits.

Example:
Let, 0101 is added with 0110.

```
  0101
+ 0110
  1011  → Invalid BCD number
+ 0110  → Add 6
---------
0001 0001 → Valid BCD number
```

Relevant Program Outcomes :

PO1,PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

1. Gain Knowledge in working with assembly language programming , finding the Opcodes for the assembly language program.
2. Identification of address locations.

Prerequisites :

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required :

1. 8085 Microprocessor Kit
2. +5V Power supply

Precautions

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections
3. Handle the Trainer kit carefully.

Algorithm/circuit/Diagram/Description:

1. Initialise the memory location of first number in HL register pair.
2. Load the first number in accumulator
3. Increment the content of HL register pair to initialise the memory location of second data
4. Addition of the content of second memory location with first data
5. Decimal adjustment of result
6. Store the result in memory location 8002H

Test cases:

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**22H**	**8003**	**89H(LSB)**
8001	**67H**	8004	00H(MSB)

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**99H**	**8003**	**00H(LSB)**
8001	**01H**	8004	01H(MSB)

Sample output:

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8000	**62H**	**8003**	**88H(LSB)**
8001	**26H**	8004	00H(MSB)

Practical Related Questions:

1. What is the function of ANI instruction?
2. What is the function of DAA instruction?
3. Do the contents of flag register will change after masking of bits?
4. Do the contents of flag register will change after DAA of bits?

Exercise Questions:

1. Using Assembly Language Programming, Write a series of processor instructions to perform the following conversions:
    a. Unpacked BCD number to packed BCD number and vice versa
    b. Unpacked BCD number to Binary number
2. Using Assembly Language Programming, Write a series of processor instructions to:
    a) mask off least significant nibble bits of a byte length number
    b) mask off most significant nibble bits of a byte length number
3. Write an assembly language program to add 2 8- bit BCD number
4. Write an assembly language program to add 2 16- bit BCD number

EXPERIMENT - 5: LOOP PROGRAMS

Write 8085 Assembly language program to find the factorial of an 8-bit number.

Practical significance :
There is no direct instruction to perform multiplication. We need to perform repetitive addition to get the result of the multiplication. In each step we are decreasing the value of B and multiply with the previous value of B. We are repeating these steps until B reaches 1. and B – 1 to 0. Thus the factorial is generated.

Relevant Program Outcomes :
PO1,PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills :
1. Gain Knowledge in working with assembly language programming , finding the Opcodes for the assembly language program.
2. Identification of address locations.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required :
1. 8085 Microprocessor Kit
2. +5V Power supply

Precautions
1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections
3. Handle the Trainer kit carefully.

Algorithm/circuit/Diagram/Description:

1. Load the data into register B
2. To start multiplication set D to 01H
3. Jump to step 7
4. Decrements B to multiply previous number
5. Jump to step 3 till value of B>0
6. Take memory pointer to next location and store result
7. Load E with contents of B and clear accumulator
8. Repeatedly add contents of D to accumulator E times
9. Store accumulator content to D
10. Go to step 4

Test cases:

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8100	05H	9000H	78H
81001	04H	9001H	18H

Practical Related Questions:

1. What is a counter?
2. Explain how counters are used in loop instructions?
3. What is meant by time delay?
4. Explain how to calculate execution delay or delay sub-routine?
5. Difference between time delay in loop and nested loop

Exercise Questions:

1. To write an ALP to implement a counter to count from '00 – 99' (UPCOUNTER) in BCD by Using a subroutine to generate a delay of one second between the counts.
2. To write an ALP to find the sum of fist n natural no.
3. To write an ALP to find the Fibonacci number

EXPERIMENT - 6: ARRAY PROGRAMS

Write an assembly language program to store the data in memory and perform the operations like find the largest number in an array, sum of all the array elements.

Practical significance :

The block of data is stored in continues memory location and can be accessed using memory pointer HL register. The loop can be implemented by initializing a register as Counter and applying the decrement and jump operation to generate a loop operation. The logic can be applied for transferring of block of data from one memory location to another, to find the smallest /Largest of a number, check even or odd etc.

Relevant Program Outcomes :

PO1,PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills :

1. Gain Knowledge in working with assembly language programming , finding the Opcodes for the assembly language program.
2. Identification of address locations.

Prerequisites :

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required :

1. 8085 Microprocessor Kit
2. +5V Power supply

Precautions

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections
3. Handle the Trainer kit carefully.

Algorithm/circuit/Diagram/Description:

1. Load the address of the first element of the array in HL pair
2. Move the count to B – reg.
3. Increment the pointer
4. Get the first data in A – reg.
5. Decrement the count.
6. Increment the pointer
7. Compare the content of memory addressed by HL pair with that of A - reg.
8. If Carry = 0, go to step 10 or if Carry = 1 go to step 9
9. Move the content of memory addressed by HL to A – reg.
10. Decrement the count
11. Check for Zero of the count. If ZF = 0, go to step 6, or if ZF = 1 go to next step.
12. Store the largest data in memory.
13. Terminate the program.

Test cases:

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8100	05H	9000H	7BH(Largest)
8101	**67H**		
8102	**50H**		
8103	**60H**		
8104	**6AH**		
8105	**7BH**		

Input Data		Result	
Memory Location	**Data**	**Memory Location**	**Data**
8100	05H	9000H	50H(smallest)
8101	**67H**		
8102	**50H**		
8103	**60H**		
8104	**6AH**		
8105	**7BH**		

Practical Related Questions:

1. What is the function of JNC and JC instruction?
2. How data is extracted from continuous memory location.
3. How do you apply loops in 8085 programming.
4. Differentiate between program counter and Data pointer?

Exercise Questions:

1. The data are stored at location 8001H onwards. The 8000H is containing the size of the block. After executing this program, it will return the largest/smallest number and store it at location 9000H.
2. Transfer a block of data from memory location 8050 to 8500.
3. Find the sum of array of data in a block of continuous memory location the data are stored at location 8002H to 8007H. The 8000H is containing the size of the block, and 8001H is holding the key value to search. After executing this program, it will return the address of the data where the item is found and store the address at location 9000H and9001H. If the item is not found, it will return FFFFH.
4. Write an assembly language program in 8085 microprocessor to separate odd and even numbers from the given list of 50 numbers. Store odd nos in another list starting from memory location 2100H. Store even nos in another list starting from memory location 2200H. Starting address of the list is 2000H.

EXPERIMENT - 7: SORTING PROGRAMS

To write an assembly language program to arrange a given series of numbers in ascending and descending order.

Practical significance:

To arrange the given numbers in ascending and descending order, the bubble sorting method is used. Initially the first number of the series is compared with the second one. If the first number is greater than second, exchange their positions in the series otherwise leave the position unchanged. Then compare the second number in the recent form of the series with third and repeat the exchange part that you are carried out for the first and second number, and for all the remaining number of the series. Repeat this procedure for complete series (n-1) times. After n-1 iterations you will get the largest number at the end of the series. Again start from the first number of the series. Repeat the same procedure

right from the first element to the last element. After n-2 iteration you will get the second highest number at the last but one place in the series. Repeat this till the complete series is arranged in ascending order.

Relevant Program Outcomes :

PO1,PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

1. Gain Knowledge in working with assembly language programming, finding the Opcodes for the assembly language program.
2. Identification of address locations.

Prerequisites :

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required :

1. 8085 Microprocessor Kit
2. +5V Power supply

Precautions

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections
3. Handle the Trainer kit carefully.

Algorithm/circuit/Diagram/Description:

1. Load size of list in C register and set D register to be 0
2. Decrement C as for n elements n-1 comparisons occur
3. Load the starting element of the list in Accumulator
4. Compare Accumulator and next element
5. If accumulator is less than or equal to the next element jump to step 8
6. Swap the two elements
7. Set D register to 1
8. Decrement C
9. If C>0 take next element in Accumulator and go to point 4
10. If D=0, this means in the iteration, no exchange takes place consequently we know that it won't take place in further iterations so the loop in exited and program is stopped
11. Jump to step 1 for further iterations

Test cases:

Ascending order

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8100	05H	8101	50H

8101	**67H**	8102	60H
8102	**50H**	8103	67H
8103	**60H**	8104	6AH
8104	**6AH**	8105	7BH
8105	**7BH**		

Descending order

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8100	05H	8101	7BH
8101	**67H**	8102	6AH
8102	**50H**	8103	67H
8103	**60H**	8104	60H
8104	**6AH**	8105	50H
8105	**7BH**		

Practical Related Questions:

1. What is the similarity and difference between Subtract and Compare Instruction?
2. What is the use of PUSH and POP instruction?
3. Differentiate between program counter and Data pointer?

Exercise Questions:

1. Compose an Assembly Language Program to Sort the given byte length numbers in ascending order and also in descending order using bubble sort or insertion sort

EXPERIMENT - 7: CODE CONVERSION

To convert two BCD numbers in memory to the equivalent HEX number using 8085 instruction set

Practical significance:

Code conversion allows user to translate a number that is representated using one coding system to other coding system. The code conversion involves operations like : 1) Binary to BCD 2) BCD to Binary 3) BCD to Hex 4) Hex to BCD 5) BCD to Seven Segment 6) Binary to ASCII 7) ASCII to Binary

Relevant Program Outcomes :

PO1,PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills :

1. Gain Knowledge in working with assembly language programming , finding the Opcodes for the assembly language program.
2. Identification of address locations.

Prerequisites :

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required :

1. 8085 Microprocessor Kit
2. +5V Power supply

Precautions

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections
3. Handle the Trainer kit carefully.

Algorithm/circuit/Diagram/Description:

1. We get the value from the user.
2. Then we take the Most Significant Digit (MSD).
3. We multiply MSD by 10, using repeated addition.
4. Then we add the Least Significant Digit (LSD) to the result obtained in previous step.
5. And finally the value is stored in the next memory location.

Test cases:

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8100	0AH	8101	10H

Input Data		**Result**	
Memory Location	**Data**	**Memory Location**	**Data**
8100	05H	8101	05H

Practical Related Questions:

1. Write about BCD system.
2. How will you convert BCD to hexadecimal?
3. What is the use if INX instruction?
4. Write various JMP operations?
5. How will you convert hexadecimal to BCD?

Exercise Questions:

1. To convert two HEX numbers in memory to the equivalent BCD number using 8085 instruction set.
2. To convert two HEX numbers in memory to the equivalent ASCII number using 8085 instruction set.
3. To convert two ASCII numbers in memory to the equivalent HEX number using 8085 instruction set.

CYCLE II : INTERFACING WITH 8085 TRAINER KIT

EXPERIMENT -1: 8255 STUDY CARD SCENARIOS (I/O MODE OPERATIONS)

Mode-0 Operation:

Write an assembly language program to configure 8255A such that Port A as an input and Port B as an output in Mode 0 operation.

Practical significance:

Port A, B and C can work either as input function or as output function. Two 8-bit ports and two 4-bit ports available. The outputs are latched but the inputs are not latched. It has interrupt handling capability. No handshaking is required.

Relevant Program Outcomes:

PO1, PO3 – Moderate
PO2, PO5, PO9, PO10- Weak

Competency and practical skills:

1. To apply the concepts in the design of microprocessor/microcontroller based systems in real time applications
2. Develop code, debug, test and execute various assembly language programs using 8255A PPI & 8085 instruction set.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required:

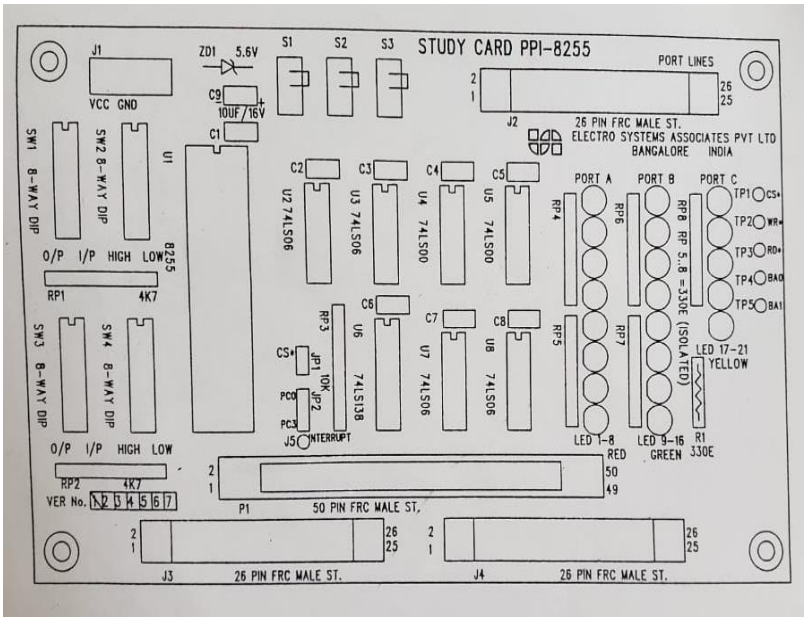
1. 8085 Microprocessor Kit
2. +5V Power supply
3. 8255A study card
4. Connectors for study card

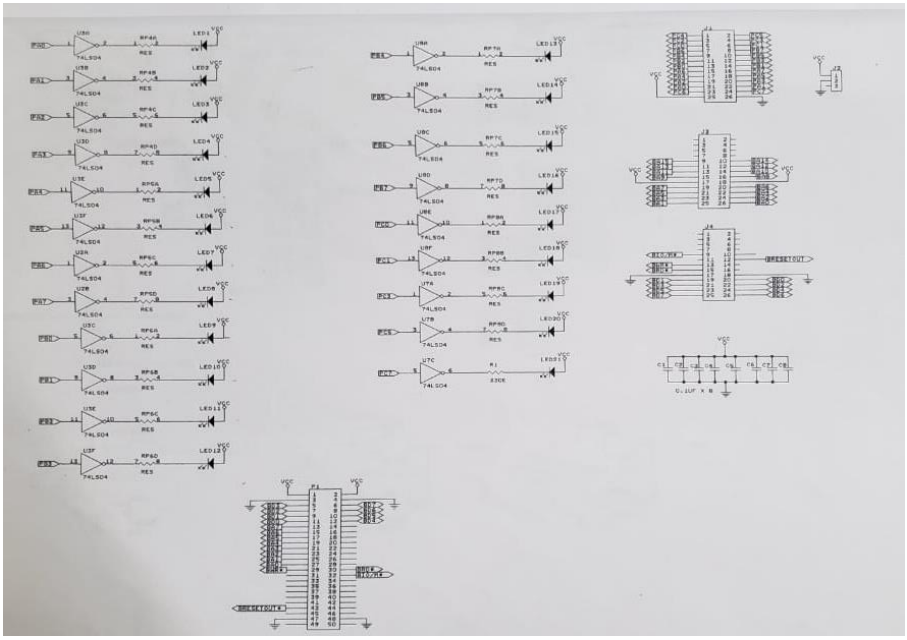
Precautions:

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections.
3. Properly connect the study card using connectors.

4. Switch off power of the trainer kit while connecting the study card.
5. Press Reset after giving power to the trainer.
6. Handle the Trainer kit carefully.

Algorithm/circuit/Diagram/Description:





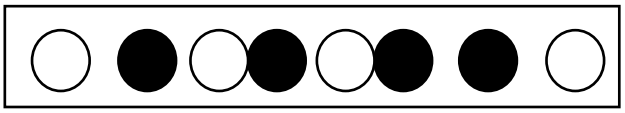
Test cases:

Input: PORT A: 55A

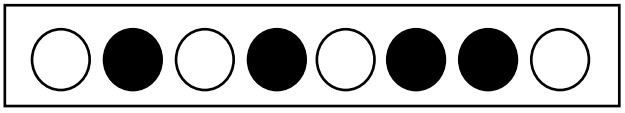
Output: Port B: 55A

Sample Result:

Port A:



Port B:



Practical Related Questions:

1. What is the feature of mode 0?
2. What are the different Modes of 8255PPI?
3. Bi-directional data transfer is available in which mode?
4. Explain about Port C pins?

Exercise Questions:

1. Configure 8255A such that Port A and Port B as an output port in Mode 0. Execute the program at 8000H operation.
2. Write an ALP code to find the complemented output on Port B in Mode 0.
3. Write an ALP code to find the 2's Complement output on Port B in Mode 0.

EXPERIMENT -2: 8255 STUDY CARD SCENARIOS (BSR MODE OPERATIONS)

Write an assembly language program to configure 8255A in BSR Mode. Write BSR control word to set bits PC7 and reset them after certain delay.

Practical significance:

This mode is used to set or reset the bits of port C only, and selected when the most significant bit (D7) in the control register is 0. This mode affects only one bit of port C at a time because, as user set the bit, it remains set until and unless user changes it. User needs to load the bit pattern in control register to change the bit.

Relevant Program Outcomes:

PO1, PO3 – Moderate
PO2, PO5, PO9, PO10- Weak

Competency and practical skills:

1. To apply the concepts in the design of microprocessor/microcontroller based systems in real time applications
2. Develop code, debug, test and execute various assembly language programs using 8255A PPI & 8085 instruction set.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

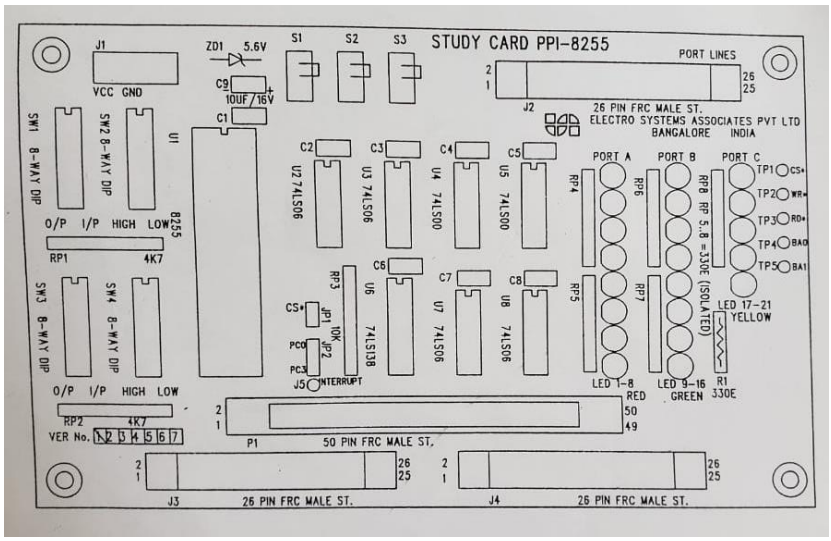
Resources required:

1. 8085 Microprocessor Kit
2. +5V Power supply
3. 8255A study card
4. Connectors for study card

Precautions:

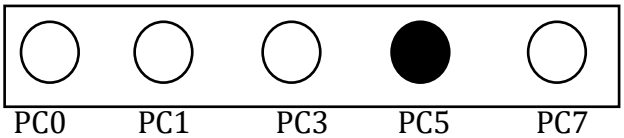
1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections.
3. Properly connect the study card using connectors.
4. Switch off power of the trainer kit while connecting the study card.
5. Press Reset after giving power to the trainer.
6. Handle the Trainer kit carefully.

Algorithm/circuit/Diagram/Description:

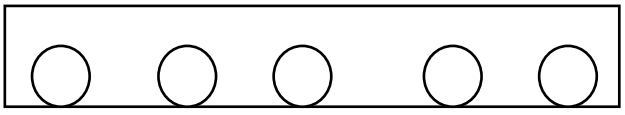


Test cases:

Port C: Set

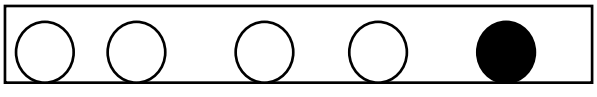


Port C : Reset

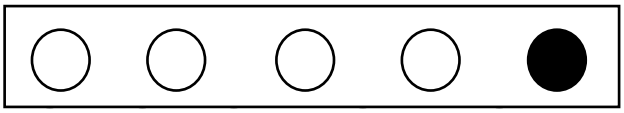


Sample output:

Port C: Set



Port C : Reset



Practical Related Questions:

1. In BSR mode, only port C can be used to?
2. How many bits can be affected with BSR control word register?
3. Does BSR control word affect the I/O mode?

Exercise Questions:

1. Write an ALP to configure 8255A in BSR Mode to Set PC3 bit.
2. Write an ALP to display even and odd lights in Port C alternatively with some time delay.

EXPERIMENT -3: 8255 STUDY CARD SCENARIOS (I/O MODE OPERATIONS)

Write an assembly language program initializes 8255A Port A as input in Mode-1 and Port B as an output in Mode-0. Read through input & output to the data field of the trainer display, press S2 switch to simulate STB_A*signal. (Using Interrupt signal RST 5.5).

Practical significance:

Both Group A and Group B can operate in Mode 1, either together, or individually, with each port containing an 8-bit latched Input or Output data port, and a 4-bit port which is used for control and status of the 8-bit port.Strobe is an active low input signal for 8255 Programming and Operation and output signal for the input device. The input device activates this signal to indicate CPU that the data to be read is already sent on the port lines of 8255 port. Upon activation of this signal 8255 loads the data from the input port lines into the input buffer of that port. Input Buffer full (IBF) is active high output signal for 8255 and an input signal for input device. This signal is generated by 8255 Programming and Operation in response to STB signal as an acknowledgment to input device.Interrupt Request (INTR) is an active high output signal generated by 8255. A 'high' on this output can be used to interrupt the CPU when an input device is requesting service. The 8255 Programming and Operation sets the INTR when STB signal is 'one', IBF signal is 'one' and INTE is 'one', indicating CPU that the data from the input device is available in the input buffer. This signal is reset by the falling edge of the RD signal i.e. immediately after reading the data from the input buffer.

Relevant Program Outcomes:

PO1, PO3 – Moderate
PO2, PO5, PO9, PO10- Weak

Competency and practical skills :

1. To apply the concepts in the design of microprocessor/microcontroller based systems in real time applications
2. Develop code, debug, test and execute various assembly language programs using 8255A PPI & 8085 instruction set.

Prerequisites :

Basic knowledge of Digital Logic Design and Computer Organization.

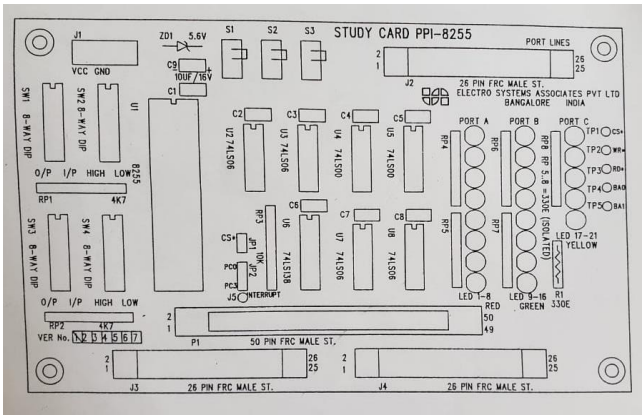
Resources required :

1. 8085 Microprocessor Kit
2. +5V Power supply
3. 8255A study card
4. Connectors for study card

Precautions :

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections.
3. Properly connect the study card using connectors.
4. Switch off power of the trainer kit while connecting the study card.
5. Press Reset after giving power to the trainer.
6. Handle the Trainer kit carefully.

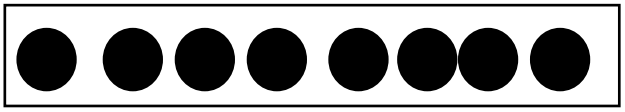
Algorithm/circuit/Diagram/Description:



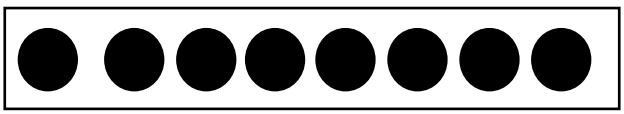
Test cases:

Run the program for different strobe signals.
Press S1 switch to simulate STB_B* signal.

Port A:



Port B:

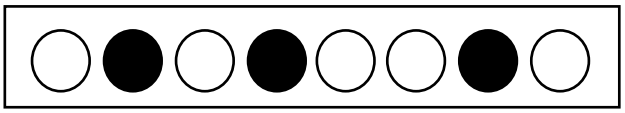


Data Field of 8255 kit:

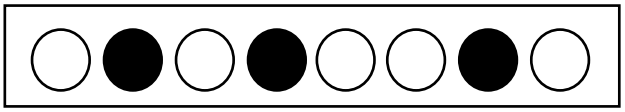
F	F

Sample output:

Port A:



Port B:



Data Field of 8255 kit:

A	2

Practical Related Questions:

1. If the value of the pin STB (Strobe Input) falls to low level, then?
2. The pulse width of the signal INIT at the receiving terminal must be more than?
3. What are the signals that are provided to maintain proper data flow and synchronization between the data transmitter and receiver?
4. The feature of mode 2 of 8255 is?

Exercise Questions:

1. Write an assembly language program initializes 8255A Port A as input in Mode-1 and Port B as an output in Mode-0. Read through input & output to the data field of the trainer display, press S2 switch to simulate STB_A*signal. (Using Interrupt signal RST 6.5).
2. Write an assembly language program initializes 8255A Port B as input in Mode-1 and Port A as an output in Mode-0. Read through Port B & output to Port A aswell as the data field of the trainer display, press S1 switch to simulate STB_A*signal.
3. Write an assembly language program initializes 8255A Port A as input in Mode-1 and Port B as an output in Mode-0. Read through input & output to the data field of the trainer display, press S2 switch to simulate STB_A*signal. (Using Interrupt signal RST 7.5).

EXPERIMENT - 4: 4X4 HEXKEY BOARD MATRIX

To write an assembly program to take input from the Hex keypad and display the numbers on 8085μp

Practical significance :

The hex keypad is a peripheral that connects to the DE2 through JP1 or JP2 via a 40-pin ribbon cable. It has 16 buttons in a 4 by 4 grid, labelled with the hexadecimal digits 0 to F. Internally; the structure of the hex keypad is very simple. Wires run in vertical columns (we call them C0 to C3) and in horizontal rows (called R0 to R3). These 8 wires are available externally, and will be connected to the lower 8 bits of the port. Each key on the keypad is essentially a switch that connects a row wire to a column wire. When a key is pressed, it makes an electrical connection between the row and column.

Relevant Program Outcomes :

PO3 – Moderate
PO1, PO2, PO9, -Weak

Competency and practical skills :

1. To apply the concepts in the design of microprocessor/microcontroller based systems in real time applications
2. Develop code, debug, test and execute various assembly language programs and various peripherals using 8255A PPI & 8085 instruction set.

Prerequisites :

Basic knowledge of Digital Logic Design and Computer Organization.

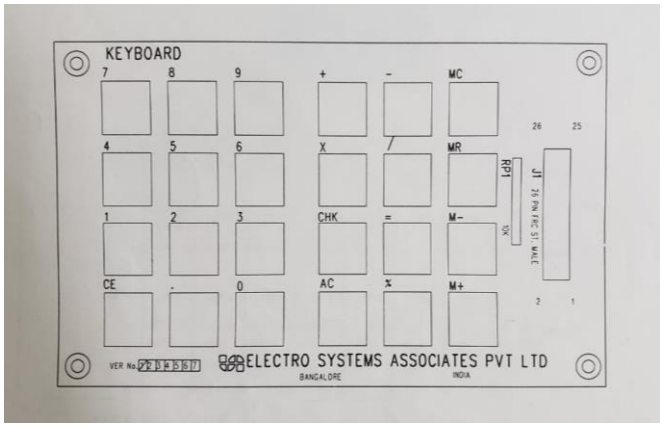
Resources required :

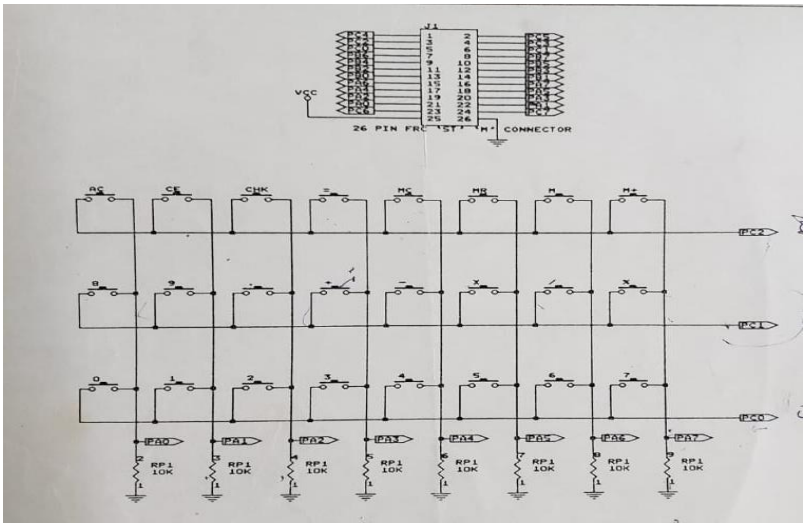
1. 8085 Microprocessor Kit
2. +5V Power supply
3. Key board Interface
4. Flat ribbon cable (FRC).

Precautions :

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections.
3. Properly connect the Key board Interface using connectors.
4. Switch off power of the trainer kit while connecting the Key board Interface.
5. Handle the Trainer kit carefully.

Algorithm/circuit/Diagram/Description:





Description : Internal wiring of Hex keypad

'E'	'D'	'E'	'F'	R3
'8'	'9'	'A'	'B'	R2
'4'	'5'	'6'	'7'	R1
'0'	'1'	'2'	'3'	R0
		C0	C1	C2 C3

Test cases:

Run the program for different Key strokes.

Sample output:

Values to be loaded 8100 ←01

8101 ←02

8102 ← 04

8103 ← 08

By pressing 6, we get

06	

8085 Display

Practical Related Questions:

1. The keyboard interfacing schemes can also be divided into two modes. What are they?
2. What is scanning in keyboard and what is scan time?
3. What is JP1, JP2 pins?
4. How a keyboard matrix is formed in keyboard interface using 8279?

Exercise Questions:

1. Assume the interface is connected over J2 of the 8085 trainer kit in keyboard or serial mode.
2. Assume that 8255 interface is connected over P4 of the trainer kit in keyboard or serial mode.

EXPERIMENT - 5: TRAFFIC LIGHT INTERFACE

Write an assembly code to for Traffic light controller to set from South to North direction.

Practical significance:

Traffic light controller interface module is designed to simulate the function of four way traffic light controller. Combinations of red, amber and green LED's are provided to indicate Halt, Wait and Go signals for vehicles. Combination of red and green LED's are provided for pedestrian crossing. 36 LED's are arranged in the form of an intersection. A typical junction is represented on the PCB with comprehensive legend printing. At the left corner of each road, a group of five LED's (red, amber and 3 green) are arranged in the form of a T-section to control the traffic of that road. Each road is named North (N), South(S), East (E) and West (W). LED's L1, L10, L19 & L28 (Red) are for the stop signal for the vehicles on the road N, S, W, & E respectively. L2, L11, L20 & L29 (Amber) indicates wait state for vehicles on the road N, S, W, & E respectively. L3, L4 & L5 (Green) are for left, strait and right turn for the vehicles on road S. similarly L12-L13-L14, L23-L22-L21 & L32-L31-L30 simulates same function for the roads E, N, W respectively. A total of 16 LED's (2 Red & 2 Green at each road) are provided for pedestrian crossing. L7-L9.L16-L18, L25-L27 & L34-L36 (Green) when on allows pedestrians to cross and L6-L8, L15-L17, L24-L26 & L33-L35 (Red) when on alarms the pedestrians to wait.

Relevant Program Outcomes :

PO1, PO3, PO9 – Moderate
PO2,PO5 -Weak

Competency and practical skills :

1. To apply the concepts in the design of microprocessor/microcontroller based systems in real time applications
2. Develop code, debug, test and execute various assembly language programs and various peripherals using 8255A PPI & 8085 instruction set.

Prerequisites :

Basic knowledge of Digital Logic Design and Computer Organization.

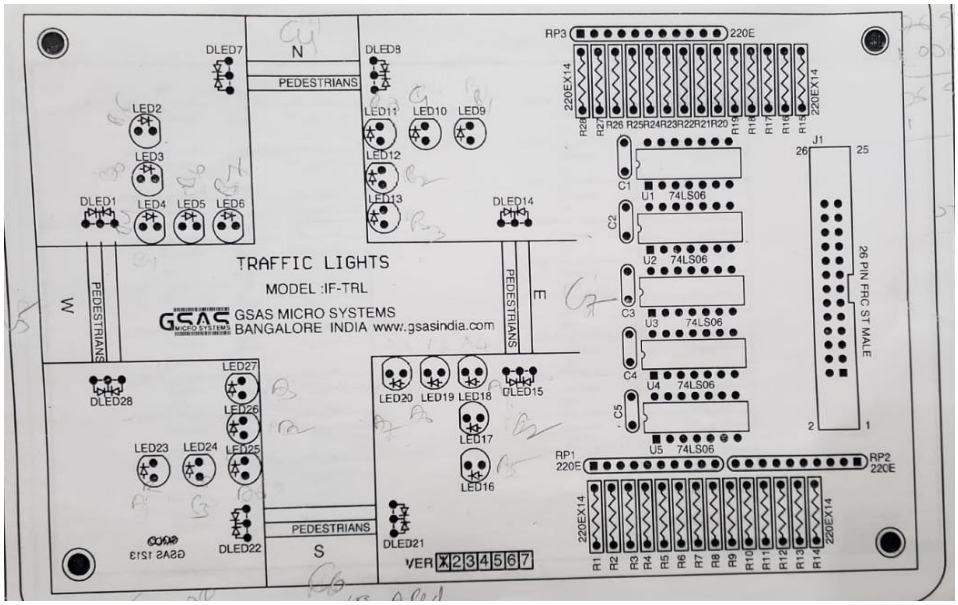
Resources required :

1. 8085 Microprocessor Kit
2. +5V Power supply
3. Personal computer (PC)
4. Traffic lights Interface
4. Flat ribbon cable (FRC).

Precautions :

1. Properly connect the Traffic lights interface to Personal computer.
2. Switch on the PC after checking connections.
3. Handle the Traffic lights interface carefully.

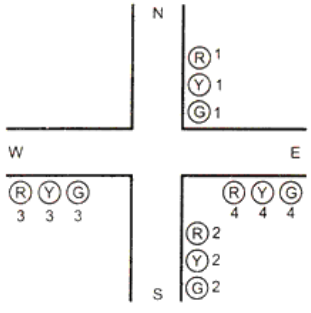
Algorithm/circuit/Diagram/Description:



Test cases:

1. Test the Traffic light controller to set from South to East direction.
2. Test the Traffic light controller to set from North to West direction.

Sample output:



Practical Related Questions:

1. How many LED's were given for traffic control on controller interface?
2. What is asynchronous data transfer scheme?

Exercise Questions:

1. Write an ALP to Traffic light controller to set from North to east direction.
2. To write an assembly program to set traffic lights in all directions with delay.

CYCLE III: INTERFACING WITH PC

EXPERIMENT - 1: STEPPERMOTOR

Write a program in PC to demonstrate the forward/reverse stepped motions of a stepper motor, using 8085 microprocessor.

Practical significance :

The stepper motors have immense applications in printing, Industrial Robotics, Precision tool motions in drilling, cutting and shaping machines, lathe etc. This project is proposed to demonstrate interfacing of stepper motor to 8085 microprocessor and to run it in continuous or stepped motion in forward or reverse direction under program control, using 8085 microprocessor.

Relevant Program Outcomes:

PO1, PO3, PO9 – Moderate
PO2, PO5 -Weak

Competency and practical skills :

To apply the concepts in the design of microprocessor/microcontroller based systems in real time applications

Develop code, debug, test and execute various assembly language programs and various peripherals using 8255A PPI & 8085 instruction set.

Prerequisites :

Basic knowledge of Digital Logic Design and Computer Organization.

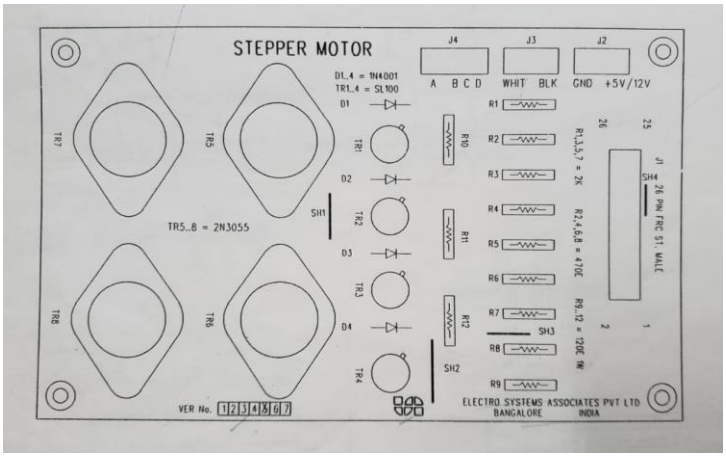
Resources required :

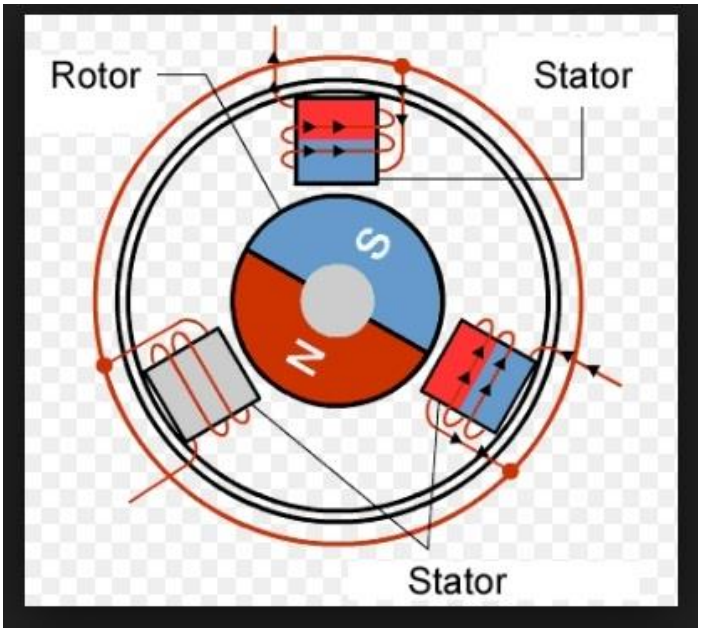
1. 8085 Microprocessor Kit
2. +5V Power supply
3. Personal computer (PC)
4. Stepper Motor
5. Stepper motor Interface Board

Precautions :

1. Properly connect the Stepper motor interface to Personal computer.
2. Switch on the PC after checking connections.
3. Do not change the motor direction at high speed
4. Handle the Stepper motor carefully.

Algorithm/circuit/Diagram/Description:





Test cases:

1. Test stepper motor to demonstrate the rotation in Clock wise rotation.
2. Test stepper motor to demonstrate the rotation in Anti Clock wise rotation.

Sample output:

Thus the interfacing of stepper motor with 8085 kit was done successfully.

Practical Related Questions:

1. Why interfacing is needed for I/O devices?
2. What is USART?

Exercise Questions:

1. Compose an Assembly Language Program to interface stepper motor to 8086 and observe the following:
   a. rotations in clockwise direction
   b. rotations in anticlockwise direction
   c. Continuous rotation in clockwise direction at much faster speed
2. Write an ALP for stepper motor to demonstrate the rotation in Anti Clock wise rotation

EXPERIMENT - 2: LOGIC CONTROLLER.

Write an assembly code to for Logic controller.

Practical significance:

We use a logic controller and it is used in industry for the process of control done by the software. Multiple inputs are typically accepted which performs a total complete sequence of operations carried out both arithmetic and logically.

Relevant Program Outcomes:

PO1, PO3, PO9 – Moderate
PO2, PO5 -Weak

Competency and practical skills :

1. To apply the concepts in the design of microprocessor/microcontroller based systems in real time applications
2. Develop code, debug, test and execute various assembly language programs and various peripherals using 8255A PPI & 8085 instruction set.

Basic knowledge of Digital Logic Design and Computer Organization.

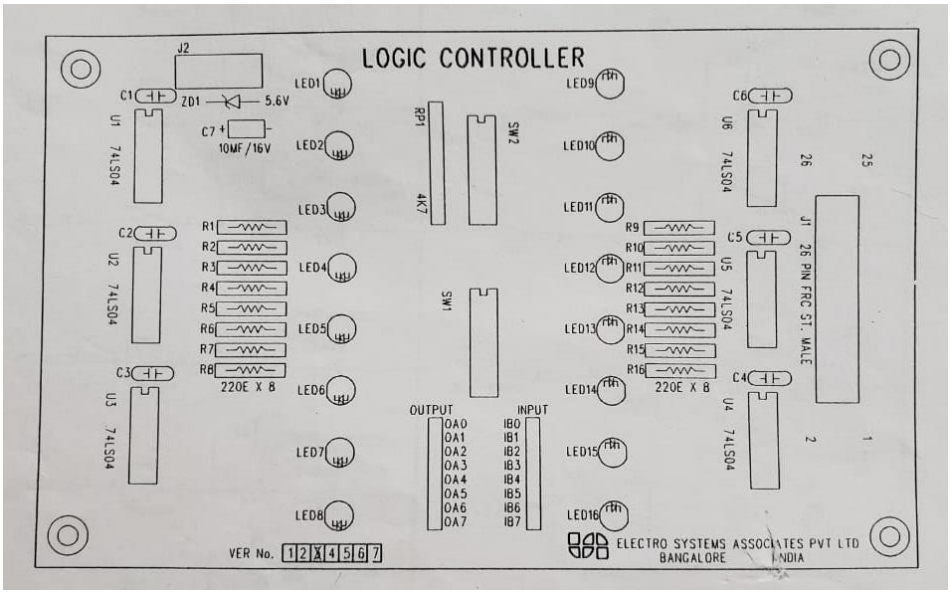
Resources required :

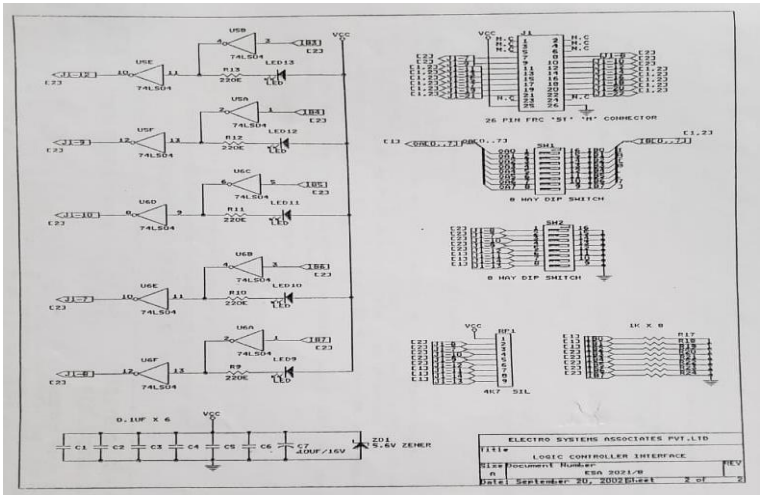
1. 8085 Microprocessor Kit
2. +5V Power supply
3. Personal computer (PC)
4. Logic controller Interface
4. Flat ribbon cable (FRC).

Precautions :

1. Properly connect the Traffic lights interface to Personal computer.
2. Switch on the PC after checking connections.
3. Handle the Logic controller interface carefully.

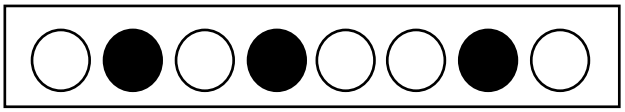
Algorithm/circuit/Diagram/Description:



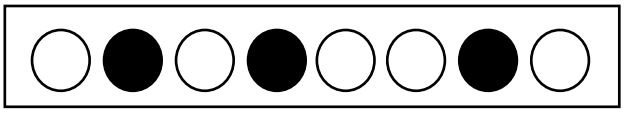


Test cases:

Input: Port B



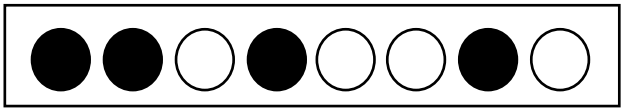
Output: Port A



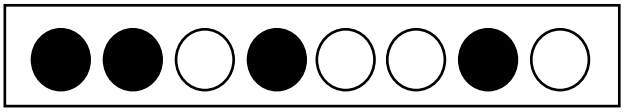
.

Sample Result:

Input: Port B



Input: Port A



Practical Related Questions:

1. Why interfacing is needed for I/O devices?
2. What is purpose of logic controller device?

Exercise Questions:

1. Write an ALP an Assembly Language Program to interface Logic for 1's Complement
2. Write an ALP an Assembly Language Program to interface Logic for 2's Complement

CYCLE IV : 8086 PROGRAMS

EXPERIMENT - 1: ADDITION OF TWO 16-BIT NUMBERS

Practical significance:

To write an assembly language program for performing addition of two 16-bit signed and unsigned numbers.

Relevant Program Outcomes:

PO1, PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

1. To apply the concepts in the design of microprocessor/microcontroller based systems in realtime applications
2. Develop code, debug, test and execute various assembly language programs using 8086 instruction set.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required:

1. TASM Software
2. PC with DOS and Debug program

Precautions

1. Make sure proper handling of equipments/ kits.
2. Make sure that all the machine codes/ mnemonics are as per the program.

Algorithm/circuit/Diagram/Description:

1. Initialize the data memory.
2. Load the first number into AX register.
3. Load the second number into BX register.
4. Add two lower digits.
5. Adjust result to valid BCD number.
6. Store the result in BL.
7. Add the two upper digits with carry.
8. Adjust result to valid BCD number.
9. Store the result in BH.
10. Display the result.
11. Stop.

Test cases:
INPUT: OPR1 = 9763H
OPR2 = A973H
OUTPUT: RES = 40D6H
Sample Input & Output

INPUT: OPR1 = 4269H
OPR2 = 1000H
OUTPUT: RES = 5269H
Practical Related Questions:

1. What is a Register?
2. What is the maximum size of any register in 8086?
3. What are the different categories of registers supported by 8086?
4. What is an addressing mode? What are different types of Addressing modes supported in 8086?
5. Define Opcode and Operand?
6. What is Physical Address of Instruction?
7. What is the purpose of the following instructions: a. MOV b. ADD c. ADC d. SBB e. SUB
   f. INT 3

Exercise Questions:

1. Formulate an assembly language program to move the content in memory location 1100h into register BX and also move to register CX, and also store the content in CX in memory location 1300h.
2. Write an assembly language program to add two 8-bit signed numbers (Use immediate and direct addressing modes).

EXPERIMENT - 2: SUBTRACTION OF TWO 16-BIT NUMBERS

Practical significance:
To write an assembly language program for performing Subtraction of two 16-bit signed and unsigned numbers.

Relevant Program Outcomes:

PO1, PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

3. To apply the concepts in the design of microprocessor/microcontroller based systems in realtime applications
4. Develop code, debug, test and execute various assembly language programs using 8086 instruction set.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required:

1. TASM Software
2. PC with DOS and Debug program

Precautions

1. Make sure proper handling of equipments/ kits.
2. Make sure that all the machine codes/ mnemonics are as per the program.

Algorithm/circuit/Diagram/Description:

1. Initialize the data memory.
2. Load the first number into AX register.
3. Load the second number into BX register.
4. Sub AX from BX.
5. Store result in AX
6. Display the result.
7. Stop

Test cases:

INPUT: OPR1 = 9763H
OPR2 = A973H
OUTPUT: RES = ODFOH

Sample Input & Output

INPUT: OPR1 = 4269H
OPR2 = 1000H
OUTPUT: RES = 3269H

Practical Related Questions:

1. Write the instruction formats for the above instructions?
2. Calculate the physical address for the given data. DS=1000h, BP=1234h.
3. What is the purpose of HLT instruction?
4. What happens if the result is greater than 16bit?
5. Give the steps to calculate physical address?
6. If carry is set to 1 before subtraction what is the instruction to be used?
7. What is the difference between MOV AX, [1100] and MOV [1200], AX?

Exercise Questions:

1. Formulate an assembly language program to move the content in memory location 1100h into register BX and also move to register CX, and also store the content in CX in memory location 1300h.
2. Write an assembly language program to subtract two 8-bit signed numbers (Use immediate and direct addressing modes).

EXPERIMENT - 3: MULTIPLICATION OF TWO 16-BIT NUMBERS

Practical significance:

To write an assembly language program to perform multiplication of two 16-bit unsigned numbers.

Relevant Program Outcomes:

PO1, PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

1. To apply the concepts in the design of microprocessor/microcontroller based systems in realtime applications
2. Develop code, debug, test and execute various assembly language programs using 8086 instruction set.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required:

1. TASM Software
2. PC with DOS and Debug program

Precautions

1. Make sure proper handling of equipments/ kits.
2. Make sure that all the machine codes/ mnemonics are as per the program.

Algorithm/circuit/Diagram/Description:

1. Initialize the data memory.
2. Load the first number into AX register.
3. Load the second number into BX register.
4. Multiply AX with BX.
5. Store lower byte in accumulator.
6. Store Upper byte in DX register
7. Display the result.
8. Stop

Test cases:
INPUT: OPR1 = 4000H OPR2 = 2000H
OUTPUT: RESLW = 0000H (AX) RESHW = 0800H (DX)

Sample Input & Output

INPUT: OPR1 = 2000H OPR2 = 4000H
OUTPUT: RESLW = 0000H (AX) RESHW = 0800H (DX)

Practical Related Questions:

1. Which registers should be used for performing MUL
2. Name the registers in which output will be stored after MUL
3. What is the purpose of the following instructions: a.MOV b.MUL c .IMUL
4. Write the instruction formats for the above instructions?
5. What is a Flag register?
6. From which address the 8086 starts execution after reset?

Exercise Questions:

1. Write an assembly language program to multiply two 8-bit signed numbers (Use immediate and direct addressing modes).

EXPERIMENT - 4: DIVISION OF 16-BIT NUMBERS
Practical significance:

To write an assembly language program for performing of two Division 16-bit signed and unsigned numbers.

Relevant Program Outcomes:

PO1, PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

1. To apply the concepts in the design of microprocessor/microcontroller based systems in realtime applications
2. Develop code, debug, test and execute various assembly language programs using 8086 instruction set.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required:

1. TASM Software
2. PC with DOS and Debug program

Precautions

1. Make sure proper handling of equipments/ kits.
2. Make sure that all the machine codes/ mnemonics are as per the program.

Algorithm/circuit/Diagram/Description:

1. Initialize the data memory.
2. Load the first number into AX register.
3. Load the second number into BX register.
4. Divide AX by BX. store Quotient in AL register.
5. Store reminder in AH register
6. Display the result.
7. Stop.

Test cases:

INPUT: OPR1 = 0008H (DIVIDEND) OPR2 = 04 (DIVISOR)
OUTPUT: RESQ = 02H (AL) RESR = 00H (AH)

Sample Input & Output

INPUT: OPR1 = 2C58H (DIVIDEND) OPR2 = 56H (DIVISOR)
OUTPUT: RESQ = 84H (AL) RESR = 00H (AH)

Practical Related Questions:

1. Derive the opcodes for the instructions used in your program?
2. List out the type of addressing modes used in your program.
3. If result exceeds 32 bit in multiplication, where is it stored?
4. What is the name given to the register combination DX: AX?
5. What is the instruction used for signed division and signed multiplication?
6. In the above program instead of DIV BX, is it possible to use DIV num2?
7. Where the remainder produced in 16 bit division is stored?

Exercise Questions:

1. Write an assembly language program to Division of 32 bit number by 16bit.

EXPERIMENT - 4: COMPLIMENT OPERATION (1's complement and 2's complement)

Practical significance:

To write an assembly language program to find one's compliment and two's compliment of given value.

Relevant Program Outcomes:

PO1, PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

1. To apply the concepts in the design of microprocessor/microcontroller based systems in realtime applications
2. Develop code, debug, test and execute various assembly language programs using 8086 instruction set.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required:

1. TASM Software
2. PC with DOS and Debug program

Precautions

1. Make sure proper handling of equipments/ kits.
2. Make sure that all the machine codes/ mnemonics are as per the program.

Algorithm/circuit/Diagram/Description:

1. The one's compliment of a given operand can be obtained by inverting 1's to 0's and vice versa.
2. The NOT instruction complements (inverts) the contents of an operand register or a memory location bit by bit.
3. The two's compliment is obtained by adding 1 to one's compliment result. This can be done by using ADD instruction or just by using INC instruction.

Test cases:

INPUT: OPR1 = 04H
OUTPUT: RESQ = OBH (1's complement)
RESQ = OCH (2's complement)

Sample Input & Output

INPUT: OPR1 = FFH
OUTPUT: RESQ = 00H (1's complement)
RESQ = 01H (2's complement)

Practical Related Questions:

1. Derive the opcodes for the instructions used in the program?
2. How many bytes of memory are needed for the following instructions?
   a) NOT AX b) NOT AL c) INC AX d) INC BL

Exercise Questions:

1. Formulate an Assembly Language Program to find one's compliment and two's compliment of a given 8-bit operand.

2. Write an Assembly Language Program to find one's compliment and two's compliment of a given 16-bit operand.

EXPERIMENT - 5: CODE CONVERSION

Practical significance:

To perform code conversion i.e. conversion of two ASCII values to packed BCD..

Relevant Program Outcomes:

PO1, PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

3. To apply the concepts in the design of microprocessor/microcontroller based systems in realtime applications
4. Develop code, debug, test and execute various assembly language programs using 8086 instruction set.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required:

1. TASM Software
2. PC with DOS and Debug program

Precautions

1. Make sure proper handling of equipments/ kits.
2. Make sure that all the machine codes/ mnemonics are as per the program.

Algorithm/circuit/Diagram/Description:

For converting 2 ASCII values to packed BCD number, the following steps should be done:

1. Convert two ASCII values into unpacked BCD format(use masking of bits concept)
2. Convert unpacked BCD numbers to packed BCD numbers(use rotate, OR operaton)

Unpacked to packed BCD: To pack a two-byte unpacked BCD number into a single byte creating a packed BCD number, shift the upper byte left four times, then OR the results with the lower byte. For example, 0000 0111 0000 1001(unpacked BCD) = 0111 1001(packed BCD) 0000 0111 << 4 = 0111 0000 (SHIFT LEFT 4) 0111 0000 + 0000 1001 = 0111 1001 (OR) Unpacked BCD to Binary: To convert from a two-byte unpacked BCD to

a binary number, multiply the most significant byte of the BCD by decimal ten, then add the product to the least significant byte.

For example, 00001001 00000010(unpacked BCD) = 01011100(base 2)
0000 1001 = 9
× 0000 1010 = 10(base 10)
0101 1010 = 90(base 10)

0101 1010 = 90(base 10)
+ 0000 0010 = 2
0101 1100 = 92(base 10)

Test cases:

INPUT: Two ascii values AL='6' BL='8'
OUTPUT: AL=68H(packed BCD format)

Sample Input & Output
INPUT: Two ascii values AL='9' BL='2'
OUTPUT: AL=92H(packed BCD format)

Practical Related Questions:

1. Derive the opcodes for the instructions used in the program?
2. Give the flags which are affected after performing the code conversion?
3. How many bytes of memory are needed for the following instructions:
   a. MOV CL,COUNT b. ROR AL,CL c. AND AL,0FH d. HLT

Exercise Questions:

Using Assembly Language Programming, Write a series of processor instructions to perform the following conversions:
a. Unpacked BCD number to packed BCD number and vice versa
b. Unpacked BCD number to Binary number

EXPERIMENT - 6: FIND BIGGER AND SMALLER NUMBER

Practical significance:

To find the largest and smallest number in an array of data.

Relevant Program Outcomes:

PO1, PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

1. To apply the concepts in the design of microprocessor/microcontroller based systems in realtime applications
2. Develop code, debug, test and execute various assembly language programs using 8086 instruction set.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required:

1. TASM Software
2. PC with DOS and Debug program

Precautions

a. Make sure proper handling of equipments/ kits.
b. Make sure that all the machine codes/ mnemonics are as per the program.

Algorithm/circuit/Diagram/Description:

1. To find the largest number in any given array, the contents of the array must be compared with an arbitrary biggest number.
2. The first number of the array is taken in a register AL.
3. The second number of the array is compared with the first one.
4. If the first one is greater than the second one, it is left unchanged.
5. However if the second one is greater than the first, the second number replaces the first one in the AL register.
6. The procedure is repeated for every number in the array and thus it requires n iterations.
7. At the end of nth iteration the largest number will reside in the register AL.
8. For smallest number the above said logic is repeated but, If the first number is smaller than the second one it is left unchanged.
9. Otherwise the second number replaces the first number in the AL register.

Test cases: (Largest)

INPUT: 09 FF 01 05 04
OUTPUT: FF

Sample Input & Output
INPUT: 09 06 01 05 04
OUTPUT: 09

Test cases: (Smallest)

INPUT: 09 FF 01 05 04
OUTPUT: 01

Sample Input & Output
INPUT: 09 06 00 05 04
OUTPUT: 00

Practical Related Questions:

1. What is the purpose of MOV DS, AX?
2. Derive the opcodes for the instructions used in your program?
3. What will be the status of flags after executing the program?
4. What are the addressing modes are used in our program?
5. What is the difference between JUMP and LOOP instructions?
6. What instructions are needed to add AL, 3L and DL together, and place the result in CL? Do not destroy BL or DL.
7. Show the instruction needed to count the number of 1's found in AL. For example if AL contains 10110001, the number of 1's is 4.
8. What is purpose served by CX register?

Exercise Questions:

1. Formulate an assembly language program to find largest of two given numbers
2. Write an assembly language program to find smallest of two given numbers
3. Compose an assembly language program to find largest and smallest number in a given array of data.
4. Write an algorithm and draw the flowchart for the above mentioned questions.

EXPERIMENT - 6: SEARCHING AND SORTING

Practical significance:
To write an assembly language program to arrange a given series of numbers in ascending and descending order.

Relevant Program Outcomes:

PO1, PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

1. To apply the concepts in the design of microprocessor/microcontroller based systems in realtime applications
2. Develop code, debug, test and execute various assembly language programs using 8086 instruction set.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required:

1. TASM Software
2. PC with DOS and Debug program

Precautions

1. Make sure proper handling of equipments/ kits.
2. Make sure that all the machine codes/ mnemonics are as per the program.

Algorithm/circuit/Diagram/Description:

1. To arrange the given numbers in ascending and descending order, the bubble sorting method is used.
2. Initially the first number of the series is compared with the second one.
3. If the first number is greater than second, exchange their positions in the series otherwise leave the position unchanged.
4. Then compare the second number in the recent form of the series with third and repeat the exchange part that you are carried out for the first and second number, and for all the remaining number of the series.
5. Repeat this procedure for complete series (n-1) times. After n-1 iterations you will get the largest number at the end of the series. Again start from the first number of the series.
6. Repeat the same procedure right from the first element to the last element.
7. After n-2 iteration you will get the second highest number at the last but one place in the series. Repeat this till the complete series is arranged in ascending order.

Test cases: (Sorting)

INPUT: 09 FF 01 05 04
OUTPUT: 01 04 05 09 FF

Sample Input & Output
INPUT: OE OD 0F OB O8
OUTPUT: 08 0B 0D 0E 0F

Practical Related Questions:

1. What is the purpose of XCHG instruction?
2. What is the use of PUSH and POP instruction?
3. Derive the opcodes for the instructions used in the programs?
4. Write an assembly language program in 8086 to sort the given array of 16-bit numbers in descending order.
5. What do square brackets means when they appear in an operand?

Exercise Questions:

1. Compose an Assembly Language Program to Sort the given byte length numbers in ascending order and also in descending order using bubble sort or insertion sort

2. Formulate an Assembly Language Program to Search for a given substring whether it is present in main string or not
3. Write an Assembly Language Program for finding given key value on given word length integers by using binary search

EXPERIMENT -7: STRING MANIPULATIONS

Practical significance:
To performing string manipulation operations on the string stored in the memory

Relevant Program Outcomes:

PO1, PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

1. To apply the concepts in the design of microprocessor/microcontroller based systems in realtime applications
2. Develop code, debug, test and execute various assembly language programs using 8086 instruction set.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required:

1. TASM Software
2. PC with DOS and Debug program

Precautions

a. Make sure proper handling of equipments/ kits.
b. Make sure that all the machine codes/ mnemonics are as per the program.

Algorithm/circuit/Diagram/Description:

1. To do any operation on strings, first string should be defined.
2. The defined string will be stored in memory with some offset address.
3. The offset address should be loaded into SI register. If you are comparing two strings or moving a string from one location to another, source string offset address should be in SI and destination offset address is stored in DI. This is done with the help of LEA instruction.
4. The length of the string should be stored in CX which serves as counter in doing any string manipulations.
5. For comparison of strings, use COMPS instruction. For moving a string from one location to another, use MOVS instruction.

Test cases:
(Moving block of data)
INPUT:
2000H: 05
12 34 56 78 9A
OUTPUT: 12 34 56 78 9A

Sample Input & Output
INPUT: OE OD 0F OB O8
OUTPUT: OE OD 0F OB O8
Practical Related Questions:

1. Derive the opcodes for the instructions used in the program?
2. Give the flags which are affected after execution of your program?
3. What does the direction flag value indicates if D=1 and also if D=0?
4. How many bytes of memory are needed for the following instructions?
    a) LEA SI, STR1 b) JNE L2 c) INC SI d) INC DI

Exercise Questions:

Using Assembly Language Programming, Write a series of processor instructions to do the following string manipulations:
a) Program to compare two given strings
b) Program for inserting a character into a string
c) Move block of data from one memory location to another memory location

EXPERIMENT -8: GCD AND FACTORIAL

Practical significance:
To find GCD and Factorial of given operand.

Relevant Program Outcomes:

PO1, PO3- Highly
PO8, PO9 - Moderate
PO2, PO4, PO5, PO10- Weak

Competency and practical skills:

1. To apply the concepts in the design of microprocessor/microcontroller based systems in realtime applications
2. Develop code, debug, test and execute various assembly language programs using 8086 instruction set.

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Resources required:

1. TASM Software
2. PC with DOS and Debug program

Precautions

1. Make sure proper handling of equipments/ kits.
2. Make sure that all the machine codes/ mnemonics are as per the program.

Algorithm/circuit/Diagram/Description:

1. GCD: Program should load two registers with two Numbers and then apply the logic for GCD of two numbers.
2. GCD of two numbers is performed by dividing the greater number by the smaller number till the remainder is zero.
3. If it is zero, the divisor is the GCD if not the remainder and the divisors of the previous division are the new set of two numbers.
4. The process is repeated by dividing greater of the two numbers by the smaller number till the remainder is zero and GCD is found.

Test cases: (GCD)

First number=AL=0Ah Second number=BL=05h GCD is 05h

Sample Input & Output

First number=AL=05h Second number=BL=02h GCD is 01h

Practical Related Questions:

1. Derive the opcodes for the instructions used in the program?
2. Give the flags which are affected after calculating GCD & Factorial?
3. How many bytes of memory are needed for the following instructions:
    a) CMP b) INC SI c) DEC DI

Exercise Questions:

1. Using Assembly Language Programming, Write a series of processor instructions to do the following string manipulations:
    a) Find GCD and LCM for given two byte length numbers
    b) Find the factorial for the given number