Master of Computer Applications (MCA)

Microprocessor Lab (DMCASE209P24)

Self-Learning Material (SEM II)

Jaipur National University
Centre for Distance and Online Education

Established by Government of Rajasthan
Approved by UGC under Sec 2(f) of UGC ACT 1956
&
NAAC A+ Accredited

TABLE OF CONTENTS

EXPERT COMMITTEE

Prof. Sunil Gupta
(Department of Computer and Systems Sciences, JNU Jaipur)

Dr. Deepak Shekhawat
(Department of Computer and Systems Sciences, JNU Jaipur)

Dr. Shalini Rajawat
(Department of Computer and Systems Sciences, JNU Jaipur)

COURSE COORDINATOR

Prof. Sudhir Sharma
(Department of Computer and Systems Sciences, JNU Jaipur)

UNIT PREPARATION

Unit Writer(s)	Assisting & Proofreading	Unit Editor
Mr. Puneet kalia (Department of Computer and Systems Sciences, JNU Jaipur)	Mr. Ram Lal Yadav (Department of Computer and Systems Sciences, JNU Jaipur)	Mr. Shish Dubey (Department of Computer and Systems Sciences, JNU Jaipur)

Secretarial Assistance

Mr. Mukesh Sharma

COURSE INTRODUCTION

The Microprocessor Lab course students is a hands-on, experiential learning component designed to complement theoretical knowledge with practical skills. This lab course aims to provide students with an in-depth understanding of microprocessor systems through direct engagement with hardware and software tools, fostering a comprehensive grasp of microprocessor functionalities and applications.

In the Microprocessor Lab, students will have the opportunity to work with various microprocessor kits and development boards, allowing them to experiment with different architectures and configurations. The lab sessions are structured to guide students through the fundamental aspects of microprocessor operations, including instruction execution, data manipulation, and control processes. By working directly with microprocessors, students can observe and analyze the real-time execution of instructions, enhancing their understanding of abstract concepts covered in lectures.

A significant focus of the lab is on assembly language programming, where students will write, debug, and execute programs on microprocessor kits. This hands-on practice is crucial for mastering low-level programming techniques and understanding how high-level instructions are translated into machine-level operations. Students will learn to develop efficient code, optimize performance, and troubleshoot common issues, skills that are essential for careers in embedded systems and hardware design.

The lab sessions also emphasize the importance of understanding and implementing interrupt-driven programming. Students will learn to handle interrupts effectively, enabling microprocessors to respond to external events promptly. This knowledge is crucial for developing real-time systems where timely responses are critical. Practical exercises in this area will help students understand the intricacies of interrupt handling and the challenges of maintaining system stability and performance.

Furthermore, the Microprocessor Lab encourages collaborative learning and teamwork. Students will often work in pairs or small groups, fostering an environment of peer learning and knowledge sharing. This collaborative approach helps students develop communication and teamwork skills, which are essential in professional settings.

The lab course also integrates project-based learning, where students will design and implement mini-projects that showcase their understanding of microprocessor systems. These projects encourage creativity and innovation, allowing students to apply their knowledge to solve practical problems. Through these projects, students will gain confidence in their ability to design, develop, and test microprocessor-based systems.

Through practical exercises, collaborative learning, and project-based assignments, students will develop a robust understanding of microprocessors and their applications, equipping them with the skills needed for successful careers in technology and engineering.

Course Outcomes:
At the completion of the course, a student will be able to:

1. Analyze the ever-changing digital environment in which e-commerce exists and its impact on operational needs, capabilities, opportunities and challenges.
2. Uniderstand digital marketing methods organizations can use in combination with other marketing methods and integrate into their international sales and marketing plan.
3. Apply the elements to consider in the design of an efficient and effective e-commerce operation, including the ability to integrate with other systems within an organization, localize for each target market and accommodate growth.
4. Assess organizational readiness to set up and support an e-commerce operation serving national or international markets.
5. Perform an e-commerce operation using components and practices that provide a storefront, a shopping cart and payment options, minimize security and privacy risks, are user friendly, and provide timely customer support and delivery.

Acknowledgements:

The content we have utilized is solely educational in nature. The copyright proprietors of the materials reproduced in this book have been tracked down as much as possible. The editors apologize for any violation that may have happened, and they will be happy to rectify any such material in later versions of this book.

Question 1: Basic Arithmetic Calculator

Program Statement:

Develop a program on the 8085 microprocessor to perform arithmetic operations (addition, subtraction, multiplication, and division) on two entered numbers.

Solution Hints:

- Use the accumulator for arithmetic operations.
- Implement error handling for division by zero.
- Display results on a seven-segment display or through the serial port.

Question 2: Data Transfer between Registers

Program Statement:

Write a program to transfer data between various registers and memory locations using different addressing modes on the 8086 microprocessor.

Solution Hints:

- Experiment with immediate, direct, register indirect, and indexed addressing modes.
- Use MOV instructions to perform data transfer.
- Validate the data transfer by displaying the register contents.

Question 3: Implement Stack Operations

Program Statement:

Create a program to demonstrate PUSH and POP operations in 8085. Simulate a stack overflow and underflow condition.

Solution Hints:

- Use the stack pointer to manage stack operations.

- Push multiple register pairs onto the stack and then pop them off.
- Implement checks to simulate and handle stack overflow and underflow.

Question 4: String Reversal

Program Statement:

Write a program on the 8086 microprocessor to reverse a string stored in memory.

Solution Hints:

- Store the string in the data segment.
- Use a loop with index registers to reverse the string in place.
- Display the original and reversed string through the console.

Question 5: BCD to Binary Conversion

Program Statement:

Develop a program in 8085 to convert a BCD number to its binary equivalent and display the result.

Solution Hints:

- Understand the difference between BCD and binary representations.
- Use double dabble or a similar algorithm for conversion.
- Use debugging or a display to check the binary output.

Question 6: Traffic Light Controller Simulation

Program Statement:

Simulate a traffic light control system using the 8085 microprocessor connected to LEDs via an interface.

Solution Hints:

- Use the programmable peripheral interface (PPI) for controlling LEDs.
- Implement a timing sequence to change the LEDs representing traffic lights.
- Consider pedestrian signals in the control logic.

Question 7: Frequency Counter

Program Statement:

Design a frequency counter using the 8086 microprocessor to measure the frequency of an input signal.

Solution Hints:

- Use the 8253 timer/counter for measuring frequency.
- Configure the counter in a suitable mode to capture the frequency.
- Display the frequency on an alphanumeric display connected via an interface.

Question 8: Multiplication of Large Numbers

Program Statement:

Write a program to multiply two 16-bit numbers in 8085 and store the result.

Solution Hints:

- Handle the carry generated from multiplying large numbers.
- Use register pairs to store the 16-bit numbers and results.
- Display the 32-bit result using multiple data registers.

Question 9: Keyboard Interface

Program Statement:

Interface a keyboard with the 8086 microprocessor and write a program to display the pressed key on a monitor.

Solution Hints:

- Use an 8255 PPI for interfacing the keyboard.
- Implement an interrupt-driven or polling-based approach to detect key presses.
- Display the character corresponding to the key press.

Question 10: Assembly Language Subroutines

Program Statement:

Create a library of subroutines for the 8085 microprocessor that includes operations like sorting an array, finding the maximum number in an array, and calculating the average.

Solution Hints:

- Write separate subroutines for each operation.
- Use the CALL and RET instructions to manage subroutine calls.
- Test each subroutine with sample data stored in memory.

Question 11: LED Blinking Pattern

Program Statement:

Program the 8086 microprocessor to control an array of LEDs to display various blinking patterns.

Solution Hints:

- Use an interface board with LEDs connected.
- Implement different patterns using loop and delay subroutines.
- Control the timing and sequence of LED activation to create visible patterns.

Question 12: Analog to Digital Conversion

Program Statement:

Interface an ADC with the 8085 microprocessor and write a program to read an analog input and display the digital value.

Solution Hints:

- Use the 8255 PPI to interface the ADC.
- Poll the ADC status or use an interrupt to read the value.
- Convert the ADC output to a readable format and display it.

Question 13: Serial Communication

Program Statement:

Implement a serial communication interface between two 8086 microprocessors to exchange data.

Solution Hints:

- Configure the serial communication interface using the 8251 USART.
- Establish a protocol for sending and receiving data.
- Write programs for both sender and receiver microprocessors.

Question 14: Factorial Calculation

Program Statement:

Write a program on the 8085 microprocessor to calculate the factorial of a number entered through the keyboard.

Solution Hints:

- Use a loop to calculate the factorial.
- Store intermediate results in registers.

- Handle larger numbers carefully to avoid overflow.

Question 15: Real-Time Clock Display

Program Statement:

Interface a real-time clock module with the 8086 microprocessor and display the time in HH:MM:SS format on a digital display.

Solution Hints:

- Use the RTC chip like DS12C887.
- Implement a routine to read the time from the RTC.
- Convert the time into a displayable format and update the display regularly.

Question 16: Palindrome Checker

Program Statement:

Develop a program in 8085 to check if a given string stored in memory is a palindrome.

Solution Hints:

- Use loop and comparison instructions to check characters from the beginning and end of the string.
- Display results as palindrome or not a palindrome.

Question 17: Interrupt Handling

Program Statement:

Write a program to handle multiple interrupts in 8086, simulating an environment where interrupts are triggered by keyboard and timer.

Solution Hints:

- Set up the interrupt vector table correctly.
- Write separate interrupt service routines for keyboard and timer.
- Ensure proper prioritization and handling of interrupts.

Question 18: Square Wave Generator

Program Statement:

Use the 8085 microprocessor to generate a square wave signal of specified frequency on a GPIO pin.

Solution Hints:

- Use a timer or software delays to control the frequency.
- Toggle the GPIO pin at the calculated frequency rate.
- Connect the pin to an oscilloscope to verify the wave shape and frequency.

Question 19: Memory Testing Program

Program Statement:

Create a program to test the memory of a system using the 8086 microprocessor, checking for faulty addresses and reporting errors.

Solution Hints:

- Write and read back values at each memory address.
- Use patterns like 0x55AA to check for stuck bits.
- Report the addresses where read values do not match written values.

Question 20: Simple Game Development

Program Statement:

Develop a simple game like tic-tac-toe on the 8085 microprocessor, using a keypad for input and a display for output.

Solution Hints:

- Design the game logic to check for win conditions.
- Interface the keypad to accept inputs and display to show the game state.
- Manage the game flow and player turns within the microprocessor.