PARVATHANENI BRAHMAYYA
SIDDHARTHA COLLEGE OF ARTS & SCIENCE
Autonomous
Siddhartha Nagar, Vijayawada–520010
Re-accredited at 'A+' by the NAAC

Course Code				**23IOMIP231**			
Title of the Course				**Microprocessor lab**			
Offered to:				B.Sc. AI(Hons), BCA(A &B)			
L	**0**	**T**	**0**	**P**	**2**	**C**	**1**
Year of Introduction:	**2024-25**			**Semester:**			**3**
Course Category:	Minor			**Course Relates to:**	Global		
Year of Revision:				**Percentage:**			
Type of the Course:				Skill development			
Crosscutting Issues of the Course :							
Pre-requisites, if any				Familiarity with basic electronic components			

Course Description:

The Microprocessor Lab provides hands-on experience with microprocessor-based systems. Students will learn to program and interface microprocessors using assembly language and high-level languages. Key activities include writing, debugging, and executing programs to control peripheral devices and implement various functionalities. The lab covers topics such as memory organization, input/output interfacing, and interrupt handling. Utilizing development boards and simulation software, students will design and test microprocessor applications. This lab aims to develop skills in embedded system design, fostering an understanding of microprocessor architecture and its applications in automation, robotics, and real-time systems.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	Understand the fundamental architecture and operation of microprocessors.).
2	Write, debug, and execute assembly language and high-level language programs for microprocessor-based systems.
3	Utilize development boards and simulation software to test and validate microprocessor designs.
4	Develop problem-solving skills related to embedded systems and microprocessor applications.
5	Design and implement microprocessor-based applications, focusing on automation, robotics, and real-time systems.

Course Outcomes

At the end of the course, the student will be able to…

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Explain the fundamental architecture and operation of microprocessors.	K4	1	1
CO2	Develop and implement microprocessor-based applications, focusing on areas such as automation, robotics, and real-time systems.	K4	1	1
CO3	Utilize development boards, simulation software, and other tools to design, test, and validate microprocessor systems.	K5	1	1
CO4	Manage memory organization, input/output interfacing, and interrupt handling in microprocessor-based systems.	K5	1	1
CO5	Interpret and produce detailed schematics, documentation, and reports for microprocessor-based projects.	K3	1	1

For BTL: K1: Remember; K2: Understand; K3: Apply; K4: Analyze; K5: Evaluate; K6: Create

CO-PO MATRIX									
CO NO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2
CO1	2							2	
CO2	2							2	
CO3	2							2	
CO4	2							2	
CO5	3							3	

Use the codes 3,2,1 for High, Moderate and Low correlation Between CO-PO-PSO respectively

Course Structure

This lab list covers the key areas of a Microprocessor lab course, providing hands-on practice with using EMU software:

1: Data Transfer Instructions

- **Objective:** Understand and practice data transfer instructions.
- **Activities:**
  - **Lecture:** Explain data transfer instructions (e.g., MOV, PUSH, POP, XCHG).
  - **Lab Exercise:**
    - **Simple Programs:** Write and test programs to transfer data between registers, memory, and I/O ports.
    - **Practice:** Use simulation software to observe how data transfer instructions work in practice.

2: Logical Instructions

- **Objective:** Explore logical instructions and their usage.
- **Activities:**
  - **Lecture:** Detailed discussion of logical instructions (e.g., AND, OR, XOR, NOT, TEST).
  - **Lab Exercise:**

        - **Basic Programs:** Write programs to perform bitwise operations and test logical instructions.
        - **Analysis:** Use a simulator to visualize how logical operations affect data.

3: Arithmetic Instructions

- **Objective:** Learn and practice arithmetic instructions.
- **Activities:**
    - **Lecture:** Overview of arithmetic instructions (e.g., ADD, SUB, MUL, DIV).
    - **Lab Exercise:**
        - **Simple Programs:** Write and run programs to perform basic arithmetic operations and handle results.
        - **Testing:** Use the simulator to verify the correctness of arithmetic operations.

4: Branch Instructions

- **Objective:** Understand branch instructions and their applications.
- **Activities:**
    - **Lecture:** Explanation of branch instructions (e.g., JMP, CALL, RET, JZ, JNZ).
    - **Lab Exercise:**
        - **Control Flow Programs:** Write programs to demonstrate conditional and unconditional branching.
        - **Simulation:** Use a simulator to trace program execution and observe branching behavior.

5: Flag Manipulation Instructions

- **Objective:** Study instructions for flag manipulation.
- **Activities:**
    - **Lecture:** Discuss flag manipulation instructions (e.g., CLC, STC, CLI, STI, CLD, STD).
    - **Lab Exercise:**
        - **Flag Operations:** Write programs to manipulate and test the status of flags.
        - **Practical Application:** Observe how flag manipulation affects program execution and control.

6: Shift and Rotate Instructions

- **Objective:** Explore shift and rotate instructions.
- **Activities:**

- **Lecture:** Explanation of shift and rotate instructions (e.g., SHL, SHR, ROL, ROR).
- **Lab Exercise:**
  - **Shift and Rotate Programs:** Write and test programs that use shift and rotate instructions for data manipulation.
  - **Simulation:** Use tools to visualize the impact of these operations on data.

7: Loop Instructions and ALP Programs

- **Objective:** Learn loop instructions and write assembly language programs for specific tasks.
- **Activities:**
  - **Lecture:** Overview of loop instructions (e.g., LOOP, JCXZ).
  - **Lab Exercise:**
    - **Loop Programs:** Write programs using loop instructions to perform repetitive tasks.
    - **ALP Programs:** Implement and test assembly language programs for the following tasks:
      - **ADD:** Add two numbers.
      - **SUB:** Subtract two numbers.
      - **MUL:** Multiply two numbers.
      - **DIV:** Divide two numbers.
      - **LARGEST:** Find the largest number in a list.
      - **SMALLEST:** Find the smallest number in a list.

8: ARM Cortex-M Development Tools and Environment

- **Objective:** Familiarize with the development tools and environment for ARM Cortex-M microcontrollers.
- **Activities:**
  - **Lecture:** Introduction to ARM development tools and environments (e.g., Keil MDK, IAR Embedded Workbench, GCC).
  - **Lab Exercise:**
    - **Tool Setup:** Install and configure a development environment (e.g., Keil MDK or GCC).
    - **Hello World Program:** Write and compile a simple "Hello World" program to familiarize with the toolchain.

9: Hands-On Programming with ARM Cortex-M

- **Objective:** Gain practical experience programming ARM Cortex-M microcontrollers.

- **Activities:**
    - **Lecture:** Basics of programming in ARM assembly and C for Cortex-M.
    - **Lab Exercise:**
        - **C Programming:** Write and test simple programs in C for the Cortex-M microcontroller, focusing on basic I/O operations.
        - **Assembly Programming:** Write and test basic ARM assembly language programs.

Week 10: Introduction to ARM Cortex-M Core Architecture

- **Objective:** Understand the ARM Cortex-M core architecture, including its structure and functionalities.
- **Activities:**
    - **Lecture:** Overview of ARM Cortex-M core architecture, including the CPU core, pipelining, and key features.
    - **Lab Exercise:**
        - **Architecture Diagram:** Analyze or draw the block diagram of the ARM Cortex-M core.
        - **Simulation:** Use simulation tools to explore the Cortex-M architecture and its components.

Lab Manual:

1. [Manual - Author(s), Year of Publication, Title, Edition, Publisher]Supplied by the Department

References:

1. [Reference 1 - Author(s), Year of Publication, Title, Edition, Publisher]
2. [Reference 2 - Author(s), Year of Publication, Title, Edition, Publisher]

LAB EXAMINATION PATTERN

23IOMIP231: Microprocessor Lab

Semester: III MAX Marks:35M **Time:3Hrs**

(A) SEE Evaluation Procedure

1. For Aim, Apparatus, circuit diagram, Flow chart	**5M**
2.For Observation table, formulas, Program	**5M**
3. Experiment Procedure and Execution	**10M**
4.Output of Experiment	10M
5. Practical Record	**2M**
6. Viva voce	**3M**

(B) CONTINUOUS ASSESMENT(Internal) 15 MARKS

15 marks for the continuous assessment (Day to day work in the laboratory shall be evaluated for 15 marks by the concerned laboratory teacher based on the regularity/ record/viva). Laboratory teachers are mandated to ensure that every student completes 80%-90% of the lab assessments.

TOTAL: (A)+(B) = 50 MARKS