Gyanmanjari Innovative University

Syllabus
Gyanmanjari Institute of Technology
Semester-3 (B. Tech)

Subject: Microprocessor & Controller-based System Design – BET2EE13306

Type of course: Professional Core

Prerequisite: Digital Electronics, Basic Programming Concepts

Rationale:

This course provides students with foundational and advanced knowledge of microprocessors and microcontrollers, which serve as the backbone of embedded systems, automation systems, industrial control, consumer electronics, and intelligent devices. Students will learn architecture, assembly language programming, interfacing techniques, and real-time control applications using the 8085 microprocessor and the 8051 microcontroller. The course bridges theoretical concepts and practical implementation to prepare students for applications in embedded systems, IoT, automation, robotics, and industrial electronics.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks		Total Marks
CI	T	P	C	SEE	CCE	
2	0	4	4	100	50	150

Legends: CI-Classroom Instructions; T-Tutorial; P -Practical; C –Credit; ESE -End Semester Examination; MSE-Mid Semester Examination; V–Viva; CA-Continuous Assessment; ALA-Active Learning Activities.

Course Content:

Sr. No.	Course Content	Hrs.	% Weightage
1	**Fundamentals of Microprocessors and 8085 Architecture** ***Topics:*** Introduction to microprocessors and microcomputers, Von Neumann architecture, architecture of 8085, pin configuration, internal registers, flag register, buses, control signals, timing diagrams, instruction cycle, machine cycle, T-states, memory interfacing basics. ***Practical:*** 1. To study the hardware components, operating features, and functional operation of the 8085 microprocessor trainer kit. 2. To identify and analyze the pin configuration of the 8085	14	15%

microprocessor and understand the function of each pin.
3. To examine the internal register organization and analyze the operation of status flags in the 8085 microprocessor.
4. To implement and verify demultiplexing of the 8085 microprocessor's multiplexed address/data bus using a latch circuit.
5. To analyze the opcode fetch machine cycle and interpret its timing diagram with respect to control and status signals.
6. To interface memory devices with the 8085 microprocessor and verify memory read and write operations.
7. To investigate various machine cycles and T-states of the 8085 microprocessor using a simulation tool and analyze their timing characteristics.

Evaluation Method:

Sr. No.	Evaluation Method	SEE (Marks)	CCE (Marks)
1.	**8085 Microprocessor Architecture Modeling using Microsoft Visio/AI Tools:** Create a professional block diagram of the 8085 microprocessor architecture using Microsoft Visio, Draw.io, Lucidchart, or an AI-based diagram generation tool. Label all functional units, indicate data/address/control signal flow, and prepare a brief explanation of the operation of each block.	10	-
2.	**8085 Timing Diagram Generation and Analysis using Digital Diagramming Tools:** Generate opcode fetch cycle, machine cycle, and T-state timing diagrams for specified 8085 instructions using Microsoft Visio, Draw.io, WaveDrom, Timing Diagram Editor, or AI-assisted diagram generation tools. Analyze the sequence of control signals and explain the relationship between machine cycles and T-states.	10	-
3.	**Comparative Architecture Analysis using AI and Diagramming Tools:** Prepare a professional comparative study of Microprocessor and Microcomputer architectures using Microsoft Visio, Draw.io, Lucidchart, Canva, or AI-assisted presentation tools. Develop architecture diagrams, highlight key differences, and	-	5

	applications, and present the analysis through a technical presentation or infographic.		
4.	**Machine Cycle, Flag Operation, and Bus Operation Visualization using Simulation Tools:** Demonstrate and analyze machine cycles, flag operations, and bus operations of the 8085 microprocessor using simulator software (e.g., GNUSim8085, Sim8085, EdSim51) or a microprocessor trainer kit. Capture outputs, generate operational flow diagrams/charts using professional diagramming tools, and present the observations with technical interpretation.	-	5
	Total	20	10

Examination Style:

8085 Microprocessor Architecture Modeling using Microsoft Visio/AI Tools:

Students will develop a professional architecture model of the 8085 microprocessor using suitable diagramming tools. They should identify and represent major functional blocks, data paths, address buses, control signals, and internal organization of the microprocessor. The assessment will evaluate the accuracy of the architecture diagram, clarity of signal flow representation, technical understanding of individual functional units, and the quality of technical documentation.

8085 Timing Diagram Generation and Analysis using Digital Diagramming Tools

Students will generate and analyze opcode fetch, machine cycle, and T-state timing diagrams for selected 8085 instructions using digital diagramming tools. They should demonstrate an understanding of timing relationships, control signal sequencing, machine cycle execution, and instruction processing. The assessment will evaluate the correctness of timing diagrams, interpretation of control signals, and analysis of machine cycle and T-state operations.

Comparative Architecture Analysis using AI and Diagramming Tools

Students will prepare a comparative analysis of microprocessor and microcomputer architectures using AI-assisted presentation and visualization tools. They should identify architectural differences, functional characteristics, advantages, limitations, and application areas through diagrams, tables, and technical illustrations. The assessment will evaluate analytical ability, technical accuracy, quality of comparison, and effectiveness of visual presentation.

	Machine Cycle, Flag Operation, and Bus Operation Visualization using Simulation Tools Students will demonstrate and analyze machine cycles, flag operations, and bus operations of the 8085 microprocessor using simulation software or trainer kits. They should interpret the behavior of status flags, data transfer mechanisms, bus operations, and instruction execution cycles through simulation outputs and technical observations. The assessment will evaluate the ability to analyze operational behavior, interpret simulation results, and present findings through appropriate technical documentation.		
2	**8085 Programming and Debugging** ***Topics:*** Instruction classification, data transfer instructions, arithmetic instructions, logical instructions, branching instructions, addressing modes, instruction format, stack operations, subroutines, assembler directives, looping, counters, delay generation, and debugging concepts. ***Practical:*** 8. To develop, execute, and verify an assembly language program for the addition of two 8-bit numbers and observe the resulting data and status flags. 9. To develop, execute, and verify an assembly language program for the subtraction of two numbers and analyze the effect on status flags. 10. To develop and execute an assembly language program to perform multiplication using the repeated addition technique and verify the obtained result. 11. To develop and execute an assembly language program to perform division using the repeated subtraction technique and determine the quotient and remainder. 12. To develop and execute assembly language programs utilizing loops and counters for repetitive operations and program flow control. 13. To develop and execute an assembly language program incorporating subroutines and demonstrate modular programming techniques. 14. To develop and verify assembly language programs for code conversion operations such as Binary-to-BCD, BCD-to-Binary, or other specified code conversions. 15. To develop and execute an assembly language program for generating a specified time delay using software-based timing techniques. 16. To identify, analyze, and rectify errors in assembly language programs using appropriate debugging techniques and tools. **Evaluation Method:** **Sr. No.** | **Evaluation Methods** | **SEE** | **CCE**	22	25%

1	**Assembly Program Development:** Write and execute programs for arithmetic operations (addition, subtraction, multiplication, and division) using the 8085 instruction set.	10	-	
2	**Code Conversion Programming:** Develop programs for BCD to Binary, Binary to BCD, ASCII to Binary, or Binary to ASCII conversions	5	-	
3	**Program Logic Analysis:** Design programs using loops, counters, branching, stack operations, and subroutines for given problem statements.	5	-	
4	**Debugging Activity:** Identify errors in given assembly programs, debug them, and verify correct output using a simulator/trainer kit.	-	5	
5	**Program Demonstration / Viva:** Demonstrate execution of developed programs and explain program flow, registers used, and output verification.	-	5	
	Total	20	10	

Examination Style:

Assembly Program Development

Students will write, execute, and verify assembly language programs for arithmetic operations such as addition, subtraction, multiplication, and division using the 8085 instruction set. They should demonstrate an understanding of instruction execution, register utilization, and result storage while ensuring correct program logic and output.

Code Conversion Programming

Students will develop and execute programs for code conversion operations such as BCD to Binary, Binary to BCD, ASCII to Binary, and Binary to ASCII. They should apply appropriate instructions and logic to perform data conversion and verify the correctness of the output.

Program Logic Analysis

Students will design and analyze programs using loops, counters, branching instructions, stack operations, and subroutines for given problem statements. They should demonstrate the ability to develop efficient program logic and explain the flow of execution.

Debugging Activity

Students will identify syntax and logical errors in given assembly language programs, perform debugging using a simulator or trainer kit.

<table>
<tr><td></td><td>and verify the corrected output. The activity should emphasize systematic error detection and troubleshooting techniques.

Program Demonstration and Viva
Students will demonstrate the execution of developed programs and explain the instruction sequence, registers used, addressing modes applied, and output verification process. They should be able to justify program logic and answer questions related to program execution.</td><td></td><td></td></tr>
<tr><td>3</td><td>Fundamentals of 8051 Microcontroller

<u>Topics:</u>
History of microcontrollers, Harvard vs Von-Neumann architecture, RISC vs CISC architecture, classification of MCS-51 family (8051, 8052, 8031, 8751, AT89C51), comparison of microprocessors and microcontrollers, architecture of 8051, registers, RAM organization, register banks, stack, ROM memory, clock circuits, reset circuits, I/O ports.

<u>Practical:</u>
17. To study the hardware components, functional features, and operational characteristics of the 8051 microcontroller trainer kit.
18. To identify and analyze the pin configuration of the 8051 microcontroller and understand the function of each pin.
19. To develop and execute programs for accessing and utilizing different register banks of the 8051 microcontroller.
20. To develop and execute programs for configuring and controlling the input/output ports of the 8051 microcontroller.
21. To examine the memory organization of the 8051 microcontroller and analyze the utilization of RAM, ROM, Special Function Registers (SFRs), and register banks.
22. To study the operation of the clock and reset circuits of the 8051 microcontroller and analyze their role in system initialization and timing control.
23. To develop and verify basic input-output interfacing programs using the 8051 microcontroller and external peripheral devices.

Evaluation Method:
<table>
<tr><th>Sr. No.</th><th>Evaluation Methods</th><th>SEE</th><th>CCE</th></tr>
<tr><td>1</td><td>8051 Microcontroller Architecture Modeling using Professional Diagramming Tools: Develop a professional architecture diagram of the 8051 microcontroller using Microsoft Visio, Draw.io, Lucidchart, or AI-assisted diagramming tools and explain the functionality of its internal blocks.</td><td>10</td><td>-</td></tr>
</table>
</td><td>14</td><td>15%</td></tr>
</table>

	2	**Comparative Architecture Study using AI-Assisted Presentation Tools:** Prepare a professional comparative analysis of Harvard vs. Von Neumann architecture, RISC vs. CISC Architecture, and Microprocessor vs. Microcontroller using AI-assisted presentation tools, Canva, PowerPoint, or infographic design platforms. Present architectural differences, advantages, limitations, and application areas through diagrams, tables, and visual representations.	5	-			
	3	**8051 Memory and I/O Organization Analysis using Simulation Tools:** Investigate the register banks, RAM organization, stack operations, and I/O port structure of the 8051 microcontroller using simulation software such as Keil µVision, Proteus, EdSim51, or other suitable simulators. Document memory mapping, stack behavior, and port operations with screenshots, block diagrams, and technical observations.	5	-			
	4	**Application-Based Quiz:** Participate in a technology-enabled quiz (Kahoot, Quizizz, Google Forms, etc.) covering MCS-51 family classification, 8051 architecture, memory organization, register banks, clock and reset circuits, I/O ports, Harvard vs. Von-Neumann architecture, RISC vs. CISC architecture, and comparison of microprocessors and microcontrollers. The quiz should emphasize conceptual understanding, architectural analysis, and application-oriented problem solving.	-	10			
		Total	20	10			

Examination Style:

8051 Microcontroller Architecture Modeling using Professional Diagramming Tools

Students will develop a professional architecture model of the 8051 microcontroller using suitable diagramming tools. They should identify and represent major functional blocks such as CPU, ALU, memory

	organization, register banks, timers, serial communication unit, interrupt structure, and I/O ports, along with their interconnections. The assessment will evaluate technical accuracy, completeness of the architecture representation, understanding of functional units, and quality of technical documentation. **Comparative Architecture Study using AI-Assisted Presentation Tools** Students will prepare a comparative analysis of Harvard and Von-Neumann architectures, RISC and CISC architectures, and Microprocessors and Microcontrollers using AI-assisted presentation and visualization tools. They should analyze architectural characteristics, advantages, limitations, and application domains through diagrams, comparison tables, and technical illustrations. The assessment will evaluate analytical skills, conceptual understanding, technical accuracy, and effectiveness of presentation. **8051 Memory and I/O Organization Analysis using Simulation Tools** Students will investigate and analyze the memory organization and I/O structure of the 8051 microcontroller using simulation software. They should examine RAM organization, register banks, stack operations, Special Function Registers (SFRs), and I/O port functionality, and document their observations through memory maps, screenshots, and technical reports. The assessment will evaluate understanding of memory architecture, interpretation of simulation results, and quality of analysis and documentation. **Application-Based Quiz** Students will participate in a technology-enabled quiz to demonstrate their understanding of the 8051 microcontroller architecture, memory organization, register banks, clock and reset circuits, I/O ports, and MCS-51 family classification. They should analyze architectural features, compare computing architectures, interpret memory and port organization, and apply fundamental microcontroller concepts to solve application-oriented and scenario-based questions.		
4	**Programming and Real-Time Applications using 8051** ***Topics:*** Addressing modes, data types, assembler directives, instruction set, and programming techniques. Timers/Counters: Registers, modes, programming; Interrupts: Interrupt vs polling, interrupt types, vector addresses, interrupt registers, programming external and timer interrupts. Interfacing: LED interfacing, seven-segment display, Relay interfacing, LCD interfacing, Stepper motor interfacing, 4×4 keyboard interfacing ***Practical:*** 24. To develop, execute, and verify an 8051 assembly language program for the addition of two 8-bit numbers and analyze the resulting	22	25%

output.

25. To develop, execute, and verify an 8051 assembly language program for the subtraction of two numbers and observe the effect on status flags.
26. To develop, execute, and verify an 8051 assembly language program for the multiplication of two numbers using the appropriate instruction set.
27. To develop and execute an 8051 assembly language program to perform multiplication using the repeated addition method and verify the obtained result.
28. To develop, execute, and verify an 8051 assembly language program for the division of two numbers using the DIV instruction and determine the quotient and remainder.
29. To develop and execute an 8051 assembly language program to perform division using the repeated subtraction method and determine the quotient and remainder.
30. To develop and execute 8051 assembly language programs utilizing loops and counters for repetitive operations and program flow control.
31. To develop and execute an 8051 assembly language program using subroutines and demonstrate modular programming techniques.
32. To interface LEDs with the 8051 microcontroller and demonstrate their operation through assembly language programming.
33. To interface a seven-segment display with the 8051 microcontroller and display numeric data through assembly language programming.
34. To interface a relay with the 8051 microcontroller and control its operation using assembly language programming.

Evaluation Method:

Sr. No.	Evaluation Methods	SEE	CCE
1	**Assembly Program Development:** Write and execute 8051 programs for arithmetic operations, loops, counters, and subroutines.	10	-
2	**Troubleshooting Exercise:** Identify errors in interfacing/programming circuits and debug hardware/software issues.	10	-
3	**Mini Project on Real-Time Applications using 8051:** Design, develop, and demonstrate a microcontroller-based application utilizing 8051 programming, timers/counters, interrupts, and interfacing techniques. Students may implement applications such as digital counters, security systems, automatic control systems, display units, monitoring systems, or other real-time embedded solutions. The project should include	-	10

			program development, hardware/simulation implementation, testing, and result analysis.			
			Total	20	10	

Examination Style:

Assembly Program Development

Students will develop, execute, and verify 8051 assembly language programs for arithmetic operations, loops, counters, and subroutines. They should demonstrate an understanding of addressing modes, instruction execution, register utilization, program flow control, and modular programming techniques while ensuring correct output and efficient program implementation.

Troubleshooting Exercise

Students will identify, analyze, and rectify errors in 8051 programming and interfacing circuits using appropriate debugging techniques, simulation tools, or hardware platforms. They should demonstrate the ability to trace program execution, diagnose logical and hardware-related faults, interpret system behavior, and verify the correctness of the implemented solution.

Mini Project on Real-Time Applications using 8051

Students will design, develop, and demonstrate a real-time embedded application using the 8051 microcontroller by applying assembly language programming, timers/counters, interrupts, and interfacing concepts. They should analyze the problem requirements, develop appropriate hardware/software solutions, implement the system using simulation tools or hardware platforms, and verify its functionality through testing and performance evaluation.

5	**Project Development and System Implementation** **Topics** Project planning and development methodology, problem identification, system design approach, block diagram development, component selection, programming techniques, hardware interfacing integration, sensor and actuator integration, real-time control implementation, debugging techniques, testing methodologies, technical documentation, prototype validation, teamwork, and presentation skills. Students will apply concepts of: • Programming • Interfacing • Sensors/Actuators • Real-time control • Debugging • Documentation	18	20%

- Teamwork

Project planning and development methodology, problem identification, system design approach, block diagram development, component selection, programming techniques, hardware interfacing integration, sensor and actuator integration, real-time control implementation, debugging techniques, testing methodologies, technical documentation, prototype validation, teamwork, and presentation skills.
Students will apply concepts of:

Project Definition

Students shall design, develop, implement, test, and demonstrate a microcontroller-based embedded system using either the 8051 microcontroller or Arduino Uno platform. The project shall address a real-world engineering problem and integrate concepts such as programming, hardware interfacing, sensors/actuators, real-time control, debugging, testing, documentation, and technical presentation. Students may select a project from the suggested project (practical) options or propose an equivalent project with faculty approval.

Project Options:

35. To interface a seven-segment display with the microcontroller and display the given input number.
36. To develop and implement a system for converting binary input data into its decimal equivalent and displaying the result.
37. To design and implement a relay control system that operates based on a specified range of binary input values.
38. To develop and demonstrate a digital counter system using a seven-segment display for real-time count visualization.
39. To design and implement a temperature monitoring system using a temperature sensor and display unit for real-time temperature indication.
40. To develop a password-based security lock system for controlled access using a microcontroller and input interface.
41. To design and implement an automatic streetlight control system based on ambient light conditions.
42. To develop a water level monitoring system for detecting and displaying the status of water levels in a storage tank.
43. To design and implement a visitor counter system for monitoring and displaying the number of entries and exits.
44. To develop a fire alarm system capable of detecting fire-related conditions and generating an appropriate alert.
45. To design and implement a motion detection system that generates an alert upon detecting movement within a specified area.
46. To develop an automatic fan control system that regulates fan operation according to measured temperature conditions.
47. To design and implement a smart parking slot indication system for

monitoring and displaying parking space availability.

48. To develop a digital notice display system for displaying programmable messages on an electronic display unit.
49. To design, develop, and demonstrate a microcontroller-based solution for a student-defined real-world problem by applying appropriate hardware and software techniques.

(Implementation can be carried out using either the 8051 microcontroller or Arduino Uno platform.)

Evaluation Method

Sr. No.	Evaluation Component	SEE (Marks)	CCE (Marks)
1	**Project Report:** Documentation including objective, circuit diagram, algorithm/flowchart, implementation details, and results.	5	-
2	**Hardware/Prototype Demonstration:** Working model demonstration with proper output verification.	5	-
3	**Viva Examination:** Oral evaluation based on programming logic, hardware interfacing, troubleshooting, and contribution.	10	-
4	**Presentation:** Project presentation with PPT/poster/demo explanation.	-	10
	Total	20	10

Examination Style:

Project Report Evaluation

Students will prepare a project report documenting the complete development process, including problem identification, objectives, system requirements, block diagram, circuit design, algorithm/flowchart, implementation methodology, testing procedures, results, and conclusions. The report should demonstrate technical understanding, proper documentation practices, and systematic project development.

Hardware / Prototype Demonstration

Students will demonstrate the working prototype of their developed system and verify its functionality under different operating conditions. They should explain the hardware configuration, software implementation, interfacing techniques, and system operation while validating the expected

	output. **Viva Examination** Students will undergo an oral examination based on project planning, programming logic, hardware interfacing, sensor and actuator integration, debugging techniques, testing methodology, and overall system implementation. They should be able to justify design decisions, explain troubleshooting approaches, and demonstrate their individual contribution to the project. **Project Presentation** Students will present their project through a PPT, poster, or live demonstration highlighting the problem statement, objectives, methodology, system architecture, implementation process, results, and future scope. The presentation should reflect technical knowledge, teamwork, communication skills, and the ability to effectively explain the developed solution.		

Suggested Specification Table with Marks (Theory): 100

Distribution of Marks (Revised Bloom's Taxonomy)						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage %	10%	15%	30%	15%	10%	20%

Course Outcome:

After learning the course, the students should be able to:	
CO1	Explain the architecture, operation, timing concepts, and memory interfacing of the 8085 microprocessor for understanding microprocessor-based systems.
CO2	Develop and debug assembly language programs using the 8085 instruction set, addressing modes, loops, counters, subroutines, and code conversion techniques.
CO3	Analyze the architecture, memory organization, and internal resources of the 8051 microcontroller and compare it with modern microcontroller architectures.
CO4	Design and implement real-time embedded applications using 8051 programming, timers, interrupts, and peripheral interfacing techniques.
CO5	Develop a real-world embedded system prototype using the 8051 microcontroller or Arduino UNO by integrating programming, interfacing, debugging, and system design concepts.

Instructional Method:

The course delivery method will depend on the requirements of the content and the needs of students. The teacher, in addition to the conventional teaching method by the blackboard, may also use any of the tools, such as demonstration, role play, quizzes, brainstorming, MOOCs, etc.

From the content, 10% of topics are suggested for flipped-mode instruction.

Students will utilize supplementary resources, including online videos, NPTEL/SWAYAM videos, e-courses, and Virtual Laboratories.

The internal evaluation will be done on the basis of the CCE-Continuous and Comprehensive Evaluation.

SEE: Semester End Evaluation will be conducted at the end of the semester for the evaluation of the performance of students in the laboratory.

Reference Books

[1] R. S. Gaonkar, *Microprocessor Architecture, Programming, and Applications with the 8085*, 6th ed. New Delhi, India: Penram International Publishing Pvt. Ltd., 2013.

[2] M. A. Mazidi, J. G. Mazidi, and R. D. McKinlay, *The 8051 Microcontroller and Embedded Systems: Using Assembly and C*, 2nd ed. Noida, India: Pearson Education, 2011.

[3] D. V. Hall, *Microprocessors and Interfacing: Programming and Hardware*, 3rd ed. New Delhi, India: McGraw-Hill Education, 2016.

[4] K. Kant, *Microprocessors and Microcontrollers: Architecture, Programming and System Design using 8085, 8086, 8051 and 8096*, New Delhi, India: PHI Learning Pvt. Ltd., 2007.

[5] S. Mathur and J. Panda, *Microprocessors and Microcontrollers*, New Delhi, India: PHI Learning Pvt. Ltd., 2016.

[6] M. Banzi and M. Shiloh, *Getting Started with Arduino*, 4th ed. Sebastopol, CA, USA: O'Reilly Media, 2022.