Gyanmanjari Innovative University

Syllabus
Gyanmanjari Degree Engineering College
Semester-5 (B.Tech.)

Subject: Computer Organization and Architecture – BET1CE15320

Type of course: Professional Core

Prerequisite: Basic understanding of digital logic design, number systems, Boolean logic, and combinational and sequential circuit fundamentals.

Rationale:
This course is structured to deliver an in-depth understanding of computer organization and architecture by developing a comprehensive understanding of computer systems and their functional components, including system characteristics, operational principles, performance considerations, and internal as well as external communication mechanisms. It enables students to explore system organization concepts such as system buses, memory hierarchy, input/output organization, and processor interaction, helping them understand how hardware components function collectively within a computing environment. The course emphasizes conceptual clarity and applied learning through the study of hardware architectural concepts and introductory assembly language programming, while also introducing students to the Evaluation: of computer hardware and the working fundamentals of modern processors. By building domain-specific foundational knowledge of microprocessors, computer system architecture, system operation, and peripheral communication, the course prepares learners with essential competencies required for hardware-oriented domains in computer engineering and related disciplines.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks		Total Marks
CI	T	P	C	SEE	CCE	
2	0	2	3	100	50	150

Legends: CI-Class Room Instructions; T – Tutorial; P - Practical; C – Credit; SEE - Semester End Evaluation; CCE-Continuous and Comprehensive Evaluation.

Course Content:

Sr. No	Course Content	Hrs.	% Weightage
1	**Fundamentals of Computer Organization** ***Theory Topics:*** Introduction to computer systems and the basic structure of a CPU. Fundamental concepts of digital electronics including flip-flops, registers, and shift registers. Study of different types of CPU registers and their applications, along with an overview of system buses used for communication within the processor. ***Practical:*** 1. Number System Conversion: Convert numbers between Binary, Decimal, Octal, and Hexadecimal systems. 2. Binary Arithmetic: Perform binary addition, subtraction, multiplication, and division. 3. Logic Gates: Verify truth tables of basic logic gates such as AND, OR, NOT, NAND, NOR, XOR, and XNOR. 4. Boolean Algebra: Simplify Boolean expressions using Boolean laws or Karnaugh Map (K-Map). 5. Flip-Flops: Study the operation of SR, JK, D, and T flip-flops using truth tables or simulation. 6. Registers: Demonstrate register operations such as load, clear, shift left, and shift right. 7. CPU Registers: Study different CPU registers and their functions in instruction execution. 8. Memory Organization: Illustrate memory hierarchy and calculate memory addressing. 9. Instruction Cycle: Demonstrate the Fetch–Decode–Execute cycle of a processor. 10. Assembly Language Basics: Write simple assembly language programs for arithmetic and data transfer operations.	18	20%

Evaluation Method:

Sr. No.	Evaluation: Method	SEE	CCE
1	**Instruction Cycle Demonstration:** Students will demonstrate the Fetch–Decode–Execute cycle for a given machine-level or assembly instruction. They will explain the role of CPU registers, control unit operations, and data transfer through system buses during instruction execution.	10	-
2	**Memory Organization & Performance Analysis:** Students will analyse different memory components such as registers, cache memory, main memory, and secondary storage. They will explain memory hierarchy, access time, storage capacity, and the effect of memory organization on overall system performance.	10	-
3	**Active Learning Active (ALA): Architecture Analysis :** Students will analyse the architecture of a given computer system or processor model and identify major components such as CPU, memory, registers, buses, and I/O units. They will explain the role of each component with a suitable diagram and submit the report in PDF format on the GMIU Web Portal.	-	10
	Total	20	10

Examination Style:

1. **Instruction Cycle Demonstration (10 Marks)**:
   - Students will be provided with a sample machine-level or assembly instruction by the faculty. They are required to illustrate the Fetch–Decode–Execute cycle by describing how the CPU processes the instruction, including the role of registers, control unit activities,

	and data transfer through system buses. The objective of this task is to evaluate students' understanding of instruction execution and internal processor operations. 2. **Memory Organization & Performance Analysis (10 Marks)**: • Students will be execute a machine-level or assembly instruction by the faculty. They must demonstrate the Fetch–Decode–Execute cycle by explaining each stage of instruction processing, highlighting the functions of CPU registers, control unit operations, and data flow through system buses. This activity aims to assess students' conceptual understanding of processor functioning and instruction execution procedures. 3. **Active Learning Activity (ALA): Architecture Analysis (10 Marks)**: • In this task, students will analyse the architecture of a given computer system or processor model provided by the faculty. Students will identify major functional units such as CPU, memory, registers, buses, and input/output components, and explain their roles using suitable diagrams. This task is designed to evaluate students' conceptual understanding of computer organization and system components. The analysis report into PDF format will be submitted by students at GMIU Web Portal.		
2	**8085 Microprocessor Architecture** ***Theory Topics:*** Introduction to microprocessors and their role in computer systems. Study of the 8085 microprocessor including pin diagram and pin functions. Overview of the internal architecture of the 8085 processor, general purpose registers, and flag register. Understanding of instruction execution and basic operation of the 8085 microprocessor.	18	20%

	Practical: 11. Introduction to 8085 Simulator: Study the interface of the 8085 simulator and identify registers, memory window, and execution controls. 12. 8085 Register Identification: Observe accumulator, general purpose registers, program counter, and stack pointer using the simulator. 13. 8085 Pin Diagram Study: Study and document pin diagram and functions of the 8085 microprocessor. 14. Data Transfer Instructions: Write and execute programs using MOV, MVI, and LXI instructions. 15. Arithmetic Operations: Write an assembly program to perform addition of two 8-bit numbers. 16. Subtraction Operation: Write an assembly program to subtract two numbers and observe flag status. 17. Flag Register Observation: Execute arithmetic operations and analyze Zero, Carry, Sign, and Parity flags. 18. Instruction Execution Study: Execute instructions step-by-step and observe instruction execution sequence. 19. Memory Operations: Write programs to store and retrieve data from memory locations. 20. Simple Program Execution: Develop a small 8085 program combining data transfer and arithmetic instructions and verify output.		

Evaluation Method:

Sr. No.	Evaluation: Method	SEE	CCE
1	**Instruction Execution Observation:** Students will execute basic 8085 instructions using a simulator and observe the step-by-step execution process. They will record changes in registers, memory locations, and flag status to understand the instruction execution flow.	10	-
2	**8085 Pin Diagram & Function Analysis:** Students will analyse the 8085 pin diagram and explain the function of important pins such as address bus, data bus, control signals, interrupts, clock, reset, and power supply pins. This task will assess students' understanding of 8085 hardware structure and signal functions.	10	-
3	**Active Learning Activity (ALA): 8085 Architecture Study:** Students will study the architecture of the 8085 microprocessor and identify its major components such as registers, ALU, control unit, buses, flags, and timing/control section. They will explain the function of each component with a suitable diagram and submit the report in PDF format on the GMIU Web Portal.	-	10
	Total	20	10

Examination Style:

1. **Instruction Execution Observation (10 Marks):**
   - Students will perform execution of basic 8085 microprocessor instructions using a simulator provided by the faculty. They are required to observe the step-by-step execution process and document the

	changes occurring in CPU registers and flag status during program execution. This task is designed to evaluate students' understanding of instruction execution flow and fundamental processor operations. 2. **8085 Pin Diagram & Function Analysis (10 Marks)**: • Students will execute selected basic 8085 microprocessor instructions using a simulator provided by the faculty. They are required to monitor the instruction execution process step-by-step and record the corresponding changes in registers and flag conditions during program execution. This task aims to assess students' conceptual understanding of instruction execution and internal processor functioning. 3. **Active learning Activity (ALA) : 8085 Architecture Study (10 Marks)**: • In this task, students will be given two program, containing logical or syntactical errors. The questions will be of unsolved, practical, or competitive coding type and will be provided by the faculty. The task aims to assess the students' debugging skills, logical reasoning, and ability to correct code to produce the desired output. The solution will be prepaid by students into PDF format and submitted to GMIU Web Portal.		
3	**8085 Instruction Set and Assembly Language Programming** ***Theory Topics:*** Introduction to instruction format including opcode and operands, and basic concepts of machine language instructions. Study of different instruction formats such as 1-byte, 2-byte, and 3-byte instructions. Understanding various types of 8085 instructions including data transfer, arithmetic, logical, branching and looping, stack operations, input/output, and machine control instructions. Overview of different addressing modes of the 8085 microprocessor and classification of interrupts along with their priority levels.	18	20%

	Practical: 21. Instruction Format Study: Identify opcode and operand fields of different 8085 instructions using the simulator. 22. 1-Byte Instruction Program: Write and execute programs using 1-byte instructions. 23. 2-Byte Instruction Program: Execute instructions involving immediate data using 2-byte instruction format. 24. 3-Byte Instruction Program: Implement instructions involving memory addressing using 3-byte instructions. 25. Data Transfer Instructions: Write programs using MOV, MVI, LDA, STA, and LXI instructions. 26. Arithmetic Instructions: Develop programs to perform addition and subtraction operations and observe flag changes. 27. Logical Instructions: Execute logical operations such as AND, OR, XOR, and Compare instructions. 28. Branching & Looping Instructions: Write programs using JMP, JZ, JNZ, and loop-based instructions. 29. Stack Operations: Demonstrate PUSH and POP instructions and observe stack behavior. 30. Interrupt & Addressing Mode Study: Demonstrate different addressing modes and study types of interrupts with priority levels using example programs.		

Evaluation Method:

Sr. No.	Evaluation: Method	SEE	CCE
1	**Addressing Modes Implementation** Students will implement basic 8085 addressing modes such as immediate, register, direct, and indirect addressing using a simulator. They will prepare example programs and explain the execution results of each addressing mode.	10	-
2	**Branching, Looping & Stack Operations Simulation** Students will execute 8085 programs based on branching, looping, and stack operations using a simulator. They will analyse program flow, stack behavior, register changes, and output results during execution.	10	-
3	**Active Learning Activity (ALA): Instruction Execution Simulation:** Students will execute 8085 programs using a simulator and observe data transfer, arithmetic, and logical instructions. They will record register and flag changes with screenshots and submit the final PDF report on the GMIU Web Portal.	-	10
	Total	20	10

Examination Style:x

1. **Active Learning Activity (ALA): Instruction Execution Simulation (10 Marks):**
   - In this task, students will use a free 8085 microprocessor simulator to execute programs containing data transfer, arithmetic, and logical instructions. Students will observe instruction execution step-by-step and record register and flag changes during execution. Students must prepare a detailed report including program code, screenshots, observations, and results. The final report shall be converted into PDF format and submitted through the GMIU Web Portal.

	2. Addressing Modes Implementation (10 Marks): • Students will execute and implement various 8085 microprocessor addressing modes using a freely available simulator. They are required to develop example programs demonstrating immediate, register, direct, and indirect addressing modes. Students must submit a report including program code, execution results, and a brief explanation of each addressing mode. This task is designed to evaluate students' practical understanding and application of addressing techniques in microprocessor programming. **3. Branching, Looping & Stack Operations Simulation (10 Marks):** • Students will design and execute simple 8085 assembly language programs incorporating branching, looping, and stack instructions using an 8085 simulator. They will analyze program execution flow and monitor stack operations during runtime. Students are required to submit a structured report including the algorithm, assembly program, execution screenshots, and output analysis. This task aims to enhance students' logical reasoning skills and understanding of processor control mechanisms.		
4	**Memory Organization** ***Theory Topics:*** Introduction to memory classification and organization in computer systems. Study of memory hierarchy and its importance in improving system performance. Overview of different types of main memory including RAM and ROM along with their characteristics and applications. Understanding various auxiliary memory devices used for secondary storage. Concepts of cache memory and its role in speeding up data access. The unit also covers virtual memory techniques and their significance in efficient memory management and execution of large programs.	18	20%

	Practical: 31. Memory Hierarchy Study: Illustrate memory hierarchy using diagrams showing registers, cache, main memory, and secondary memory. 32. RAM and ROM Comparison: Study characteristics of RAM and ROM and prepare comparison based on speed, volatility, and applications. 33. Main Memory Organization: Demonstrate memory addressing and storage organization using CPU simulator tools. 34. Auxiliary Memory Study: Analyze different auxiliary storage devices such as HDD, SSD, and optical storage using system specifications. 35. Memory Access Time Analysis: Observe and compare access time differences between cache, main memory, and secondary storage. 36. Cache Memory Simulation: Simulate cache memory operation and observe hit and miss conditions using cache simulator. 37. Cache Mapping Techniques: Study direct mapping, associative mapping, and set-associative mapping using simulation examples. 38. Virtual Memory Concept Demonstration: Observe virtual memory usage and page allocation using operating system memory monitoring tools. 39. Paging Mechanism Study: Demonstrate paging concept and logical-to-physical address translation through simulation. 40. Memory Performance Analysis: Analyze how memory hierarchy improves system performance and prepare observation report.		

Examination Style:

Sr. No.	Evaluation: Method	SEE	CCE
1	**Cache Memory Mapping Simulation:** Students will use a cache simulation tool to implement and compare Direct Mapping, Fully Associative Mapping, and Set-Associative Mapping techniques. They will analyse cache hit, miss behavior, and performance differences using sample memory reference sequences.	10	-
2	**Virtual Memory & Paging Implementation:** Students will demonstrate virtual memory and paging concepts using operating system utilities or online simulators. They will perform logical-to-physical address translation and explain memory allocation, paging workflow, and observations during execution.	10	-
3	**Active Learning Activity (ALA): Memory Hierarchy Modeling using Simulation Tool:** Students will model the memory hierarchy of a computer system using a diagram or simulation tool. They will represent registers, cache memory, main memory, and auxiliary memory with access time and storage capacity comparison, and submit the final PDF report on the GMIU Web Portal.	-	10
	Total	20	10

	Examination Style: 1. **Active Learning Activity (ALA): Memory Hierarchy Modelling using Simulation Tool (10 Marks):** • In this task, students will model the memory hierarchy of a computer system using free diagram or simulation software. Students will represent registers, cache memory, main memory, and auxiliary memory along with access time and storage capacity comparison. The report including system model, screenshots, explanation, and conclusions must be prepared in PDF format and submitted through the GMIU Web Portal. This task evaluates students' understanding of memory organization and performance optimization. 2. **Cache Memory Mapping Simulation Task (10 Marks):** • Students will execute and utilize a free cache simulation tool to implement and compare different cache mapping techniques, including Direct Mapping, Fully Associative Mapping, and Set-Associative Mapping. They will execute sample memory reference sequences, evaluate cache hit and miss behavior, and analyze performance differences among the mapping methods. Students are required to submit a structured report containing simulator configuration details, execution screenshots, observations, and comparative performance analysis. This task aims to strengthen students' understanding of cache organization and memory performance optimization. 3. **Virtual Memory & Paging Implementation Task (10 Marks):** • Students will execute the concepts of virtual memory management using operating system utilities or suitable online simulators. They will perform examples of paging operations and logical-to-physical address translation, and analyze how memory allocation occurs during program execution. Students must prepare a structured technical report including workflow steps, execution screenshots, observations, and conceptual explanations. This task is designed to evaluate students' understanding of virtual memory organization and memory management mechanisms.		

5	**Input–Output Organization** ***Theory Topics:*** Introduction to input–output organization and its role in computer system operation. Study of input–output interfaces and mechanisms used for communication between CPU and peripheral devices. Understanding various modes of data transfer including programmed I/O, interrupt-driven I/O, and direct memory access (DMA). The unit also covers CPU–I/O processor communication techniques and data exchange methods for efficient input and output operations. ***Practical:*** 41. Input–Output Interface Study: Illustrate the basic structure of an input–output interface showing CPU, I/O devices, data bus, address bus, and control signals using diagram tools. 42. Programmed I/O Operation: Demonstrate programmed I/O data transfer using simple simulation or microcontroller programs and observe CPU polling behavior. 43. Interrupt Driven I/O Study: Analyze interrupt-based I/O communication by simulating interrupt request and interrupt service routine execution. 44. Direct Memory Access (DMA) Concept Study: Illustrate DMA data transfer mechanism and compare it with programmed I/O and interrupt-driven I/O using simulation diagrams. 45. Serial Communication Demonstration: Perform serial data transfer between system and I/O device using terminal or serial communication software. 46. Parallel Data Transfer Study: Demonstrate parallel mode data transfer using logic simulation tools and observe simultaneous multi-bit communication. 47. CPU–I/O Communication Analysis: Study communication between CPU and peripheral devices through control signals, status signals, and data exchange operations. 48. I/O Addressing Techniques: Analyze memory-mapped I/O and isolated I/O concepts using system architecture diagrams or simulator tools. 49. Data Transfer Modes Comparison: Compare programmed I/O, interrupt-driven I/O, and DMA transfer modes based on CPU utilization and efficiency.	18	20%

50. I/O System Performance Observation: Analyze how different I/O data transfer techniques affect overall system performance and prepare an observation report.

Examination Style:

Sr. No.	Evaluation: Method	SEE	CCE
1	**Interrupt Handling Case Study** Students will study interrupt handling in a selected computer system or operating system example. They will explain how interrupts are generated, prioritized, and processed by the CPU using suitable flow diagrams or illustrations.	10	-
2	**I/O System Design Proposal :** Students will design a conceptual I/O organization model for a system such as a smart classroom, automated library, or IoT setup. They will propose suitable input/output devices, interfaces, and data transfer techniques for efficient CPU–I/O communication.	10	-
3	**Active Learning Activity (ALA): I/O Performance Evaluation using System Monitoring Tools** Students will evaluate the performance of different I/O devices using system monitoring tools. They will observe CPU utilization, data transfer rate, I/O throughput, and device activity, prepare comparison tables with screenshots, and submit the final PDF report on the GMIU Web Portal.	-	10
	Total	20	10

	Examination Style: 1. **Interrupt Handling Case Study (10 Marks):** • Students will execute an analytical study on the role of interrupts in modern computer systems by selecting an appropriate operating system or embedded system example. Students will explain how interrupt requests are generated, prioritized, and executed by the CPU, supported with flow diagrams or conceptual illustrations. This task is designed to evaluate students' understanding of system-level interrupt execution and processor response mechanisms. 2. **I/O System Design Proposal (10 Marks):** • Students will execute the design of a conceptual I/O organization model for a hypothetical computer system (such as a smart classroom system, automated library, or IoT monitoring setup). Students are required to propose appropriate input/output devices, communication interfaces, and data transfer techniques, and explain how efficient CPU–I/O communication is executed within the system. This task evaluates students' understanding of I/O organization and system integration concepts. 3. **Active Learning Activity (ALA): I/O Performance Evaluation using System Monitoring Tools (10 Marks):** • In this assignment, students will perform a technical evaluation of various input and output devices such as keyboard, storage drive, printer, and network interface using free system monitoring and analysis tools (e.g., Windows Task Manager, Resource Monitor, Linux System Monitor, or similar open-source utilities).Students will observe device activity including CPU utilization, data transfer rate, interrupt activity, and I/O throughput during different operations such as file transfer, printing, or network communication. Based on experimental observations, students will analyze and identify the most suitable I/O data transfer technique — Programmed I/O, Interrupt-Driven I/O, or Direct Memory Access (DMA) — for each device. The assignment must include screenshots of software observations, performance comparison tables, technical analysis, and justified conclusions. Students shall compile their findings into a structured technical report and submit it in PDF format through the GMIU Web Portal.		

Suggested Specification table with Marks (Theory):100

Distribution of Theory 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	Understand computer organization and CPU basics.
CO2	Explain 8085 architecture and instruction execution.
CO3	Analyze addressing modes, interrupts, and ALP.
CO4	Evaluate memory organization and performance.
CO5	Apply I/O organization and data transfer methods.

Instructional Method:

The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

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

Students will use supplementary resources such as online videos, NPTEL/SWAYAM videos, e-courses, Virtual Laboratory.

The internal evaluation will be done on the basis of Active Learning Assignment.

Practical/Viva examination will be conducted at the end of semester for evaluation of performance of students in laboratory.

Reference Books:

[1] *Computer Organization and Architecture: Designing for Performance, by William Stallings, Pearson Education.*
[2] *Computer System Architecture*, by M. Morris Mano, Pearson Education.
[3] *Structured Computer Organization*, by Andrew S. Tanenbaum, Pearson Education.
[4] *Computer Organization*, by Carl Hamacher, Zvonko Vranesic, and Safwat Zaky, McGraw-Hill Education.
[5] *Microprocessor Architecture, Programming and Applications with 8085*, by Ramesh S. Gaonkar, Penram International Publishing.

Suggested Assessment Guidelines:

Module-1: Fundamentals of Computer Organization

Instruction Cycle and System Component Analysis (20 Marks)		
Criteria	**Description**	**Marks**
Concept Understanding	Clear understanding of CPU, memory, registers, buses, and I/O components.	5
Instruction Cycle Explanation	Proper explanation of Fetch–Decode–Execute cycle with correct sequence.	5
Register and Bus Role Analysis	Correct identification of register usage, control unit activity, and data transfer.	5
Diagram and Presentation	Neat diagram, structured explanation, and accurate technical presentation.	5
	Total	10

Module-2: 8085 Microprocessor Architecture

8085 Architecture and Instruction Execution Analysis (20 Marks)		
Criteria	**Description**	**Marks**
Architecture Understanding	Correct explanation of 8085 functional blocks and internal structure.	5
Instruction Execution Observation	Proper observation of basic instruction execution using simulator or example.	5
Register and Flag Analysis	Accurate recording of changes in registers, memory, and flag status.	5
Pin Diagram Understanding	Correct explanation of important 8085 pins and their functions.	5
	Total	20

Module-3: Dynamic Programming and Search Techniques

8085 Programming and Addressing Mode Implementation (20 Marks)		
Criteria	**Description**	**Marks**
Instruction Implementation	Correct execution of data transfer, arithmetic, and logical instructions.	5
Addressing Mode Application	Proper use of immediate, register, direct, and indirect addressing modes.	5
Branching and Stack Operations	Correct implementation of branching, looping, and stack-based instructions.	5
Output and Execution Analysis	Accurate output, register changes, flag status, and execution screenshots.	5
	Total	20

Module-4: Memory Organization

Memory Hierarchy and Cache Performance Analysis (20 Marks)		
Criteria	**Description**	**Marks**
Memory Hierarchy Understanding	Correct explanation of registers, cache, main memory, and secondary memory.	5
Cache Mapping Implementation	Proper demonstration of direct, associative, and set-associative mapping.	5
Hit/Miss and Performance Analysis	Accurate analysis of cache hit, miss, access time, and performance impact.	5
Virtual Memory and Paging	Correct explanation of paging and logical-to-physical address translation.	5
	Total	20

Module-5: Input–Output Organization

I/O Organization and Data Transfer Technique Analysis (20 Marks)		
Criteria	**Description**	**Marks**
I/O Organization Understanding	Correct explanation of I/O modules, interfaces, and CPU–I/O communication.	5
Data Transfer Techniques	Proper explanation of programmed I/O, interrupt-driven I/O, and DMA.	5
Interrupt and DMA Analysis	Accurate analysis of interrupt handling, DMA operation, and data flow.	5
Diagram and Technical Clarity	Neat diagram, correct terminology, and clear structured presentation.	5
	Total	20