SEMESTER-III

COURSE 7: COMPUTER ORGANIZATION

Theory **Credits: 3** **3 hrs/week**

Course Objectives

1. To introduce foundational concepts of register transfer language and micro-operations, enabling understanding of basic computer organization.
2. To examine CPU architecture and the control unit's design through hardwired and microprogrammed approaches.
3. To explore various memory organization strategies, including hierarchy, cache mapping, and associative techniques.
4. To understand input-output systems and data transfer mechanisms using I/O interfaces and DMA methods.
5. To analyze arithmetic algorithms and the principles of parallel and pipelined processing.

Course Outcomes

At the end of the course, students will be able to:

1. Interpret register-level operations and perform arithmetic, logic, and shift micro-operations within a basic computer framework.
2. Illustrate CPU functionality, addressing modes, and control unit design with both hardware-based and microprogramming techniques.
3. Categorize types of memory and explain memory hierarchy, access methods, and mapping techniques.
4. Describe I/O organization, including asynchronous transfer modes, DMA, and interrupt-driven processing.
5. Apply arithmetic algorithms and demonstrate understanding of parallel and pipelined processing models.

Unit 1. Register Transfer Language and Micro Operations:
Introduction- Functional units, computer registers, register transfer language, register transfer, bus and memory transfers, arithmetic, logic and shift micro-operations, arithmetic logic shift unit. Basic Computer Organization and Design: Instruction codes, instruction cycle. Register reference instructions, Memory – reference instructions, input – output and interrupt.

Unit 2. CPU and Micro Programmed Control:
Central Processing unit: Introduction, instruction formats, addressing modes.Control memory, address sequencing, design of control unit - hard wired control, micro programmed control.

Unit 3. Memory Organization:
Memory hierarchy, main memory, auxiliary memory, associative memory, cache Memory and mappings.

Unit 4. Input-Output Organization:
Peripheral Devices, input-output interface, asynchronous datatransfer, modes of transfer-programmed I/O, priority interrupt, direct memory access, Input – Output Processor (IOP).

Unit 5. Computer Arithmetic:
Data representation- fixed point, floating point, addition and subtraction, multiplication and division algorithms.

Text Books:

1. Computer Systems Architecture, M. Moris Mano, 3rd edition, Pearson/ PHI
2. Computer Organization and Architecture, William Stallings, 8th edition, Pearson/PHI

Reference Books:

1. Computer Organisation and Architecture, V. Rajaraman, T. Radha Krishnan, PHI
2. Computer Organization and Architecture Hamacher, Vranesic, Zaky, 5th edition, McGraw Hill

Activities:

Outcome: Interpret register-level operations and perform arithmetic, logic, and shift micro-operations within a basic computer framework.

Activity: Use a simulation tool (e.g., Logisim or Digital Works) to design a simple 4-bit ALU that performs:

- Addition, subtraction
- AND, OR
- Logical and arithmetic shifts

Evaluation Method: Demonstration-based assessment where students explain each operation using test inputs. Include a short worksheet with expected vs. actual outputs.

Outcome: Illustrate CPU functionality, addressing modes, and control unit design with both hardware-based and microprogramming techniques.

Activity: Create a flowchart or block diagram showing:

- CPU components (ALU, registers, control unit)
- Addressing modes (immediate, direct, indirect, etc.)
- Control unit types (hardwired vs. microprogrammed)

Evaluation Method:

Oral presentation or peer-reviewed poster session. Use a rubric to assess clarity, completeness, and correct identification of modes and control logic on 1 10-point scale.

Outcome: Categorize types of memory and explain memory hierarchy, access methods, and mapping techniques.

Activity: Build a memory hierarchy chart using coloured cards or digital tools. Include:

- Registers, cache, RAM, secondary storage
- Access methods (direct, associative, etc.)
- Mapping techniques (direct, associative, set-associative)

Evaluation Method: Quiz with matching and short-answer questions. Include a scenario-based question where students choose the best memory type for a given task.

Outcome: Describe I/O organization, including asynchronous transfer modes, DMA, and interrupt-driven processing.

Activity: Role-play simulation: Assign students' roles (CPU, DMA controller, I/O device) and simulate data transfer using cards or tokens to represent data and control signals.

Evaluation Method: Reflective worksheet where students describe the flow of control and data in each mode. Include a diagram labelling key signals and steps.

Outcome: Apply arithmetic algorithms and demonstrate floating point arithmetic operations.

Activity: Use a spreadsheet or visual tool to simulate floating point arithmetic operations.

Evaluation Method: Submission of simulation results with explanation. Evaluate the report on a 10-point scale.

SEMESTER-III

COURSE 7: COMPUTER ORGANIZATION

Practical **Credits: 1** **2 hrs/week**

List of Experiments:

1. Simulate Register Transfer Using Logisim
2. Build a microprogrammed control unit using Logisim or VHDL (GHDL).
3. Simple Register Transfers and Arithmetic Operations using EMU8086 Simulator
4. Simulate a Simple Instruction Cycle in Emu8086
5. Demonstrate Addressing Modes (Immediate, Register, Direct, Indexed) in Emu8086
6. Implement Subroutines Using CALL and RET in Emu8086
7. Simulate DMA Transfer in Ripes or Logisim
8. Simulate Booth's Algorithm for multiplication and restoring division in Python/C or Logisim
9. Write assembly language code for A+B*(C-D) using various instruction formats in any open-source assembler.
10. Write assembly language code for A+B*C using various addressing modes in any open-source assembler.