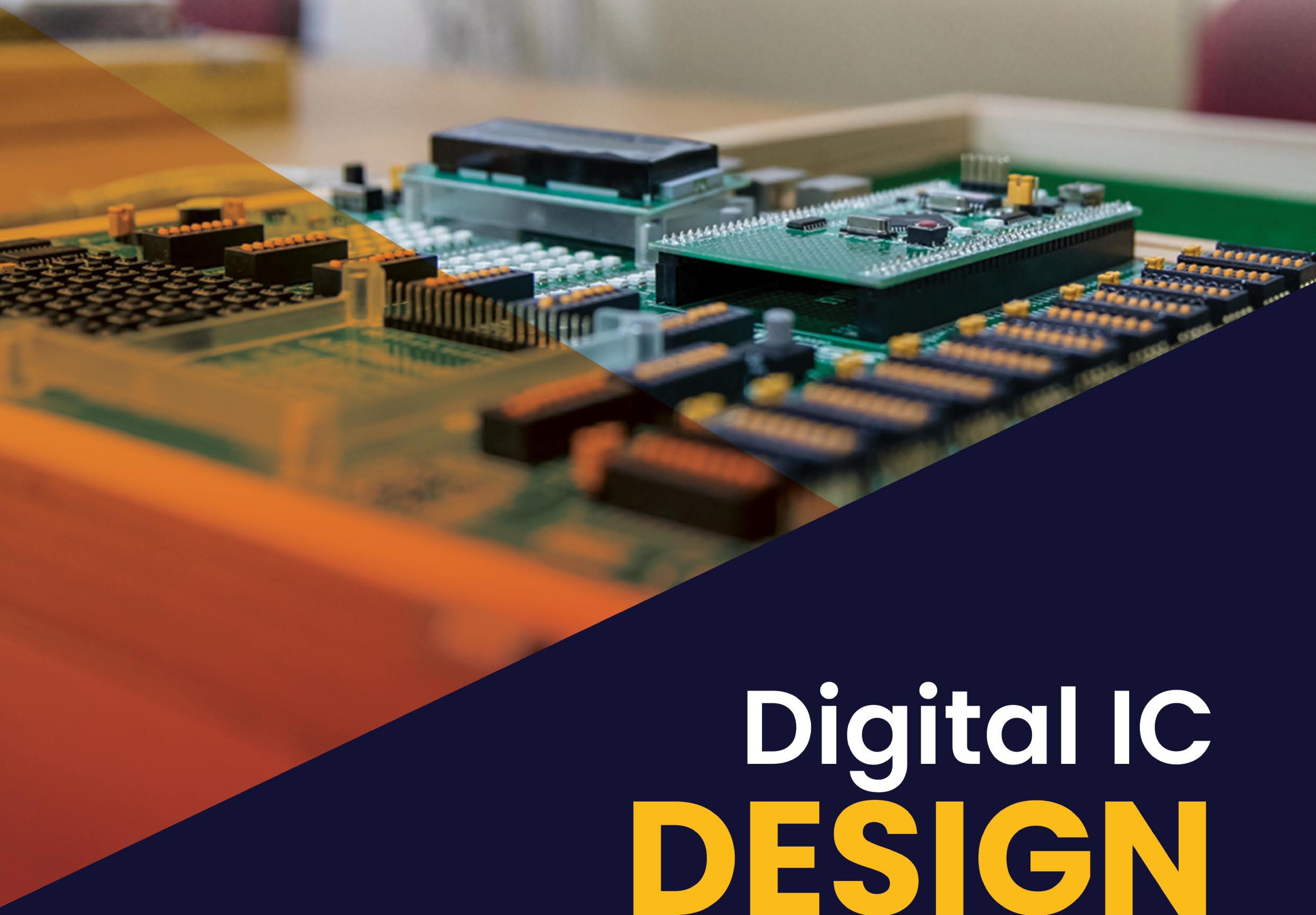
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

Deemed to be University U/S 3 of the UGC Act, 1956

SCHOOL OF ELECTRONICS ENGINEERING

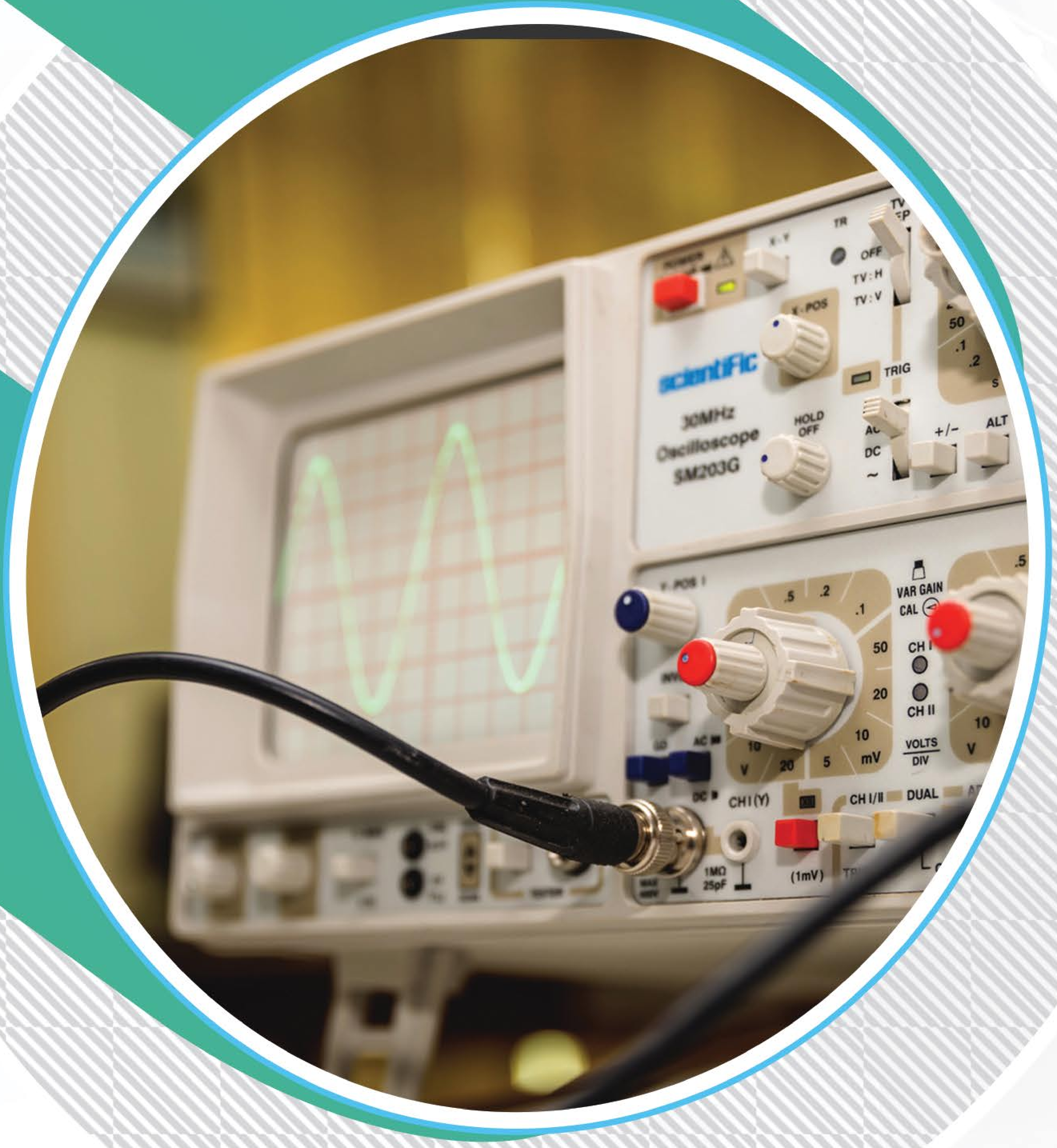
Established : 1997



Digital IC DESIGN

RTL to GDS flow using EDA Tools

SUMMER TRAINING 2025



COURSE OBJECTIVE

This course provides a comprehensive exploration of RISC-V core design and FPGA implementation. Students will learn to translate architectural specifications into synthesizable hardware using HDLs like Verilog. Practical exercises will cover memory interfacing and peripheral integration. By the end, students will be able to design, simulate, and deploy a functional RISC-V core on an FPGA platform, gaining valuable skills in embedded system development and hardware acceleration. The course emphasizes hands-on experience and real-world application of RISC-V principles.

- ✓ CERTIFICATES ON COMPLETION
- ✓ HYBRID MODE TRAINING
- ✓ 20 HOURS | 40+ PROJECT

COURSE OUTCOME

RISC-V Implementation

Students will successfully develop and demonstrate a working RISC-V processor core on an FPGA, capable of executing basic programs and interacting with peripherals

Verilog Proficiency

Students will demonstrate proficiency in using hardware description languages (Verilog) and FPGA synthesis tools to translate architectural designs into implementable hardware.

System Integration

Students will be able to integrate a RISC-V core with memory and peripherals, debug hardware and software interactions, and analyze system performance on an FPGA platform.

COURSE FEE

KIIT Affiliation	Rs. 4000/-*
Other Affiliation	Rs 5000/-*

*No Fooding & Lodging

INSTRUCTORS

- Dr. Jitendra Kr. Das
- Dr. Sarita Nanda
- Dr. Subir Kr. Maity

MODULE DESCRIPTOR FOR THE COURSE

MODULE 1

Fundamentals of Computer Architecture and RISC-V
Von Neumann Architecture, CPU, Memory, I/O, Instruction Set Architecture (ISA), Register, Memory Addressing. History and Philosophy of RISC-V, Advantages of RISC-V, RISC-V Specifications (Base ISA, Extensions), RV32I, RV64I, and other common extensions.

MODULE 2

RISC-V Base Integer Instruction Set (RV32I): Register Conventions, Arithmetic Instructions (ADD, SUB, MUL, DIV), Logical Instructions (AND, OR, XOR), Shift Instructions (SLL, SRL, SRA), Immediate Instructions.
Memory access instructions: Load (LB, LH, LW), store (SB, SH, SW), addressing modes.
Control flow instructions: branch instructions (BEQ, BNE, BLT, BGE), jump instructions (JAL, JALR), basic assembly programs (e.g., Sum of Numbers, Factorial), intro to assembly language calling conventions.

MODULE 3

Verilog HDL fundamentals: Introduction to Verilog HDL, gate level, data flow and behavioral modeling style, and arithmetic and logical operators in Verilog.
Concatenation operator and ternary operator: Procedural block, if-else, and case statement in Verilog. Memory design (RAM) in Verilog.
Design of finite state machines (FSM): (Mealy type) with an example (3-bit sequence detector). Implementation of a simple 8-bit ALU using dataflow and behavioral modeling style and its testing using a testbench.

MODULE 4

Implementing a Basic RISC-V Core in Verilog and testing
Basic Core Architecture (Fetch → Decode → Execute → Writeback): Designing the Instruction Memory. Implementing a Basic ALU for Addition & logical operations, Implementing a simple RISC-V core in Verilog, and testing it with a basic addition/subtraction instruction. and modify the core to support logical operations. Connecting Instruction Memory to the Core. Simulating the core with a test bench. Load the hex file into the Verilog Core and verify the execution of an assembly program in Verilog. Modify the RISC-V Verilog core and testbench to store the output in memory and verify correctness.
Introduction to RISC-V GCC toolchain: Toolchain setup, writing a C code, and generating a HEX file. Loading the HEX file into the generated RISC-V Verilog core and testing it with a testbench. Sample program: addition of a series of numbers, factorial, multiplication using functions.
Adding GPIO peripheral to RISC-V core: Adding LEDs and switches to the RISC-V core. Test the core for GPIO operation. FPGA implementation of the core with GPIO support.