# School of Electrical and Electronics Engineering
# Department of (Electronics Engineering)
# Electronics and Computer Science

**Multidisciplinary Minor (MDM) Track: Integrated Circuit Design (IC design)**

| Sr. No. | Semester | Course Code | Course Name | Offered To (Name of the Department) |
|---|---|---|---|---|
| 1 | III | 24EE01TH0305 | Basics of Chip Design using Verilog HDL | All - CSE, EC,EE, BME,IT |
| 2 | IV | 24EE01TH0405 | MIPS Processor Design and Testing | All -CSE, EC,EE, BME,IT |
| 3 | V | 24EE01TH0505 | Chip Verification using System Verilog | All -CSE, EC,EE, BME,IT |
| 4 | VI | 24EE01TH0605 | VLSI Physical Design | All -CSE, EC,EE, BME,IT |

| | | | | |
|---|---|---|---|---|
| **Course Code** | **24EE01TH0305** | | | |
| **Category** | **Multidisciplinary Minor -1** | | | |
| **Course Title** | **Basics of Chip Design using Verilog HDL** | | | |
| **Scheme & Credits** | **L** | **P** | **Credits** | **Semester** |
| | **3** | **0** | **3** | **III** |

Course Outcomes:

Upon completion of this course, students will demonstrate the ability to:

1. Realize the digital systems using Verilog HDL
2. Apply the testing strategies using HDL
3. Write a synthesizable HDL code for EDA tools
4. Analyze the timing issues in digital systems
5. Implement the digital systems on FPGA platforms.

# Syllabus:

**Module I (6 Hrs):**

Digital System Design Flow, FPGA Architecture, Introduction to FPGA Development Board, Introduction to HDL, Basic Language Elements, Syntax and Semantics of HDL

**Module II (8 Hrs):**

Gate level, Dataflow and Behavioral Modeling for combinational circuits like Multiplexer, De-multiplexer, Encoder-Decoder, Flip-Flop, Counter, Writing Test Benches and Handling Text files to test the Circuits.

**Module III (6 Hrs):**

Design and Analysis of Standard Combinational Blocks, Algorithm to Architectural Translation for Arithmetic Circuits-Adders, Subtractor, Multiplier, Divider, Shifter, ALU and Comparator

**Module IV (6 Hrs):**

Design and analysis of standard sequential blocks, Finite State Machine Design.

**Module V (6 Hrs):**

Design of Data Path and Control unit with Case Studies.

**Module VI (6 Hrs):**
Logic Synthesis and Optimization Techniques for Area, Power and Delay, Timing analysis-Setup and Hold Violations, Synthesis of HDL code on FPGA platforms, Concepts of Critical Path Delay

**Text Book:**


1. Verilog HDL: A Guide to Digital Design and Synthesis; Samir Palnitkar, Prentice Hall PTR; 2nd Edition

2. Fundamentals of Digital Logic with Verilog; Stephen Brown and ZvonkoVranesic; McGraw Hill, 2nd Edition


**Reference Books:**


1. Digital Systems Design Using Verilog; Charles Roth, Lizy K. John, ByeongKil Lee; Cengage Learning 2nd Edition

2. A Verilog HDL Primer: J Bhaskar; Star Galaxy Publishing; 2nd Edition.

| Course Code | 24EE01TH0405 | | | |
|---|---|---|---|---|
| Category | Multidisciplinary Minor -2 | | | |
| Course Title | MIPS Processor Design and Testing | | | |
| Scheme & Credits | L | P | Credits | Semester |
| | 3 | 0 | 3 | IV |

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. Understand the architecture and working of a single-cycle and multi-cycle MIPS processor, including the data path and control path design.
2. Design individual components of the MIPS processor in Verilog HDL and integrate them into a complete processor.
3. Implement and simulate the MIPS data path and control unit in Verilog HDL on FPGAs.
4. Develop testbenches in Verilog HDL to verify the functionality of individual modules and the integrated processor.
5. Perform step-by-step debugging and testing of a MIPS processor using waveform analysis and behavioral simulation.

Syllabus:

**Module-I: Introduction to MIPS Processor and Design Flow**

Introduction to computer system and its sub modules, Introduction to RISC and CISC paradigm, overview of the MIPS Instruction Set Architecture (ISA), registers, instruction formats, and addressing modes, single-cycle and multi-cycle execution, breakdown of the MIPS data path into instruction fetch, decode, execute, memory, and writeback stages.

**Module-II: Designing the MIPS ALU and Register File**

Hardware modeling for Arithmetic and logical operations performed by the MIPS ALU, implementation of a register file, testbench creation for verifying the ALU and register file operations.

**Module-III: Instruction Fetch and Decode Stage Implementation**

The instruction fetch stage, design of program counter (PC) and instruction memory components, instruction decode stage, generation of control signals, Integration of the instruction memory with the register file.

**Module-IV: Execution, Memory, and Writeback Stages**

Implementation of the execution stage, ALU operation and branching logic, memory access stage, design of control logic for load store instructions, implementation of write back stage, testing of R-type, I-type, and J-type instructions using data and control path.

**Module-V: Integration and Testing of Single-Cycle MIPS Processor**

Assembling the complete single-cycle MIPS processor, develop a complete Verilog testbench for the entire processor, run test programs, and debug issues that arise during simulation, analyze timing diagrams and waveforms using simulation tools.

**Module-VI**

Overview of Pipelining and Parallel processing concepts, multiprocessors and its characteristics, Input/Output Subsystem:-Interfaces and buses, I/O Operations, Designing I/O Systems, Overview of Domain-Specific Architectures

**Text Books:**

1. Computer Organization and Design Edition - The Hardware/Software Interface, David A. Patterson, John L. Hennessy, 5th Edition, 2014.

**Reference Books**

1. Computer Architecture and Organization; J. P. Hayes; Third Edition (Fifth Reprint), McGraw Hill, 2012.
2. Computer Architecture And Parallel Processing; Kai Hawang, Faye A. Briggs, McGraw Hill, 2012