# Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering and Technology, Baramati

## Department of Electronics and Telecommunication Engineering

**Honor courses E&TC Engineering A.Y. 2026-27**
**(As per NEP 2020)**
**S.Y. B.Tech. 2025 Pattern**
**T.Y. B.Tech. 2024 Pattern**
**Final Year B.Tech. 2023 Pattern**

# Vidya Pratishthan's
# Kamalnayan Bajaj Institute of Engineering and Technology
## Board of Studies: E&TC Engineering
### Syllabus: Honor / Double Minor Program
### 2025 Pattern S.Y. Sem III A.Y. 2026-27
### SEMESTER-III

| Semester | Course Code | Course Name | Teaching Scheme | | | Examination Scheme and Marks | | | | | | | Credits | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | TH | PR | TUT | CAA | ISE | ESE | TW | PR | OR | Total | TH | PR | TUT | Total |
| III | ET25281TH | Advanced Digital Design | 2 | - | - | 10 | - | 60 | - | - | - | 70 | 2 | - | - | 3 |
| | ET25281PR | Advanced Digital Design | - | 2 | - | - | - | - | - | 30 | - | 30 | - | 1 | - | |
| | Total | | 2 | 2 | - | 10 | - | 60 | - | 30 | - | 100 | 2 | 1 | - | 3 |

HOD E&TC
Dr. J. S. Rangole

Dean Academics
Dr. S. M. Bhosle

Controller of Examination
Dr. A. H. Kolekar

Principal
Dr. S. B. Lande

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING
VPKBIET, BARAMATI

# Vidya Pratishthan's
# Kamalnayan Bajaj Institute of Engineering and Technology
## Board of Studies: E&TC Engineering
### Syllabus: Honor / Double Minor Program
### 2025 Pattern S.Y.Sem IV A.Y. 2026-27
### SEMESTER-IV

| Semester | Course Code | Course Name | Teaching Scheme | | | Examination Scheme and Marks | | | | | | | Credits | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | TH | PR | TUT | CAA | ISE | ESE | TW | PR | OR | Total | TH | PR | TUT | Total |
| IV | ET25291TH | ASIC Design and SoC | 2 | - | - | 10 | - | 60 | - | - | - | 70 | 2 | - | - | 3 |
| | ET25291PR | ASIC Design and SoC | - | 2 | - | - | - | - | - | 30 | - | 30 | - | 1 | - | |
| | Total | | 2 | 2 | - | 10 | - | 60 | - | 30 | - | 100 | 2 | 1 | - | 3 |

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING
VPKBIET, BARAMATI

| HOD E&TC | Dean Academics | Controller of Examination | Principal |
|---|---|---|---|
| Dr. J. S. Rangole | Dr. S. M. Bhosle | Dr. A. H. Kolekar | Dr. S. B. Lande |

# Vidya Pratishthan's
# Kamalnayan Bajaj Institute of Engineering and Technology
## Board of Studies: E&TC Engineering
### Syllabus: Honor / Double Minor Program
### 2024 Pattern T.Y. Sem V A.Y. 2026-27

| Semester | Course Code | Course Name | Teaching Scheme | | | Examination Scheme and Marks | | | | | | Credits | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | TH | PR | TUT | CAA | ISE | ESE | TW | PR | OR | Total | TH | PR | TUT | Total |
| V | ET24381TH | System Verilog for Design and Verification | 3 | - | - | 10 | 30 | 60 | - | - | - | 100 | 3 | - | - | 4 |
| | ET24381PR | System Verilog for Design and Verification | - | 2 | - | - | - | - | - | 30 | - | 30 | - | 1 | - | |
| | Total | | 3 | 2 | - | 10 | 30 | 60 | - | 30 | - | 130 | 3 | 1 | - | 4 |

HOD E&TC
Dr. J. S. Rangole

Dean Academics
Dr. S. M. Bhosle

Controller of Examination
Dr. A. H. Kolekar

Principal
Dr. S. B. Lande

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING
VPKBIET, BARAMATI

# Vidya Pratishthan's
# Kamalnayan Bajaj Institute of Engineering and Technology
## Board of Studies: E&TC Engineering
### Syllabus: Honor / Double Minor Program
### 2024 Pattern T.Y. Sem VI A.Y. 2026-27

| Semester | Course Code | Course Name | Teaching Scheme | | | Examination Scheme and Marks | | | | | | | Credits | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | TH | PR | TUT | CAA | ISE | ESE | TW | PR | OR | Total | TH | PR | TUT | Total |
| VI | ET24391TH | · Advanced Verification Methodologies and scripting | 3 | - | - | 10 | 30 | 60 | - | - | - | **100** | 3 | - | - | 4 |
| | ET24391PR | Advanced Verification Methodologies and scripting | - | 2 | - | - | - | - | | 30 | - | 30 | - | 1 | - | |
| | **Total** | | 3 | 2 | - | 10 | 30 | 60 | | 30 | - | 130 | 3 | 1 | - | 4 |

HOD E&TC
Dr. J. S. Rangole

Dean Academics
Dr. S. M. Bhosle

Controller of Examination
Dr. A. H. Kolekar

Principal
Dr. S. B. Lande

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING
VPKBIET, BARAMATI

# Vidya Pratishthan's
# Kamalnayan Bajaj Institute of Engineering and Technology
## Board of Studies: E&TC Engineering
### Syllabus: Honor / Double Minor Program
### 2023 Pattern B.Tech. A.Y. 2026-27
### SEMESTER-VIII

| Semester | Course Code | Course Name | Teaching Scheme | | | Examination Scheme and Marks | | | | | | | Credits | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | TH | PR | TUT | CAA | ISE | ESE | TW | PR | OR | Total | TH | PR | TUT | Total |
| VIII | ET23481 TH | Advanced CMOS VLSI Technology | 3 | - | - | 10 | 30 | 60 | - | - | - | 130 | 3 | - | - | 4 |
| | ET23481 PR | Advanced CMOS VLSI Technology | - | 2 | - | - | - | - | - | 30 | - | 30 | - | 1 | - | |
| | Total | | 3 | 2 | - | 10 | 30 | 60 | - | 30 | - | 130 | 3 | 1 | - | 4 |

HOD E&TC
Dr. J. S. Rangole

Dean Academics
Dr. S. M. Bhosle

Controller of Examination
Dr. A. H. Kolekar

Principal
Dr. S. B. Lande

**SEM III**

| **Advanced Digital Design** | |
|---|---|
| **Teaching Scheme:**<br><br>**TH: 02 Hrs./week**<br><br>**PR: 02 Hrs./week** | **Credits 03**<br><br>**Examination Scheme**<br><br>**In - Semester: -**<br>**End - Semester: 60 Marks**<br>**PR: 30 Marks**<br>**Activity:10 Marks** |

**Course Objectives:**

1. To use Verilog HDL and thus model tasks & functions at the behavioral level.
2. To Apply the state machines approach to design digital circuits.
3. To illustrate combinational circuits and sequential circuits using Verilog
4. Explain the types of programmable logic devices and building blocks of FPGA families and Implement digital circuits on FPGA

**Course Outcomes:**

1. Write Verilog HDL code and thus model tasks & functions at the behavioral level
2. Design the state machines using D and JK Flip Flops and model using Verilog
3. To model combinational circuits and sequential circuits using Verilog.
4. Differentiate FPGAs and Implement digital design on FPGA

**Course Contents**

**Unit I Verilog HDL**

Data Flow & Structural Modeling-Lexical Conventions - Ports and Modules – Operators - Gate Level Modeling - Data Flow Modeling - System Tasks & Compiler Directives - Test Bench. Behavioral level Modeling- Procedural Assignment Statements- Blocking and Non-Blocking Assignments -Tasks & Functions - Useful Modeling Techniques.

**Unit II State Machine Design**

Definition of state machines -State machine as a sequential controller- Analysis of state machines using D and JK flip-flops - Design of state machines- State table and State assignment - Transition/excitation table - excitation maps and equations - logic realization- Design examples, FSM Verilog modeling of Sequence detector - Serial adder - Vending machine. Design Guidelines for Sequential Circuits

**Unit III: Combinational Circuits and Sequential Circuits**

Adders: Ripple Carry Adder, Carry Look-Ahead Adder, Higher Bit Adders Using CLA, Carry Skip Adder, Carry Increment Adder, Carry Select Adder, Carry Save Addition, Multipliers: Sequential Multiplication, Array Multipliers, Booth's Multiplication, Multiplication Using Look-Up Table Sequential Circuits: BCD up-down counter, Synchronous and Asynchronous FIFO – Single port and Dual port ROM and RAM -

**Unit VI: FPGA Architecture and Design Flow**

Types of Programmable Logic Devices: PLA, PAL, CPLD, FPGA Architecture, Programming Technologies, Chip I/O, Programmable Logic Blocks, Fabric and Architecture of FPGA. Impact of logic block functionality on FPGA performance, Model for measuring delay.

FPGA Families: Artix, Kintex, Virtex, EDA tools, Design Flow, FPGA Design Guidelines

**Books & Other Resources:**

**Text Books:**

1. Samir Palnitkar, Verilog HDL, 2nd Edition, Pearson Education, 2009.

2. Advanced Digital System Design A Practical Guide to Verilog-based FPGA and ASIC Implementation.

**Reference Books:**

1. Data sheets of Artix-7, Kintex-7, Virtex-7.

2. Digital Logic Design Using Verilog, Coding and RTL Synthesis by Vaibhav Taraate, Second Edition Springer

3. Advanced Digital Logic Design using Verilog, State Machines & Synthesis for FPGA - Sunggu Lee, Cengage Learning, 2012.

**NPTEL:**

1. Digital Design with Verilog By Prof. Chandan Karfa, Prof. Aryabartta Sahu | IIT Guwahati https://onlinecourses.nptel.ac.in/noc24_cs61/preview

2. System Design Through VERILOG By Prof. Shaik Rafi Ahamed | IIT Guwahati https://onlinecourses.nptel.ac.in/noc21_ee97/preview

| List of Laboratory Experiments/Assignments | |
|---|---|
| 1. | Introduction to FPGA design flow using Verilog with a multigate IC |
| 2. | Design and implementation of Adder and compare the performance with other adder circuits. |
| 3. | Design and implement a multiplier and compare its performance with other multiplier circuits. |
| 4. | Design and implement BCD up-down counter |
| 5.<br>6.<br>7. | Design and implement FIFO<br>Design and implement Serial adder/Sequence Detector/Vending Machine<br>Mini Project: Design and implement Microcontroller OR Communication Protocol or any equivalent design on BASYS 3 Kit. A group of 4-5 students will design the individual project modules and integrate the complete design. |

**List of Activities for reference:**

**1.** Intel Certification on Verilog
https://learning.intel.com/developer/learn/courses/235/verilog-hdl-basics
2. Survey of Semiconductor Mission of India and Fabrication plants in India
3. Presentation on VLSI Industries, Job Profiles and Alumni in VLSi Industries
4. Development of system design using verilog and upload to github
5. https://www.udemy.com/topic/verilog-hdl-programming/
6. NPTEL Courses: Digital Design with Verilog By Prof. Chandan Karfa, Prof. Aryabartta Sahu | IIT Guwahati https://onlinecourses.nptel.ac.in/noc24_cs61/preview
7. System Design Through VERILOG By Prof. Shaik Rafi Ahamed | IIT Guwahati https://onlinecourses.nptel.ac.in/noc21_ee97/preview

**Vidya Pratishthan's**

**Kamalnayan bajaj Institute of Engineering & Technology, Baramati**

**Department of Electronics and Telecommunication**

# Course Title: ASIC Design and System on Chip

| Teaching Scheme:<br>TH : 02 Hrs./week<br>PR : 02 Hrs./week | Credits<br>03 | Examination Scheme<br><br>End-of-Semester: 60 Marks<br>PR : 30 Marks,<br>Activity: 10 marks |
|---|---|---|

## Course Objectives:

- To provide a fundamental understanding of ASIC design principles and System-on-chip (SoC) architecture.
- To familiarize students with the design flow of digital integrated circuits using CAD tools.
- To introduce practical skills for implementing designs using Microwind, MATLAB, and PYNQ boards.
- To bridge the gap between theoretical knowledge and practical applications in VLSI and embedded systems.

---

## Course Outcomes:

CO1: Analyze the working principles and design techniques of digital CMOS circuits

CO2: Apply fabrication technology concepts, including layout design and design issues

CO3: Design and evaluate System on Chip (SoC) architectures

CO4: Explore and implement emerging trends in ASIC and SoC design,

**Unit 1: Digital CMOS Circuits**

Types of ICs: Full Custom, Semi-Custom, and Programmable Logic Devices. Types of ASICs: Standard Cell-Based, Gate Array-Based, Full Custom Design.

MOSFET parasitic, Technology scaling, Channel length modulation, Hot electron effect, Velocity saturation. CMOS Inverter, Device sizing, CMOS combinational logic design, Power dissipations, Power delay product, Body Effect, Rise and fall times, Latch-Up effect, Transmission gates

**Unit 2: Fabrication Technology for ASICs**

Lambda rules, Design Rule Check, Fabrication methods of circuit elements, Layout of cell, Library cell designing for NAND & NOR, Circuit Extraction, Electrical Rule Check, Layout Vs. Schematic, Post-layout Simulation and Parasitic extraction, Design Issues like Antenna effect, Electro migration effect, Cross talk and Drain punch through, Timing analysis

**Unit 3: System on Chip (SoC) Design**

SoC: Components and Architectures, Processor Cores in SoC: Soft Cores vs. Hard Cores. Communication Protocols: UART, I2c, SPI, AMBA (APB, AHB), AXI. Memory Subsystems and IP Integration. High-Level Synthesis Tools for SoC Design.

**Unit 4: Emerging Trends in ASIC and SoC Design**

Low Power ASIC Design Techniques.Heterogeneous SoC Architectures, Role of Machine Learning in SoC Development, AI and ML Accelerators in SoC, Chiplets and 3D IC Integration, Hardware Acceleration using FPGA and PYNQ Boards.Green and Sustainable ASIC/SoC Design, Case Studies: Real-World Applications of ASICs and SoCs.

---

## Practical Syllabus

**Practicals Using Microwind**

1. **Introduction to Microwind**: Design and simulate CMOS inverter, NAND and NOR layout.
2. Design and simulate Half Adder & Full Adder
3. **Design and Simulate 2:1 Mux using logic gates & transmission gates**
4. **Design and Simulate One bit SRAM Cell**

5. Develop a project to implement any System on Chip using Simulink, HDL Coder and FPGA Board

---

## Textbooks & References

1. "Application-Specific Integrated Circuits" by Michael John Sebastian Smith.
2. "CMOS VLSI Design: A Circuits and Systems Perspective" by Neil H.E. Weste and David Harris.
3. "Digital Design and Computer Architecture" by David Harris and Sarah Harris.
4. "System-on-Chip Design with Arm Cortex-M Processors" by Joseph Yiu.
5. MATLAB and Microwind Tool Manuals.

---

## SEM V

| ET23301 SYSTEM VERILOG FOR DESIGN AND VERIFICATION | | |
|---|---|---|
| **Teaching Scheme:**<br>**TH : 03 Hrs./week**<br>**PR:2 Hr/week** | **Credits**<br>**04** | **Examination Scheme**<br>**In - In-Semester: 30 Marks**<br>**End-Semester: 60 Marks**<br>**PR-30 Marks**<br>**Activity: 10 Marks** |

**Course Objectives:**

1. To understand System Verilog Fundamentals and explore enhancements over Verilog, including dynamic casting, tasks, and functions.
2. To develop Testbenches and Verification Environments and implement testbenches using interfaces, modports, clocking blocks, and event schedulers.
3. To apply Advanced Verification Techniques and utilize constrained random verification (CRV), functional coverage, and assertions to verify digital designs.
4. To explore Functional Coverage and Assertion-Based Verification

**Course Outcomes: On completion of the course, the learner will be able to–**

1. Demonstrate the use of System Verilog data types for digital system design and functional verification.
2. Demonstrate the skill of writing test benches for the design of digital systems and connecting them with the design
3. Apply object-oriented analysis in System Verilog and advanced features of System Verilog
4. Verify and analyze the complete systems through robust verification methods such as assertion-based verification.

**Course Contents**

**Unit I: Introduction to SystemVerilog (8 Hours)**

Overview of SystemVerilog, SystemVerilog standards and enhancements over Verilog, Data types: literals, enumerated types, structures, unions, Operators, expressions, and procedural statements, Control flow: loops, conditional statements, Processes in SystemVerilog: always, always_ff, always_comb, initial, final, Packages and type conversion: dynamic casting, static casting, Tasks and functions.

**Unit II: Connecting the Testbench and Design: ( 6 Hours )**

Verilog interface signals and their limitations. SystemVerilog interfaces and modports, SystemVerilog port connections and interface instantiation, Tasks and functions in interfaces, Verilog Event Scheduler vs. SystemVerilog Event Scheduler, Clocking blocks and input-output skews, Typical testbench environment: stimulus, DUT, monitors, checkers, Developing a verification plan

**Unit II: Object Oriented Analysis In System Verilog ( 7 Hours )**

Introduction to Object-Oriented Programming (OOP) in SystemVerilog, Objects, properties, methods, and constructors, Encapsulation, data hiding, and inheritance, Polymorphism and method overriding, Randomization techniques: random constraints, randomization methods, Constraint control: inline constraints, disabling random variables, Scope-based randomization and controlling constraint solver, Inter-process synchronization and communication techniques, Scheduling semantics and clocking blocks

**Unit IV: Functional Coverage and Assertion-Based Verification (6 Hours)**

**Functional Coverage:** Coverage Definition: Code Coverage vs. Functional Coverage, Covergroups: Creating Cover Group Instances, Coverpoints and bins (explicit, implicit, illegal, ignore), Cross coverage and intersect operations, Coverage analysis and built-in coverage methods

**Assertion-Based Verification:** Introduction to Assertions: Immediate vs. Concurrent Assertions, Writing assertion properties, Sequences and sequence composition, Logical operations (and, or, intersect) for complex assertions, SystemVerilog Assertion API and Coverage API

**UNIT V: Advanced Verification Concepts (6 Hours)**

Introduction to TLM: Concept of abstraction in testbenches, TLM vs. RTL-based simulation, TLM Interfaces and Communication Mechanisms, Implementation of simple TLM-based testbenches. Scoreboards and Checkers: Need, Concept of a Reference Model in verification, Designing a Scoreboard: FIFO-based comparison, Transaction matching, Data integrity checks, Self-checking Testbenches using Scoreboards, Debugging and Analyzing Verification Code, Debugging Constrained Randomization Issues, Waveform Debugging in SystemVerilog, Handling Race Conditions and Timing Issues, Assertions for Debugging – Debugging complex verification scenarios, Testbench Optimization Strategies, Best Practices for Writing Readable and Maintainable Testbenches

**UNIT VI: Industry Applications and Practical Case Studies (6 Hours)**

Case Studies: FIFO Verification, ALU Verification, Memory Controller Verification, Best Practices for Scalable and Reusable Testbenches, Overview of Industry-Standard EDA Tools: Synopsys VCS, Cadence Xcelium, Mentor QuestaSim

**Text Books:**

**Text Books:**

1. SystemVerilog for Verification: A Guide to Learning the Testbench Language Features by Chris Spear, Greg Tumbush, *Publisher:* Springer
2. Verilog HDL: A Guide to Digital Design and Synthesis by Samir Palnitkar, *Publisher:* Pearson

**Reference Books:**

1. Writing Testbenches: Functional Verification of HDL Models *by* Janick Bergeron, *Publisher:* Springer
2. Principles of VLSI RTL Design: A Practical Guide by Sanjay Churiwala *Publisher:* Springer
3. System Verilog Assertions and Functional Coverage: Guide to Language, Methodology and Applications *by* Ashok B. Mehta, *Publisher:* Springer
4. Digital Design and Computer Architecture by David Money Harris, Sarah L. Harris, *Publisher:* Morgan Kaufmann

**SWAYAM Course:**

**Digital VLSI Testing, By Prof. Santanu Chattopadhyay (IIT Kharagpur)**

**https://onlinecourses.nptel.ac.in/noc25_ee25/preview**

**VLSI Design Verification and test**

**https://archive.nptel.ac.in/courses/117/103/117103125/**

## Lab Assignments

## SYSTEM VERILOG FOR DESIGN AND VERIFICATION

Students will perform all ten practicals or any five practicals and a mini-project

## Practicals

**1. Tasks and Functions in SystemVerilog**

- Develop a SystemVerilog program with tasks and functions.
- Use task chaining and function overloading for modular verification.

**2. SystemVerilog Interfaces and Port Connections**

- Design a Verilog interface and understand its limitations.
- Implement SystemVerilog interface with modports and references.

**3. Clocking Blocks and Skew Analysis**

- Implement clocking blocks in SystemVerilog.
- Explore input and output skews with different timing constraints.

**4. Verification Environment Design (Basic Testbench)**

- Develop a basic SystemVerilog testbench with a DUT, driver, monitor, and scoreboard.
- Implement transaction-based communication using tasks and functions.

**5. Object-Oriented Programming in SystemVerilog**

- Implement class properties, methods, constructors, and method chaining.
- Demonstrate inheritance and polymorphism in verification environments.

**6. Constrained Randomization and Control**

- Implement random constraints and inline constraints.
- Use randomization control (weighting, disabling, priority constraints).

**7. Inter-Process Synchronization and Communication**

- Implement event-based communication in SystemVerilog using semaphores and mailboxes.
- Develop a scheduler for inter-process synchronization.

**8. Functional Coverage in SystemVerilog**

- Implement covergroups, coverpoints, and bins for a simple ALU.
- Analyze cross-coverage and conditional expressions (iff statements).

**9. Immediate and Concurrent Assertions**

- Implement immediate assertions for signal monitoring.
- Develop concurrent assertions with properties and sequences.

**10. Advanced Assertion-Based Verification (ABV)**

- Implement sequence composition (and, or, intersect) for FSM verification.
- Use illegal bins and ignore bins for coverage refinement.

---

# Mini-Projects

**Here are some sample mini-projects to be performed along with any five experiments above.**

## Project 1: FIFO Verification Using SystemVerilog

Develop a FIFO verification environment using transaction-based modeling.
Implement functional coverage and assertions for overflow, underflow, and reset.

## Project 2: UART Protocol Verification Using SystemVerilog

Build a SystemVerilog testbench to verify a UART transmitter and receiver.
Use interface-based communication and implement random stimulus generation.

## Project 3: ALU Verification with Functional Coverage

Verify a 4-bit ALU supporting ADD, SUB, AND, OR, XOR operations.
Implement functional coverage for different operation scenarios.

## Project 4: Memory Controller Verification Using OOP and UVM Basics

Develop a class-based testbench to verify a simple RAM module.
Implement randomized transactions using constrained random verification.

## Project 5: Assertion-Based Verification of a Traffic Light Controller

Implement assertions for state transitions in an FSM-based traffic light controller.
Use covergroups to ensure all transitions are tested.

**SEM VI**

| **ET24391 Advanced Verification Methodologies and Scripting** | | |
|---|---|---|
| **Teaching Scheme:**<br>**TH : 03 Hrs./week**<br>**PR : 02 Hrs./week** | **Credits**<br>**04** | **Examination Scheme**<br>**In - Semester: 30 Marks**<br>**End-of-Semester: 60 Marks**<br>**PR : 30 Marks,**<br>**Activity: 10 marks** |

**Course Objectives:**

The objectives of this course are:

1. To introduce Universal Verification Methodology (UVM) and its role in modern VLSI verification
2. To understand the UVM testbench architecture, components, and TLM communication mechanisms
3. To learn PERL scripting for automating verification tasks, regression testing, and log analysis.
4. To develop Python-based verification solutions, including EDA tool integration, data visualization, and coverage analysis.
5. To explore advanced UVM concepts, including register abstraction, debugging, coverage closure, and industry best practices.
6. To explore with real-world UVM-based projects and industry case studies.

**Course Outcomes: On completion of the course, the learner will be able to–**

At the end of the course, the student will be able to

**CO1:** Explain the need for standardized verification methodologies and implement basic UVM testbench components for structured verification.

**CO2:** Design reusable and scalable UVM testbench architectures using TLM interfaces, agents, and configurable environments.

**CO3:** Develop and apply PERL scripts for regression automation, log analysis, and integration with UVM-based testbenches.

**CO4:** Apply Python programming to automate verification flows, generate coverage reports, and visualize verification metrics.

**CO5:** Implement advanced UVM concepts such as virtual sequences, RAL models, debugging strategies, and coverage closure methods for optimized verification.

**CO6:** Analyze real-world verification case studies and apply emerging trends such as AI/ML to design end-to-end automated verification solutions.

## Unit I: Foundations of Modern Verification & UVM Basics (7 Hrs)

Directed Testing vs. Constrained Random Testing, Need for Standardized Verification Methodologies, Overview of SystemVerilog Verification Features, UVM Testbench Architecture, UVM Components: Transaction, Sequence & Sequencer, Driver, Monitor, Agent, Scoreboard, Checker, UVM Phases: Build Phase, Run Phase, Cleanup Phase, Factory Pattern: Registration and Object Configuration

## Unit II: UVM Testbench Design & Transaction-Level Modeling (7 Hrs)

UVM Data Flow and Communication Mechanisms, Transaction-Level Modeling (TLM): Interfaces, Ports, Exports, Analysis Ports, TLM FIFO, Building Reusable UVM Agents and Environments, Hierarchical Testbench Development, Configurability and Reuse in UVM, Verification Planning and Testbench Scalability

## Unit III: PERL for Verification & Automation (7 Hrs)

PERL Basics: Syntax, Variables, Arrays, Hashes, Control Structures, File Handling, Regular Expressions, Pattern Matching, Advanced Topics: Process Management, Directory Access, Formats, Writing PERL Scripts for Regression Automation, Testbench Log Analysis, Report Parsing & Report Generation, Interfacing PERL with SystemVerilog/UVM, Case Study: Regression and Report Automation with PERL

## Unit IV: Python for Verification & Data-Driven Automation (7 Hrs)

Object-Oriented Programming in Python, File Handling and Log Analysis, Automating UVM Regression Flows, Python Integration with EDA Tools, Coverage Analysis & Report Generation, Data Visualization of Verification Metrics, Python for Test Pattern Generation, Case Study: Automated Verification Flow using Python

## Unit V: Advanced UVM Techniques & Debugging Strategies (7 Hrs)

UVM Sequences & Virtual Sequences, Creating Custom and Layered Stimulus, Virtual Sequencers, UVM Register Abstraction Layer (RAL): Register Models, Configuration, Accessing & Manipulating Registers, Debugging in UVM: Waveform Debugging, Logging, Tracing Testbench Failures, Performance Considerations: Simulation Optimization, Resource Management, Achieving Coverage Closure: Strategies and Best Practices, Regression Testing & Result Analysis

**Unit VI: Industry Applications, Case Studies & Emerging Trends (6 Hrs)**

Real-World UVM Projects: AXI Bus Protocol Verification, UVM-Based Cache Memory Verification, SoC Verification Flows, RISC-V Processor Verification, Complete UVM-Based End-to-End Verification Flow, Industry Practices: Regression Farms, Coverage Closure in Practice, Emerging Trends: Portable Stimulus, Formal + Simulation Hybrid Flows, AI and Machine Learning in VLSI Verification, Mini Project: End-to-End Automated Verification Environment Integrating UVM, PERL, Python, Flow of UVM-Based End-to-End Verification, Industry Trends & Future of Verification, AI and Machine Learning in VLSI Verification

**Text Books:**

1. Larry Wall, Tom Christiansen, John Orwant, Programming PERL, Oreilly Publications, Fourth Edition, 2012.
2. Christian B Spear, SystemVerilog for Verification: A guide to learning the Test bench language features, Springer publications, Third Edition, 2012.
3. Michael H. Gonzalez Harbour – *"Python for Test Engineers"*

---

**Reference Books:**

1. Ray Salmei, The UVM Primer: A Step-by-Step Introduction to the Universal Verification Methodology, First Edition, Boston Light Press, 2013.
2. Vanessa R. Copper, Getting started with UVM: A Beginner's Guide, Verilab Publishing, First Edition, 2013
3. Richard Sharp – *"Practical Electronics for Inventors"*
4. Mark Lutz – *"Learning Python",* Publisher: O'Reilly Media, Fifth Edition (2013)

**Practicals: Any 10 Practicals or Any 5 Practicals and a miniproject**

1. Basic UVM Testbench Implementation (Create a simple UVM testbench with transactions, drivers, and monitors)
2. Build a UVM-Based Verification Environment (Integrate sequencers, agents, and scoreboards)
3. Implement Coverage-Driven Verification in UVM (Functional and code coverage using SystemVerilog and UVM)
4. Write and Debug a UVM Register Model (Configure UVM RAL for register verification)
5. Write a Basic PERL Script for Log Parsing (Extract error messages and warnings from simulation logs)
6. Use PERL for Regression Testing Automation (Automate execution of multiple verification tests)
7. Implement Testbench Data Processing with PERL (Analyze waveform dump files and extract key parameters)
8. Write Python Scripts for EDA Tool Automation (Run simulations, collect results, and generate reports)
9. Automate UVM Regression Execution Using Python (Trigger multiple tests and analyze results)
10. Python for Coverage Analysis and Data Visualization (Plot verification coverage data using Matplotlib)
11. Interface Python with SystemVerilog Testbenches (Use Python to send/receive data from a SystemVerilog environment)
12. Mini-Project: Complete Automated Verification Flow (Integrating UVM, PERL, and Python for a real-world verification task)

**SEM VIII**

| **ET23481 Advanced CMOS VLSI Technology** | | |
|---|---|---|
| **Teaching Scheme:**<br>**TH : 03 Hrs./week**<br>**PR : 02 Hrs./week** | **Credits**<br>**04** | **Examination Scheme**<br>**In - Semester: 10 Marks**<br>**End-of-Semester: 60 Marks**<br>**PR : 30 Marks,**<br>**Activity: 10 marks** |

**Course Objectives:**The objectives of this course are:

1. Introduce advanced CMOS process technologies and device scaling challenges.
2. Provide knowledge of low-power circuit design methodologies for modern VLSI.
3. Familiarize students with interconnect modeling, parasitics, and reliability issues in nanometer CMOS.
4. Develop understanding of advanced memory design including SRAM, DRAM, Flash, and emerging memories.
5. Train students in device modeling and simulation for nanoscale CMOS technologies.
6. Expose students to future trends in CMOS and beyond-CMOS VLSI, including 3D integration and neuromorphic/quantum devices.

**Course Outcomes: On completion of the course, the learner will be able to–**

- **CO1**: Analyze scaling challenges and process innovations in advanced CMOS.
- **CO2**: Apply low-power design techniques in CMOS circuits and systems.
- **CO3**: Evaluate interconnect parasitics and reliability issues in nanoscale CMOS.
- **CO4**: Design and analyze advanced memory architectures in CMOS and emerging technologies.
- **CO5**: Use device modeling and simulation techniques for nanoscale CMOS circuits.
- **CO6**: Explore and assess emerging trends in CMOS and beyond-CMOS VLSI design.

**Unit I: Advanced CMOS Process Technology**

Deep submicron and nanoscale CMOS challenges, high-k dielectrics and metal gate technology, strained silicon and mobility engineering, silicon-on-insulator (SOI) and fully depleted SOI (FDSOI) technology, multi-gate devices including FinFETs and gate-all-around FETs, short channel effects and scaling limits.

**Unit II: Low Power CMOS Design**

Sources of power dissipation in CMOS, static and dynamic power reduction techniques, multi-threshold and multi-voltage design methods, clock gating and power gating approaches, adiabatic logic and energy recovery techniques, dynamic voltage and frequency scaling (DVFS) for power-performance optimization.

**Unit III: Interconnects and Reliability Issues**

Interconnect scaling and delay models including RC and RLC models, crosstalk and signal integrity problems in deep submicron designs, electromigration and IR drop issues, joule heating in interconnects, use of low-k and ultra low-k dielectrics, interconnect materials such as copper, carbon nanotubes, and graphene, reliability challenges in nanoscale regimes.

**Unit IV: Memory Design in CMOS**

Design and stability analysis of SRAM cells, DRAM cells and refresh mechanisms, Flash memory architectures including NOR and NAND, emerging non-volatile memories such as MRAM, FeRAM, PCM and ReRAM, techniques for low power and high-density memory design.

**Unit V: Device Modeling and Simulation**

Compact MOS models including BSIM, EKV and PSP, noise modeling in CMOS devices, process variation and statistical design analysis, Monte Carlo simulations for variability, technology CAD (TCAD) for device and process simulation, reliability modeling including NBTI, PBTI, hot carrier injection (HCI) and time-dependent dielectric breakdown (TDDB).

**Unit VI: Future Directions in CMOS VLSI**

Three-dimensional ICs and monolithic integration, through-silicon vias (TSVs) and interposer technologies, neuromorphic circuits implemented in CMOS, compatibility of quantum devices with CMOS technology, beyond-CMOS devices including CNTFETs, tunnel FETs and spintronics, sustainable and green VLSI design practices.

**Text Books:**


1. Neil H. E. Weste, David Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, Pearson
2. Sung-Mo Kang, Yusuf Leblebici, *CMOS Digital Integrated Circuits: Analysis and Design*, McGraw Hill
3. Rabaey, Chandrakasan, Nikolic, *Digital Integrated Circuits: A Design Perspective*, Pearson.


**Reference Books:**


1. Yuan Taur, Tak H. Ning, *Fundamentals of Modern VLSI Devices*, Cambridge University Press.
2. A. P. Chandrakasan, R. W. Brodersen, *Low Power CMOS Design*, Wiley-IEEE Press.
3. M. J. S. Smith, *Application-Specific Integrated Circuits*, Addison-Wesley.
4. S. M. Sze, Kwok K. Ng, *Physics of Semiconductor Devices*, Wiley.


**NPTEL Courses**

1. NPTEL – Advanced VLSI Design by Prof. S. Chakravarty (IIT Kharagpur).

2. NPTEL – Low Power VLSI Circuits and Systems by Prof. S. Mohan (IIT Madras).

**Virtual Labs (IIT/NITs)**

- VLSI Design Virtual Lab (IIT Roorkee) – vlab.co.in
- CMOS Inverter and Combinational Circuits Simulation – Virtual Electronics Lab (Amrita Vishwa Vidyapeetham).
- Digital VLSI Design Virtual Lab – NITK Surathkal.
- **Analog and Digital VLSI Circuit Design Lab** – IIT Bombay (available via VLAB portal).

# Lab Practicals – Advanced CMOS VLSI Technology

**List of Practicals:** Any 10 Practicals or Any 5 Practicals and a miniproject using tools: Cadence Virtuoso, Tanner EDA, Microwind (for intro FinFET/DRAM), or TCAD (for device-level analysis) or any Evaluation or open source tool

**Practical 1: Layout of CMOS Transmission Gate**

Design and simulate a transmission gate, verify functionality as an analog switch.

**Practical 2: Layout and Simulation of CMOS Schmitt Trigger**

Implement a Schmitt Trigger layout, analyze noise immunity.

**Practical 3: Low Power CMOS Inverter using Multi-Vth Technique**

Implement inverter layouts with high-Vth and low-Vth transistors, compare leakage power.

**Practical 4: Layout of 1T1C DRAM Cell**

Design DRAM cell layout with capacitor and access transistor, analyze refresh requirement.

**Practical 5: Interconnect RC Parasitics Extraction**

Design interconnects of varying lengths/widths, extract RC parasitics, and measure propagation delay.

**Practical 6: Layout and Simulation of FinFET Inverter**

Draw FinFET-based inverter layout and compare delay, power, and leakage with bulk CMOS inverter.

**Practical 7: Layout of Low-Power Clock Gated Flip-Flop**

Implement layout of a clock-gated D flip-flop and measure power savings compared to conventional design.

**Practical 8: Reliability Study in Layout (Electromigration/IR Drop)**

Simulate interconnects with different widths/materials, analyze electromigration, and IR drop impact.

**Practical 9: Layout of Non-Volatile Memory Cell (ReRAM or MRAM)**

Implement the conceptual layout of emerging non-volatile memory (e.g., ReRAM/MRAM) and compare with Flash.

**Practical 10: Layout of 3D IC Interconnect using TSVs**

Draw a layout structure representing Through-Silicon Vias (TSVs) and simulate inter-die connectivity.