Ramdeobaba University (RBU)

RBU

RAMDEOBABA UNIVERSITY, NAGPUR

Formerly Shri Ramdeobaba College of Engineering & Management (RCOEM) Est. 1984

School of Electrical and Electronics Engineering

Scheme and Syllabus
Master of Technology
(VLSI Design)
Session:2026-27

About the Programme

The Master of Technology (M.Tech.) in VLSI Design is a specialized postgraduate programme designed to develop highly skilled professionals capable of designing, implementing, verifying, and optimizing modern integrated circuits and semiconductor systems. The programme provides a comprehensive understanding of contemporary Very Large Scale Integration (VLSI) design methodologies, encompassing digital, analog and mixed-signal circuit design, hardware modelling, ASIC and FPGA design flows, embedded systems, design verification, physical design, low-power design, and emerging semiconductor technologies.

The curriculum integrates strong theoretical foundations with extensive laboratory practice using industry-standard Electronic Design Automation (EDA) tools, modern hardware platforms, and software development environments. Students gain practical exposure to the complete integrated circuit design lifecycle—from device-level concepts and RTL development to verification, implementation, system integration, and prototype validation—thereby preparing them for complex engineering challenges in the semiconductor industry.

The programme is carefully aligned with the evolving requirements of the global semiconductor ecosystem and supports emerging technology domains. Through project-based learning, laboratory-intensive courses, internships, and research-oriented activities, students develop analytical, design, implementation, and problem-solving skills required for next-generation chip design and electronic system development.

Graduates of the programme are well prepared for rewarding careers in semiconductor product companies, electronic design automation (EDA) organizations, semiconductor design service companies, embedded system industries, research laboratories, startups, and academia. They are equipped to contribute to technological innovations across diverse application domains.

Unique Selling Proposition (USP)

A contemporary, industry-oriented M.Tech. programme that integrates advanced VLSI design, ASIC and FPGA implementation, embedded systems, verification methodologies, physical design, low-power design, and emerging semiconductor technologies through extensive laboratory practice, project-based learning, and industry-relevant curriculum.

Key Features:

- Comprehensive curriculum covering the complete VLSI design flow, including digital, analog and mixed-signal IC design, hardware modelling, ASIC and FPGA implementation, design verification, embedded systems, physical design, and low-power design.
- Extensive laboratory experience using industry-standard Electronic Design Automation (EDA) tools, FPGA platforms, embedded development environments, and modern engineering software.
- Specialized electives in emerging domains.
- Strong emphasis on practical implementation through engineering practice laboratories, capstone projects, internships, and industry-oriented assignments.
- Exposure to modern semiconductor design methodologies, verification techniques, and hardware-software co-design practices aligned with current industrial requirements.
- Opportunities for interdisciplinary research, and innovation through advanced laboratories and faculty-guided research activities.
- Curriculum designed to support lifelong learning, professional development, and continuous adaptation to rapidly evolving semiconductor technologies.

Opportunities:

Graduates of the programme can pursue diverse technical and research careers in the global semiconductor and electronics industries, including positions such as:

- ASIC Design Engineer
- RTL Design Engineer
- FPGA Design Engineer
- Design Verification Engineer
- Analog and Mixed-Signal Design Engineer
- Physical Design Engineer
- Embedded Systems Engineer
- SoC Design and Integration Engineer
- Hardware Security Engineer
- Low-Power Design Engineer
- Electronic Design Automation (EDA) Engineer
- Semiconductor Research Engineer
- Technical Consultant and Product Development Engineer
- Doctoral Researcher and Academician

(CHOICE BASED CREDIT SYSTEM)

Programme Educational Objectives (PEOs)

PEO1: Apply advanced knowledge of VLSI design, semiconductor devices, embedded systems, and electronic design automation to analyze, design, implement, and optimize integrated circuits and electronic systems.

PEO2: Develop innovative, reliable, and energy-efficient semiconductor solutions by employing modern engineering tools, research methodologies, and industry-standard design practices to address real-world engineering challenges.

PEO3: Demonstrate professional competence, ethical responsibility, effective communication, teamwork, and leadership while pursuing lifelong learning, research, innovation, entrepreneurship, or higher studies in semiconductor and allied technologies.

Programme Outcomes (POs)

PO1: Apply advanced knowledge of semiconductor devices, VLSI design, embedded systems, and electronic design automation to solve complex engineering problems.

PO2: Analyze, model, design, verify, and optimize integrated circuits, embedded systems, and semiconductor-based solutions using modern engineering principles and industry-standard tools.

PO3: Independently conduct research, investigate complex engineering problems, develop innovative solutions, and evaluate their technical, economic, environmental, and societal impact.

PO4: Effectively communicate technical ideas through professional reports, research publications, presentations, and collaborative engineering practices while demonstrating ethical responsibility and commitment to lifelong learning.

Programme Specific Outcomes (PSOs)

PSO1: Design and develop analog, digital, mixed-signal, ASIC, FPGA, and electronic systems using contemporary VLSI design methodologies and electronic design automation (EDA) tools.

PSO2: Develop embedded electronic systems by integrating hardware, software, and emerging technologies to address contemporary engineering challenges.

SESSION 2026-27

I SEMESTER

Sr. No.	Category	Code	Course	L	P	Credits	Internal Evaluation (Th/Lab)	Mid Sem Exam (30)	End Sem Exam (50)	Continuous Evaluation (25)	Total Marks
1	PCC	26EE51TH1101	CMOS Digital Circuit Design	4	0	4	20	30	50	-	100
2	PCC	26EE51PR1101	CMOS Digital Circuit Design Lab	0	2	1	25	-	-	25	50
3	PCC	26EE51TH1102	Hardware Modelling with HDL	4	0	4	20	30	50	-	100
4	PCC	26EE51PR1102	Hardware Modelling with HDL Lab	0	2	1	25	-	-	25	50
5	PCC	26EE51TH1103	Semiconductor Devices	4	0	4	20	30	50	-	100
6	PCC	26EE51PR1103	Semiconductor Devices Lab	0	2	1	25	-	-	25	50
7	PCC	26EE51TH1104	Embedded System and RTOS	4	0	4	20	30	50	-	100
8	PCC	26EE51PR1104	Embedded System and RTOS Lab	0	2	1	25	-	-	25	50
9	PCC	26EE51TH1105	Design for Testability	3	0	3	20	30	50	-	100
10	SEC	26EE51PR1106	Laboratory Practice I	0	2	1	25	-	-	25	50
			Total	**19**	**10**	**24**					

DEPARTMENT OF ELECTRONICS ENGINEERING
M. TECH. (VLSI DESIGN)
II SEMESTER

Sr. No.	Category	Code	Course	L	P	Credits	Internal Evaluation (Th/Lab)	Mid Sem Exam (30)	End Sem Exam (50)	Continuous Evaluation (25)	Total Marks
1	PCC	26EE51TH1201	Analog & Mixed-Signal IC Design	4	0	4	20	30	50	-	100
2	PCC	26EE51PR1201	Analog & Mixed-Signal IC Design Lab	0	2	1	25	-	-	25	50
3	PCC	26EE51TH1202	System Verilog & UVM for Verification	4	0	4	20	30	50	-	100
4	PCC	26EE51PR1202	System Verilog & UVM for Verification Lab	0	2	1	25	-	-	25	50
5	PCC	26EE51TH1203	Digital ASIC Design	3	0	3	20	30	50	-	100
6	PCC	26EE51PR1203	Digital ASIC Design Lab	0	2	1	25	-	-	25	50
7	PCC	26EE51TH1204	SoC Design	3	0	3	20	30	50	-	100
8	PCC	26EE51PR1204	SoC Design Lab	0	2	1	25	-	-	25	50
9	SEC	26EE51PR1205	Laboratory Practice II	0	2	1	25	-	-	25	50
10	AEC	26EE51PR1206	Seminar	0	2	1	25	-	-	25	50
11	OE	26EE51TH1207	Open Elective	3	0	3	20	30	50	-	100
			Total	**17**	**12**	**23**					

Course Code	Open Elective-I
26EE51TH1207-01	VLSI Signal Processing
26EE51TH1207-02	High Level Synthesis
26EE51TH1207-03	Machine Learning Engineering
26EE51TH1207-04	Industry offered Elective
26EE51TH1207-05	Equivalent Swayam NPTEL/MOOCs Course approved by the Department

DEPARTMENT OF ELECTRONICS ENGINEERING
M. TECH. (VLSI DESIGN)
III SEMESTER

Sr. No.	Category	Code	Course	L	P	Credits	Internal Evaluation (Th/Lab)	Mid Sem Exam (30)	End Sem Exam (50)	Continuous Evaluation (25)	Total Marks
1	MLC	26EE51TH1301	Research Methodology & IPR	3	0	3	20	30	50	-	100
2	PEC	26EE51TH1302	Programme Elective-I	4	0	4	20	30	50	-	100
3	PEC	26EE51PR1302	Programme Elective-I Lab	0	2	1	25	-	-	25	50
4	PEC	26EE51TH1303	Programme Elective-II	4	0	4	20	30	50	-	100
5	PEC	26EE51TH1304	Programme Elective-III	3	0	3	20	30	50	-	100
6	PRJ	26EE51PR1305	Capstone Project Phase-I	0	8	4	100	-	-	100	200
7	VSEC	26EE51TH1306	Participatory Learning	0	2	1	25	-	-	25	50
8	SEC	26HS02TH1375	Technical Communication	1	0	1	10	15	-	25	50
			Total	**15**	**12**	**21**					

OR

Sr. No	Category	Code	Course	L	P	Credits	Internal Evaluation (Th/Lab)	Mid Sem Exam (30)	End Sem Exam (50)	Continuous Evaluation (25)	Total Marks
1	MLC	26EE51TH1307	Research Methodology& IPR/MOOC course	3	0	3	20	30	50	-	100
2	PCC	26EE51TH1308	MOOCs Course approved by the Department	3	0	3	20	30	50	-	100
3	PCC	26EE51TH1309	MOOCs Course approved by the Department	3	0	3	20	30	50	-	100
4	PRJ/VSEC	26EE51PR1310	Industry Internship /Research Internship	-	24	12	100	-	-	100	200
			Total	**9**	**24**	**21**					

Course Code	Programme Elective – I
26EE51TH1302-01/26EE51PR1302-01	FPGA Based System Design/ FPGA Based System Design Lab
26EE51TH1302-02/26EE51PR1302-02	Low Power VLSI Design/ Low Power VLSI Design Lab
26EE51TH1302-03/26EE51PR1302-03	RF Circuit Design/ RF Circuit Design Lab

Course Code	Programme Elective – II
26EE51TH1303-01	MEMS Design and Fabrication
26EE51TH1303-02	TinyML & Edge AI System Design
26EE51TH1303-03	VLSI Physical Design
26EE51TH1303-04	Equivalent Swayam NPTEL/MOOCs Course approved by the Department

Course Code	Programme Elective – III
26EE51TH1304-01	Nanoelectronics
26EE51TH1304-02	Hardware Security
26EE51TH1304-03	Advanced Computer Architecture
26EE51TH1304-04	Embedded Linux and Device Drivers
26EE51TH1304-05	Equivalent Swayam NPTEL/MOOCs Course approved by the Department

DEPARTMENT OF ELECTRONICS ENGINEERING
M. TECH. (VLSI DESIGN)

IV SEMESTER

Sr. No.	Code	Course	L	P	Credits	Internal Evaluation (Th/Lab)	Mid Sem Exam (30)	End Sem Exam (50)	Continuous Evaluation (25)	Total Marks
1	26EE51PR1401	Capstone Project Phase-II	0	24	12	200	-	-	200	400
		Total	**0**	**24**	**12**					

OR

Sr. No.	Code	Course	L	P	Credits	Internal Evaluation (Th/Lab)	Mid Sem Exam (30)	End Sem Exam (50)	Continuous Evaluation (25)	Total Marks
1	26EE51PR1402	Industry Internship / Research Internship	0	24	12	200	-	-	200	400
		Total	**0**	**24**	**12**					

SYLLABUS OF SEMESTER I, M. Tech. (VLSI DESIGN)

Course Code	26EE51TH1101			
Category	PCC			
Course Title	CMOS Digital Circuit Design			
Scheme & Credits	L	P	Credits	Semester
	4	0	4	I

Course Outcomes:

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

CO1: Analyze nanoscale MOS transistor metrics, short-channel effects, and non-planar multi-gate architectures (FinFETs, GAAFETs, Nanosheets) for static and dynamic IC operation.

CO2: Calculate interconnect parasitics, RC timing delays, charge-sharing hazards, and power dissipation across diverse dynamic and static CMOS logic structures.

CO3: Design optimized datapath subsystems (adders, multipliers, shifters, dynamic memory, ALUs) and FSM controllers subject to area, power, and timing budgets.

CO4: Evaluate sub-micron scaling limitations, noise margins, clock skew/jitter, and low-power management architectures (clock/power gating).

CO5: Synthesize, simulate, and generate physical IC layouts for digital blocks using industry-standard EDA platforms, verifying DRC, LVS, and parasitic extraction.

Syllabus:

Module I: Advanced Semiconductor Devices & Submicron Scaling (07 Hours)

Introduction to planar CMOS logic and scaling laws (Constant Field vs. Constant Voltage). Short-Channel Effects (SCEs): DIBL, Drain-Induced Barrier Lowering, Velocity Saturation, Hot-Carrier Injection, Gate-Induced Drain Leakage (GIDL), and Subthreshold Conduction. Silicon-on-Insulator (SOI) technology: Fully Depleted (FDSOI) vs. Partially Depleted (PDSOI). Advanced 3D Transistor Architectures: FinFET operational physics, Gate-All-Around (GAAFET), Nanosheet, and Forksheet configurations.

Module II: Circuit Characterization & Performance Estimation (08 Hours)

Inverter VTC characteristics, Noise Margins, and Transistor Sizing. Parasitic Estimation: Resistance extraction, Capacitance components (junction, overlap, gate, sidewall). Delay Analysis: RC Delay Models, Elmore Delay Model, Linear Delay Model, and Logical Effort Methodology for critical path optimization. Power Dissipation: Static leakage currents, Dynamic switching power, Short-circuit power. Low-Power Foundations: Multi-VDD design, Clock-Gating architectures, and Power-Gating (Sleep Transistors). Introduction to Dynamic Voltage and Frequency Scaling (DVFS).

Module III: Combinational Logic Families & Circuit Optimization (07 Hours)

Static CMOS Logic design, Ratioed Logic (Pseudo-NMOS), and Pass-Transistor Logic (PTL). Transmission Gate (TG) logic and complementary TG circuits. Dynamic CMOS Logic: Precharge-Evaluate mechanics, Charge Sharing hazards, Noise Margins, and Dynamic Leakage. High-Performance Dynamic Styles: Domino Logic, Dual-Rail Domino, and Zipper CMOS. I/O Structures and driving large capacitive loads.

Module IV: Sequential Logic Circuits & Clocking Strategies (08 Hours)

Static vs. Dynamic Latches and Flip-Flops. Timing Metrics: Setup time, Hold time, Propagation delay, Clock-to-Q delay. Dynamic Timing Hazards: Metastability, Clock Skew, and Clock Jitter

analysis. Advanced Sequencing Elements: Pulsed Latches, Sense-Amplifier Flip-Flops, Dual-Edge Triggered Flip-Flops. Clock Distribution Networks: H-tree topologies, Grid structures, and Clock-Tree Synthesis (CTS) concepts.

Module V: CMOS Subsystem & Datapath Architecture (08 Hours)

Datapath Arithmetic: High-speed Adders (Ripple Carry, Carry-Lookahead, Carry-Select, Carry-Bypass, Kogge-Stone Parallel Prefix). Multipliers: Array Multipliers, Booth Encoding, Wallace Tree Reduction Networks. High-Speed Shifters and Comparators. Control Unit Implementation: Finite State Machine (FSM) encoding and synthesis. Memory Array Architectures: SRAM 6T cell operation (Read/Write noise margins), DRAM 1T1C cell, Register Files, CAM, SIPO/FIFO structures, and ROM.

Module VI: Physical Design & EDA Layout Engineering (07 Hours)

The ASIC Backend: Introduction to Standard Cell methodology and the complete RTL-to-GDSII design flow. Overview of the VLSI fabrication flow from GDSII to packaged IC, including mask generation, wafer fabrication, wafer probing (wafer sort), packaging, and final testing. Layout Topologies: Stick diagrams, Euler paths, and Common-Centroid layout concepts. Interconnect Engineering: Cross-talk noise, RC interconnect modeling, and Buffer insertion. Power Delivery & Reliability Sign-off: IR Drop analysis, Electromigration (EM) rules, and deep-submicron reliability hazards (NBTI and TDDB). Physical Verification Flow: DRC, LVS, and Parasitic Extraction (PEX).

Textbooks:

1. Neil H. E. Weste, David Money Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th Edition, Pearson Education, 2015
2. Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd Edition, Prentice Hall India (PHI), 2003.
3. Sung-Mo Kang, Yusuf Leblebici, Chulwoo Kim, *CMOS Digital Integrated Circuits: Analysis and Design*, 4th Edition, McGraw-Hill Education, 2014.

Reference Books:

1. R. Jacob Baker, *CMOS: Circuit Design, Layout, and Simulation*, 4th Edition, Wiley-IEEE Press, 2019.
2. Ivan Sutherland, Bob Sproull, David Harris, *Logical Effort: Designing Fast CMOS Circuits*, Morgan Kaufmann, 1999.
3. Selected IEEE Transactions Literature: Recent publications on GAAFET/Nanosheet cell layout and dynamic clock gating architectures.

Course Code	26EE51PR1101			
Category	PCC			
Course Title	CMOS Digital Circuit Design Lab			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Indicative List of Experiments (The instructor may modify/add experiments based on the availability of EDA tools and emerging technologies.)

Experiment 1: CMOS Inverter Characterization
Objective:
Design and simulate a CMOS inverter and extract the Voltage Transfer Characteristics (VTC), switching threshold, propagation delay, and noise margins.

Experiment 2: CMOS Gate Design and Performance Analysis
Objective:
Design CMOS NAND, NOR, XOR and XNOR gates and compare transistor count, delay, power consumption, and propagation characteristics.

Experiment 3: Logical Effort Based Gate Sizing
Objective:
Design an inverter chain driving a large capacitive load and optimize transistor sizing using the Logical Effort methodology. Compare optimized and unoptimized designs in terms of delay.

Experiment 4: Power Characterization of CMOS Circuits
Objective:
Measure static, dynamic and short-circuit power under different supply voltages, frequencies and load capacitances.

Experiment 5: Dynamic Logic and Charge Sharing
Objective:
Design dynamic CMOS NAND/NOR logic and investigate charge sharing. Evaluate circuit stability using keeper (bleeder) transistors.

Experiment 6: Arithmetic Datapath Design
Objective:
Design and simulate one of the following:
16-bit Carry Lookahead Adder
16-bit Kogge-Stone Adder
16-bit Arithmetic Logic Unit (ALU)
Evaluate propagation delay, power consumption and area.
(Offer one option depending on available time.)

Experiment 7: CMOS Memory Cell Design
Objective:
Design and simulate a 6T SRAM bit-cell. Perform Read, Write and Hold analysis and determine Static Noise Margin (SNM) using Butterfly Curves.

Experiment 8: CMOS Layout Design
Objective:
Create full-custom layout of a CMOS inverter, NAND gate or D Flip-Flop and verify: DRC, LVS, Parasitic Extraction (PEX). Compare schematic and post-layout simulation results.

Experiment 9: Layout Optimization

Objective:

Optimize layout using guard rings, power rails, well contacts, device matching, routing optimization. Evaluate area and parasitic improvements.

Mini Design Project:

Objective:

Design, simulate and verify a moderately complex CMOS subsystem such as

- 4-bit ALU
- Register File
- FIFO Controller
- Multiplier
- Counter

Perform: Schematic simulation, layout DRC, LVS, PEX post-layout verification

Course Code	26EE51TH1102			
Category	PCC			
Course Title	Hardware Modelling with HDL			
Scheme & Credits	L	P	Credits	Semester
	4	0	4	I

Course Outcomes (COs)

CO1 : Develop basic RTL models of digital hardware using Verilog HDL constructs and modeling styles targeting FPGA platforms.
CO2 : Design synthesizable and parameterized RTL models for combinational and sequential logic circuits using modular and hierarchical design techniques, and verify their functionality using appropriate test benches.
CO3: Design and optimize finite state machine (FSM)-based digital controllers using appropriate state encoding techniques, RTL implementation styles, and functional verification methods.
CO4: Develop modular RTL architectures for complex data path and control path systems by integrating arithmetic units and register-transfer operations.
CO5: Implement FPGA-based digital systems by synthesizing RTL designs into optimized gate-level netlists and analysing technology mapping and FPGA resource utilization.
CO6 : Evaluate and optimize FPGA implementations using static timing analysis, timing closure techniques, resource utilization, power estimation, and design optimization methods to satisfy specified timing and performance constraints.

Syllabus:

Module I: FPGA Architecture, Design Flow and Fundamentals of Verilog HDL

Introduction to Hardware Description Languages (HDLs) and their role in modern digital system design, Comparison between FPGA and ASIC design methodologies, FPGA architecture blocks, FPGA design flow, Fundamentals of Verilog HDL including module structure, design hierarchy, identifiers, constants, numbers, data types, operators, compiler directives, modelling styles procedural blocks, blocking and non-blocking assignments

Module II: Modeling of Combinational Logic Circuits

Modeling of basic combinational logic circuits using continuous assignment statements and procedural modeling techniques, Arithmetic circuits – Adders , Subtractor, Multiplier, Divider, parameterized hardware modeling using parameters and generate statements, scalable RTL design, combinational circuit optimization, coding practices to avoid inferred latches, development of synthesizable RTL, functional simulation, waveform analysis, and verification using self-contained Verilog test benches.

Module III: Modeling of Sequential Logic Circuits

Modeling of sequential digital circuits using edge-triggered and level-sensitive storage elements, Design and implementation of latches, D, JK and T flip-flops, registers, shift registers, synchronous and asynchronous counters, up/down loadable counters, Modeling of memories including ROM, single-port RAM, dual-port RAM, and register files, Clocking methodologies, synchronous and asynchronous reset techniques, clock enable implementation, setup and hold time considerations, RTL coding guidelines for sequential circuits, synthesizable sequential logic, avoidance of race conditions and unintended latch inference, development of reusable sequential modules, generation of clocks and reset signals in test benches, self-checking verification techniques, and waveform-based debugging of sequential circuits.

Module IV: Finite State Machine (FSM) Design and Modelling

Concepts of finite state machines (FSMs) for digital controller design including Moore and Mealy machine models, state diagrams, RTL implementation of FSMs using one-process, two-process, and three-process coding styles. State encoding techniques including binary, Gray, one-hot, Johnson, and automatic synthesis-based

encoding. State minimization, state assignment, safe FSM design, reset strategies, and optimization of sequential controllers. Design and implementation of practical digital controllers such as sequence detectors, traffic light controllers, elevator controllers, vending machines, and protocol controllers. Functional verification of FSMs through simulation, state transition validation, waveform analysis, and debugging methodologies.

Module V: Modelling of Complex Datapath and Control Path Architectures

RTL modelling of Datapath and Control path architectures, Design of arithmetic and logical Datapath including arithmetic logic units (ALUs), barrel shifters, Multibit unsigned-signed Adders and Multipliers, multiply-accumulate (MAC) units, shift register files, buses, multiplexed data paths, and register transfer operations. Case Study- RISC V processor data path and control path design

Module VI: RTL Synthesis, Timing Analysis and Optimization

RTL synthesis methodology and conversion of Verilog descriptions into gate-level netlists, Technology mapping, synthesis constraints, resource utilization analysis, and generation of synthesis reports. Static timing analysis (STA), setup and hold time analysis, clock uncertainty, timing reports, timing closure techniques, and identification of critical paths. RTL optimization techniques, clock gating, fan-out optimization, and area-performance-power trade-offs. Coding styles for efficient synthesis, identification of synthesizable and non-synthesizable constructs, post-synthesis and post-implementation simulation, FPGA place-and-route, power estimation, hardware debugging using integrated logic analyzers, and best practices for developing optimized, portable, and reusable RTL designs suitable for FPGA.

Reference books:

1. The Verilog Hardware Description Language, Fifth Edition: Donald E. Thomas, Philip R.Moorby, Kluwer Academy Publisher. (2002).
2. Digital Systems Design Using VHDL, Second Edition: Charles H. Roth. Jr., L Kurian John,Cengage Learning, (2008).
3. Logic Synthesis using Synopsys, Second edition, P. Kurup and T. Abbasi, Kluwer, (1996)
4. Logic synthesis and verification algorithms: Gary D. Hachtel, Fabio Somenzi, Springer (1996)
5. An Engineering Approach to Digital Design: W. Fletcher. Prentice Hall

Course Code	26EE51PR1102			
Category	PCC			
Course Title	Hardware Modelling with HDL Lab			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Lab Experiments:

1. Design a parameterized N-bit adder architecture using Verilog HDL. Each student shall implement a different adder architecture (such as Ripple Carry Adder, Carry Look-Ahead Adder, Carry Select Adder, Carry Skip Adder, Carry Save Adder, Kogge-Stone Adder, Brent-Kung Adder, Han-Carlson Adder, etc.) for the same operand width. Develop a self-checking Verilog testbench for functional verification, perform RTL simulation, synthesize the design, and analyze the synthesis reports for area, timing (delay), and power consumption. Implement the design on an FPGA development board, validate its functionality on hardware, and compare the performance metrics of different adder architectures.
2. Four sensors continuously generate 8-bit parallel data representing measurements acquired from different locations in a monitoring system. Design a parameterized combinational packet formation unit using Verilog HDL that selects one of the sensor inputs based on a control signal, appends the corresponding Sensor ID, generates parity/error detection bits, compares the sensor reading with programmable threshold values, performs left or right shift operations for data scaling, and packs the processed information into a 16-bit data packet ready to be written into memory.

   Expected Design Components: Parameterized Multiplexer, Sensor ID Encoder/Packet Formatter, Threshold Comparator, Logical/Arithmetic Shifter, Parity/Error Bit Generator, Packet Generator (16-bit) ,Self-checking Testbench
3. A monitoring system consists of four remote sensors that continuously transmit serial measurement data. Design a parameterized data acquisition controller that receives the serial data from the selected sensor, converts it into N-bit parallel data using shift registers, generates a valid-data signal after receiving a complete data word, stores the received packet into a memory using sequential address generation, and provides status signals indicating memory full and data ready. Develop the complete RTL using modular Verilog HDL. Verify the functionality using a self-checking testbench. Synthesize the design, analyse area, timing, and power reports, and implement the controller on an FPGA development board.

   RTL Components: Shift Register, Bit Counter, Word Counter, Address Generator, Memory Interface, Register File, Status Register.
4. Design and implement a parameterized synchronous First-In First-Out (FIFO) buffer using Verilog HDL for temporary storage and transfer of streaming data between producer and consumer modules operating on a common clock. The FIFO shall support configurable data width and depth, sequential write and read operations, programmable Full, Empty, Almost Full, and Almost Empty status flags, and detection of Overflow and Underflow conditions. Implement circular buffer addressing using read and write pointers and appropriate control logic to ensure reliable data transfer.

   Develop the complete RTL using modular and reusable design practices. Create a self-checking Verilog testbench to verify normal operation as well as boundary conditions, including simultaneous read/write, FIFO full, FIFO empty, overflow, and underflow scenarios. Perform RTL simulation, synthesize the design, analyse FPGA resource utilization, timing, and power reports, and demonstrate the functionality on an FPGA development board.

5. Design and implement a parameterized Arithmetic Execution Unit (AEU) consisting of a multi-port Register File, Arithmetic Logic Unit (ALU), Barrel Shifter, Multiplier, and Output Multiplexer using modular Verilog HDL. The execution unit shall support configurable data width and perform arithmetic, logical, comparison, and shift operations based on programmable control signals. The Register File shall support simultaneous dual-read and single-write operations to facilitate efficient operand access.

   Develop reusable RTL modules using hierarchical design techniques. Verify the functionality using a self-checking Verilog testbench covering all arithmetic, logical, and register operations. Perform RTL synthesis, analyse FPGA resource utilization, timing, and power reports, and implement the execution unit on an FPGA development board.

6. A complete RTL implementation of a parameterized single-cycle RISC-V processor supporting a subset of the RV32I instruction set is provided. Study the processor architecture, including the data path and hardwired control unit, and analyse the execution of arithmetic, logical, load/store, branch, and jump instructions.

   Develop a self-checking Verilog testbench to execute a given instruction program, monitor the contents of the Program Counter, Register File, Data Memory, and control signals, and verify the correct execution of each instruction. Demonstrate the flow of instructions through the datapath by correlating simulation waveforms with the expected instruction execution sequence. Synthesize the processor, analyze FPGA resource utilization, timing, and power reports, and implement the design on an FPGA development board to validate its functionality.

7. A synthesizable Verilog HDL implementation of a DSP hardware architecture (20-Tap FIR Filter or 8-Point FFT Processor) is provided. Analyze the given RTL architecture and implement suitable optimization techniques to improve the design with respect to area, delay, throughput, and power consumption while preserving functional correctness.

   Develop a self-checking Verilog testbench to verify the optimized design. Compare the original and optimized implementations using synthesis reports, timing analysis, power estimation, and FPGA resource utilization. Demonstrate the optimized architecture on an FPGA development board.

8. An open-source FPGA implementation of a DSP accelerator (e.g., Moving Average Filter, FIR Filter, FFT Processor, or Image Processing Filter) is provided through a GitHub repository. Study the repository structure, configure the FPGA project, simulate the RTL, synthesize the design, and implement it on an FPGA development board.

   Perform hardware validation using Integrated Logic Analyzer (ILA) or SignalTap Logic Analyzer to observe internal signals in real time, verify functional correctness, identify timing or synchronization issues, and optimize the implementation based on hardware measurements.

   Prepare a technical report describing the complete FPGA design flow from repository setup to hardware validation.

Course Code	26EE51TH1103			
Category	PCC			
Course Title	Semiconductor Devices			
Scheme & Credits	L	P	Credits	Semester
	4	0	4	I

Course Outcomes

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

CO1: **Analyze**: Analyze semiconductor material properties, carrier transport phenomena, and recombination mechanisms governing semiconductor device operation

CO2: **Evaluate:** Evaluate the electrical characteristics and operational behavior of p-n junctions and metal-semiconductor junctions under various biasing conditions.

CO3: **Analyze:** Analyze the operation of MOS capacitors and MOSFETs using charge control concepts, device physics, and current-voltage characteristics.

CO4: **Evaluate:** Assess the impact of short-channel effects and apply compact SPICE/BSIM device models for CMOS device analysis and simulation

CO5: **Evaluate:** Evaluate advanced CMOS technologies including SOI, High-k Metal Gate, FinFET, and multi-gate MOSFETs for nanoscale VLSI applications

Syllabus:

Module I: Basic Semiconductor Physics (10 Hours)

Fundamental concepts of semiconductor materials and device physics. Crystal lattice structures, semiconductor crystal properties, energy band theory, density of states, and carrier statistics using Maxwell-Boltzmann and Fermi-Dirac distribution functions. Intrinsic and extrinsic semiconductors, doping mechanisms, carrier concentration, and Fermi level movement. Various carrier transport mechanisms including drift, diffusion, thermionic emission, and quantum tunneling with current continuity equations. Excess carriers, carrier generation and recombination processes, carrier lifetime, Shockley-Read-Hall (SRH) recombination, Auger recombination, and their significance in semiconductor devices.

Module II: p-n Junctions and Metal-Semiconductor Junctions (09 Hours)

Physics, fabrication, and operation of p-n junctions and metal-semiconductor contacts including fabrication techniques for p-n junctions, equilibrium conditions, forward and reverse bias operation, depletion region analysis, current-voltage characteristics, charge control model, transient response, junction capacitances, and reverse breakdown mechanisms including Zener and avalanche breakdown. SPICE modeling of p-n junction diode, metal-semiconductor junction fabrication, Schottky barrier formation, rectifying and ohmic contacts, work function concepts, barrier height, and current transport mechanisms with of I-V characteristics.

Module III: MOS Capacitors and MOSFET Fundamentals (09 Hours)

Structure, fabrication, and operation of MOS capacitors and MOSFETs. The MOS capacitor through energy band diagrams, surface potential, accumulation, depletion, inversion, threshold voltage derivation, flat-band voltage, oxide charges, and low-frequency and high-frequency capacitance-voltage (C-V) characteristics. MOSFET fabrication technology and device operation using the gradual channel approximation, simple charge control model (SCCM), Pao-Sah model, and Schichman-Hodges model. Current-voltage characteristics under different operating regions and analysis of device behavior in integrated circuits.

Module IV: MOSFET Scaling, Short Channel Effects and Compact Models (09 Hours)

The non-ideal behavior of nanoscale MOSFETs and compact device modeling. Velocity saturation, mobility degradation, channel length modulation, drain-induced barrier lowering (DIBL), punch-through, charge sharing, hot carrier effects, gate oxide tunneling, leakage currents, and subthreshold conduction. Introduction of Advanced charge control using the Unified Charge Control Model (UCCM). Compact transistor models including SPICE Level-1, Level-2, Level-3, and Berkeley Short-Channel IGFET Model (BSIM), emphasizing their applications in modern CMOS circuit simulation and design.

Module V: Advanced MOSFET Technologies (08 Hours)

Advanced CMOS device technologies used in deep submicron and nanometer VLSI fabrication. Silicon-on-Insulator (SOI) MOSFETs, partially and fully depleted SOI devices, High-k dielectric and Metal Gate (HKMG) technology. FinFETs, tri-gate devices, multi-gate MOSFETs, gate-all-around (GAA) transistor concepts, scaling

challenges, leakage reduction, and performance enhancement techniques. Comparative analysis of planar MOSFETs and advanced multi-gate transistor architectures.

Text Books

1. Ben G. Streetman and Sanjay Banerjee, *Solid State Electronic Devices*, 8th Edition, Pearson.
2. Donald A. Neamen, *Semiconductor Physics and Devices: Basic Principles*, 4th Edition, McGraw-Hill.
3. Yuan Taur and Tak H. Ning, *Fundamentals of Modern VLSI Devices*, 2nd Edition, Cambridge University Press.
4. S. M. Sze and Kwok K. Ng, *Physics of Semiconductor Devices*, 3rd Edition, Wiley.

Reference Books

1. Robert F. Pierret, *Semiconductor Device Fundamentals*, Pearson.
2. Chenming Hu, *Modern Semiconductor Devices for Integrated Circuits*, Pearson.
3. Massoud Pedram and Jan Rabaey (Editors), *Leakage in Nanometer CMOS Technologies*, Springer.
4. Stephen A. Campbell, *The Science and Engineering of Microelectronic Fabrication*, Oxford University Press.
5. Yannis Tsividis and Colin McAndrew, *Operation and Modeling of the MOS Transistor*, 3rd Edition, Oxford University Press.
6. Dimitrijev Sima, *Principles of Semiconductor Devices*, Oxford University Press.

Course Code	26EE51PR1103			
Category	PCC			
Course Title	Semiconductor Devices Lab			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Experiment List:

1. Simulation of Semiconductor Material Properties and Energy Band Structure: Simulate intrinsic and extrinsic silicon by varying doping concentration. Analyze energy band diagrams, carrier concentration, Fermi level position, and conductivity under different temperatures and doping levels. (CO1)
2. Simulation of Carrier Transport Mechanisms in Semiconductors: Investigate drift and diffusion transport mechanisms under applied electric fields. (CO1)
3. TCAD Simulation and Characterization of a Silicon p–n Junction Diode: Create a p–n junction structure, simulate forward and reverse bias characteristics, determine depletion width, electric field distribution, junction capacitance, and reverse breakdown behavior. (CO2)
4. Simulation and Analysis of Schottky Barrier and Ohmic Contacts: Develop metal-semiconductor junctions with different metal work functions. Analyze Schottky barrier height, rectifying and ohmic contacts, and compare their I–V characteristics. (CO2)
5. TCAD Analysis of MOS Capacitor Characteristics: Simulate a MOS capacitor and study accumulation, depletion, and inversion regions. Extract threshold voltage, flat-band voltage, oxide capacitance, and low/high-frequency C–V characteristics. (CO3)
6. Design and Characterization of Planar MOSFET Using TCAD: Design an NMOS transistor, simulate transfer (Id–Vgs) and output (Id–Vds) characteristics, and extract threshold voltage, transconductance, mobility, and ON/OFF current ratio. (CO3)
7. Investigation of Short-Channel Effects in Nanoscale MOSFETs: Study the impact of channel length scaling on threshold voltage roll-off, Drain-Induced Barrier Lowering (DIBL), velocity saturation, hot-carrier effects, and subthreshold leakage. (CO4)
8. Simulation and Performance Evaluation of Advanced MOSFET Structures: Simulate and compare Planar MOSFET, SOI MOSFET, and FinFET devices. Evaluate ON/OFF current ratio, subthreshold swing, leakage current, and electrostatic control. (CO5)
9. Comparative Study of High-k Gate Dielectric MOSFET and Conventional SiO_2 MOSFET: Replace the conventional SiO_2 gate oxide with High-k dielectric materials (e.g., HfO_2). Compare threshold voltage, gate leakage current, oxide capacitance, and overall device performance. (CO5)
10. TCAD Process Simulation and Fabrication Flow of a Planar CMOS/MOSFET Device: Simulate the complete fabrication sequence of a planar MOSFET using TCAD Process Simulator. Perform substrate preparation, oxidation, photolithography, ion implantation, diffusion/annealing, gate oxide formation, polysilicon gate deposition, source/drain formation, spacer formation, and metallization. Generate the final device structure and verify critical process parameters such as junction depth, oxide thickness, channel length, and doping profiles before exporting the structure for electrical characterization. (CO3)

Course Code	26EE51TH1104			
Category	PCC			
Course Title	Embedded System and RTOS			
Scheme & Credits	L	P	Credits	Semester
	4	0	4	I

Course Outcomes:

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

CO1: Apply ARM Cortex-M architecture, programming concepts, and CMSIS to develop and debug applications for ARM Cortex-M based embedded systems.

CO2: Design and develop embedded applications by interfacing peripherals and integrating hardware and software components.

CO3: Analyze real-time embedded systems using scheduling algorithms, synchronization mechanisms, inter-task communication, and resource management techniques to satisfy deterministic system requirements.

CO4: Design and implement multitasking embedded applications using RTOS task management, synchronization, communication, memory management, and interrupt handling mechanisms.

CO5: Evaluate the performance of real-time embedded applications based on timing constraints, scheduling behavior, processor utilization, and memory utilization.

Syllabus

Module I: Embedded Systems Fundamentals (08 Hours)

Introduction to Embedded Systems, characteristics, classifications, design methodology, and design challenges. Review of computer architecture, RISC principles, instruction pipelining, memory organization, startup sequence, linker script basics, embedded software development flow, and introduction to the development environment.

Module II: ARM Cortex-M4 Architecture and Programming (08 Hours)

ARM Cortex-M processor family and Cortex-M4 architecture. Programming model, register organization, stack operation, operation modes, exception model, interrupt handling, Nested Vectored Interrupt Controller (NVIC), SysTick timer, bus architecture, memory map, Memory Protection Unit (MPU), Cortex Microcontroller Software Interface Standard (CMSIS), ARM assembly language fundamentals, debugging, breakpoints, and watchpoints.

Module III: Peripheral Programming and Hardware Abstraction (08 Hours)

Hardware Abstraction Layer (HAL), CMSIS drivers, General Purpose Input/Output (GPIO), external interrupts, timers, Pulse Width Modulation (PWM), Analog-to-Digital Converter (ADC), Digital-to-Analog Converter (DAC) overview, Universal Asynchronous Receiver Transmitter (UART), Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I^2C), Direct Memory Access (DMA), watchdog timer, Real-Time Clock (RTC), clock tree configuration, PLL configuration, and low-power operating modes.

Module IV: Real-Time Operating System Fundamentals (08 Hours)

Introduction to Real-Time Operating Systems (RTOS), characteristics and classifications, process and task models, task states, context switching, scheduling algorithms, Rate Monotonic Scheduling (RMS), Earliest Deadline First (EDF) overview, interrupt service routines, critical sections, inter-task communication, synchronization mechanisms, shared resources, priority inversion, priority inheritance, deadlock, starvation, response time, latency, jitter, and Worst-Case Execution Time (WCET).

Module V: FreeRTOS Architecture and Application Development (08 Hours)

FreeRTOS kernel architecture, task creation and management, Task Control Blocks (TCBs), scheduler operation, queues, binary and counting semaphores, mutexes, recursive mutexes, event groups, direct task notifications, stream buffers, message buffers, software timers, memory management schemes, tickless idle mode, interrupt handling in FreeRTOS, and development of multitasking embedded applications using FreeRTOS on ARM Cortex-M processors.

Module VI: Embedded Linux and Modern Embedded Systems (06 Hours)

Embedded Linux architecture, boot process, bootloader overview, Device Tree, cross-compilation toolchain, Linux kernel configuration and build process, root file system, BusyBox, introduction to embedded Linux device drivers, embedded software development workflow, version control using Git, and an overview of secure boot and firmware update mechanisms.

Text Books

1. Joseph Yiu, The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors, 3rd Edition, Elsevier, 2013.
2. Yifeng Zhu, Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C, 4th Edition, 2023.
3. Richard Barry, Mastering the FreeRTOS™ Real Time Kernel (Latest LTS Edition).
4. Christopher Hallinan, Embedded Linux Primer, 2nd Edition, Prentice Hall, 2010

Reference Books

1. Cem Ünsalan, Hüseyin Deniz Gürhan, Mehmet Erkin Yücel, Embedded System Design with ARM Cortex-M Microcontrollers, Springer, 2022.
2. Jonathan W. Valvano, Embedded Systems: Real-Time Operating Systems for ARM Cortex-M Microcontrollers, 4th Edition, 2017.
3. Andrew N. Sloss, Dominic Symes, Chris Wright, ARM System Developer's Guide, Morgan Kaufmann, 2004.
4. Tammy Noergaard, Embedded Systems Architecture, 2nd Edition, Newnes / Elsevier 2013
5. Chris Simmonds, Mastering Embedded Linux Programming, 2nd Edition, Packt Publishing, 2017

Course Code	26EE51PR1104			
Category	PCC			
Course Title	Embedded System and RTOS Lab			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Indicative List of Experiments (The instructor may modify/add experiments based on the availability of EDA tools and emerging technologies.)

Experiment No. 1: Development environment setup using STM32CubeIDE and STM32CubeMX; familiarization with the STM32F407G-DISC1 Discovery Kit and project creation.

Experiment No. 2: GPIO programming using register-level programming and HAL for LED control, push-button interfacing, and external interrupt handling.

Experiment No. 3: Timer and SysTick programming for precise delay generation, periodic interrupts, and event scheduling.

Experiment No. 4: PWM generation and applications for LED brightness control and servo/DC motor interfacing.

Experiment No. 5: Analog-to-Digital Converter (ADC) interfacing for sensor data acquisition and monitoring.

Experiment No. 6: UART communication for serial data transmission, debugging, and command-line interface implementation.

Experiment No. 7: SPI and I²C interfacing with external peripherals such as EEPROM, sensors, or display modules.

Experiment No. 8: Direct Memory Access (DMA) based peripheral data transfer and comparative performance analysis with CPU-driven transfer.

Experiment No. 9: Nested Vectored Interrupt Controller (NVIC) programming and interrupt priority management using CMSIS.

Experiment No. 10: FreeRTOS task creation, scheduling, and context switching for multitasking embedded applications.

Experiment No. 11: Inter-task communication using FreeRTOS queues, semaphores, mutexes, event groups, and direct task notifications.

Experiment No. 12: FreeRTOS software timers, memory management schemes, and CPU utilization analysis.

Experiment No. 13: Performance evaluation of real-time embedded applications by measuring execution time, latency, processor utilization, and memory footprint.

Experiment No. 14: Cross-compilation and execution of a simple Embedded Linux application using a Linux-based development environment.

Experiment No. 15: Mini Project – Design, implement, and evaluate a multitasking real-time embedded application integrating multiple peripherals and FreeRTOS services on the STM32F407G-DISC1 Discovery Kit.

Course Code	26EE51TH1105			
Category	PCC			
Course Title	Design for Testability			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	I

Course Outcomes:

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

1. Identify and model different types of faults in VLSI designs
2. Apply the Testability measures for Combinational & Sequential Circuits
3. Apply various algorithms for test pattern generation and simulation
4. Model different types of memory faults and develop effective testing strategies to detect them
5. Use techniques such as built-in self-test (BIST), boundary scan testing, and fault injection to improve testability

Syllabus

Module-1: (05 Hrs.)
Scope of testing: Fundamentals of VLSI testing, verification in VLSI design process. Issues in test and verification of complex chips, embedded cores and SOCs. VLSI Technology Trends Affecting Testing.

Module-2: (08 Hrs.)
Fault modeling and Simulation: Fault Equivalence, Fault Collapsing, Fault Dominance, Algorithms for Fault Simulation, Serial Fault Simulation, Parallel Fault Simulation, Deductive Fault Simulation, Concurrent Fault Simulation, IDDQ testing, Delay testing

Module-3: (08 Hrs.)
Testability measures and Test pattern generation: controllability and observability SCOAP, D-Algorithm, PODEM Algorithm,

Module-4: (06 Hrs.)
Memory Testing: Memory fault model, stuck at fault, Transition Faulty, Coupling Fault, In version Coupling Fault, Idempotent Coupling Faults. Address Decoder Faults. Neighborhood Pattern Sensitive fault, Memory testing Algorithms.

Module-5: (06 Hrs.)
Design for testability: Tradeoffs, Adhoc Design for Testability Techniques, Scan design. LSSD, Test interface and boundary scan.

Module-6: (07 Hrs.)
Built in Self-Test (BIST): Test Pattern Generation for BIST, BIST Architectures, MCM testing, Yield models.

TextBooks :

1. Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits: M. Bushnell and V. D. Agrawal, Kluwer Academic Publishers, 2000.

2. Digital Systems Testing and Testable Design: M. Abramovici, M. A. Breuer and A. D. Friedman, IEEE Press, 1990.

Reference Books:

1. Introduction to Formal Hardware Verification: T. Kropf, Springer Verlag, 2000.
2. System-on-a-Chip Verification-Methodology and Techniques: P. Rashinkar, Paterson and L. Singh, Kluwer Academic Publishers, 2001.

Course Code	26EE51PR1106			
Category	SEC			
Course Title	Laboratory Practice I			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Outcomes

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

CO1. Utilize Linux operating system utilities and command-line tools for engineering and software development workflows.

CO2. Develop Bash shell scripts to automate routine engineering tasks and software workflows.

CO3. Apply version control, build automation, and debugging tools for collaborative software development.

CO4. Develop Linux-based engineering workflows for improving productivity and software development.

Indicative List of Experiments (The instructor may modify/add experiments based on the availability of EDA tools and emerging technologies.)

Experiment No. 1

Linux environment, file system organization, file and directory management.

Experiment No. 2

Linux command-line utilities: grep find sed awk cut sort uniq tar gzip

Experiment No. 3

User management, file permissions, process management and job control.

Experiment No. 4

Bash shell scripting: Variables, Operators, Loops, Conditional statements, Functions.

Experiment No. 5

Advanced Bash scripting: File processing, Batch execution, Log analysis, and Automation

Experiment No. 6

Version control using Git: Repository creation, Commit, Branch, Merge, Conflict resolution

Experiment No. 7

Build automation using Makefiles.

Experiment No. 8

GNU software development tools: GCC , GDB, objdump, readelf

Experiment No. 9

Remote Linux development: SSH, SCP, Secure file transfer

Experiment No. 10

Engineering workflow automation using Linux utilities and shell scripting.

Experiment No. 11

Mini Project:
Develop a Linux-based automated engineering workflow integrating shell scripting, Git, and build automation.

SYLLABUS OF SEMESTER II, M. Tech.(VLSI DESIGN)

Course Code	26EE51TH1201			
Category	PCC			
Course Title	Analog & Mixed-Signal IC Design			
Scheme & Credits	L	P	Credits	Semester
	4	0	4	II

Course Outcomes:

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

1. **Apply** mathematical models of MOS transistors to evaluate their behavior in deep sub-micron analog circuits.
2. **Analyze** and design fundamental analog building blocks, including current mirrors, references, and differential amplifiers.
3. **Design** stable single-stage and two-stage Operational Amplifiers by analyzing frequency response and compensation techniques.
4. **Evaluate** mixed-signal architectures, including Switched-Capacitor circuits and high-performance Data Converters (ADC/DAC).

Syllabus

Analog Design Foundations & Sub-micron Challenges: Introduction to analog VLSI and design issues in modern CMOS technologies. Short-channel effects in analog design, device mismatch, and layout techniques for analog circuits (common-centroid, interdigitation).

Basic Analog Building Blocks: Switches, active resistors, current sources and sinks. Advanced current mirrors (Cascode, Wilson). Voltage references, temperature-independent references, and Bandgap Reference (BGR) design.

Single-Stage & Differential Amplifiers: Common Source, Source Follower, Common Gate, and Cascode amplifiers. Frequency Response: Miller Effect, pole-zero association. Differential Amplifiers: Basic differential pair, common-mode response, CMRR, MOS load differential pairs, and the Gilbert Cell.

Operational Amplifier Design: Design of single-stage and two-stage OPAMPs. Gain boosting techniques. Frequency compensation, phase margin, slew rate, and power supply rejection ratio (PSRR).

Switched-Capacitor Circuits: General considerations for mixed-signal interfaces. Sampling switches, charge injection, and clock feedthrough. Switched-capacitor integrators and discrete-time filters.

Data Converter Fundamentals: DAC/ADC Specifications (INL, DNL, SNR, ENOB). DAC Architectures: Resistor String, Charge-Scaling, Cyclic, and Pipeline DACs. ADC Architectures: Flash, Two-Step Flash, Pipeline, Successive Approximation Register (SAR) ADC, and introduction to Sigma-Delta oversampling ADCs.

Text books:

1. Design of Analog CMOS Integrated Circuits: Behzad Razavi, McGraw-Hill, (2/E) 2016
2. CMOS Circuit Design, Layout, and Simulation: R. Jacob Baker, IEEE Press/Wiley, (4/E) 2019

3. Analog Integrated Circuit Design: Tony Chan Carusone, David Johns, Kenneth Martin, Wiley, (2/E) 2011

Reference books:

1. VLSI Design Techniques for Analog and Digital Circuits: Randall L. Geiger, Phillip E. Allen, Noel R. Strader, McGraw-Hill, (1/E) 2010
2. Analysis and Design of Analog Integrated Circuits: Paul R. Gray, Paul J. Hurst, Stephen H. Lewis, Robert G. Meyer, Wiley, (5/E) 2009

Course Code	26EE51PR1201			
Category	PCC			
Course Title	Analog & Mixed-Signal IC Design Lab			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	II

Indicative List of Experiments (The instructor may modify/add experiments based on the availability of EDA tools and emerging technologies.)

Experiment No. 1
Device Characterization

Familiarization with PDK , NMOS/PMOS characterization, Id-Vgs, Id-Vds ,Threshold voltage extraction, gm analysis.

Experiment No. 2
CMOS Current Mirrors
Design and simulate: Basic Current Mirror, Cascode Current Mirror, and Widlar Current Mirror
Evaluate current accuracy and output resistance.

Experiment No. 3
Single-Stage CMOS Amplifier
Design a Common Source Amplifier.
Measure: Voltage Gain, Bandwidth, Input/Output Swing, and Power Consumption

Experiment No. 4
Differential Amplifier
Design a CMOS Differential Pair.
Evaluate: Differential Gain, Common Mode Gain, CMRR, and Offset

Experiment No. 5
Operational Amplifier Design
Design a Two-Stage CMOS Operational Amplifier.
Analyze: DC Gain, Unity Gain Bandwidth, Phase Margin, Slew Rate, and Power Dissipation

Experiment No. 6
Frequency Compensation
Implement Miller Compensation.
Study: Stability, Phase Margin, Pole-Zero movement.

Experiment No. 7
Comparator Design
Design: Static Comparator and Dynamic Latch Comparator
Evaluate: Offset, Delay, and Power

Experiment No. 8
Bandgap Reference / Voltage Reference
Design and simulate a CMOS voltage reference circuit.
Evaluate: Temperature stability, Supply sensitivity, and Output variation

Experiment No. 9
CMOS Ring Oscillator
Design

- 3-stage
- 5-stage Ring Oscillator

Measure: Oscillation Frequency, Phase Noise, and Power Consumption

Experiment No. 10
Mixed-Signal Building Block
Design one of the following: Sample-and-Hold Circuit OR CMOS Transmission Gate Switch OR Track-and-Hold Circuit.
Evaluate switching performance.

Experiment No. 11
Layout Design and Physical Verification
Create layout for Differential Pair OR Current Mirror
Perform: DRC and LVS

Experiment No. 12
Post-Layout Simulation
Extract parasitics (PEX).
Compare: Pre-layout and Post-layout
Analyze: Gain degradation, Bandwidth, and Delay

Mini Design Project

Design, layout, and verify one analog building block such as an Operational Amplifier, Comparator, Voltage Reference, Ring Oscillator, or Phase-Locked Loop (PLL) sub-block to meet specified design requirements.

Course Code	26EE51TH1202			
Category	PCC			
Course Title	System Verilog & UVM for Verification			
Scheme & Credits	L	P	Credits	Semester
	4	0	4	II

Course Outcomes (COs)

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

CO1: Apply Object-Oriented Programming (OOP) concepts of SystemVerilog to develop reusable and scalable verification components.

CO2: Design constrained-random verification environments using SystemVerilog inter-process communication mechanisms and randomization techniques.

CO3: Develop assertion-based verification environments and perform functional coverage analysis to validate digital hardware designs.

CO4: Architect and implement industry-standard verification environments using the Universal Verification Methodology (UVM) for functional verification of digital systems.

Course Syllabus

Module I: SystemVerilog Verification Foundations

Introduction to Hardware Verification. Verification Guidelines, Verification Process and Verification Planning. Verification Methodologies. Simulation-based Verification Flow. Basic Testbench Architecture. SystemVerilog Data Types: Built-in data types, literals, arrays (Fixed, Dynamic, Associative), Queues, Structures, Unions, Enumerated Types, Typedef, Packages and Compilation Units. Procedural Blocks, Procedural Statements, Operators, Tasks and Functions.

Module II: Object-Oriented Programming in SystemVerilog

Object-Oriented Programming Concepts. Classes, Objects, Handles, Constructors and Methods. Data Hiding and Encapsulation. Inheritance, Polymorphism and Virtual Methods. Static Members, Abstract Classes, Virtual Classes and Parameterized Classes. Deep Copy and Shallow Copy. Class-based Verification Environment Development. Reusable Verification Components.

Module III: Constrained Random Verification

Threads and Process Control. Inter-Process Communication (IPC): Events, Mailboxes and Semaphores. Directed Testing versus Constrained-Random Verification. Random Variables and Randomization Methods. Constraint Blocks, Constraint Inheritance, Constraint Override, Inline Constraints, State-dependent Constraints, Weighted Distribution and Random Stability. Development of Reusable Constrained-Random Testbenches.

Module IV: Assertion-Based Verification and Functional Coverage

Assertion-Based Verification Concepts. Immediate Assertions and Concurrent Assertions. Sequences, Properties, Implication Operators, Repetition Operators, Assertion Control and Disable Conditions. Assertion-Based Protocol Checking. Functional Coverage Fundamentals. Covergroups, Coverpoints, Bins, Automatic Bins, Transition Bins, Ignore Bins, Illegal Bins, Cross Coverage, Coverage Collection and Coverage Closure.

Module V: Universal Verification Methodology (UVM)

Introduction to Universal Verification Methodology (UVM). UVM Architecture and Verification Flow. UVM Base Classes (uvm_object, uvm_component). UVM Phases and Objection Mechanism. UVM Factory and Factory Overrides. Configuration Database (uvm_config_db). Sequence Items, Sequences and Sequencers. Drivers, Monitors, Agents, Environments and Scoreboards. Virtual Interfaces and Virtual Sequences. Building Reusable UVM Verification Environments. Introduction to Regression Execution and Python-based EDA Automation for Simulation, Log Parsing and Regression Testing.

Text Books

1. Chris Spear and Greg Tumbush, SystemVerilog for Verification: A Guide to Learning the Testbench Language Features, 3rd Edition, Springer, 2012.
2. Accellera Systems Initiative, Universal Verification Methodology (UVM) 1.2 User's Guide, 2015.
3. Kathleen Meade and Sharon Rosenberg, A Practical Guide to Adopting the Universal Verification Methodology (UVM), 2nd Edition, Cadence Design Systems, 2013.

Reference Books

1. Janick Bergeron, Writing Testbenches using SystemVerilog, Springer, 2006.
2. Harry D. Foster, Adam C. Krolnik and David J. Lacey, Assertion-Based Design, 2nd Edition, Springer, 2004.
3. Stuart Sutherland, Simon Davidmann and Peter Flake, SystemVerilog for Design, 2nd Edition, Springer, 2006.
4. Donald Norris, Python for Microcontrollers: Getting Started with MicroPython, McGraw-Hill, 2016 (for EDA scripting and automation).

Course Code	26EE51PR1202			
Category	PCC			
Course Title	System Verilog & UVM for Verification Lab			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	II

Experiment List:

1. Development of a Basic SystemVerilog Verification Testbench: Create a verification environment for a 4-bit ALU using interfaces, packages, tasks, functions, and testbench modules.
2. Implementation of Object-Oriented Programming in SystemVerilog: Develop reusable verification components using classes, inheritance, polymorphism, virtual methods, and parameterized classes to verify a multiplexer or counter.
3. Design of Constrained-Random Verification Environment: Generate randomized stimulus with constraint blocks, inline constraints, constraint override, and randomization methods to verify an Arithmetic Logic Unit (ALU).
4. Implementation of Inter-Process Communication (IPC): Verify a FIFO using Events, Mailboxes, Semaphores, and multiple concurrent processes for synchronized transaction-level communication.
5. Assertion-Based Verification of Sequential Circuits: Develop Immediate and Concurrent SystemVerilog Assertions (SVA) to verify protocol timing and functionality of a FIFO or UART controller.
6. Functional Coverage Analysis: Create Covergroups, Coverpoints, Transition Bins, Cross Coverage, Ignore Bins, and Illegal Bins to evaluate verification completeness of a digital controller.
7. Development of a Basic UVM Verification Environment: Build a UVM testbench including sequence item, sequence, sequencer, driver, monitor, and agent for verification of an ALU.
8. Construction of a Reusable UVM Testbench: Develop a reusable UVM environment consisting of Environment, Scoreboard, Subscriber, Functional Coverage Collector, and Virtual Interface for verification of a synchronous FIFO
9. Implementation of Advanced UVM Features: Apply UVM Factory Override, Configuration Database (uvm_config_db), Virtual Sequences, Callbacks, and Objection Mechanism to build configurable verification environments.
10. Automation of UVM Regression and Coverage Analysis: Develop Python scripts to automate simulation execution, regression testing, log parsing, waveform management, and functional coverage report generation.

Course Code	26EE51TH1203			
Category	PCC			
Course Title	Digital ASIC Design			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	II

Course Outcomes (COs):

On successful completion of the course, students will be able to:

CO1: Analyze the logic synthesis process, Clock Domain Crossing (CDC) methodologies, and the role of timing constraints (SDC) in generating an optimized netlist.

CO2: Evaluate macro floorplanning strategies, power grid distribution planning, and standard cell placement algorithms for congestion and area optimization.

CO3: Examine the algorithms behind Clock Tree Synthesis (CTS) and global/detailed routing, and analyze their impact on Signal Integrity (SI) and electromigration.

CO4: Apply Static Timing Analysis (STA) principles to identify setup and hold violations, and comprehend the physical sign-off checks (DRC, LVS) required for final GDSII generation.

Course Syllabus

Module 1: Synthesis, CDC, and Constraints [7 Hrs]

Translating logic to gate-level netlists. Clock Domain Crossing (CDC) analysis, multi-flop synchronizers, and asynchronous FIFOs. Writing Synopsys Design Constraints (SDC) for clocks, false paths, and I/O delays.

Module 2: Floorplanning and Power Planning [6 Hrs]

Core area estimation, macro placement heuristics, and routing blockages. Designing the Power Grid/Rings. Multi-VDD domains, power-gating, and interpreting Unified Power Format (UPF).

Module 3: Placement and Clock Tree Synthesis (CTS) [8 Hrs]

Placement algorithms (simulated annealing, min-cut). Congestion-driven placement and tie-cell insertion. Goals of CTS: minimizing skew and latency. Buffer insertion, delay balancing, and post-CTS hold-time fixing.

Module 4: Signal Routing and Signal Integrity (SI) [7 Hrs]

Global vs. detailed routing algorithms (Maze routing). Handling Design Rule Constraints (spacing, via rules). Mitigating SI issues in layout: crosstalk delay, crosstalk noise (glitch), and Electromigration (EM) fixing.

Module 5: Static Timing Analysis (STA) and Sign-Off [7 Hrs]

Exhaustive setup and hold timing closure using STA engines across On-Chip Variation (OCV) and PVT corners. Parasitic extraction (SPEF). Physical verification for tape-out: Design Rule Checking (DRC) and

Layout Versus Schematic (LVS), concluding with final GDSII / OASIS database generation for foundry manufacturing.

Reference Books:

1. VLSI Physical Design: From Graph Partitioning to Timing Closure (2nd Edition) by Andrew B. Kahng, Jens Lienig, Igor L. Markov, & Jin Hu, Springer, 2022.

2. Advanced ASIC Chip Synthesis: Using Synopsys Design Compiler, Physical Compiler, and Primetime (2nd Edition) by Himanshu Bhatnagar, Springer, 2002.

3. Static Timing Analysis for Nanometer Designs: A Practical Approach by J. Bhasker & Rakesh Chadha, Springer, 2009.

4. Physical Design Essentials: An ASIC Design Implementation Perspective by Khosrow Golshan, Springer, 2007.

5. Constraining Designs for Synthesis and Timing Analysis: A Practical Guide to Synopsys Design Constraints (SDC) by Sridhar Gangadharan & Kattamuri Ekanath Seth, Springer, 2013

Course Code	26EE51PR1203			
Category	PCC			
Course Title	Digital ASIC Design Lab			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	II

Indicative List of Experiments (The instructor may modify/add experiments based on the availability of EDA tools and emerging technologies.)

Experiment No. 1: RTL Synthesis and Gate-Level Netlist Generation

a) Synthesize a given RTL design using an industry-standard synthesis tool.
b) Examine the generated gate-level netlist.
c) Analyze area, timing and cell utilization reports.
d) Identify combinational and sequential cells in the synthesized design.

Experiment No. 2: Clock Domain Crossing Analysis

a) Analyze a design containing multiple clock domains.
b) Identify CDC violations.
c) Implement and evaluate:
    a. Multi-flop synchronizers
    b. Pulse synchronization
    c. Asynchronous FIFO
d) Compare safe and unsafe CDC implementations.

Experiment No. 3: Timing Constraints using SDC

a) Define clocks and generated clocks.
b) Apply:
    a. Input/output delays
    b. Clock uncertainty
    c. False paths
    d. Multicycle paths
c) Perform timing analysis before and after applying constraints.
d) Examine the effect of constraints on synthesis and timing reports.

Experiment No. 4: Floorplanning and Macro Placement

a) Define the core and die area.
b) Determine target utilization and aspect ratio.
c) Place macros and memories.
d) Define placement blockages and routing channels.
e) Analyze:

    - Area utilization
    - Macro placement
    - Congestion
    - Timing implications

Experiment No. 5: Power Planning and Power Distribution Network

a) Create power rings and power straps.

b) Connect standard cells and macros to the power network.
c) Examine power-grid connectivity.
d) Analyze IR-drop considerations.
e) Introduce multiple voltage domains at a conceptual or tool-supported level.

Experiment No. 6: Standard Cell Placement and Congestion Analysis

a) Perform global and detailed placement.
b) Analyze placement density and congestion.
c) Study the effect of:
    a. Cell utilization
    b. Placement blockages
    c. Macro locations
    d. High-fanout nets
d) Optimize placement for area and congestion.

Experiment No. 7: Clock Tree Synthesis and Clock Optimization

a) Perform Clock Tree Synthesis (CTS).
b) Analyze:
    a. Clock skew
    b. Clock latency
    c. Clock transition
    d. Clock fanout
c) Perform buffer insertion and clock balancing.
d) Analyze post-CTS timing.
e) Identify and fix hold violations introduced after CTS.

Experiment No. 8: Global and Detailed Routing

a) Perform global routing followed by detailed routing.
b) Examine routing congestion and overflow.
c) Study routing constraints involving:
    a. Metal layers
    b. Spacing
    c. Via rules
    d. Routing blockages
d) Analyze the final routed design.

Experiment No. 9: Signal Integrity and Crosstalk Analysis

a) Identify critical nets susceptible to crosstalk.
b) Analyze crosstalk-induced:
    a. Delay variation
    b. Noise/glitches
c) Investigate the effect of:
    a. Wire spacing
    b. Buffer insertion
    c. Routing changes
d) Compare the design before and after SI optimization.

Experiment No. 10: Electromigration and Power Integrity Analysis

a) Analyze power-network reliability.

b) Examine current density and potential electromigration violations.
c) Identify vulnerable power/interconnect regions.
d) Apply appropriate fixes such as:
    a. Wider power routes
    b. Additional vias
    c. Improved power distribution
e) Re-run analysis to evaluate the improvement.

Experiment No. 11: Static Timing Analysis and Timing Closure

a) Perform post-route STA.
b) Analyze:
    a. Setup violations
    b. Hold violations
    c. Slack
    d. Critical paths
c) Perform timing analysis across appropriate PVT corners.
d) Introduce OCV/AOCV concepts where supported.
e) Apply timing optimization techniques and verify timing closure.

Experiment No. 12: Physical Verification and Final GDSII Generation

a) Perform parasitic extraction and generate SPEF.
b) Perform post-layout timing analysis.
c) Run:
    a. DRC
    b. LVS
d) Identify and resolve physical verification violations.
e) Generate the final GDSII/OASIS database.

Mini Project: Complete RTL-to-GDSII Implementation

Take a moderate-sized RTL design such as:

a) 16/32-bit RISC-V processor
b) UART controller
c) AES core
d) FIR filter
e) ALU/datapath
f) Small SoC subsystem

and perform: RTL↓ Synthesis↓ SDC Constraints↓ Floorplanning↓Power Planning↓Placement↓CTS↓Routing↓STA↓SI/EM Analysis↓DRC/LVS↓GDSII/OASIS

Give report for: Area, Cell utilization, Worst setup slack, Worst hold slack, Clock skew, Routing congestion, Power IR drop, DRC/LVS status, Final GDSII generation status.

Course Code	26EE51TH1204			
Category	PCC			
Course Title	SoC Design			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	II

Course Outcomes
Upon successful completion of the course, students will be able to:
CO1: Apply systems-level design principles to analyze hardware/software partitioning, programmability, performance, power, and product trade-offs in System-on-Chip (SoC) design.
CO2: Analyze the architecture and microarchitecture of RISC-V processor cores and their memory hierarchy in contemporary SoC platforms.
CO3: Evaluate on-chip communication architectures, including AMBA bus protocols and Network-on-Chip (NoC) interconnection strategies for efficient SoC design.
CO4: Develop and integrate bare-metal firmware and embedded software for FPGA-based RISC-V SoC platforms by interfacing processors with peripherals and custom IP blocks.
CO5: Describe the architectural concepts of Multiprocessor SoCs (MPSoCs), heterogeneous computing, and AI/ML hardware accelerators in modern System-on-Chip architectures.

Syllabus

Module I: System-on-Chip Design Fundamentals (05 Hours)

Introduction to System-on-Chip (SoC) design methodology. System architecture overview. Hardware/software co-design and partitioning. Programmability versus performance. Product economics and design trade-offs. Managing design complexity through IP reuse and platform-based design. Chip design considerations including area, power, performance, reliability, configurability, and design productivity.

Module II: RISC-V Architecture and Processor Microarchitecture (07 Hours)

Overview of the RISC-V ISA (RV32I). Processor architecture and microarchitecture. Pipeline organization and hazard handling. Memory hierarchy in SoCs including cache memories, tightly coupled memories (TCM), and memory management units (MMU). Control and Status Registers (CSR), privilege modes, exception handling, and interrupt mechanisms.

Module III: On-Chip Communication Architectures (06 Hours)

Need for standardized on-chip communication. Bus architectures, arbitration techniques, and memory-mapped communication. Advanced Microcontroller Bus Architecture (AMBA): AHB, APB, and AXI protocols. Introduction to TileLink protocol. Address decoding, bus bridging, and memory-mapped I/O.

Module IV: IP Integration and Embedded Software Development (06 Hours)

Design and integration of reusable IP cores. Standard bus interfacing and peripheral integration including GPIO, Timers, UART, SPI, and I²C. Direct Memory Access (DMA) controllers and interrupt controllers. Bare-metal firmware development for FPGA-based RISC-V SoCs. Hardware-software integration and verification of embedded applications.

Module V: Network-on-Chip (NoC) Architectures (06 Hours)

Introduction to Network-on-Chip (NoC). NoC architectures and topologies. Switching techniques. Routing algorithms. Flow control techniques. Router microarchitecture. Layered architecture and Network Interface Units (NIU). Performance metrics including latency, throughput, bandwidth, and scalability.

Module VI: Emerging SoC Architectures (05 Hours)

Introduction to Multiprocessor Systems-on-Chip (MPSoCs). Performance and flexibility considerations in MPSoC design. Heterogeneous computing architectures. Overview of AI and Machine Learning hardware accelerators in modern SoCs. Emerging trends in System-on-Chip architectures.

Text Books

1. RISC-V System-on-Chip Design, David Harris, James Stine, Sarah Harris, Rose Thompson, Morgan Kaufmann, First Edition, 2026.
2. Digital Design and Computer Architecture: RISC-V Edition, Sarah L. Harris and David Harris, Morgan Kaufmann, 2021.
3. Computers as Components: Principles of Embedded Computing System Design, Wayne Wolf, Morgan Kaufmann, Fourth Edition, 2016

Reference Books

1. Networks-on-Chips: Technology and Tools, Giovanni De Micheli and Luca Benini, Morgan Kaufmann, 2006.
2. Multiprocessor System-on-Chips, Ahmed Jerraya and Wayne Wolf, Morgan Kaufmann, 2004.
3. Computer Organization and Design: RISC-V Edition – The Hardware/Software Interface, David A. Patterson and John L. Hennessy, Morgan Kaufmann, 2020.
4. System-on-Chip Design with Arm Cortex-M Processors, Joseph Yiu, Arm Education Media, 2019.
5. Selected research papers from IEEE Transactions on VLSI Systems (TVLSI), IEEE Transactions on Computer-Aided Design (TCAD), and premier conferences such as DAC, ISCA, MICRO, DATE, and ASP-DAC.

Course Code	26EE51PR1204			
Category	PCC			
Course Title	SoC Design Lab			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	II

Indicative List of Experiments (The instructor may modify/add experiments based on the availability of EDA tools and emerging technologies.)

Experiment No. 1: RVfpga-SoC Environment Initialization and Baseline Synthesis

Problem Statement:
Configure the RISC-V software toolchain and FPGA development environment. Synthesize the baseline SweRVolf SoC architecture, generate the FPGA bitstream, program the target FPGA board, and verify successful hardware bring-up.

Experiment No. 2: Bare-Metal Memory-Mapped I/O Interfacing

Problem Statement:
Develop a bare-metal C application to interface with memory-mapped peripherals. Read hardware switch inputs and control GPIO peripherals such as LEDs or seven-segment displays through memory-mapped registers.

Experiment No. 3: RISC-V Exception and Interrupt Handling

Problem Statement:
Configure the Core Local Interruptor (CLINT) and Platform-Level Interrupt Controller (PLIC). Develop an interrupt-driven application utilizing timer and external interrupts, and implement efficient Interrupt Service Routines (ISRs) without blocking the main execution flow.

Experiment No. 4: SoC Architecture Exploration using FuseSoC

Problem Statement:
Explore the SweRVolf SoC architecture using FuseSoC. Analyze the hierarchical hardware organization, identify processor, bus interconnects, memory organization, peripheral subsystems, and interpret the system memory map.

Experiment No. 5: Hardware IP Customization and Driver Development

Problem Statement:
Modify the SweRVolf SoC by integrating or customizing a memory-mapped peripheral or hardware IP block. Develop the corresponding embedded software driver and verify correct hardware-software interaction through functional testing.

Experiment No. 6: DMA Controller Configuration and Performance Analysis

Problem Statement:
Configure the Direct Memory Access (DMA) controller for peripheral or memory transfers. Compare DMA-

based data transfer with processor-driven transfer by evaluating execution time, processor utilization, and overall system performance.

Experiment No. 7: Zephyr RTOS Porting and System Boot

Problem Statement:
Configure the device tree and build environment to port the Zephyr Real-Time Operating System to the SweRVolf SoC. Successfully boot the Zephyr kernel and establish serial console communication.

Experiment No. 8: Multi-threaded Embedded Application using Zephyr RTOS

Problem Statement:
Develop a multitasking embedded application using Zephyr RTOS. Implement preemptive thread scheduling and inter-thread communication using synchronization primitives such as mutexes, semaphores, or message queues.

Experiment No. 9: Edge AI Inference on RISC-V SoC

Problem Statement:
Cross-compile and deploy a pre-trained TensorFlow Lite for Microcontrollers (TFLM) model on the SweRVolf SoC. Execute AI inference, evaluate memory footprint, execution latency, and analyze computational performance on the RISC-V processor.

Experiment No. 10: Mini Project – FPGA-Based RISC-V System-on-Chip Design

Problem Statement:
Design and implement a complete FPGA-based RISC-V embedded system by integrating custom hardware IP, memory-mapped peripherals, interrupt handling, bare-metal firmware or Zephyr RTOS, and application software. Validate the complete hardware-software co-designed system through functional and performance evaluation.

Course Code	26EE51PR1205			
Category	SEC			
Course Title	Laboratory Practice II			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	II

Course Outcomes

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

CO1. Develop Python programs for scientific computing, engineering data analysis, and visualization.

CO2. Develop Tcl scripts to automate engineering tasks and Electronic Design Automation (EDA) workflows.

CO3. Prepare professional technical documents and research manuscripts using LaTeX.

CO4. Integrate scripting, data visualization, and technical documentation tools to improve engineering and research productivity.

Indicative List of Experiments (The instructor may modify/add experiments based on the availability of EDA tools and emerging technologies.)

Experiment No. 1:
Python fundamentals for engineering applications.

Experiment No. 2
Scientific computing using NumPy.

Experiment No. 3
Engineering data analysis using Pandas.

Experiment No. 4
Engineering visualization using Matplotlib: Line plots, Scatter plots, Histograms, Bar charts, Publication-quality figures.

Experiment No. 5
Engineering data processing and automation using Python.

Experiment No. 6
Introduction to Tcl programming: Variables, Lists, Procedures, Loops, File handling.

Experiment No. 7

Tcl scripting for engineering automation.

Experiment No. 8
EDA workflow automation using Tcl scripts.

Experiment No. 9
Introduction to LaTeX: Document preparation, Equations, Tables Figures.

Experiment No. 10

Preparation of IEEE-format technical papers using LaTeX.

Experiment No. 11
Preparation of technical reports, dissertations, and presentations using LaTeX.

Experiment No. 12
Mini Project:
Develop an integrated engineering workflow involving Python for data analysis and visualization, Tcl for automation, and LaTeX for preparing a professional technical report.

Course Code	26EE51TH1207-01			
Category	OE			
Course Title	VLSI Signal Processing			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	II

Course Outcomes:
Students will demonstrate the ability to

1. Analyze discrete-time signal processing systems using data flow graphs, dependence graphs and iterative algorithms
2. Design high-throughput DSP architectures using pipelining and parallel processing techniques
3. Optimize DSP architectures using retiming techniques for clock period and register minimization under design constraints.
4. Design area-efficient DSP architectures using folding and unfolding transformations.
5. Analyze and evaluate systolic architectures for high-performance DSP and AI accelerator applications

Syllabus

Module I: Introduction to Digital Signal Processing (DSP) systems, sampling theorem, discrete time signal & systems, representation of discrete systems in Z domain, basic DSP algorithms and its mathematical representation, representation of DSP algorithms using block diagram, data flow graph, dependence graph, signal flow graph, Loop bound and Iteration bound algorithms, difference between recursive(IIR) and non-recursive (FIR) systems

Module II: Basics of pipelining and parallel processing, cut-set theory, data broadcast structure and transpose structure of FIR systems, fine-grain and course-grain pipelining techniques, realizing a parallel architecture for FIR systems, pipelining and parallel processing for low power.

Module III: Retiming algorithm for IIR systems, cut set retiming and pipelining, retiming for clock period minimization, retiming for register minimization.

Module IV: Unfolding algorithms, properties, unfolding for sample period reduction, word level processing, bit level processing, register minimization techniques using folding transformation.

Module V: Folding algorithms, properties, register minimization techniques using folding transformation, Folding equation, Lifetime analysis, Register allocation

Module VI: Systolic architecture design, block diagram, space representation, edge mapping table, low level implementations, space time representation for systolic arrays, Applications of systolic architectures in matrix multiplication, convolution, neural network accelerators and tensor processing units.

Text book:

1. VLSI Digital Signal Processing Systems: Keshab K. Parhi. Wiley-Inter Sciences. (1999).
2. Analog VLSI signal and information processing: Mohammed Ismail, Terri, Fiez, McGraw Hill.(1994).
3. VLSI Digital signal processing system Design and implementation: Keshab. Parthi, Wiley-Interscience, (1999).

Reference Books:

1. VLSI and Modern signal processing: kung. S. Y., H. J. While house T. Kailath,

prentice hall,(1985).

2. Design of Analog - Digital VLSI circuits for telecommunications and signal processing: Jose E.France, YannisTsividls, Prentice Hall, (1994).

Course Code	26EE51TH1207-02			
Category	OE			
Course Title	High Level Synthesis			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	II

Course Outcomes

On successful completion of the course, students will be able to:

1. Understand the High-Level Synthesis (HLS) process and its role in converting C-code to hardware.
2. Apply scheduling, allocation, and binding techniques for efficient hardware synthesis.
3. Optimize C-code for improved hardware performance using loop transformations and dataflow optimizations.
4. Implement verification techniques to ensure correctness and equivalence between C and RTL.
5. Explore advanced topics like HLS for security, FPGA technology mapping, and recent trends in VLSI design.

Module I:

Introduction to High-Level Synthesis (HLS) (5 Hours)

Introduction to High-Level Synthesis (HLS), fundamentals of HLS design flow, scheduling, resource allocation, binding, and controller synthesis, introduction to scheduling techniques, ILP formulation for MLRC and MRLC scheduling, introduction to multiprocessor scheduling and its algorithms, list-based scheduling for MLRC and MRLC, Forced Directed Scheduling, Forced Directed MLRC and MRLC Scheduling Algorithm, Path-Based Scheduling techniques, role of constraints in HLS, and performance trade-offs in scheduling.

Module II:

Allocation and Binding Techniques in HLS (5 Hours)

Understanding the allocation and binding problem formulation**,** Left Edge Algorithm for binding**,** ILP formulation of allocation and binding, register allocation and binding for efficient design, hierarchical graph-based allocation and binding, multi-port binding problem and its impact on resource sharing, impact of clock constraints on binding**,** datapath and controller synthesis techniques, interconnect synthesis and trade-offs in datapath architecture.

Module III: HLS for Arrays and Loops (5 Hours)

High-Level Synthesis for array-based designs, handling memory in HLS, HLS for loops, impact of loop transformations on hardware, HLS for loop pipelining**,** loop unrolling, loop fusion, software pipelining, initiation interval minimization, design trade-offs in loop optimizations, scheduling loops for minimum latency and area, impact of memory hierarchy on loop optimizations, loop tiling and array partitioning, and array dependency analysis in HLS.

Module IV:

Optimizing C-Code for Hardware Efficiency (5 Hours)

Hardware-efficient C coding principles, writing C-code with HLS-aware optimizations, dataflow optimization in HLS, control and data dependency analysis, frontend optimizations in C, handling function calls and recursion in HLS, effects of compiler optimizations on circuit performance, hardware/software partitioning, area-power-performance trade-offs in C-based HLS, resource sharing and function inlining, impact of precision tuning and bit-width reduction in C-based hardware design.

Module V:

Verification and Reverse Engineering in HLS (5 Hours)
Simulation-based verification strategies for HLS-generated designs, phase-wise verification of HLS flow, verification of synthesized RTL, testbench creation for HLS-based designs, equivalence between C and RTL, debugging and refining synthesized designs, validation challenges in C-to-RTL transformation, RTL to C reverse engineering, synthesis validation methodologies, formal verification for HLS, and integrating HLS verification into existing ASIC/FPGA flows.

Module VI:
Advanced Topics in C-Based VLSI Design (5 Hours)
Introduction to hardware security, attacks on RTL logic locking and countermeasures, introduction to logic synthesis, FPGA technology mapping and optimizations, introduction to physical synthesis, circuit-level optimizations for power and performance, recent advances in C-based VLSI design, emerging trends in HLS-based ASIC and FPGA design, case studies on commercial HLS tools and research directions in high-level synthesis.

Text Book:

1. Mike Fingeroff, High-Level Synthesis Blue Book, Mentor Graphics Corporation, 2010.

2. D. D. Gajski, N. D. Dutt, A.C.-H. Wu and S.Y.-L. Lin, High-Level Synthesis: Introduction to Chip and System Design, Springer, 1st edition, 1992

Reference Books:

1. G. De Micheli. Synthesis and optimization of digital circuits, McGraw Hill, India Edition, 2003.

2. Philippe Coussy and Adam Morawiec, High-level Synthesis from Algorithm to Digital Circuit, Springer, 2008

Course Code	26EE51TH1207-03			
Category	OE			
Course Title	Machine Learning Engineering			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	II

Course Outcomes:

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

- Understand the fundamental concepts of machine learning, and get an insight of when to apply aparticular machine learning approach.
- Comprehend the underlying mathematical relationship within/across Machine Learningalgorithms.
- Apply machine-learning algorithms to complex engineering problems, optimize the modelslearned and report on the expected accuracy that can be achieved by applying the models.
- Design and implement deep neural networks for solving real-world problems in various domainsand test them with benchmark data sets.

Syllabus

Foundations and paradigms of Machine Learning

Supervised learning: K-Nearest Neighbors, Decision trees, Linear and Logistic Regression – Bias/Variance Trade-off, Overfitting, Regularization, Variants of Gradient Descent, Support Vector Machines, boosting and bagging, Ensemble methods such as Random Forest and Ada Boost.

Artificial Neural Networks: Perceptron, Multilayer networks, Backpropagation algorithm, Optimization algorithms,Introduction to Deep Neural networks, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), Brief introduction to ML applications in computer vision, and natural language processing using Tensorflow/Pytorch.

Probabilistic Machine Learning- Bayesian learning and Bayesian networks, Naive Bayes classifier; Bayes optimal classifiers, Maximum Likelihood Estimation, MAP; Gaussian Discriminant Analysis.

Unsupervised learning: Clustering, Expectation Maximization, and Gaussian Mixture Models. Dimensionality Reduction-PCA, LDA, and Feature Selection, PAC Learnability.

Text Book:

1. Understanding Machine Learning: From Theory to Algorithms, by Shai Shalev-Shwartz, ShaiBen-David, Third edition, Cambridge University Press, 2015.

2. Pattern Recognition and Machine Learning by Christopher M. Bishop, First edition, Springer,2006.
3. The Elements of Statistical Learning Data Mining, Inference, and Prediction by Trevor Hastie,Robert Tibshirani, Jerome Friedman, Second Edition, Springer, 2009.

Reference Books:

1. Machine learning, by Mitchell Tom, First edition, McGraw Hill, 1997.
2. Deep Learning by Ian Goodfellow, YoshuaBengio, Aaron Courville, & Francis Bach, MIT Press,2017.
3. Machine Learning: An Algorithmic Perspective by Stephen Marsland, Second Edition, Chapmanand Hall/CRC, 2014
4. Richard O. Duda, Peter E. Hart, David G. Stork. Pattern classification, Wiley, New York, 2001.
5. Machine Learning: A Probabilistic Perspective by Kevin P. Murphy, Francis Bach; MIT Press,2012.
6. Recent Research Papers from Reputed Journals and Conferences such as ICLR, NIPS, ICML,CVPR, PAMI etc.