COEP Technological University, Pune

(A Unitary Public University of Govt. of Maharashtra)

School of Engineering and Technology

Detailed Syllabus

T.Y. B. Tech.

Electronics and Telecommunication Engineering

Effective from: A.Y. 2026-27

Digital Signal Processing

Teaching Scheme	2-0-2-1 (L-T-P-S)
Credits	3
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE 50 | ESE 50

Course Outcomes

Students will be able to:

1. Analyse and characterize discrete time signals and systems.
2. Apply Fourier analysis concepts for frequency domain representation and analysis.
3. Implement discrete time systems using appropriate realization structures.
4. Design digital FIR and IIR filters for given specifications.
5. Calculate spectral density parameters and measure signal processing performance.

Unit I: Introduction to Discrete Time Signals and Systems [4 Hrs]

Introduction to discrete-time sequences and elementary signals, similarities and differences between CT and DT sinusoids. Z-transform, introduction to systems and their properties, LTI systems, linear convolution.

Unit II: Frequency Domain Analysis [6 Hrs]

DTFT, Discrete Fourier Transform. Direct computation of the DFT, Radix-2 Fast Fourier Transform, Goertzel algorithm, Chirp-z transform.

Unit III: Design of Digital FIR Filters [6 Hrs]

Magnitude and phase response of digital filters, frequency response of linear phase FIR filters, design techniques for FIR (lowpass, highpass, bandpass and bandreject) filters. Design of optimal linear phase FIR filters, design of minimum phase FIR filters.

Unit IV: Design of Digital IIR Filters [8 Hrs]

IIR filter design by approximation of derivatives, impulse invariant approach and bilinear transformation. Butterworth, Chebyshev, inverse Chebyshev and elliptic filters; design of lowpass, highpass, bandpass and bandreject IIR filters. Spectral transformation of IIR filters, effects of finite word length in digital filters. Spectral estimation, energy density spectrum, estimation of autocorrelation and power spectrum, DFT in spectral estimation, parametric and non-parametric methods for power spectrum estimation.

Unit V: Implementation of Discrete Time Systems [4 Hrs]

Structure for the realization of discrete time FIR and IIR systems: direct form, cascade form, frequency sampling structure, lattice structure. State space system analysis and structures. Round-off effects in digital filters.

Self-Study / Application

Introduction to AI and ML for DSP applications like speech, biomedical signal, radar signal processing.

Textbooks

[1] Discrete Time Signal Processing – Oppenheim & Schafer, PHI Ltd, Third Edition.

Reference Books

[1] Digital Signal Processing: Principles, Algorithms and Applications – Proakis & Manolakis.
[2] Digital Signal Processing: A Computer-Based Approach – Sanjit K. Mitra, McGraw Hill Education.

Note

[1] To measure CO1, questions may be of the type: define, identify, state, classify, list, represent using block diagram or signal flow graph, solve basic difference equations, etc.
[2] To measure CO2, questions may be of the type: explain, describe, illustrate frequency domain concepts, compute DFT/FFT, analyze spectral characteristics, compare different FFT algorithms, evaluate computational efficiency, etc.
[3] To measure CO3, questions will be based on implementation of discrete-time systems including realization structures, state-space representation, and analysis of finite word length and round-off effects.
[4] To measure CO4, questions may be of the type: design FIR/IIR filters for given specifications, analyze magnitude and phase response, compare different filter design techniques, justify selection of filter types, etc.
[5] To measure CO5, some questions may be based on computation and analysis of spectral density parameters, estimation of autocorrelation and power spectrum, comparison of parametric and non-parametric methods, and may also include self-study topics.

Digital Signal Processing Lab

Lab Course Outcomes

Students will be able to:

1. Verify & demonstrate basic concepts of digital signal processing.
2. Design and implement FIR filters (lowpass, highpass, bandpass, bandreject) using windowing techniques and evaluate their frequency response.
3. Design and implement IIR filters (Butterworth, Chebyshev) using impulse invariant and bilinear transformation methods and compare their performance.
4. Estimate the power spectral density of signals using parametric and non-parametric methods, and evaluate autocorrelation functions.
5. Apply DSP techniques to applications like speech, biomedical signal, etc.

List of Experiments

1. Generation of basic signals using C / Python and MATLAB.
2. Verification of Sampling theorem.
3. Program to obtain Linear Convolution of two finite length sequences and verify the properties of linear convolution.
4. Program to obtain Circular Convolution of two finite length sequences using FFT method using C / Python and MATLAB.
5. Program for computing auto correlation of two finite length sequences and obtain the energy of the input sequence.
6. Program for computing cross correlation of two finite length sequences and using it to find the delay between signals using C / Python and MATLAB.
7. To perform spectral analysis of discrete signal using C / Python and MATLAB.
8. Implementation of Low Pass and High Pass FIR filter for a given sequence.
9. Implementation of Low Pass and High Pass IIR filter for a given sequence using C / Python and MATLAB.
10. Implementation of Decimation Process.
11. Implementation of Interpolation Process using C / Python and MATLAB.

12. Generation of Sinusoidal signal through filtering using C / Python and MATLAB.
13. To perform audio/speech processing using C / Python and MATLAB.
14. To perform ECG signal processing using C / Python and MATLAB.
15. Implementing Linear Convolution and Circular Convolution in TMS320 C6748 DSP Processor.
16. DFT with FFT using TMS320 C6748 DSP Processor.
17. Noise removal: Add noise above 3 kHz and then remove; Interference Suppression using 400 Hz Tone using TMS320 C6748 DSP Processor.

RTL Design and FPGA Implementation

Teaching Scheme	3-0-2-2 (L-T-P-S)
Credits	4
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE 50 | ESE 50

Course Outcomes

Students will be able to:

1. Write codes for digital building blocks using Verilog HDL.
2. Design Finite State Machines.
3. Model digital designs including FSMs to processor architectures using knowledge of HDL language.
4. Identify data-path and control path for MIPS Processor.
5. Use IP cores for early development of product.
6. Discuss the concept of Neuromorphic computing in VLSI.

Unit I: HDL and Digital Building Blocks [10 Hrs]

Hardware Description Language: HDL fundamentals and design entry by Verilog / System Verilog, test benches. Digital Building Blocks: combinatorial and sequential building blocks using System Verilog. Simulation region: system tasks, active events, inactive events, non-blocking events and monitor events for System Verilog.

Unit II: Sequential Logic Design [10 Hrs]

Top-down approach to design, synchronous FSM design (Mealy and Moore machines), STA – Static Timing Analysis, meta-stability, clock issues, asynchronous FSM design. IP in various forms: soft IP, hard IP, case studies. Configurable logic devices: fused and anti-fused technologies, PAL, PLA, FPGA implementation.

Unit III: Data Path and Control Path Design [6 Hrs]

Unsigned multiplication, signed multiplication, GCD computation.

Unit IV: Instruction Set Architecture [4 Hrs]

Types of ISA, MIPS ISA, instruction encoding, MIPS general purpose registers, signed and unsigned instructions, floating point instructions, pseudo instructions.

Unit V: HDL Representation for MIPS Architecture [6 Hrs]

Addressing modes, exceptions, HDL representation of single cycle CPU data path and control path. Control path and data path design of multi-cycle processor and a pipelined processor, performance analysis.

Unit VI: Concepts of Neuromorphic Computing in VLSI [6 Hrs]

Roadmap on neuromorphic computing and engineering, Spiking Neural Networks (SNN) in digital circuits, digital synapses, digital neural models. Applications: robotics and autonomous systems, health and brain-computer interface (BCI), edge AI and IoT devices, cybersecurity and anomaly detection, space and aerospace. Advantages, challenges and future prospects.

Self-Study (SS) [6 Hrs]

Clock domain crossover, memory technologies, chiplet design, RISC-V.

Textbooks

[1] Stephen Brown and Zvonko Vranesic, "Fundamentals of Digital Logic with Verilog Design", McGraw-Hill.
[2] David Harris and Sarah Harris, "Digital Design and Computer Architecture", 2nd Edition, 2013, Morgan Kaufmann.

Reference Books

[1] Zainalabedin Navabi, "Verilog Digital System Design: RT Level Synthesis, Testbench and Verification", 2nd Edition, 2006, Tata McGraw-Hill.

[2] William Fletcher, "An Engineering Approach to Digital Design", Prentice Hall.

[3] Douglas Smith, "HDL Chip Design: A Practical Guide for Designing, Synthesizing & Simulating ASICs & FPGAs Using VHDL or Verilog", Doone Publications.

[4] David A. Patterson, John Hennessy, "Computer Organization and Design", Elsevier/Morgan Kaufmann.

[5] IEEE Standard HDL Based on Verilog HDL, IEEE.

Note

[1] To measure CO1, questions may be of the type: define, identify, state, match, list, name, etc.

[2] To measure CO2, questions may be of the type: explain, describe, illustrate, evaluate, give examples, compute, etc.

[3] To measure CO3, questions will be based on applications of core concepts.

[4] To measure CO4, questions may be of the type: true/false with justification, theoretical fill in the blanks, theoretical problems, prove implications or corollaries of theorems, etc.

[5] To measure CO5, some questions may be based on self-study topics and comprehension of unseen passages.

RTL Design and FPGA Implementation – Laboratory

Lab Course Outcomes

Students will be able to:

1. Use Verilog HDL and testbench to simulate, synthesize and implement Boolean functions, combinatorial and sequential designs on FPGA using EDA tools like Quartus II and Vivado.

2. Know the structure and steps of operation for implementation on FPGA.

3. Perform the static timing analysis on sequential designs.

4. Integrate IPs in their design.

List of Experiments / Assignments

1. Behavioral and structural modeling with various constructs for given specifications: Verilog implementation of Mux/Demux, Full Adder.
2. Verilog implementation of magnitude comparator, encoder/decoder, priority encoder, parity generator, code converters.
3. Verilog implementation of D Flip-Flop, shift registers (SISO, SIPO, PISO, bidirectional).
4. Verilog implementation of counters.
5. Verilog implementation of sequence generator/detectors, synchronous FSM – Mealy and Moore machines.
6. Verilog implementation of vending machine and traffic light controller.
7. Verilog implementation of single port SRAM.
8. Verilog implementation of generic building blocks like decoder, adder, shifter, register file and ALU used by MIPS architecture.
9. Assignment: HDL code for ATM and elevator control.
10. Assignment: HDL code for UART, SPI, I2C and Arbiter.
11. Assignment: HDL code for single cycle processor data path and control path.
12. Assignment: HDL code for MIPS instruction memory and data memory.

Embedded System Design

Teaching Scheme	2-0-2-1 (L-T-P-S)
Credits	3
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE 50 | ESE 50

Course Outcomes

Students will be able to:

1. Understand the fundamentals of processor architecture including RISC/CISC and ARM design philosophy.
2. Demonstrate the fundamentals of processor memory system including memory map and exception model.
3. Analyse and apply programming techniques with microcontrollers and I/O device interfacing.
4. Design and develop small embedded applications using microcontrollers.

Unit I: Introduction to Processor [8 Hrs]

CISC and RISC architecture, embedded processor/microcontroller architecture, processor (ARM) design philosophy, instruction set, features, architecture revision, processor families.

Unit II: Memory System [6 Hrs]

Programming model, memory system features, memory map, memory requirements, memory access attributes, exception model, fault handling, internal memory.

Unit III: I/O Devices [6 Hrs]

GPIO, PWM, timers, ADC, WDT, UART, SSI, I2C, CAN/USB.

Unit IV: Internal Peripheral Interfacing [4 Hrs]

System timers, NVIC, SCB, MPU, FPU, register map, system control, power management.

Self-Study: Embedded Software Development

Development suites, compilation process, software flow, microcontroller software interface standards, toolchain, debugger, adaptors, simulators, emulators. Note: Teacher Assessment (20M) based on Self-Study.

Textbooks

[1] Joseph Yiu and Reinhard Keil, "The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors", 3rd Edition, Newnes, ISBN: 978-0124080829.

[2] Mazidi & Naimi, "TI Tiva ARM Programming For Embedded Systems", 1st Edition, MicroDigitalEd, 2017.

[3] Sloss, Symes & Wright, "ARM System Developer's Guide: Designing and Optimizing System Software", Morgan Kaufmann, 2004.

Reference Books

[1] Marilyn Wolf, "Embedded System Interfacing: Design for the IoT and CPS", Morgan Kaufmann.

[2] Shibu K. V., "Introduction to Embedded Systems", 2nd Edition, McGraw Hill Education (India), 2016.

[3] Louis E. Frenzel, "Handbook of Serial Communications Interfaces", 1st Edition.

Note

[1] To measure CO1, questions may be of the type: define, identify, state, match, list, name, etc.

[2] To measure CO2, questions may be of the type: explain, describe, illustrate, evaluate, give examples, compute, etc.

[3] To measure CO3, questions will be based on applications of core concepts.

[4] To measure CO4, questions may be of the type: true/false with justification, theoretical problems, design problems, etc.

Embedded System Design – Laboratory

Lab Course Outcomes

Students will be able to:

1. Interpret datasheets related to microcontrollers and associated peripheral devices.
2. Design, develop and test applications using microcontrollers and associated peripherals.
3. Design and implement experiments with different paradigms such as polled mode and interrupt driven mode.

List of Experiments / Assignments

1. LED blinking using GPIO.
2. Display switch presses count on seven segment display.
3. Blink an LED with software delay; delay generated using the SysTick timer.
4. System clock real time alteration using PLL modules.
5. UART echo test.
6. Control an LED flashing using switch by polling method and by interrupt method.
7. Control intensity of an LED / speed of a motor using PWM.
8. Capture analog readings of a rotary potentiometer connected to an ADC channel.
9. Read, write and verify a serial EEPROM using I2C.
10. Read, write and verify a serial EEPROM using SPI interface.
11. Evaluate various sleep modes by putting core in sleep and deep sleep modes.
12. Evaluate and understand the functionality of Watchdog Timer (WDT).
13. Calling C function from assembly program and vice versa.

PEC-01: Program Specific Elective – I

Students select ONE of the following options

PEC-01(a): Digital Image Processing and Applications

Teaching Scheme	3-0-2-1 (L-T-P-S)
Credits	4
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE 50 | ESE 50

Course Outcomes

Students will be able to:

1. Explain the fundamental concepts and techniques of digital image processing.
2. Apply image enhancement and restoration techniques in both spatial and frequency domains.
3. Implement image segmentation and feature extraction methods for various applications.
4. Analyse and apply appropriate image compression techniques.
5. Develop practical solutions for real-world problems using image processing algorithms.

Unit I: Introduction to Image Formation, Acquisition and Digital Image Processing [6 Hrs]

Image acquisition, image formation using pinhole camera, image magnification, image formation by lens, lens defocus, depth of field, effect of depth of field by aperture, hyperfocal distance. Fundamental steps in image processing, image acquisition and sampling, quantization and image representation, applications of digital image processing.

Unit II: Image Enhancement Techniques [6 Hrs]

Spatial domain: intensity transformations (contrast stretching, gray level slicing, negative transformation), spatial filtering (box filter, averaging filter, smoothing and noise reduction), histogram modification (histogram equalization, contrast enhancement). Frequency domain: frequency transformation, LPF for smoothing and ringing artifacts, HPF for edge enhancement, BPF for selective frequency enhancement, homomorphic filtering.

Unit III: Image Segmentation and Feature Extraction [8 Hrs]

Point, line, and edge detection using first-order and second-order derivative operators; Sobel and Prewitt gradient-based edge detection; Laplacian, LoG, DoG; Canny edge detector; edge linking; Hough transform. Thresholding: global, adaptive, and optimal. Region-based segmentation: region growing and region splitting/merging. Morphological processing: dilation, erosion, opening, closing. Feature extraction: Harris corner detector, texture-based features, shape descriptors, boundary descriptors.

Unit IV: Image Degradation and Restoration [6 Hrs]

Image restoration vs. enhancement, image degradation models, types of noise (Gaussian, salt & pepper, Poisson, speckle), degradation function and PSF, estimation of degradation function. Motion blur and atmospheric turbulence. Restoration techniques: inverse filtering, Wiener filtering, CLS filtering, blind deconvolution. Performance evaluation metrics (MSE, PSNR).

Unit V: Image Compression [6 Hrs]

Compression models, compression ratio, relative redundancy. Lossless: run-length coding, Huffman coding, Shannon-Fano, arithmetic coding, bit-plane coding, LZW, predictive coding. Lossy: lossy predictive coding, transform coding (DCT-based), vector quantization. JPEG compression standard, rate-distortion concepts.

Textbooks

[1] Rafael C. Gonzalez and Richard Woods, "Digital Image Processing", 3rd Edition, Pearson.
[2] Anil K. Jain, "Fundamentals of Digital Image Processing".

Reference Books

[1] Ethem Alpaydin, "Introduction to Machine Learning", MIT Press, 3rd Edition, 2014.
[2] H. Stark & J. W. Woods, "Probability and Random Processes with Applications to Signal Processing", 2014.

Note

[1] To measure CO1, questions may be of the type: define, identify, state, match, list, name, etc.
[2] To measure CO2, questions may be of the type: explain, describe, illustrate, evaluate, give examples, compute, etc.
[3] To measure CO3, questions will be based on applications of core concepts.
[4] To measure CO4, questions may be of the type: true/false with justification, theoretical fill in the blanks, theoretical problems, prove implications or corollaries of theorems, etc.
[5] To measure CO5, some questions may be based on self-study topics and comprehension of unseen passages.

Digital Image Processing and Applications – Laboratory

Lab Course Outcomes

Students will be able to:

1. **1.** Implement fundamental image processing operations such as grayscale conversion, enhancement, and filtering.
2. **2.** Analyse and apply various spatial domain techniques for improving image quality.
3. **3.** Design and implement edge detection algorithms and interpret outputs for feature extraction.
4. **4.** Compare performance of different image processing techniques based on visual and quantitative criteria.
5. **5.** Develop image processing pipelines integrating multiple operations to solve real-world problems.

List of Experiments / Assignments

1. Acquire grayscale and colour images, perform basic manipulations (resize, crop), and display using image processing software.
2. Enhance image contrast using histogram equalization and apply spatial filters (smoothing and sharpening).
3. Implement the DFT on an image and apply low-pass and high-pass filtering in the frequency domain.
4. Simulate common noise models (Gaussian, Salt & Pepper) and apply restoration techniques (median, Gaussian, Wiener filtering).
5. Apply geometric operations: scaling, rotation, translation, and affine transformation.
6. Detect edges using gradient-based operators: Sobel, Prewitt, and Canny edge detectors.
7. Segment images using thresholding, region growing, and morphological-based segmentation.
8. Convert images between colour spaces (RGB to HSI) and apply pseudo-colouring and colour enhancement.
9. Perform binary morphological operations (erosion, dilation, opening, closing) for shape analysis.
10. Implement basic image compression techniques: run-length encoding and JPEG compression concepts.

PEC-01(b): Information Theory and Coding

Teaching Scheme	3-0-2-1 (L-T-P-S)
Credits	4
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE: 30 ESE: 50

Course Outcomes

Students will be able to:

1. Design, implement and compare coding techniques for memoryless and stationary discrete sources.
2. Implement binary block and convolutional channel coding techniques for error detection and correction.
3. Estimate error detection and correction capabilities of block and convolutional channel codes.
4. Explore calculations of channel capacity to support error-free transmission using source and channel coding algorithms.

Unit I: Information Measures [4 Hrs]

Discrete source models – memoryless and stationary. Mutual information, self information, conditional information, average mutual information, entropy, entropy of the block, conditional entropy, information measures for analog sources.

Unit II: Coding Techniques for Discrete Sources [8 Hrs]

Shannon's source coding theorem, prefix-free codes, Kraft inequality. Huffman coding algorithm – design and optimality. Arithmetic coding. Shannon-Fano coding. Lempel-Ziv algorithms (LZ77, LZ78, LZW). Efficiency and redundancy calculations.

Unit III: Channel Capacity and Channel Coding Theorem [6 Hrs]

Channel models – BSC, BEC, AWGN channel. Channel capacity calculations. Shannon's channel coding theorem and achievability. Capacity of AWGN channel, Shannon limit.

Unit IV: Linear Block Codes [8 Hrs]

Binary block codes – Hamming distance, error detection and correction capability. Generator matrix, parity check matrix, syndrome decoding. Cyclic codes: CRC, BCH codes. Hamming codes, Golay codes. Performance of block codes over BSC.

Unit V: Convolutional Codes [6 Hrs]

Convolutional encoder structure, state diagram, trellis diagram. Viterbi algorithm for maximum likelihood decoding. Systematic and recursive systematic convolutional codes. Turbo codes: principle and iterative decoding overview. Puncturing and rate compatibility.

Self-Study

LDPC codes overview; polar codes and their application in 5G standards; comparison of modern coding schemes.

Textbooks

[1] David J. C. MacKay, "Information Theory, Inference, and Learning Algorithms", Cambridge University Press.
[2] Thomas M. Cover and Joy A. Thomas, "Elements of Information Theory", Wiley, 2nd Edition.

Reference Books

[1] Rolf Johannesson and Kamil Sh. Zigangirov, "Fundamentals of Convolutional Coding", IEEE Press.
[2] Shu Lin and Daniel J. Costello, "Error Control Coding", Pearson, 2nd Edition.

Note

[1] To measure CO1, questions may be of the type: define, identify, state, match, list, name, etc.
[2] To measure CO2, questions may be of the type: explain, describe, illustrate, evaluate, give examples, compute, etc.
[3] To measure CO3, questions will be based on applications of core concepts.
[4] To measure CO4, some questions may be based on self-study topics and comprehension of unseen passages.

Information Theory and Coding Laboratory

Lab Course Outcomes

Students will be able to:

1. Design and deploy various encoding and decoding techniques for discrete sources using programming languages such as C/C++.
2. Build the hardware and evaluate performance of block and convolutional encoders and decoders.

List of Experiments / Assignments

1. Entropy computations – Average Self Information (Entropy), Conditional entropy, Average mutual information for binary and M-ary sources.
2. Design and Implement Huffman encoder and decoder for the given string.
3. Design and Implement Shannon-Fano encoder and decoder for the given string.
4. Design and Implement Lempel-Ziv-Welch encoder and decoder for the given string.
5. Build and test the performance of Linear Block Codes: Encoder and Decoder.
6. Build and test the performance of Cyclic Codes: Encoder and Decoder
7. Build and test the performance of Convolution Codes: Encoder and Decoder

NOTE: The programming is to be carried out using C, C++. The hardware implementation is to be done using digital circuits/kits available in the lab.)

PEC-01(c): Analog and Mixed Signal IC Design

Teaching Scheme	3-0-2-1 (L-T-P-S)
Credits	4
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: None

Course Outcomes

Students will be able to:

1. Analyse and design advanced analog building blocks including current mirrors and amplifiers.
2. Implement data conversion systems (ADC and DAC) and verify designs through CAD tools.
3. Develop mixed-signal subsystems and verify designs through CAD simulation tools.
4. Evaluate fundamental limitations including noise, bandwidth-power trade-offs and technology constraints.

Unit I: Fundamentals and MOS Modeling [8 Hrs]

Second-order effects: impact of short-channel effects, noise, and frequency-dependent parameters in CMOS technologies. Current sources and mirrors: basic current mirrors, cascode, Widlar and Wilson current mirrors. Reference circuits: current reference, startup circuits, bandgap reference circuit, current mode bandgap reference. Self-Study: MOS device physics, secondary order effects.

Unit II: Analog Amplifier Design [10 Hrs]

Single-stage amplifiers: common-source, common-gate, common-drain (source follower), cascode amplifiers. Differential pair and tail current source. Multistage amplifiers: Miller compensation, cascaded stages. Frequency response and stability. Op-amp design: single-stage and two-stage CMOS op-amps, output stages, frequency compensation.

Unit III: Feedback and Noise in Analog Circuits [6 Hrs]

Feedback topologies: series-series, shunt-shunt, series-shunt, shunt-series. Stability analysis: Bode plots, phase margin, gain margin. Noise fundamentals: thermal noise, flicker (1/f) noise, shot noise. Input-referred noise, noise figure, noise in CMOS amplifiers.

Unit IV: Data Conversion Systems [8 Hrs]

D/A converters: resistor string, binary-weighted, R-2R ladder, current steering DACs; DNL and INL. A/D converters: flash, successive approximation (SAR), pipeline, delta-sigma converters. Resolution, sampling rate, SNDR, ENOB, aperture jitter. Anti-aliasing filter design.

Unit V: Mixed-Signal Design and Layout [6 Hrs]

Mixed-signal design challenges: substrate noise, supply bounce, grounding strategies. Clock distribution and jitter. Phase-locked loops (PLLs): VCO, phase detector, loop filter, charge pump PLL. Layout techniques: matching, shielding, guard rings, differential routing.

Self-Study

Switched capacitor circuits; overview of RF front-end design; CMOS image sensors.

Textbooks

[1] Behzad Razavi, "Design of Analog CMOS Integrated Circuits", McGraw-Hill, 2nd Edition.

[2] Paul R. Gray, Paul J. Hurst, Stephen H. Lewis, Robert G. Meyer, "Analysis and Design of Analog Integrated Circuits", Wiley, 5th Edition.

Reference Books

[1] Willy Sansen, "Analog Design Essentials", Springer.
[2] David A. Johns and Ken Martin, "Analog Integrated Circuit Design", Wiley.

Note

[1] To measure CO1, questions may be of the type: define, identify, state, match, list, name, etc.
[2] To measure CO2, questions may be of the type: explain, describe, illustrate, evaluate, give examples, compute, etc.
[3] To measure CO3, questions will be based on applications of core concepts.
[4] To measure CO4, some questions may be based on self-study topics and comprehension of unseen passages.

Analog and Mixed Signal IC Design Laboratory

Lab Course Outcomes

Students will be able to:

1. Differentiate between the DC and transient performance characteristics of fundamental CMOS building blocks, such as MOSFETs, inverters, and various current mirror topologies, using industry-standard EDA simulation tools.
2. Design complex analog and mixed-signal integrated circuits, including two-stage operational amplifiers, comparators, and data converters (ADC/DAC), to meet specific engineering constraints such as gain, bandwidth, and phase margin.
3. Appraise the reliability of circuit designs by performing pre- and post-layout simulations, justifying design trade-offs through the verification of Design Rule Checks (DRC), Layout vs. Schematic (LVS), and parasitic extraction (PEX).

List of Experiments / Assignments

1. Lab 1 - MOSFET I-V Characteristics: Study ID vs. VGS and ID vs. VDS for NMOS and PMOS transistors. Analyse the effect of substrate potential (body effect) on threshold voltage.
2. Lab 2 - CMOS Inverter Analysis: Perform DC and transient analysis to determine noise margins, switching thresholds, and propagation delay.
3. Lab 3 - Current Mirrors: Design and simulate simple, cascode, and Wilson current mirrors to evaluate output resistance and current matching performance.
4. Lab 4 - Single-Stage Amplifiers: Design Common-Source, Common-Gate, and Common-Drain (Source Follower) amplifiers. Compare resistive loads vs. active (current mirror) loads.
5. Lab 5 - Differential Amplifiers: Characterize differential gain, common-mode gain, and Common-Mode Rejection Ratio (CMRR).
6. Lab 6 - Two-Stage CMOS Op-Amp: Design a complete operational amplifier to meet specific gain, bandwidth (GBW), and phase margin requirements. This often includes implementing Miller frequency compensation.
7. Lab 7 - Comparators: Design high-speed and high-gain comparators, often focusing on regenerative latches and offset cancellation techniques.
8. Lab 8 - Switched-Capacitor Circuits: Implement basic switched-capacitor integrators or filters, observing effects like charge injection and clock feed-through.
9. Lab 9 - Oscillators and PLLs: Design 3-stage or 5-stage Ring Oscillators or Voltage Controlled Oscillators (VCO). Advanced labs may include a basic Phase-Locked Loop (PLL) design.
10. Lab 10 - Digital-to-Analog Converters (DAC): Design and simulate resistor-string or current-steering DAC architectures.
11. Lab 11 - Analog-to-Digital Converters (ADC): Implement architectures such as Flash ADCs (using multiple comparators and an encoder), SAR ADCs, or Pipelined ADCs.
12. Lab 12 - Pre- and Post-Layout Simulation: After schematic design, create the physical layout. Perform Design Rule Checks (DRC), Layout vs. Schematic (LVS) verification, and parasitic extraction (PEX) to compare post-layout performance with initial simulations.

Simulations using industry-standard EDA tools like Cadence Virtuoso, Spectre, or LTspice for schematic capture, simulation, and layout.

PEC-01(d): Automotive Electronics

Teaching Scheme	3-0-2-1 (L-T-P-S)
Credits	4
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE:50 ESE:50

Course Outcomes

Students will be able to:

1. Understand the architecture and working of automotive microcontrollers.
2. Implement structured programming constructs for real-time automotive systems.
3. Explain vehicle communication architectures and protocol stacks.
4. Evaluate security mechanisms such as authentication, encryption, and secure communication protocols.
5. Evaluate emerging automotive technologies and industry trends.

Unit I: Hardware Layer / Embedded Systems [8 Hrs]

Introduction to microcontrollers and processors; sensors and actuators in automotive systems: wheel speed sensors, radar, LiDAR, cameras, temperature sensors; brake systems; steering motors. Comparison: power, cost, uses and applications.

Unit II: Embedded Software and Programming [8 Hrs]

Introduction to C/Python programming: variables and data types, conditional statements, loops, functions, I/O. Embedded system programming: digital inputs, analog inputs, sensor data reading, actuator control.

Unit III: Vehicle Communication Architectures [8 Hrs]

Overview of automotive communication bus systems. CAN protocol: frame structure, arbitration, error handling, CAN-FD. LIN bus: frame structure and applications. FlexRay and Ethernet in automotive. ECU architecture and AUTOSAR framework overview.

Unit IV: Automotive Cybersecurity [6 Hrs]

Threat landscape in connected vehicles. ISO/SAE 21434 overview. Cryptographic fundamentals: symmetric and asymmetric encryption, hashing. Secure boot, key management, certificate-based authentication. Intrusion detection in vehicular networks.

Unit V: Emerging Technologies in Automotive [6 Hrs]

ADAS and autonomous driving: sensor fusion, V2X communication. Electric vehicles (EV) and battery management systems (BMS). Over-the-air (OTA) software updates. AI/ML applications in automotive: object detection, path planning. ISO 26262 functional safety overview.

Self-Study

Electric powertrain architecture; V2G (vehicle-to-grid) concepts; recent automotive ECU case studies.

Textbooks

[1] Konrad Reif, "Automotive Mechatronics: Automotive Networking, Driving Stability Systems, Electronics", Springer.
[2] Nicolas Navet and Francoise Simonot-Lion, "Automotive Embedded Systems Handbook", CRC Press.

Reference Books

[1] William B. Ribbens, "Understanding Automotive Electronics", Butterworth-Heinemann, 8th Edition.
[2] ISO 26262 Functional Safety for Road Vehicles – Standard Documentation.

Note

[1] To measure CO1, questions may be of the type: define, identify, state, match, list, name, etc.

[2] To measure CO2, questions may be of the type: explain, describe, illustrate, evaluate, give examples, compute, etc.

[3] To measure CO3, questions will be based on applications of core concepts.

[4] To measure CO4, questions may be of the type: true/false with justification, theoretical fill in the blanks, theoretical problems, etc.

[5] To measure CO5, some questions may be based on self-study topics and comprehension of unseen passages.

Automotive Electronics Laboratory

Lab COs:

1. Simulate and analyse communication systems using NS-3, SUMO, and Wireshark, and evaluate network performance.
2. Design, simulate, and evaluate communication systems and network protocols in both traditional and vehicular environments.
3. Develop programming ability using C/C++/Python/Arduino.
4. Design and develop secure automotive applications using microcontrollers and sensor interfaces.

List of Experiments:

1. To interface analog sensors using ADC and control motor/actuator operation using PWM in a microcontroller-based automotive system.
2. To implement CAN communication by sending and receiving frames using the python-can library.
3. To capture and analyze network packets using Wireshark for basic protocol inspection.
4. Integrate SUMO traffic simulation with NS-3 network simulation to model realistic vehicular communication systems, enabling the analysis of mobility-aware network performance in vehicular environments.
5. To model and simulate communication networks using NS-3 and study the behaviors of network protocols under varying traffic and topology conditions.
6. To model and simulate communication networks using NS-3 and study the behaviors of network protocols under varying traffic and topology conditions.
7. To evaluate the performance of network protocols in NS-3 by analysing key metrics such as throughput, latency, packet delivery ratio (PDR), and packet loss under different simulation scenarios.
8. To demonstrate security weaknesses in automotive networks due to a lack of authentication, including Man-in-the-Middle attacks.
9. To demonstrate security weaknesses in automotive networks due to a lack of authentication, including DoS attacks.
10. Mini-Project

IoT and Edge Computing (OEC-03)

Teaching Scheme	2-0-0-1 (L-T-P-S)
Credits	2
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: None

Course Outcomes

Students will be able to:

1. Understand IoT architectures and the evolution toward edge/fog computing.
2. Design low-latency, scalable IoT systems.
3. Integrate AI/ML into IoT pipelines.
4. Analyse modern connectivity technologies including 5G and beyond.
5. Build secure, intelligent IoT applications.

Unit I: Internet of Things Concepts and Architectures [4 Hrs]

Introduction to IoT: definition, evolution, and real-world applications. IoT architecture models: 3-layer, 5-layer, and cloud-centric architectures. Core components: sensors, actuators, embedded systems. Data flow: Device to Gateway to Cloud to User. IoT platforms and middleware. Shift from cloud to edge/fog computing. Self-Study: latency, bandwidth, and privacy concerns.

Unit II: Introduction to Edge Computing Architectures [4 Hrs]

Edge computing fundamentals. Edge vs. fog vs. cloud computing. Edge nodes and edge data centers. Real-time processing requirements. Round trip time (RTT) and latency optimization. Edge AI basics. Self-Study: case study – resource-constrained computing.

Unit III: Networking for IoT and Data-Driven Systems [4 Hrs]

IoT communication models: request-response, publish-subscribe. Protocols: MQTT, CoAP, HTTP, AMQP. Wireless technologies: Zigbee, LoRaWAN, NB-IoT, Bluetooth Low Energy. Network topologies: star, mesh, hybrid. IPv6 and 6LoWPAN. Data aggregation and gateways. Self-Study: case study with LoRaWAN.

Unit IV: 5G and Beyond – IoT Connectivity [5 Hrs]

Introduction to 5G architecture. Key features: eMBB, URLLC, mMTC. Network slicing in 5G. Edge computing integration with 5G. Beyond 5G (B5G) and 6G concepts. IoT in autonomous vehicles and smart defence systems. Self-Study: mobile edge computing (MEC) in beyond 5G and 6G.

Unit V: Integration of ML and Data Science in IoT [5 Hrs]

Data lifecycle in IoT. Data preprocessing and feature engineering. ML models: regression, classification, clustering. Edge AI vs. cloud AI. Real-time analytics. Predictive maintenance models. Self-Study: examples with TinyML and embedded AI.

Unit VI: IoT Security using ML and Deep Learning [6 Hrs]

IoT security challenges: device vulnerabilities, data breaches. Cryptography basics: symmetric and asymmetric encryption. Authentication and access control. Blockchain for IoT security. ML-based anomaly detection. Intrusion detection systems (IDS). Deep learning in cybersecurity. Self-Study: case study for secure edge computing.

Textbooks

[1] Perry Lea, "IoT and Edge Computing for Architects", Packt Publishing, 2020.

[2] Raj Kamal, "Internet of Things: Architectures, Protocols and Applications", McGraw Hill Education, 2017.

[3] Perry Lea, "Edge Computing: From Hype to Reality", Manning Publications, 2018.

[4] Deepsubhra Guha Roy et al., "Edge Computing for IoT: Architectures, Use Cases and Innovations", 2025.

Reference Books

[1] Rajkumar Buyya & Satish Narayana Srirama, "Fog and Edge Computing: Principles and Paradigms", Wiley, 2019.
[2] Pethuru Raj and Anupama C. Raman, "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press.
[3] Rute C. Sofia and John Soldatos, "Shaping the Future of IoT with Edge Intelligence", River Publishers, 2024.

Note

[1] To measure CO1, questions may be of the type: define, identify, state, match, list, name, etc.
[2] To measure CO2, questions may be of the type: explain, describe, illustrate, evaluate, give examples, compute, etc.
[3] To measure CO3, questions will be based on applications of core concepts.
[4] To measure CO4, questions may be of the type: true/false with justification, theoretical fill in the blanks, theoretical problems, prove implications or corollaries of theorems, etc.
[5] To measure CO5, some questions may be based on self-study topics and comprehension of unseen passages.

Internship I

Teaching Scheme	– – – – (L-T-P-S)
Credits	1
Examination Scheme	CIE 60 | ESE 40

Course Outcomes

Students will be able to:

1. Apply theoretical knowledge from Semesters 1–4 to professional or research work contexts.
2. Develop professional communication skills and workplace conduct.
3. Critically reflect on personal learning and identify gaps in knowledge.
4. Produce a structured technical report documenting internship experience.
5. Present the work carried out.

Duration and Mode

Minimum 6-8 weeks during the summer break after 4th Semester. Mode: in-person, hybrid, or remote — all acceptable, subject to faculty mentor's approval.

Eligible Formats

[1] Industry/Company Internship.
[2] Internal Institute Internship (formal offer letter required).
[3] Research/Academic Institute (IIT, NIT, IISc, IISER, government R&D lab).
[4] Entrepreneurship/Self-Started Venture (registered with institute entrepreneurship cell).
[5] Open Source/Self-Directed Technical Project (faculty pre-approval required).
NOT ELIGIBLE: Purely administrative/sales/HR roles with zero technical component; internships completed before enrolment.

Assessment

A. Internship Report: 60 marks (structure, technical content, reflection, curriculum mapping, references).
B. Presentation & Viva: 40 marks (communication clarity, technical understanding, Q&A).
Total: 100 marks.

Textbooks

[1] AICTE Guidelines on Internships for Technical Education, 2023.

Reference Books

[1] National Internship Portal – https://internship.aicte-india.org

Note

[1] To measure all COs (CO1–CO5), assessment is based on the internship report and presentation/viva evaluation.

Mini Project

Teaching Scheme	0-0-4-2 (L-T-P-S)
Credits	3
Examination Scheme	Project: CIE 60 | ESE 40 | Total: 100

Course Outcomes

Students will be able to:

1. Integrate and apply knowledge from at least two courses studied in Semesters I–IV to design a working system or prototype.

2. Identify a well-scoped engineering problem, define clear objectives, and develop a structured project plan.

3. Design, build, test, and document a technical solution using appropriate tools, methodologies, and best practices.

4. Work effectively as a team with defined roles, maintaining a weekly progress log and demonstrating consistent engagement.

5. Present the completed project through a formal end-semester panel presentation, viva, and a structured project report.

CIE – Continuous Internal Evaluation (60 Marks)

Guide Continuous Evaluation – regularity, discipline, initiative, quality of weekly interaction, iterative improvement (30 marks). Project Proposal and Weekly Progress Log – completeness, regularity, quality (10 marks). Internal Working Demonstration – prototype/system completeness and functionality assessed mid-to-late semester by internal panel (20 marks).

ESE – End-Semester Evaluation (40 Marks)

Panel Presentation – structured 15-minute presentation with Q&A (20 marks). Final Project Report – written quality, completeness, IEEE format, integration of theory and results (10 marks). Viva Voce – individual technical depth, ability to explain design decisions and defend results (10 marks).

Textbooks

[1] IEEE format guide for technical reports.

Reference Books

[1] Project management and documentation guidelines issued by the Programme Coordinator.

Note

[1] Registration for Mini Project is compulsory. Non-submission results in an F grade.

[2] Guide CIE marks (30 marks) must be submitted to the Programme Coordinator at least one week before the ESE date.

[3] The ESE Evaluation Panel must have a minimum of two faculty members; the project guide does not serve as panel examiner.

[4] Declaration of AI-generated code/content is mandatory. Undisclosed use is treated as plagiarism.

[5] A team member who fails to participate in the ESE will receive zero for the ESE component.

PEC-02: Program Specific Elective-II (Select Any ONE)

Fiber Optic Communications	
Teaching Scheme (L-T-P-S)	3-0-2-1 (L-T-P-S)
Credits	4
Course Code	–
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE 50 | ESE 50

Course Outcomes

Students will be able to:

1. **1.** Explain the principles, advantages, and key elements of fiber optic communication systems.
2. **2.** Describe the modes propagating in optical fiber using mode theory and classify fiber types.
3. **3.** Analyze the transmission characteristics including attenuation, dispersion, and polarization effects in optical fibers.
4. **4.** Analyze the performance of optical sources, detectors, and design optical links using power and rise time budgets.
5. **5.** Illustrate the concepts and components used in WDM including passive and active optical components.
6. **6.** Design an optical network and analyze its performance including SONET/SDH and passive optical networks.

Unit 1: Overview of Optical Fiber Communications [5 Hrs]

Motivations for lightwave communications, optical spectral bands, decibel units, network information rates, key elements of optical fiber systems, standards for optical fiber communications.

Unit 2: Optical Fibers – Structures, Waveguiding and Fabrication [7 Hrs]

Introduction to vector nature of light, basic optical laws and definitions, optical fiber modes and configurations, mode theory of circular waveguide, single mode fibers, graded index fibers, fiber materials, photonic crystal fibers, fiber fabrication, mechanical properties of fiber.

Unit 3: Transmission Characteristics of Optical Fiber [8 Hrs]

Attenuation, material absorption loss, scattering loss, bending loss, dispersion, chromatic dispersion, intermodal dispersion, polarization mode dispersion, dispersion modified single mode fibers. International fiber standards.

Unit 4: Optical Sources, Detectors and Link Design [8 Hrs]

LEDs and laser diodes, photodetectors PIN-diodes, APDs, detector responsivity, noise, optical receivers. Optical link design, power budget and rise time budget.

Unit 5: WDM Concepts and Components [6 Hrs]

Overview of Wavelength Division Multiplexing, passive optical couplers, isolators and circulators, fiber grating filters, active optical components, tunable light sources, Erbium Doped Fiber Amplifier, Raman Amplifier.

Unit 6: Optical Networks [8 Hrs]

Network concepts, network topologies, SONET/SDH, high speed lightwave links, Optical Add/Drop Multiplexing, optical switching, passive optical networks, optical Ethernet, optical fiber system performance monitoring and measurement, Optical Time Domain Reflectometer (OTDR).

Self-Study (SS): Modulation, BER [6 Hrs]

Advanced modulation techniques for optical systems, bit error rate analysis, performance metrics in optical communication links.

Textbooks

[1] Gerd Keiser, "Fibre Optic Communication", McGraw-Hill, 5th Edition, 2010.
[2] John M Senior, "Optical Fiber Communications Principles and Practice", Pearson, 3rd Edition.

Reference Books

[1] Siva Ram Murthy, Mohan Guruswamy, "WDM Optical Networks Concepts Design and Algorithms", PHI Eastern Economy Edition, 2001.
[2] Nakazawa, Masataka, Kazuro Kikuchi, and Tetsuya Miyazaki, eds. High Spectral Density Optical Communication Technologies, Vol. 6, Springer Science & Business Media, 2010.

Note

[1] To measure CO1, questions may be of the type – define, identify, state, explain, list etc. [2] To measure CO2, questions may be of the type – describe, classify, differentiate etc. [3] To measure CO3, questions may involve calculation of attenuation, dispersion and related parameters. [4] To measure CO4, questions will be based on link design and budget calculations. [5] To measure CO5, questions may involve analysis and selection of WDM components. [6] To measure CO6, questions may include design and performance analysis of optical networks.

Fiber Optic Communications Laboratory

Lab Course Outcomes

Students will be able to:

1. Prepare optical fiber ends and experimentally determine the numerical aperture, understanding light propagation in optical fibers.
2. Design and deploy various optical networks and analyze their performance.
3. Measure fiber characteristics, fiber damage and splice loss/connector loss using OTDR.
4. Analyze and measure different types of losses in optical fibers, including attenuation and bending losses.
5. Set up and evaluate a fiber optic digital communication link, demonstrating basic transmission principles.
6. Perform splicing of optical fibers and measure splice loss, understanding practical joining techniques.
7. Characterize laser diode performance by plotting P-I and V-I characteristics and determining slope efficiency.

List of Experiments / Assignments

1. Study of Basics of Optical Fiber (FOC) Cables and Connectors.
2. Measurement of numerical aperture of a fiber after preparing the fiber ends.
3. Study of losses in optical fiber.
4. Setting up of fiber optic digital link.
5. Preparation of splice joint and measurement of splice loss.
6. Power vs. current (P-I) characteristics and measurement of slope efficiency of laser diode.
7. Voltage vs. current (V-I) characteristics of laser diode.
8. Power vs. current (P-I) characteristics and measurement of slope efficiency of LED.
9. Voltage vs. current (V-I) characteristics of LED.
10. Characteristics of photodiode and measurement of responsivity.
11. Characteristics of avalanche photodiode (APD) and measurement of responsivity.
12. Measurement of fiber characteristics, fiber damage and splice loss/connector loss by OTDR.

Power Electronic Devices

Teaching Scheme (L-T-P-S)	3-0-2-1 (L-T-P-S)
Credits	4
Course Code	–
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE 50 | ESE 50

Course Outcomes

Students will be able to:

1. Understand the working, characteristics, ratings, and protection of modern power semiconductor devices.
2. Analyse and design DC-DC converters and choppers for industrial and EV applications.
3. Evaluate performance of different power converter topologies under varying load conditions for inverters (single phase and three phase).
4. Apply concepts of digital electronics controllers and processors in the control of power electronic devices.
5. Analyze modern soft-switched converters and understand EMI/EMC standards applicable to power converters.
6. Simulate and analyze power converter circuits using software tools such as MATLAB/PSIM.

Unit 1: Power Devices and Their Properties [7 Hrs]

Structure, static and switching characteristics, trigger requirements, ratings, protection and snubber circuit and application of: Power Diode, SCR, IGBT, Power MOSFET. Wide Bandgap (WBG) devices and comparison: GaN, SiC devices, IPM.

Unit 2: DC-DC Converter [8 Hrs]

MOSFET based choppers: Buck, Boost, Buck-Boost converters – working, duty cycle equation, output waveforms, performance analysis, continuous and discontinuous conduction mode, Voltage Mode Control, Current Mode Control, modeling of VM in CCM, power loss and efficiency calculation.

Unit 3: Modern Power Converters [8 Hrs]

Hard switched and soft switched converters, ZVS and ZCS converter, LLC Converter, Phase Shifted Full Bridge Converter, Bidirectional DC-DC converter, Flyback SMPS. Introduction to EMI/EMC: basic concepts, sources of EMI, and types of coupling mechanisms. EMC Standards and Regulations: CISPR, MIL-STD, FCC.

Unit 4: AC-DC Converter [8 Hrs]

Principle of phase controlled converter; uncontrolled, semi-controlled and fully controlled converters in single-phase and three-phase configurations. Performance parameters and input-output waveforms for R, R-L loads, harmonic analysis. Improved power quality AC-DC converters (introduction to PFC), single phase and three phase unity power factor converters.

Unit 5: DC-AC Converter [8 Hrs]

PWM Inverters: single phase and three phase circuits, principle of operation, performance parameters. Multilevel Inverters: Cascaded H Bridge, Neutral Point Clamped. Sinusoidal PWM, Space Vector PWM. Algorithm for PWM generation for power converters.

Unit 6: Digital Control of Power Converters [8 Hrs]

FPGA, Microcontroller and DSP-based converter control; Closed-loop control: PI, PID, Model Predictive Control. Case Study: Microcontroller-Based DC-DC Converter; Case Study: FPGA-Based Inverter Control; Case Study: Closed-Loop Buck Converter.

Textbooks

[1] Ned Mohan, Siddharth Raju, "Simulations and Laboratory Implementations", Wiley, 2nd Edition, 2022.
[2] P. C. Sen, "Power Electronics", Tata McGraw Hill Publication, 2nd Edition, 2017.
[3] M. H. Rashid, "Power Electronics", PHI Publication, 3rd Edition, 2004.
[4] L. Umanand, Power Electronics: Essentials and Applications, Wiley India Pvt. Ltd., 1st Edition, 2009.

Reference Books

[1] Robert W. Erickson and Dragan Maksimovic, Fundamentals of Power Electronics, Springer, 2nd Edition, 2001.
[2] Keith H. Sueker, "Power Electronic Design: A Practitioner's Guide", Elsevier Publication.
[3] Kumar L. Ashok, "Power Electronics with MATLAB", Cambridge University Press.
[4] B W Williams, "Power Electronics: Devices, Drivers, Applications and Passive Components", Mac-Millan Publication.

Note

[1] To measure CO1, questions may be of the type – define, identify, state, match, list, name etc. [2] To measure CO2, questions may be of the type – explain, describe, illustrate, evaluate, give examples, compute etc. [3] To measure CO3, questions will be based on applications of core concepts. [4] To measure CO4, questions may be of the type – design and implement digital control algorithms. [5] To measure CO5, questions may be based on EMI/EMC standards and soft-switching concepts. [6] To measure CO6, questions may involve simulation and analysis of converter circuits.

Power Electronic Devices Laboratory

Lab Course Outcomes

Students will be able to:

1. Evaluate the V-I characteristics for different power semiconductor devices.
2. Demonstrate the operation and control techniques of power converters.
3. Analyze the waveforms exhibited at the input and output ports of the converters.
4. Simulate and analyze different converters with their control strategies.

List of Experiments / Assignments

1. To evaluate SCR characteristics and their turn-on methods.
2. To evaluate IGBT/MOSFET characteristics, loss calculations and measurement of Rds-on and parasitic capacitances including Miller capacitor.
3. To evaluate the performance of single-phase full bridge diode rectifiers with R and RL load.
4. To evaluate the performance of single-phase full bridge controlled rectifiers with R and RL load.
5. To evaluate the open loop performance of DC-DC Buck converter.
6. To evaluate the open loop performance of DC-DC Boost converter.
7. To evaluate the open loop performance of DC-DC Flyback converter.
8. To evaluate the performance of LLC converter.
9. To evaluate the performance of three phase inverter using different PWM techniques.
10. To evaluate the performance of PWM rectifier.
11. To perform MATLAB/PSIM Simulation of Unity Power Factor Converter. (Perform any two from): MATLAB/PSIM Modeling and analysis of DC-DC converter; MATLAB/PSIM Simulation of Multilevel Inverter or various PWM strategies; Introduction to LTSPICE circuit simulator.

Verification using System Verilog

Teaching Scheme (L-T-P-S)	3-0-2-1 (L-T-P-S)
Credits	4
Course Code	–
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE 50 | ESE 50

Course Outcomes

Students will be able to:

1. Understand and apply SystemVerilog data types, advanced arrays, and testbench layers including the layered testbench architecture.

2. Analyze and implement OOP concepts including classes, inheritance, polymorphism, and inter-process communication (IPC) mechanisms in SystemVerilog.

3. Design and create constrained random stimulus using SystemVerilog randomization constructs and connect testbench to DUT using interfaces.

4. Evaluate and compare functional coverage using covergroups and SystemVerilog Assertions (SVA) to measure verification completeness.

Unit 1: Verification Guidelines and Foundations [8 Hrs]

Verification Roadmap: verification plans, stimulus vs. response, and the layered testbench architecture (Generator, Driver, Monitor, Scoreboard). SystemVerilog Data Types: 2-state vs. 4-state logic, enumerated types (enum), and user-defined structures (struct). Advanced Arrays: static vs. dynamic arrays, associative arrays, and queues for managing transaction data. Procedural Blocks: tasks vs. functions, automatic storage, and argument passing (by value, reference, and name).

Unit 2: Object-Oriented Programming (OOP) and Inter-Process Communication [8 Hrs]

Class Basics: defining classes, objects, handles, and the new() constructor. Advanced OOP: inheritance, polymorphism, and virtual methods for creating reusable verification components. Threads and Concurrency: fine-grained process control using fork...join, fork...join_any, and fork...join_none. Synchronization (IPC): using Events, Semaphores (for resource sharing), and Mailboxes (for message passing between testbench layers).

Unit 3: Constrained Random Verification (CRV) and Interfaces [8 Hrs]

Randomization: using rand and randc modifiers; the randomize() method and post_randomize functions. Constraint Design: simple constraints, distribution constraints (dist), implication (->), and solve...before constructs. Inline Constraints: overriding class constraints using the with block. Connecting to the DUT: the interface construct, modports, and Clocking Blocks to eliminate race conditions and define timing skews.

Unit 4: Assertions and Functional Coverage [8 Hrs]

SystemVerilog Assertions (SVA): immediate vs. concurrent assertions; sequences, properties, and the assert, assume, and cover statements. Functional Coverage: defining covergroups, coverpoints, and bins (explicit, transition, and cross-coverage). Verification Management: analyzing coverage reports to identify holes in the verification plan. Virtual Interfaces: bridging the gap between static design modules and dynamic class-based testbenches.

Textbooks

[1] System Verilog LRM (Language Reference Manual).

[2] Chris Spear, Gregory J Tumbush, "System Verilog for Verification – A Guide to Learning the Testbench Language Features", Springer, 2012.

Reference Books

[1] System Verilog LRM (Language Reference Manual).
[2] Chris Spear, Gregory J Tumbush, "System Verilog for Verification – A Guide to Learning the Testbench Language Features", Springer, 2012.

Note

[1] To measure CO1, questions may be of the type – Explain, Apply. [2] To measure CO2, questions may be of the type – Analyze, Implement, Construct. [3] To measure CO3, questions will be based on Design and Develop. [4] To measure CO4, questions may be of the type – Evaluate and Compare.

Verification using System Verilog Laboratory

Lab Course Outcomes

Students will be able to:

1. Implement advanced SystemVerilog data structures, including associative arrays, dynamic arrays, and mailboxes, to model complex hardware behaviors and inter-process communication in a verification environment.

2. Examine the efficiency of verification strategies by utilizing SystemVerilog Assertions (SVA) and Functional Coverage (Covergroups) to identify protocol violations and measure progress of the verification plan.

3. Design a complete layered, constrained-random verification environment comprising generators, drivers, monitors, and scoreboards to validate the functionality of complex digital designs like network routers or FIFOs.

List of Experiments / Assignments

1. Data Type Characterization: Implement code to compare 2-state (bit, int) and 4-state (logic, integer) variables; perform operations on Dynamic Arrays and Queues including resizing, sorting, and searching.

2. Associative Array Modeling: Model a sparse memory by adding 50 integer values at random locations in an Associative Array and verifying data retrieval.

3. Class-Based Transaction Modeling: Define a Transaction class with data members, custom constructor, and print method; perform Shallow Copy and Deep Copy operations between objects.

4. IPC Synchronization: Develop a producer-consumer model where two threads communicate using Mailboxes and synchronize using Events or Semaphores.

5. Constrained Random Stimulus Generation: Write constraints to generate unique, sorted, or one-hot random values; use Inline Constraints (with blocks) to override default behaviors.

6. Interface-Based Connection: Verify a Synchronous FIFO or Dual-Port RAM using the interface construct with modports and clocking blocks.

7. Protocol Verification with SVA: Implement Concurrent Assertions to check handshake protocols (e.g., Request-Acknowledge) and FIFO overflow/underflow conditions.

8. Functional Coverage Implementation: Define covergroups and coverpoints with specific bins to track the percentage of the verification plan completed.

9. Capstone Project: Layered Testbench for a Network Router or FIFO – Integrate classes and constrained random stimulus into a full layered environment (Generator, Driver, Monitor, Scoreboard). Simulations using Questa Sim, NC Verilog, ModelSim or equivalent tools.

Real-Time Operating System using QNX

Teaching Scheme (L-T-P-S)	3-0-2-1 (L-T-P-S)
Credits	4
Course Code	–
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE 50 | ESE 50

Course Outcomes

Students will be able to:

1. Describe the QNX OS architecture and its microkernel-based design including process/thread model and scheduling.

2. Develop and debug QNX-based applications using appropriate tools including the QNX Momentics IDE.

3. Apply process/thread management and synchronization techniques using mutexes, semaphores, and condition variables in QNX.

4. Implement inter-process communication methods including message passing, pulses, and shared memory for real-time systems.

5. Configure and build QNX boot/OS images for specific hardware platforms including startup code, kernel, drivers, and scripts.

6. Develop strong knowledge on POSIX standards that support system application development in embedded real-time environments.

Unit 1: Introduction to QNX OS Architecture [7 Hrs]

Overview of QNX OS architecture: microkernel, process manager, and standards. Protected address spaces, process/thread model, and scheduling. Introduction to inter-process communication (IPC) and synchronization. Resource managers and shared objects.

Unit 2: Processes, Threads, and Synchronization [6 Hrs]

Process management: creation, termination, and memory protection. Thread management: creation, termination, and synchronization. Synchronization techniques: mutexes, semaphores, and condition variables. Hands-on exercises: process/thread creation and synchronization.

Unit 3: Inter-Process Communication (IPC) [8 Hrs]

Overview of IPC methods in QNX: message passing, pulses, and shared memory. Comparing IPC methods: advantages and disadvantages. Practical implementation of IPC in QNX. Hands-on exercises: message passing and shared memory.

Unit 4: Hardware Programming and Timing [6 Hrs]

Hardware access methods: IO-mapped and memory-mapped IO. Interrupt handling and DMA-safe memory allocation. Timing architecture: periodic timing, one-shot timing, and timeouts. Hands-on exercises: interrupt handling and timing mechanisms.

Unit 5: Building and Configuring QNX Boot/OS Images [8 Hrs]

Overview of QNX boot/OS image structure. Components of a boot image: startup code, kernel, drivers, and scripts. Building and loading boot images onto target hardware. Introduction to resource managers and their implementation.

Textbooks

[1] QNX Neutrino RTOS User's Guide, QNX Software Systems.

[2] Michael Barr, "Programming for Embedded Systems", O'Reilly Media.
[3] Brian Amos, "Hands-on RTOS with Microcontrollers", Packt Publishing, 2020.
[4] Abraham Silberschatz, Peter B. Galvin, Greg Gagne, "Operating System Concepts", 9th Edition, Wiley, 2018.

Note

[1] To assess CO1, questions shall focus on fundamental understanding of QNX OS architecture, microkernel design, definitions, and conceptual recall. [2] To assess CO2, questions shall evaluate application development and debugging skills using QNX Momentics IDE. [3] To assess CO3, questions shall require analysis of process/thread management, synchronization techniques, mutexes, semaphores, and hands-on implementation scenarios. [4] To assess CO4, questions shall involve design and implementation of IPC methods, including message passing, shared memory, and comparative analysis. [5] To assess CO5 and CO6, questions shall focus on boot image construction, hardware platform configuration, POSIX standards, and comprehension of unseen problem scenarios.

Real-Time Operating System using QNX Laboratory

Lab experiments will be received from QnX.

Internet of Things

Teaching Scheme (L-T-P-S)	3-0-2-1 (L-T-P-S)
Credits	4
Course Code	–
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE 50 | ESE 50

Course Outcomes

Students will be able to:

1. Elaborate on the fundamentals of IoT such as paradigms, architecture, possibilities, and challenges.
2. Identify suitable hardware, sensors, actuators, and interfaces for IoT deployments.
3. Compare IoT protocols for communication including M2M, LPWAN, MQTT, CoAP, and IPv6/6LoWPAN.
4. Develop cloud computing models and service options for IoT including SDN, NFV, and fog computing.
5. Illustrate data analytics and security mechanisms including cryptographic algorithms for IoT systems.
6. Design an IoT application in the form of a prototype integrating sensing, communication, and cloud connectivity.

Unit 1: IoT Introduction and Fundamentals [6 Hrs]

Deciphering the term IoT; applications where IoT can be deployed; benefits/challenges of deploying IoT; IoT components: digital signal processing, data transmission, choice of channel (wired/wireless), back-end data analysis; understanding packaging and power constraints for IoT implementation.

Unit 2: Signals, Sensors, Actuators, Interfaces [8 Hrs]

Introduction to sensors and transducers; introduction to electrodes and biosensors; static and dynamic characteristics of sensors; different types of sensors; selection criteria for sensors/transducers; commercial IoT sensors/transducers; signal conditioning modules; energy and power considerations; introduction to actuators; different types of actuators; interfacing challenges; specification sheets; modules of data acquisition system; wireless sensor node structure; positioning topologies for IoT infrastructure.

Unit 3: Communication and Networking in IoT [8 Hrs]

Review of communication networks; challenges in networking of IoT nodes; Machine-to-Machine (M2M) and IoT technology fundamentals; MAC protocols for M2M; communications standards for IoT; basics of 5G cellular networks and 5G IoT communications; Low-Power Wide Area Networks (LPWAN); wireless communication for IoT: channel models, power budgets, data rates; networking and communication aspects: IPv6, 6LoWPAN, CoAP, MQTT; operating systems needs and requirements for IoT.

Unit 4: Modern Networking [6 Hrs]

SDN and NFV for IoT; Cloud Computing: introduction, history, cloud service options, cloud deployment models, business concerns in the cloud, hypervisors, comparison of cloud providers; Cloud and Fog Ecosystem for IoT; review of architecture.

Unit 5: IoT Data Analytics and Security [6 Hrs]

OLAP and OLTP; NoSQL databases; row and column oriented databases; introduction to Columnar DBMS CStore; Run-Length and Bit Vector Encoding; IoT Data Analytics. Cryptographic algorithms; analysis of lightweight cryptographic solutions; IoT security; key exchange using Elliptical Curve Cryptography; comparative analysis of cryptographic library for IoT.

Unit 6: IoT Applications [6 Hrs]

IoT applications including: Home Automation, Precision Agriculture, Smart Vehicles, Smart Grid, Industry 5.0, Healthcare.

Textbooks

[1] D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, J. Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", 1st Edition, Pearson India Pvt. Ltd., 2018.

[2] A. Bahga, V. Madisetti, "Internet of Things: A Hands-on Approach", 1st Edition, Universities Press (India) Pvt. Ltd., 2015.

Reference Books

[1] Rajkumar Buyya, Amir Vahid Dastjerdi, "Internet of Things Principles and Paradigms", Elsevier Inc., 2016.

[2] Y. Kanetkar, S. Korde, "21 Internet of Things (IoT) Experiments: Learn IoT, the Programmer's Way", 1st Edition, BPB Publications, 2018.

[3] Marinescu, Dan C., "Cloud Computing: Theory and Practice", Elsevier/Morgan Kaufmann, 2017.

[4] William Stallings, "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud", Addison-Wesley, 2015.

Note

[1] To measure CO1, questions may be of the type – define, list, identify, state, and explain fundamental concepts of IoT. [2] To measure CO2, questions may be of the type – describe, compare, classify, and illustrate various IoT hardware components, sensors, actuators, and interfaces. [3] To measure CO3, questions may focus on comparison, analysis, and selection of IoT communication protocols. [4] To measure CO4, questions may include explain, analyze, and evaluate cloud computing models and service architectures for IoT. [5] To measure CO5, questions may be of the type – analyze, interpret, and apply data analytics techniques and security mechanisms for IoT systems. [6] To measure CO6, questions will be application-oriented such as – design, develop, and propose IoT-based solutions or prototypes.

Internet of Things Laboratory

Lab Course Outcomes

Students will be able to:

1. Apply embedded programming and interfacing techniques using IoT hardware platforms and development environments to build basic IoT applications.

2. Implement sensor, actuator, and peripheral interfacing using appropriate communication protocols and enable wireless connectivity in IoT systems.

3. Develop IoT systems with node-to-node, gateway, and cloud communication, and perform data acquisition and visualization.

4. Design and realize an end-to-end IoT system integrating sensing, communication, cloud connectivity, data analysis, and automated control for real-world applications.

List of Experiments / Assignments

1. Study IoT hardware kits, development boards, and embedded programming environments using standard IDEs.

2. Understand and implement peripheral interfacing techniques for IoT systems using sensors, ADCs, and communication protocols such as SPI and I2C.

3. Interface basic input/output devices such as LEDs, switches, keypads, and displays with the development board.

4. Interface and program multiple sensors (e.g., temperature and soil moisture) simultaneously for data acquisition.

5. Interface and control actuators such as relays, motors, and buzzers based on programmed logic.

6. Interface wireless modules and establish communication for IoT systems using suitable protocols.

7. Connect IoT gateway devices to cloud platforms and implement monitoring/control using dashboards or web interfaces.

8. Implement communication protocols for data exchange between nodes, gateways, and cloud platforms.

9. Analyze sensor and cloud data for decision-making: implement closed-loop control, perform basic classification or prediction, and visualize data using appropriate tools.

10. Design and develop a complete IoT system with multiple nodes, gateway connectivity, cloud integration, data analysis, and actuator control.

Mobile Communication and Networks

Teaching Scheme	2-0-2-1 (L-T-P-S)
Credits	3
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE 50 | ESE 50

Course Outcomes

Students will be able to:

1. **1.** Explain the concepts of cellular system, mobility, multiple access and the supporting technologies.
2. **2.** Apply concepts of GSM and CDMA technologies to understand design of higher generation technologies like 4G, 5G.
3. **3.** Compare performance of various higher generation and latest technologies towards their usage as per user demands.
4. **4.** Analyse radio propagation in wireless systems with understanding of different indoor and outdoor propagation models related to diverse types of fading and path loss models.

Unit I: Cellular Concept – System Design Fundamentals [4 Hrs]

Cellular system design, frequency reuse, co-channel and adjacent channel interference, interference reduction techniques, methods to improve cell coverage, frequency management and channel assignment, capacity improvement, cell splitting concepts, handover in cellular systems.

Unit II: GSM Architecture and Interfaces [6 Hrs]

Introduction to GSM subsystems, GSM architecture, interfaces, logical channels, data encryption in GSM, mobility management, call flows, spectral efficiency with TDMA and FDMA. Self-Study: functions and working of GSM blocks (MS, BSS, switching subsystems, HLR, VLR, EIR, echo canceller).

Unit III: Code Division Multiple Access (CDMA) [8 Hrs]

CDMA technology, RAKE receiver, IS-95 system architecture, air interface, forward and reverse link, physical and logical channels, call processing, soft handoff, GSM-CDMA comparison, spectral efficiency. Evolution to CDMA 2000, 3G example WCDMA. Self-Study: comparison and usage of GSM and CDMA technologies.

Unit IV: Mobile Radio Propagation [8 Hrs]

Large-scale path loss and small-scale fading. Free space propagation, ground reflection (two-ray) model, diffraction, scattering. Practical link budget design, outdoor and indoor propagation models, building penetration. Small-scale fading and multipath: impulse response, multipath parameters, types of fading: flat/frequency selective, fast/slow fading. Self-Study: large-scale path loss vs. small-scale fading (MATLAB implementation).

Unit V: Higher Generation Cellular Standards [8 Hrs]

Evolution from 2.5G to 4G. GPRS architecture, node functionalities, interfaces, channels, call setup procedures. EDGE (2.75G). 3G network architecture and services. LTE enabler technologies: OFDMA, SC-FDMA, MIMO. Adaptive antenna techniques, QoS requirements. LTE network architecture, interfaces, protocol stack, channels, call procedures. Self-Study: comparative study of 2G, 2.5G, 3G, 4G.

Unit VI: Introduction to 5G [6 Hrs]

Drivers for 5G, 5G roadmap and vision, enabler technologies, key building blocks. 5G numerology, O-RAN, CP-OFDM, network slicing, 5G protocols, frequency bands and modulation techniques. Recent trends in telecommunications. Self-Study: AI and 5G integration for 6G application development.

Textbooks

[1] Vijay K. Garg, "Wireless Communication and Networking", Elsevier/Morgan Kaufmann, 2012.

[2] Vijay K. Garg and J. E. Wilkes, "Principles and Applications of GSM", Pearson Education.
[3] Vijay K. Garg, "IS-95 CDMA and CDMA 2000", Pearson Education.
[4] T. S. Rappaport, "Wireless Communications: Principles and Practice", PHI, 2nd Edition, 2006.

Reference Books

[1] T. L. Singal, "Wireless Communications", Tata McGraw Hill, 2010.
[2] William C. Y. Lee, "Mobile Cellular Telecommunication Systems", Tata McGraw Hill, 2nd Edition.
[3] Jonathan Rodriguez, "Fundamentals of 5G Mobile Networks", John Wiley & Sons, 2015.

Note

[1] To measure CO1, questions may be of the type: define, identify, state, match, list, name, etc.
[2] To measure CO2, questions may be of the type: explain, describe, illustrate, evaluate, give examples, compute, etc.
[3] To measure CO3, questions will be based on applications of core concepts.
[4] To measure CO4, questions may be of the type: true/false with justification, theoretical fill in the blanks, theoretical problems, prove implications or corollaries of theorems, etc.

Mobile Communication and Networks – Laboratory

Lab Course Outcomes

Students will be able to:

1. Explain GSM, GPRS, LTE, and 5G architectures.
2. Analyse wireless channel and system performance.
3. Simulate and evaluate mobile networks.
4. Design real-world use cases using modern communication technologies.

List of Experiments / Assignments

1. Study fundamentals of cellular systems and implement frequency reuse concept using Python code.
2. Study of wireless channel modelling – AWGN channel modulation using Python code.
3. BER performance analysis of modulation schemes using MATLAB.
4. Study GSM and CDMA network concepts (call management, call setup, call release, handover, GSM security, power control, RAKE receiver, CDMA capacity) using wireless communication software.
5. Design any two using GNU Radio (one Tx-Rx pair is compulsory): AM Tx-Rx, Narrow-band FM Tx-Rx, SSB Tx-Rx, BPSK Tx-Rx, filter design (LPF/HPF/BPF).
6. Study 4G kit sections in detail; observe fault insertion; study AT commands and develop an APP using AT commands.
7. Study of DSSS technique for CDMA; observe effect with variation of PN codes, chip rate, spreading factor, processing gain.
8. AT command concepts using Proteus software: interface GSM 900 modem to serial monitor; interface to Arduino; perform SMS send/receive.
9. Experience 4G connectivity features using 4G modules and Arduino.
10. 5G Experience: 5G core and RAN; 5G camera integration (face/object detection); 5G IoT sensors with RUTx50 IoT gateway.

Data Communication and Networking

Teaching Scheme	2-0-2-1 (L-T-P-S)
Credits	3
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE 50 | ESE 50

Course Outcomes

Students will be able to:

1. Explain the fundamental underlying principles of layered network architecture.
2. Comprehend the congestion and routing protocols at various network layers.
3. Design a network considering the QoS parameters.
4. Analyse the performance of various communication protocols and networks.
5. Implement flow control and congestion control in the network.
6. Illustrate the interfacing of components in existing networks.

Unit I: Basics of Data Communication [4 Hrs]

Introduction to protocols and standards, layered network protocol architectures, internet devices, topology, LAN, MAN, WAN. Data transmission – types, media and impairments, data rate and throughput.

Unit II: Data Link Layer [4 Hrs]

Framing, flow control, error control mechanisms, error detection. Multiple access protocols: CSMA/CA, CSMA/CD. ARP, RARP.

Unit III: Network Layer [6 Hrs]

IPv4, IPv6, IP addresses and subnetting, transitioning from IPv4 to IPv6. ICMP, IGMP, DHCP, NAT.

Unit IV: Transport Layer [6 Hrs]

Connectionless transport – UDP. Connection-oriented transport – TCP. Congestion control mechanisms, flow control mechanisms, Quality of Service (QoS).

Unit V: Wireless Networks [4 Hrs]

Bluetooth, Zigbee, Ethernet, Wi-Fi, Wireless Sensor Networks, Ad Hoc networks. Circuit and packet switching.

Unit VI: Application Layer [4 Hrs]

The Web and HTTP, file transfer (FTP), Domain Name System (DNS).

Textbooks

[1] J. F. Kurose and K. W. Ross, "Computer Networking – A Top-Down Approach Featuring the Internet", Pearson Education, 5th Edition.
[2] L. Peterson and B. Davie, "Computer Networks – A Systems Approach", Elsevier/Morgan Kaufmann, 5th Edition.

Reference Books

[1] B. A. Forouzan, "Data Communications and Networking", Tata McGraw Hill, 4th Edition.
[2] Andrew Tanenbaum, "Computer Networks", Prentice Hall.
[3] William Stallings, "Data and Computer Communications", Prentice Hall.
[4] S. Keshav, "An Engineering Approach to Computer Networking", Pearson Education.
[5] D. Comer, "Computer Networks and Internet/TCP-IP", Prentice Hall.

Note

[1] To measure CO1, questions may be of the type: define, identify, state, match, list, name, etc.
[2] To measure CO2, questions may be of the type: explain, describe, illustrate, evaluate, give examples, compute, etc.
[3] To measure CO3, questions will be based on applications of core concepts.
[4] To measure CO4, questions may be of the type: true/false with justification, theoretical fill in the blanks, theoretical problems, prove implications or corollaries of theorems, etc.
[5] To measure CO5, some questions may be based on self-study topics and comprehension of unseen passages.

Data Communication and Networking – Laboratory

Lab Course Outcomes

Students will be able to:

1. Understand the network simulator environment; visualize and observe performance of network topologies.
2. Design and implement network applications such as data transmission and file transfer.
3. Understand TCP/IP configuration for Windows and Linux.
4. Understand various routing protocols and internetworking algorithms.
5. Implement packet capture and traffic analysis using Wireshark.

List of Experiments / Assignments

1. Study of NS2 Simulator and installation of NS2.
2. Transfer of packets between two nodes: a) UDP, b) TCP.
3. Forward packets via single router: a) UDP, b) TCP.
4. Transfer packets using 4 nodes, assign colour to nodes and traffic for UDP and TCP.
5. Transfer packets using 5 nodes – router broadcasting UDP/TCP packets.
6. Transfer packets using 6 nodes – Queue Management for UDP and TCP.
7. Transfer UDP and TCP packets on Ring Topology Network.
8. Distance Vector Routing – Throughput and X-graph analysis.
9. Implement Dijkstra's shortest path algorithm on NS2 (greedy approach).
10. Determine shortest path using C/C++/Python.
11. Implement sliding window protocol in C/C++/Python.
12. Capture packets using Wireshark and analyse them at all layers of the network.

Digital IC Design

Teaching Scheme	3-0-2-1 (L-T-P-S)
Credits	4
Examination Scheme	Theory: TA 20 | MSE 30 | ESE 50 Laboratory: CIE 50 | ESE 50

Course Outcomes

Students will be able to:

1. Illustrate, infer, and interpret the importance of CMOS manufacturing process, energy band diagrams of MOS, scaling technology and secondary effects in MOSFETs.

2. Model MOS transistors, small geometry effects and interconnect wire for performance analysis; evaluate impact of technology scaling on robustness, performance, and energy/power dissipation of CMOS inverter.

3. Design, formulate and estimate combinational logic circuits optimized for area, speed, power, glitch-free operation and reduced supply voltages.

4. Design and analyse CMOS sequential circuits by applying bi-stability principles and conducting Static Timing Analysis (STA) to ensure robust data storage and timing closure.

5. Design and optimize high-density SRAM systems by evaluating trade-offs between 6T, 8T, and 10T cell topologies and integrating row/column peripheral circuitry.

6. Evaluate and classify fundamental types of semiconductor memory by analysing architectural differences, volatility, and performance metrics to select appropriate storage solutions.

Unit I: Introduction to VLSI [8 Hrs]

Evolution of VLSI. Energy band diagram of MOS capacitor. Operation and working of MOS transistor. Derivation of threshold voltage equation. Technology scaling and roadmap, scaling issues, secondary effects in MOSFETs. Standard 4-mask NMOS fabrication process. High-k dielectric, metal gate technology, FinFET. IC layout design and tools: CMOS n-well process design rules, stick diagram, ASIC design flow.

Unit II: Electrical Wire Models and CMOS Inverter [6 Hrs]

MOS capacitances. Modelling of MOS transistors using SPICE Level I and II equations, BSIM models. Ideal wire, lumped RC model, distributed RC model, Elmore delay model. Quality metrics of digital design: cost, functionality, robustness, power, and delay. CMOS inverter: switching threshold, noise margin, dynamic behavior, propagation delay, RC delay, dynamic and static power consumption, low power design criteria.

Unit III: Combinational Logic [8 Hrs]

Static CMOS design, pass transistor logic, transmission gate logic. Logic effort delay, standard cell design, sizing of gates. Estimating and optimizing delay in standard cell and clock distribution networks. Dynamic logic: speed and power dissipation, cascading dynamic gates.

Unit IV: Sequential Logic [4 Hrs]

Timing constraints in CMOS latches and registers. Bi-stability principle, MUX-based latches, static SR flip-flops, master-slave edge-triggered register, dynamic latches and registers. STA and max/min delay constraints for flip-flops.

Unit V: Array Subsystems [3 Hrs]

6T/8T/10T SRAM cell design, row circuitry, column circuitry, SRAM cell array design discussions from technical papers.

Unit VI: Introduction to Storage Devices for In-Memory Computation [4 Hrs]

Mass data storage devices and technology. Flash memory – fundamental operating physics, read/write mechanisms. Performance metrics comparison for emerging memory devices: FeRAM, MRAM, RRAM, PCM, nano-magnetic and spintronic devices, superconducting electronic memory for quantum information processing (QIP).

Textbooks

[1] N. Weste and D. Harris, "CMOS VLSI Design: A Circuits and Systems Perspective", 4th Edition, Pearson.
[2] J. M. Rabaey, A. Chandrakasan, B. Nikolic, "Digital Integrated Circuits: A Design Perspective", Pearson.

Reference Books

[1] Sung Mo Kang and Yusuf Leblebici, "CMOS Digital Integrated Circuits", Tata McGraw Hill.
[2] Baker, Li and Boyce, "CMOS Circuit Design, Layout, and Simulation", Wiley, 2nd Edition.
[3] Neil E. Weste and Kamran Eshraghian, "Principles of CMOS VLSI Design", Pearson Education.

Note

[1] To measure CO1, questions may be of the type: illustrate, interpret, infer, describe, calculate, etc.
[2] To measure CO2, questions may be of the type: explain, describe, illustrate, evaluate, give examples, compute, etc.
[3] To measure CO3, questions may be of the type: design, formulate, estimate, etc.
[4] To measure CO4, questions may be of the type: design and analyse, etc.
[5] To measure CO5, questions may be of the type: design, optimize and integrate, etc.
[6] To measure CO6, questions may be of the type: evaluate, analyse, classify, etc.

Digital IC Design – Laboratory

Lab Course Outcomes

Students will be able to:

1. Interpret the behavior of MOS transistors using SPICE tools.
2. Perform DC and transient analysis of CMOS inverter.
3. Carry out physical design and verification of CMOS inverter.
4. Design digital circuits, clock and buffer distribution networks for standard cell design according to given specifications.

List of Experiments / Assignments

1. DC analysis of NMOS and PMOS transistors using NGSPICE. Estimate ION, IOFF, subthreshold swing (S) and lambda for 180nm/130nm/90nm transistors.
2. Threshold voltage (Vth) analysis of NMOS and PMOS transistors using various methods.
3. DC and transient analysis of CMOS inverter using NGSPICE. Estimate VIL, VIH, VOL, VOH, NML, NMH (DC) and tpHL, tpLH (transient).
4. Design of 5-stage and 7-stage ring oscillators using NGSPICE. Estimate oscillation frequency (fosc).
5. Schematic design of CMOS inverter and its DC/transient analysis using Cadence EDA Tool.
6. Schematic to symbol generation using Cadence EDA Tool.
7. Schematic to layout of CMOS inverter using Cadence EDA Tool.
8. Post-layout simulation of CMOS inverter and parasitic extraction.
9. Design of basic gates and/or combinational circuits using Cadence EDA Tool.
10. Design of a 6-transistor SRAM cell using Cadence EDA Tool. Estimate CR, PR, read SNM and write SNM.
11. Design of sequential circuits (D flip-flop) and simple clock/buffer distribution networks for standard cell design using Cadence EDA Tool.

Minor Project

Teaching Scheme	0-0-6-2 (L-T-P-S)
Credits	6
Examination Scheme	Project: CIE 60 | ESE 40 | Total: 100

Course Outcomes

Students will be able to:

1. Analyse a real-world or industry-grade problem, conduct a requirements study, and formulate clear project objectives and deliverables.

2. Design and develop a complete technical solution by applying competencies gained across the programme.

3. Execute all phases of the project lifecycle – design, development, testing, validation, and documentation – to engineering standards.

4. Engage professionally with industry mentors, incorporate structured feedback, and demonstrate understanding of industry practices and constraints.

5. Communicate technical work effectively through written reports, a mid-semester review, and an end-semester evaluation before internal and external examiners.

6. Demonstrate professional integrity by adhering to ethical practices, confidentiality obligations (NDA where applicable), and intellectual property norms.

CIE – Continuous Internal Evaluation (50 Marks)

Guide Continuous Evaluation – regularity, initiative, iterative improvement (15 marks).

Project Registration Form and Weekly Progress Log (5 marks).

Mid-Semester Review by industry mentor or internal panel (15 marks).

Internal Working Demonstration – completeness and functionality (15 marks).

ESE – End-Semester Evaluation (50 Marks | Internal 60% : External 40%)

Technical Presentation (20-min talk + live demo) – Internal: 15 | External: 10 = 25 marks. Final Project Report – Internal: 10 | External: 5 = 15 marks. Viva Voce – individual technical depth – Internal: 5 | External: 5 = 10 marks.

Textbooks

[1] IEEE format guide for technical reports (minimum 50 pages).

Reference Books

[1] Project management and documentation guidelines issued by the Programme Coordinator.

Note

[1] Registration for the Minor Project is compulsory in Semester VI.

[2] The MoU between the institution and the industry partner must be executed before Industry Track teams commence work.

[3] The ESE panel must have at least one internal examiner and one external examiner. The project guide does not serve as an examiner.

[4] Mid-semester review marks must be submitted by the guide or mentor to the Programme Coordinator by Week 9.

[5] AI-generated content (code, text, images) must be declared and cited. Undisclosed use is treated as plagiarism and results in disqualification from ESE.