Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

(Course Structure and Curriculum of B.Tech. (ECE) Third Year)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Course Structure and Curriculum of B.Tech. (ECE) Third Year

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Course Structure of B. Tech (ECE) Third Year
Third Year

Semester – V

S. No.	Course Domain	Course Title	Credits*			Total Credits
			L	T	P	
1.	DSC-13	Control Systems Engineering	3	0	1	4
2.	DSC-14	Digital Signal Processing	3	0	1	4
3.	DSC-15	Analog Communication Systems	3	0	1	4
4.	DSE-3	Select a course from the specified list of DSE - 3				4
5.	GE-5	Select a course from the specified list of GE - 5				4
6.	SEC or IAPC	Choose one SEC or Internship/Apprenticeship/Project/Community Outreach (IAPC)				2
Total Credits						22

Semester – VI

S. No.	Course Domain	Course Title	Credits*			Total Credits
			L	T	P	
1.	DSC-16	Hands – On CMOS VLSI Design	3	0	1	4
2.	DSC-17	Embedded Systems and Applications	3	0	1	4
3.	DSC-18	Digital Communication Systems	3	0	1	4
4.	DSE-4	Select a course from the specified list of DSE - 4				4
5.	GE-6	Select a course from the specified list of GE - 6				4
6.	SEC or IAPC	Choose one SEC or Internship/Apprenticeship/Project/Community Outreach (IAPC)				2
Total Credits						22

**Credits*
L (01 Credit) is equivalent to 01 contact hour per week
T (01 Credit) is equivalent to 01 contact hour per week
P (01 Credit) is equivalent to 02 contact hours per week

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Pool of DSEs offered by the Department

S. No.	Semester	DSE	Course Title
1.	V	DSE - 3	Spread Spectrum Communication
2.			Network Technologies and Interfacing
3.			Digital Image Processing
4.	VI	DSE - 4	Wireless Sensor Networks
5.			Artificial Intelligence in Electronics
6.			Coding and Data Compression Techniques
7.			Electromagnetic Compatibility: Principles and Applications

List of SECs offered by the Department

S. No.	Semester	Course Title
1.	V	IAPC
2.	VI	Radar and Antenna Workshop

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Specialization and Minors offered by the Department

Semester	DSE/ GE	ECE Minor (Open only for CSE/ EE)	Specializations for ECE/ Minors for EE and CSE			
			Telecommunication Networks	VLSI Technology and System Design	IoT System Design	Computer Vision
V	DSE-3/ GE-5	Communication Architecture	Wireless Communication and Mobile Networks	Advanced Digital VLSI Circuits and Physical Design	Introduction to Security of Cyber-Physical Systems	Intelligent Imaging
VI	DSE-4/ GE-6	Antenna and Wave Propagation	Satellite Communication	CMOS Analog IC Design	Ubiquitous Sensing, Computing and Communication	Deep Learning for Image Analysis
		Elements of Wireless Communication	Cognitive Radio & Networks	Current Mode Analog VLSI Circuits	Introduction to Embedded Systems for IoT	Deep & Reinforcement Learning

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Core (DSC) Courses for B. Tech. (ECE) – Semester V

Control Systems Engineering (DSC - 13)

(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Control Systems Engineering	4	3	0	1	Signals and Systems

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- Understand core concepts of linear, nonlinear, time-invariant, and time-varying control systems.
- Develop skills in modeling physical systems using differential equations, transfer functions, and block diagrams.
- Analyze system behavior in both time and frequency domains.
- Apply stability criteria and root-locus techniques for system analysis.
- Design and tune controllers (P, PI, PID) and compensators (lag, lead, lead-lag).
- Utilize state-space methods to analyze controllability and observability.

Course Outcomes:

After completing the course, the students should be able to:

1. Classify and model various control systems.
2. Evaluate time response and steady-state errors.
3. Assess stability using Routh-Hurwitz and root-locus methods.
4. Interpret frequency response plots (Bode, Nyquist) for stability and performance.
5. Design appropriate controllers and compensators to meet specifications.
6. Apply state-space techniques to analyze and design advanced control systems.

Unit - I

Introduction to Control System: Linear, Non Linear, Time Varying and Linear Time Invariant System, Mathematical Modelling of Electrical and Mechanical Systems, Differential Equations of Physical Systems, Transfer Functions, Block Diagram and Signal Flow Graphs. Feedback and Non feedback Systems. Reduction of Parameter Variations by use of Feedback Control Over System Dynamics. Feedback Control of Effects of Disturbance.

Control Systems and Components DC and AC Servomotors, Synchro Error Detector, Tacho Generator and, Stepper Motors etc.

Unit - II

Time Response Analysis: Standard Test Signals, Time Response of First-order Systems, Time Response of Second-Order Systems, Steady-State Error and Error Constants, Effect of Adding a Pole/ Zero to a System, Design Specifications of Second-Order Systems and Performance Indices. The Concept of Stability, Necessary Conditions for Stability, Hurwitz Stability Criterion, Routh Stability Criterion and relative Stability Analysis. The Root Locus Concept, Construction of Root Loci, Root Contours, Systems with Transportation Lag, Sensitivity of the Roots of the Characteristic equation.

Unit - III

Frequency Response Analysis: Correlation Between Time and Frequency Response, Bode Plots, Polar Plots. Stability in Frequency Domain: Mathematical Preliminaries, Nyquist Stability Criterion, Calculation of Gain Margin and Phase Margin, Assessment of Relative Stability Using Nyquist Criterion and Closed-Loop Frequency Response.

Unit - IV

P, PI and PID Control Action and Their Effect, Compensator and Controller Design: Design of Lag, Lead, Lead Lag, Feedback compensator, Preliminary Considerations of Classical Design, Realization of Basic Compensators, Cascade Compensation in Time Domain Cascade Compensation in Frequency Domain, Tuning of PID Controllers.

Control Systems Analysis in State Space: State-Space Representations of Transfer-Function Systems, Solving the Time-Invariant State Equation, Controllability, Observability.

Suggested Readings

1. Control Systems Engineering by Norman S. Nise (Wiley)
2. Modern Control Engineering by Katsuhiko Ogata (Pearson)
3. Modern Control Systems by Richard C. Dorf and Robert H. Bishop (Pearson)
4. Feedback Control of Dynamic Systems by Gene Franklin, J. Da Powell, and Abbas Emami-Naeini (Pearson)
5. Control System Engineering by I J Nagrath (New Age Publishers)
6. Automatic Control Systems by B. C. Kuo (Wiley)

List of Experiments (Hardware and Software using Matlab/Simulink)

1. Transient and Steady-State Analysis: analyze responses of first- and second- order systems using MATLAB.
2. PID Controller Design: Implement and tune PID controllers.
3. Utilize tuned PID controller for DC motors using MATLAB.
4. PID Controller Design: Implement PID control for a DC-DC converter using Arduino.
5. State-Space Analysis and Design: Perform state feedback control design and implementation using Raspberry Pi.
6. Analysis of a SISO and MIMO system
7. Frequency Response Analysis: Generate and interpret Bode, Nyquist, and Nichols plots using MATLAB.
8. Frequency Response Analysis using Bode Plot of a given transfer function using Hardware.
9. Design of Lag Compensator and Lead Compensator using MATLAB and Hardware.
10. Design of Lag and Lead Compensator using MATLAB and Hardware.

(Note: Course instructor may add/update new experiments in addition to the above suggested practical exercises.)

Digital Signal Processing (DSC - 14)

(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Digital Signal Processing	4	3	0	1	Signals and Systems

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To understand the fundamentals of digital signal processing and its applications in real-world systems.
- To learn techniques for analyzing and designing digital filters and systems.
- To gain hands-on experience with modern tools and platforms for implementing DSP algorithms.
- To explore advanced topics such as adaptive filtering, multi-rate signal processing.
- To equip students with skills to solve industry-relevant problems using DSP techniques.

Course Outcomes:
After completing the course, the students should be able to:
1. Analyze and process discrete-time signals.
2. Design and implement digital filters for various applications.
3. Apply multi-rate signal processing techniques in communication systems.
4. Work with adaptive signal processing algorithms for noise cancellation and predictive analytics.

Unit - I
Fundamentals of DSP: Introduction to DSP and its applications, Discrete-Time Signals and Systems: Sampling, quantization, aliasing, Discrete-Time Fourier Transform (DTFT) and its properties, Linear Time-Invariant Systems: Convolution and stability analysis. Wavelet Transform: Continuous wavelets, discrete wavelets and its advantages.

Unit - II
Digital Filter Design (for both FIR and IIR): Design of Linear Phase FIR Filters: Window methods, frequency sampling methods, IIR Filters: Butterworth, Chebyshev, and elliptic filters, Filter realization: Direct form, cascade form, and parallel form, Applications in image compression.

Unit - III
Advanced DSP Techniques: Fast Fourier Transform (FFT) and its applications, Multi-rate Signal Processing: Decimation, interpolation, and polyphase structures, Adaptive Signal Processing: Adaptive Filtering – Kalman Filters, Wiener Filters, LMS and RLS algorithms.

Unit - IV
Real-Time DSP Applications: Real-Time Processing: Signal reconstruction, anti-aliasing filters and its applications for signal processing. Introduction to Digital Signal Processors: Fixed Point and Floating-Point processors, architectures. TMS 320C54XX and TMS320C67XX Architecture, Memory, Addressing Modes.

Suggested Readings

1. Digital Signal Processing by John G. Proakis and Dimitris K. Manolakis (Pearson)
2. Discrete-Time Signal Processing by Alan V. Oppenheim and Ronald W. Schafer (Pearson)
3. Digital Signal Processing: Principles, Algorithms, and Applications by John G. Proakis (Pearson)
4. DSP First by James H. McClellan, Ronald W. Schafer, and Mark A. Yoder (Pearson)
5. Digital Signal Processing: A Computer-Based Approach by Sanjit K. Mitra (Tata McGraw Hill)
6. Adaptive Filter Theory by Simon Haykin (Pearson)

List of Experiments

1. Sampling and Reconstruction of Signals
   Requirements - MATLAB/Python, DAC-ADC Converter, Oscilloscope.
   Objective - Understand the effects of under-sampling and aliasing.
2. Design a digital Differentiator using a Hamming Window
3. Design a digital Hilbert Transformer using a Hamming Window
4. Wavelet Transform for Signal Denoising
   Requirements - MATLAB Wavelet Toolbox, noisy biomedical signals.
   Objective - Analyze the denoising capability of wavelet transforms.
5. Implementation of FIR Filters
   Requirements - MATLAB, Python (Scipy), Filter Design Toolbox.
   Objective - Design filters and analyze their performance in suppressing noise.
6. Implementation of IIR Filters
   Requirements - MATLAB, Python (Scipy), Filter Design Toolbox.
   Objective - Design filters and analyze their performance in suppressing noise.
7. Adaptive Noise Cancellation
   Requirements - DSP Kit (e.g., TMS320C6713), MATLAB, microphone, and speakers.
   Objective - Implement LMS algorithm for noise removal in audio.
8. Adaptive Filtering Using RLS Algorithm
   Requirements - DSP Kit (e.g., TMS320C6713), MATLAB, noisy communication signals.
   Objective - Implement RLS for dynamic noise cancellation.
9. Multi-rate Signal Processing
   Requirements - MATLAB, Python, signal generator, decimation/interpolation circuits.
   Objective - Explore decimation and interpolation in multi-rate systems.
10. Filter Implementation using Fixed- and Floating-Point processors.

(Note: Course instructor may add/update new experiments in addition to the above suggested practical exercises.)

Analog Communication Systems (DSC - 15)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Analog Communication Systems	4	3	0	1	Electronic Devices and Circuits, Signals and Systems, Network Analysis and Synthesis

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Introduce the core principles and analysis techniques for analog communication systems.
- Develop the skills to design, implement, and test analog modulation, transmission, and reception circuits.
- Expose students to emerging analog communication technologies, including SDR (Software-Defined Radio), cognitive radio, quantum communication influences, IoT integration, and cutting-edge modulation techniques.
- Enhance problem-solving and critical-thinking abilities through extensive hands-on lab work, projects, and case studies.

Course Outcomes:
After completing the course, the students should be able to:
1. Understand and analyze fundamental and advanced analog modulation techniques.
2. Design and simulate analog communication circuits using modern EDA tools and SDR platforms.
3. Integrate analog front-ends with IoT and edge computing systems, addressing real-world constraints.
4. Explore and critically assess emerging research fields such as cognitive radio, quantum-influenced communication, and OAM-based modulation.
5. Implement complex analog experiments, interpret results, and propose improvements, preparing for contemporary industry requirements.

Unit - I
Fundamentals of Analog Communication: Basics of Communication Systems: Representation of signals, frequency spectrum, and noise considerations, Amplitude Modulation (AM): DSB, SSB, VSB – generation, demodulation, bandwidth, power considerations, Angle Modulation: Frequency Modulation (FM) and Phase Modulation (PM) – principles, Carson's rule, generation, and detection, Signal-to-Noise Ratio (SNR), Noise Figure, and Noise in Communication Channels.

Unit - II
Advanced Analog Modulation & System Architecture: Pulse Modulation: PAM, PWM, PPM – characteristics, generation, and detection, Superheterodyne Receiver Architecture: Mixers, Local Oscillators, IF stages, Software Defined Radio (SDR) Basics: Architecture, reconfigurable front-ends, FPGA-based implementations, IoT Integration: Analog front-ends for low-power IoT devices, sensor signal conditioning, Introduction to Advanced RF Components: Filters, low-noise amplifiers (LNAs), impedance matching.

Unit - III

Transmission, Reception & System Integration: Transmission Lines and Wave Propagation: Parameters, reflections, impedance matching techniques, Multiplexing Techniques: Frequency Division Multiplexing (FDM), Time Division Multiplexing (TDM) in analog systems, Antennas in Analog Communications: Basic antenna theory, analog link budgeting, Analog Communication Over Fiber: Basic principles, intensity modulation, optical detectors, Introduction to Advanced RF Design: oscillator, mixer, and amplifier design.

Unit - IV

Emerging Trends and Research Directions: Cognitive Radio Systems: Spectrum sensing, dynamic spectrum allocation, energy-efficient analog signal processing, Satellite and CubeSat Analog Systems: Analog link design, small-satellite telemetry, and OAM (Orbital Angular Momentum) modulation for high capacity, Noise-Shaping and Advanced Filtering: Techniques to improve SNR and signal integrity.

Suggested Readings

1. Principles of Communication Systems by Herbert Taub and Donald Schilling
2. Communication Systems by Simon Haykin
3. Modern Digital and Analog Communication Systems by B.P. Lathi and Zhi Ding
4. RF Microelectronics by Behzad Razavi
5. Cognitive Radio Communications and Networks by A. Wyglinski, M. Nekovee

List of Experiments (Hardware on Breadboard / Software using NI Multisim)

1. Amplitude Modulation and Demodulation
   Objective: Implement and analyze standard AM.
2. Frequency Modulation using Varactor Diodes
   Objective: Generate and observe FM signals.
3. Generation and Analysis of Pulse Amplitude Modulation (PAM)
   Objective: To generate and study the characteristics of Pulse Amplitude Modulation (PAM) and understand its applications in communication systems.
   Requirements: Hardware
4. Generation and Analysis of Pulse Width Modulation (PWM)
   Objective: To generate and analyze Pulse Width Modulation (PWM) signals and observe how the pulse width varies with the amplitude of the modulating signal.
   Requirements: Hardware
5. Generation and Analysis of Pulse Position Modulation (PPM)
   Objective: To generate and analyze Pulse Position Modulation (PPM) signals and understand the impact of modulating signal amplitude on pulse position.
   Requirements: Hardware
6. Demodulation of PAM, PWM, and PPM Signals
   Objective: To demodulate the PAM, PWM, and PPM signals and recover the original modulating signal using appropriate techniques.
   Requirement: Hardware
7. Superheterodyne Receiver Simulation
   Objective: Design and simulate superheterodyne receiver stages.
   Requirements: MATLAB/Simulink.
8. SDR-Based AM/FM Modulation
   Objective: Implement modulation schemes using SDR (GNU Radio + USRP).
   Requirements: SDR hardware (e.g., USRP), GNU Radio.

9. Analog Signal over Fiber
   Objective: Transmit and receive analog signals over fiber optics.
   Requirements: Fiber optic trainer kit, Laser diode, Photodiode.
10. RF Amplifier Design and Analysis
    Objective: Design an RF amplifier and analyze gain, noise figure.
    Requirements: Keysight ADS, RF components.
11. QAM under Noise Conditions
    Objective: Generate QAM signals and analyze their performance under noise.
    Requirements: Signal Generator, GNU Radio, Spectrum Analyzer.
12. Noise Cancellation with Analog Filters
    Objective: Implement active filters to enhance SNR.
13. Orbital Angular Momentum (OAM) Modulation Setup
    Objective: Generate and analyze OAM-modulated analog signals.
    Requirements: OAM modulation kit, Signal Generator, MATLAB.
14. Satellite Analog Link Simulation (CubeSat scenario)
    Objective: Simulate an analog downlink for a CubeSat telemetry system.
    Requirements: CubeSat communication module simulator, MATLAB.
15. Noise-Shaping Filter Design
    Objective: Implement filters to shape noise spectrum and improve SNR.
    Requirements: Filter Design Toolkit, Multisim.

(Note: Course instructor may add/update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Elective (DSE) Courses for B. Tech. (ECE) – Semester V

Spread Spectrum Communication (DSE – 3)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Spread Spectrum Communication	4	3	0	1	Signals and Systems, Mathematics - II

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To understand the fundamental concepts of Spread Spectrum (SS) communication systems.
- To design and analyze Spread Spectrum techniques and their applications in modern communication systems.
- To develop practical skills in implementing SS systems using hardware and software tools.
- To explore advanced topics such as Code Division Multiple Access (CDMA), frequency hopping, and their role in 5G and satellite communication systems.

Course Outcomes:
After completing the course, the students should be able to:
1. Explain the principles of Spread Spectrum communication and its advantages over conventional methods.
2. Design and simulate basic SS communication systems using MATLAB/Simulink or Python.
3. Implement hardware-based SS systems using tools like SDR, FPGA, and microcontrollers.
4. Analyze the role of Spread Spectrum techniques in CDMA, GPS, 5G, and secure communication.
5. Integrate advanced Spread Spectrum concepts into real-world communication challenges.

Unit - I
Fundamentals of Spread Spectrum Communication: Introduction to Spread Spectrum Systems, Definition, history, and applications, Benefits: interference rejection, multipath fading reduction, and security, Types of Spread Spectrum Techniques, Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), Spreading Codes, Pseudorandom noise (PN) sequences: properties and generation, Gold codes, Walsh codes, and Kasami sequences.

Unit - II
System Design and Performance Analysis: DSSS System Design, Transmitter and receiver architecture, Spreading and despreading processes, Processing gain and Bit Error Rate (BER) analysis, FHSS System Design, Hopping patterns and synchronization, BER analysis and jamming resistance, Multipath and Interference Effects, Rake receivers and diversity combining techniques, Interference rejection capabilities.

Unit - III

Applications of Spread Spectrum: Code Division Multiple Access (CDMA), Principles, Walsh codes, and orthogonality, Forward and reverse link analysis, Global Positioning System (GPS), Spread Spectrum in GPS signal structure, Synchronization and tracking, Secure Communications, Anti-jamming and eavesdropping resilience, Military and commercial applications.

Unit - IV

Advanced Topics and Emerging Trends: Spread Spectrum in 5G and Beyond, Role in massive MIMO and millimeter-wave communication, Integration with OFDM and NOMA, Cognitive Radio and Spectrum Sharing, Spread Spectrum for dynamic spectrum allocation, Hardware Implementation, SDR-based SS communication systems, FPGA and microcontroller-based DSSS/FHSS systems.

Suggested Readings

1. Digital Communication by John G. Proakis and Masoud Salehi.
2. Wireless Communications: Principles and Practice by Theodore S. Rappaport.
3. Spread Spectrum Communications Handbook" by Marvin K. Simon, Jim K. Omura, Robert A. Scholtz.
4. Principles of Communication Systems by Herbert Taub and Donald Schilling.
5. CDMA: Principles of Spread Spectrum Communication by Andrew J. Viterbi

List of Experiments (Software using NI Multisim/Matlab/Simulink)

1. Generation and Analysis of PN Sequences.
2. Simulation of DSSS System.
3. Frequency Hopping Spread Spectrum Simulation.
4. BER Analysis of DSSS under Noise
5. Implementation of DSSS Transmitter and Receiver on SDR (Tools: GNU Radio, USRP SDR)
6. Design of a Simple CDMA System.
7. Implementation of a Rake Receiver.
8. Spread Spectrum in GPS Signal Simulation
9. Study and implementation of FHSS.
10. SDR-based Real-Time Spread Spectrum Communication (Tools: USRP, GNU Radio)

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Network Technologies and Interfacing (DSE – 3)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Network Technologies and Interfacing	4	3	0	1	Fundamentals of Computer Programming

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To understand the principles of computer networks and interfacing systems.
- To learn about modern communication protocols, IoT technologies, and network design principles.
- To gain hands-on experience with hardware-software interfacing and network performance analysis.
- To explore advanced topics such as 5G technologies, network security, and AI-driven network optimization.

Course Outcomes:
After completing the course, the students should be able to:
1. Design and implement basic and advanced computer networks using industry-standard tools.
2. Interface hardware and software components to create efficient communication systems.
3. Analyze and optimize network performance for IoT and 5G applications.
4. Develop solutions for network security and manage modern interfacing challenges using AI/ML tools.

Unit - I
Fundamentals of Network Technologies: Network architectures: OSI and TCP/IP models, Communication protocols: IP, UDP, TCP, HTTP, FTP, Overview of IPv4 and IPv6, Switching techniques: Circuit switching, packet switching, and virtual circuits, Wireless and wired network design, Introduction to Network Simulation Tools (NS2, NS3, Wireshark). Network security: Firewalls, VPNs, and intrusion detection systems.

Unit - II
Interfacing Systems and IoT: Basics of hardware-software interfacing, Embedded systems for networked communication, IoT protocols: MQTT, CoAP, Zigbee, LoRaWAN, Interfacing microcontrollers (e.g., Arduino, STM32) with sensors and actuators, Networked IoT systems using Raspberry Pi, Use of cloud platforms (AWS IoT, Azure IoT) for real-time monitoring.

Unit - III
Advanced Network Technologies: Fundamentals of 5G: Architecture, protocols, and applications, SDN (Software-Defined Networking) and NFV (Network Function Virtualization), Performance evaluation using traffic models and QoS metrics, Network Optimization.

Unit - IV
Emerging Technologies and Applications: Vehicular networks and VANETs, Underwater and satellite communication networks, Blockchain in network security, Industrial IoT (IIoT) applications in smart factories, Network automation tools (e.g., Ansible, Puppet), AI and ML for network optimization.

Suggested Readings

1. Computer Networking: A Top-Down Approach by Kurose and Ross (Pearson).
2. Computer Networks by A. S. Tanenbaum (Pearson).
3. Computer Networks: A Top – Down Approach by Forouzan (McGraw Hill).
4. Data and Computer Communications by William Stallings (Pearson)
5. Mastering Networks: An Internet Lab Manual by Jorg Liebeherr and Magda El Zarki.
6. 5G Mobile and Wireless Communications Technology by Afif Osseiran et al.
7. Hands-On Networking Fundamentals by Michael Palmer.
8. Python for Data Analysis by Wes McKinney (for AI/ML in networks).
9. Introduction to Network Simulator NS2 by Teerawat Issariyakul and Ekram Hossain

List of Experiments (Based on Hardware/ Simulation)

1. Packet Sniffing and Analysis using Wireshark: Analyze network traffic and identify protocols.
2. Network Topology Design: Simulate wired and wireless networks using NS2/NS3.
3. IoT Device Communication: Establish an MQTT-based communication system using Raspberry Pi.
4. IPv6 Configuration: Design and configure an IPv6 network for efficient communication.
5. Firewall and VPN Setup: Implement a basic firewall and VPN for secure communication.
6. QoS Analysis: Analyze network performance using QoS metrics and traffic models.
7. 5G Network Simulation: Simulate basic 5G network architecture using MATLAB or NS3.
8. Sensor Interfacing with Microcontrollers: Interface sensors and collect data using Arduino/STM32.
9. IoT Cloud Integration: Connect Raspberry Pi to AWS IoT and monitor real-time data.
10. AI-based Network Optimization: Use Python to develop basic ML models for network performance analysis.
11. Network Automation: Automate network configurations using Ansible.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Digital Image Processing (DSE – 3)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Digital Image Processing	4	3	0	1	Fundamentals of Computer Programming, Mathematics – I

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Introduce fundamental concepts and techniques of digital image processing.
- Provide students with practical exposure to image enhancement, compression, and segmentation.
- Explore advanced topics like deep learning applications in image processing.
- Develop proficiency in using industry-standard tools for solving real-world image processing
- problems.

Course Outcomes:
After completing the course, the students should be able to:
1. Understand and implement fundamental image processing techniques such as filtering, transformation, and restoration.
2. Analyze and interpret image data for applications in various domains like healthcare, surveillance, and entertainment.
3. Develop image processing solutions using modern frameworks and libraries (e.g., OpenCV, MATLAB, Python).
4. Explore and apply machine learning and deep learning algorithms for advanced image processing applications.

Unit - I
Introduction to Digital Image Processing: Fundamentals of Image Processing, Image Formation, Sampling, and Quantization, Types of Images: Grayscale, RGB, Multispectral, Image Representation: Pixels, Bit Depth, and Resolution, Color Models: RGB, HSV, CMY, YUV, File Formats: BMP, JPEG, PNG, TIFF.

Unit - II
Image Enhancement and Restoration: Spatial Domain Techniques, Histogram Equalization, Spatial Filtering: Smoothing, Sharpening, Edge Detection, Frequency Domain Techniques, Fourier Transform and Its Applications, Frequency Filters: Low-Pass, High-Pass, Band-Pass, Noise Reduction and Image Restoration Techniques.

Unit - III
Image Analysis and Compression: Image Segmentation, Thresholding (Global, Adaptive), Region-Based Segmentation, Morphological Operations: Erosion, Dilation, Opening, Closing, Image Features and Descriptors, SIFT, SURF, HOG, Image Compression, Lossless vs. Lossy Compression, JPEG Compression Steps and Implementation.

Unit - IV

Advanced Topics in Digital Image Processing: Image Classification and Object Detection, Introduction to Convolutional Neural Networks (CNNs), Pretrained Models (VGG, ResNet, YOLO), Image Denoising using Autoencoders, Applications in Healthcare and Surveillance, Introduction to Video Processing and Motion Analysis.

Suggested Readings

1. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods (Pearson).
2. Computer Vision: Algorithms and Applications by Richard Szeliski.
3. Deep Learning for Computer Vision with Python by Adrian Rosebrock.
4. Programming Computer Vision with Python by Jan Erik Solem.

List of Experiments (Based on Hardware/ Simulation)

1. Basics of Image Processing
   Objective: Load, display, and convert images between color spaces.
   Requirements: Python (OpenCV, PIL), MATLAB.
2. Image Enhancement
   Objective: Implement histogram equalization and various spatial filters.
   Requirements: Python/NumPy, MATLAB, sample images.
3. Frequency Domain Filtering
   Objective: Apply Fourier transforms and design frequency filters.
   Requirements: Python (SciPy), MATLAB, signal processing toolbox.
4. Noise Removal
   Objective: Remove noise using different filters and compare results.
   Requirements: Python, MATLAB.
5. Image Segmentation
   Objective: Apply thresholding and morphological operations to segment images.
   Requirements: Python (OpenCV), MATLAB.
6. Feature Detection
   Objective: Detect features using SIFT, SURF, and HOG descriptors.
   Requirements: OpenCV with Python, dataset of sample images.
7. Image Compression
   Objective: Implement JPEG compression steps manually.
   Requirements: Python (NumPy, PIL), MATLAB.
8. CNN-based Image Classification
   Objective: Build a CNN for handwritten digit classification (MNIST dataset).
   Requirements: Python (TensorFlow/Keras), GPU support (optional).
9. Object Detection using YOLO
   Objective: Detect objects in real-time using YOLO.
   Requirements: Python, pretrained YOLO weights, webcam.
10. Motion Tracking in Video
    Objective: Implement motion tracking using optical flow.
    Requirements: Python (OpenCV), video datasets.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Core (DSC) Courses for B. Tech. (ECE) – Semester VI

Hands – On CMOS VLSI Design (DSC – 16)

(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Hands – On CMOS VLSI Design	4	3	0	1	Physics, Introduction to Electrical and Electronics Engineering

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- Provide a solid grounding in MOSFET device physics, CMOS technology, and fabrication flows, enabling students to understand how semiconductor devices are realized at the transistor level.
- Equip students with the ability to design, analyze, and optimize CMOS circuits, both digital and analog, and to understand their trade-offs in terms of power, speed, area, and reliability.
- Encourage maximum practical exposure by integrating simulation, layout, prototyping, and measurement, ensuring students feel engaged and enjoy learning through experiential labs.
- Introduce a broad range of CMOS circuit applications—oscillators, amplifiers, memory cells, frequency dividers, data recovery circuits—to help students appreciate the diversity and complexity of real-world VLSI systems.

Course Outcomes:

After completing the course, the students should be able to:

1. Demonstrate a thorough understanding of MOSFET operation, CMOS fabrication steps, and the principles underlying CMOS inverters and logic gates.
2. Design, simulate, and layout fundamental CMOS building blocks and assess their performance via SPICE simulations, post-layout verification, and basic prototyping.
3. Apply CMOS principles to implement and analyze advanced circuit blocks such as latches, memory cells, oscillators, and frequency dividers, understanding their practical roles in larger systems.

Unit - I

Introduction to CMOS Devices and Fabrication: Review of MOSFET. CMOS Inverter (response and characteristics), CMOS basic fabrication flow. Layout Fundamentals: Design rules, stick diagrams, layout for matching and symmetry. SPICE simulation basics.

Unit - II

Core CMOS Digital Circuits: Basic Gates: NAND, NOR, XOR, and their transistor-level implementations. Tapered Buffer: Design concepts, driving large capacitive loads efficiently. CMOS Latches and Flip-Flops: T-Gate Static Latch Tri-state Static Latch SR Latch and Dynamic Latch. Memory Elements: Static Memory Cell (e.g., 6T SRAM cell).

Unit - III
Analog and Mixed-Signal CMOS Circuits: LNA (input matching, noise figure, and gain optimization). The Self-Based Inverter & Duty Cycle Correction Circuits. Active Inductor: Realizing inductive behavior using CMOS transistors. Transimpedance Amplifier (TIA): Current-to-voltage conversion, bandwidth, and noise considerations.

Unit - IV
Oscillators: Ring Oscillator, Differential & Quadrature Ring Oscillator, Current-Controlled.
Clock and Data Recovery (CDR): Burst-mode CDR circuits, lock time, jitter performance. Frequency Dividers: Feedforward frequency divider, frequency divider with quadrature output.

Suggested Readings

1. CMOS VLSI Design: A Circuits and Systems Perspective by Neil H. E. Weste and David Harris (Pearson)
2. Digital Integrated Circuits: A Design Perspective by Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolić (Prentice Hall)
3. Design of Analog CMOS Integrated Circuits by Behzad Razavi (McGraw-Hill)
4. CMOS Analog Circuit Design by Phillip E. Allen and Douglas R. Holberg (Oxford University Press)
5. Analysis and Design of Analog Integrated Circuits by Paul R. Gray, Paul J. Hurst, Stephen H. Lewis, and Robert G. Meyer (Wiley)
6. The Art of Analog Layout by Alan Hastings (Prentice Hall)
7. RF Microelectronics by Behzad Razavi (Prentice Hall)
8. VLSI Technology by S. M. Sze (McGraw-Hill)
9. CMOS IC Layout: Concepts, Methodologies, and Tools by Dan Clein (Newnes)
10. VLSI Design Techniques for Analog and Digital Circuits by R. L. Geiger, P. E. Allen, and N. R. Strader (McGraw-Hill)

List of Experiments
(Will be based on open-source/ Proprietary EDA tools such as Silvaco's TCAD or Synopsys Sentaurus)

1. MOSFET Characterization
   Objective: Extract threshold voltage, measure I-V curves, and observe channel length modulation.
2. CMOS Inverter DC and Transient Response
   Objective: Simulate and measure inverter transfer characteristics, noise margins, and propagation delay.
3. Inverter Layout and Verification (Cadence Virtuoso)
   Objective: Draw layout of a CMOS inverter, run DRC/LVS, extract parasitics, and re-simulate.
4. Tapered Buffer Design
   Objective: Optimize buffer chain for driving large loads; compare delay and power.
5. T-Gate Static Latch Implementation
   Objective: Implement a T-Gate latch using FPGA logic slices, verify timing and functionality.
6. Tri-State Static Latch and SR Latch
   Objective: Construct and test latch behavior, measure setup/hold times.
7. Dynamic Latch and Static Memory Cell Simulation
   Objective: Simulate and analyze read/write operations and cell stability.

8. Low-Noise Amplifier Design
   Objective: Design and simulate an LNA at a given frequency, measure gain and noise figure.
9. Self-Based Inverter & Duty Cycle Correction Simulation
   Objective: Adjust duty cycle and observe waveform integrity in simulation.
10. Active Inductor Implementation
    Objective: Build an active inductor using discrete transistors or op-amps and measure frequency response.
11. Transimpedance Amplifier Testing
    Objective: Connect a photodiode to a TIA and measure output voltage for various light intensities.
12. Ring Oscillator Frequency Measurement
    Objective: Simulate a ring oscillator, vary supply voltage and measure frequency changes.
13. Quadrature Ring Oscillator Simulation
    Objective: Achieve quadrature outputs, measure phase offset.
14. Current-Controlled / Digitally Controlled Oscillator
    Objective: Vary control signals and observe frequency tuning range.
15. Burst-Mode CDR Simulation
    Objective: Analyze lock time and jitter in a burst-mode CDR circuit.
16. Feedforward Frequency Divider Implementation
    Objective: Design and simulate a CMOS-based frequency divider, measure output frequency.
17. Frequency Divider with Quadrature Output
    Objective: Generate and verify quadrature phase signals.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Embedded Systems and Applications (DSC - 17)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Embedded Systems and Applications	4	3	0	1	Fundamentals of Computer Programming

Course Hours: L – 03, T – 00, P – 02

Course Objectives:

- To understand the architecture and programming of microprocessors and microcontrollers.
- To gain insight into interfacing peripheral devices with processors/microcontrollers.
- To familiarize students with ARM CPU architecture and programming.
- To develop the ability to design and implement embedded systems for real-world applications.

Course Outcomes:

After completing the course, the students should be able to:

1. Analyze the architecture and instruction set of 8085, 8086, and 8051 processors/microcontrollers.
2. Develop assembly-level programs for solving computational problems.
3. Design and interface peripheral devices using protocols like SPI, I2C, and UART.
4. Implement embedded solutions using ARM architecture and modern software development tools.

Unit - I

Introduction to 8085 basic/applications: 8086 Microprocessor: Internal architecture, Real mode memory addressing, Instruction Format. Addressing modes: Data-Addressing modes, Program Memory, Addressing modes, Stack Memory-Addressing modes.

Unit - II

Instruction Set: Programming 8086 using: Data movement instructions: MOV, PUSH/POP, Load-Effective Address, String data transfers, miscellaneous data transfer instructions, Arithmetic and logic instructions, BCD and ASCII arithmetic, Shift and Rotate, String comparisons, Program control instructions, Introduction to interrupts.

Unit - III

Peripheral Devices: 8255-Programmable Peripheral Interface, 8254- Programmable interval Timer. DMA: Introduction to Direct memory Access. Interrupts: Basic interrupt processing, Interrupt instructions, Operation of real mode interrupt, interrupt flag bits, Hardware interrupts. Introduction to 8051 microcontrollers. RISC vs. CISC Architecture.

Unit – IV

ARM CPU Architecture: Introduction to ARM Architecture Programmer’s Model for ARM CPU Operating Modes, Instruction Set, Exception Handling, and Pipelining. ARM Cores (e.g., ARMv4). Peripheral-Processor Interfacing: Concepts of Peripheral Interfacing. Review of Peripheral Interface Protocols: SPI, I2C, UART, and One-Wire. Interface with Sensors, Radios, and ADCs. Embedded System Features: Hardware Timers and Interrupt Handling, Interrupt Service Routines.

Suggested Readings

1. The Intel Microprocessors: Architecture, Programming, and Interfacing by B. Brey (Pearson Education)
2. Microprocessors and Interfacing: Programming and Hardware by D. V. Hall (Tata McGraw Hill)
3. The x86 PC: Assembly Language, Design & Interfacing by Mazidi, Mazidi & Causey (Pearson Education)
4. An Embedded Software Primer by David E. Simon (Pearson Education)
5. Embedded Systems: Architecture, Programming, and Design by Raj Kamal (Tata McGraw Hill)
6. ARM System Developer's Guide by Andrew N. Sloss (Elsevier)
7. Embedded System Design by Frank Vahid and Tony Gwargie (John Wiley & Sons)
8. Embedded System Design by Steve Heath (Elsevier, Second Edition)
9. Embedded System Architecture: A Comprehensive Guide for Engineers and Programmers by Tammy Noergaard (Elsevier)
10. Programming Embedded Systems in C and C++ by Michael Barr (O'Reilly)

List of Experiments (Hardware Based)

1. Write an assembly language program to add two 8-bit and 16-bit hexadecimal numbers.
2. Write an assembly language program to transfer a block of data.
3. Write an assembly language program to multiply two 8-bit hexadecimal numbers.
4. Write an assembly language program to generate a Fibonacci series.
5. Write an assembly language program to sort hexadecimal numbers in ascending/descending order.
6. Study the working of IC 8255/8254 interfaced with the 8086 microprocessor.
7. Design and implement SPI and I2C-based communication for interfacing sensors with a microcontroller.
8. Develop an ARM-based interrupt-driven application using hardware timers.
9. Write an assembly language program to toggle an LED using the 8051 microcontroller.
10. Interface a 7-segment display with the 8051 microcontroller and display numbers.
11. Design a digital clock using the 8051 microcontroller and interfaced LCD.
12. Implement serial communication using the UART protocol on the 8051 microcontroller.
13. Write an ARM assembly program to add two numbers and display the result.
14. Implement an ARM-based program for a simple digital thermometer using ADC and a sensor interface.
15. Develop an ARM-based system to control a DC motor using PWM.
16. Interface an external EEPROM using I2C with an ARM microcontroller and perform read/write operations.
17. Implement interrupt-driven GPIO control on an ARM microcontroller.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Digital Communication Systems (DSC – 18)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Digital Communication Systems	4	3	0	1	Mathematic – II, Signals and Systems, Analog Communication Systems

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To introduce students to the fundamental concepts of digital communication systems.
- To enable the analysis and design of communication systems, focusing on performance under practical constraints like noise and interference.
- To incorporate hands-on experience with both hardware and software tools for implementing communication systems.
- To explore advanced and emerging topics like adaptive modulation, MIMO systems, and spread spectrum techniques.

Course Outcomes:
After completing the course, the students should be able to:
1. Design and analyze digital communication systems using theoretical concepts and practical tools.
2. Implement digital modulation, waveform coding, and baseband shaping using MATLAB, Simulink, and hardware kits.
3. Evaluate the performance of communication systems under various noise and channel conditions.
4. Explore and simulate advanced techniques like adaptive modulation, spread spectrum, and MIMO communication.

Unit - I
Transition from Analog to Digital Communication (Advantages/Disadvantages), Waveform Coding: Sampling Theorem for baseband and bandpass signals, Quantization and coding techniques: PCM, DPCM, Delta Modulation, and Adaptive Delta Modulation, Design and performance analysis of waveform coding systems.

Unit - II
Baseband Shaping: Discrete PAM signals and their power spectra. Pulse shaping and intersymbol interference (ISI). Nyquist criterion for distortionless transmission. Eye diagram analysis and equalization techniques. Concepts of Gram-Schmidt orthogonalization, Geometric Representation of Signals.

Unit - III
Detection and Estimation: Review of Gaussian Random Process, Detection of Known Signals in Noise, Optimum Threshold Detection, Optimum Receiver for AWGN Channel, Matched Filter and Correlation Receivers, Decision Procedure: Maximum A- Posteriori Probability Detector- Maximum Likelihood Detector, Probability of Error, Bit Error Rate.

Unit - IV
Digital modulation schemes: Shift Keying Methods (including ASK, FSK, Q/DQ/OQ/(pi/4)/8-/16-PSK, 16/64-QAM). Coherent M-ary Schemes, Non-Coherent Schemes.
Uncertainty, Information, Entropy, Source Coding, Huffman Coding, Shannon Fano Coding. Introduction to error control coding, Linear Block Code, Convolution Code.

Suggested Readings

1. Digital Communications by John G. Proakis (McGraw-Hill)
2. Principles of Communication Systems by Taub and Schilling (McGraw-Hill)
3. Wireless Communications: Principles and Practice by Theodore S. Rappaport (Pearson)
4. Information Theory, Inference, and Learning Algorithms by David J.C. MacKay (Cambridge)
5. Digital Communication Systems by Simon Haykin (John Wiley)
6. Modern Digital and Analog Communication by B.P. Lathi (Oxford)
7. Digital Communication by Sklar (Pearson)

List of Experiments (Hardware and Software using NI Multisim)

1. Design and Implementation of PAM Modulation and Demodulation Circuit
   Objective: Design a circuit for generating and demodulating a Pulse Amplitude Modulated (PAM) signal.
2. Design and Implementation of PWM and PPM Circuits
   Objective: Generate PWM and PPM signals using a monostable multivibrator and observe their waveforms.
3. Sampling and Reconstruction of Signals
   Objective: Design a sampling circuit using a sample-and-hold IC and reconstruct the original signal using a low-pass filter.
4. PCM Encoding and Decoding
   Objective: Implement PCM encoding and decoding using discrete components and compare the results with simulated output.
5. Delta Modulation and Demodulation
   Objective: Design and implement a delta modulation system using discrete components.
6. Eye Diagram Observation Using Oscilloscope
   Objective: Generate digital signals with ISI and observe their eye diagram on an oscilloscope.
7. Analyze the performance of waveform coding scheme.
8. Design and implement pulse shaping filters in MATLAB and verify Nyquist criterion.
9. Perform equalization experiments using DSP kits.
10. Implement digital modulation schemes: ASK
    Objective: Design and implement a circuit to perform ASK.
11. Implement digital modulation schemes: FSK
    Objective: Design and implement a circuit to perform FSK
12. Implement digital modulation schemes: PSK
    Objective: Design and implement a circuit to perform PSK.
13. Perform BER analysis for different modulation schemes under noise using MATLAB.
14. Matched Filter Receiver Design
    Objective: Design and analyze a matched filter receiver for detecting signals in noise.
15. Signal Detection in Noise
    Objective: Implement a signal detection circuit using thresholding and analyze its performance with varying noise levels.

16. QPSK Modulation and Demodulation
    Objective: Design a QPSK modulator and demodulator circuit using hardware.
17. Perform adaptive modulation experiments using MATLAB.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Elective (DSE) Courses for B. Tech. (ECE) – Semester VI

Wireless Sensor Networks (DSE – 4)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Wireless Sensor Networks	4	3	0	1	Fundamentals of Computer Programming, Signals and Systems

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To understand principles of sensor networks and its difference with mobile ad hoc networks.
- To evaluate computations related to energy saving using different routing schemes.
- To analyse different MAC protocols used for different communication standards in WSN.
- To design small sensor networks for different applications.

Course Outcomes:
After completing the course, the students should be able to:
1. Understand principles of sensor networks and its difference with mobile ad hoc networks.
2. Understand the concepts of Middleware and Transmission Technologies.
3. Evaluate computations related to energy saving using different routing schemes.
4. Analyse different MAC protocols used for different communication standards in WSN.
5. Design small sensor networks for different applications.

Unit - I
Cellular and Ad hoc wireless Networks: Mobile Ad-Hoc Networks, Sensor Networks. Applications, Categories, Issues and challenges in designing, Operating environment (Propagation). Architecture: Sensor node technology, Hardware and Software, Performance Metrics.

Unit - II
Middleware Functions, Architecture, Data management functions, Operating Systems, Design issues, Available wireless Technologies: WSN, Bluetooth, WLAN, Zigbee, WiMax, 3G and beyond. Performance modelling of WSN.

Unit - III
Fundamentals of MAC Protocols: Requirements and Design Constraints. Overview of MAC Protocols for WSN. Schedule-Based Protocols: SMAC, LEACH, and TRAMA. Contention-Based Protocols: CSMA and Slotted CSMA-CA Protocol. Non-beaconed Mode in MAC. Sensor MAC Protocol: Overview, Periodic Listen and Sleep Operations. Schedule Selection and Coordination, Synchronization, and Adaptive Listening. Access Control, Data Exchange, and Message Passing.

Unit - IV

Routing Protocols: Challenges, Issues, and Data Dissemination/Gathering. Location Discovery and Routing Strategies: Flooding, Gossiping, SPIN, PEGASIS, GEAR. Attribute-Based Routing: Direct Diffusion, Rumor Routing, Geographic Hash Tables. Transport Protocols: Design Issues and Feasibility of TCP/UDP for WSN. Advanced Protocols: CODA and GARUDA. Network Management: Techniques for Managing WSN Performance.

Suggested Readings

1. Fundamentals of Wireless Sensor Networks: Theory and Practice by Waltenegus Dargie and Christian Poellabauer (John Wiley & Sons Publications)
2. SENSORS Handbook by Sabrie Soloman (McGraw Hill Publication)
3. Wireless Sensor Networks by Feng Zhao and Leonidas Guibas (Elsevier Publications)
4. Wireless Sensor Networks: Technology, Protocols, and Applications by Kazem Sohraby and Daniel Minoli (Wiley-Interscience)

List of Experiments

1. Sensor Node Architecture Exploration
   Objective: Understand sensor node hardware and software.
   Requirements: Arduino/NodeMCU, temperature sensor, humidity sensor, multimeter.
2. Signal Propagation in Wireless Networks
   Objective: Study propagation impairments in WSNs.
   Requirements: RF transceiver modules, spectrum analyzer, wireless signal simulator.
3. Middleware Functions for WSNs
   Objective: Explore middleware functions and data management.
   Requirements: Raspberry Pi, Python, SQLite database.
4. Performance Modeling of WSN
   Objective: Analyze performance metrics of WSNs.
   Requirements: MATLAB/NS3 simulator.
5. Bluetooth and Zigbee Communication
   Objective: Implement WSN communication using Bluetooth and Zigbee.
   Requirements: Bluetooth modules, Zigbee modules, Arduino/NodeMCU.
6. Implementation of SMAC Protocol
   Objective: Implement and evaluate SMAC for energy-efficient communication.
   Requirements: NS3 simulator
7. Study PHY and MAC layer operations of IEEE 802.15.4.
   Requirements: Zigbee modules, NS3 simulator.
8. Explore LEACH and TRAMA protocols.
   Requirements: NS3 simulator, PCs.
9. Routing Protocol Implementation
   Objective: Implement and compare different routing strategies.
   Requirements: NS3 simulator, PCs.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Artificial Intelligence in Electronics (DSE – 4)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Artificial Intelligence in Electronics	4	3	0	1	Fundamentals of Computer Programming, Mathematics - I

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Introduce fundamental AI and machine learning concepts relevant to electronics and signal processing.
- Explore AI algorithms on electronic hardware platforms (microcontrollers, FPGAs, ASICs, SoCs) for real-time applications.
- Apply ML techniques to sensor signals (vision, audio, RF, biomedical) to develop intelligent systems.
- Expose students to advanced concepts like neuromorphic computing, low-power edge AI, and AI accelerators.
- Provide hands-on experience from simulation to hardware prototyping, ensuring job-ready skills.

Course Outcomes:
After completing the course, the students should be able to:
1. Implement AI/ML models for classification, regression, and pattern recognition in electronics.
2. Integrate and optimize ML algorithms for embedded platforms considering power, performance, and area.
3. Design AI-based intelligent embedded solutions like smart sensors and predictive maintenance modules.
4. Use AI/ML libraries (TensorFlow, PyTorch, scikit-learn) and hardware tools.

Unit - I
Foundations of AI and ML for ECE: AI, ML, and Deep Learning: Key concepts and relevance to ECE. Supervised & Unsupervised Learning: Linear regression, logistic regression, k-means clustering. Classification & Regression Techniques: SVMs, Decision Trees, Random Forests. Basics of Neural Networks: Perceptron, Multi-layer Perceptrons, activation and loss functions. Gradient Optimization, Backpropagation, Regularization Techniques. Python-based ML frameworks: scikit-learn, TensorFlow.

Unit - II
Embedded and Hardware-Aware AI: Hardware Basic: FPGAs, and AI accelerators. Real-time Constraints: Latency, throughput, memory optimization. Model Compression & Quantization for Edge Devices. AI on Microcontrollers & Low-power SoCs (ARM Cortex-M, RISC-V). FPGA-based AI design flows and high-level synthesis (HLS) for ML.

Unit - III
Intelligent Signal Processing & Sensor Fusion: Signal Processing with AI: Denoising, filtering, feature extraction. Computer Vision & Image Processing: CNNs, object detection, classification. Speech & Audio Processing: RNNs, LSTMs, Transformer-based models. Sensor Fusion & IoT: Multi-sensor integration and ML algorithms. Applications: Smart homes, wearables, biomedical, industrial IoT. Model Distillation.

Unit - IV

Advanced Topics & Emerging Trends: Neuromorphic Computing: Spiking Neural Networks, event-driven sensors. Edge AI & TinyML: Low-power inference, federated learning. Neural Architecture Search (NAS) & AutoML: Automated model optimization. Domain-Specific AI Accelerators: ASIC design for ML inference. Case Studies: Robotics, drones, biomedical diagnostics. Model Distillation.

Suggested Readings

1. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (MIT Press, 2016)
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurelien Geron (O'Reilly, 2nd Edition)
3. Neural Networks and Learning Machines by Simon Haykin (Pearson, 3rd Edition)
4. Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig (Pearson, 4th Edition)
5. TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers by Pete Warden and Daniel Situnayake (O'Reilly)
6. Vivado Design Suite User Guides (Xilinx) and Intel Quartus Documentation for HLS
7. RISC-V-Based SoC Design by S. Wong et al. (Springer)

List of Experiments (Based on Simulation)

1. Basic ML Implementation
   Objective: Train a simple linear regression and classification model on a given dataset.
2. Neural Network for Signal Classification
   Objective: Implement a small MLP to classify sine vs. square waves or detect a known waveform.
3. Edge AI on a Microcontroller
   Objective: Deploy a compressed NN model (tiny model trained offline) onto an Arduino or STM32 board to classify simple sensor data (e.g., temperature thresholds or vibration patterns).
   Requirements: Trained tiny model (tflite), Arduino/STM32 board.
4. Computer Vision on Embedded Platform (Raspberry Pi + Camera)
   Objective: Run a CNN-based object detection or face recognition model on a Raspberry Pi with a camera feed.
   Requirements: Raspberry Pi, camera module, pre-trained CNN model (MobileNet or YOLO-tiny), OpenCV.
5. Audio Keyword Spotting
   Objective: Train a simple keyword spotting model (e.g., detecting the word "Hello") and deploy it on an NVIDIA Jetson Nano.
   Requirements: Microphone, recorded audio samples, TensorFlow Lite, Jetson Nano.
6. FPGA-based Inference Accelerator (to lean hardware acceleration)
   Objective: Implement a basic neural network inference engine on an FPGA using HLS.
   Requirements: FPGA board (PYNQ-Z2), Vivado, a small pre-trained NN model.
7. Sensor Fusion Project
   Objective: Combine data from an IMU (accelerometer/gyroscope) and a camera feed, run a simple fusion model to determine object orientation.
   Requirements: Raspberry Pi/STM32 with IMU and camera, Python, fusion algorithm.
8. Spiking Neural Network
   Objective: Implement a Spiking Neural Network simulation and observe event-driven processing.
   Requirements: Python-based SNN simulator (e.g., Brian2), synthetic spiking data.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Coding and Data Compression Techniques (DSE – 4)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Coding and Data Compression Techniques	4	3	0	1	Fundamentals of Computer Programming, Mathematics – II, Signals and Systems

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Understand the fundamental concepts of information theory, entropy, and mutual information.
- Learn about source coding, data compression techniques, and error control coding.
- Explore the theoretical and practical aspects of channel capacity and coding theorems.
- Gain proficiency in advanced coding techniques like convolutional codes, Turbo coding, and universal source coding.
- Develop analytical skills to compute rate-distortion functions and implement algorithms such as Lempel-Ziv and Blahut-Arimoto.

Course Outcomes:
After completing the course, the students should be able to:
1. Analyze and calculate information measures such as entropy and mutual information.
2. Design efficient codes for data compression and error control.
3. Evaluate the capacity of different communication channels and understand their theoretical limits.
4. Implement universal coding algorithms for data compression.
5. Apply advanced error control and source coding techniques to practical communication problems.

Unit - I
Entropy, Relative Entropy, and Mutual Information: Entropy, Joint Entropy and Conditional Entropy, Relative Entropy and Mutual Information, Chain Rules, Data-Processing Inequality, Fano's Inequality. Source Coding and Data Compression: Kraft Inequality, Huffman Codes, Optimality of Huffman Codes, Shannon Fano Codes.

Unit - II
Channel Capacity: Symmetric Channels, Properties of Channel Capacity, Jointly Typical Sequences, Channel Coding Theorem, Fano's Inequality and the Converse to the Coding Theorem. Variable to Block Length Coding: The Asymptotic Equipartition Property Block to Block Coding of Discrete Memoryless Sources (DMS). Rate Distortion Theory, Blahut-Arimoto Algorithm for Channel Capacity and Rate-Distortion Function.

Unit - III
Standard Error control coding- Block codes: Definitions and Principles: Hamming weight, Hamming distance, Minimum distance decoding - Single parity codes, Hamming codes, Repetition codes -Linear block codes. Cyclic codes - Syndrome calculation, Encoder and decoder – CRC

Unit - IV

Error control coding- convolution codes: code tree, trellis, state diagram, Encoding – Decoding: Sequential search and Viterbi algorithm – Principle of Turbo coding. Universal Source Coding: Lempel-Ziv Algorithm, Introduction to Reed Solomon (RS) Codes.

Suggested Readings

1. Elements of Information Theory by T. M. Cover, and J. A. Thomas (Wiley).
2. Channel Codes: Classical and Modern by William Ryan, and Shu Lin (Cambridge).
3. Information Theory and Reliable Communication by Robert Gallager (Wiley).
4. Error Control Coding by Shu Lin and Daniel Costello Jr (Pearson).
5. Theory and Practice of Error Control Codes by Richard E. Blahut (Addison-Wesley).

List of Experiments (Hardware/ Software based)

1. Implement entropy calculation and mutual information for given data sets.
2. Design Huffman and Shannon-Fano codes for a given source.
3. Simulate the channel coding theorem.
4. Evaluate Gaussian channel capacity and compute rate-distortion functions using the Blahut-Arimoto algorithm.
5. Analyze the performance of single parity codes.
6. Design and implement Hamming codes.
7. Implement cyclic redundancy check (CRC) encoding and decoding.
8. Implement Lempel-Ziv algorithms for text compression.
9. Simulate linear block codes.
10. Simulate convolutional codes.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Electromagnetic Compatibility: Principles and Applications (DSE – 4)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Electromagnetic Compatibility: Principles and Applications	4	3	0	1	Physics, Network Analysis and Synthesis, Signals and Systems, Electromagnetic Theory

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To provide a comprehensive understanding of electromagnetic compatibility (EMC) principles and challenges in electronic systems.
- To enable students to analyze and design electronic systems that minimize electromagnetic interference (EMI) and ensure compliance with EMC standards.
- To familiarize students with EMC testing, measurement techniques, and the use of computational modeling tools.
- To equip students with hands-on skills in designing noise-suppressed circuits, effective PCB layouts, and implementing shielding and ESD protection strategies.

Course Outcomes:
After completing the course, the students should be able to:
1. Develop the ability to identify and classify EMC problems in electronic systems and propose appropriate mitigation strategies.
2. Gain proficiency in utilizing simulation tools and experimental setups to evaluate and optimize EMC performance.
3. Design and implement EMC-compliant PCBs, shielding solutions, and noise suppression techniques in practical applications.
4. Demonstrate knowledge of international EMC standards, testing protocols, and regulatory requirements for electronic systems.

Unit - I
Fundamentals of EMC and Standards: Introduction to EMC: Problem classifications, physical and electrical dimensions of components. Common EMC units and their significance. Basics of transmission line theory for EMC applications. Overview of EMC signal sources and their characterization. EMC Standards: Conducted emissions standards and testing. Radiated emissions standards and testing. Regulatory frameworks: FCC, CISPR. Antenna factors and their importance in EMC measurements.

Unit - II
Component Behavior and Noise Suppression: Low- and high-frequency approximations of circuit components. Internal impedance of round wires and PCB traces. External inductance, capacitance, and conductance in coaxial and parallel wires. Nonideal behavior of resistors, capacitors, and inductors. Common mode vs. differential mode currents. Use of ferrites and common mode chokes for noise suppression. Digital circuit devices and their role in EMC.

Unit - III

Signal Spectra and Radiated Emissions: Classification of signals: Periodic and aperiodic. Fourier series and transforms for signal spectrum analysis. Spectra of digital clock waveforms and their implications. Emission models for wires, PCB lands, and their corresponding signal spectra. Measured spectra and antenna factor effects on emissions.

Unit - IV

Crosstalk, Shielding, and PCB Design for EMC: Crosstalk in multi-conductor systems: Per-unit-length parameters and time-domain analysis. Shielding: Far-field and near-field effectiveness. EMC/EMI computational modeling: FDTD and Method of Moments techniques. PCB Design for EMC: Board stack-up issues. Component placement and signal isolation strategies. Electrostatic Discharge (ESD): Mechanisms, models (Human Body Model), and protection techniques.

Suggested Readings

1. Introduction to Electromagnetic Compatibility by Clayton R. Paul (John Wiley & Sons, 2nd Ed., 2006).
2. Noise Reduction Techniques in Electronic Systems by Henry W. Ott (John Wiley & Sons, 2nd Ed., 1988).
3. PCB Design for Real-World EMI Control by Bruce R. Archambeault (Springer Science + Business Media, LLC, 2002).
4. EMC for Product Designers by Tim Williams (Elsevier, 5th Ed., 2016).
5. EMC of Analog Integrated Circuits by Christoph Schubert & Bruno Schumacher (Springer, 2015).
6. Electromagnetic Compatibility: Principles and Applications by David Weston (Marcel Dekker, Inc., 2nd Ed., 2001).

List of Experiments (Hardware/ Software based)

1. Signal Characterization:
   Generate and analyze periodic and aperiodic signals using oscilloscopes and spectrum analyzers.
   Software implementation of Fourier transforms for signal spectra (MATLAB/Python).
2. PCB Design for EMC:
   Design and test PCB layouts for minimal crosstalk and enhanced signal isolation.
3. Noise Suppression Techniques:
   Construct circuits with ferrites, capacitors, and inductors to suppress common mode noise.
4. ESD Testing:
   Measure and protect against electrostatic discharges using protection circuits.
5. EMC Shielding:
   Evaluate shielding effectiveness for near-field and far-field scenarios.
6. Simulation of Transmission Line Behavior:
   a. Use simulation tools (e.g., LTSpice, ADS, or CST Studio) to model and analyze the behavior of transmission lines under different signal frequencies.
   b. Evaluate the effects of impedance mismatches on signal integrity.
7. Modeling EMC Signal Sources:
   a. Simulate different EMC signal sources, such as sinusoidal, square, and pulse waveforms, using MATLAB/Simulink or Python.
   b. Analyze their spectra and relate them to real-world EMC challenges.
8. Analysis of Non-Ideal Component Behavior:
   a. Simulate inductors, capacitors, and resistors at high frequencies using tools like LTSpice or COMSOL Multiphysics.
   b. Compare the simulated performance with ideal models and observe deviations.

9. Fourier Transform Analysis of Signal Spectra:
   Objective: Implement Fourier transforms in MATLAB or Python to analyze the spectra of periodic and aperiodic signals.
10. Crosstalk Simulation:
    a. Model multi-conductor transmission lines in CST Studio or Ansys HFSS.
    b. Simulate and visualize crosstalk between adjacent traces or cables and identify mitigation strategies.
11. Shielding Effectiveness Analysis:
    a. Simulate shielding effectiveness in near-field and far-field scenarios using CST Studio or COMSOL Multiphysics.
    b. Compare different materials and configurations for their shielding performance.
12. Modeling Electrostatic Discharge (ESD) Events:
    a. Use SPICE-based tools or MATLAB to simulate ESD events and observe their impact on circuits.
    b. Implement and test ESD protection circuits virtually.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

List of Discipline Specific Elective (DSE)/ Generic Elective (GE) courses offered for Minors/ Specializations by the Department in Second Year

1. **Minor in ECE (Offered only to CSE and EE)**
   a) DSE - 3/ GE - 5: Communication Architecture
   b) DSE - 4/ GE - 6: Antenna and Wave Propagation
   Or
   Elements of Wireless Communication

2. **Minor/ Specialization in Telecommunication Networks (Offered to ECE, CSE, and EE)**
   a) DSE - 3/ GE - 5: Wireless Communication and Mobile Networks
   b) DSE - 4/ GE - 6: Satellite Communication
   Or
   Cognitive Radio & Networks

3. **Minor/ Specialization in VLSI Technology and System Design (Offered to ECE, CSE, and EE)**
   a) DSE - 3/ GE - 5: Advanced Digital VLSI Circuits and Physical Design
   b) DSE - 4/ GE - 6: CMOS Analog IC Design
   Or
   Current Mode Analog VLSI Circuits

4. **Minor/ Specialization in IoT System Design (Offered to ECE, CSE, and EE)**
   a) DSE - 3/ GE - 5: Introduction to Security of Cyber-Physical Systems
   b) DSE - 4/ GE - 6: Ubiquitous Sensing, Computing and Communication
   Or
   Introduction to Embedded Systems for IoT

5. **Minor/ Specialization in Computer Vision (Offered to ECE, CSE, and EE)**
   a) DSE - 3/ GE - 5: Intelligent Imaging
   b) DSE - 4/ GE - 6: Deep Learning for Image Analysis
   Or
   Deep & Reinforcement Learning

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Generic Elective (GE) courses offered for Minors/ Specializations by the Department in Semester V

Communication Architecture (DSE – 3/ GE – 5)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Communication Architecture	4	3	0	1	NIL

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To understand the fundamental principles of analog and digital communication systems.
- To familiarize students with communication architectures and their practical implementations.
- To develop hands-on skills in designing and analyzing communication systems using modern tools.
- To prepare students for advanced topics in wireless communication and antenna theory.

Course Outcomes:
After completing the course, the students should be able to:
1. Explain the structure and functionality of analog and digital communication systems.
2. Design and simulate basic communication architectures using appropriate software and hardware tools.
3. Analyze and troubleshoot communication systems for practical applications.
4. Transition smoothly to advanced topics like antennas and wireless communication.

Unit - I
Fundamentals of Communication Systems: Introduction to communication: Analog vs. Digital communication. Basic elements of a communication system: Transmitter, channel, receiver. Signal modulation: Amplitude Modulation (AM), Frequency Modulation (FM), and Phase Modulation (PM). Noise in communication systems and signal-to-noise ratio (SNR). Practical communication channels and their characteristics.

Unit - II
Digital Communication Systems: Sampling theorem and quantization. Pulse Code Modulation (PCM), Differential PCM, and Delta Modulation. Digital Modulation Techniques: ASK, PSK, FSK, QAM. Introduction to error detection and correction. Basics of Multiplexing: Time Division Multiplexing (TDM) and Frequency Division Multiplexing (FDM).

Unit - III
Communication Architectures and Protocols: Introduction to layered communication architecture (OSI and TCP/IP models). Physical and Data Link Layer protocols. Basics of wired communication standards (Ethernet) and wireless standards (Wi-Fi, Bluetooth). Signal propagation and channel impairments.

Unit - IV
Advanced Communication Concepts: Introduction to Spread Spectrum Techniques: DSSS and FHSS. Basics of MIMO (Multiple Input Multiple Output) systems. Introduction to IoT communication protocols: MQTT and Zigbee. Overview of Software Defined Radio (SDR) and Cognitive Radio.

Suggested Readings

1. Modern Digital and Analog Communication Systems by B.P. Lathi (Oxford).
2. Communication Systems by Simon Haykin (Wiley).
3. Digital Communications by John G. Proakis (McGraw-Hill).
4. Wireless Digital Communications by K. Feher (Prentice Hall).
5. Wireless Communications by Andrea Goldsmith (University Press).
6. Wireless Communications: Principles and Practice by Theodore S. Rappaport (Pearson).

List of Experiments (Hardware on Breadboard / Software using NI Multisim)

1. Design and implement an AM transmitter and receiver.
2. Design and implement Pulse Code Modulation (PCM).
3. Design and Simulate ASK (MATLAB or Simulink and Hardware).
4. Design and Simulate PSK (MATLAB or Simulink and Hardware).
5. Design and Simulate QAM (MATLAB or Simulink and Hardware).
6. Simulate a basic wireless communication link using SDR tools.
7. Measure signal impairments in a communication link using spectrum analyzers.
8. Build a simple IoT communication system using MQTT on NodeMCU.
9. Simulate a basic MIMO system using MATLAB or Simulink.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Wireless Communication and Mobile Networks (DSE – 3/ GE – 5)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Wireless Communication and Mobile Networks	4	3	0	1	NIL

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To introduce the basic cellular concepts.
- To understand the various signal propagation effects.
- To study various multiple access schemes.
- To familiarize with various mobile standards.
- To study the implementation of MIMO systems for high-speed communication

Course Outcomes:
After completing the course, the students should be able to:
1. Understand cellular concepts and signal propagation in mobile communication.
2. Understand the fundamental principles of signal propagation in wireless communication.
3. Compare and contrast various multiple access techniques and their suitability for different applications. Evaluate the advantages and limitations of different modulation schemes
4. Understand the architecture of mobile communication networks, including GSM, CDMA, 3G, 4G LTE, and 5G.
5. Design MIMO configurations to enhance wireless communication systems.

Unit - I
Wireless Channel Characteristics & Cellular Concepts: Cellular concepts: Cell structure, frequency reuse, cell splitting, channel assignment, handoff mechanisms, interference management, capacity planning, power control. Antennas for mobile communication: Mobile terminal antennas (monopole, PIFA), base station antennas, antenna arrays. Propagation mechanisms: Reflection, refraction, diffraction, scattering Large-scale signal propagation models and lognormal shadowing.

Unit - II
Fading Channels & Their Characterization: Fading channels: Multipath and small-scale fading, Doppler shift, statistical channel models. Narrowband and wideband fading models, power delay profile, RMS delay spread. Coherence bandwidth, coherence time, flat vs. frequency-selective fading, slow vs. fast fading, average fade duration, level crossing rate. Channel capacity for flat and frequency-selective fading channels.

Unit - III
Multiple Access Techniques & Receiver Architectures: Multiple access techniques: FDMA, TDMA, CDMA, SDMA, OFDMA. Receiver design: Diversity receivers (selection, MRC), RAKE receiver, Equalization (linear-ZFE, adaptive, DFE). Transmit diversity: Alamouti scheme. Introduction to spread spectrum techniques. Brief introduction to WLAN (Wi-Fi) and PAN (Bluetooth, Zigbee)

Unit - IV

Wireless Standards, Advanced Technologies & Network Architectures: Cellular standards evolution: Overview of 2G (GSM), 3G (WCDMA/UMTS), 4G (LTE), 5G NR, and emerging 6G. System examples: GSM, EDGE, GPRS, IS-95, CDMA2000, WCDMA. MIMO and space-time signal processing: Spatial multiplexing, diversity/multiplexing trade-off. Performance measures: Outage probability, average SNR, BER, QoS considerations. Network architectures: Mobile IP, mobility management, LTE/EPC architecture (MME, SGW, PGW), planning & optimization

Suggested Readings

1. 4G, LTE-Advanced Pro and The Road to 5G (Third Edition, 2016) by Erik Dahlman (Academic Press)
2. 5G NR: Architecture, Technology, Implementation, and Operation of 3GPP New Radio Standards (2019) by Sassan Ahmadi (Academic Press)
3. Wireless Communication and Networking by Vijay K. Garg (Elsevier, Morgan Kaufmann, 2012)
4. Wireless Communications: Principles and Practice by T. S. Rappaport (PHI, 2006)
5. Mobile Cellular Telecommunications: Analog and Digital Systems by William Lee (McGraw Hill Education, 2017)
6. Fundamentals of Wireless Communication by David Tse and Pramod Viswanath (Cambridge University Press, 2005)
7. Mobile Communications by Jochen Schiller (Addison-Wesley, 2003)
8. LTE for UMTS: Evolution to LTE-Advanced by Harri Holma and Antti Toskala (Wiley, 2007)
9. Wireless Communication Networks and Systems by Cory Beard and William Stallings (Pearson, 2016)

<u>List of Experiments</u> (Hardware on Breadboard / Software using NI Multisim)

1. Measure path loss, received signal strength, and antenna radiation patterns using a signal generator and spectrum analyzer.
   Evaluate monopole and PIFA antennas for mobile terminals.
2. Channel emulators to observe multipath fading effects, measure delay spreads, and analyze fading statistics.
3. Implement simple modulation/demodulation schemes (QPSK, OFDM) and observe BER under varying channel conditions.
4. Simulate cellular networks, analyze handover strategies, and evaluate resource scheduling algorithms.
5. Antenna Orientation Experiment:
   a. Attach a simple dipole or whip antenna to a transmitter and measure received power at a fixed receiver position.
   b. Rotate the antenna at various angles and observe how polarization and orientation affect the signal.
6. Basic Network Simulator Introduction (NS-3):
   a. Run a basic NS-3 simulation for a simple point-to-point wireless scenario.
   b. Compare throughput and delay under different traffic loads and link distances.
7. Indoor/Outdoor Path Loss Comparison:
   a. Use a simple RF transmitter-receiver pair (e.g., low-cost FM module or a simple 2.4 GHz transceiver) to measure signal strength indoors and outdoors.
   b. Record how the signal attenuates with distance in each environment.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Advanced Digital VLSI Circuits and Physical Design (DSE – 3/ GE – 5)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Advanced Digital VLSI Circuits and Physical Design	4	3	0	1	Physics, Microelectronics Design

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Equip students with advanced knowledge of CMOS logic families, dynamic logic, and memory elements beyond basic combinational and sequential logic.
- Expose students to the back-end design flow, including floorplanning, placement, routing, design rule checks, and layout verification.
- Enable students to use industry-standard EDA tools for implementing, verifying, and optimizing digital circuits at both the schematic and layout levels.
- Familiarize students with timing closure, advanced node scaling challenges, and emerging trends like 3D IC integration and advanced packaging.
- Provide a learning experience that balances theory and hands-on labs, encouraging creative problem-solving and teamwork through mini-projects and gamified activities.

Course Outcomes:
After completing the course, the students should be able to:
1. Create and optimize advanced digital logic circuits (dynamic logic gates, memory cells, arithmetic blocks) with a clear understanding of performance, area, and power.
2. Perform floorplanning, placement, routing, and post-layout verification using professional EDA tools, ensuring DRC/LVS compliance and timing closure.
3. Understand and apply methods for static timing analysis, power optimization (clock gating, multi-V_t design), and signal integrity checks.
4. Gain awareness of advanced nodes, FinFET/3D integration, advanced packaging, and their implications on design methodologies.

Unit - I
Advanced Digital CMOS Circuits: Review of CMOS Logic & Static CMOS: Brief refresher of CMOS inverter, NAND, NOR, complex gates, Dynamic Logic & Domino Circuits: Principles, advantages/ disadvantages, charge sharing, noise margins, Pass-Transistor & Transmission Gate Logic: Design techniques, area and speed considerations, Memory Elements (SRAM, ROM, CAM Basics): 6T SRAM cell layout, read/write operations, timing margins, Case Studies: Arithmetic circuits (adders, multipliers), pipeline registers.

Unit - II
Physical Design Flow & Methodologies: Introduction to ASIC & Full-Custom Flows: Standard-cell methodology, IP reuse, PDKs, Floorplanning & Partitioning: Hierarchical design, IO and macro placement, minimizing wirelength and congestion, Placement & Routing: Timing-driven placement, global vs. detailed routing strategies, crosstalk avoidance, Design Rule Checking (DRC) & Layout vs. Schematic (LVS): Physical verification fundamentals, running DRC/LVS tools, fixing violations.

Unit - III

Timing Closure & Power Optimization: Static Timing Analysis (STA): Setup/hold timing, clock skew, Clock Tree Synthesis & Optimization: Balancing skew, buffering, clock gating for power reduction, Power Optimization Techniques: Multi-threshold CMOS, power gating, dynamic voltage/frequency scaling, Signal Integrity & Parasitic Extraction: Crosstalk, IR drop, RC extraction, post-layout optimization.

Unit - IV

Advanced Transistor Technologies: FinFETs, Gate-All-Around FETs, impact on design rules and PPA (Power-Performance-Area), 3D ICs & Advanced Packaging: Through-Silicon Vias (TSVs), chiplets, heterogeneous integration, Reliability & Variability: Aging, NBTI, EM, process variation, methods to mitigate reliability issues.

Suggested Readings

1. CMOS VLSI Design: A Circuits and Systems Perspective by N.H.E. Weste and D. Harris (Pearson).
2. CMOS Digital Integrated Circuits: Analysis and Design by Sung-Mo (Steve) Kang, Yusuf Leblebici (McGraw-Hill).
3. VLSI Physical Design Automation by Sadiq M. Sait and Habib Youssef (McGraw-Hill).
4. Synthesis and Optimization of Digital Circuits Giovanni De Micheli (McGraw-Hill).
5. Digital Integrated Circuits: A Design Perspective by Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolic (Pearson).

<u>List of Experiments</u> (Hardware / Software based)

1. Complex Gate Design & Simulation:
   Task: Design and simulate a complex AOI/OAI gate at transistor-level using SPICE. Analyze delay, power, and voltage transfer characteristics.
2. Dynamic Logic Implementation:
   Task: Implement a domino logic gate and verify its operation for a given digital function. Investigate charge leakage and noise margins.
3. SRAM Cell Layout & Verification:
   Task: Layout a 6T SRAM bit-cell and perform DRC and LVS checks. Extract parasitics and run post-layout simulation to verify read/write stability.
4. RTL-to-Gates Synthesis (Simple ALU Block):
   Task: Take a behavioral RTL code of an 8-bit ALU, synthesize it to gates.
5. RTL-to-Gate Synthesis
   Task: Synthesize an RTL description of a small arithmetic module (e.g., a 4-bit multiplier) into a gate-level netlist using a standard-cell library.
6. Floorplanning & Placement (Software: ICC2 or Cadence Innovus)
   Task: Perform floorplanning and placement of the synthesized netlist.
7. Routing & DRC/LVS Checks (Software: ICC2 + Calibre/ICV)
   Task: Route the design and verify it against foundry DRC/LVS rules.
8. Advanced Node Exploration (FinFET PDK):
   Task: Using a simplified FinFET PDK, implement a small inverter and compare area, delay, and power to planar CMOS.

9. FinFET vs. Planar CMOS Device Simulation (Software: Silvaco)
   Task: Simulate a FinFET structure and compare its I-V characteristics, threshold voltage, and short-channel effects to a traditional planar MOSFET.

10. Reliability & Aging Simulation
    Task: Model NBTI aging in a PMOS device and observe how threshold voltage shifts over time.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Introduction to Security of Cyber-Physical Systems (DSE – 3/ GE – 5)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Introduction to Security of Cyber-Physical Systems	4	3	0	1	Introduction to IoT, Introduction to IoT System Design

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To introduce fundamental concepts of security in IoT-based Cyber-Physical Systems (CPS).
- To understand security challenges, vulnerabilities, and risks in CPS.
- To explore practical solutions for securing CPS using hands-on implementations.

Course Outcomes:
After completing the course, the students should be able to:
1. Identify security challenges and threats in CPS.
2. Implement basic security mechanisms for IoT systems.
3. Analyze case studies and scenarios for real-world CPS security issues.

Unit - I
Fundamentals of CPS Security: Overview of Cyber-Physical Systems (CPS) and IoT. Security needs in CPS: Confidentiality, Integrity, Availability (CIA). Common threats and attack vectors in IoT and CPS (DDoS, Malware, Eavesdropping, Physical Tampering). Overview of Security Standards: NIST Cybersecurity Framework, ISO/IEC 27001.

Unit - II
Authentication and Access Control in CPS: Cryptographic Basics: Hashing, Encryption (AES, RSA), Digital Signatures. Authentication mechanisms for IoT devices. Access Control Policies: Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC). Secure communication using TLS/SSL and HTTPS.

Unit - III
Intrusion Detection and Mitigation Techniques: Intrusion Detection Systems (IDS) for IoT. Malware analysis and prevention. Secure device boot and firmware updates. Lightweight security protocols for resource-constrained devices.

Unit - IV
Software-Defined Networks: Introduction, Security. CSP – Platform, Components, Implementation Issues, Intelligent CPS, Secure Deployment of CPS. Designing a secure CPS system: Practical insights. Introduction to Blockchain for IoT security.

Suggested Readings

1. Cybersecurity for IoT by Brian Russell and Drew Van Duren.
2. Securing the Internet of Things by Li Da Xu, and Shancang Li (Syngress).
3. IoT Security Issues by Alasdair Gilchrist (De Gruyter).
4. The Internet of Risky Things by Sean Smith (O'Reilly).

<u>List of Experiments</u> (Hardware on Breadboard / Software using NI Multisim)

1. Encryption and Decryption using Python (AES/RSA algorithms).
   Software: Python libraries (PyCrypto, Cryptography).
   Hardware: Raspberry Pi.
2. Configuring HTTPS for IoT Devices using NodeMCU.
   Software: Arduino IDE, OpenSSL.
3. Intrusion Detection Simulation using Wireshark and Snort.
   Software: Wireshark, Snort.
4. Secure Device Communication using TLS with MQTT.
   Hardware: NodeMCU.
5. Firmware Update Simulation for IoT device.
   Hardware: ESP32.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Intelligent Imaging (DSE – 3/ GE – 5)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Intelligent Imaging	4	3	0	1	Mathematics – I, Fundamentals of Computer Programming, Mathematics – II, Fundamentals of Image Processing, Image Filtering and Restoration

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Introduce advanced imaging techniques (e.g., hyperspectral, 3D imaging, light-field imaging) and their roles in intelligent systems.
- Apply state-of-the-art machine learning and deep learning models for image interpretation, object recognition, and scene understanding.
- Reinforce hands-on proficiency with industry-standard libraries (OpenCV, TensorFlow/PyTorch) and hardware (embedded vision systems, specialized sensors).
- Encourage students to experiment, prototype, and innovate intelligent imaging solutions for real-world problems (e.g., medical imaging, autonomous navigation, surveillance, robotics).
- Introduce concepts of fairness, privacy, and explainability in AI-driven imaging solutions.

Course Outcomes:
After completing the course, the students should be able to:
1. Explain advanced imaging modalities and select appropriate imaging techniques for a given application.
2. Implement and fine-tune state-of-the-art computer vision algorithms and deep learning models for intelligent imaging tasks.
3. Design and execute end-to-end intelligent imaging pipelines, from sensor data acquisition to interpretation and decision-making.
4. Analyze and evaluate system performance using quantitative metrics, improve models through iterative experimentation, and ensure robustness and reliability.
5. Apply ethical considerations and understand regulatory frameworks governing intelligent imaging solutions.

Unit - I
Advanced Imaging Techniques and Modalities: Recap of standard 2D imaging and introduction to advanced modalities. Hyperspectral & Multispectral Imaging: Concepts, sensors, data cubes, and applications (agriculture, remote sensing). Light-field Imaging: Principles, plenoptic cameras, and depth estimation. 3D Imaging: Structured light, Time-of-Flight sensors, LIDAR, and stereo vision. Computational Photography: High Dynamic Range (HDR), panoramic stitching, synthetic aperture, and refocusing.

Unit - II

Intelligent Image Representation and Feature: Feature Representation: SIFT, SURF, ORB, and advanced local descriptors. Introduction to Deep Feature Extractors: Convolutional Neural Networks (CNNs), transfer learning, and feature maps. Dimensionality Reduction & Manifold Learning: PCA, t-SNE, UMAP for image embedding. Sparse and Dense Representations: Autoencoders, Variational Autoencoders (VAE) for image compression and generation.

Unit - III

Intelligent Image Interpretation via Machine Learning: Deep Learning Architectures for Vision: CNNs, ResNets, EfficientNets, Vision Transformers. Object Detection & Recognition: YOLO, Faster R-CNN, Mask R-CNN, DETR. Semantic and Instance Segmentation: U-Net, DeepLab, Segment-Anything Model. Explainability and Fairness in Vision: Grad-CAM, LIME, bias in datasets. Domain Adaptation and Transfer Learning: Fine-tuning pre-trained models for new tasks.

Unit - IV

Intelligent Imaging Systems and Applications: Embedded and Real-Time Vision Systems: Edge computing, NVIDIA Jetson. Industrial and Medical Applications: robotics navigation, autonomous vehicles, medical diagnostics. Performance Evaluation & System Optimization: Accuracy metrics, speed benchmarks, memory optimization, and model quantization.

Suggested Readings

1. Computer Vision: Algorithms and Applications by Richard Szeliski (Springer).
2. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods (Pearson).
3. Digital Image Processing by Bernd Jähne (Springer).

List of Experiments (Hardware on Breadboard / Software using NI Multisim)

1. Familiarization with OpenCV scripts
   Tasks: Run OpenCV scripts, explore TensorFlow
2. Advanced Imaging Modalities:
   Objective: Work with light-field and hyperspectral image samples.
   Tasks: Load hyperspectral datasets, perform spectral band selection, and reconstruct views from light-field data.
3. Computational Photography:
   Objective: Implement HDR merging and panoramic stitching.
   Tasks: Capture multiple exposure images using a camera module, merge into HDR; stitch overlapping images into a panorama.
4. Feature Extraction and Matching:
   Objective: Extract and match features for image registration or stitching.
   Tasks: Implement SIFT/SURF features, match keypoints between two images, visualize correspondences.
5. Deep Feature Representation:
   Objective: Compare handcrafted features with CNN-based features.
   Tasks: Extract features from a pre-trained CNN, use them for classification of a small dataset; compare accuracy with SIFT/SURF.
6. Object Detection and Recognition:
   Objective: Implement YOLO or Faster R-CNN on a given dataset.
   Tasks: Fine-tune a pre-trained object detector on a custom dataset (e.g., lab images of objects).

7. Segmentation Task:
   Objective: Implement semantic segmentation with U-Net or DeepLab.
   Tasks: Train a segmentation model on a subset of annotated images (e.g., medical or aerial images).
8. Explainable AI for Vision:
   Objective: Visualize model decisions.
   Tasks: Use Grad-CAM to highlight regions in an image that influence a CNN's classification.
9. Embedded Vision Implementation:
   Objective: Deploy a trained CNN model on an NVIDIA Jetson or Raspberry Pi.
   Tasks: Real-time object detection/inference with a camera feed, optimizing model size.
10. Capstone Project:
    Objective: Integrate multiple techniques into a cohesive intelligent imaging application.
    Tasks: Students work in teams to propose, design, and implement an end-to-end solution (e.g., a smart surveillance camera that detects intruders and segments foreground, or a spectral-based crop health analyzer).

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Generic Elective (GE) courses offered for Minors/ Specializations by the Department in Semester VI

Antenna and Wave Propagation (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Antenna and Wave Propagation	4	3	0	1	Physics, Mathematics – I, Electromagnetic Theory

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- Learn the basic parameters of an antenna and its radiation mechanism
- Understand the design and analyze of various wire antenna, and antenna arrays
- Understand the design and analyze of various broadband and planar antennas
- Learn the concept of aperture as well as high gain reflector antennas
- Express the basic concepts of ground, space, sky wave propagation mechanism

Course Outcomes:
After completing the course, the students should be able to:

1. Understand antenna characteristics for different applications.
2. Understand the concepts of Aperture and Slot Antennas
3. Analyze and design different types of antennas.
4. Design antenna arrays and understand operation of smart antennas.
5. Investigate different modes of propagation and their suitability for wireless communication.

Unit - I
Electromagnetic Radiation and Antenna Fundamentals- Review of Maxwell's equations: Retarded vector potential, Solution of wave equation in retarded case, Concept of radiation, Antenna equivalent circuits, Antenna characteristics: Radiation pattern, Beam solid angle, Radiation intensity, Directivity, Gain, Input impedance, Polarization, Bandwidth, Effective aperture, Antenna effective height, Antenna temperature.

Unit - II
Wire Antenna and Antenna Arrays-Wire antennas: Hertzian dipole, short dipole, Radiation resistance and Directivity, Half wave Dipole, Monopole, Small loop antennas. Antenna Arrays: Linear Array and Pattern Multiplication, Two-element Array, Uniform Array, Array with non-uniform Excitation, Binomial Array.

Unit - III

Special and Broad band Antennas-Special Antennas: Long wire, V and Rhombic Antenna, Yagi-Uda Antenna, Turnstile Antenna, Helical Antenna- Axial and Normal mode helix, Bi- conical Antenna, Frequency Independent Antenna, Log periodic Dipole Array, Spiral Antenna, Microstrip.

Aperture Antennas- Aperture Antennas: Slot antenna, Horn Antenna, Pyramidal Horn Antenna, Reflector Antenna- Flat reflector, Corner Reflector, Common curved reflector shapes, parabolic reflector, Lens Antenna, Patch Antennas.

Unit - IV

Radio Wave Propagation- Ground Wave Propagation, Free-space Propagation, Ground Reflection, Surface waves, Diffraction, Wave propagation in complex Environments, Tropospheric Propagation, Space waves, Ionosphere propagation: Structure of ionosphere, Skywaves, Skip distance, Virtual height, Critical frequency, MUF, Electrical properties of ionosphere, Effects of earth's magnetic fields, Faraday rotation.

Modern Antennas- Phase Array Antennas, Smart Antennas for Mobile Communication, MIMO Antennas for 5G Communication System, Reconfigurable Antenna

Suggested Readings

1. Antennas by J.D. Kraus (McGraw Hill).
2. Antenna Theory - Analysis and Design by C.A. Balanis (John Wiley).
3. Antennas and Radio Wave Propagation by R.E. Collin (McGraw Hill).
4. Antenna Engineering Handbook by R.C. Johnson and H. Jasik (McGraw Hill).
5. Electromagnetic Waves by R.K. Shevgaonkar (McGraw Hill).
6. Antenna Theory and Design by Stutzman, W.L. and Thiele, H.A (John Wiley & Sons).
7. Electromagnetic waves and Radiating Systems by E.C. Jordan and Balmain (Pearson Education).

List of Experiments

1. Electric Field Mapping: To map the electric field lines around different charge distributions and study the concept of electric flux density and Gauss's law.
2. Transmission Line Parameters: To measure the characteristic impedance and propagation constant of transmission lines and study the wave propagation in lossless and conducting media.
3. Electromagnetic Wave Propagation: To study the reflection and refraction of plane waves at the interface of different dielectric materials and measure the depth of penetration of electromagnetic waves.
4. Polarization of Electromagnetic Waves: To study the polarization states of electromagnetic waves and demonstrate linear, circular, and elliptical polarization.
5. Impedance measurement and Frequency measurement using rectangular waveguide.
6. Using simulation software
   a) Plot radiation pattern of dipole antenna
   b) Plot radiation pattern of monopole antenna
   c) Wire Antenna and Antenna Arrays-Wire antennas
   d) Reflector Antenna
   e) MIMO Antennas for 5G Communication System
   f) Reconfigurable Antenna

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Elements of Wireless Communication (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Elements of Wireless Communication	4	3	0	1	Mathematics – II, Signals and Systems

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To introduce the fundamental concepts and techniques of wireless communication systems.
- To provide an understanding of wireless channel characteristics and modulation techniques.
- To familiarize students with wireless network protocols and standards.
- To equip students with hands-on experience in designing and analyzing wireless communication systems.

Course Outcomes:
After completing the course, the students should be able to:
1. Explain the basic principles of wireless communication systems and technologies.
2. Analyze wireless channel characteristics and their impact on communication performance.
3. Apply modern modulation and multiple access techniques for wireless systems.
4. Design and simulate wireless communication links and networks using software tools.
5. Understand the evolution of wireless standards and the basics of current and emerging technologies.

Unit - I
Introduction to Wireless Communication: Overview of wireless communication systems. Evolution of wireless technologies (1G to 6G, IoT). Basic structure of wireless communication systems: Transmitter, channel, receiver. Key performance metrics: Bandwidth, latency, throughput, and reliability. Introduction to frequency spectrum and channel allocation.

Unit - II
Wireless Channel Characteristics: Propagation models: Free space, two-ray, and multipath. Large-scale fading: Path loss and shadowing. Small-scale fading: Types, Doppler effect, and coherence time. Rayleigh and Rician Fading. Channel capacity and diversity techniques.

Unit - III
Modulation and Multiple Access Techniques: Review of Digital modulation techniques: shift keying and QAM. Spread spectrum techniques: DSSS and FHSS. OFDM. FEC Techniques: Convolutional and Turbo Codes. Multiple access techniques: FDMA, TDMA, CDMA, and OFDMA.

Unit - IV
Wireless Standards and Networks: Overview of wireless communication standards: GSM, CDMA2000, LTE, 5G. Basics of WLAN (Wi-Fi), Bluetooth, ZigBee, and LoRaWAN. Introduction to MIMO systems (Role in Wireless and Beamforming). Future trends in wireless communication: mmWave, IoT, and 6G.

Suggested Readings

1. Modern Digital and Analog Communication Systems by B.P. Lathi (Oxford).
2. Communication Systems by Simon Haykin (Wiley).
3. Digital Communications by John G. Proakis (McGraw-Hill).
4. Wireless Digital Communications by K. Feher (Prentice Hall).
5. Wireless Communications by Andrea Goldsmith (University Press).
6. Wireless Communications: Principles and Practice by Theodore S. Rappaport (Pearson).
7. Fundamentals of Wireless Communication by David Tse and Pramod Viswanath (Cambridge).
8. Modern Wireless Communications by Simon Haykin and Michael Moher (Pearson).
9. Wireless Communications and Networks by William Stallings (Pearson).

List of Experiments

1. Set up a basic wireless link using SDR and measure signal strength.
2. Analyze wireless channel allocation in a real environment using Wireshark.
3. Experiment with indoor and outdoor path loss measurements.
4. Simulate Rayleigh and Rician fading channels using MATLAB.
5. Simulate and compare OFDM with single-carrier modulation using MATLAB or Python.
6. Implement a basic FEC system using coding algorithms in MATLAB or Python.
7. Simulate a MIMO system using MATLAB and measure capacity gains.
8. Design a basic IoT communication system using LoRa or Zigbee modules.

Hardware Requirements:

1. Software-Defined Radio (SDR) kits (e.g., RTL-SDR, HackRF, or USRP).
2. Zigbee and LoRa modules for IoT experiments.
3. NodeMCU or Raspberry Pi for IoT applications.
4. Antennas for diversity and beamforming experiments.
5. Spectrum analyzers for propagation and channel measurements.

Software Requirements:

1. MATLAB/Simulink or GNU Octave for simulation.
2. GNURadio for SDR programming.
3. Python with libraries such as NumPy, SciPy, and Matplotlib.
4. Wireshark for network protocol analysis.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Satellite Communication (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Satellite Communication	4	3	0	1	Mathematics – I, Electromagnetic Theory

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To provide in-depth knowledge of satellite communication principles, system design, and applications.
- To enhance problem-solving skills and foster innovation through hands-on learning in satellite communication systems.
- To familiarize students with advanced technologies and emerging trends in satellite networks.
- To prepare students for careers in telecommunications, aerospace, and related industries.

Course Outcomes:
After completing the course, the students should be able to:
1. Apply the principles of orbital mechanics and satellite subsystems in real-world satellite communication systems.
2. Design and analyze satellite links, including uplink, downlink, and link budgets, with a focus on performance optimization.
3. Evaluate multiple access techniques and advanced technologies in satellite networks.
4. Develop skills in designing, testing, and troubleshooting satellite communication systems using simulation tools and hardware.

Unit - I
Fundamentals of Satellite Communication and Orbital Mechanics Satellite types and applications. Orbital mechanics: Orbital elements, equations of motion, orbital perturbations. Satellite constellations and inter-satellite links. Look angle determination, limits of visibility, eclipse, and sun transit outage. Spacecraft technology: Structural design, power systems, attitude and orbit control, thermal management, propulsion, telemetry, tracking, command, and communication subsystems. Launch vehicles and procedures.

Unit - II
Satellite Link Design and Earth Station Technology Earth station architecture: Transmitters, receivers, and antenna systems. Basic transmission theory: Uplink and downlink design. Link budget analysis: Eb/No calculations, system noise, and intermodulation effects. Performance impairments: Propagation characteristics and reliability. Case studies: Design for IMMARSAT, INTELSAT, and other satellites.

Unit - III
Multiple Access Techniques and Optical Communication in Satellites FDMA, TDMA, CDMA: Concepts, system design, and comparison. Satellite switch TDMA, SPADE systems, and backoff considerations. Optical communication in satellite networks: Inter-satellite links, laser communication, beam acquisition, tracking, and pointing. Advanced topics: Quantum communication via satellites.

Unit - IV

Packet satellite networks and services, fixed satellite services, broadcast satellite services, mobile satellite services- VSAT, GPS, maritime satellite services, gateways, ATM over satellite, role of satellite in future network. Integration of satellites in next-gen networks: 5G/6G and IoT.

Suggested Readings

1. Satellite Communications by Timothy Pratt and Jeremy Allnutt (Wiley).
2. Satellite Communication by Dennis Roddy (McGraw Hill).
3. Satellite Communications by Varsha Agrawal, Anil K. Maini (Wiley).
4. Digital Satellite Communications by Tri T. Ha (Tata McGraw Hill).

List of Experiments

1. Understanding Orbital Mechanics
   Objective: Calculate orbital elements and simulate satellite orbits using MATLAB/Simulink.
2. Satellite Link Budget Analysis
   Objective: Design uplink and downlink with link budget calculations.
3. Antenna Design and Analysis for Earth Stations
   Objective: Design and test parabolic and horn antennas for satellite communication.
4. Modulation Techniques for Satellites
   Objective: Implement and test QPSK, BPSK, and QAM modulation schemes.
5. Simulation of FDMA, TDMA, and CDMA Systems
   Objective: Simulate and compare multiple access techniques.
   Requirements: MATLAB, NS3, or similar simulators.
6. Laser Communication System Simulation
   Objective: Design and analyze optical communication links for satellites.
   Requirements: Optical simulation tools (OptiSystem), photodetectors, laser diodes.
7. Satellite IoT Network Implementation
   Objective: Design a small satellite network for IoT applications using LoRaWAN.
   Requirements: LoRa modules,
8. AI in Satellite Communication
   Objective: Implement an AI-based fault detection system for satellite telemetry data.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Cognitive Radio & Networks (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Cognitive Radio & Networks	4	3	0	1	Signals and Systems, Wireless Communication and Mobile Networks

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Understand the principles of cognitive radio, including spectrum scarcity, dynamic spectrum access, and the cognitive cycle.
- Analyze spectrum sensing and management techniques and their application in dynamic spectrum sharing.
- Explore advanced concepts such as cross-layer design, OFDM, and MIMO systems in cognitive radio networks.
- Gain practical skills through experiments and understand the regulatory frameworks and emerging technologies in cognitive radio.

Course Outcomes:
After completing the course, the students should be able to:
1. Demonstrate an understanding of cognitive radio fundamentals, including its architecture and functionalities.
2. Apply spectrum sensing and sharing techniques to design efficient cognitive radio systems.
3. Analyze regulatory standards and integrate advanced technologies like OFDM and UWB into cognitive radio networks.
4. Develop practical skills for implementing and evaluating cognitive radio systems using hardware and software tools.

Unit - I
Introduction to Cognitive Radio and Networks: spectrum scarcity and Spectrum white space, Fixed spectrum allocation, Software Defined Radio (SDR): Concept, Limitations, Evolution to Cognitive Radio. Dynamic Spectrum Access and Cognitive Cycle, Functions of Cognitive Radio: Spectrum Sensing, Spectrum Management, Spectrum Mobility, Cognitive Radio Architecture.

Unit - II
Spectrum Sensing and Management: Hypothesis Model for Spectrum Sensing. Types of Spectrum Sensing: Non-cooperative Sensing, Cooperative Sensing, Interference-based Sensing. Detection Techniques: Matched Filter Detection, Energy Detection, Cyclostationary Feature Detection. Dynamic Spectrum Access (DSA): Models and Architectures, Opportunistic Spectrum Access (OSA).

Unit - III
Cognitive Radio Networks (CRN): Cognitive Radio Cross-Layer Design: Adaptation and Optimization Security Challenges in Cognitive Radio Networks, MIMO Systems, Smart Antennas, and Beamforming in Cognitive Radio, OFDM-based Cognitive Radio: suitability, challenges, and Multiband OFDM. Standards and Technologies for Cognitive-OFDM.

Unit - IV

Standardization, Regulation, and Emerging Technologies: Regulatory Issues and New Spectrum Management Regimes. Spectrum Planning and Authorization. Standards. Ultra-Wideband (UWB) Cognitive Radio: Fundamentals of Impulse Radio UWB. Cognitive Radio Requirements vs. IR-UWB. Merging Impulse Radio with Cognitive Radio.

Suggested Readings

1. Dynamic Spectrum Access and Spectrum Management in Cognitive Radio Networks by Ekram Hossain, Dusit Niyato, and Zhu Han, 1st Edition (Cambridge University Press, 2009)
2. Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems by Huseyin Arslan, 1st Edition (Springer, 2007)
3. Cognitive Radio Technology by Bruce A. Fette, 2nd Edition (Academic Press, 2009)
4. Suggested Reference Books:
5. Cognitive Radio Networks by Yang Xiao and Fei Hu, 1st Edition (CRC Press, 2008)
6. Essentials of Cognitive Radio by Linda E. Doyle, 1st Edition (Cambridge University Press, 2009)
7. Cognitive Radio Communications and Networks by Alexander M. Wyglinski, Maziar Nekovee, and Thomas Hou (Academic Press)
8. Wireless Communications: Principles and Practice by Theodore S. Rappaport (Pearson)
9. Principles of Modern Wireless Communication Systems by Aditya K. Jagannatham (McGraw Hill)
10. Other Useful Resources:
11. GIAN IIT Kanpur course contents

<u>List of Experiments</u>

1. Basic SDR Implementation Using GNU Radio and USRP
   Objective: Demonstrate dynamic spectrum access and identify available frequency bands using SDR hardware.
   Requirements: GNU Radio software, USRP hardware, antennas, signal source.
2. Spectrum Sensing Using GNU Radio
   Objective: Implement and compare energy detection and cyclostationary feature detection.
   Requirements: GNU Radio, USRP, spectrum analyzer.
3. OFDM-Based Cognitive Radio Simulation
   Objective: Model an OFDM-based cognitive radio system and study the impact of spectrum fragmentation and interference on system performance.
4. Cooperative Spectrum Sensing
   Objective: Evaluate performance improvements with distributed sensing.
   Requirements: Multiple USRP devices, GNU Radio.
5. Dynamic Spectrum Access
   Objective: Simulate dynamic spectrum access and interference management.
   Requirements: NS-3, MATLAB.
6. MAC Protocol Simulation
   Objective: Design and analyze a spectrum-aware MAC protocol.
   Requirements: NS-3, Python.
7. QoS Optimization
   Objective: Implement algorithms to enhance QoS in CRNs.
   Requirements: Python, MATLAB.
8. 5G Cognitive Radio Simulation
   Objective: Simulate the integration of cognitive radio in 5G networks.
   Requirements: MATLAB, NS-3.

9. Real-Time Communication System
   Objective: Prototype a small-scale cognitive radio communication system.
   Requirements: USRP, GNU Radio.
10. Energy Efficiency Analysis
    Objective: Measure and optimize energy consumption in cognitive radios.
    Requirements: MATLAB, power meters.
11. Smart Antenna and Beamforming Experiment
    Objective: Design and test a basic MIMO system with smart antennas for efficient spectrum sharing.
    Requirements: SDR platform, antenna arrays, MATLAB/Python for algorithm testing.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

CMOS Analog IC Design (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

<table>
<tr><th rowspan="2">Course Title</th><th rowspan="2">Credits</th><th colspan="3">Credit Distribution of the Course</th><th rowspan="2">Prerequisite of the course (if any)</th></tr>
<tr><th>Lecture</th><th>Tutorial</th><th>Practical</th></tr>
<tr><td>CMOS Analog IC Design</td><td>4</td><td>3</td><td>0</td><td>1</td><td>Physics, Electronic Devices and Circuits, Digital Electronics – I, Linear Integrated Circuits</td></tr>
</table>

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To understand the principles and building blocks of CMOS analog ICs.
- To develop skills in analyzing and designing CMOS amplifiers and op-amps.
- To explore advanced CMOS architectures and nonlinearity cancellation techniques.
- To design and implement MOSFET-C circuits for filters, oscillators, and multipliers.
- To apply switched-capacitor design principles in practical applications.
- To gain hands-on experience with simulation tools and circuit optimization.
- To foster innovative problem-solving in analog IC design.

Course Outcomes:
After completing the course, the students should be able to:
1. Demonstrate a thorough understanding of CMOS device characteristics and their relevance in analog integrated circuits.
2. Analyze and design CMOS analog building blocks such as current mirrors, amplifiers, and op-amps, including their frequency response and large/small signal behavior.
3. Design and analyze CMOS operational amplifiers (op-amps) and operational transconductance amplifiers (OTAs) with specific emphasis on performance metrics like gain bandwidth, slew rate, and impedance.
4. Apply MOSFET-C circuit design techniques to create filters, oscillators, and multipliers with linear and nonlinear functionalities.
5. Design and simulate switched-capacitor circuits including universal filters, VCOs, and integrators, with a focus on practical implementation challenges.
6. Utilize modern simulation tools to evaluate and optimize the performance of CMOS analog circuit designs.

Unit - I
Basic building blocks of CMOS analog ICs: Review of PMOS, NMOS and CMOS devices, characteristics and circuit models; Importance of CMOS analog circuits; diode-connected MOSFET; Current sources and current mirrors (basic, Wilson, Cascode and others); NMOS and PMOS differential amplifiers with active loads and their large signal analysis; small signal analysis; determination of Ad, Ac and CMRR; CS amplifier as a gain stage; Cascode amplifier and its frequency response analysis; Differential to single-ended converter; Source follower as output stage and level shifter; the IC op-amp architecture.

Unit - II
CMOS op-amps and OTAs: Various CMOS op-amp architectures; Two-stage CMOS op-amp design; large signal and small signal analysis; frequency response analysis-determination of gain bandwidth product, slew rate, Zin and Zout; Basic CMOS OTAs; Non-linearity cancellation techniques; Design of CMOS linear transconductors.

Unit - III

Fully-integrable MOSFET-C Circuits: The basic topologies of nonlinearity cancellation in MOSFET circuits and their analysis; fully differential continuous-time MOSFET-C filters based on CMOS op-amps; MOSFET-C filters design using CMOS Current Conveyors and Current feedback op-amps; MOSFET-C oscillators; CMOS analog multipliers and other linear/nonlinear functional circuits.

Unit - IV

MOS Switched-capacitor networks: The switched-capacitor resistor equivalence; Switched- capacitor integrator; generation of two-phase non-overlapping clock; basic building blocks of switched-capacitor networks; Switched-Capacitor universal biquad filter design; Switched-capacitor filter design using generalised impedance converter (GIC) and bilinear transform; design of the switched-capacitor linear VCOs.

Suggested Readings

1. Design of Analog CMOS Integrated Circuits by Razavi (McGraw-Hill Education).
2. CMOS Analog Circuit Design by Allen and Holberg (Oxford University Press).
3. Analysis and Design of Analog Integrated Circuits by Gray and Meyer (John Wiley & Sons).
4. Analog Filter Design by Schoumann and Van Valkenburg (Oxford University Press).

List of Experiments

1. Evaluation of the performance of various MOS current mirrors
2. SPICE simulation and studies of (i) differential to singe ended converter and (ii) level shifter
3. Frequency response analysis of CS amplifier and cascade amplifier
4. Evaluation of the performance of a two-stage CMOS op-amp
5. Simulation of a CMOS linear transconductor
6. Design of a second order switched-capacitor filter
7. Design of a Switched-capacitor VCO
8. MOSFET-C biquad filter design and simulation
9. Design and simulation of a MOSFET-C VCO
10. Simulation of the CMOS Gilbert Multiplier

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Current Mode Analog VLSI Circuits (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Current Mode Analog VLSI Circuits	4	3	0	1	Physics, Electronic Devices and Circuits, Linear Integrated Circuits

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- Develop a strong theoretical foundation in the principles and advantages of current-mode circuit design compared to traditional voltage-mode circuits.
- Enable students to design, analyze, and implement various current-mode circuits, such as multipliers, dividers, squarers, and other translinear circuits.
- Study and apply various current-mode building blocks like Current Conveyors (CCs) and Current Feedback Op-Amps (CFOAs).
- Equip students with the ability to utilize CMOS technologies for implementing current-mode circuits, addressing challenges like nonlinearity and achieving high performance.
- Explore the concepts and practical implementations of fully integrable current-mode filters and dynamic translinear circuits.

Course Outcomes:

After completing the course, the students should be able to:

1. Evaluate the advantages and limitations of current-mode circuits over traditional voltage-mode circuits.
2. Apply the translinear principle to design and synthesize circuits like analog multipliers, dividers, and square-root circuits.
3. Implement and analyze current conveyors and their applications in practical analog designs.
4. Analyze and design CMOS-based current-mode circuits with nonlinearity cancellation and explore applications in analog signal processing.
5. Design and synthesize fully integrable current-mode filters, including log-domain and square-root domain filters.
6. Use current-mode design principles to tackle complex challenges in analog VLSI design, contributing to advanced IC development.

Unit - I

Voltage-mode versus Current-mode circuits; advantages of current mode circuit design; Current-mode circuits from voltage-mode op-amps using supply-current sensing techniques; the operational mirrored amplifiers (OMA) and their applications; The Translinear principle and circuits; static translinear circuit design- analog multipliers, analog dividers, squarers, square-rooters; geometric mean and harmonic mean circuits; the compete translinear squarer design; Synthesis of a 4-quadrant current-mode analog multiplier.

Unit - II

Current-mode analog building blocks: The current Conveyors and their variants; Current-feedback op-amp; the mixed translinear cell; the translinear implementation of CCs and CFOAs; the gain bandwidth conflict resolution and attainment of high slew rates; the basic analog circuits realisable from CCs and CFOAs; Examples of IC CCs and CFOAs; Miscellaneous current-mode building blocks.

Unit - III

CMOS current-mode circuits: Review of MOS Current mirrors; the translinear principle for MOS circuits; CMOS mixed translinear cell; CMOS implementation of CCs and CFOAs; Nonlinearity cancellation in MOS analog circuits and their applications; CMOS Voltage-controlled linear resistors; CMOS nonlinear functional circuits.

Unit - IV

Fully- integratable Current-mode filters: The dynamic translinear circuits; Log domain and translinear filters; MOS geometric mean circuits, squarers and squarer/divider circuits; Introduction to fully-integratable square-root domain CMOS analog filters.

Suggested Readings

1. Analog IC Design: The Current-Mode Approach by Toumazou, Lidgey, and Haigh (Institution of Engineering and Technology, 1990).
2. Integrated Circuits for Analog Signal Processing by Esteban Tlelo-Cuautle (Springer, 2009).
1. Analog Integrated Circuit Design by Tony Chan Carusone, David Johns, and Kenneth Martin (Wiley, 2011).
2. Design of Analog CMOS Integrated Circuits by Razavi (McGraw-Hill Education, 2017).
3. Analog VLSI: Circuits and Principles by Shih-Chii Liu, Jörg Kramer, Giacomo Indiveri, Tobias Delbrück, and Rodney Douglas (MIT Press, 2002).

<u>List of Experiments</u>

1. Realisation of Current followers using operational-mirrored amplifier
2. Realisation of constant-bandwidth, variable gain amplifiers using AD844 CFOA
3. Design of a Translinear Squarer and verification of its operation
4. Realisation of a CMOS linear VCRs
5. CMOS Gilbert multiplier and determination of its performance
6. Current-mode analog biquad filter design using AD844 CFOA
7. Design and verification of an instrumentation amplifier using CC/CFOA
8. Design of a single resistance controlled oscillator using CFOAs
9. Design of a log domain low pass filter
10. Design of a CMOS square-root domain all pass filter

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Ubiquitous Sensing, Computing and Communication (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Ubiquitous Sensing, Computing and Communication	4	3	0	1	Fundamentals of Computer Programming

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- To provide an in-depth understanding of ubiquitous sensing, computing, and communication in IoT systems.
- To explore advanced topics such as context-aware computing, location-based services, and mobile applications.
- To understand the integration of IoT with data analytics and real-time processing for actionable insights.

Course Outcomes:
After completing the course, the students should be able to:

1. Demonstrate knowledge of ubiquitous sensing and computing frameworks.
2. Design and develop context-aware and location-aware IoT systems.
3. Analyze IoT data using modern analytics tools for decision-making and problem-solving.

Unit - I
Fundamentals of Ubiquitous IoT Systems: Introduction to IoT and Ubiquitous Computing. Overview of IoT Networking Basics: NFC, Wireless LAN, LPWAN. Challenges in Ubiquitous IoT Systems. Fundamentals of Location-aware Computing: Concepts, Architecture, and Applications. Location-based Services (LBS): Social Networks, Recommendations, and Real-World Applications.

Unit - II
Context-Aware and Mobile Ubiquitous Computing: Context and Context-Aware Computing: Definitions and Challenges. Developing Context-Aware Applications: System Architecture and Implementation. Wearable Computing: Devices and Applications (Smart Glasses, Augmented Reality, Digital Pen). Mobile Social Networking and Crowd Sensing: Case Studies. Event-Based Social Networks and Applications.

Unit - III
Privacy, Security, and Energy in Ubiquitous IoT: Privacy and Security Challenges in Ubiquitous IoT Systems. Energy Constraints in IoT and Strategies for Efficiency. Human Activity and Emotion Sensing for Mobile Applications. Smart Homes and Intelligent Buildings: Design and Applications. Mobile Peer-to-Peer Computing and Cloud-Centric IoT.

Unit - IV
IoT Data Management and Analytics: IoT Data Management: Cleaning, Processing, and Storage Models. Advanced Search Techniques: Semantic Sensor Web, Deep Web, and Semantic Web Data Management. Real-Time and Big Data Analytics for IoT: High-Dimensional and Heterogeneous Data Processing. Parallel and Distributed Data Processing: Techniques and Tools. QoS and QoE in IoT: Concepts and Protocols.

Suggested Readings

1. Ubiquitous Computing Fundamentals by John Krumm (CRC Press).
2. Internet of Things: Principles and Paradigms by Rajkumar Buyya and Amir Vahid Dastjerdi (Morgan Kaufmann).
3. Real-Time Analytics by Byron Ellis (Wiley)
4. Ubiquitous Computing and Computing Security of IoT by N. Jeyanthi, Ajith Abraham, and Hamid Mcheick (Springer).
5. Enterprise IoT by Dirk Slama (O'Reilly Publisher).

List of Experiments

1. Interfacing DHT11 and soil moisture sensor with Arduino/ESP32.
   Hardware: Arduino/ESP32, Sensors.
2. Uploading sensor data to ThingSpeak and creating real-time dashboards.
   Software: ThingSpeak, Python.
3. Deploying a simple ML model on Raspberry Pi for anomaly detection.
4. Setting up LoRa modules for data transmission.
5. AIoT Application: Using Google Colab for predictive analytics with IoT data.
6. NFC Communication: Implementing a basic NFC-based interaction between devices.
7. Location-Aware Application: Developing a GPS-based location tracking system for personal assistants.
   Hardware: Raspberry Pi, GPS module.
8. Smart Home System: Developing a prototype for smart home automation using NodeMCU and MQTT.
   Hardware: NodeMCU, sensors (temperature, motion), actuators (relay).

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Introduction to Embedded Systems for IoT (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Introduction to Embedded Systems for IoT	4	3	0	1	Fundamentals of Computer Programming

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- To understand the purpose, architecture, and design principles of embedded systems in IoT.
- To develop proficiency in interfacing sensors, actuators, and communication modules with embedded platforms.
- To explore IoT-enabling technologies and integrate them with cloud platforms for real-world applications.

Course Outcomes:

After completing the course, the students should be able to:

1. Design embedded systems tailored for IoT applications.
2. Interface IoT devices with sensors, actuators, and other hardware components.
3. Implement communication protocols and cloud integration for IoT solutions.

Unit - I

Fundamentals of Embedded IoT Systems: Introduction to Embedded IoT: Purpose and Requirement Specification. IoT Level Specifications: Functional View, Operational View. Device and Component Integration: Overview and Challenges. Pillars of Embedded IoT: The Internet of Devices and IoT Levels.

Unit - II

Design of Embedded Systems: Core Components: Common Sensors and Actuators. Embedded Processors and Microcontroller Architectures. Memory Architectures: Flash, EEPROM, and RAM. Software Architectures: RTOS vs Bare-Metal Programming.

Unit - III

Inputs and Outputs in Embedded Systems: Digital Inputs/Outputs: GPIO, BusIn, BusOut, BusInOut. Analog Inputs/Outputs: ADC and DAC Interfacing. Pulse Width Modulation (PWM) for Motor and Light Control. Advanced Peripherals: Accelerometer, Magnetometer, SD Card Integration, and File Systems. Sensor and Actuator Integration: Temperature, Humidity, Motion Sensors, and Relays.

Unit - IV

IoT-Enabling Technologies: Communication Technologies: RFID, NFC, Bluetooth Low Energy (BLE), LiFi, ZigBee, Z-Wave, LoRa, and 6LoWPAN. IoT Protocols: HTTP, WebSocket, MQTT, CoAP, XMPP. IoT Platforms: Node-RED, IBM Watson IoT, AWS IoT, Microsoft Azure IoT, Google Cloud IoT, ThingWorx.

Suggested Readings

1. Embedded Systems: Architecture, Programming, and Design by Raj Kamal (McGraw-Hill Education).
2. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things by David Hanes et al. (Cisco Press).
3. Exploring Arduino: Tools and Techniques for Engineering Wizardry by Jeremy Blum (Wiley).
4. Mastering Internet of Things by Peter Waher (Packt Publishing).
5. Internet of Things: Principles and Paradigms by Rajkumar Buyya and Amir Vahid Dastjerdi (Elsevier).

List of Experiments

1. Digital I/O: Blinking LEDs and reading button input using ESP32.
2. Analog Interfacing: Reading sensor data (e.g., potentiometer) and controlling LEDs with PWM.
3. Sensor Integration: Interfacing temperature and motion sensors to display readings on an LCD.
4. IoT Protocol Implementation: Sending data to a cloud platform (ThingSpeak) using MQTT.
5. Cloud Integration: Building a weather monitoring system with live data visualization on AWS IoT.
6. Communication Technologies: Setting up BLE communication between ESP32 and a smartphone.
7. File Systems: Implementing data logging on an SD card and reading the logged data.
8. Smart Home Prototype: Developing a web interface to control home appliances using ESP32.
9. Node-RED: Creating a flow-based application for monitoring IoT device data.
10. Energy Efficiency Analysis: Measuring power consumption of IoT devices under different workloads.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Deep Learning for Image Analysis (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Deep Learning for Image Analysis	4	3	0	1	Fundamentals of Computer Programming, Mathematics – II, Intelligent Imaging

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To provide in-depth knowledge of advanced deep learning architectures and their application to complex image analysis tasks.
- To enable students to understand and implement state-of-the-art object detection, segmentation, and generative models.
- To introduce vision transformers, self-supervised learning, and other frontier topics in deep learning for images.
- To equip students with practical skills for model training, optimization, deployment, and interpretation in real-world scenarios.

Course Outcomes:
After completing the course, the students should be able to:
1. Understand and critically evaluate advanced deep neural network architectures for image-related tasks.
2. Implement and fine-tune cutting-edge models for object detection, segmentation, and generative image synthesis.
3. Explore and apply vision transformers and self-supervised techniques to improve model performance and reduce labeling costs.
4. Employ interpretability methods, perform model compression, and deploy models on resource-constrained devices.
5. Design and conduct experiments to address real-world image analysis challenges, iterating through data preparation, model training, validation, and optimization.

Unit - I
Advanced CNN Architectures: Revisiting ResNet, DenseNet, MobileNet, EfficientNet. Training Optimization Techniques. Regularization & Generalization: DropBlock, label smoothing, and large-batch training strategies. Hyperparameter Tuning & Transfer Learning. Distributed & Parallel Training: Introduction to multi-GPU training, mixed precision, and cloud-based training pipelines.

Unit - II
Object Detection Models: Faster R-CNN, YOLO families, RetinaNet - architecture insights and training approaches. Image Segmentation: U-Net, Mask R-CNN, and newer variants for medical and satellite imaging. Vision Transformers (ViT): Fundamentals of transformer architecture applied to images, training and fine-tuning ViTs, comparison with CNNs. Advanced Topics in Detection & Segmentation: Handling imbalanced datasets, small object detection, and domain adaptation.

Unit - III

Generative Adversarial Networks (GANs): StyleGAN, CycleGAN, and conditional GAN variants for image generation and domain translation. Variational Autoencoders (VAEs): Concept and applications in image compression and feature disentanglement. Self-Supervised & Semi-Supervised Learning: SimCLR, MoCo, BYOL, DINO - methods to leverage unlabeled data, pretext tasks, and contrastive learning.

Unit - IV

Model Interpretability & Explainability: Grad-CAM, Guided Backprop, SHAP, LIME for understanding decision-making processes. Model Compression & Acceleration. Edge & Embedded Vision: Implementing lightweight models on GPUs, TPUs, and microcontrollers (e.g., NVIDIA Jetson, Google Coral).

Suggested Readings

1. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (MIT Press).
2. Deep Learning with Python by François Chollet (Manning).
3. CS231n Notes by Justin Johnson and Fei-Fei Li (Online).
4. Deep Residual Learning for Image Recognition by He et al. (ResNet).
5. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale by Dosovitskiy et al. (ViT).
6. U-Net: Convolutional Networks for Biomedical Image Segmentation by Ronneberger et al.
7. Generative Adversarial Networks by Goodfellow et al. (Original GAN Paper).
8. A Simple Framework for Contrastive Learning of Visual Representations by Chen et al. (SimCLR).

<u>List of Experiments</u>

1. CNN Fine-Tuning
   Objective: Fine-tune a pre-trained EfficientNet on a custom dataset.
      a. Implement data augmentation strategies (mixup, cutmix)
      b. Optimize hyperparameters using grid search or Bayesian optimization
2. Object Detection with YOLOv5
   Objective: Implement and train YOLO on a small custom dataset (e.g., campus surveillance images).
      a. Annotate a small dataset using labeling tools
      b. Train YOLOv5 and evaluate precision-recall, mAP
      c. Attempt domain adaptation (e.g., day vs. night images)
3. Image Segmentation using U-Net
   Objective: Segment objects in microscopy or satellite imagery.
      a. Train U-Net on a medical imaging dataset (e.g., cell nuclei segmentation)
      b. Evaluate with IoU, Dice scores
      c. Experiment with augmentations to handle class imbalance
4. Vision Transformers
   Objective: Fine-tune a ViT model on a small dataset.
      a. Use Hugging Face Transformers library
      b. Compare ViT performance vs. a standard CNN on the same dataset
5. GAN-based Image Generation
   Objective: Train a DCGAN or StyleGAN to generate synthetic images (e.g., anime faces, fashion items).
      a. Implement GAN training loop
      b. Explore latent space manipulations

6. Self-Supervised Learning
   Objective: Implement SimCLR to learn image embeddings without labels.
   a. Pre-train model using contrastive learning on unlabeled dataset
   b. Fine-tune on a small labeled subset
7. Model Interpretability
   Objective: Use Grad-CAM to visualize which regions of an image influence model predictions.
   a. Implement Grad-CAM on a trained model
   b. Analyze differences between classes and confirm correctness of reasoning
8. Model Compression and Deployment
   Objective: Compress and run a trained model on a Jetson Nano or Raspberry Pi.
   a. Apply pruning or quantization
   b. Deploy model and compare inference speed before and after compression

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Deep & Reinforcement Learning (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Deep & Reinforcement Learning	4	3	0	1	Fundamentals of Computer Programming, Mathematics – II, Intelligent Imaging

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Introduce the foundational concepts of Reinforcement Learning (RL), including Markov Decision Processes and basic RL methods.
- Demonstrate how deep learning can enhance RL, enabling agents to handle high-dimensional state spaces.
- Cover essential deep RL algorithms like DQN, policy gradients, and PPO, making students comfortable in both theory and coding.
- Expose students to practical considerations such as stability, exploration strategies, and environment setup.
- Allow learners to experiment with state-of-the-art tools and environments in a step-by-step manner, culminating in a final project.

Course Outcomes:
After completing the course, the students should be able to:
1. Understand core RL principles (policy, value function, reward) and how they map to real-world decision-making tasks.
2. Implement fundamental deep RL algorithms, from DQN to PPO, using modern frameworks.
3. Appreciate the challenges in training stable and efficient RL agents and employ techniques to address them.
4. Experiment with continuous action spaces and policy optimization methods.
5. Develop and present a small RL-based project, demonstrating the ability to apply learned techniques to a new problem.

Unit - I
Foundations of Reinforcement Learning: Basic Concepts: Agents, environments, states, actions, rewards, and returns. Markov Decision Processes (MDPs): Defining the RL problem, Bellman equations. Value-based Methods (Tabular): State-value and action-value functions, Q-learning basics. Limitations of Tabular Methods: Why we need function approximation.

Unit - II
Introducing Deep Reinforcement Learning: Neural Network Function Approximators: Representing value functions with neural nets. Deep Q-Network (DQN): How DQN works, replay buffers, target networks. Improving DQN: Double DQN, Dueling Networks for stability and better performance.

Unit - III

Policy-Based and Actor-Critic Methods: Policy Gradients (REINFORCE): Learning policies directly, basics of gradient-based policy optimization. Actor-Critic Methods: Advantage Actor-Critic (A2C), motivation and benefits. Proximal Policy Optimization (PPO): Conceptual understanding of PPO, clipping to maintain stable updates. Continuous Action Spaces: Brief introduction to handling continuous controls.

Unit - IV

Exploration Strategies: Epsilon-greedy, entropy bonuses, curiosity-driven exploration. Stability and Debugging: Identifying training instability, tuning hyperparameters. Basic Model-Based and Multi-Agent Concepts (Introductory): Light overview of model-based RL and the idea of multi-agent settings (no deep dive). Ethical and Safety Considerations: Brief discussion on when and how RL can impact real-world scenarios.

Suggested Readings

1. Reinforcement Learning: An Introduction by Sutton and Barto (MIT Press).
2. Spinning Up in Deep RL by OpenAI (Documentation and Tutorials).
3. Playing Atari with Deep Reinforcement Learning by Mnih et al. (DQN).
4. Proximal Policy Optimization Algorithms by Schulman et al. (PPO).

List of Experiments

1. Tabular Q-Learning
   Objective: Implement Q-learning to navigate a simple gridworld environment.
2. Deep Q-Networks (DQN) for CartPole
   Objective: Train a neural network-based agent to balance a pole on a cart using Deep Q-Networks (DQN).
3. Enhancing DQN
   Objective: Apply Double DQN or Dueling DQN to improve learning stability.
4. Hyperparameter Tuning
   Objective: Explore the effect of changing learning rates, batch sizes, and gamma values on agent performance.
5. Cliff-Walking Problem
   Objective: Demonstrate the concept of state-value functions using a cliff-walking example.
6. Design a simple neural network to predict action probabilities.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)