NATIONAL INSTITUTE OF ELECTRONICS AND INFORMATION TECHNOLOGY

DEEMED TO BE UNIVERSITY UNDER DISTINCT CATEGORY

(AN AUTONOMOUS INSTITUTION UNDER MINISTRY OF ELECTRONICS & IT, GOVT. OF INDIA)

Curriculum and Syllabi for

B.Tech – Electronics and Communication Engineering (VLSI Design and Technology)

(2026-27 Scheme)

रा.इ.सू.प्रौ.सं

NIELIT

Deemed to be University

Main Campus at Ropar and Constituent Units at Aizawl, Agartala, Aurangabad, Calicut, Gorakhpur, Imphal, Itanagar, Kekri, Kohima, Patna and Srinagar

Vision

To be a nationally recognized center of excellence in VLSI Design and Technology, nurturing skilled, innovative, and ethically responsible engineers who drive advancements in semiconductor and integrated circuit design for the digital future.

Mission

1. To deliver quality education in Electronics and Communication Engineering with a specialized focus on VLSI Design and Embedded Systems.
2. To equip students with rigorous theoretical foundations and hands-on skills in digital, analog, and mixed-signal IC design.
3. To promote interdisciplinary research, innovation, and entrepreneurship in semiconductor technologies.
4. To instill professional ethics, teamwork, and communication skills for holistic engineering development.
5. To enable graduates to address industry and societal challenges through responsible and cutting-edge VLSI solutions.

Program Education Objectives

Graduates of the B.Tech (ECE – VLSI Design and Technology) program will:
1. Apply knowledge of electronics, mathematics, and VLSI principles to design integrated circuits for diverse application domains.
2. Pursue successful careers in the semiconductor industry, academia, or entrepreneurial ventures in VLSI and embedded systems.
3. Demonstrate professional ethics, leadership, and communication abilities in multidisciplinary engineering environments.
4. Engage in lifelong learning and advanced studies to remain current with evolving EDA tools and semiconductor technologies.
5. Contribute responsibly to societal and technological advancement through innovative IC design solutions.

Program Outcomes

Upon successful completion of the program, students will be able to:
1. Apply Engineering Knowledge: Apply mathematics, physics, and electronics principles to analyze and design complex VLSI systems.
2. Design VLSI Solutions: Design digital, analog, and mixed-signal integrated circuits satisfying functional and performance specifications.
3. Modern Tool Usage: Utilize EDA tools and hardware platforms for IC design, simulation, verification, and implementation.
4. Problem Analysis: Identify, formulate, and solve semiconductor design problems using appropriate tools and methodologies.
5. Ethics and Communication: Demonstrate ethical responsibility, teamwork, and effective technical communication skills.
6. Lifelong Learning: Engage in continuous learning and adapt to rapidly evolving semiconductor technologies and design practices.

Program Specific Outcomes

Graduates of the B.Tech (ECE – VLSI Design and Technology) program will be able to:
1. Design and implement digital, analog, and mixed-signal VLSI circuits using standard cell libraries and HDL languages.
2. Perform physical design, layout, and verification of integrated circuits using industry-standard EDA tools such as Cadence, Synopsys, and Mentor Graphics.
3. Develop and test FPGA-based prototypes and demonstrate understanding of the complete IC design flow from RTL to GDSII.

CATEGORY WISE CREDIT DISTRIBUTION

Category	Category Code	Credits
Audit Courses (without grade or credit)	AU	0
Value Added Courses	VA	7
Professional Elective Courses	PE	20
Ability Enhancement Courses	AE	8
Open Elective Courses	OE	12
Skill / Employment Enhancement Courses (Project/Internship/Seminar/Dissertation/Research)	EE	32
Program Core Courses	PC	88
Total Credits (Common track)		**167**

EVALUATION AND ASSESSMENT

1. Performance of a student in a semester shall be evaluated through continuous Class assessment, Tutorial/Lab assessment, Mid-Semester Examination (MSE) and End-Semester Examination (ESE). Both the MSE and ESE shall be the University examination and will be conducted as notified by the Controller of Examinations (CoE) of the University.

2. The continuous assessment shall be based on assignments, tutorials, paper presentation / quizzes/ viva-voce / flipped classes, lab work / projects / fieldwork and attendance, etc.

3. The MSE/ESE shall be comprising of written papers.

4. The overall assessment of the students will be done as per the following scheme:

S. No.	Assessment Type	Marks Weightage
1.	Mid Semester Examination	20
2.	End Semester Examination	40
3.	Continuous Assessment - Practical /Lab/ Tutorial	25
4.	Continuous Assessment - Theory	15
Total		**100**

For more details, please refer the applicable ordinance for this programme and Assessment SOPs.

CURRICULUM

Semester 1						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTECVL-101	DASH250038	Engineering Mathematics I	3	1	0	4
PC-BTECVL-102	DOEE260025	Fundamentals of Electronics	3	1	0	4
PC-BTECVL-103	DOAI250044	Probability and Statistical Techniques	3	1	0	4
EE-BTECVL-104	DCSE260001	Programming using C and Python	2	0	4	4
OE-BTECVL-105	Elective	Open Elective				4
AE-BTECVL-106	DASH250016	Communication Skills	1	0	2	2

Semester 2						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTECVL-201	DASH250039	Engineering Mathematics II	3	1	0	4
PC-BTECVL-202	DOEE250059	Digital Electronics	3	0	2	4
EE-BTECVL-203	DCSE250055	Data Structures	3	0	2	4
PE-BTECVL-204	Elective	Program Elective				4
OE-BTECVL-205	Elective	Open Elective				4
VA-BTECVL-206	DASH250101	Sports and Yoga	0	0	4	2

Semester 3						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTECVL-301	DOEE260024	Introduction to VLSI Technology	3	1	0	4
PC-BTECVL-302	DOEE260026	Electronics Devices and Circuits	3	1	0	4
EE-BTECVL-303	DOEE260001	HDL Programming: Verilog and VHDL	3	0	2	4
PE-BTECVL-304	Elective	Program Elective				4
OE-BTECVL-305	Elective	Open Elective				4
VA-BTECVL-306	DASH250105	Universal Human Values-II: Understanding Harmony And Ethical Human Conduct	3	0	0	3
AU-BTECVL-307	Elective	Audit Elective				0

Semester 4						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTECVL-401	DOEE250050	Control System Engineering	3	1	0	4
PC-BTECVL-402	DOEE250044	Computer Organization and Architecture	3	0	2	4
PC-BTECVL-403	DOEE250183	Signals and Systems	2	1	2	4
PC-BTECVL-404	DOEE250012	Analog Electronics	2	1	2	4
PC-BTECVL-405	DOEE260002	CMOS VLSI Design	3	0	2	4
AE-BTECVL-406	DASH250026	Design Thinking	1	0	2	2

Semester 5						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTECVL-501	DOEE250062	Digital IC Design	3	0	2	4
PC-BTECVL-502	DOEE260010	FPGA Design and Implementation	2	0	4	4
PC-BTECVL-503	DOEE250015	Analog IC Design	3	1	0	4
PC-BTECVL-504	DOEE260009	AI/ML for Hardware	3	0	2	4
PE-BTECVL-505	Elective	Program Elective				4
AE-BTECVL-506	DASH250104	Technical Presentation and Report Writing	0	0	4	2
VA-BTECVL-507	DASH250030	Economics for Engineers	1	1	0	2

Semester 6						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTECVL-601	DOEE260005	Physical Design of VLSI Circuits	2	0	4	4
PC-BTECVL-602	DOEE260008	Semiconductor Fabrication	3	0	2	4
PC-BTECVL-603	DOEE260011	Embedded Systems Design for Engineers	3	0	2	4
PE-BTECVL-604	Elective	Program Elective				4
AE-BTECVL-605	DASH250049	Entrepreneurial Development and Innovation	1	1	0	2
EE-BTECVL-606	EE	Internship				4

Semester 7						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTECVL-701	DOEE260012	Advanced VLSI Design and Low Power Techniques	3	0	2	4
PC-BTECVL-702	DOEE260007	Testing and Verification of VLSI Circuits	2	0	4	4
PC-BTECVL-703	DOEE260013	EDA Tools and Design Automation	3	0	2	4
PE-BTECVL-704	Elective	Program Elective				4
EE-BTECVL-705	EE	Minor Project				4

Semester 8						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
EE-BTECVL-801	EE	Major Project				12

Elective Courses

Elective Courses for PE Category						
Course Code	**Course Title**	**L**	**T**	**P**	**Cr**	**For Sem./Code**
DOAI260001	AI (Industry)-Applied Ethics for AI	3	0	2	4	
DOAI260002	AI (Industry)-Introduction to Artificial Intelligence	3	0	2	4	
DOAI260003	AI (Industry)-Introduction to Generative AI	3	0	2	4	
DOAI260004	AI (Industry)-Introduction to Machine Learning	3	0	2	4	
DCSA250056	Numerical Methods	3	1	0	4	
DCSE250020	Cloud Computing	3	0	2	4	
DOAI250005	Artificial Intelligence and Machine Learning	2	0	4	4	
DOAI250007	Fundamentals of Artificial Intelligence	2	0	4	4	
DOAI250013	Machine Learning Fundamentals	2	0	4	4	
DOAI250029	Artificial Intelligence with Python	2	0	4	4	
DOAI250031	Introduction to High Performance Computing	3	1	0	4	
DOEE250011	Analog Communication	2	1	2	4	
DOEE250057	Digital Communication Systems	3	0	2	4	
DOEE250124	IOT and its Applications	3	1	0	4	
DOEE250128	Linear Electrical Networks	2	2	0	4	
DOEE250134	Low Power VLSI Design	3	1	0	4	
DOEE250138	Medical Electronics	3	0	2	4	
DOEE250175	Reconfigurable Computing	3	0	2	4	
DOEE250203	VLSI with VHDL	3	0	2	4	
DOEE260001	HDL Programming: Verilog and VHDL	3	0	2	4	
DOEE260014	Hardware Security and Trust	3	0	2	4	
DOEE260015	Neuromorphic Computing	3	0	2	4	
DOEE260016	Digital Filters and DSP Processors	3	0	2	4	
DOEE260017	Power Electronics Integrated Circuits	3	0	2	4	
DOEE260018	Photonics and Optoelectronic Devices	3	1	0	4	
DOEE260019	AI/ML for VLSI Design	3	0	2	4	
DOEE260020	Advanced CMOS Analog Design	3	0	2	4	
DOEE260021	Fault-Tolerant Computing	3	1	0	4	
DOEE260022	System on Chip (SoC) Design	3	0	2	4	
DOEE260023	Memory Design and Technology	3	1	0	4	

Elective Courses for OE Category

Students may opt courses under MOOC, NPTEL, SWAYAM, NSQF/NCVET aligned courses or other relevant technical subjects not included in their core curriculum. The exercise of option shall be subject to the Course Schedule of the respective Constituent Unit, the credit requirements and availability of seats. Guidelines for choosing MOOC/Online courses are given separately and shall have to be adhered to.

Elective Courses for AU (Audit) Category

Course Code	Course Title	L	T	P	Cr
DASH240027	Disaster Management	2	0	0	0
DASH240047	English for Research Paper Writing	2	0	0	0
DASH240052	Environmental Science	3	0	0	0
DASH240085	Pedagogy Studies	2	0	0	0
DASH240086	Personality Development through Life Enlightenment Skills	2	0	0	0
DASH240097	Sanskrit for Technical Knowledge	2	0	0	0
DASH240103	Stress Management by Yoga	2	0	0	0
DASH240104	Technical Presentation and Report Writing	0	0	2	0
DASH240106	Value Education	2	0	0	0
DASH240124	Constitution of India - AU	2	0	0	0
DASH250023	Constitution of India	2	1	0	0
DASH250027	Disaster Management	3	0	0	0
DASH250034	Energy and Environmental Engineering	3	0	0	0
DASH250047	English for Research Paper Writing	2	0	0	0
DASH250054	Essence of Indian Knowledge and Tradition	2	0	0	0
DASH250085	Pedagogy Studies	2	0	0	0
DASH250086	Personality Development	2	0	0	0
DASH250097	Sanskrit for Technical Knowledge	3	0	0	0
DASH250103	Stress Management by Yoga	3	0	0	0
DASH250106	Value Education	2	0	0	0
DASH250117	Innovative Thinking and Entrepreneurship Skills	1	0	0	0
DASH250118	Intellectual Property Rights	2	0	0	0
DCSA240080	Training on Computer Networking	0	0	2	0
DCSA240304	Lab Based on Statistical Package	0	0	2	0
Any other Scheduled Course as per the Course Schedule of the respective Constituent Unit can also be chosen, subject to availability of seats. However, there shall be no assessment and grading for the course chosen as Audit Course.					

The choice of all type of Electives available shall be as per the Course Schedule of the respective Constituent Unit.

Guidelines for Massive Open Online Courses (MOOC) / approved Online Courses

1. Relevant Swayam, MOOC, NPTEL, NIELIT NSQF and any other online courses approved by UGC/AICTE shall also be allowed as Open Electives(OE).

2. One credit of MOOC course should be minimum of 30 hours.

3. A student can choose any approved online course as Open Elective, subject to minimum credit requirements.

4. The students willing to opt OE under MOOC in their next semester, will submit their request to the MOOC Coordinator at their respective campus.

5. Once approved, the campus shall display the list of accepted courses.

6. The student has to submit certificate issued by the institution offering the MOOCs course, along with the number of credits and grades, to get credits transferred into his/her marks certificate issued by NDU.

7. The transfer of credits shall be as adopted by NDU.

Detailed Syllabus

Course Code	Course Name	L	T	P	Cr
DASH250038	Engineering Mathematics I	3	1	0	4

Course Outcomes:

CO1 : Apply matrix operations to solve systems of linear equations, find eigenvalues/eigenvectors, and use the Cayley-Hamilton theorem.
CO2 : Analyze functions of a single variable for limit, continuity and differentiability, Mean value theorems, definite integrals in engineering applications such as areas and volumes.
CO3 : Solve first and second-order ordinary differential equations, including linear, exact, and simultaneous forms, and interpret their geometric and analytical solutions.
CO4 : Differentiate multivariable functions, use Euler's theorem, Jacobians, and Taylor series, and solve problems involving maxima and minima of functions of two variables.
CO5 : Evaluate double and triple integrals in different coordinate systems and apply them to compute areas and volumes, including changing the order of integration when required.

Module 1:

Matrices (9 hrs)
Rank of a matrix, Consistency of a system of linear equations, Characteristic equation of a matrix, Eigen values and Eigen vectors, Cayley-Hamilton theorem (excluding proof), Verification, Application (Finding Inverse of a matrix), Diagonalization of a matrix by orthogonal transformation.

Module 2:

Differential Calculus and Applications (9 hrs)
Limit, Continuity and differentiability of a function of single variable, Rolle's theorem, Lagrange's theorem, and Cauchy's mean value theorem (Statements and simple applications), Applications of definite integrals to evaluate surface areas and volumes of revolutions of curves in Cartesian coordinates.

Module 3:

Differential Equations (7 hrs)
Ordinary Differential Equations (ODE): First order differential equations, Exact differential equations and integrating factor, Linear differential equations of first and second order, Simultaneous linear differential equations by elimination method.

Module 4:

Multivariable Calculus (Partial Differentiation and Applications) (11 hrs)
Partial differentiation, Partial derivatives of first, second and third order, Partial differentiation of implicit functions, Euler's theorem, Total derivative, Jacobian and its Properties, Applications: Maxima and minima of functions of two variables.

Module 5:

Multivariate Calculus (Multiple Integrals) (9 hrs)
Double integral (Cartesian, polar form), Application of double integral: Area by double integration, Change of Order of Integration. Triple Integral in Cartesian coordinate only, Application of triple integral: Volume by triple integration.

Labs / Practicals:

N/A

References:

1. Grewal, B. S. Higher Engineering Mathematics (43rd ed.). Khanna Publishers.
2. Bali, N. P., & Goyal, M. A Text Book of Engineering Mathematics. Laxmi Publications (P) Ltd.
3. Kreyszig, E. Advanced Engineering Mathematics (9th ed.). John Wiley & Sons.
4. Ramana, B. V. Higher Engineering Mathematics. Tata McGraw-Hill.
5. Jeffery, A. Advanced Engineering Mathematics. Academic Press.

Course Code	Course Name	L	T	P	Cr
DOEE260025	Fundamentals of Electronics	3	1	0	4

Course Outcomes:

CO1. Ability to apply various techniques and theorems for analysis of electrical circuits and systems
CO2. Analyze basic DC and AC electrical circuits using Ohm's Law, Kirchhoff's Laws, and network theorems
CO3. Gain understanding about several semiconductor diodes.
CO4. Ability to apply knowledge of diodes for its circuit applications.

Module 1:

Introduction to Circuit Theory:
Basic concepts about voltage, current, power, energy. Introduction to circuit elements like Resistors, Capacitors, Inductors, and Voltage/Current sources, Kirchoff's Voltage Law, Kirchoff's Current Law, magnetically coupled circuits, Loop current and node voltage methods for circuit analysis. Types of Electrical Energy Sources: Independent and Dependent Voltage and Current sources. Analysis of AC and DC Circuits. Transient and Steady State Analysis

Module 2:

Network Theorems:
Superposition Theorem, Thevenin and Norton Theorem, Maximum Power Transfer Theorem, Tellegen's Theorem, Millman's Theorem, Reciprocity Theorem, Compensation Theorem.

Module 3:

AC Circuits and Measurement:
Sinusoidal voltage and current: amplitude, frequency, phase, time period. RMS and average values, form factor, peak factor. Phasor representation of sinusoidal quantities. AC analysis of pure R, L, and C components. RLC series and parallel circuits. Power in AC circuits: real, reactive, and apparent power. Power factor and its significance.

Module 4:

Semiconductor Diodes:
Semiconductors, , Effect of Temperature on Semiconductors, types of semiconductor materials, Review of PN Junction Diode, Diode Equivalent Circuits, V-I Characteristics, Ideal diode, Zener Diodes, Light-Emitting Diodes (LEDs), Varactor Diodes, Power Diodes, Tunnel Diodes, Photodiodes

Module 5:

Applications:
Halfwave, Fullwave and Bridge rectifiers, Passive filters, Regulators, Line regulator and load regulators, Zener Diode as voltage regulators, concept of switched mode power supply (SMPS)

Labs / Practicals:

NA

References:

1. Van-Valkenberg M E "Network Analysis", PHI, New Delhi, Third Edition, (1999).
2. Nahvi M, Edminister J, "Electric Circuits (Schaum's Outline Series)", Tata McGraw Hill, Fourth Edition, (2017).
3. S P Ghosh, A K Chakraborty, "Network Analysis and Synthesis" Tata McGraw Hill, (2010).
4. M.S. Sukhija, T.K. Nagsarkar – Basic Electrical and Electronics Engineering, Oxford University Press.
5. D.P. Kothari, I.J. Nagrath – Basic Electrical and Electronics Engineering, McGraw-Hill Education.

Course Code	Course Name	L	T	P	Cr
DOAI250044	Probability and Statistical Techniques	3	1	0	4

Course Outcomes:

CO1: Summarize and represent raw data using graphical tools, central tendency, dispersion, and shape measures for effective data interpretation.
CO2: Understand probability concepts and apply probability functions to analyse discrete and continuous random variables.
CO3: Identify and apply standard probability distributions and generating functions to solve problems involving random events and statistical modelling.
CO4: Analyse joint distributions, apply transformation techniques, and interpret multivariate dependence using covariance, correlation, and conditional expectations.
CO5: Compute and interpret moments, covariance and correlation matrices, and understand Central Limit Theorem.

Module 1:

Descriptive Statistics & Data Representation
Variables and Raw Data
Graphical Representation: Plots, charts, histograms, frequency polygons
Frequency Distributions: Relative, cumulative, and frequency curves
Measures of Central Tendency: Mean, median, mode
Measures of Dispersion: Range, standard deviation, quartile deviation, mean/median absolute deviation
Measures of shape: Skewness and Kurtosis

Module 2:

Probability Fundamentals and Random Variables
Introduction to Probability: Sample space, events, axioms
Random Variables: Definition, types (discrete/continuous), independence
Probability Functions: PMF (Probability Mass Function), PDF (Probability Density Function), CDF (Cumulative Distribution Function)
Joint Probability Distributions: Marginal and conditional distributions
Expectation and Variance: Linearity of expectation, covariance, correlation

Module 3:

Probability Distributions and Generating Functions
Discrete Distributions: Binomial, Poisson (properties, applications)
Continuous Distributions: Normal, Uniform, Exponential, Gamma, Beta
Relationship between Binomial and Normal, Poisson and Normal distributions
Generating Functions: Probability generating functions (PGF), Moment generating functions (MGF), Characteristic functions (properties and applications)

Module 4:

Multivariate Distributions and Transformations
Bivariate Random Variables: Joint distributions, conditional distributions.
Transformation Techniques: Single variable , Multiple variables (Jacobian method)
Conditional Expectation and Variance
Covariance and Correlation.

Module 5:

Moments: Raw vs. central moments, interpretation of Moments.
Covariance Matrix: Definition, properties, and eigenvalues.
Correlation Matrix, Central Limit Theorem.

Labs / Practicals:

N/A

References:

1. Fundamentals of Mathematical Statistics by S.C. Gupta & V.K. Kapoor
2. Probability, Random Variables, and Stochastic Processes by Athanasios Papoulis and S. Unnikrishna Pillai (4th Ed.)
3. An Introduction to Probability and Statistics by Vijay K. Rohatgi & A.K. Md. Ehsanes Saleh (3rd Ed.)
4. Statistics for Management by Richard I. Levin & David S. Rubin.
5. Applied Multivariate Analysis by S. Dasgupta.

Course Code	Course Name	L	T	P	Cr
DCSE260001	Programming using C and Python	2	0	4	4

Course Outcomes:

CO1: Master the syntax, semantics, and basic programming constructs of the C & Python languages.
CO2: Develop problem-solving skills through algorithmic thinking and programming exercises.
CO3: Understand memory management, pointers, and data structures for efficient programming.
CO4: Gain proficiency in modular programming, debugging, and testing techniques

Module 1:

Flow charts, Data types and Storage classes, Scope of Variables, Arithmetic operators, Assignment operators, Conditional operators, Arithmetic expressions, Enumerated data types, Decision making, Branching, Looping, Switch concept, Function and parameter passing, Recursive functions, Macros.

Module 2:

Programs to illustrate basic language constructs in C like Factorial, Sine/cosine and other mathematical series, Fibonacci series, calculating square-root of a number, calculating GCD of 2 integers (Euclid's method and otherwise), Calculating LCM of 2 integers and similar such programs.

Module 3:

Introduction to Python Installing Python, How a Program Works using Python, Program Development Cycle, Input, Processing and Output, Displaying Output with the Print Function, Comments, Variables, Reading Input from the Keyboard, Performing Calculations, Operators, Type Conversions, Expressions, More about Data Output, Decision Structures and Boolean Logic: if, if-else, if-elif-else Statements, Nested Decision Structures, Comparing Strings, Logical Operators, Boolean Variables, Repetition Structures: Introduction, while loop, for loop, Calculating a Running Total, Input Validation Loops, Nested Loops, Data Types and Expressions: Strings, Assignment and Comments, Numeric Data Types and Character Sets, Expressions, Functions and Modules.

Module 4:

Control statements Definite Iteration, Formatting Text for Output, Selection, Conditional Iteration. File and Exceptions: Introduction to File Input and Output, Using Loops to Process Files, Processing Records, Exceptions. Functions: Introduction, Defining and Calling a Void Function, designing a Program to Use Functions, Local Variables, Passing Arguments to Functions, Global Variables and Global Constants, Value-Returning Functions-Generating Random Numbers, the math Module, Storing Functions in Modules.

Module 5:

Introduction to one dimensional and 2-D array with examples. Representing a polynomial using 1- D array and polynomial operations, use of 2-D array to represent a matrix and matrix operations. Character arrays (strings): String related functions (strlen, strcpy, strcat, strcmp, atoi, itoa, reverse, strstretc) and their function definitions, Basic concept, array of structures and its applications. Introduction (declaration and initialization), Pointers and Arrays, Concept of Dynamic Memory Allocation, use of pointers to represent variable- sized 1-D and 2-D arrays, pointers to structures, Introduction to file operations, Opening and Closing files, Reading from and Writing to files, File modes, formatted and unformatted file I/O. Bitwise AND, OR, XOR, NOT, shifting operations, applications of bitwise operators in programming. Use of #include, #define, macros with arguments, conditional compilation, and use of predefined macros. malloc, calloc, realloc, and free.

Labs / Practicals:

1. Write a C program to check whether a number is even or odd using ternary operator.
2. Write a C program to find the sum of individual digits of a positive integer.
3. Write a C program to print the numbers in triangular form.
4. Write a C program to find the second largest integer in a list of integers.
5. Write C programs that use both recursive and non-recursive functions.

6. Write a C program to perform arithmetic using Switch Statement.
7. Write a C program to perform Factorial.
8. Write a C program to print Fibonacci no
9. Installation of Python, and learning interactively at command prompt and writing simple programs.
10. Random number generations, and problems based on random numbers.
11. Handling tuples and exercises based on tuples.
12. Functions and files.
13. Matrix addition, multiplications, and unity matrix.

References:

1. Mark Lutz, “Learning Python”, O'Reilly Media, 5th Edition, 2016.
2. White, “Hadoop: The Definitive Guide”, Third Edition - O’Reilly, 2012.
3. Brandon Rhodes and John Goerzen, “Foundations of Python Network Programming: The Comprehensive Guide to Building Network Applications with Python”, Apress, Second Edition, 2016.
4. Kerninghan; Ritchie, “C programming Language”, PHI.
5. Theraja B.L., Electrical Technology, S. Chand Publishers.
6. Balguruswamy, “Programming in ANSI C”, Tata Mcgraw Hill Publishing.
7. Kakde and Deshpande, “C and data Structure”, Charles River Media Publisher.
8. Dromey R G, “How to Solve it by Computer”, PHI.

Course Code	Course Name	L	T	P	Cr
DASH250016	Communication Skills	1	0	2	2

Course Outcomes:

CO1 : Understand the need and importance of English language.
CO2 : Develop proficiency in the language.
CO3 : Able to use technology to enrich communication skills.
CO4 : Gain self-confidence with improved command over English.
CO5 : Apply English communication skills effectively in academic, professional, and social environments.

Module 1:

Communication Fundamentals
Communication ;Meaning, Process and importance of Communication, Types and Channels of communications , Scope and Significance of Listening ,Speaking, Reading, Writing Skills.

Module 2:

Writing Skills
Basics of Grammar: Parts of Speech, Uses of Tenses, Active Passive, Narration.

Module 3:

Vocabulary Building
Word Formation, Synonyms , Antonyms, Words often Confused, One Word Substitutes, Idioms and Phrasal verbs, Abbreviations of Scientific and Technical Words.

Module 4:

Speaking Skills
Introduction to Phonetic Sounds & Articulation, Word Accent, Rhythm and Intonation, Interpersonal Communication, Oral Presentation, Body Language and Voice Modulation (Para linguistics and Non Verbal), Negotiation and Persuasion, Group Discussion, Interview Techniques (Telephonic and Video conferencing).

Module 5:

Technical Writing
Job application, CV writing, Business Letters, Notices, Report Writing, Minutes, E-mail Etiquette, Blog Writing.

Labs / Practicals:

1. Introducing Oneself, Exercise on Parts of Speech & Exercise on Tense.
2. Exercise on Agreement, Narration, Active Passive Voice & Dialogue Conversation.
3. Exercise on Writing Skills and Listening Comprehension.
4. Practice of Phonemes, Word Accent, Intonation, JAM Session.
5. Individual Presentation, Extempore and Picture Interpretation.
6. Vocabulary Building Exercises (One Word Substitute, Synonyms, Antonyms, Words Often Confused etc.) & Group Discussion.

References:

1. "The Essence of Effective Communication", Ludlow R. and Panton F., Pubs: Prentice Hall.
2. “Effective Communication Skills”, Kulbhushan Kumar, Khanna Publishing House.
3. “A University Grammar of English”, Quirk R. and Sidney G., Pubs: Pearson Education.
4. “High School English Grammar”, Wren and Martin, Pubs: S. Chand & Company Ltd.
5. “Essentials of Business Communication”, Guffrey M.E., Pubs: South-Western College Publishing.
6. “Technical Communication: Principles and Practice”, Raman M. and Sharma S., Pubs: Oxford University Press.
7. “Effective Business Communication”, Rodrigues M.V., Pubs: Concept Publishing Company, Delhi.

8. “English Vocabulary in Use”, McCarthy M. and Felicity O’ Dell.

Course Code	Course Name	L	T	P	Cr
DASH250039	Engineering Mathematics II	3	1	0	4

Course Outcomes:

CO1 : Analyze the behavior of sequences and series using convergence tests and apply them in evaluating infinite processes.
CO2 : Construct Fourier series representations of periodic functions and apply them in engineering and data modeling contexts.
CO3 : Compute Laplace transforms of various functions including step and impulse functions, and use them for solving engineering problems.
CO4 : Perform inverse Laplace transforms and use them effectively to solve initial value problems in engineering models.
CO5 : Use Fourier transforms and their properties to switch between time and frequency domains, aiding AI applications such as image/signal processing and neural network optimization.

Module 1:

Sequences and Series (9 hrs)
Limit of a sequence, monotone and Cauchy sequences, properties of convergent sequences, examples. Infinite series, positive series, tests for convergence and divergence, integral test, alternating series, Leibnitz test.

Module 2:

Fourier Series (9 hrs)
Fourier Series of Periodic functions (Euler's Formulas and Dirichlet's conditions), Even and odd function expansions (Half range sine cosine series), Fourier series over arbitrary intervals (Change of interval techniques).

Module 3:

Laplace Transformation (7 hrs)
Laplace Transforms: Laplace Transform of standard functions, First shifting theorem, Second shifting theorem, Unit step function, Dirac delta function, Laplace transforms of functions when they are multiplied and divided by 't', Laplace transforms of derivatives (First and second derivative) of functions, Laplace Transformation of integral of functions, Evaluation of integrals by Laplace transforms.

Module 4:

Inverse Laplace Transformation and Applications (11 hrs)
Inverse Laplace transform by partial fractions and shifting theorem, convolution theorem (without proof). Applications: solving Initial value problems of first order by Laplace Transform method.

Module 5:

Fourier Transformation (9 hrs)
Introduction to Fourier Transform, Properties- Linearity, Time and frequency shifting property (1st and 2nd shifting theorem), Differentiation and integration in time domain and their transforms, Fourier Integral theorem (statement only), Fourier transform and its inverse, Convolution theorem (Statement and applications).

Labs / Practicals:

N/A

References:

1. Bali, N. P., & Goyal, M. A Text Book of Engineering Mathematics. Laxmi Publications (P) Ltd.
2. Kreyszig, E. Advanced Engineering Mathematics (9th ed.). John Wiley & Sons, New Delhi.
3. Ramana, B. V. Higher Engineering Mathematics. Tata McGraw-Hill, New Delhi.
4. Grewal, B. S. Higher Engineering Mathematics. Khanna Publishers, New Delhi.
5. H.K. Dass, Advanced Engineering Mathematics. S. Chand & Company Pvt. Ltd., New Delhi.
6. Jain, R. K., & Iyengar, S. R. K. Advanced Engineering Mathematics. Narosa Publishing House, New Delhi.
7. Dass, H. K., & Verma, R. Engineering Mathematics. S. Chand & Company Pvt. Ltd., New Delhi.
8. Pundir, R. S., & Pundir, S. Engineering Mathematics. Laxmi Publications (P) Ltd.
9. Ghosh, S. C. Engineering Mathematics. PHI Learning Pvt. Ltd.
10. Gupta, C. S., & Malik, A. Mathematical Analysis and Applications. Wiley Eastern / Narosa Publishing House

Course Code	Course Name	L	T	P	Cr
DOEE250059	Digital Electronics	3	0	2	4

Course Outcomes:

CO1: Perform conversions between number systems and design logic functions using basic and universal gates.
CO2: Simplify Boolean expressions using Karnaugh Maps and Boolean algebra for circuit minimization.
CO3: Design and analyze combinational logic circuits like adders, subtractors, multiplexers, and decoders.
CO4: Understand and implement sequential circuits such as flip-flops, counters, and shift registers.
CO5: Understand the architecture and applications of memory units and programmable logic devices.

Module 1:

Number System and Logic Gates: Representation and interconversion among binary, octal, decimal, and hexadecimal number systems, Binary arithmetic operations: addition, subtraction, multiplication, and division, Binary Coded Decimal (BCD), Excess-3, and Gray codes, Signed and unsigned binary number representation,1's and 2's complement techniques for binary subtraction, Code conversions: Binary to BCD, Excess-3, and Gray codes, and vice versa, Logic gates: AND, OR, NOT, NAND, NOR, XOR, XNOR, Universal gates and implementation of basic logic functions using universal gates.

Module 2:

Boolean Algebra and Logic Simplification: Boolean laws and theorems, DeMorgan's theorems, SOP and POS forms, Canonical forms, Karnaugh Map (2, 3, 4 variables) – minterms and maxterms, Quine-McCluskey method (basic idea), Logic expression simplification and gate minimization.

Module 3:

Combinational Logic Circuits: Adders: Half adder, full adder, ripple carry adder, carry look-ahead adder, Subtractors: Half and full subtractors, Comparators, Multiplexers (MUX) and De-multiplexers (DEMUX) Encoders and Decoders, BCD to 7-segment decoder, Design of combinational logic using MUX, DEMUX.

Module 4:

Sequential Logic Circuits: Latches and Flip-flops: SR, D, T, JK, Master-Slave, Characteristic tables and excitation tables, Edge and level triggering, Timing diagrams, Counters: Asynchronous (ripple) and Synchronous counters, Up/down counters, modulo-N counters, Shift registers: SISO, SIPO, PISO, PIPO, Ring counter and Johnson counter, Mealy and Moore Finite State Machines (FSMs).

Module 5:

Memory & Logic Families: Classification of memories: RAM, ROM, PROM, EPROM, EEPROM, Memory decoding and address space, Overview of programmable logic devices: PLA, PAL, CPLD, FPGA (introductory), Logic families: TTL, CMOS. Characteristics: propagation delay, power dissipation, noise margin, fan-out, fan-in.

Labs / Practicals:

1. Verification of Logic Gates: Implement and verify the truth tables of AND, OR, NOT, NAND, NOR, XOR, and XNOR gates.
2. Universal Gates Implementation: Realize basic logic functions using only NAND or NOR gates.
3. Boolean Expression Simplification: Simplify given Boolean expressions and implement the minimized circuits using logic gates.
4. Design of Half Adder and Full Adder: Construct and test circuits for half and full adders using logic gates.
5. Design of Half Subtractor and Full Subtractor: Implement and verify half and full subtractor circuits.
6. Multiplexer and Demultiplexer: Design and test 4:1 MUX and 1:4 DEMUX circuits.
7. Flip-Flop Implementations: Implement and observe SR, D, T, and JK flip-flops using basic gates or ICs.
8. Counters: Design and test asynchronous and synchronous up/down counters (e.g., mod-8, mod-10).
9. Shift Registers: Construct and analyze SIPO, SISO, PIPO, and PISO shift registers.

10. BCD to 7-Segment Display: Design a decoder to display BCD inputs on a 7-segment LED display.

References:

1. M. Morris Mano, Michael D. Ciletti – Digital Design, Pearson Education
2. R.P. Jain – Modern Digital Electronics, McGraw-Hill Education
3. Thomas L. Floyd – Digital Fundamentals, Pearson Education
4. John F. Wakerly – Digital Design: Principles and Practices, Pearson
5. Tokheim – Digital Electronics: Principles and Applications, McGraw-Hill Education

Course Code	Course Name	L	T	P	Cr
DCSE250055	Data Structures	3	0	2	4

Course Outcomes:

CO1: Remember the different types of linear and non-linear data structures.

CO2: Understand how structuring of data can reduce the average access time.

CO3: Apply concrete data structures to implement abstract data structures.

CO4: Analyze the fundamental operations supported by various data structures.

CO5: Evaluate the number of steps required to do an operation on a data structure.

CO6: Create efficient implementations that are suitable for real-world applications.

Module 1:

Multi-dimensional arrays, addressing in row-major and column-major orderings.
Searching: linear search, binary search, uniform binary search, interpolation search.
Sorting by comparison: selection, bubble, insertion, Shell sort, quicksort, mergesort.
Non-comparison sorting: counting sort, pigeonhole sort, radix sort (LSD and MSD).

Module 2:

Linked lists types: linear and circular, unidirectional and bidirectional.
Operations on linked lists: create, insert, delete, traverse, reverse, find.
Open address hashing, collision, linear probing, quadratic probing, double hashing.
Hashing with separate chaining, load factor, applications of hash tables.

Module 3:

Abstract data structures, concrete implementation using arrays and linked lists.
Stacks: LIFO access, operations (push, pop, top), call stack, expression evaluation.
Queues: FIFO access, operations (enqueue, dequeue, peek), linear queue, ring buffer.
Double-ended queue (input-restricted, output-restricted), priority queue, DEPQ.

Module 4:

Graphs (undirected and directed), adjacency matrix, adjacency list, weighted graphs.
Graph traversals: breadth-first search, depth-first search, iterative deepening search.
Graph connectivity, directed acyclic graphs, topological sorting, spanning trees.
Arborescence, branching factor, binary trees (skew, strict, complete, perfect).

Module 5:

Tree traversal: pre-order (NLR), in-order (LNR), post-order (LRN), reverse variants.
Binary heap (min-heap and max-heap), Fibonacci heap, binomial heap, heapsort.
Binary search tree (BST), sorting with BST, threaded BST (single and double).
Disjoint-set data structure, operations: makeset, find, union (by rank or by size).

Labs / Practicals:

Each data structure and supporting algorithms should be implemented in C or Python.

01. Use arrays to store univariate polynomials for doing addition and multiplication.

02. Write a function to find min/max element and use it to implement selection sort.

03. Implement an adaptive variant of bubble sort that runs in linear time for best case.

04. Implement insertion sort with a type-agnostic design (like qsort from stdlib.h).

05. Perform non-comparative sorting with counting sort and radix sort (LSD and MSD).

06. Create unidirectional and bidirectional linked lists with linear and circular variants.

07. Create open address hash tables with linear probe, quadratic probe, double hashing.

08. Create hash table with an array of pointers for separate chaining using linked lists.

09. Implement stack, queue (linear and circular), deque using arrays and linked lists.

10. Convert among prefix, infix, postfix expressions, and evaluate the postfix notation.

11. Perform breadth-first traversal, depth-first traversal, topological sort on a digraph.

12. Implement priority queue using binary max heap, supporting insertion and deletion.

13. Heapify an array of strings in linear time and do heap sort in lexicographic order.

14. Perform insert and delete on a binary search tree threaded for in-order traversal.

15. Implement disjoint-set data structure supporting makeset, find, union operations.

References:

1. "Think Data Structures" by Allen Benjamin Downey
https://greenteapress.com/thinkdast/thinkdast.pdf

2. "Open Data Structures: An Introduction" by Pat Morin
https://www.aupress.ca/app/uploads/120226_99Z_Morin_2013-Open_Data_Structures.pdf

3. "An Open Guide to Data Structures and Algorithms" by Paul W. Bible and Lucas Moser
https://pressbooks.palni.org/anopenguidetodatastructuresandalgorithms

4. "Computer Science II" by Chris Bourke
https://cse.unl.edu/~cbourke/ComputerScienceTwo.pdf

5. "Data Structures and Algorithms" by Alfred Vaino Aho, John Edward Hopcroft, Jeffrey David Ullman
https://dl.acm.org/doi/10.5555/577958

6. "Data Structures and Algorithm Analysis in C (Second Edition)" by Mark Allen Weiss
https://dl.acm.org/doi/10.5555/248735

Course Code	Course Name	L	T	P	Cr
DASH250101	Sports and Yoga	0	0	4	2

Course Outcomes:

CO1: Understand the principles of physical education, fitness, wellness, and positive lifestyle habits essential for maintaining overall health.

CO2: Demonstrate knowledge of basic anatomy, physiology, and posture correction, and apply this understanding to enhance body mechanics and prevent injuries.

CO3: Practice fundamental yoga postures, breathing techniques, and relaxation methods to improve flexibility, concentration, and mental well-being.

CO4: Apply yoga and fitness practices to manage lifestyle-related health conditions and promote physical and psychological balance.

CO5: Participate actively in sports and fitness activities, demonstrating teamwork, discipline, and awareness of safety, nutrition, and lifelong fitness practices.

Module 1:

Foundations of Physical Education and Fitness
Meaning & definition of Physical Education, Aims & Objectives of Physical Education, Changing trends in Physical Education, Meaning & importance of Physical Fitness and Wellness, Components of physical fitness and health-related fitness, Components of wellness, Concept of positive lifestyle, Preventing health threats through lifestyle changes.

Module 2:

Anatomy, Physiology and Posture
Introduction to Anatomy and Physiology, Importance of Anatomy and Physiology in Physical Education, Effects of exercise on circulatory, respiratory, and neuromuscular systems, Meaning & concept of posture, Causes and correction of bad postures, Postural deformities: Knock knee, Flat foot, Lordosis, Kyphosis, etc., Weight training: pros and cons.

Module 3:

Yoga-Theory and Practice
Meaning & importance of Yoga, Elements of Yoga: Asanas, Pranayama, Meditation, Kriyas, Yoga for concentration and related Asanas (Sukhasana, Tadasana, Padmasana, Shashankasana), Relaxation techniques (Yog-Nidra).

Module 4:

Yoga for Lifestyle and Health Conditions
Preventive role of Asanas, Yoga for hypertension, obesity, back pain, diabetes, asthma, Benefits, procedures & contraindications for relevant Asanas (e.g., Bhujangasana, Pavan Muktasana, Tadasana, Chakrasana, etc.).

Module 5:

Games and Sports Fundamentals
Select any one sport: Athletics, Basketball, Kabaddi, etc., History and development of the sport, General rules and regulations, Specifications of playfields and equipment, Major tournaments, venues, and personalities, Importance of proper sports gear.

Labs / Practicals:

1. Practice of Sukhasana, Padmasana, and Tadasana for improving concentration and posture.

2. Shashankasana and Bhujangasana for relaxation and spinal flexibility.
3. Introduction to and practice of basic Pranayama techniques (Anulom-Vilom, Bhramari).
4. Practice of Yog Nidra (deep relaxation technique).
5. Asanas for lifestyle diseases: practice of Pavan Muktasana, Chakrasana, and Tadasana for obesity and diabetes control.
6. Demonstration and correction of bad postures (knock knee, flat foot, lordosis, kyphosis).
7. Light warm-up and stretching routines before yoga or sports activities.
8. Basic fitness tests: measurement of flexibility, endurance, strength, and body composition.
9. Participation in any one selected sport: demonstration of rules, gear, warm-up drills.
10. Practice of the chosen sport (e.g., Athletics, Basketball, Kabaddi) including basic skills and gameplay
11. Group activity: Demonstration of importance of proper sports gear and warm-up before sports.
12. Poster or presentation on a Yoga-based lifestyle for common health issues.
13. Introduction to circuit training and its benefits for physical fitness.
14. Meditation sessions and reflective journaling on mental well-being.
15. Nutritional awareness activity: discussion on diet for fitness and yoga lifestyle.

References:

1. Modern Trends and Physical Education by Prof. Ajmer Singh.
2. Light On Yoga by B.K.S. Iyengar.
3. Health and Physical Education – NCERT (11TH and 12th Classes).
4. Back to health through YOGA by Ramesh Bijlani.

Course Code	Course Name	L	T	P	Cr
DOEE260024	Introduction to VLSI Technology	3	1	0	4

Course Outcomes:

This course aims at understanding the manufacturing methods and their underlying scientific principles in the context of technologies used in VLSI chip fabrication.

1. Explain the basics of VLSI technology, scaling trends, and the evolution of integrated circuits.
2. Describe key fabrication steps such as oxidation, diffusion, ion implantation, photolithography, and etching.
3. Understand CMOS fabrication, design rules, and basic layout techniques for digital circuits.

Module 1:

Crystal Growth, Wafer manufacturing and Clean rooms: CMOS Process flow starting from Substrate selection to multilevel metal formation, comparison between bulk and SOI CMOS technologies. Crystal structure, Czochralski and FZ growth methods, Wafer preparation and specifications, SOI Wafer manufacturing, Clean rooms, wafer cleaning and gettering: Basic concepts, manufacturing methods and equipment, Measurement methods.

Module 2:

Photolithography and Oxidation:
Photolithography: Light sources, Wafer exposure systems, Photoresists, Baking and development, Mask making, Measurement of mask features and defects, resist patterns and etched features.
Oxidation: Wet and Dry oxidation, growth kinetics and models, defects, measurement methods and characterization.

Module 3:

Diffusion and Ion-implantation:
Diffusion: Models for diffused layers, Characterization methods, Segregation, Interfacial dopant pileup, oxidation enhanced diffusion, dopant-defect interaction.
Ion-implantation: Basic concepts, High energy and ultralow energy implantation, shallow junction formation & modeling, Electronic stopping, Damage production and annealing, RTA Process & dopant activation

Module 4:

Thin film Deposition: Chemical and physical vapour deposition, epitaxial growth, manufacturing methods and systems, deposition of dielectrics and metals commonly used in VLSI, Modeling deposition processes.

Module 5:

Etching Technologies and Back-end Technology:
Etching Technologies: Wet etching, Plasma etching, RIE, Etching of materials used in VLSI, Modeling of etching,
Back-end Technology: Contacts, Vias, Multi-level Interconnects, Silicided gates and S/D regions, Reflow & planarization, Multi-chip modules and packaging.

References:

1. James Plummer, M. Deal and P.Griffin, Silicon VLSI Technology, Prentice Hall Electronics, 2000.
2. Stephen Campbell, The Science and Engineering of Microelectronics, Oxford University Press, 1996.
3. S. M. Sze (Ed), VLSI Technology, McGraw Hill, Second Edition, 1988.
4. S.K. Gandhi, VLSI Fabrication Principles, John Wiley Inc., New York, 1983

Course Code	Course Name	L	T	P	Cr
DOEE260026	Electronics Devices and Circuits	3	1	0	4

Course Outcomes:

CO1. Understand the principles and characteristics of different types of semiconductor devices.
CO2. Understand the fabrication process of semiconductor devices.
CO3. Utilize the mathematical models of semiconductor junctions and MOS transistors for circuits and systems
CO4. Understand the constructional details and characteristics of Diode, BJT, and MOS Devices.
CO5. Understand the characteristics of special diodes.

Module 1:

Semiconductor Basics:
Bonding forces, Energy bands in Solids, Metals, Semiconductors and Insulators, Direct and Indirect semiconductors, Electrons and Holes, Intrinsic and Extrinsic materials, Conductivity and Mobility, Drift and Resistance, Effects of temperature and doping on mobility, Hall Effect..

Module 2:

P-N Junctions:
Forward and Reverse biased junctions- Qualitative description of Current flow at a junction, reverse bias, Reverse bias breakdown- Zener breakdown, avalanche breakdown, Rectifiers, Optoelectronic Devices Photodiodes: Current and Voltage in an Illuminated Junction, Solar Cells, Photo detectors, Light Emitting Diode.

Module 3:

Bipolar Junction Transistor:
Fundamentals of BJT operation, Amplification with BJTS, BJT Fabrication, The coupled Diode model (Ebers-Moll Model), Switching operation of a transistor, Cutoff, saturation, switching cycle, specifications, Drift in the base region, Base narrowing, Avalanche breakdown.

Module 4:

Field Effect Transistors:
Basic JFET Operation, Equivalent Circuit and Frequency Limitations, MOSFET Two terminal MOS structure- Energy band diagram, Ideal Capacitance – Voltage Characteristics and Frequency Effects, Basic MOSFET Operation- MOSFET structure, Current-Voltage Characteristics.

Module 5:

Fabrication of P-N Junctions:
Thermal Oxidation, Diffusion, Rapid Thermal Processing, Ion implantation, chemical vapour deposition, photolithography, Etching, metallization.

Labs / Practicals:

NA

References:

1. Ben. G. Streetman, Sanjay Kumar Banergee, “Solid State Electronic Devices”, 7th Edition, Pearson Education.
2. Donald A Neamen, Dhrubes Biswas, “Semiconductor Physics and Devices”, 4th Edition, MCGraw Hill Education, 2012.

Course Code	Course Name	L	T	P	Cr
DOEE260001	HDL Programming: Verilog and VHDL	3	0	2	4

Course Outcomes:

CO1: Understand the hardware description language paradigm and its role in digital design.
CO2: Write Verilog and VHDL programs for combinational and sequential circuits.
CO3: Model design at structural, dataflow, and behavioral abstraction levels.
CO4: Develop testbenches and perform simulation-based verification of HDL designs.
CO5: Implement HDL designs on FPGA using synthesis tools.

Module 1:

Introduction to HDL and Verilog Basics

Hardware description languages: purpose, role in digital design flow. Verilog basics: modules, ports, wire, reg, integer, parameter. Lexical conventions, operators. Structural modeling: primitive gates, user-defined modules. Dataflow modeling: assign statements, continuous assignment.

Module 2:

Behavioral Verilog

Procedural blocks: always, initial. Sequential statements: if-else, case, for, while, repeat, forever. Blocking and non-blocking assignments. Tasks and functions. System tasks: $display, $monitor, $dumpvars. Verilog testbench: stimulus generation, response checking.

Module 3:

VHDL Fundamentals

VHDL entities, architectures, signal declaration, variable, constant. VHDL data types: std_logic, std_logic_vector, integer, boolean. Concurrent signal assignment, conditional signal assignment, selected signal assignment. Process statement, sequential statements in VHDL. Component instantiation and port mapping.

Module 4:

FSM and Advanced HDL Modeling
Finite state machine design in Verilog and VHDL: Mealy and Moore machines. One-hot encoding, binary encoding. Parameterized modules in Verilog, generics in VHDL. Memories and register files in HDL. Pipeline design using HDL. HDL coding guidelines for synthesis.

Module 5:

Synthesis and FPGA Implementation

Logic synthesis flow: HDL → netlist. Synthesis constraints: timing, area, power. Timing analysis basics: setup and hold constraints. FPGA architecture overview: LUTs, flip-flops, DSP blocks, block RAM. FPGA design flow using Xilinx Vivado or Intel Quartus. Place and route, bitstream generation.

Labs / Practicals:

1. Verilog structural model of 2-input gates and combinational circuits.
2. Dataflow model for half adder, full adder.
3. Behavioral Verilog model for 4-bit counter.
4. Verilog model of 4×1 multiplexer and 2-to-4 decoder.
5. Verilog FSM design: traffic light controller.
6. VHDL entity and architecture for basic combinational circuits.
7. VHDL process-based sequential circuit (D flip-flop, register).

8. Verilog testbench for functional verification of a 4-bit ALU.
9. FPGA implementation: LED blinking and 7-segment display.
10. FPGA implementation of a parameterized N-bit counter with synthesis report analysis.

References:

1. Palnitkar S. – Verilog HDL: A Guide to Digital Design and Synthesis, Prentice Hall PTR.
2. Chu P.P. – RTL Hardware Design Using VHDL, John Wiley & Sons.
3. Brown S. and Vranesic Z. – Fundamentals of Digital Logic with Verilog Design, McGraw-Hill.
4. Ciletti M.D. – Advanced Digital Design with the Verilog HDL, Pearson Education.

Course Code	Course Name	L	T	P	Cr
DASH250105	Universal Human Values-II: Understanding Harmony And Ethical Human Conduct	3	0	0	3

Course Outcomes:

CO1: Understand human aspirations and distinguish between needs of the self and the body.
CO2: Analyze the concept of harmony in self, family, society, and nature.
CO3: Apply ethical principles to real-life situations and emerging challenges in science and technology.
CO4: Evaluate responsible behavior in professional and personal domains with sustainability and justice in mind.
CO5: Develop a holistic perspective for ethical human conduct in the context of AI, data use, and digital society.

Module 1:

Introduction to Value Education and Human Aspirations
Purpose of education, the role of values in personal and professional life; Understanding happiness and prosperity; Distinction between needs of the Self ('I') and Body; right understanding and right feelings; AI ethics and human well-being.

Module 2:

Harmony in the Human Being – Understanding the Self
Self-exploration as the process for value education; Understanding the activities of the Self and harmony in them; Program to ensure harmony; Applying this to deal with digital distractions, identity crisis, and mental well-being.

Module 3:

Harmony in Family and Society
Values in human relationships – trust and respect; Nine universal feelings in relationships; harmony in society through justice, mutual fulfillment, and participation; Ethical dilemmas in teamwork, AI product development, and organizational conduct.

Module 4:

Harmony with Nature and Sustainable Living
Interconnectedness of human and natural systems; Four orders of nature; Co-existence and Eco-balance; Sustainable production, consumption, and innovation in AI/ML; Responsible use of computing resources and data centers.

Module 5:

Ethical Human Conduct and Universal Order
Universal human order (Sarvabhauma Vyavastha); Ethical competence; vision for a humane society; Professional ethics for technologists – data privacy, bias in AI, ethical design of systems; role of engineers in promoting equity, well-being, and digital justice.

Labs / Practicals:

N/A

References:

Textbooks & Guides:
1. R.R. Gaur, R. Sangal, G.P. Bagaria – A Foundation Course in Human Values and Professional Ethics, Excel Books (AICTE
Recommended)
2. Ritu Sinha – Understanding Human Values, Laxmi Publications
3. B. L. Bajpai – Indian Ethos and Modern Management, New Royal Book Co.
4. R. Subramanian – Professional Ethics, Oxford University Press India
5. K. Mohan & M. Banerjee – Developing Soft Skills and Personality, McGraw Hill India

Course Code	Course Name	L	T	P	Cr
DOEE250050	Control System Engineering	3	1	0	4

Course Outcomes:

CO1. Understand the principles and concepts of control systems including feedback, stability, and performance criteria.
CO2. Analyze and design control systems using various techniques such as root locus, frequency response, and state-space methods.
CO3. Gain proficiency in modeling dynamic systems and obtaining transfer functions for control system design.
CO4. Explore the application of control systems in engineering disciplines such as robotics, aerospace, and industrial automation.
CO5. Develop practical skills in implementing control algorithms, tuning controllers, and analyzing system responses through simulations and experiments.

Module 1:

Introduction to Control Systems: Types of Control Systems, Effect of Feedback Systems, Differential equation of Physical Systems – Mechanical Systems, Electrical Systems, Analogous Systems. Block diagrams and signal flow graphs: Transfer functions, Block diagram algebra and Signal Flow graphs.

Module 2:

Time Response of feedback control systems: Standard test signals, Unit step response of First and Second order Systems. Time response specifications, Time response specifications of second order systems, steady state errors and error constants. Introduction to PI, PD and PID Controllers (excluding design).

Module 3:

Stability analysis: Concepts of stability, Necessary conditions for Stability, Routh stability criterion, Relative stability analysis: more on the Routh stability criterion, Introduction to Root-Locus Techniques, The root locus concepts, Construction of root loci.

Module 4:

Frequency domain analysis and stability: Correlation between time and frequency response, Bode Plots, Experimental determination of transfer function. Introduction to Polar Plots, (Inverse Polar Plots excluded) Mathematical preliminaries, Nyquist Stability criterion, (Systems with transportation lag excluded) Introduction to lead, lag and lead-lag compensating networks (excluding design).

Module 5:

Introduction to Digital Control System: Introduction, Spectrum Analysis of Sampling process, Signal reconstruction, Difference equations. Introduction to State variable analysis: Introduction, Concept of State, State variables & State model, State model for Linear Continuous & Discrete time systems, Diagonalization.

Labs / Practicals:

NA

References:

Control systems, K.R. Varmah, McGraw hill
Control System Engineering, D. Roy Chowdhuri, PHI
Digital Control system, B.C. Kuo, Oxford University Press.
Control System Engineering, I. J. Nagrath& M. Gopal. New AgeInternational Publication
Modern Control Engineering, K. Ogata, 4th Edition, Pearson Education

Course Code	Course Name	L	T	P	Cr
DOEE250044	Computer Organization and Architecture	3	0	2	4

Course Outcomes:

CO1: Explain processor fundamentals, instruction representation, and hardware operations including arithmetic and logical functions.
CO2: Analyze instruction set architectures, addressing modes, and computer arithmetic with parallelism and SIMD extensions.
CO3: Design basic datapath and pipelined architectures, and evaluate instruction-level parallelism and hazards in modern processors.
CO4: Demonstrate memory mapping techniques, cache design, memory hierarchy, and virtual memory mechanisms.
CO5: Evaluate memory management strategies including RAID levels, disk storage, multithreading, and multiprocessor systems.

Module 1:

Introduction of Processor:Introduction, Technologies for building Processors and Memory, Performance, The Power Wall, Operations of the Computer Hardware, Operands Signed and Unsigned numbers, Representing Instructions, Logical Operations, Instructions for Making Decisions

Module 2:

Instructions Set:MIPS Addressing for 32-Bit Immediate and Addresses, Parallelism and Instructions: Synchronization, Translating and Starting a Program, Addition and Subtraction, Multiplication, Division, Floating Point, Parallelism and Computer Arithmetic: Sub word Parallelism, Streaming SIMD Extensions and Advanced Vector Extensions in x86.

Module 3:

Architecture Building Block: Logic Design Conventions, building a Datapath, A Simple Implementation Scheme, overview of Pipelining, Pipelined Datapath, Data Hazards: Forwarding versus Stalling, Control Hazards, Exceptions, Parallelism via Instructions, The ARM Cortex – A8 and Intel Core i7 Pipelines, Instruction –Level Parallelism and Matrix Multiply Hardware Design language.

Module 4:

Memory Mapping: Memory Technologies, Basics of Caches, Measuring and Improving Cache Performance, dependable memory hierarchy, Virtual Machines, Virtual Memory, Using FSM to Control a Simple Cache, Parallelism and Memory Hierarchy: Redundant Arrays of Inexpensive Disks, Advanced Material: Implementing Cache Controllers.

Module 5:

Memory Management: Disk Storage and Dependability, RAID levels, performance of storage systems, Introduction to multi- threading clusters, message passing multiprocessors.

Labs / Practicals:

Module 1 & 2: Processor Basics and Instruction Set
1. Simulation of Basic Arithmetic Operations in MIPS Assembly
o Objective: Write MIPS assembly code to perform addition, subtraction, multiplication, and division of two integers.
o Tool: MIPS Simulator (e.g., QtSpim, MARS)
2. Binary and Hexadecimal Conversion & Representation
o Objective: Implement a program (in C or Assembly) to convert between binary, hexadecimal, and decimal formats; demonstrate signed and unsigned number representations.

3. Instruction Encoding and Decoding
o Objective: Manually encode and decode sample MIPS instructions and verify using a simulator.

Module 3: Datapath and Pipelining
4. Design a Basic Arithmetic Logic Unit (ALU) Using VHDL/Verilog
o Objective: Implement an ALU capable of performing AND, OR, ADD, SUB, and SLT operations.
o Tool: Xilinx Vivado / ModelSim / Logisim
5. Simulate Single-Cycle and Multi-Cycle Datapaths
o Objective: Model a simplified processor datapath (single-cycle and pipelined) and compare performance.
o Tool: Digital Simulator / Logisim / Ripes (RISC-V simulator)
6. Pipeline Hazard Detection and Resolution
o Objective: Simulate and analyze data and control hazards in a pipelined processor; demonstrate stalling and forwarding.

Module 4 & 5: Memory Mapping and Management
7. Cache Memory Simulation
o Objective: Simulate direct-mapped, fully-associative, and set-associative caches; compute hit/miss ratio.
o Tool: Custom Python/C program or existing cache simulator tools.
8. Implementation of Virtual Memory using Paging
o Objective: Simulate address translation using page tables; demonstrate page faults and TLBs.
o Tool: Python or Java-based simulation
9. Disk Scheduling Algorithm Simulation
o Objective: Implement and compare performance of FCFS, SSTF, SCAN, and LOOK disk scheduling algorithms.
10. RAID Level Simulation and Performance Comparison
• Objective: Simulate different RAID levels (RAID 0, 1, 5) and evaluate redundancy, speed, and fault tolerance.

References:

1. David A. Patterson and John L. Hennessey, “Computer organization and design, The Hardware/Software interface”, Morgan Kauffman / Elsevier, Fifth edition, 2014
2. V. Carl Hamacher, Zvonko G. Varanesic, and Safat G. Zaky, “Computer Organization “, 6 th edition, McGraw-Hill Inc, 2012.
3. William Stallings, “Computer Organization and Architecture”, 8th Edition, Pearson Education, 2010

Course Code	Course Name	L	T	P	Cr
DOEE250183	Signals and Systems	2	1	2	4

Course Outcomes:

CO1: Explain the concepts of vector spaces, subspaces, linear dependence, bases, dimensions, and linear transformations with matrix representation.
CO2: Classify continuous-time and discrete-time signals and systems, and analyze their fundamental properties.
CO3: Apply Fourier Transform and Fourier Series for spectral analysis of signals and characterization of LTI systems.
CO4: Demonstrate sampling theorem, DTFT, DFT, and their properties for signal analysis in discrete domain.
CO5: Analyze continuous and discrete-time LTI systems using Laplace transform and Z-transform with stability considerations

Module 1:

Vector spaces: Field-definition-Finite field arithmetic. Vector Spaces – Subspaces – Linear Combinations - Linear Span – Linear Dependence - Linear Independence – Bases and Dimensions. Linear Transformations. Matrix representation of linear transformation.

Module 2:

Classification of signals and systems: Continuous time signals (CT signals)- Discrete time signals (DT signals) – Step, Ramp, Pulse, Impulse, Exponential - classification of CT and DT signals – periodic and aperiodic signals, random signals, Energy & Power signals - CT and DT systems, Classification of systems. LINEAR TIME INVARIANT SYSTEMS.

Module 3:

Spectral analysis of continuous time signals : Representation of a periodic Signals by continuous FT, FT of periodic signals, convolution and multiplication property of continuous FT, systems characterized by Linear Constant Coefficient Differential Equations. Magnitude and phase representation of FT, Magnitude and phase response of LTI systems, Time domain and Frequency domain aspects of ideal and non-ideal filters.
Fourier Series representation of periodic signals.

Module 4:

Sampling: Sampling theorem. Effect of under sampling. DTFT and Spectral analysis of sampled signal. DFT. Properties of DTFT and DFT, convolution property, multiplication property, Duality, Systems characterized by Linear Constant Coefficient Difference Equations.

Module 5:

Characterization of continuous time and discreate time LTI Systems: Definition-ROC-Properties-Inverse Laplace transforms-the S-plane and BIBO stability-Transfer functions- System Response to standard signals-Solution of differential equations with initial conditions. Application in continuous time signal and system analysis.

Z-transform, Region of convergence and its properties, Inverse Z transform, properties of ZT, Analysis and characterization of LTI systems using ZT, LTI Systems, System function algebra and block diagram representations. Application in discreate time system analysis.

Labs / Practicals:

Module 1: Signal Basics and LTI Systems

1. Generation and Visualization of Standard Signals

Objective: Generate and plot continuous-time and discrete-time signals: step, ramp, exponential, sinusoidal, and impulse.

Tools: MATLAB/Scilab/Python

2. Operations on Signals

Objective: Perform and plot operations like scaling, shifting, folding, and addition on signals.

3. Verification of Linearity and Time Invariance of Systems

Objective: Test whether a given discrete-time system is linear and time-invariant using example inputs and outputs.

Module 2 & 3: Fourier Series and Fourier Transform

4. Fourier Series Representation of Periodic Signals

Objective: Compute and plot the Fourier series coefficients of a square/triangular waveform and reconstruct the waveform using limited harmonics.

5. Computation and plotting of Fourier Transform

Objective: Find the Fourier Transform of a given time-domain signal (e.g., rectangular pulse) and plot its magnitude and phase spectrum.

Module 4: Laplace Transform

6. System Response using Laplace Transform

Objective: Use Laplace transform to find system response to standard inputs (step, impulse) and compare time-domain and Laplace-domain results.

Module 5: DTFT and DFT, Z transform

7. Sampling and Reconstruction of Signals

Objective: Demonstrate the effect of sampling and reconstruction using sinc interpolation. Analyze aliasing with under-sampling.

8. Compute and Plot DTFT and DFT of Discrete-Time Signals

Objective: Use MATLAB/Python to compute DTFT and DFT, and visualize the spectral content of a discrete-time signal.

9. Computation and Plotting of Z-Transform and Inverse Z-Transform

Objective: Determine and plot the Z-transform of basic sequences. Find inverse Z-transform using partial fractions.

Integrated Application

10. Analysis of LTI System using Convolution and Z-Transform

Objective: Implement an LTI system described by a difference equation. Use convolution and Z-transform to compute and verify the system's response.

References:

Alan V. Oppenheim, Alan S. Willsky, S. Hamid Nawab, Signals and Systems Prentice Hall India, 2nd Edition, 2009.

John G. Proakis, Dimitris G. Manolakis, Digital Signal Processing, Principles, Algorithms, and Applications, 4th Edition, PHI, 2007.

B.P. Lathi, —Signals, Systems & Communications‖, 2009,BS Publications.

Simon Hykin , " Signals and Systems", John Wiley

Robert A. Gable, Richard A. Roberts, Signals & Linear Systems, 3rd Edition, John Wiley, 1995.

Harish Parthasarathy,"Signals and Systems" JK International Second Edition.

HWEI P.HSU Schaum's, " Signals and Systems" TMH

M.J.Roberts,"Signals and Systems" TMH Ed 2003

Course Code	Course Name	L	T	P	Cr
DOEE250012	Analog Electronics	2	1	2	4

Course Outcomes:

CO1: Explain the characteristics, operating regions, and biasing techniques of BJTs and design stable biasing circuits.
CO2: Analyze small-signal models of BJTs and evaluate amplifier performance (gain, impedance, frequency response, bandwidth).
CO3: Illustrate the concepts of feedback in amplifiers and design sinusoidal oscillators using RC and LC circuits.
CO4: Demonstrate MOSFET characteristics, biasing methods, and analyze their amplifier configurations and digital applications.
CO5: Classify power amplifiers and voltage regulators, analyze their operation, efficiency, and design protection mechanisms.

Module 1:

BJT Characteristics and Biasing Techniques: Overview of BJT operation: active, cut-off, and saturation regions, Input and output characteristics of CE, CB, and CC configurations, DC load line, operating point, Biasing techniques: Fixed bias, Collector-to-base bias, Voltage divider bias, Stability factors and thermal runaway, Design of BJT biasing circuits.

Module 2:

BJT Small Signal Models and Amplifiers: Small signal equivalent models (hybrid-π and r_e models), CE, CB, CC amplifier analysis: voltage gain, input/output impedance, Multistage amplifier concepts: cascade and cascode configurations, Frequency response of BJT amplifiers: low and high-frequency analysis, Miller effect and bandwidth estimation.

Module 3:

Feedback Amplifiers and Oscillators: Concept of negative and positive feedback in amplifiers, Voltage series, voltage shunt, current series and current shunt feedback configurations, Oscillators: Barkhausen criterion, RC phase shift oscillator, Wien bridge oscillator, LC oscillators: Colpitts and Hartley.

Module 4:

MOSFET Characteristics and Amplifier Configurations:Overview of MOSFET types (Enhancement and Depletion), MOSFET characteristics: output and transfer characteristics, Biasing techniques for MOSFETs, Small signal analysis of common source, common drain, and common gate configurations. MOSFET as an amplifier and switch, Frequency response of MOSFET amplifiers, Comparison with BJT amplifier performance. Introduction to CMOS inverter and digital logic applications.
Analog applications: active loads, current sources, analog switches

Module 5:

Power Amplifiers and Linear Regulators: Classification of power amplifiers: Class A, B, AB, and C, Class A power amplifier with resistive and transformer load, Class B push-pull amplifier and crossover distortion, Class AB operation and efficiency.
Linear Voltage Regulators: Types of voltage regulators-series and shut. Working and design. Load and line regulation. Short circuit protection and fold back protection.

Labs / Practicals:

Design and Implementation of Voltage Divider Biasing Circuit
To design a voltage divider bias circuit for a BJT and determine its operating point (Q-point).

Single Stage CE Amplifier – Gain and Impedance Measurement
To design and implement a CE amplifier and measure voltage gain, input/output impedance.

Two-Stage RC Coupled Amplifier
To design and analyze a two-stage RC coupled amplifier and determine overall gain and bandwidth.

Frequency Response of CE Amplifier
To study low-frequency and high-frequency response and determine bandwidth and cutoff frequencies.

RC Phase Shift Oscillator using BJT
To design and implement a phase shift oscillator and verify frequency of oscillation using Barkhausen criterion.

Wien Bridge Oscillator using Op-Amp or BJT
To construct a Wien bridge oscillator and analyze output waveform and frequency.

Colpitts Oscillator using BJT
To design and implement a Colpitts oscillator circuit and verify its frequency of oscillation.

MOSFET Output and Transfer Characteristics
To obtain the output and transfer characteristics of an enhancement-type MOSFET and extract threshold voltage.

Common Source MOSFET Amplifier
To implement and analyze a common source amplifier and measure voltage gain and frequency response.

MOSFET as a Switch
To demonstrate the switching behavior of a MOSFET for digital signal interfacing.

Class A Power Amplifier with Transformer Coupling
To design and analyze a Class A power amplifier and determine efficiency.

Design and Testing of Linear Voltage Regulator Circuit
To design and implement a series-type voltage regulator and measure line/load regulation and short-circuit protection.

References:

1 Sedra, Adel S. and Smith, Kenneth C.
Microelectronic Circuits
Oxford University Press, India
2 Millman, Jacob and Halkias, Christos C.
Electronic Devices and Circuits
McGraw-Hill Education (India)
3 Boylestad, Robert L. and Nashelsky, Louis
Electronic Devices and Circuit Theory
Pearson Education, India
4 Mottershead, Allen
Electronic Devices and Circuits: An Introduction
Pearson Education, India
5 Salivahanan, S. and Kumar, N. Suresh
Electronic Devices and Circuits
McGraw-Hill Education (India)

Course Code	Course Name	L	T	P	Cr
DOEE260002	CMOS VLSI Design	3	0	2	4

Course Outcomes:

CO1: Understand MOS transistor theory, CMOS fabrication process, and technology parameters.
CO2: Analyze CMOS logic gate design for static and dynamic behavior.
CO3: Design CMOS combinational and sequential circuits at the transistor level.
CO4: Evaluate power dissipation, delay, and noise margins in CMOS circuits.
CO5: Apply design rules and layout techniques for CMOS integrated circuits.

Module 1:

MOS Transistor Theory and Fabrication

MOS capacitor: threshold voltage, flat band voltage, body effect. Long-channel MOSFET I-V characteristics: triode and saturation regions. Short-channel effects: velocity saturation, DIBL, punchthrough. CMOS fabrication process: oxidation, diffusion, ion implantation, CVD, PVD, CMP, lithography. Process parameters, lambda-based design rules.

Module 2:

CMOS Logic Design

Static CMOS inverter: voltage transfer characteristic, noise margins, static power. CMOS NAND and NOR gates, AOI, OAI complex gates. Transistor sizing for symmetric response. Pass transistor logic, transmission gate logic. Tri-state buffers, multiplexers using CMOS.

Module 3:

Dynamic CMOS and Power Dissipation

Dynamic CMOS logic: precharge-evaluate phases, charge sharing, ratioed logic, domino logic, NP-CMOS. Power dissipation: dynamic switching power, short-circuit power, static leakage power. Power-delay product, energy-delay tradeoff. Power reduction techniques: supply scaling, threshold voltage scaling.

Module 4:

CMOS Sequential Circuits

SR latch, D latch, D flip-flop (static and dynamic implementations). Master-slave flip-flop. Setup time, hold time, clock-to-Q delay. Timing analysis: setup and hold constraints. Registers and shift registers in CMOS. Clock distribution and skew.

Module 5:

CMOS Layout and Design Rules
Layout design rules: minimum width, minimum spacing, overlap rules. Stick diagrams, Euler graphs for optimal layout. Layout of CMOS inverter, NAND, NOR gates. Interconnect parasitics: resistance, capacitance, inductance. Elmore delay model. Design rules check (DRC), layout versus schematic (LVS) concepts.

Labs / Practicals:

1. CMOS inverter simulation: VTC and noise margin (using LTSPICE/NGSPICE).
2. CMOS NAND and NOR gate transistor sizing.
3. Transmission gate multiplexer design.
4. CMOS D flip-flop simulation and timing parameter extraction.
5. Setup and hold time violation simulation.
6. Layout of CMOS inverter using Microwind or similar tool.

7. DRC and LVS on a simple CMOS cell.
8. Dynamic CMOS domino logic gate design.
9. Power dissipation estimation using simulation.
10. Ring oscillator design and frequency measurement.

References:

1. Weste N.H.E. and Harris D. – CMOS VLSI Design: A Circuits and Systems Perspective, Pearson Education.
2. Uyemura J.P. – Introduction to VLSI Circuits and Systems, John Wiley & Sons.
3. Kang S-M. and Leblebici Y. – CMOS VLSI Design, McGraw-Hill.
4. Taur Y. and Ning T.H. – Fundamentals of Modern VLSI Devices, Cambridge University Press.

Course Code	Course Name	L	T	P	Cr
DASH250026	Design Thinking	1	0	2	2

Course Outcomes:

CO1 : Understand the core principles and process of design thinking including empathy, ideation, and prototyping.
CO2 : Apply user-centric research methods to identify and frame real-world problems.
CO3 : Generate and evaluate innovative ideas through brainstorming and collaborative design activities.
CO4 : Build and test low-fidelity prototypes to explore solution possibilities.
CO5 : Present and communicate design solutions effectively using appropriate tools and storytelling techniques.

Module 1:

Introduction to Design Thinking
Definition and Importance of Design Thinking, Comparison with Traditional Problem Solving, Mindsets for Innovation, Stanford d.school Design Process, Applications in Engineering, Business, and Social Sectors.

Module 2:

Empathize – Understanding Users
Importance of Empathy, Observation and Interview Techniques, User Persona Creation, Empathy Mapping, Journey Maps, Capturing Insights from Field Research.

Module 3:

Define and Ideate
Problem Definition and Framing: Point of View (POV) Statements, Ideation Techniques (Brainstorming, SCAMPER, Mind Mapping), Idea Evaluation and Selection Criteria.

Module 4:

Prototype and Test
Prototyping Tools and Techniques, Rapid Prototyping, Paper Prototypes, Role-Playing, Usability Testing, Feedback Integration and Iteration.

Module 5:

Communication and Implementation
Storyboarding, Pitching Ideas, Design Thinking for Entrepreneurship, Real-life Case Studies, Ethical and Sustainable Design Considerations.

Labs / Practicals:

1. Conduct user interviews and observations to develop empathy maps.
2. Create user personas and problem statements.
3. Brainstorm solutions for a chosen problem and organize ideas using mind maps.
4. Develop low-fidelity prototypes for the selected ideas.
5. Test prototypes with users and gather feedback for refinement.
6. Present project using storyboards and pitch decks.
7. Group project: Apply the design thinking process to a real-world challenge and document the journey.
8. Use tools like Canva, Miro, or Figma for visualizing and presenting ideas.
9. Role-play-based usability tests.
10. Peer review and reflection journal on the design process.

References:

1. Tim Brown, Change by Design, Harper Business.
2. Rolf Faste, Design Thinking Methodology, Stanford University Materials.
3. Jeanne Liedtka & Tim Ogilvie, Designing for Growth: A Design Thinking Toolkit for Managers, Columbia

Business School Publishing.
4. Nigel Cross, Design Thinking: Understanding How Designers Think and Work, Berg Publishers.
5. Plattner, Meinel, Leifer, Design Thinking: Understand – Improve – Apply, Springer.

Course Code	Course Name	L	T	P	Cr
DOEE250062	Digital IC Design	3	0	2	4

Course Outcomes:

CO1. Demonstrate proficiency in digital IC design concepts, methodologies, and tools.
CO2. Write synthesizable HDL code and simulate digital IC designs using industry-standard simulation tools.
CO3. Design and implement digital IC layouts that meet design requirements and constraints.
CO4. Convert digital IC designs into FPGA prototypes for verification and testing.

Module 1:

CMOS VLSI Design:
Overview of VLSI technology and its significance, Semicustom VLSI Design flow: frontend and back-end. CMOS Transistor Theory- MOSFET operation, characteristics and scaling. CMOS inverter operation and voltage transfer characteristics, Inverter sizing and optimization techniques

Module 2:

CMOS Logic Design:
Basic logic gates: AND, OR, NAND, NOR, XOR. Designing complex logic functions using CMOS logic gates. Gate Delays and Logical Effort- Gate delay modeling and estimation, logical effort analysis, Determining the best delay-power trade-off for logic gates using P/N ratio. Transistor Level Schematics and Layouts- Transistor level design techniques, Schematic design and layout considerations, Design rules, metal layers, and interconnect routing.

Module 3:

Hardware modelling and verification using Verilog HDL:
Introduction to Verilog HDL- syntax and constructs. Hardware modelling-behavioral, dataflow and structural. Register Transfer Level (RTL) design of digital circuits-modeling using Verilog HDL. Functional verification overview. Writing Verilog test benches and RTL verification

Module 4:

RTL to GDSII:
RTL to GDSII design flow. Overview of Logic synthesis. RTL coding for synthesis. Constraints and RTL synthesis. Physical design- Floor plan and power planning, Placement and clock tree synthesis, Routing, Physical Verification and DFM Checks, static timing analysis and ECO fixing, timing closure and Tape out

Module 5:

FPGA Prototyping of Digital ICs:
Overview of FPGAs-architecture and components. Advantages of FPGAs. FPGA design flow-design entry, synthesis, floor planning, placement, routing, and bitstream generation. Programming FPGA. Embedded Logic analyzers and debugging. Xilinx 7-series FPGA-a case study.

Labs / Practicals:

1. Modelling and simulation of CMOS Inverter
2. Modelling and simulation of basic gates.
3. CMOS Inverter layout design, simulation and analysis.
4. Layout design and analysis of basic gates.
5. Hardware modelling and simulation of ALU, 8 bits up down counter.

6. Hardware modeling and verification of a decade counter.
7. RTL synthesis and analysis of a decade counter.
8. RTL to GDSII generation of a decade counter
9. FPGA programming of a decade counter
10. Decade counter debugging in FPGA fabric using Embedded logic analyzers

References:

1. Digital Integrated Circuits – A Design Perspective, Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolic, 2nd Ed., PHI.
2. Introduction to VLSI Systems: A Logic, Circuit and System Perspective – Ming-BO Lin, CRC Press, 2011.
3. Verilog HDL - A guide to Digital Design and Synthesis by Samir Palnitkar.
4. Writing Testbenches: Functional Verification of HDL Models by Janick Bergeron
5. FPGA-Based System Design by Wayne Wolf
6. Advanced FPGA Design Architecture, Implementation and optimization by Kilts

Course Code	Course Name	L	T	P	Cr
DOEE260010	FPGA Design and Implementation	2	0	4	4

Course Outcomes:

CO1: Understand FPGA architecture and its reconfigurable resources.
CO2: Implement digital designs on FPGA using hardware description languages and EDA tools.
CO3: Utilize FPGA-specific resources: DSP slices, BRAM, clock management tiles.
CO4: Develop an FPGA prototype for a real-time digital system.
CO5: Analyze and optimize FPGA design for timing, power, and area.

Module 1:

FPGA Architecture

FPGA overview: history, market, applications. Look-up table (LUT) based architecture. Xilinx and Intel FPGA families. Configurable Logic Block (CLB): LUTs, flip-flops, carry chains, multiplexers. Interconnect architecture: switch matrices, routing resources. I/O blocks, IDELAYCTRL. Configuration memory and bitstream.

Module 2:

FPGA Design Flow

FPGA design flow: RTL entry → synthesis → implementation → bitstream → download. Xilinx Vivado or Intel Quartus Prime overview. IP core integration: Vivado IP Catalog. Constraint files: XDC/SDC (timing and pin constraints). Design rule checks, timing analysis in FPGA tools.

Module 3:

Advanced FPGA Resources

Block RAM (BRAM): single-port, dual-port, configuration. DSP slices: multiplier-accumulator (MAC) unit mapping. Clock management: PLL, MMCM, global clock buffers, clock gating. SERDES for high-speed I/O. Hard IP: PCIe, Ethernet, DDR controllers.

Module 4:

High-Level Synthesis

Introduction to High-Level Synthesis (HLS): concept and motivation. HLS tools: Xilinx Vitis HLS, Intel HLS Compiler. C/C++ to RTL flow. Directives and pragmas for pipeline, unroll, dataflow. Interface synthesis: AXI master, AXI stream. Performance estimation and area report.

Module 5:

FPGA Applications and Prototyping

Signal processing on FPGA: FIR filter, FFT core. Soft-core processors: MicroBlaze, NIOS II, RISC-V. FPGA-based image processing: sobel edge detector. Partial reconfiguration concept. FPGA in embedded systems: Zynq SoC (PS-PL interaction). Real-time control system prototyping on FPGA.

Labs / Practicals:

1. LED blinking and switch debouncing on FPGA board.
2. 7-segment display controller with multiplexed digits.
3. 4-bit binary counter with enable and reset.
4. PWM generator for motor/LED brightness control.
5. UART transmitter and receiver implementation.
6. SPI master controller for interfacing sensor.

7. FIR low-pass filter using DSP slice (test with on-board ADC if available).
8. BRAM-based FIFO implementation.
9. VGA display controller (text or simple graphics).
10. Mini project: FPGA-based real-time digital clock or reaction timer

References:

1. Chu P.P. – FPGA Prototyping by Verilog Examples, John Wiley & Sons.
2. Maxfield C. – The Design Warrior's Guide to FPGAs, Newnes.
3. Xilinx Vivado Design Suite User Guide (UG910, UG949).
4. Intel Quartus Prime Design Software Handbook.

Course Code	Course Name	L	T	P	Cr
DOEE250015	Analog IC Design	3	1	0	4

Course Outcomes:

CO1. To acquaint the students with basic CMOS analog building blocks and sub-system design.
CO2. To develop the ability to design and analyze MOS based Analog VLSI circuits to draw the equivalent circuits of MOS based Analog VLSI and analyze their performance

Module 1:

Introduction: Basic MOS Device Physics – General Considerations, MOS I/V Characteristics, Second Order effects, MOS Device models. Short Channel Effects and Device Models. Single Stage Amplifiers – Basic Concepts, Common Source Stage, Source Follower, Common Gate Stage, Cascode Stage.

Module 2:

Current Mirrors, Current and Voltage Reference:
Basic current mirrors, cascode current mirrors, active current mirrors, low current biasing, supply insensitive biasing, temperature insensitive biasing, impact of device mismatch

Module 3:

Frequency Response of Amplifiers:
Miller effect, CS amplifier, source follower, CG amplifier, cascade stage, differential amplifier, Multistage amplifier

Module 4:

Operational Amplifiers: Performance parameters, One-stage and two stage Op Amps, gain boosting, comparison, common mode feedback, input range, slew rate, power supply rejection, noise in Op Amps, Buffered Op-amps, High speed/Frequency Op-amps

Module 5:

Feedback Amplifiers:
General Considerations, Feedback Topologies, Effect of Loading. Operational Amplifiers – General Considerations, One Stage Op Amps, Two Stage Op Amps, Gain Boosting, Common – Mode Feedback, Input Range limitations, Slew Rate, Power Supply Rejection, Noise in Op Amps. Stability and Frequency Compensation

Labs / Practicals:

NA

References:

1. B.Razavi, “Design of Analog CMOS Integrated Circuits”, 2nd Edition, McGraw Hill Edition 2016
2. Paul. R.Gray and Robert G. Meyer, “Analysis and Design of Analog Integrated Circuits”, Wiley, 5th Edition, 2009.
3. R. Jacob Baker, “CMOS Circuit Design, Layout, and Simulation”, 3rd Edition, Wiley, 2010.
4. T. C. Carusone, D. A. Johns and K. Martin, “Analog Integrated Circuit Design”, 2nd Edition, Wiley, 2012
5. P.E.Allen and D.R. Holberg, “CMOS Analog Circuit Design”, 3rd Edition, Oxford University Press, 2011.

Course Code	Course Name	L	T	P	Cr
DOEE260009	AI/ML for Hardware	3	0	2	4

Course Outcomes:

CO1: Apply fundamental concepts of machine learning and deep learning to analyze and design AI-driven hardware optimization pipelines.
CO2: Evaluate hardware-aware neural network design strategies including quantization, pruning, and knowledge distillation for resource-constrained deployment.
CO3: Implement and analyze neural network accelerator architectures such as systolic arrays, dataflow architectures, and in-memory computing paradigms.
CO4: Design and optimize FPGA and ASIC-based inference engines using HLS tools and hardware-software co-design methodologies.
CO5: Apply AI/ML techniques to automate EDA tasks including floorplanning, placement, routing, and timing closure in VLSI design flows.
Got it — here's your syllabus rewritten as continuous text (not in table form), following the same style as your earlier reference:

Module 1:

Foundations of ML & Deep Learning for Hardware

Supervised, unsupervised, and reinforcement learning overview; regression, classification, and clustering with hardware context; feature engineering for hardware datasets; ML pipeline for EDA: dataset preparation, model training, evaluation; introduction to scikit-learn and PyTorch for hardware applications; hardware performance metrics as ML targets (PPA: Power, Performance, Area); feedforward networks, CNNs, RNNs, LSTMs, Transformers; computational graphs and tensor operations; hardware-aware architecture design; roofline model and operational intensity; memory hierarchy and bandwidth bottlenecks; workload characterization for DNN accelerators.

Module 2:

Model Compression and Hardware-Efficient AI

Quantization: fixed-point, INT8, mixed-precision, quantization-aware training; pruning: structured and unstructured, magnitude and gradient-based; weight sharing and knowledge distillation; neural architecture search (NAS) for hardware constraints; TensorRT, ONNX, TFLite deployment; benchmarking on edge devices; MLPerf inference benchmarks; FPGA resource-aware model design (LUT/DSP budgeting context).

Module 3:

Hardware Accelerator Architecture

Systolic arrays and matrix multiply units (Google TPU, NVIDIA TensorCore); dataflow paradigms: weight stationary, output stationary, row stationary; memory hierarchy and bandwidth bottlenecks for DNN accelerators; in-memory and near-memory computing; Processing-in-Memory (PIM) architectures; reconfigurable architectures — FPGAs for DNN inference; case studies: Google TPU, NVIDIA GPU, Apple Neural Engine, Graphcore IPU; design space exploration of accelerators; batch normalization, dropout — hardware implications.

Module 4:

FPGA and ASIC Implementation of AI Models

HLS-based design flow: Vivado HLS / Vitis HLS; hls4ml framework for neural network synthesis; FPGA resource estimation: LUT, DSP, BRAM budgeting; pipelining and loop unrolling for throughput; hardware-software co-design and partitioning; ASIC custom accelerator design flow; place and route for AI accelerator blocks; power analysis and estimation for inference engines.

Module 5:

AI/ML for EDA and VLSI Design Automation

ML-driven floorplanning and placement (Google Chip Placement with RL); GNN-based congestion and timing prediction; ML for PD signoff: IR drop, EM, CTS optimization; reinforcement learning for routing; generative models for standard cell layout; AI-assisted ECO (Engineering Change Order); industry case studies: Synopsys DSO.ai, Cadence Cerebrus, NVIDIA Perforce.

Labs / Practicals:

1. ML model training for timing prediction using open-source VLSI dataset (OpenROAD) — Python / scikit-learn
2. CNN inference workload profiling: roofline analysis and bottleneck identification — PyTorch / NVIDIA Nsight
3. INT8 quantization of a pre-trained ResNet model and accuracy-latency trade-off study — TensorRT / TFLite
4. Structured pruning of a CNN and re-training to recover accuracy — PyTorch
5. Systolic array simulation for matrix multiplication — cycle-accurate modeling — Python / SCALE-Sim
6. DNN accelerator design using hls4ml — synthesize a small MLP to FPGA — Vivado HLS / hls4ml
7. FPGA resource optimization: loop unrolling and pipelining for CNN layer — Vitis HLS
8. Congestion map prediction using GNN on standard benchmark circuits — PyTorch Geometric
9. RL-based floorplan exploration on a small netlist (mimic Google chip placement) — OpenAI Gym / Python
10. End-to-end mini project: train a timing closure predictor using RTL features — Python / OpenROAD

References:

1. Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. S. — Efficient Processing of Deep Neural Networks, MIT Press / Morgan & Claypool, 2020.
2. Patterson, D., & Hennessy, J. — Computer Organization and Design (ARM / RISC-V Edition), Morgan Kaufmann, 2021.
3. Goodfellow, I., Bengio, Y., & Courville, A. — Deep Learning, MIT Press, 2016.
4. Weste, N. H. E., & Harris, D. — CMOS VLSI Design: A Circuits and Systems Perspective, 4th Ed., Pearson, 2015.
5. Huang, G. et al. — Machine Learning for Electronic Design Automation: A Survey, ACM TODAES, 2021.
6. Mirhoseini, A. et al. — A Graph Placement Methodology for Fast Chip Design, Nature, 2021.
7. Synopsys — DSO.ai Documentation; Cadence — Cerebrus Intelligent Chip Explorer Documentation (vendor references).
8. IIT Madras NPTEL — VLSI Design; IIT Bombay NPTEL — Deep Learning, online course materials.

Course Code	Course Name	L	T	P	Cr
DASH250104	Technical Presentation and Report Writing	0	0	4	2

Course Outcomes:

CO1 : Understand the principles and formats of technical communication including different types of reports and presentations.
CO2 : Prepare clear, concise, and well-structured technical documents for academic and professional purposes.
CO3 : Design and deliver effective oral presentations using appropriate tools and visual aids.
CO4 : Demonstrate proficiency in grammar, technical vocabulary, and formal style required in technical writing.
CO5 : Collaborate in teams to produce technical content and receive peer feedback constructively.

Module 1:

Introduction to Technical Communication
Principles and characteristics of technical communication, Purpose and types of technical documents, Features of a good technical report, Audience analysis.

Module 2:

Writing Skills for Technical Documents
Structure of reports: Title Page, Abstract, Table of contents, Introduction, Body, Conclusion, and References. Writing instructions, lab reports, Project Reports, and Proposals. Use of charts, graphs, and tables.

Module 3:

Grammar and Language in Technical Writing
Use of formal tone and objective language, sentence construction, common errors, punctuation, and vocabulary building for technical writing. Editing and proofreading techniques.

Module 4:

Preparing Technical Presentations
Planning a presentation, structuring content, use of slides, visual aids, and multimedia tools (e.g., PowerPoint, Prezi). Non-verbal communication and handling audience questions.

Module 5:

Practice and Evaluation
Student individual and group presentations on technical topics, peer reviews, instructor feedback. Writing mini technical reports based on lab/project work or case studies.

Labs / Practicals:

1. Analyzing Samples of Technical Documents
Objective: Identify the structure and features of different technical documents (e.g., lab report, project report, instruction manual).
2. Writing a Lab Report Based on Practical Work
Objective: Write a complete lab report including abstract, objectives, procedure, observations, and conclusion.
3. Creating a Project Proposal Document
Objective: Prepare a short technical proposal on a selected topic including problem statement, objectives, methodology, and timeline.
4. Designing Visual Elements (Graphs, Charts, Tables)
Objective: Use data to create visual aids using Excel/PowerPoint and integrate them into a technical report.
5. Grammar and Proofreading Workshop
Objective: Identify and correct errors in sample texts focusing on sentence construction, punctuation, and vocabulary.
6. Preparing and Delivering an Individual Presentation
Objective: Create and present a 5-minute talk on a technical topic using PowerPoint or Prezi.

7. Group Presentation on a Technical Case Study
Objective: Collaboratively research and present findings on a real-world technical issue or innovation.
8. Peer Review and Feedback Session
Objective: Evaluate and provide constructive feedback on peers' written reports and oral presentations.
9. Editing and Improving a Poorly Written Report
Objective: Revise and rewrite a given flawed report to meet academic/professional standards.
10. Writing a Mini Technical Report on a Chosen Topic
Objective: Submit a short report (3–5 pages) on a lab experiment, research article, or emerging technology.

References:

1. Meenakshi Raman and Sangeeta Sharma – Technical Communication: Principles and Practice, Oxford University Press.
2. M. Ashraf Rizvi – Effective Technical Communication, Tata McGraw-Hill.
3. Sharma, R.C. and Krishna Mohan – Business Correspondence and Report Writing, Tata McGraw-Hill.
4. Lesikar, Raymond V. et al. – Basic Business Communication, Tata McGraw-Hill.
5. David F. Beer and David McMurrey – A Guide to Writing as an Engineer, Wiley India.

Course Code	Course Name	L	T	P	Cr
DASH250030	Economics for Engineers	1	1	0	2

Course Outcomes:

CO1 : Understand basic economic concepts, including demand, supply, cost, and market structures.
CO2 : Analyze the economic feasibility of engineering projects using cost-benefit and break-even analysis.
CO3 : Evaluate different investment alternatives using engineering economic tools like NPV, IRR, and Payback Period.
CO4 : Apply knowledge of macroeconomic indicators and fiscal/monetary policies in the context of engineering.
CO5 : Develop the ability to assess the economic impact of engineering decisions in real-world scenarios.

Module 1:

Basic Economic Concepts
Introduction to Economics: Micro vs Macro, Utility, Demand and Supply Analysis, Elasticity of Demand and Supply, Equilibrium, Law of Diminishing Returns, Opportunity Cost.

Module 2:

Cost and Production Analysis
Types of Costs: Fixed, Variable, Marginal, Average, Total, Cost-Output Relationship, Break-even Analysis, Law of Variable Proportions, Returns to Scale, Cost-Volume-Profit Analysis.

Module 3:

Market Structures and Pricing
Types of Market: Perfect Competition, Monopoly, Oligopoly, Monopolistic Competition, Pricing Strategies, Price Discrimination, Government Policies and Pricing.

Module 4:

Engineering Economics
Time Value of Money, Present Worth and Future Worth, Net Present Value (NPV), Internal Rate of Return (IRR), Payback Period, Comparison of Investment Alternatives, Depreciation and Inflation Considerations.

Module 5:

Macroeconomic Concepts
National Income, GDP, GNP, Inflation, Deflation, Unemployment, Fiscal and Monetary Policy, Role of RBI and Government in Economic Planning, Economic Indicators Relevant to Engineers.

Labs / Practicals:

N/A

References:

1. R. Panneerselvam, Engineering Economics, PHI Learning.
2. D.M. Mithani, Managerial Economics, Himalaya Publishing House.
3. M.L. Jhingan, Microeconomic Theory, Vrinda Publications, Latest Edition.
4. Chan S. Park, Contemporary Engineering Economics, Pearson Education.
5. Paul A. Samuelson and William D. Nordhaus, Economics, McGraw Hill.

Course Code	Course Name	L	T	P	Cr
DOEE260005	Physical Design of VLSI Circuits	2	0	4	4

Course Outcomes:

CO1: Understand the VLSI physical design flow from netlist to GDSII.
CO2: Perform floor planning, power planning, and pin assignment.
CO3: Analyze and execute placement algorithms and optimization techniques.
CO4: Perform clock tree synthesis and routing in multi-layer designs.
CO5: Apply design rule checking (DRC), layout versus schematic (LVS), and parasitic extraction.

Module 1:

Physical Design Flow Overview

Physical design flow: netlist input → floorplan → placement → CTS → routing → signoff. Technology file: LEFF files, design rules, LEF/DEF formats. Standard cell library: cell view, timing model, power model. Design partitioning, hierarchical design. Tool flow: Cadence Innovus, Synopsys ICC2 overview.

Module 2:

Floor planning and Power Planning

Floor planning objectives: aspect ratio, die area estimation. Block placement, macro placement, blockages. I/O pad placement. Power planning: IR drop analysis, power grid design, ring and stripe strategy. Decoupling capacitors. Well taps and end-cap cells. Aspect ratio and utilization factor.

Module 3:

Placement Algorithms

Placement objectives: timing, wirelength, routability. Placement algorithms: min-cut bipartitioning (Fiduccia-Mattheyses), simulated annealing, force-directed. Legalization and detailed placement. Timing-driven placement. Congestion analysis and spreading. Placement quality metrics.

Module 4:

Clock Tree Synthesis and Routing

Clock tree objectives: minimize skew, latency, power. Clock tree topologies: H-tree, X-tree, spine, fishbone. CTS algorithms, clock buffers and inverters. Post-CTS optimization: hold fixing, useful skew. Global routing: maze routing, channel routing. Detailed routing: Lee algorithm, A* algorithm. Metal layer assignment, via minimization.

Module 5:

Signoff and Verification

Parasitic extraction: RC extraction, SPEF file. Post-layout STA: MMMC analysis, on-chip variation (OCV). IR drop simulation. Physical verification: DRC, LVS, ERC. Antenna checking, density checks. Tape-out flow: GDSII generation, data preparation, foundry submission. Yield analysis basics.

Labs / Practicals:

1. Floor planning of a small design block using Cadence Innovus or open-source tool.
2. Power grid design and IR drop analysis.
3. Placement of a standard cell design and congestion analysis.
4. CTS: clock tree construction and skew measurement.
5. Global routing and layer assignment.
6. DRC on a CMOS cell using Calibre or equivalent.

7. LVS verification on a simple circuit.
8. SPEF-based post-layout STA.
9. Parasitic extraction and timing comparison (pre vs post layout).
10. End-to-end P&R of a small combinational block.

References:

1. Lavagno L., Scheffer L., Martin G. – Electronic Design Automation for IC Implementation, Circuit Design, and Process Technology, CRC Press.
2. Sherwani N.A. – Algorithms for VLSI Physical Design Automation, Springer.
3. Cadence Innovus User Guide (selected chapters).
4. Rabaey J., Chandrakasan A., Nikolic B. – Digital Integrated Circuits, Pearson Education.

Course Code	Course Name	L	T	P	Cr
DOEE260008	Semiconductor Fabrication	3	0	2	4

Course Outcomes:

CO1: Describe the fundamental principles of semiconductor physics, crystal growth techniques, and wafer preparation processes used in microelectronics manufacturing.
CO2: Explain and apply various thin film deposition techniques such as CVD, PVD, thermal oxidation, and epitaxy to model process parameters.
CO3: Implement photolithography and etching processes to pattern semiconductor substrates and analyze the effect of process variations on device geometry.
CO4: Analyze diffusion, ion implantation, and annealing processes to design doping profiles and evaluate their impact on device electrical
CO5: Integrate metallization, interconnect formation, and packaging processes to evaluate complete device fabrication flows and propose process improvements.

Module 1:

Introduction to Semiconductor Technology and Crystal Growth

Overview of semiconductor industry and VLSI technology roadmap (Moore's Law, ITRS); classification of semiconductor materials — elemental (Si, Ge) and compound (GaAs, InP, GaN); crystal structure, Miller indices, and wafer orientation; point and line defects in crystals; silicon wafer specifications (resistivity, doping, orientation, flat/notch).

Module 2:

Thermal Oxidation and Thin Film Deposition

Thermal oxidation of silicon: dry and wet oxidation; Deal-Grove model; oxidation kinetics; oxide charges; applications in gate dielectric and field oxide formation.

Module 3:

Photolithography and Pattern Transfer

Photolithography is the core patterning technique in IC fabrication, enabling the transfer of circuit patterns from masks to wafers with nanometer precision

Module 4:

Doping Processes — Diffusion and Ion Implantation

Controlled introduction of impurities (dopants) into semiconductor substrates to modify electrical properties is central to device fabrication.

Module 5:

Metallization, Interconnects, Packaging, Process Integration and Device Fabrication

Formation of electrical contacts, metal interconnect layers, and final device encapsulation to complete the fabrication process. End-to-end integration of individual process steps to fabricate complete semiconductor devices.

Labs / Practicals:

1. Cleanroom orientation, safety protocols, and contamination control
2. RCA cleaning of silicon wafers and surface preparation
3. Dry and wet thermal oxidation of silicon wafers; oxide thickness measurement

4. Thin film deposition by RF sputtering; film thickness measurement
5. Spin coating of photoresist; soft bake; exposure and development
6. Wet etching of SiO2 and metal films; etch rate determination
7. Four-point probe measurement of sheet resistance; resistivity calculation
8. Diffusion simulation using TCAD (Sentaurus/Silvaco) — dopant profile
9. CMOS process flow simulation — complete device fabrication walk-through
10. Mini-project: fabrication report and presentation on selected process module

References:

1. S. M. Sze and M. K. Lee, Semiconductor Devices: Physics and Technology, 3rd ed., John Wiley & Sons, 2012.
2. C. Y. Chang and S. M. Sze (Eds.), ULSI Technology, McGraw-Hill, 1996.
3. R. C. Jaeger, Introduction to Microelectronic Fabrication, 2nd ed., Prentice Hall, 2002.
4. D. Neamen, Semiconductor Physics and Devices, 4th ed., McGraw-Hill, 2011.
5. W. Maly, Atlas of IC Technologies, Benjamin-Cummings, 1987.
6. S. Wolf and R. N. Tauber, Silicon Processing for the VLSI Era, Vols. 1–4, Lattice Press.
P. van Zant, Microchip Fabrication, 6th ed., McGraw-Hill, 2014.

Course Code	Course Name	L	T	P	Cr
DOEE260011	Embedded Systems Design for Engineers	3	0	2	4

Course Outcomes:

CO1: Understand embedded system architecture and distinguish between MCU, DSP, and FPGA platforms.
CO2: Write embedded C programs for ARM Cortex-M microcontrollers with peripherals.
CO3: Design real-time systems using RTOS concepts: tasks, scheduling, and IPC mechanisms.
CO4: Design low-power embedded systems using hardware and software power management.
CO5: Apply IoT and communication protocols in embedded system design.

Module 1:

Embedded System Architecture

Embedded systems definition, characteristics, and applications. Microcontroller vs. microprocessor vs. DSP vs. FPGA. ARM Cortex-M architecture: pipeline, registers, exception model. Memory map, bus interfaces: AHB, APB. CMSIS and HAL libraries. Development tools: Keil MDK, STM32CubeIDE.

Module 2:

Peripheral Interfacing and Programming

GPIO, timers, PWM generation and input capture. Serial communication: UART, SPI, I2C protocols and programming. ADC and DAC interfacing. DMA controllers: use cases, configuration. Interrupt handling: NVIC, priority, preemption. External peripherals: sensors (temperature, accelerometer), displays.

Module 3:

Real-Time Operating Systems

RTOS concepts: tasks, threads, scheduling (preemptive, cooperative). FreeRTOS: task creation, priorities, task states. Synchronization: semaphores, mutexes, event groups. Message queues and mailboxes. Memory management in RTOS. Deadlock and priority inversion. Timing analysis: worst-case execution time (WCET).

Module 4:

Power Management and Safety

MCU power modes: sleep, deep sleep, standby. Peripheral power gating. Low-power programming techniques. Watchdog timers and system health monitoring. Functional safety: IEC 61508, ISO 26262 (overview). Error detection: ECC memory, CRC. Safe state design. Secure boot and firmware update.

Module 5:

Connectivity and IoT

Wireless protocols: Wi-Fi (ESP32), Bluetooth LE, Zigbee, LoRa. MQTT and CoAP protocol implementation. HTTPS and TLS for embedded security. Cloud connectivity: AWS IoT, Azure IoT Hub. Edge AI on microcontrollers: TensorFlow Lite for Microcontrollers (TFLite Micro). Gesture recognition, keyword spotting on MCU. CI/CD pipeline for embedded firmware.

Labs / Practicals:

1. GPIO toggle and LED control on STM32/ESP32 using HAL.
2. UART communication: echo and formatted output.
3. I2C interfacing of temperature sensor (e.g., BMP280).
4. SPI interfacing of an SD card or display.
5. ADC reading from potentiometer with display output.

6. PWM generation for servo motor control.
7. FreeRTOS task creation: LED blinking with two tasks.
8. FreeRTOS semaphore-based task synchronization.
9. Low-power mode demonstration: measure current in sleep vs. active.
10. Mini project: temperature + humidity monitoring system with MQTT publishing.

References:

1. Yiu J. – The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors, Elsevier.
2. Mazidi M.A., Mazidi J.G., McKinlay R.D. – The 8051 Microcontroller and Embedded Systems, Pearson.
3. Barry R. – Mastering the FreeRTOS Real Time Kernel, Real Time Engineers Ltd.
4. STMicroelectronics STM32 Reference Manuals and HAL Driver Documentation.

Course Code	Course Name	L	T	P	Cr
DASH250049	Entrepreneurial Development and Innovation	1	1	0	2

Course Outcomes:

CO1: Describe fundamental concepts and the ecosystem of entrepreneurship and innovation in the Indian context.
CO2: Analyze market opportunities and prepare a basic business model canvas.
CO3: Apply innovation frameworks and ideation techniques for product or service development.
CO4: Explain government schemes, startup support systems, and funding mechanisms.
CO5: Demonstrate teamwork, presentation, and proposal writing skills through entrepreneurial assignments.

Module 1:

Introduction to Entrepreneurship:
Definition, characteristics and types of entrepreneurs, entrepreneurial mindset, role of entrepreneurship in economic development, examples of Indian entrepreneurs in electronics, IT, and embedded domains, relevance for Diploma and B.Tech engineers.

Module 2:

Innovation and Design Thinking:
Definition of innovation, types (product, process, business model), TRIZ basics, Design Thinking framework, Ideation techniques, Creative problem solving, Case studies from Indian startups using electronics/ICT innovations.

Module 3:

Business Model and Opportunity Analysis:
Idea screening, opportunity identification, Value proposition design, Business model canvas, Basics of marketing, pricing and distribution, competitors and SWOT analysis, Application to electronics/service/product ideas.

Module 4:

Startup Ecosystem and Support in India:
Startup India initiative, Digital India, Make in India, Atmanirbhar Bharat, MeitY TIDE scheme, role of MSME, DST, AIM, incubation centers, accelerators, Funding sources – angel, VC, Bank loans, grants, Basics of intellectual property (IPR).

Module 5:

Proposal Writing, Teamwork and Entrepreneurial Communication:
Writing simple project proposals (idea, team, cost, timeline), team formation and role distribution, pitching and elevator pitch, communication and leadership skills for entrepreneurs, tools like Canva, PPT, video pitch.

Labs / Practicals:

N/A

References:

1. Vasant Desai - Dynamics of Entrepreneurial Development and Management, Himalaya Publishing.
2. C.B. Gupta & N.P. Srinivasan - Entrepreneurship Development, Sultan Chand & Sons.
3. S.S. Khanka - Entrepreneurial Development, S. Chand.

Course Code	Course Name	L	T	P	Cr
DOEE260012	Advanced VLSI Design and Low Power Techniques	3	0	2	4

Course Outcomes:

CO1: Understand sources of power dissipation in CMOS and apply reduction techniques.
CO2: Design low-power digital circuits using voltage scaling, clock gating, and power gating.
CO3: Apply adiabatic and sub-threshold circuit design techniques.
CO4: Design for reliability: manage electromigration, oxide breakdown, hot carriers, and latch-up.
CO5: Understand advanced node design challenges: FinFET, gate-all-around, and 3D ICs.

Module 1:

Power Dissipation Analysis

CMOS power: dynamic (switching), short-circuit, static (leakage) power. Activity factor estimation. Power analysis tools. Power-delay product, energy-delay product. Voltage and frequency scaling: dynamic voltage and frequency scaling (DVFS). Power domains, voltage islands. Adaptive body biasing.

Module 2:

Low-Power Design Techniques

Multi-threshold CMOS (MTCMOS), power gating, sleep transistors. Clock gating: coarse-grain and fine-grain. Operand isolation, memory banking. Data encoding for bus power reduction: bus invert coding, gray coding. Approximate computing for power reduction. Power management units (PMUs) in SoCs.

Module 3:

Sub-threshold and Adiabatic Design

Sub-threshold operation: WID variations, energy-efficiency. Sub-threshold logic design: CMOS, domino. Ultra-low-power design examples. Adiabatic logic: 2N-2N2P, ECRL, PAL. Energy recovery principles. Comparison with conventional CMOS.

Module 4:

Reliability in VLSI

Electromigration: mechanism, Black's equation, design rules. Time-dependent dielectric breakdown (TDDB). Hot carrier injection (HCI). Negative bias temperature instability (NBTI), PBTI. Latch-up: mechanism, guard rings. ESD protection structures. Soft errors: alpha particles, neutrons, SER mitigation. Aging-aware design.

Module 5:

Advanced Node Design

FinFET design: multi-fin, fin pitch, fin height. FinFET cell design: standard cells, SRAM. Gate-all-around (GAA) FET and nanosheet transistors. 2D material-based FETs (overview). 3D ICs: through-silicon vias (TSVs), wafer bonding, thermal challenges. Chiplet design: die-to-die interconnects (UCIe, AIB). Near-memory and in-memory computing concepts.

Labs / Practicals:

1. CMOS power dissipation analysis with simulation tools (Synopsys PrimePower or equivalent).
2. Clock gating implementation in RTL and power comparison.
3. Power domain and power intent coding using UPF.
4. DVFS implementation: variable clock/voltage simulation.
5. Sub-threshold logic gate simulation at sub-100mV supply.

6. FinFET standard cell simulation in SPICE (using BSIM-CMG models).
7. Electromigration-aware routing guidelines analysis.
8. SER estimation for flip-flop arrays.
9. Power gating: retention flip-flop design.
10. Mini project: low-power ALU with clock and operand gating.

References:

1. Rabaey J., Chandrakasan A., Nikolic B. – Digital Integrated Circuits: A Design Perspective, Pearson.
2. Roy K. and Prasad S.C. – Low Power CMOS VLSI Circuit Design, John Wiley & Sons.
3. Weste N.H.E. and Harris D. – CMOS VLSI Design, 4th ed., Pearson.
4. Keane J. and Kim C.H. – Transistor Aging, IEEE Spectrum.

Course Code	Course Name	L	T	P	Cr
DOEE260007	Testing and Verification of VLSI Circuits	2	0	4	4

Course Outcomes:

CO1: Understand the economic and technical challenges of VLSI testing.
CO2: Apply fault models and test generation algorithms for digital circuits.
CO3: Implement design-for-testability (DFT) techniques including scan and BIST.
CO4: Understand functional verification methodologies: simulation, emulation, and formal.
CO5: Apply SVA (SystemVerilog Assertions) and UVM-based verification methods.

Module 1:

Introduction to VLSI Testing

Testing motivation: defect density, yield, test economics. Fault models: stuck-at, bridging, delay, transient faults. Fault simulation methods: serial, parallel, deductive, concurrent. Fault collapsing, fault equivalence. Test quality metrics: fault coverage, ATPG efficiency. IC life-cycle testing: wafer, package, board.

Module 2:

Test Pattern Generation

Combinational ATPG: D-algorithm, PODEM, FAN algorithm. Boolean satisfiability (SAT)-based ATPG (overview). Sequential circuit test generation: test sequence derivation. Redundancy identification. Path delay fault testing. Test compaction and test ordering.

Module 3:

Design for Testability

Ad hoc DFT techniques. Structured DFT: full-scan, partial-scan. Scan chain architecture: multiplexed flip-flop, clocked-scan. Scan compression: X-bounding. Memory BIST: MARCH algorithms, LBIST architecture. Logic BIST: LFSR, multiple-input signature register (MISR). IEEE 1149.1 (JTAG) boundary scan

Module 4:

Functional Verification

Verification plan, coverage metrics: code coverage, functional coverage. Directed testing vs. constrained random verification. HDL simulation: Cadence Xcelium, Synopsys VCS. Formal verification: model checking, property checking. Equivalence checking: RTL vs gate-level. Emulation and rapid prototyping on FPGA.

Module 5:

System Verilog and UVM

System Verilog data types, OOP features (class, inheritance, polymorphism). Interfaces, modports, clocking blocks. Assertions: immediate and concurrent, sequences and properties. UVM methodology: UVM testbench architecture, agents, sequencer, driver, monitor, scoreboard. Factory pattern, configuration database. UVM sequences and sequence items. UVM phasing.

Labs / Practicals:

1. Fault simulation on a combinational circuit using a simulation tool.
2. Stuck-at fault ATPG on a benchmark circuit.
3. Scan chain insertion in Verilog design and simulation.
4. LBIST circuit design and fault coverage estimation.
5. JTAG boundary scan implementation.

6. UVM testbench for a simple ALU.
7. SVA: write and check assertions on a counter.
8. Constrained random testbench for a FIFO.
9. Formal equivalence check of RTL vs. synthesized netlist.
10. Coverage-driven verification plan for an interface block.

References:

1. Wang L-T, Chang Y-W, Cheng K-T – Electronic Design Automation, Morgan Kaufmann.
2. Bhunia S. and Tehranipoor M. – The Hardware Trojan War, Springer.
3. Spear C. and Tumbush G. – SystemVerilog for Verification, Springer.
4. IEEE Std 1149.1-2013 (JTAG) Standard.

Course Code	Course Name	L	T	P	Cr
DOEE260013	EDA Tools and Design Automation	3	0	2	4

Course Outcomes:

CO1: Understand the role of EDA tools in the IC design flow.
CO2: Apply graph-theoretic and algorithmic methods to logic optimization and technology mapping.
CO3: Implement placement and routing algorithms for VLSI physical design.
CO4: Apply formal verification and equivalence checking methods.
CO5: Write Tcl scripts to automate EDA tool flows.

Module 1:

Introduction to EDA

EDA overview: history, market, tool categories. IC design flow and EDA touchpoints. Representation of logic: Boolean functions, BDDs (Binary Decision Diagrams), AIGs. Logic simulation: event-driven, cycle-accurate. Verilog and SPICE simulation engines. EDA data formats: Verilog, VHDL, SPICE, Liberty, LEF/DEF, GDS.

Module 2:

Logic Synthesis Algorithms

Two-level logic optimization: Quine-McCluskey, Espresso. Multi-level logic: factoring, decomposition, substitution. Sequential optimization: retiming. Technology mapping: tree covering, dynamic programming, FPGA mapping. Area and delay optimization algorithms. Power optimization in synthesis

Module 3:

Physical Design Algorithms

Partitioning: Kernighan-Lin, Fiduccia-Mattheyses algorithms. Placement: simulated annealing, quadratic placement, analytical placement. Global routing: Steiner tree routing, maze routing. Detailed routing: channel routing, Lee algorithm. Compaction. Clock network synthesis. Timing optimization in P&R.

Module 4:

Formal Verification Methods
Satisfiability (SAT) problem: DPLL algorithm, CDCL. SAT solvers: MiniSAT, PicoSAT. SAT applications in ATPG, equivalence checking. Model checking: CTL and LTL temporal logic. BDD-based model checking. Bounded model checking (BMC). Sequential equivalence checking (SEC). Property specification.

Module 5:

EDA Scripting and Custom Tools

Tcl scripting for EDA tools: Design Compiler, Innovus. Python for EDA automation: reading LEF/DEF, post-processing reports. Open-source EDA ecosystem: Yosys, OpenROAD, Magic, KLayout, ngspice. Machine learning in EDA: routability prediction, timing estimation, DRC hotspot detection (overview). Cloud-based EDA and CI/CD for chip design.

Labs / Practicals:

1. Two-level minimization using Espresso tool.
2. Multi-level logic synthesis using Yosys.
3. Technology mapping: FPGA LUT mapping with ABC tool.
4. Placement algorithm implementation in Python (force-directed).
5. Global routing visualization on a sample design.
6. SAT-based combinational equivalence checking (MiniSAT).

7. BDD construction for Boolean functions.
8. Tcl script for batch synthesis flow using Yosys.
9. Formal assertion checking using SymbiYosys.
10. OpenROAD flow: RTL to GDS of a small design (open-source).

References:

1. De Micheli G. – Synthesis and Optimization of Digital Circuits, McGraw-Hill.
2. Wang L-T, Chang Y-W, Cheng K-T – Electronic Design Automation, Morgan Kaufmann.
3. Lavagno L., Scheffer L., Martin G. – Electronic Design Automation for IC Implementation, CRC Press.
4. OpenROAD documentation: https://openroad.readthedocs.io

Course Code	Course Name	L	T	P	Cr
DOAI260001	AI (Industry)-Applied Ethics for AI	3	0	2	4

Course Outcomes:

CO1. Understand and be able to define and explain the terms 'AI for good' and 'responsible AI'. Describe the significance of responsible AI and recognize the potential benefits of AI in society.
CO2. Discuss and examine the impact of ethics on AI decision-making processes. Identify common ethical pitfalls in AI development and deployment. Explore methods for ethical AI practices.
CO3. Understand to apply an ethics-first approach to AI project planning and development. Compare different approaches to ethical AI design which include design of rules based on ethics and virtues. Identify factors influencing ethical decisions in AI development.
CO4. Apply Utilitarian and Kantian ethical principles to AI case studies, scenarios, and dilemmas. Explain the purpose and benefits of AI ethics codes. Develop a code of ethics for AI within an organization.
CO5. Identify the importance of interpretability in AI models. Evaluate the strengths and weaknesses of three different explainable AI tools: SHAP, LIME, and Tree Interpreter.
CO6. Conduct impact assessments for AI solutions. Apply ethical considerations in designing an AI system or solution. Implement measures to mitigate ethical risks and challenges post deployment.

Module 1:

Introduction to AI Ethics – meaning and importance, Role of Ethics in Responsible AI – defining 'AI for good' and 'responsible AI', significance of responsible AI in society, Common Ethical Pitfalls in AI – bias, fairness, accountability, transparency issues in AI development and deployment, Principles of Ethical AI – core principles guiding ethical AI, methods for ethical AI practices in development and deployment.

Module 2:

Ethics-First Approach in the AI Project Cycle – applying ethics at every stage of AI project planning and development, Approaches for Designing Ethical AI – rules-based ethics, values-based design, Introduction to Ethical Frameworks – overview of major ethical frameworks applied to AI, Virtue-Based Ethics – virtue ethics principles and the 4-box method for analysing ethical dilemmas, Applying Bio-Ethics Framework in AI – bioethical principles applied to AI in healthcare and biotechnology.

Module 3:

Utilitarianism for Ethical AI – utilitarian principles applied to AI case studies and dilemmas, Kantianism for Ethical AI – Kantian ethical principles applied to AI scenarios, Code of AI Ethics for Organizations – purpose, benefits and development of an organizational AI code of ethics, Data Protection and AI – risks of data breaches, privacy violations, responsible data handling.

Module 4:

Explainable AI – importance of interpretability in AI models, Intel XAI Toolkit, impact of explainability on decision-making and trust, Explainable AI in Practice – SHAP, LIME and Tree Interpreter tools: strengths, weaknesses and methods for applying them to analyze and interpret AI applications.

Module 5:

Impact Assessment of AI Solutions – conducting impact assessments for AI systems, Mitigating Ethical Issues Post Deployment – strategies to identify and resolve ethical risks after deployment, Project Creation – creating an AI startup with data card, model card, code of ethics, explainability report, impact self-assessment and transparency report.

Labs / Practicals:

1. Develop comprehensive Data Cards for datasets, as outlined in the Data Cards Playbook by Google, to ensure transparency and accountability in AI training data.
2. Design Responsible AI Best Practices.

3. Apply Human-Centered AI Canvas to design AI systems that address user needs and ethical considerations.
4. Learn to use The Box to embed AI principles into organizational workflows and operations.
5. Apply bio-ethics principles to AI case studies, scenarios, and dilemmas; evaluate the impact of virtue-based ethics on AI systems.
6. Develop AI ethics cards for given AI scenarios. Analyse a variety of ethical dilemmas by following the 4-box method from Virtue Ethics.
7. Analyse a variety of ethical dilemmas by walking through the ethical frameworks: Virtue Ethics, Bioethics, Utilitarian, Kantian, and Moral System Agnostic.
8. Develop an AI Code of Ethics for a fictitious company.
9. Use the UXAI Toolkit to brainstorm and plan different explainable interfaces for AI applications. Use the explainable AI library SHAP to create an explainability report for income prediction on the standard adult census income dataset.
10. Conduct impact assessments for AI solutions; create your own AI startup, develop business documents, write documentation (data card, model card, code of ethics, explainability report, impact self-assessment) and create and analyze transparency report.

References:

1. Intel AI for Future Workforce – Applied Ethics for AI Course Content, Intel Digital Readiness Program.
2. Virginia Eubanks – Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor, St. Martin's Press.
3. Cathy O'Neil – Weapons of Math Destruction, Crown Publishers.
4. Kate Crawford – Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence, Yale University Press.
5. Google Data Cards Playbook – https://pair-code.github.io/datacardsplaybook/

Course Code	Course Name	L	T	P	Cr
DOAI260002	AI (Industry)-Introduction to Artificial Intelligence	3	0	2	4

Course Outcomes:

CO1. Describe what AI is and what it is not. Appreciate the history of AI and its evolution over time.
CO2. Identify and analyze current trends in AI by correlating them with other technologies like IoT, Big Data and 5G.
CO3. Identify 3 common domains of AI – Natural Language Processing, Computer Vision and Statistical Data – based on the type of underlying data.
CO4. Examine and appreciate typical steps involved in an AI Project through the AI Project Cycle.
CO5. Examine the immediate impacts of AI practices on society and appreciate the ethical concerns of AI applications.
CO6. Classify Supervised Learning, Unsupervised Learning and Reinforcement Learning with the help of examples. Discuss the roles played by different key stakeholders in a typical AI team.

Module 1:

Demystification of AI – what AI is and what it is not, History and Evolution of AI, Current trends in AI, AI and Technology convergence, Industry 4.0 and Digitalization, Role of IoT, Big Data and 5G in AI, Domains of AI – Natural Language Processing, Computer Vision and Statistical Data.

Module 2:

AI Project Cycle-I – Problem Scoping and Data Acquisition, AI Project Cycle-II – Modelling and Evaluation, Societal Impact of AI – ethical concerns and responsible practices, Elements of ML – Supervised Learning, Unsupervised Learning and Reinforcement Learning with examples.

Module 3:

Elements of AI – types and characteristics, Data as Fuel for AI – structured and unstructured data, methods of data mining and data storage, AI/Data Science Team – roles and responsibilities of key stakeholders.

Module 4:

Tools to implement AI – No-Code tools (Teachable Machine, Orange, Roboflow, Chatfuel) and Code-based tools, Introduction to Neural Networks – working of simple Neural Networks, difference between common Deep Learning models with examples and applications.

Module 5:

AI Project-I – Natural Language Processing use case using no-code tool Chatfuel, AI Project-II – Statistical Data use case using no-code tool Orange Data Mining, AI Project-III – Computer Vision use case using no-code tool Roboflow, Future Possibilities of AI – emerging software and hardware technology trends and their direct impact on development of AI.

Labs / Practicals:

1. Familiarize with AI tools and techniques, including statistical data analysis using raw graphs, computer vision applications like Vision API, and generative AI tools such as DALL·E, Stable Diffusion, Gemini, and ChatGPT.
2. Apply the AI Project Cycle to real-world scenarios by identifying problems, collecting data, developing models, and evaluating their effectiveness to propose impactful AI-based solutions.
3. Deconstruct the working behind simple Neural Networks and outline the difference between some common DL models with the help of examples and applications.
4. Explore and utilize no-code tools, such as Teachable Machine, to build AI projects.
5. Project Building using No-code tool – Orange Data Mining for Statistical Data Use cases.
6. Project Building using No-code tool – Roboflow for Computer Vision Use cases.
7. Project Building using No-code tool – Chatfuel for Natural Language Processing Use cases.
8. Classify the type of data into structured and unstructured and evaluate data quality using real datasets.

9. Explore various AI domains by running Vision API experiments and comparing outcomes from NLP, CV, and Statistical Data use cases.
10. Demonstrate the AI Project Cycle on a chosen real-world problem from an industrial or social impact perspective and present findings.

References:

1. Intel AI for Future Workforce – Introduction to AI Course Content, Intel Digital Readiness Program.
2. Stuart Russell and Peter Norvig – Artificial Intelligence: A Modern Approach, Pearson Education.
3. Andreas Mueller and Sarah Guido – Introduction to Machine Learning with Python, O'Reilly Media.
4. Ian Goodfellow, Yoshua Bengio, and Aaron Courville – Deep Learning, MIT Press.
5. Aurélien Géron – Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media.

Course Code	Course Name	L	T	P	Cr
DOAI260003	AI (Industry)-Introduction to Generative AI	3	0	2	4

Course Outcomes:

CO1. Define Gen AI and distinguish its role in the broader context of AI. Describe how the basic model of Gen AI works. Discuss the wider societal impact of Gen AI.
CO2. Understand how NLP has evolved over time and the shift from Traditional AI to Gen AI. Understand Prompt Engineering and its impact on the output of NLP tasks.
CO3. Learn the limitations of current AI generation tools for video and audio. Recognize the challenges in ethical considerations and responsibilities when using generative AI tools.
CO4. Describe the stages of a generative AI project and learn to select and use appropriate AI tools for development.
CO5. Understand the basic structure of neural networks, architecture of transformer models, workings of diffusion models, and the pre-training process for generative AI models.
CO6. Learn to fine-tune generative AI models, understand RLHF, prototype Gen AI applications, and apply principles of responsible and ethical use of LLMs.

Module 1:

Introduction to Gen AI – definition and scope, role of Gen AI in the broader context of AI, how the basic model of Gen AI works, societal impact of Gen AI, Demystifying Gen AI Models – types of generative models (GANs, VAEs, Diffusion Models, LLMs), evolution of NLP to Gen AI, shift from Traditional AI to Gen AI.

Module 2:

Prompt Engineering-I – introduction to prompt engineering, basic applications and impact on NLP task outputs, text generation and conversation initiation, Prompt Engineering-II – advanced prompt engineering techniques for diverse NLP applications, real-world examples such as Mint Mobile Commercial, Basics of Video and Audio for Generation using Gen AI for Productivity-I – introduction to video generation tools (Runway ML, Genmo), text-to-video and image-to-video generation, limitations of current AI generation tools for video and audio.

Module 3:

Basics of Video and Audio for Generation using Gen AI for Productivity-II – audio and music generation using AIVA, code generation using Codeium, text generation using ChatGPT, Gemini and Perplexity, Generative AI-Ethics – ethical considerations, responsibilities and challenges when using generative AI tools, Generative AI Project Lifecycle – stages of a Gen AI project, selection of appropriate AI tools, Gen AI Tools for Development – open-source tools such as Stable Diffusion XL, proprietary tools such as Bing Image Creator powered by DALLE3.

Module 4:

Neural Network Fundamentals – basic structure and functioning of neural networks, how neural networks process data to make predictions, Transformers and the Evaluation of LLMs – architecture of transformer models, methods to evaluate large language models for performance and reliability, walkthrough of OpenAI Playground and parameter exploration, Diffusion Models and the Evaluation of Text to Image Generation – workings of diffusion models, evaluation of quality of text-to-image generation, Pre-Training Gen AI Models-I – pre-training process for generative AI models, importance of pre-training in building effective AI systems, modifying classification labels of Cohere datasets and its impact on fine-tuning and model confidence.

Module 5:

Fine-Tuning Gen AI Models – techniques for fine-tuning generative AI models to adapt them for specific tasks and applications, RLHF – Evaluating and Optimizing Gen AI Models: reinforcement learning with human feedback, evaluation and improvement of performance of generative AI models, Prototyping Gen AI Applications – skills to design, develop and test prototypes of generative AI applications, LangChain components – LLM, Agents, Prompts, Chains, Document loaders and utils with code-based implementation, LLM Application and Responsible use of AI – practical applications of large language models, ethical and responsible use of LLMs, building an AI

Tutor using open-source models such as Dolly 3B and Intel Neural Chat 7B.

Labs / Practicals:

1. Apply prompt engineering skills in basic NLP tasks, such as text generation, conversation initiation, and simple problem-solving.
2. Develop advanced prompt engineering skills for diverse NLP applications. Examine real-world prompt engineering examples, such as the Mint Mobile Commercial, to discern its role in marketing.
3. Explore cutting-edge Gen AI tools, their features, and capabilities for content generation including video and audio.
4. Explore open-source models such as Stable Diffusion XL and proprietary tools such as Bing Image Creator (powered by DALLE3) for image generation.
5. Text-to-video generation and image-to-video generation using tools like Runway ML and Genmo. Generate audio and music files using the tool AIVA.
6. Generate code by prompts using the tool Codeium. Generate text using tools like ChatGPT, Gemini, and Perplexity.
7. Walkthrough of OpenAI Playground. Explore how changing parameters can affect responses generated by different models.
8. Explore how modifying classification labels of Cohere datasets impacts fine-tuning and model confidence.
9. Explore basic components of LangChain like LLM, Agents, Prompts, Chains, and Document loaders and utils, along with a code-based implementation.
10. Utilize open-source text generation models such as Dolly 3B and Intel Neural Chat 7B for building an AI Tutor application.

References:

1. Intel AI for Future Workforce – Introduction to Generative AI Course Content, Intel Digital Readiness Program.
2. Antony Hagiwara – Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play, O'Reilly Media.
3. Ashish Vaswani et al. – Attention is All You Need, Advances in Neural Information Processing Systems.
4. OpenAI – GPT-4 Technical Report, OpenAI.
5. Harrison Chase – LangChain Documentation – https://docs.langchain.com/

Course Code	Course Name	L	T	P	Cr
DOAI260004	AI (Industry)-Introduction to Machine Learning	3	0	2	4

Course Outcomes:

CO1. Describe Machine Learning and how Deep Learning is a subset of the wider field of Machine Learning.
CO2. Understand the importance of dashboards and discuss various applications of dashboards across different industries.
CO3. Explain the mathematical working behind different ML models. Implement different classification and regression ML models using Python.
CO4. Outline the difference between supervised, unsupervised and reinforcement learning algorithms. Examine the working of different reinforcement learning models.
CO5. Describe the working of Neural Networks and contrast it with biological Neurons. Describe various applications of neural networks. Outline different methods to overcome variance and bias in DL models.
CO6. Discuss and interpret the future of ML based on current and upcoming trends. Develop Python-based AI Projects incorporating Supervised Learning, Unsupervised Learning and Deep Neural Networks.

Module 1:

Introduction to Machine Learning – definition, types and applications, Relationship between AI, ML and Deep Learning, Tools to Implement AI (Python & Mathematics) – I: Python basics for ML, NumPy, Pandas, data manipulation.

Module 2:

Tools to Implement AI (Python & Mathematics) – II: mathematical foundations – linear algebra, statistics and probability, No-Code tool for Data Exploration – Tableau dashboard creation, data visualization techniques, statistical concepts implementation using no-code visualization tools.

Module 3:

AI (Modeling) Supervised Learning-I – Linear Regression, Logistic Regression, mathematical working behind models, classification and regression using Python, AI (Modeling) Supervised Learning-II – SVM, Decision Tree, industrial applications of ML models, AI (Modeling) Unsupervised Learning-I – K-Means Clustering, mathematics of clustering algorithms, AI (Modeling) Unsupervised Learning-II – Recommendation Systems, mathematics behind recommendation system, applications of recommendation systems.

Module 4:

Reinforcement Learning – definition, difference from supervised and unsupervised learning, working of reinforcement learning models with applications, Neural Network and Deep Learning-I – biological vs artificial Neurons, working of Neural Networks, applications of neural networks, Deep Learning-II – Convolutional Neural Networks, Recurrent Neural Networks, methods to overcome variance and bias in DL models, implementation using Python and deep learning frameworks, exploring OpenAI's Gym for reinforcement learning.

Module 5:

ML Project – Supervised Learning AI models: end-to-end Python project using classification/regression, ML Project – Unsupervised Learning AI models: end-to-end Python project using clustering, ML Project – Deep Neural Networks: building and training DNN in Python, Future of Machine Learning and Deep Learning – current and upcoming trends in ML and DL.

Labs / Practicals:

1. Interpret and understand the basic tools required for building ML Projects through a selected list of topics from Mathematics and Python Programming.
2. Develop and create a simple dashboard for visualizing data through Tableau.
3. Implement statistical concepts using the no-code visualization tool and create your own dashboard with the help of a no-code visualization tool.

4. Implement supervised machine learning algorithms, such as SVM and Decision Tree, using Python to develop a deeper understanding of their practical applications.
5. Implement unsupervised machine learning algorithms, such as K-Means, using Python to develop a deeper understanding of their practical applications.
6. Implement neural networks and create simple generative AI models using Python and deep learning frameworks, including exploring OpenAI's Gym for hands-on experience with AI and reinforcement learning.
7. Develop Python-based use cases and AI Projects incorporating Supervised Learning methods.
8. Develop Python-based use cases and AI Projects incorporating Unsupervised Learning methods.
9. Develop Python-based use cases and AI Projects incorporating Deep Neural Networks.
10. Implement a recommendation system using Python and evaluate its performance on a real-world dataset.

References:

1. Intel AI for Future Workforce – Introduction to Machine Learning Course Content, Intel Digital Readiness Program.
2. Andreas Mueller and Sarah Guido – Introduction to Machine Learning with Python, O'Reilly Media.
3. Ian Goodfellow, Yoshua Bengio, and Aaron Courville – Deep Learning, MIT Press.
4. Aurélien Géron – Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media.
5. Christopher Bishop – Pattern Recognition and Machine Learning, Springer.

Course Code	Course Name	L	T	P	Cr
DCSA250056	Numerical Methods	3	1	0	4

Course Outcomes:

CO1: Solve algebraic and transcendental equations using appropriate numerical methods.
CO2: Apply interpolation techniques to estimate values for given discrete data.
CO3: Use numerical techniques to solve systems of linear equations efficiently.
CO4: Compute derivatives and definite integrals using numerical differentiation and integration methods.
CO5: Solve ordinary differential equations using numerical approaches and analyze the accuracy of solutions.

Module 1:

Solution of Equations and Interpolation:
Bisection method, Regula-Falsi method, Newton-Raphson method, Forward & Backward differences, Newton's & Gauss interpolation, Lagrange interpolation, Divided difference method

Module 2:

Numerical Differentiation and Integration:
Numerical differentiation using finite differences
Numerical integration using: Trapezoidal rule, Simpson's 1/3 rule, Simpson's 3/8 rule

Module 3:

Solution of Ordinary Differential Equations (ODEs):
Taylor series method, Euler's method, Modified Euler's method, Runge-Kutta methods (2nd and 4th order)

Module 4:

Predictor-Corrector Methods:
Milne's predictor-corrector method, Adams-Bashforth and Adams-Moulton methods

Module 5:

Numerical Solution of Partial Differential Equations (PDEs):
Finite difference method for Laplace and Poisson equations, Heat equation (Explicit and Crank-Nicholson methods), Wave equation using explicit method

Labs / Practicals:

N/A

References:

1. P. Kandasamy, K. Thilagavathy & K. Gunavathi, Numerical Methods (S. Chand, 2nd Ed, 2012)
2. S. S. Sastry, Introductory Methods of Numerical Analysis (PHI, 4th Ed, 2005)
3. E. Kreyszig, Advanced Engineering Mathematics (Wiley, 9th Ed, 2006)
4. B. S. Grewal, Higher Engineering Mathematics (Khanna, 35th Ed, 2010

Course Code	Course Name	L	T	P	Cr
DCSE250020	Cloud Computing	3	0	2	4

Course Outcomes:

CO1: Explain the evolution, architecture, and key characteristics of cloud computing, including service models (IaaS, PaaS, SaaS) and deployment types (public, private, hybrid).

CO2: Describe various types and implementation levels of virtualization, including CPU, memory, storage, and network virtualization in cloud environments.

CO3: Analyze layered cloud architecture and address design challenges related to inter-cloud resource management and platform deployment.

CO4: Apply distributed and parallel programming paradigms such as MapReduce and Hadoop for cloud-based application development and understand various cloud platforms like AWS, OpenStack, and Google App Engine.

CO5: Evaluate cloud security challenges and solutions, including data and application security, VM security, and risk management strategies in cloud environments.

Module 1:

Introduction to Cloud Computing

Evolution of Cloud Computing, System Models for Distributed and Cloud Computing, NIST Cloud Computing Reference Architecture, Features of Cloud Computing, Cloud Services – IaaS, PaaS, SaaS, Cloud service Providers – Public , Private and Hybrid Clouds.

Module 2:

Introduction to component virtualization

Basics of Virtualization, Types of Virtualization, Implementation Levels of Virtualization, Virtualization of CPU, Memory, I/O Devices, Desktop Virtualization, Server Virtualization, Storage Virtualization, Network Virtualization.

Module 3:

Architectural Design of Compute and Storage Clouds

Layered Cloud Architecture Development, Design Challenges, Inter Cloud Resource Management, Resource Provisioning and Platform Deployment, Global Exchange of Cloud Resources.

Module 4:

Parallel and Distributed Programming Paradigms

Map Reduce, Twister and Iterative MapReduce, Hadoop Library from Apache, Mapping Applications, Programming Support, Google App Engine, Amazon AWS, Cloud Software Environments - Eucalyptus, Open Nebula, OpenStack.

Module 5:

Security Overview

Cloud Security Challenges, Software-as-a-Service Security, Security Governance, Risk Management, Security Monitoring, Security Architecture Design, Data Security, Application Security, Virtual Machine Security.

Labs / Practicals:

1. Exploring Cloud Deployment and Service Models
2. Simulating Public, Private, and Hybrid Clouds
3. Virtualization of CPU and Memory
4. Network and Storage Virtualization
5. Deployment of a Layered Cloud Architecture
6. Resource Provisioning and Scaling
7. Deploying a Web Application
8. Simulating Security Attacks and Defenses in a Cloud VM
9. Implementing Access Control and Encryption on Cloud Storage

References:

[1] R. Buyya, C. Vecchiola, and T. Selvi, Mastering Cloud Computing: Foundations and Applications Programming. New Delhi, India: McGraw-Hill Education, 2013.
[2] A. T. Velte, T. J. Velte, and R. Elsenpeter, Cloud Computing: A Practical Approach. New Delhi, India: McGraw-Hill Education, 2010.
[3] K. Hwang, G. C. Fox, and J. J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. Burlington, MA, USA: Morgan Kaufmann, 2012.
[4] T. Erl, R. Puttini, and Z. Mahmood, Cloud Computing: Concepts, Technology & Architecture. Boston, MA, USA: Pearson Education, 2013.
[5] G. Reese, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. Sebastopol, CA, USA: O'Reilly Media, 2009.

Course Code	Course Name	L	T	P	Cr
DOAI250005	Artificial Intelligence and Machine Learning	2	0	4	4

Course Outcomes:

CO1: Understand AI problem-solving techniques, heuristic search algorithms, and reinforcement learning.
CO2: Apply knowledge representation, logic-based reasoning, and probabilistic models in AI systems.
CO3: Develop and evaluate machine learning models using supervised and unsupervised learning techniques.
CO4: Design and implement deep learning architectures for vision, language, and sequential data applications.
CO5: Analyze advanced AI methods, ethical challenges, and emerging trends in artificial intelligence.

Module 1:

AI Techniques and Search Strategies
History of AI, problem-solving using search, uninformed vs. informed search, heuristic search techniques (Hill Climbing, Simulated Annealing, A* Algorithm), adversarial search (Minimax Algorithm, Alpha-Beta Pruning), AI-driven game strategies (Chess, Chinese Checkers), Reinforcement Learning fundamentals (Markov Decision Processes, Q-learning), heuristic optimization techniques (Genetic Algorithms, Particle Swarm Optimization).

Module 2:

Knowledge Representation, Reasoning, and Probabilistic Models
Propositional logic, inference mechanisms, forward and backward chaining, probabilistic reasoning techniques (Bayesian Networks, Markov Models, Probabilistic Graphical Models), logic-based AI vs. machine learning-driven reasoning, Natural Language Processing (NLP) fundamentals (tokenization, embeddings, language models), AI-driven text processing, machine translation.

Module 3:

Machine Learning Fundamentals and Neural Networks
Supervised and unsupervised learning paradigms, classification and regression techniques (decision trees, support vector machines, ensemble learning methods: Random Forest, XGBoost), artificial neural networks (activation functions, loss functions, back propagation, multilayer perceptions), overfitting, bias-variance tradeoff, hyperparameter tuning, performance evaluation metrics (precision-recall, ROC curves, F1-score).

Module 4:

Deep Learning and Advanced Neural Architectures
Convolutional Neural Networks (CNNs) for image classification and object detection (convolution layers, pooling, fully connected layers), Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks for sequential data processing, Transformer models (BERT, GPT) for NLP applications, optimization techniques (batch normalization, dropout regularization, transfer learning), deep learning applications (medical imaging, self-driving vehicles, generative AI).

Module 5:

Unsupervised Learning, Generative Models, and Ethical AI
Clustering techniques (K-Means, Hierarchical Clustering), Autoencoders, Generative Adversarial Networks (GANs), self-supervised learning, foundation models in AI, AI applications in healthcare, finance, and autonomous systems, AI ethics (fairness, bias mitigation, adversarial attacks, explainable AI), advancements in AI (quantum computing, federated learning, AI governance frameworks).

Labs / Practicals:

1. Implement a basic uninformed search algorithm (e.g., Breadth-First Search, Depth-First Search).
2. Implement informed search algorithms using heuristics (e.g., A* Algorithm for pathfinding).
3. Implement the Hill Climbing algorithm for local search problems.
4. Develop a solution using Simulated Annealing for optimization problems.
5. Implement the Minimax algorithm to solve a simple game (e.g., Tic-Tac-Toe).
6. Implement Alpha-Beta Pruning for optimizing the Minimax algorithm.
7. Build a Reinforcement Learning agent using Q-learning to solve a basic maze problem.
8. Implement Genetic Algorithms to solve optimization problems (e.g., travelling salesman).
9. Build a Particle Swarm Optimization model for function optimization.
10. Implement propositional logic and inference mechanisms (e.g., forward and backward chaining).
11. Develop a basic Bayesian Network for probabilistic reasoning.

References:

1.Tom M. Mitchell: Machine Learning. McGraw Hill
2.Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig
3.Pattern Recognition and Machine Learning by Christopher M. Bishop
4.Machine Learning: A Probabilistic Perspective by Kevin P. Murphy
5.Reinforcement Learning: An Introduction by Richard S. Sutton and Andrew G. Barto
6.Natural Language Processing with Python by Steven Bird, Ewan Klein, and Edward Loper
7.The Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman
8.Michael Nielsen: Neural Networks and Deep Learning (free online book)
9.Stuart Russel & Peter Norvig: Artificial Intelligence– A Modern Approach. Pearson, 4th edition, 2023
10.Deep Learning (Adaptive Computation and Machine Learning series) Illustrated Edition, by Ian Goodfellow, Yoshua Bengio, Aaron Courville, MIT Press, London

Course Code	Course Name	L	T	P	Cr
DOAI250007	Fundamentals of Artificial Intelligence	2	0	4	4

Course Outcomes:

CO1. Understand the fundamentals of Artificial Intelligence and review Python programming concepts.
CO2. Apply AI search algorithms such as uninformed, informed, heuristic, and adversarial search.
CO3. Demonstrate knowledge of various machine learning techniques and implement relevant algorithms.
CO4. Employ text processing and NLP techniques in real-world tasks.
CO5. Build basic AI applications for domains such as education, healthcare, and automation.

Module 1:

Fundamentals of AI & Python Basics
Introduction to AI, Evolution & Revolution of AI, Structure of AI, Intelligent Agents & Environments.

Python: Introduction, Basic Syntax, Data Types, Variables, Operators, Input/output, Flow of Control (If, If-else, Nested if-else), Loops (For, While, Nested loops, Pass), Strings, List, Dictionary & Tuples Comparison, OOPs Concepts.

Module 2:

Problem Solving by Search:
Searching for solution, Uninformed Search, Breadth First Search, Informed Search (Heuristic & Meta-heuristics), Alpha-beta pruning.

Module 3:

Learning: The Brain and the Neuron, Machine Learning types (Supervised, Unsupervised, Reinforcement), Find S Algorithm, Classification, Linear Regression.

Module 4:

Natural Language Processing
Basics of Text Processing, Tokenization, Word Types, Morphology, Lemmatization, Morphemes, Stemming, POS Tagging.

Module 5:

Applications of Artificial Intelligence
Applications in Education, Healthcare, Transportation, Robotics, Data Analysis, Autonomous Vehicles, Agriculture, Gaming.

Labs / Practicals:

1. Familiarization with Python libraries used in machine learning
2. Write a Python program to check if a number is even or odd
3. Generate random numbers using Python
4. Extract data from a database using Python
5. Implement Breadth First Search using Python
6. Solve a simple knapsack problem using heuristic algorithm
7. Implement Alpha-beta pruning game-playing algorithm
8. Implement Find S algorithm using Python

9. Perform K-means clustering on synthetic GPS data
10. Implement regression model (y = mx + c) using dataset
11. Remove stop words from text using Python
12. Perform tokenization on input text
13. Implement noise removal in text using Python
14. Generate an image from text using an AI image generator
15. Create a speedometer using TK dial in Python
16. Analyze smartwatch data using Python

References:

Textbooks:
1. Stuart J. Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2010.
2. Rupinder Singh, Introduction to Artificial Intelligence, 1st Edition, Notion Press, 2021.
3. T.R. Sharika and P.S. Archana, Introduction to Artificial Intelligence, 1st Edition, Sharika T.R. Publisher, 2023.

Course Code	Course Name	L	T	P	Cr
DOAI250013	Machine Learning Fundamentals	2	0	4	4

Course Outcomes:

CO1. Understand and differentiate types of learning in machine learning.
CO2. Apply simple regression and classification techniques to practical problems.
CO3. Analyze and cluster data using unsupervised learning models.
CO4. Understand and implement Naive Bayes and basic sequence models.
CO5. Describe the structure of a neural network and use simple deep learning ideas.
CO6. Use basic ML libraries such as Scikit-learn for implementation.

Module 1:

Introduction to Machine Learning:
Definition of ML, Applications and Use Cases, Types of Machine Learning (Supervised, Unsupervised, Reinforcement), Basic Learning Concepts, Concept and Hypothesis Space, Introduction to Learning Models

Module 2:

Supervised Learning Methods:
Linear Regression, Overfitting and Underfitting, Cross Validation, Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, Random Forest, Model Evaluation Metrics (Accuracy, Precision, Recall)

Module 3:

Unsupervised Learning Methods:
Clustering Algorithms: K-Means and Hierarchical Clustering, Dimensionality Reduction using PCA, Basics of Recommendation Systems

Module 4:

Probability-Based Models:
Naïve Bayes Classifier, Maximum Likelihood, Basics of Bayesian Networks, Introduction to Hidden Markov Models (HMM) and Sequence Modeling

Module 5:

Introduction to Neural Networks & Deep Learning:
Perceptron, Feedforward Networks, Activation Functions, Concept of Backpropagation, Limitations of Classical ML, Introduction to Deep Learning and Use Cases

Labs / Practicals:

1. Introduction to Python libraries for ML (NumPy, Pandas, Matplotlib, Scikit-learn).
2. Build and test a Linear Regression model for house price prediction.
3. Apply Logistic Regression to binary classification (e.g., email spam detection).
4. Implement and compare K-Nearest Neighbors (KNN) with different k-values.
5. Train and evaluate a Decision Tree Classifier on a basic dataset.
6. Implement K-Means clustering on a synthetic dataset and visualize the clusters.
7. Apply Principal Component Analysis (PCA) on an image dataset for dimensionality reduction.
8. Use Naive Bayes Classifier for text classification (e.g., sentiment analysis).

9. Introduction to training a simple neural network using TensorFlow/Keras.

References:

1. Aurelien Geron, Hands-On Machine Learning with Scikit-Learn and TensorFlow, O'Reilly, 2nd Ed.
2. Stephen Marsland, "Machine Learning: An Algorithmic Perspective", Chapman & Hall/CRC, 2nd Edition, 2014.

3. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press.

Online Resources
1. https://scikit-learn.org
2. https://www.kaggle.com/learn/intro-to-machine-learning

Course Code	Course Name	L	T	P	Cr
DOAI250029	Artificial Intelligence with Python	2	0	4	4

Course Outcomes:

CO1: Define AI concepts, its history, types, and distinguish AI from human intelligence.
CO2: Implement and evaluate machine learning models using Python for supervised and unsupervised tasks.
CO3: Develop basic deep learning models using neural networks and understand CNNs for image classification.
CO4: Apply NLP techniques such as text preprocessing, sentiment analysis, and language modeling.
CO5: Demonstrate fundamental computer vision tasks using OpenCV for image processing and analysis.
CO6: Able to design and implement various machine learning algorithms in a range of real-world applications.

Module 1:

Introduction to Artificial Intelligence
•Definition and Goals of AI
•History and Applications of AI
•Types of AI (Narrow, General, Super)
•Intelligent Agents and Environments
•AI vs Human Intelligence
•Python for AI: Libraries Overview (NumPy, Pandas, Matplotlib, Scikit-learn)

Module 2:

Machine Learning
•Overview of Machine Learning and Types (Supervised, Unsupervised, Reinforcement)
•Data Preprocessing and Visualization
•Supervised Learning Algorithms: Linear Regression, k-NN, Decision Trees
•Unsupervised Learning: k-Means Clustering
•Model Evaluation: Accuracy, Confusion Matrix, Cross-validation

Module 3:

Introduction to Deep Learning
•Introduction to Neural Networks (Perceptrons, FeedForward Networks, Gradient Descent).
•Basic Concepts of Deep Learning (Layers, Activation Function)
•Introduction to Convolutional Neural Networks (CNNs) for Image classification.
•Simple Neural Network Implementation in Python

Module 4:

Natural Language Processing and its Applications
•Introduction to NLP
•Text Preprocessing (Tokenization, Stop Words, Lemmatization)
•Bag of Words and TF-IDF
•Sentiment Analysis using Python
•Real-world AI Applications: Chatbots, Image Recognition, Recommendation Systems

Module 5:

Computer Vision and its Applications
•Introduction to Computer Vision
•Image Processing Basics (Image Presentation, Pixels, Color Models).
•Image Filtering (Blurring, Sharpening, Edge Detection (Sobel, Canny)
•OpenCV Library for Computer Vision in Python

Labs / Practicals:

List of 10 Practical Exercises

1.Using a simple Linear Regression Model (Scikit-Learn) predict house prices based on a single feature (eg. Square Footage) using a synthetic or small real-world dataset.

2.Using KNN Model (Scikit-Learn) classify the iris dataset. Evaluate its accuracy using a train-test, split and visualize the decision boundaries (if possible, for 2 features).

3.Apply the K-Means clustering algorithm to a given dataset (eg Customer segmentation data). Determine an appropriate number of cluster using the elbow method.

4.Build a simple Feed-Forward neural network (using Tensorflow or Keras) with atleast one hidden layer to classify the MNIST handwritten digit dataset. Train the model and report its accuracy and loss.

5.Given a collection of raw text document. Perform tokenization, lowercasing and stop word removed. Implement stemming or lemmatization. Calculate word frequencies and display the top 10 most frequent word (use NLTK Library).

6.Develop a basic sentiment analysis model using NLTK and Scikit-Learn. Train on a small dataset.

7.Load an image and perform

a. Convert to Gray Scale.

b. Apply a Gaussian blur filter.

c. Detect edges using the Sobel and Canny edge detector.

d. Display the original and processed image.

8.Using a simple CNN model to perform an image classification on a dataset.

9.Using Decision Tree Model, classify the MNIST digit dataset. Evaluate the model.

10.Using logistic regression model, classify the iris dataset and evaluate the model.

References:

1."Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig

2."Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili

3."Deep Learning with Python" by François Chollet

4.Artificial Intelligence, Author: Saroj Kaushik
Publisher: Cengage Learning India

5.Programming in Python, Author: T. R. Padmanabhan
Publisher: Universities Press

6.Machine Learning, Author: V.K. Jain
Publisher: Khanna Book Publishing

7.Artificial Intelligence with Python, Author: Prateek Joshi (Indian-origin, Packt Publishing)

8.Introduction to Machine Learning, Author: Alok Sharma
Publisher: Katson Publishing House

9."Natural Language Processing with Python" by Steven Bird, Ewan Klein, and Edward Loper (NLTK Book).

10."Learning OpenCV: Computer Vision with the OpenCV Library" by Gary Bradski and Adrian Kaehler

Course Code	Course Name	L	T	P	Cr
DOAI250031	Introduction to High Performance Computing	3	1	0	4

Course Outcomes:

CO1: Understand instruction-level parallelism, pipelining techniques, hazards, and compiler/hardware methods for ILP exploitation.
CO2: Analyze multiprocessors, thread-level parallelism, vector architectures, GPU architectures, and CUDA programming concepts.
CO3: Evaluate cache performance, virtual memory, and advanced memory optimization techniques.
CO4: Apply parallel programming platforms (OpenMP, MPI, OpenCL, OpenACC) for performance improvement in AI/ML applications.
CO5: Explain interconnection networks, clusters, SoC interconnects, and NoC architectures for parallel systems.

Module 1:

Introduction to pipelining – Types of pipelining – Hazards in pipelining - Introduction to instruction level parallelism (ILP) – Challenges in ILP - Basic Compiler Techniques for exposing ILP - Reducing Branch costs with prediction - Overcoming Data hazards with Dynamic scheduling - Hardware-based speculation - Exploiting ILP using multiple issue and static scheduling - Exploiting ILP using dynamic scheduling, multiple issue and speculation - Tomasulo's approach, VLIW approach for multi-issue

Module 2:

Introduction to multi processors and thread level parallelism - Characteristics of application domain - Systematic shared memory architecture - Distributed shared – memory architecture – Synchronization
– Multithreading - Multithreading-fined grained and coarse grained, superscalar and super pipelining, hyper threading. Vector architectures; organizations and performance tuning; GPU architecture and internal organization, Elementary concepts in CUDA programming

Module 3:

Introduction to cache performance - Cache Optimizations - Virtual memory - Advanced optimizations of Cache performance - Memory technology and optimizations - Protection: Virtual memory and virtual machines - multi-banked caches, critical word first, early restart approaches, hardware pre-fetching, write buffer merging.

Module 4:

Introduction to parallel computing platforms; (Open MP, MPI, Open CL, Open ACC) with performance improvement analysis done using real-life AI and ML applications

Module 5:

Introduction to inter connection networks and clusters - interconnection network media - practical issues in interconnecting networks- examples - clusters - designing a cluster – System on Chip (SoC) Interconnects – Network on Chip (NOC).

Labs / Practicals:

NA

References:

1. Wang, Endong, Qing Zhang, Bo Shen, Guangyong Zhang, Xiaowei Lu, Qing Wu, and Yajuan Wang. "High-performance computing on the Intel Xeon Phi." Springer 5, 2014.
2. Sanders, Jason, and Edward Kandrot. "CUDA by example: an introduction to general-purpose GPU programming", Addison-Wesley Professional, 2010.
3. Chandra, Rohit, Leo Dagum, David Kohr, Ramesh Menon, Dror Maydan, and Jeff McDonald. Parallel programming in Open MP. Morgan kaufmann, 2001.

4. Kaeli, David R., Perhaad Mistry, Dana Schaa, and Do

Course Code	Course Name	L	T	P	Cr
DOEE250011	Analog Communication	2	1	2	4

Course Outcomes:

CO1: Explain the fundamentals of amplitude modulation, its types (AM, DSB-SC, SSB, VSB, QAM) and demodulation techniques with applications.
CO2: Analyze angle modulation techniques (FM, PM), their bandwidth requirements, generation and demodulation schemes.
CO3: Evaluate the effect of noise in analog communication systems and compare system performance (SNR, noise figure) under different modulation schemes.
CO4: Apply the principles of analog pulse modulation (PAM, PWM, PPM) and demodulation with spectral analysis.
CO5: Illustrate the working of radio receivers (AM, FM, superheterodyne) and associated circuits (PLL, Costas loop, FDM) with performance metrics like sensitivity and selectivity.ls.

Module 1:

Basics of Amplitude Modulation: Elements of communication systems, Information, Messages and Signals, Modulation, Modulation Methods, Modulation Benefits and Applications. Amplitude Modulation & Demodulation: Baseband and carrier communication, Amplitude Modulation (AM), Rectifier detector, Envelope detector, Double sideband suppressed carrier (DSB-SC) modulation & its demodulation, Switching modulators, Ring modulator, Balanced modulator, Frequency mixer, sideband and carrier power of AM, Generation of AM signals, Quadrature amplitude modulation (QAM), Single sideband (SSB) transmission, Time domain representation of SSB signals & their demodulation schemes (with carrier, and suppressed carrier), Generation of SSB signals, Vestigial sideband (VSB) modulator & demodulator, Illustrative Problems.

Module 2:

Angle Modulation &Demodulation: Concept of instantaneous frequency, Generalized concept of angle modulation, Bandwidth of angle modulated waves – Narrow band frequency modulation (NBFM); and Wide band FM (WBFM), Phase modulation, Verification of Frequency modulation bandwidth relationship, Features of angle modulation, Generation of FM waves – Indirect method, Direct generation; Demodulation of FM, Band pass limiter, Practical frequency demodulators, Small error analysis,

Module 3:

Noise in Communication Systems: Types of noise, Time domain representation of narrowband noise, Filtered white noise, Quadrature representation of narrowband noise, Envelope of narrowband noise plus sine wave, Signal to noise ratio & probability of error, Noise equivalent bandwidth, Effective noise temperature, and Noise figure, Baseband systems with channel noise, Performance analysis (i.e. finding SNR expression) of AM, DSB-SC, SSB-SC, FM, PM in the presence of noise, Illustrative Problems.

Module 4:

Analog pulse modulation schemes: Pulse amplitude modulation – Natural sampling, flat top sampling and Pulse amplitude modulation (PAM) & demodulation, Pulse-Time Modulation – Pulse Duration and Pulse Position modulations, and
demodulation schemes, PPM spectral analysis, Illustrative

Module 5:

Radio Receivers : Pre-emphasis, & De-emphasis filters, FM receiver, FM Capture Effect,. Carrier Acquisition-phased locked loop (PLL), Costas loop, Frequency division multiplexing (FDM), and Super-heterodyne AM receiver, Sensitivity and selectivity , selection of IF ,Illustrative Problems.

Labs / Practicals:

Amplitude Modulation and Demodulation

Frequency Modulation and Demodulation

Characteristics of Mixer

Pre-emphasis and De-emphasis

Pulse Amplitude Modulation and Demodulation

Pulse Width Modulation and Demodulation

Pulse Position Modulation and Demodulation

Radio Receiver Measurements – Sensitivity, Selectivity, and Fidelity

Sampling Theorem – Verification

Time Division Multiplexing

References:

B. P. Lathi, “Modern Digital and Analog Communication Systems,” Oxford Univ.
press, 3rd Edition, 2006
Sham Shanmugam, “Digital and Analog Communication Systems”, WileyIndia
edition, 2006.
A. Bruce Carlson, & Paul B. Crilly, “Communication Systems – An Introduction to Signals & Noise in Electrical Communication”, McGraw-Hill International Edition,
5th Edition, 2010.
Simon Haykin, “Communication Systems”, Wiley-India edition, 3 rd edition, 2010.
Herbert Taub& Donald L Schilling, “Principles of Communication Systems”, Tata
McGraw-Hill, 3rd Edition, 2009.
George Kennedy and Bernard Davis, “Electronics & Communication System”, TMH, 2004.

Course Code	Course Name	L	T	P	Cr
DOEE250057	Digital Communication Systems	3	0	2	4

Course Outcomes:

CO1: Explain the concepts of random variables, random processes, and their statistical properties in communication systems.
CO2: Analyze the fundamentals of digital communication, line coding, pulse shaping, and performance measures.
CO3: Compare and evaluate digital modulation techniques such as ASK, FSK, PSK, DPSK, and QPSK.
CO4: Apply equalization and detection techniques for digital communication over band-limited and noisy channels.
CO5: Demonstrate understanding of information theory, entropy, mutual information, coding techniques, and channel capacity.

Module 1:

Random Variables and Random Process: Deterministic and Random Signal: Types of random variables, cumulative distribution function and probability density functions, Standard distributions: Gaussian, exponential, Rayleigh, uniform, Bernoulli, binominal, Poisson, discrete uniform and conditional distributions. Functions of one random variable: distribution, mean, variance, moments and characteristics functions.
Random Processes: Random processes, stationary processes, mean and covariance functions, periodicity, linear filtering of random processes, power spectral density, examples of random processes: white noise process and white noise sequence, Gaussian process, Poisson process, Markov process.

Module 2:

Digital communication and modulation basics: Band pass and Low pass signal, Introduction to Digital communication systems, Pulse code modulation
, differential pulse code modulation, delta modulation, adaptive delta modulation, PSD of Line Coding schemes, Pulse shaping, Scrambling, Eye diagram, Gram-Schmidt orthogonalization scheme.

Module 3:

Digital Modulation Techniques: Modulation and Demodulation of Digital modulation schemes-ASK, FSK, PSK, DPSK, QPSK and Constellation diagram. Introduction to M-ary communication.

Module 4:

Digital Communication Through Band Limited Channels and Digital Receiver: Characteristic and signal Design of band Limited Channels. Optimum Receiver for Channel with ISI and AWGN. Linear Equalization, Decision Equalization, Adaptive Equalization. Introduction of Multichannel and Multicarrier System. Optimum threshold detection, Concept of Matched Filters, BER analysis of BASK, BFSK, BPSK, Model of Spread Spectrum Digital Communication. Direct Sequence Spread Spectrum Signal (DS-SS), Frequency Hopped Spread Spectrum Signal (FH-SS).

Module 5:

Information theory and Coding: Discrete Source models – Memoryless and Stationary, Mutual Information, Self-Information, Conditional Information, Average Mutual Information, Entropy, Entropy of the block, Conditional Entropy, Information Measures for Analog Sources. Review of probability theory Entropy Mutual information Data compression Huffman coding Asymptotic equipartition property Universal source coding Channel capacity Differential entropy Block codes and Convolutional codes.

Labs / Practicals:

1 Design and Generation of random binary signals

2 Study of impairments of signals generated in experiment 1 on passing through a simulated channel by observing Eye Pattern.
3 Generation Unipolar NRZ, Polar NRZ, Unipolar RZ and Polar RZ line codes.
4 Generation Manchester and AMI line codes
5 Conversion of analogue signal into PCM format and its study
6 Design and implementation of Delta Modulator for analogue signals
7 Design, implementation and study of BASK Modulator and demodulator
8 .Design, implementation and study of BPSK Modulator and demodulator
9 Design, implementation and study of BFSK Modulator and demodulator
10 Design, implementation and study of multiplexer and de-multiplexer of digital signals using TDM.
11 study of spread spectrum signal

References:

1 John G. Proakis & Masoud Salehi, “Digital Communications”, 5th Edition, McGraw Hill
2 B.P. Lathi, “Modern Digital and Analog Communication Systems”, 4th Edition, Oxford University Press
3 H. Taub, D L Schilling, Gautam Saha, “Principles of Communication”, 4th Edition, McGraw Hill
4 Singh & Sapre, Analog & Digital Communication Systems, 3th Edition, McGraw Hill
5 John G. Proakis,"Communication Systems Engineering 2nd Edition, Pearson Education, 2015
6 (Schaum's Outline Series) H P HSU & D Mitra, “Analog and Digital Communications”, McGraw Hill 3rd Edition
7 Bernard Sklar, Digital Communications, Pearson Education
8 Simon Haykin, “Communication Systems”, 5th Edition, Wiley India

Course Code	Course Name	L	T	P	Cr
DOEE250124	IOT and its Applications	3	1	0	4

Course Outcomes:

CO1: Understand the basics of IoT.
CO2: Implement the state of the Architecture of an IoT.
CO3: Understand design methodology and hardware platforms involved in IoT.
CO4: Understand how to analyze and organize the data.
CO5: Compare IOT Applications in Industrial & realworld.

Module 1:

FUNDAMENTALS OF IoT- Evolution of Internet of Things, Enabling Technologies, M2M Communication, IoT World Forum (IoTWF) standardized architecture, Simplified IoT Architecture, Core IoT Functional Stack, Fog, Edge and Cloud in IoT, Functional blocks of an IoT ecosystem, Sensors, Actuators, Smart Objects and Connecting Smart Objects.

Module 2:

IoT PROTOCOLS- IoT Access Technologies: Physical and MAC layers, topology and Security of IEEE 802.15.4, 802.11ah and Lora WAN, Network Layer: IP versions, Constrained Nodes and Constrained Networks,6LoWPAN, Application Transport Methods: SCADA, Application Layer Protocols: CoAP and MQTT.

Module 3:

DESIGN AND DEVELOPMENT- Design Methodology, Embedded computing logic, Microcontroller, System on Chips, IoT system building blocks
IoT Platform overview: Overview of IoT supported Hardware platforms such as: Raspberry pi, Arduino Board details

Module 4:

DATA ANALYTICS AND SUPPORTING SERVICES:
Data Analytics: Introduction, Structured Versus Unstructured Data, Data in Motion versus Data at Rest, IoT Data Analytics Challenges, Data Acquiring, Organizing in IoT/M2M,
Supporting Services: Computing Using a Cloud Platform for IoT/M2M Applications/Services, Everything as a service and Cloud Service Models.

Module 5:

CASE STUDIES/INDUSTRIAL APPLICATIONS: IoT applications in home, infrastructures, buildings, security, Industries, Home appliances, other IoT electronic equipments, Industry 4.0 concepts.

Labs / Practicals:

NA

References:

1. The Internet of Things – Key applications and Protocols, Olivier Hersent, David Boswarthick, Omar Elloumi and Wiley, 2012 (for Unit2).
2. “From Machine-to-Machine to the Internet of Things – Introduction to a New Age of Intelligence”,Jan Ho¨ ller, VlasiosTsiatsis, Catherine Mulligan, Stamatis,

Karnouskos, Stefan Avesand. David Boyle and Elsevier, 2014.
3. Architecting the Internet of Things,Dieter Uckelmann, Mark Harrison, Michahelles and Florian (Eds), Springer,2011.
4. Recipes to Begin, Expand, and Enhance Your Projects, 2nd Edition,Michael Margolis,Arduino Cookbook and O"Reilly Media,2011.

Course Code	Course Name	L	T	P	Cr
DOEE250128	Linear Electrical Networks	2	2	0	4

Course Outcomes:

CO1: Understand the principles of linear circuit analysis including Ohm's Law, Kirchhoff's Laws, and network theorems.
CO2: Analyze and solve linear electrical networks consisting of resistors, capacitors, inductors, and independent sources.
CO3: Gain proficiency in analyzing AC circuits using phasor techniques, impedance, and admittance concepts.
CO4: Explore the behavior of linear networks under sinusoidal steady-state conditions and transient responses.
CO5: Develop problem-solving skills for analyzing and designing linear electrical networks for various engineering applications.

Module 1:

Basic concept of circuit theory & Network Theorems: Review of circuit analysis using Kirchoff's laws, nodal and mesh analysis, solution by classical method and
Laplace transform, concept of independent and dependent sources, analysis of special signal waveforms, and duality of networks, Brief review of Signals and Systems. Superposition and Reciprocity theorem, Thevenin's and Norton's theorem, Millman's theorem, maximum power transfer theorem, compensation, Tellegan's theorem, analysis of circuits using theorems.

Module 2:

Transient Analysis of Networks: Network elements, Transient response of R-L, R-C, R-L- C for DC and sinusoidal excitation, Initial
condition, Solution using differential equation approach and Laplace transform method. Behaviors of series and parallel resonant circuits

Module 3:

RLC circuit and Resonance: Laplace transforms and properties: Partial fraction, singularity functions, waveform synthesis, analysis of RC, RL, and RLC networks with and without initial conditions with Laplace. Resonance conditions, quality factor.

Module 4:

Two Port Networks and Graph theory: Two Port Networks, Graph theoretic analysis for Large scale networks, Formulation and solution of network graph of simple networks, State space representation, Analysis using NGSPICE.

Module 5:

Passive Filter Design: Butter worth and Chebyshev approximations, Normalized specifications, Frequency transformations,
Frequency and impedancede-normalisation, Types of frequency selective filters, Linear phase filters.

Labs / Practicals:

NA

References:

"Network and systems" by D.Roy - Choudhary
"Circuit Analysis - with computer applications to problem solving" by

Someshwar C. Gupta, Jon W. Bayless, Behrouz Peikari.
Franklin F. Kuo, “Network Analysis and Synthesis ", John Wiley.
Vanvalkenburg, “Network Analysis ", Printice Hall of India Pvt. Ltd., New Delhi, 1994.
A William Hayt, “Engineering Circuit Analysis,” 8th Edition, McGraw-Hill Education.
A. Anand Kumar, “Network Analysis and Synthesis,” PHI publication, 2019.
Sudhakar, A., Shyammohan, S. P., “Circuits and Network,” Tata McGraw-Hill New Delhi, 1994.

Course Code	Course Name	L	T	P	Cr
DOEE250134	Low Power VLSI Design	3	1	0	4

Course Outcomes:

CO1. To understand the concept of VLSI circuit for low power consumption
CO2. To design various circuits with optimal power consumption
CO3. To get an insight on low power issues and challenges at various design levels
CO4. To design a system with multiple supply and threshold voltages applicable for various applications

Module 1:

Introduction to Low Power Design: Sources of Power dissipation in Ultra Deep Submicron CMOS Circuits – Static, Dynamic and Short circuit components, Effects of scaling on power consumption, Low power design flow, Normalized Figure of Merit, Overview of power optimization at various levels

Module 2:

Power Estimation:
Calculation of Steady state probability, Transition probability, Conditional probability, Transition probability of correlated inputs, Transition density, Estimation of Switching activity, Estimation of glitching power

Module 3:

Design Level Optimization:
Computer arithmetic techniques for low power, Software level power optimization, Pipelining, Parallel Processing and retiming approaches for power minimization, Multiple supply voltage for low power , Optimal drivers of high speed low power ICs

Module 4:

Leakage Power Reduction:
Leakage power reduction techniques – stacking techniques, sleepy keeper technique, and super cut off CMOS, VTCMOS, MTCMOS, DTCMOS, Energy constrained and delay constrained

Module 5:

Sleep Transistor Design: Switch and area efficiency, IR drop, normal Vs reverse body bias
Low Power Techniques Automation:
Low power design techniques automation levels, Power-Aware design flow, Low power design examples using UPF

Labs / Practicals:

NA

References:

1. Gary K. Yeap, Practical Low Power Digital VLSI Design, 2010, First Edition, Springer, USA
2. Jan M. Rabaey, Massoud Pedram, Low power Design methodologies, 2014, First Edition, Springer, US
3. Soudris, Dimitrios, Christian Piguet, Goutis, Costas, Designing CMOS circuits for low power, 2011, First Edition, Springer, USA
4. Michael Keating, David Flynn, Robert Aitken, Alan Gibbons, Kaijian Shi, Low power methodology manual: for system-on-chip design, 2007, Springer, New York
5. Abdelatif Bellaouar, Mohamed Elmasry, Low-Power Digital VLSI Design: Circuits and Systems, 2019, Springer, New York
6. S. Ramamurthy, Medetc, Low Power Digital VLSI Design Circuits and Systems, June 2014

Course Code	Course Name	L	T	P	Cr
DOEE250138	Medical Electronics	3	0	2	4

Course Outcomes:

CO1. Understand human body systems and transducers used in medical instrumentation.
CO2. Identify and explain diagnostic and monitoring equipment used in hospitals.
CO3. Recognize therapeutic devices and their applications.
CO4. Understand and describe modern imaging techniques used in clinical diagnosis.
CO5. Apply knowledge of biotelemetry systems and safety practices in handling medical electronics.

Module 1:

Introduction to Human Physiology & Biomedical Instrumentation:
Bio-potentials and their generation (resting and action potential), Block Diagrams of CNS, cardiovascular, respiratory, and urinary systems, Electrodes: Micro, Skin-Surface, Needle, Biomedical sensors and transducers, Signal conditioning and amplification

Module 2:

Diagnostic & Monitoring Instruments:
ECG (Electrocardiography), EEG (Electroencephalography), EMG (Electromyography), Pulse oximeter (SpO2 monitoring), Blood Pressure monitoring (Sphygmomanometer), Electroretinography (ERG), Audiometry

Module 3:

Therapeutic Instruments:
Electrotherapy devices, Implantable Cardioverter Devices: Pacemakers, ICD, Dialysis machines: Hemodialysis and Peritoneal, Respiratory therapy: Ventilators, Heart-Lung Machine

Module 4:

Modern Imaging Techniques
Ultrasonic imaging: Axial Echo Cardiography, X-ray Imaging, CT, PET, Infrared imaging, MRI technique,Laser Endoscopy

Module 5:

Biotelemetry and Patient Safety
Biotelemetry systems: Block diagram, single and multi-channel telemetry, Chronic disease management, telemedicine, Electrical safety: physiological effects, micro/macro shock, Safety in electrosurgical units: grounding, burns, high-frequency current hazards

Labs / Practicals:

1. Measurement of heart rate using fingertip Pulse Sensor and display on LCD
2. Acquisition and visualization of ECG signal using a basic ECG module and oscilloscope
3. Measurement of body temperature using LM35/DS18B20 and display using microcontroller
4. Demonstration of blood pressure measurement using a digital sphygmomanometer
5. Interfacing of a biomedical sensor (e.g., photoplethysmography - PPG) with Arduino
6. Study and demonstration of ultrasonic imaging using simulation software
7. Demonstration of basic electrical safety measures for biomedical equipment (e.g., checking grounding and insulation resistance)

References:

Textbooks
1. R.S. Khandpur, Handbook of Biomedical Instrumentation, 3rd Edition, McGraw Hill Education, 2014
2. M. Arumugam, Biomedical Instrumentation, Anuradha Publications, 2017
3. Lesile Cromwell, Fred J. Weibell, Erich A. Pfeiffer, Biomedical Instrumentation and Measurement, 2nd Edition, Prentice-Hall of India, 2008

Course Code	Course Name	L	T	P	Cr
DOEE250175	Reconfigurable Computing	3	0	2	4

Course Outcomes:

CO 1: Able to select suitable hardware for an application
CO 2: Able to understand the design methodology of micro-processor system on Chip (SoC) buses, memory peripherals on FPGA
CO 3: Build reconfigurable system using FPGAs
CO 4: Able to evaluate hardware accelerator and achieve acceleration factor for a specific application.
CO 5: Perform partial reconfiguration for various applications using peripheral devices.
CO 6: Demonstrate an embedded system on FPGA using IP blocks.

Module 1:

Introduction to Reconfigurable Computing
Reconfigurable Architectures: Classification of Reconfigurable Architectures. FPGA Technology and Architectures, LUT devices and Mapping, Placement and Partitioning. Programming Technology: HDL Based Programming and High level Synthesis using C, Partial Reconfiguration. Intellectual Property Based Design: Soft core, Firm core and Hard Core, Software tools.

Module 2:

System on chip (SoC) system in FPGA devices
Embedded computer organization and methodology of System on chip (SoC) system in FPGA devices. Design challenges and Differences GPP, DSP, ASIC and FPGA based System on Chip platforms. Application profiling and partitioning, FPGAs vs. Multi-core processor architectures. Xilinx Zynq 7000 family programmable SoC (system on chip) in particular - hybrid device with ARM + FPGA architecture.

Module 3:

Overview of AXI Bus protocol
Design of Master and Slave Bus protocols based IPs.Design Metrics, General purpose peripherals (interrupt, timer, clock, DMA etc.) and special purpose peripherals Serial Transmission protocols & Standards, and advanced high speed buses. Debugging methodologies.

Module 4:

Emulating SoC Architectures on FPGAs
Emphasis on different embedded processors and multiprocessor and architectures. Coprocessor creation, hardware design for System-On-a-Chip. Memory and peripheral interfacing. System level design Tradeoffs, Power, Energy, Performance and Area.

Module 5:

High level synthesis and system modeling
Overview of high level synthesis (HLS). Exploration of HLS tools, System Modeling. Models of Computation and System Specification Languages, High Level Computation/Behavioral Synthesis. Application case study like FFT, JPEG.

Labs / Practicals:

Hands-on Sessions on FPGA development board.

1. To design and implement various combinational logic circuits on an FPGA development board

• Basic logic gates, 2x1 Mux, 4x1 Mux, 3 bit Comparators, Half adder, 3 bit Full adder, Ripple Carry adder using Structural Modelling
• Code converters – 4 bit BCD, Gray Code, Gray to Excess-3 code
• Encoders and Decoders

2. To design and implement various sequential logic circuits on an FPGA development board
• D Flip-flop with Reset
• Up, Down, Up-down Counter
• Ring & Johnson Counter
• Traffic Light Controller implementation with FSM
• IP CORE - Binary Counter
• IPCORE - Adder/Subtractor

3. System on chip (SoC) implementation on FPGA Devices : Zynq 7000
• GPIO configuration and control of LEDs on Zynq Board
• Configuration and data transmission with UART on Zynq Board
4. System Modelling with High Level Synthesis
• Matrix multiplication using the concept of HLS. Tool used : Vivado HLS
• FFT process using concept of HLS. Tool used : Vivado HLS

5. Demonstration of debugging methodologies
• Capture and display internal signal waveforms in real-time on the FPGA using Integrated Logic Analyzer & Virtual Input Output IP Cores

References:

Text Books
1. R. Sass and A. G. Schmidt. Embedded Systems Design with Platform FPGAs Principles and Practices. Elsevier Inc, USA, 2010.
2. The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC, Strathclyde Academic Media , UK ,2014.

Reference Books

1. S. Hauck and A. DeHon, Reconfigurable Computing: The Theory and Practice of FPGA-Based Computing, Morgan Kaufmann, 2008.
2. Cardoso, João M. P.; Hübner, Michael (Eds.), Reconfigurable Computing: From FPGAs to Hardware/Software Codesign, Springer, 2011.

Course Code	Course Name	L	T	P	Cr
DOEE250203	VLSI with VHDL	3	0	2	4

Course Outcomes:

CO1: Explain VLSI design flow, design styles, MOS structure, and MOSFET characteristics.
CO2: Analyze the operation of MOS and CMOS inverters under different load conditions.
CO3: Design and implement combinational and sequential MOS logic circuits.
CO4: Apply VHDL for structural, behavioral, and dataflow modeling of digital systems.
CO5: Develop VHDL codes for combinational and sequential circuits like adders, multiplexers, and counters.

Module 1:

Digital System and MOS Transistor: VLSI design flow, Y chart, Practical design flow, Design Hierarchy Structural Decomposition in
the physical (geometrical) domain, FPGA, Gate Array Design, Standard Cell Based Design, Full
Custom Design, MOS structure, MOS system under external bias , Structure and operation of
MOSFET transistor, MOSFET current-voltage Characteristics.

Module 2:

MOS Inverters: MOS Inverter : Concept and working, Resistive load Inverter, Inverter with n-type MOSFET Load, Enhancement load NMOS, Depletion Load NMOS, Enhancement load and Depletion Load NMOS, CMOS Inverter: Circuit operation and description, Cascaded CMOS Inverter stage

Module 3:

MOS Circuits: Combinational MOS Logic Circuits, CMOS logic circuits, Complex logic circuit, Sequential
MOS circuit, VLSI Technology Environment & Processes in brief.

Module 4:

Introduction to VHDL: Data flow, behavioural, structural, Logic operations viz.AND,OR,NOR,NAND,NOT,EXOR ,EXNOR etc., Adder and Subtractor.

Module 5:

VHDL Programming: Multiplexer and Demultiplexer, Decoder and Encoder,4 bit Parallel Adder, Parity Generator and paritychecker, Basic sequential circuits- SR D Latch, RS, T, JK Flip flop, Parallel input Parallel output
, Shift Register, Up Counter, Down Counter

Labs / Practicals:

1. To design and simulate a digital system using structural decomposition in FPGA, and understand the VLSI design flow with standard cell and gate array perspectives.
2. To study the structure and operation of a MOSFET and plot its I-V characteristics under different gate voltages.
3. To construct and analyze the operation of different NMOS inverter configurations:Resistive Load Inverter, Enhancement-mode NMOS Load Inverter, Depletion-mode NMOS Load Inverter.
4. To construct and analyze a CMOS inverter circuit and observe the behavior of a cascaded CMOS inverter chain.
5. To design and simulate basic CMOS combinational logic gates (NAND and XOR) and understand the working of CMOS logic circuits.
6. To construct a basic sequential circuit (SR latch or D flip-flop) using CMOS gates and to understand the brief process steps in VLSI technology.
7. To write and simulate VHDL code for basic logic gates (AND, OR, NOT, NAND, NOR, XOR, XNOR) using Dataflow modeling, Behavioral modeling, Structural modeling.
8. To design, write VHDL code, and simulate 4-bit Adder using Dataflow modeling.
9. To implement and simulate the following sequential circuits using VHDL:D and JK Flip-Flop, 4-bit SISO Shift

Register, 4-bit Up/Down Counter.
10. To write, simulate, and verify VHDL programs for the following combinational logic circuits: 4:1 Multiplexer, 2:4 Decoder.

References:

1. Basic VLSI Design : Systems and Circuits, Doublas A. Puckness & Kamran Eshraghian, Prentice Hall of India Private Ltd., New Delhi, 1989.
2. Introduction to VLSI System, C. Mead & L. Canway, Addison Wesley Pub Co. 1990.
3. Introduction to NMOS & VLSI System Design, A. Mukharjee, Prentice Hall, 1986.
4. VLSI Design Techniques for analog and digital circuits, R.L. Geiger, P.E. Allen & N.R.Streder, McGraw Hill Int. 1990.
5. VHDL: Programming by Example, Douglas L. Perry
6. Digital Design with RTL Design, VHDL, and Verilog, Frank Vahid

Course Code	Course Name	L	T	P	Cr
DOEE260001	HDL Programming: Verilog and VHDL	3	0	2	4

Course Outcomes:

CO1: Understand the hardware description language paradigm and its role in digital design.
CO2: Write Verilog and VHDL programs for combinational and sequential circuits.
CO3: Model design at structural, dataflow, and behavioral abstraction levels.
CO4: Develop testbenches and perform simulation-based verification of HDL designs.
CO5: Implement HDL designs on FPGA using synthesis tools.

Module 1:

Introduction to HDL and Verilog Basics

Hardware description languages: purpose, role in digital design flow. Verilog basics: modules, ports, wire, reg, integer, parameter. Lexical conventions, operators. Structural modeling: primitive gates, user-defined modules. Dataflow modeling: assign statements, continuous assignment.

Module 2:

Behavioral Verilog

Procedural blocks: always, initial. Sequential statements: if-else, case, for, while, repeat, forever. Blocking and non-blocking assignments. Tasks and functions. System tasks: $display, $monitor, $dumpvars. Verilog testbench: stimulus generation, response checking.

Module 3:

VHDL Fundamentals

VHDL entities, architectures, signal declaration, variable, constant. VHDL data types: std_logic, std_logic_vector, integer, boolean. Concurrent signal assignment, conditional signal assignment, selected signal assignment. Process statement, sequential statements in VHDL. Component instantiation and port mapping.

Module 4:

FSM and Advanced HDL Modeling
Finite state machine design in Verilog and VHDL: Mealy and Moore machines. One-hot encoding, binary encoding. Parameterized modules in Verilog, generics in VHDL. Memories and register files in HDL. Pipeline design using HDL. HDL coding guidelines for synthesis.

Module 5:

Synthesis and FPGA Implementation

Logic synthesis flow: HDL → netlist. Synthesis constraints: timing, area, power. Timing analysis basics: setup and hold constraints. FPGA architecture overview: LUTs, flip-flops, DSP blocks, block RAM. FPGA design flow using Xilinx Vivado or Intel Quartus. Place and route, bitstream generation.

Labs / Practicals:

1. Verilog structural model of 2-input gates and combinational circuits.
2. Dataflow model for half adder, full adder.
3. Behavioral Verilog model for 4-bit counter.
4. Verilog model of 4×1 multiplexer and 2-to-4 decoder.
5. Verilog FSM design: traffic light controller.
6. VHDL entity and architecture for basic combinational circuits.
7. VHDL process-based sequential circuit (D flip-flop, register).

8. Verilog testbench for functional verification of a 4-bit ALU.
9. FPGA implementation: LED blinking and 7-segment display.
10. FPGA implementation of a parameterized N-bit counter with synthesis report analysis.

References:

1. Palnitkar S. – Verilog HDL: A Guide to Digital Design and Synthesis, Prentice Hall PTR.
2. Chu P.P. – RTL Hardware Design Using VHDL, John Wiley & Sons.
3. Brown S. and Vranesic Z. – Fundamentals of Digital Logic with Verilog Design, McGraw-Hill.
4. Ciletti M.D. – Advanced Digital Design with the Verilog HDL, Pearson Education.

Course Code	Course Name	L	T	P	Cr
DOEE260014	Hardware Security and Trust	3	0	2	4

Course Outcomes:

CO1: Identify hardware security threats and attack models.
CO2: Implement cryptographic hardware accelerators.
CO3: Detect and mitigate hardware Trojans in ICs.
CO4: Apply PUF-based authentication and key generation.
CO5: Analyze side-channel attacks and countermeasures.

Module 1:

Introduction to Hardware Security

Hardware vs software security. Attack taxonomy: invasive, semi-invasive, non-invasive. Supply chain threats. Counterfeit ICs. Threat models and trust boundaries. Security metrics for hardware.

Module 2:

Cryptographic Hardware

AES hardware implementation: SubBytes, ShiftRows, MixColumns. RSA and ECC hardware accelerators. Hash function hardware: SHA-256. True Random Number Generators (TRNG). Lightweight cryptography for IoT.

Module 3:

Hardware Trojans

Trojan taxonomy: trigger, payload, location. Insertion in design and fabrication phases. Detection methods: side-channel analysis, functional testing, formal verification. Trojan-resistant design techniques. DARPA TRUST program.

Module 4:

Physical Unclonable Functions (PUF)

PUF concept: uniqueness, reliability, unclonability. Arbiter PUF, Ring Oscillator PUF, SRAM PUF. PUF-based key generation and authentication. Machine learning attacks on PUFs. PUF reliability enhancement.

Module 5:

Side-Channel Attacks and Countermeasures

Power analysis: SPA, DPA, CPA. Timing attacks. Electromagnetic analysis. Fault injection attacks. Countermeasures: masking, hiding, logic-level. Secure coding practices for hardware.

Labs / Practicals:

Lab using FPGA: AES implementation, PUF characterization, power side-channel measurement, hardware Trojan insertion and detection exercises

References:

1. Tehranipoor M. and Koushanfar F. – A Survey of Hardware Trojan Taxonomy and Detection, IEEE Design & Test.
2. Mangard S., Oswald E., Popp T. – Power Analysis Attacks, Springer.
3. Bhunia S. and Tehranipoor M. – Hardware Security: A Hands-on Learning Approach, Morgan Kaufmann.
4. Kocher P. et al. – Differential Power Analysis, CRYPTO 1999.

Course Code	Course Name	L	T	P	Cr
DOEE260015	Neuromorphic Computing	3	0	2	4

Course Outcomes:

CO1: Understand biological neural systems and their VLSI analogs.
CO2: Design silicon neuron and synapse circuits.
CO3: Implement spiking neural network architectures.
CO4: Apply neuromorphic chips to pattern recognition problems.
CO5: Evaluate performance of neuromorphic vs conventional computing.

Module 1:

Biological Neural Systems

Neuron anatomy and function. Action potential generation: Hodgkin-Huxley model. Synapse types: electrical, chemical. Plasticity: Hebbian learning, STDP. Neural coding: rate coding, temporal coding.

Module 2:

Silicon Neuron and Synapse Circuits

Integrate-and-fire neuron: circuit implementation. Conductance-based neuron models. CMOS synapse circuits. Log-domain processing. Subthreshold CMOS design for low-power neural circuits.

Module 3:

Spiking Neural Networks
Spiking neuron models: LIF, Izhikevich. SNN training: SpikeProp, STDP-based learning. Rate-to-spike conversion. Spike encoding methods. SNN vs ANN comparison. Simulation using NEST/Brian.

Module 4:

Neuromorphic Architectures

IBM TrueNorth architecture. Intel Loihi chip. SpiNNaker platform. Memristive synapses. In-memory computing for neural networks. Analog vs digital neuromorphic design trade-offs.

Module 5:

Applications and Future Trends

Pattern recognition using neuromorphic chips. Sensory processing: vision, audition. Robotic control applications. Brain-machine interfaces. Challenges: scalability, programmability, learning. Roadmap for neuromorphic computing.

Labs / Practicals:

Simulation using NEST/PyNN: spiking neural network design, STDP learning rule implementation, pattern recognition tasks, FPGA-based silicon neuron demonstration.

References:

1. Mahowald M. – VLSI Analogs of Neural Visual Processing, Caltech PhD Thesis.
2. Indiveri G. et al. – Neuromorphic Silicon Neuron Circuits, Frontiers in Neuroscience.
3. Davies M. et al. – Loihi: A Neuromorphic Many-core Processor, IEEE Micro.
4. Gerstner W. and Kistler W. – Spiking Neuron Models, Cambridge University Press.

Course Code	Course Name	L	T	P	Cr
DOEE260016	Digital Filters and DSP Processors	3	0	2	4

Course Outcomes:

CO1: Design FIR and IIR digital filters using standard methods.
CO2: Implement filters on DSP processors and FPGAs.
CO3: Analyze fixed-point arithmetic effects in DSP systems.
CO4: Apply multirate signal processing techniques.
CO5: Evaluate DSP processor architectures for real-time applications

Module 1:

Digital Filter Design – FIR

FIR filter characteristics: linear phase, stability. Window method: rectangular, Hamming, Hanning, Blackman windows. Frequency sampling method. Optimal equiripple design (Parks-McClellan algorithm). Computational complexity of FIR filters.

Module 2:

Digital Filter Design – IIR

IIR filter design from analog prototypes: Butterworth, Chebyshev, Elliptic. Impulse invariance and bilinear transformation. Direct form I, II structures. Cascade and parallel forms. Comparison of FIR and IIR filters.

Module 3:

Fixed-Point DSP and Quantization

Fixed-point vs floating-point arithmetic. Quantization error analysis. Coefficient quantization effects. Round-off noise in IIR filters. Limit cycles and overflow oscillations. Scaling to prevent overflow.

Module 4:

Multirate Signal Processing

Decimation and interpolation. Polyphase filter structures. Filter banks: analysis and synthesis. Quadrature Mirror Filters (QMF). Wavelet transforms and filter bank relationship. Applications in audio/video coding.

Module 5:

DSP Processor Architecture

Harvard architecture. MAC unit and barrel shifter. Special addressing modes: bit-reverse, circular. Pipeline and parallelism in DSP. TI C6000 and ARM Cortex-M DSP extensions. Real-time implementation considerations.

Labs / Practicals:

MATLAB/Python filter design and analysis, FPGA implementation of FIR/IIR filters, TI DSK board experiments, fixed-point simulation of DSP algorithms.

References:

1. Proakis J.G. and Manolakis D.G. – Digital Signal Processing, Pearson.
2. Oppenheim A.V. and Schafer R.W. – Discrete-Time Signal Processing, Pearson.
3. Mitra S.K. – Digital Signal Processing: A Computer-Based Approach, McGraw-Hill.
4. Wanhammer L. – DSP Integrated Circuits, Academic Press.

Course Code	Course Name	L	T	P	Cr
DOEE260017	Power Electronics Integrated Circuits	3	0	2	4

Course Outcomes:

CO1: Understand power conversion topologies and their IC implementations.
CO2: Design gate driver circuits for power semiconductor devices.
CO3: Analyze PWM control ICs for switching regulators.
CO4: Design DC-DC converters using IC building blocks.
CO5: Apply power management ICs in battery-operated systems.

Module 1:

Power Semiconductor Devices and Gate Drivers

Power MOSFET, IGBT, GaN, SiC device characteristics. Gate charge and switching losses. Gate driver IC design: bootstrap circuits, level shifters, isolation. Dead-time control. Desaturation detection and protection.

Module 2:

PWM Control ICs

Voltage-mode and current-mode control. PWM IC architecture: UC3825, TL494. Error amplifier design. Slope compensation. Soft-start circuits. Frequency synchronization. Protection circuits: OCP, OVP, thermal shutdown.

Module 3:

DC-DC Converter ICs

Buck, boost, buck-boost converter ICs. Integrated power stage design. Synchronous rectification. Light-load efficiency techniques: PFM, burst mode. Multiphase converters for microprocessor power. Package and thermal design.

Module 4:

Linear Regulators and Battery Management

LDO regulator design: pass element, error amplifier, stability. PSRR optimization. Charge pump circuits. Li-ion battery charger ICs: CC-CV charging. Battery fuel gauge ICs. USB power delivery controllers.

Module 5:

AC-DC and Motor Drive ICs

PFC controller ICs: boost PFC. Flyback and LLC resonant controller ICs. Class-D audio amplifier ICs. Brushless DC motor driver ICs. Stepper motor driver ICs. System-in-Package (SiP) for power management.

Labs / Practicals:

Lab experiments: gate driver design and testing, DC-DC converter IC characterization, LDO design simulation (Cadence/LTSpice), PWM controller breadboard implementation.

References:

1. Mohan N., Undeland T., Robbins W. – Power Electronics, Wiley.
2. Erickson R. and Maksimovic D. – Fundamentals of Power Electronics, Springer.
3. Kazimierczuk M. – Pulse-Width Modulated DC-DC Power Converters, Wiley.
4. Texas Instruments Application Notes – Power Management ICs.

Course Code	Course Name	L	T	P	Cr
DOEE260018	Photonics and Optoelectronic Devices	3	1	0	4

Course Outcomes:

CO1: Explain principles of light-matter interaction in semiconductors.
CO2: Analyze p-n junction optoelectronic devices.
CO3: Design and evaluate laser and LED structures.
CO4: Understand photodetector operation and performance metrics.
CO5: Apply integrated photonics concepts to communication systems.

Module 1:

Optical Fundamentals and Semiconductors

Electromagnetic spectrum. Photon-semiconductor interaction: absorption, emission. Direct and indirect bandgap semiconductors. Optical joint density of states. Carrier recombination: radiative and non-radiative. Quantum confinement effects.

Module 2:

Light Emitting Diodes

LED operating principle. Heterojunction LED structures. Quantum well LEDs. External quantum efficiency and internal quantum efficiency. LED materials: GaAs, AlGaAs, InGaN for visible LEDs. Organic LEDs (OLEDs). LED modulation bandwidth.

Module 3:

Semiconductor Lasers

Stimulated emission and optical gain. Fabry-Perot laser cavity. Threshold condition. DFB and DBR lasers. VCSEL structure and operation. Modulation characteristics: rate equations. Laser linewidth and noise.

Module 4:

Photodetectors

p-n photodiode, p-i-n diode, avalanche photodiode (APD). Responsivity, quantum efficiency, bandwidth. Noise in photodetectors: shot noise, thermal noise. MSM photodetector. Phototransistor. Detector circuits: transimpedance amplifier.

Module 5:

Integrated Photonics

Silicon photonics platform. Waveguides: ridge, rib structures. Modulators: Mach-Zehnder, ring resonator. On-chip photodetectors. Optical interconnects. Photonic integrated circuits (PICs) for data center applications.

Labs / Practicals:

N/A – Theory and tutorial course. Computer simulations of LED/laser characteristics and waveguide modes using COMSOL or Lumerical as tutorials.

References:

1. Saleh B. and Teich M. – Fundamentals of Photonics, Wiley.
2. Chuang S.L. – Physics of Optoelectronic Devices, Wiley.
3. Kasap S.O. – Optoelectronics and Photonics, Pearson.
4. Reed G. and Knights A. – Silicon Photonics: An Introduction, Wiley.

Course Code	Course Name	L	T	P	Cr
DOEE260019	AI/ML for VLSI Design	3	0	2	4

Course Outcomes:

CO1: Apply machine learning techniques to EDA problems.
CO2: Use ML models for timing, power, and area prediction.
CO3: Implement floorplanning and routing using reinforcement learning.
CO4: Design neural network hardware accelerators.
CO5: Evaluate AI-driven design automation tools.

Module 1:

Introduction to ML in EDA

Overview of VLSI design flow. Challenges in modern EDA. ML basics: supervised, unsupervised, reinforcement learning. Feature engineering for EDA datasets. ML model evaluation: cross-validation, metrics. Public EDA datasets and benchmarks.

Module 2:

ML for Pre-Silicon Prediction

Timing closure prediction using regression. Power estimation using ML. IR drop prediction. Congestion prediction during placement. GNN-based net delay models. Fast path slack prediction.

Module 3:

ML for Physical Design

Floorplanning using RL: Google's chip placement paper. Placement optimization using GNNs. RL for global routing. Lithography simulation using ML. Hotspot detection using CNNs. OPC acceleration.

Module 4:

Neural Network Hardware Accelerators

Systolic array architecture for matrix multiplication. Dataflow models: weight stationary, output stationary. Accelerator design: Google TPU, NVIDIA GPU architecture for DNN. Memory bandwidth bottleneck. Quantization-aware hardware design.

Module 5:

AI-EDA Tools and Future Trends

Commercial AI-EDA tools: Synopsys DSO.ai, Cadence Cerebrus. Generative AI for RTL coding. LLM-assisted verification. Neural architecture search (NAS) for hardware. Future of autonomous chip design.

Labs / Practicals:

Python/TensorFlow labs: timing prediction models, GNN for netlist analysis, RL-based floorplanning simulation, TPU-style accelerator design on FPGA.

References:

1. Mirhoseini A. et al. – A Graph Placement Methodology for Fast Chip Design, Nature 2021.
2. Huang T. et al. – Machine Learning for Electronic Design Automation, IEEE TCAD.
3. Chen Y. et al. – DNN Accelerator Survey, ACM Computing Surveys.
4. Goodfellow I. et al. – Deep Learning, MIT Press.

Course Code	Course Name	L	T	P	Cr
DOEE260020	Advanced CMOS Analog Design	3	0	2	4

Course Outcomes:

CO1: Analyze advanced CMOS analog circuit topologies.
CO2: Design high-performance operational amplifiers.
CO3: Implement data converters: ADC and DAC.
CO4: Design phase-locked loop and frequency synthesizer circuits.
CO5: Apply noise analysis techniques to analog circuits.

Module 1:

CMOS Device Models and Advanced Topologies

MOSFET small-signal model. Short-channel effects. Cascode amplifiers: telescopic and folded. Current mirrors: Wilson, cascode, regulated cascode. Differential pair with active load. Body effect in analog design.

Module 2:

Operational Amplifiers

Single-stage and two-stage op-amp design. Gain-bandwidth product. Frequency compensation: Miller compensation, nested Miller. Slew rate and settling time. Rail-to-rail input and output stages. Noise-power trade-off in op-amp design.

Module 3:

Data Converters

ADC architectures: flash, SAR, pipeline, sigma-delta. DAC architectures: resistor string, R-2R, current steering. Static specifications: DNL, INL, offset. Dynamic specifications: SNDR, SFDR, ENOB. Design of a SAR ADC. Sigma-delta modulator design.

Module 4:

Phase-Locked Loops

PLL fundamentals: phase detector, VCO, loop filter. Linear model and loop dynamics. Charge pump PLL. Frequency synthesizer architectures. Fractional-N PLL. PLL noise analysis: phase noise, jitter. DLL design.

Module 5:

Noise and Mismatch in Analog Circuits

Noise types: thermal, flicker (1/f), shot noise. Noise models of MOSFET. Noise figure and noise optimization. Mismatch in CMOS: pelgrom model. Impact of mismatch on analog circuits. Layout techniques for low noise and mismatch.

Labs / Practicals:

Cadence Virtuoso simulations: two-stage op-amp design, SAR ADC behavioral model, PLL transient simulation, noise analysis of differential pair.

References:

1. Razavi B. – Design of Analog CMOS Integrated Circuits, McGraw-Hill.
2. Allen P. and Holberg D. – CMOS Analog Circuit Design, Oxford University Press.
3. Gray P., Hurst P., Lewis S., Meyer R. – Analysis and Design of Analog Integrated Circuits, Wiley.
4. Razavi B. – RF Microelectronics, Pearson.

Course Code	Course Name	L	T	P	Cr
DOEE260021	Fault-Tolerant Computing	3	1	0	4

Course Outcomes:

CO1: Classify faults and failure modes in digital systems.
CO2: Apply error detection and correction coding techniques.
CO3: Design redundant hardware structures for fault tolerance.
CO4: Implement software-based fault tolerance techniques.
CO5: Evaluate dependability metrics of computing systems.

Module 1:

Faults, Errors, and Failures

Fault taxonomy: permanent, transient, intermittent. Physical causes: aging, radiation, manufacturing defects. Failure modes in VLSI circuits. System dependability attributes: reliability, availability, safety. Fault models: stuck-at, delay, bridging faults.

Module 2:

Error Detecting and Correcting Codes

Parity codes. Hamming codes: SEC-DED. BCH codes. Reed-Solomon codes. CRC codes. LDPC codes. Arithmetic error codes: Berger codes, AN codes. Implementation of ECC in memory systems.

Module 3:

Hardware Redundancy Techniques

Static redundancy: TMR, NMR voting. Dynamic redundancy: standby sparing, reconfiguration. Hybrid redundancy. Self-checking circuits: totally self-checking, fail-safe. Concurrent error detection. RAID storage systems.

Module 4:

Software Fault Tolerance

N-version programming. Recovery blocks. Algorithm-based fault tolerance (ABFT). Checkpoint and rollback recovery. Software implemented hardware fault tolerance (SIHFT). Fault tolerance in real-time systems.

Module 5:

Dependability Evaluation

Reliability modeling: exponential, Weibull distributions. MTTF, MTTR, MTBF. Reliability block diagrams. Markov models for repairable systems. Fault tree analysis. FMEA. Accelerated life testing. Dependability benchmarking.

Labs / Practicals:

N/A – Theory and tutorial course. Reliability modeling using MATLAB/Python, fault simulation exercises, Markov chain analysis as tutorials.

References:

1. Pradhan D.K. – Fault-Tolerant Computer System Design, Pearson.
2. Johnson B.W. – Design and Analysis of Fault-Tolerant Digital Systems, Addison-Wesley.
3. Avizienis A. et al. – Basic Concepts and Taxonomy of Dependable and Secure Computing, IEEE TDSC.
4. Lin T. and Costello D. – Error Control Coding, Pearson.

Course Code	Course Name	L	T	P	Cr
DOEE260022	System on Chip (SoC) Design	3	0	2	4

Course Outcomes:

CO1: Understand the architecture and components of a modern SoC.
CO2: Design and integrate IP cores using standard bus protocols.
CO3: Apply hardware-software co-design methodologies.
CO4: Implement on-chip interconnects and memory subsystems.
CO5: Verify SoC designs using simulation and formal methods.

Module 1:

SoC Architecture and IP Cores

Evolution of SoC design. Processor subsystems: ARM, RISC-V cores. IP core types: hard, soft, firm. IP integration and reuse. Platform-based design. SoC design flow overview.

Module 2:

On-Chip Interconnects and Buses

AMBA protocol: AHB, APB, AXI. AXI4 bus architecture: channels, handshaking. Network-on-Chip (NoC): topologies, routing. Bus arbitration and bandwidth analysis. Cache coherency protocols.

Module 3:

Memory Subsystems

On-chip SRAM, ROM, Flash. External memory interfaces: DDR, LPDDR. Memory controller design. Cache hierarchy: L1, L2, L3. TLB and virtual memory. DMA controllers.

Module 4:

Hardware-Software Co-design

HW/SW partitioning. Virtual prototyping using SystemC/TLM. Embedded software integration. Boot sequence and firmware. BSP development. Real-time OS integration (FreeRTOS, Linux).

Module 5:

SoC Verification and Physical Design

Functional verification: UVM testbenches. Formal verification methods. Emulation and FPGA prototyping. Physical implementation: floorplanning, power distribution, clock tree synthesis. Tape-out flow and DFT.

Labs / Practicals:

Lab exercises using Vivado/Quartus: AXI IP integration, SoC prototyping on FPGA, memory interface design, UVM-based testbench development.

References:

1. Keating M. and Bricaud P. – Reuse Methodology Manual for System-on-a-Chip Designs, Springer.
2. Flynn J. and Kolinski R. – AMBA AXI and ACE Protocol Specification, ARM.
3. Wolf W. – FPGA-Based System Design, Prentice Hall.
4. Gajski D. et al. – Embedded System Design, Springer.

Course Code	Course Name	L	T	P	Cr
DOEE260023	Memory Design and Technology	3	1	0	4

Course Outcomes:

CO1: Explain different memory technologies and their operating principles.
CO2: Design SRAM and DRAM cells and peripheral circuits.
CO3: Analyze emerging non-volatile memory technologies.
CO4: Apply techniques for memory reliability and yield improvement.
CO5: Evaluate memory system performance parameters

Module 1:

Memory Fundamentals and SRAM

Memory taxonomy: volatile vs non-volatile. SRAM cell: 6T cell operation, static noise margin, read/write stability. Sense amplifiers: latch-type, current-mirror. Word line and bit line design. SRAM arrays and hierarchy.

Module 2:

DRAM and Flash Memory

DRAM cell: 1T-1C structure, refresh operation. SDRAM, DDR SDRAM interface. Flash memory: NOR and NAND architectures. Floating gate and charge trap cells. Program, erase, and read operations. Wear leveling and error correction.

Module 3:

Emerging Non-Volatile Memories

Phase Change Memory (PCM). Resistive RAM (ReRAM/RRAM). Spin-Transfer Torque RAM (STT-MRAM). Ferroelectric RAM (FeRAM). Comparison of NVM technologies: endurance, retention, density, speed.

Module 4:

Memory Peripheral Circuits

Row and column decoders. Charge pumps and voltage regulators. Write drivers and read circuits. ECC: Hamming codes, BCH codes, LDPC for NAND flash. Redundancy circuits for yield improvement.

Module 5:

Memory System Design and Testing

Memory controller architecture. Memory testing: March algorithms, memory BIST. Fault models: stuck-at, transition, coupling faults. 3D memory stacking: HBM, HMC. In-memory computing concepts.

Labs / Practicals:

N/A – Theory and tutorial course. SPICE simulations of SRAM cells and sense amplifiers may be used as tutorials.

References:

1. Itoh K. – VLSI Memory Chip Design, Springer.
2. Prince B. – High Performance Memories, Wiley.
3. Micheloni R. et al. – Inside NAND Flash Memories, Springer.
4. Weste N. and Harris D. – CMOS VLSI Design, Addison-Wesley.