गोंय विद्यापीठ
ताळगांव पठार,
गोंय - ४०३ २०६
फोन : + ९१ - ८६६९६०९०४८

Goa University
Taleigao Plateau,Goa - 403 206
Tel : +91-8669609048
Email : registrar@unigoa.ac.in
Website : www.unigoa.ac.in

(Accredited by NAAC with Grade A+)

GU/Acad –PG/BoS -NEP/2026-27/207 Date: 09.07.2026

CIRCULAR

Ref. No.: GU/Acad –PG/BoS -NEP/2024/538 dated 24.09.2024
GU/Acad –PG/BoS -NEP/2025-26/266 dated 24.07.2025

In continuation to the above referred Circulars, the syllabus for Semester V & VI of the **Bachelor of Engineering in Electronics and Computer Engineering** Programme as approved by the Academic Council in its meeting held on 20th February 2026 is attached.

The Dean, Faculty of Engineering and Principals of affiliated Colleges offering the **Bachelor of Engineering in Electronics and Computer Engineering** Programme are requested to take note of the above and bring the contents of the Circular to the notice of all concerned.

ASHWIN V LAWANDE Digitally signed by ASHWIN V LAWANDE Date: 2026.07.09 17:43:36 +05'30'

(Ashwin V. Lawande)
Deputy Registrar – Academic

To,

1. The Dean, Faculty of Engineering, Goa University.
2. The Principals of Affiliated Engineering Colleges.

Copy to:

1. Chairperson, BoS in Electronics & Computer Engineering, Goa University.
2. Controller of Examinations, Goa University.
3. Assistant Registrar Examinations (Technical and Allied), Goa University
4. Directorate of Internal Quality Assurance, Goa University for uploading the Syllabus on the University website.

ELECTRONICS AND COMPUTER ENGINEERING AY 2024-2025

SEMESTER - V							
Sr. No.	**Course Category**	**Course Code**	**Title of the Course**	**L**	**T**	**P**	**TCr**
1	**Major**	**ECM-300**	Operating Systems	3	0	0	3
		ECM-301	Operating Systems Lab	0	0	1	1
2	**Major**	**ECM-302**	Fundamentals of VLSI Design	3	0	0	3
		ECM-303	VLSI Design Lab	0	0	1	1
3	**Major**	**ECM-304**	Industrial Automation and Instrumentation	2	0	0	2
		ECM-305	Industrial Automation and Instrumentation Lab	0	0	2	2
4	**Major**	**ECM-306**	Computer organization and Architecture	3	0	0	3
		ECM-307	Computer organization and Architecture Lab	0	0	1	1
5	**Prof. Electives**	**ECM-321**	Information Theory and Coding	3	0	0	3
		ECM-322	Information Theory and Coding Lab	0	0	1	1
		OR					
		ECM-323	ARM Microcontroller and Embedded systems	3	0	0	3
		ECM-324	ARM Microcontroller and Embedded systems Lab	0	0	1	1
		OR					
		ECM-325	Digital Signal Processing	3	0	0	3
		ECM-326	Digital signal Processing Lab	0	0	1	1
			TOTAL	**14**	**0**	**6**	**20**

Approved by: Academic Council on 20th February 2026

SEMESTER - VI							
Sr. No.	**Course Category**	**Course Code**	**Title of the Course**	**L**	**T**	**P**	**TCr**
1	**Major**	**ECM-308**	Digital System Design using Verilog HDL	3	0	0	3
		ECM-309	Digital System Design using Verilog HDL Lab	0	0	1	1
2	**Major**	**ITH-310**	Augmented and Virtual Reality	2	0	0	2
		ITH-311	Augmented and Virtual Reality Lab	0	0	2	2
3	**Major**	**ECS-311**	Computer Networks	3	0	0	3
		ECS-312	Computer Networks Lab	0	0	1	1
4	**Major**	**ECM-310**	Design and Analysis of Algorithms	3	0	0	3
		ECM-311	Design and Analysis of Algorithms Lab	0	0	1	1
5	**Prof. Electives**	**ECM-327**	Soft Computing Techniques	3	0	0	3
		ECM-328	Soft Computing Lab	0	0	1	1
		OR					
		ECM-329	Theory of Computations	3	0	0	3
		ECM-330	Theory of Computations Lab	0	0	1	1
		OR					
		ECM-331	Software Engineering and testing	3	0	0	3
		ECM-332	Software Engineering and testing Lab	0	0	1	1
			TOTAL	**14**	**0**	**6**	**20**

SEMESTER V

Major Courses

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ECM-300**
Title of the Course : **Operating Systems**
Number of Credits : **3**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Basics of Computer	
Course Objectives:	**This course will enable students to:** 1. Describe control structures and techniques used in a typical operating system for process management. 2. Understand the classic problems of Process Synchronization and resource Allocation in case of Deadlocks. 3. Understand memory and file management techniques in operating systems and analyze strategies for efficient resource utilization. 4. Understand file system and storage structures, and apply Linux commands with shell programming for efficient system interaction.	
Contents:		**No of Hours**
UNIT- 1	**Introduction to Operating System**: OS objectives and functions, System calls, Types of System calls. **Process description & control**: Process, process states: creation & termination of processes, two & five model process model, process description: OS control structures, process control structures, process attributes, process control: modes of execution, creation of process, process switching, context switching, **Threads:** processes and threads, types of threads. **CPU Scheduling:** Basic Concepts, Scheduling Criteria, Scheduling Algorithms: FCFS, SJF, SRTF / SRTN, Priority Scheduling, Round Robin Scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling.	**13**
UNIT-2	**Process Synchronization:** Background, The Critical – Section Problem, Peterson's solution, Mutex locks, Semaphores, classic problems of Synchronization: The Bounded Buffer Problem, the Readers-Writers Problem, The Dining- Philosophers Problem. **Deadlocks:** System model, deadlock characterization, methods for handling deadlocks, deadlock prevention, deadlock	**10**

	avoidance, deadlock detection, recovery from deadlock.	
UNIT -3	**Memory Management:** Memory management requirements, Memory Partitioning: Fixed Partitioning, Dynamic Partitioning, Memory Allocation Strategies: Best-Fit, First Fit, Worst Fit, Paging and Segmentation. **Virtual Memory:** Background, Demand Paging, Page Replacement: Basic Scheme, FIFO, Optimal, LRU. **File System:** File Concept, Access methods, Directory Structure.	**11**
UNIT -4	**Implementing File systems**: File system structure, Implementation, Directory implementation, Allocation methods. **Secondary Storage structure:** Disk structure and attachment, Disk scheduling; FCFS, SSTF, SCAN, CSCAN, LOOK. Disk management **Linux Commands:** Basic Linux commands, Essential Shell Programming.	**11**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**TEXTBOOKS:** 1. Abraham Silberschatz, Peter B Galvin, Greg Gagne, Operating System Concepts, 10th Edition, John Wiley & Sons, 2018, ISBN: 9781118063330 2. Sumitabha Das, UNIX Concepts and Applications, 4th Edition, McGraw Hill Education, 2017, ISBN: 9780070635463 3. William Stallings, Operating Systems Internals and Design Principles, 9th Edition, Pearson, 2018, ISBN: 9780134670959 **REFERENCE BOOKS:** 1. Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 5th Edition, Pearson, 2022, ISBN: 9780137618873 2. Dhananjay M Dhamdhere, Operating Systems A Concept-Based Approach, 3rd Edition, McGraw Hill Education India, 2012, ISBN: 9781259005589 3. Milan Milenkovic, Operating Systems Concepts and Design, 2nd Edition, Tata McGraw Hill, 2001, ISBN: 9780074632727 4. William E Shotts Jr, The Linux Command Line A Complete Introduction, 2nd Edition, No Starch Press, 2019, ISBN: 9781593279523	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Explain the objectives of an Operating System, process cycle and Compare various CPU Scheduling algorithms. **CO2:** Apply synchronization mechanisms to solve classic concurrency problems and evaluate techniques for handling deadlocks in operating	

	systems. **CO3:** Analyze memory allocation and page replacement strategies, file access methods, and directory structures to achieve efficient data and resource management. **CO4:** Analyze disk scheduling algorithms for performance, and apply Linux commands with basic shell scripting to manage operating system functionalities.

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ECM-301**
Title of the Course : **Operating Systems Lab**
Number of Credits : **1**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Basic Programming skills	
Course Objectives:	**This course will enable students to:** 1. Understand the fundamental concepts and functionalities of operating systems. 2. Implement and analyze various CPU scheduling algorithms and memory management techniques for efficient resource utilization. 3. Develop problem-solving skills through the implementation of inter-process communication, synchronization, and deadlock-handling mechanisms. 4. Gain hands-on experience in programming, shell scripting, and the use of system calls for process management.	
Contents:	**List of Programs /Experiments**	**No of Hours**
	1. Implementation of scheduling algorithm- FCFS. 2. Implementation of scheduling algorithm- SJF 3. Implementation of scheduling algorithm- RR 4. Implementation of System Calls 5. Implementation of Producer consumer problem 6. Implementation of Petersons Solution. 7. Implementation of Bankers Algorithm 8. Implementation of memory allocation in memory management 9. Implementation of segmentation in memory management 10. Shell script programming 11. Study and execution of Linux commands 12. Implementation of Disk Scheduling Algorithms.	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**TEXTBOOKS:** 1. Abraham Silberschatz, Peter B Galvin, Greg Gagne, Operating System Concepts, 10th Edition, John Wiley & Sons, 2018, ISBN: 9781118063330 2. Sumitabha Das, UNIX Concepts and Applications, 4th Edition, McGraw Hill Education, 2017, ISBN: 9780070635463	

	3. William Stallings, Operating Systems Internals and Design Principles, 9th Edition, Pearson, 2018, ISBN: 9780134670959 **REFERENCE BOOKS:** 1. Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 5th Edition, Pearson, 2022, ISBN: 9780137618873 2. Dhananjay M Dhamdhere, Operating Systems A Concept-Based Approach, 3rd Edition, McGraw Hill Education India, 2012, ISBN: 9781259005589 3. Milan Milenkovic, Operating Systems Concepts and Design, 2nd Edition, Tata McGraw Hill, 2001, ISBN: 9780074632727 4. William E Shotts Jr, The Linux Command Line A Complete Introduction, 2nd Edition, No Starch Press, 2019, ISBN: 9781593279523
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Demonstrate understanding of operating system structures, types, and core functions. **CO2:** Apply commands and shell scripts to automate system tasks and manage processes effectively. **CO3:** Design and implement solutions to classical synchronization and concurrency problems using semaphores, system calls, and synchronization algorithms. **CO4:** Implement and evaluate CPU scheduling, memory allocation, and disk scheduling algorithms.

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ECM-302**
Title of the Course : **Fundamentals of VLSI Design**
Number of Credits : **3**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Basic Electronic Engineering, Knowledge of Digital System Design	
Course Objectives:	**This course will enable students to:** 1. Understand of VLSI design flow, MOS devices, and CMOS logic circuits. 2. analyse CMOS circuits in terms of DC behaviour, switching characteristics, power, and noise margins. 3. Demonstrate proficiency in standard VLSI design practices and tools, including SPICE modelling and basic layout concepts.. 4. Exposure to contemporary VLSI technologies and implementation platforms, such as advanced CMOS processes, FinFETs, Verilog HDL, and FPGA architectures.	
Contents:		**No of Hours**
UNIT- 1	**Introduction to VLSI**: VLSI Design Flow. **MOS transistors:** Structures, MOS system under external bias, operation of MOS transistor (MOSFET), MOS transistors: Threshold voltage MOSFET current-voltage characteristics (CGA), channel length modulation, substrate bias effect. **Measurements of parameters** - K_N, V_{TO} & γ. Overview of MOSFET capacitances. **CMOS inverter design:** operation, DC characteristics, calculation of VIL, VIH, VTH, VOH and VOL. Noise margins power and area considerations. Latch up and its prevention	**12**
UNIT-2	**Switching Circuit Characteristics**: Rise, fall and delay time, gate delays, transistor sizing, static and dynamic power dissipations CMOS logic gate design: Fan in and fan out. Modelling of MOS transistor circuits using SPICE. (Level 1 model equations). **CMOS transistor switches:** CMOS logic- Inverter, NOR, NAND and combinational logic, Compound gates, Boolean Expressions using CMOS logic. **Transmission gates:** Multiplexers, logic gates, Latches and Registers. Implementation of Boolean Expressions using transmission gates.	**11**
UNIT -3	Stick diagrams and Layout of Inverter, NOR, NAND, OR and AND gates. Complex logic gates and their layouts (Euler paths). MOSIS layout design rules.	**11**

	Silicon semiconductor technology: Wafer processing, oxidation, epitaxy, deposition, etching, Photolithography, Ion-implantation, and diffusion. Silicon gate process, Chemical Vapour Deposition, Metallization. **Basic CMOS technology:** n-well and p-well CMOS process. Silicon on insulator. Basic structure of FinFET and cross section of FinFET and List of VLSI Tech nodes (10um to 3nm)	
UNIT -4	**Introduction to Verilog language**. Verilog Programs and test benches for Adder, Subtractor, Decoder, Encoder, Multiplexer, Demultiplexer (using dataflow and gate level modelling). **Introduction to FPGA**: Basic FPGA Design Flow, FPGA Device Overview, FPGA types, SRAM –Based FPGA Architecture, FPGA Logic Block Structure, FPGA Routing Matrix and Global Signals, FPGA Clock Resources, FPGA Memory, Advance FPGA Features and Generic FPGA architecture.	**11**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**TEXTBOOKS** 1. Neil H E Weste, David Harris, CMOS VLSI Design A Circuits and Systems Perspective, 4th Edition, Pearson, 2015, ISBN: 9789332542884 2. R C Cofer, Ben Harding, Rapid System Prototyping with FPGAs Accelerating the Design Process, 1st Edition, Elsevier Newnes, 2005, ISBN: 9780750678667 3. S Palnitkar, Verilog HDL A Guide to Digital Design and Synthesis, 2nd Edition, Prentice Hall, 2003, ISBN: 9780130449115 4. Sung-Mo Steve Kang, Yusuf Leblebici, CMOS Digital Integrated Circuits Analysis and Design, 4th Edition, McGraw-Hill Education, 2014, ISBN: 9780073380629 **REFERENCES** 1. Jan M Rabaey, Anantha Chandrakasan, Borivoje Nikolic, Digital Integrated Circuits A Design Perspective, 2nd Edition, Pearson Education, 2003, ISBN: 9780131207646 2. Samar K Saha, FinFET Devices for VLSI Circuits and Systems, 1st Edition, CRC Press, 2020, ISBN: 9781138586093 3. Wayne Wolf, Modern VLSI Design Systems on Silicon, 3rd Edition, Prentice Hall, 2002, ISBN: 9780130179739	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Explain MOSFET operation, current–voltage characteristics, CMOS inverter theory, and the fundamentals of advanced FPGA architectures. **CO2:** Describe the various MOSFET fabrication processes used in CMOS technology.	

	CO3: Model MOSFET-based circuits and implement complex combinational functions using CMOS logic. **CO4:** Design and verify combinational digital circuits using hardware description languages.

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ECM-303**
Title of the Course : **VLSI Design Lab**
Number of Credits : **1**
Effective from AY : **2024-2025**

Pre-requisites for the course:	Knowledge of basic electronics, C programming and digital logic.	
Course Objectives:	**This course will enable students to:** 1. Understand the fundamentals of RTL design and functional verification using Verilog modeling styles. 2. Know simulation and verification techniques to validate digital circuits using automated testbenches and FPGA platforms. 3. analyse MOSFET device characteristics and CMOS circuit behaviour through SPICE-based electrical simulations. 4. design and evaluate CMOS layouts and standard cells using VLSI CAD tools, ensuring DRC, LVS, and post-layout correctness.	
		No. of Hours
Content:	**List of Experiments** **FRONT-END RTL DESIGN & FUNCTIONAL VERIFICATION** **(using tools like Icarus, Xilinx Vivado)** 1. **RTL Design: Combinational Circuit** Design a combinational circuit using gate level, dataflow and behavioral Verilog. 2. **Self-Checking Testbench for 4×1 MUX** Develop a complete automated testbench using $display to report pass/fail status. 3. **Sequential Circuits** Implement synchronous sequential logic using procedural blocks and clock edges. 4. **Structural Modeling: 4-bit Ripple-Carry Adder** Create a hierarchical design by instantiating four Full-Adder modules. **CIRCUIT PHYSICS & SPICE SIMULATION** **(using tools like LT SPICE)** 1. **MOSFET V_T and I–V Characterization** Obtain I_{DS}–V_{GS} and I_{DS}–V_{DS} characteristics and extract the threshold voltage. 2. **CMOS Inverter Voltage Transfer Characteristic (VTC)**	**30**

	Perform a DC sweep, plot the VTC, and determine V_{IH}, V_{IL}, V_{OH}, V_{OL}. 3. **CMOS Inverter Delay Analysis** Measure rise time, fall time, t_{pHL}, t_{pLH}, and propagation delay. 4. **Static & Dynamic Power Measurement** Calculate leakage (static) power and switching (dynamic) power of the inverter. **MANUAL LAYOUT DESIGN** **(using tools like Electric VLSI Design, magic)** 1. **CMOS Inverter Layout** Create layout, perform DRC, extract the netlist, and simulate post-layout behavior. 2. **2-Input CMOS NAND Gate Layout** Use shared diffusion, complete DRC, and extract the layout netlist.	
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**TEXTBOOKS** 1. R C Cofer, Ben Harding, Rapid System Prototyping with FPGAs Accelerating the Design Process, 1st Edition, Elsevier Newnes, 2005, ISBN: 9780750678667 2. Samir Palnitkar, Verilog HDL A Guide to Digital Design and Synthesis, 2nd Edition, Prentice Hall Pearson, 2003, ISBN: 9780130449115 3. Sung-Mo Steve Kang, Yusuf Leblebici, CMOS Digital Integrated Circuits Analysis and Design, 3rd Edition, McGraw-Hill Education, 2003, ISBN: 9780072460537 **REFERENCES** 1. Jan M Rabaey, Anantha Chandrakasan, Borivoje Nikolic, Digital Integrated Circuits A Design Perspective, 3rd Edition, Pearson, 2023, ISBN: 9780137432158 2. Samar K Saha, FinFET Devices for VLSI Circuits and Systems, 1st Edition, CRC Press, 2020, ISBN: 9781138586093 3. Wayne Wolf, Modern VLSI Design Systems on Silicon, 4th Edition, Pearson, 2008, ISBN: 9780131549784	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Design combinational and sequential digital circuits using appropriate hardware description language modelling styles and testbenches. **CO2:** Analyze transistor characteristics, inverter voltage transfer characteristics, delay, and power consumption using circuit-level simulations.	

	CO3: Implement hierarchical register-transfer level modules and validate their functionality through hardware prototyping. **CO4:** Create basic integrated circuit layouts by performing design rule checking, layout-versus-schematic verification, and post-layout simulation.

Name of the Programme : **B.E in Electronics and Computer Engineering**
Course Code : **ECM-304**
Title of the Course : **Industrial Automation and Instrumentation**
Number of Credits : **02**
Effective from AY : **2024-2025**

Pre-requisites for the course	Knowledge of Basic Electronics	
Course Objectives:	**This course will enable students to:** 1. Understand the principles and operation of modern digital measurement, visualization, and data acquisition systems used in instrumentation and automation. 2. Gain insight into transducer technologies and sensing principles. 3. Comprehend industrial automation concepts through the study of Programmable Logic Controllers (PLCs), including system architecture and programming methodologies. 4. Acquire knowledge of supervisory control architectures and industrial communication paradigms employed in distributed and networked automation systems.	
Content:		**No of Hours**
UNIT- I	**Digital Voltmeters**: Non-integrating type: Successive Approximation. Integrating type: Dual Slope Integrating Voltmeter; Sensitivity & Resolution of a DVM. **Oscilloscope:** Block diagram, CRT features, Probes compensation **Digital Storage Oscilloscope**: Block diagram and working **Data Acquisition systems (DAS):** Objective & basic block diagram, Signal conditioning of the inputs	**08**
UNIT-2	Factors for selecting transducers, types of transducers **Temperature Measurement Transducers:** Resistance Temperature Detectors, Thermistors, Thermocouples. **Displacement Transducers:** Strain Gauge, Linear Variable Differential Transformer, Potentiometric **Pressure Transducers:** Inductive, Resistive and Capacitive **Optical transducers:** Photoresistor, Photodiode, Phototransistor	**07**
UNIT-3	**Programmable Logic Controllers (PLC)**: PLC Advantages & Disadvantages, Overall PLC System, CPU & Programmable Monitors, PLC input & Output Modules (Interfaces), Factors considered for selecting PLC. **General PLC Programming Procedure:** Proper Construction of PLC Ladder diagrams, Process Scanning considerations.	**08**

UNIT-4	**PLC Counter functions:** PLC Counters, Examples of Counter Functions, Industrial applications. **PLC data handling instructions:** Move, Conditional Jump, Call Subroutine instructions. **SCADA systems:** SCADA systems, Remote terminal units. **Basic standards:** RS-232 and RS-485, Electrical signal characteristics **Modbus:** General overview, Modbus protocol structure. **Fieldbus:** Introduction, General Fieldbus architecture, Basic requirements of Fieldbus standard.	**07**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**TEXT BOOKS:** 1. Albert D Helfrick, William D Cooper, Modern Electronic Instrumentation and Measurement Techniques, 1st Edition, Prentice Hall of India, 1990, ISBN: 9780135940488 2. H S Kalsi, Electronic Instrumentation, 3rd Edition, Tata McGraw Hill, 2011, ISBN: 9780070702066 3. John W Webb, Ronald A Reis, Programmable Logic Controllers Principles and Applications, 5th Edition, Prentice Hall of India, 2002, ISBN: 9780130607188 **REFERENCE BOOK** 1. D. Patranabis, Principles of Industrial Instrumentation, 3rd Edition, Tata McGraw Hill, 2010, ISBN: 9780070699717 2. G. Clarke, D. Reynders, E. Wright, Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems, 1st Edition, Newnes (Elsevier), 2004, ISBN: 9780750657976	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Explain the operating principles and performance characteristics of digital measuring instruments, oscilloscopes, and data acquisition systems used in industrial applications. **CO2:** Analyze the characteristics and suitability of transducers for measuring temperature, displacement, pressure, and optical parameters in instrumentation systems. **CO3:** Apply Programmable Logic Controller concepts and ladder programming techniques to implement basic industrial automation and control tasks. **CO4:** Design and simulate various industrial control applications using the programmable logic controllers.	

Name of the Programme : **B.E in Electronics and Computer Engineering**
Course Code : **ECM-305**
Title of the Course : **Industrial Automation and Instrumentation Lab**
Number of Credits : **02**
Effective from AY : **2024-2025**

Pre-requisites for the course	Basics of Electrical and Electronics	
Course Objectives:	**This course will enable students to:** 1. understand of the working of DSO advanced functions and Spectrum Analyzer 1. determine the characteristics of the different types of the transducers. 2. construct the virtual instrumentation system. 3. develop PLC Ladder diagrams for industrial control mechanisms and integrate it with SCADA	
Content:	**List of Experiments**	**No. of Hours**
	A. Instrumentation and Measurement Systems 1. Characteristics of Platinum Resistance Temperature Detector (RTD) 2. Characteristics of Thermistor / Temperature Sensors 3. Characteristics of Light Dependent Resistor (LDR) 4. Displacement Measurement Using Linear Variable Differential Transformer (LVDT) 5. Measurement of Strain Using Strain Gauge Load Cell 6. Measurement Using Capacitive and Inductive Sensors 7. Signal Conditioning for Sensors (Amplification, Filtering) **B. Data Acquisition and Programming** 8. Programming Loops and Conditional Operations for Sensor Data Acquisition 9. Handling Arrays and Groups of Sensor Signals 10. Data Logging and Real-Time Monitoring of Sensor Data 11. Development of Modular Functions / Subroutines for Measurement Automation 12. Implementation of Basic Signal Processing Algorithms (Moving Average, Filtering) **C. Programmable Logic Controller (PLC) Programming** 13. PLC Programming – Implementation of Logic Gates 14. PLC Programming – Latching, Jogging, and Interlocking Operations 15. PLC Programming – Timer Functions 16. PLC Programming – Counter Functions 17. PLC Programming – Sequential Control of Simple Processes **D. Supervisory Control and Automation Systems**	**60**

	18. SCADA System Applications – Tank Level Monitoring 19. SCADA System Applications – Automated Process Control 20. PLC Interfacing with SCADA Systems	
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**TEXT BOOKS:** 1. H S Kalsi, Electronic Instrumentation and Measurements, 4th Edition, McGraw Hill Education, 2019, ISBN: 9789353162511 2. John W Webb, Ronald A Reis, Programmable Logic Controllers Principles and Applications, 5th Edition, PHI Learning, 2009, ISBN: 9788120323087 3. Jovitha Jerome, Virtual Instrumentation Using LabVIEW, 1st Edition, PHI Learning Pvt Ltd, 2010, ISBN: 9788120340305	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Explain working principles and characteristics of transducers. **CO2:** Develop and implement VI programs. **CO3:** Construct and troubleshoot PLC logic circuits **CO4:** Integrate PLC with SCADA	

Name of the Programme : **B.E in Electronics and Computer Engineering**
Course Code : **ECM-306**
Title of the Course : **Computer organization and Architecture**
Number of Credits : **3**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Basics of Computer, Logic design, Programming Knowledge	
Course Objectives:	**This course will enable students to:** 1. Understand the structure, functions, and characteristics of computer systems. 2. Identify the elements of instruction sets and their impact on processor design. 3. Explain the functions of each level of the memory hierarchy and I/O mechanisms. 4. Evaluate processor structures and pipelining techniques to assess computer architecture efficiency.	
Contents		**No of hours**
UNIT-1	**Introduction to Computer Organization:** Computer components, Functions, interconnection Structure, Bus Interconnection. **Computer Arithmetic:** Integer Arithmetic: Addition, Subtraction, Multiplication unsigned, signed (Booths Algorithm), Division-unsigned, signed. Floating-Point Representation: IEEE 32 bits, 64 bits. Floating-Point Arithmetic: Addition, Subtraction, Multiplication, Division.	**10**
UNIT-2	**Microprocessors:** Organization of Microprocessor-Based system, Pin description and Internal architecture of Intel 8085A microprocessor: 8085 Programming model **Programming the 8085:** 8085 Machine Cycles and Bus Timings: Opcode fetch, memory and I/O read/write Instruction Set and ALP: Addressing modes; Data transfer operations, arithmetic operations, logic operations, branch operations, Counters and time delays.	**11**
UNIT-3	**Memory System**: Basic concepts, Characteristics, Hierarchy, Semiconductor RAM Memories, Internal Organization of Memory Chip, Read-Only Memories, Static RAM, Dynamic RAM, Difference between SRAM and DRAM, Asynchronous DRAM, Synchronous DRAM, Cache Memory Principles, Elements of Cache Design, Mapping functions, Replacement Algorithm, Unified versus Split Cache	**13**

	Input/Output: External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access	
UNIT-4	**Instruction Pipeline:** Basic Concepts of Pipelining, Pipelining Strategy, Performance and Hazards. **Architecture:** Instruction Execution Characteristics, Use of Large Register File, Compiler based register optimization, Reduced Instruction Set Architecture, RISC v/s CISC **Control Unit Operation:** Micro Operations, Control of the CPU, Hardwired Implementation. **Micro programmed Control:** Basic Concepts, Microinstruction Sequencing, and Microinstruction Execution.	**11**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References/ Readings:	**TEXTBOOKS** 1. Morris Mano, Computer System Architecture, 3rd Edition, Pearson Education, 2017, ISBN: 9789332555610 2. Ramesh S Gaonkar, Microprocessor Architecture Programming and Applications with the 8085, 5th Edition, Penram International Publishing, 2007, ISBN: 9788187972062 3. William Stallings, Computer Organization and Architecture Designing for Performance, 11th Edition, Pearson Education, 2022, ISBN: 9781292420103 **REFERENCE BOOKS** 1. Andrew S Tanenbaum, Todd Austin, Structured Computer Organization, 6th Edition, Pearson, 2013, ISBN: 9780132916524 2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Naraig Manjikian, Computer Organization and Embedded Systems, 6th Edition, McGraw Hill Education, 2012, ISBN: 9780073380650 3. David A Patterson, John L Hennessy, Computer Organization and Design The Hardware Software Interface, 6th Edition, Morgan Kaufmann, 2021, ISBN: 9780128201091 4. Douglas V Hall, Microprocessors and Interfacing Programming and Hardware, 2nd Edition, McGraw Hill Education, 2006, ISBN: 9780070601677	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Explain the concepts of computer organization and computer arithmetic to solve basic computational and architectural problems. **CO2:** Describe the architecture, instruction set, and machine cycle operations of the 8085 microprocessors to develop and execute assembly language programs. **CO3:** Analyze the memory hierarchy concepts, cache organization, I/O mechanisms, and processor functions	

	CO4: Evaluate pipelining techniques, control unit design, and RISC/CISC architectural features to assess processor efficiency and architectural trade-offs**.**

Name of the Programme : **B.E in Electronics and Computer Engineering**
Course Code : **ECM-307**
Title of the Course : **Computer organization and Architecture Lab**
Number of Credits : **1**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Basics Knowledge of Computer, Awareness regarding any programming language like 'C'	
Course Objectives:	**This course will enable students to:** 1. Gain practical exposure to assembly language programming for implementing arithmetic, logical, and data transfer operations on a microprocessor. 2. Develop hands-on experience in implementing algorithmic solutions such as signed multiplication and floating-point operations at the assembly level. 3. Understand program control mechanisms, including branching, looping, counters, and time-delay generation. 4. Acquire skills in manipulating memory and data structures through array processing, comparison, and block data transfer operations	
	List of Experiments	**No of hours**
Contents	1. Write an assembly language program to perform arithmetic and logical operations and observe the effect on condition flags 2. Implement a program to perform Multiplication of two binary numbers using Booth's Algorithms 3. Implement a program to perform floating-point addition on two numbers. 4. Implement a program to perform floating-point multiplication on two numbers. 5. Write an assembly language code to implement data transfer instruction. 6. Write an assembly language code to store numbers in reverse order in memory location. 7. Write an assembly language program to compare a set of numbers and determine the largest and smallest values stored in memory. 8. Write an assembly language code to count the number of positive and negative numbers in a given array of signed integers. 9. Write assembly language programs to implement counters and time delay routines using looping constructs	**30**

	10. Write an assembly language program to simulate basic memory operations such as block data transfer or memory content manipulation	
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**TEXTBOOKS** 1. Morris Mano, Computer System Architecture, 3rd Edition, Pearson Education, 2017, ISBN: 9789332555610 2. Ramesh S Gaonkar, Microprocessor Architecture Programming and Applications with the 8085, 5th Edition, Penram International Publishing, 2007, ISBN: 9788187972062 3. William Stallings, Computer Organization and Architecture Designing for Performance, 11th Edition, Pearson Education, 2022, ISBN: 9780137522679. **REFERENCES:** 1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Naraig Manjikian, Computer Organization and Embedded Systems, 6th Edition, McGraw Hill Education, 2012, ISBN: 9780073380650 2. John L Hennessy, David A Patterson, Computer Architecture A Quantitative Approach, 6th Edition, Morgan Kaufmann, 2019, ISBN: 9780128119051 3. Kai Hwang, Advanced Computer Architecture Parallelism Scalability Programmability, 2nd Edition, McGraw Hill Education, 2013, ISBN: 9780073380407	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Write assembly language programs to perform arithmetic, logical, and data transfer operations. **CO2:** Implement algorithm-based programs such as signed multiplication and floating-point operations at the assembly level. **CO3:** Apply branching, looping, counters, and delay routines to control program execution. **CO4:** Manipulate memory and data structures using array processing, comparison, and block data transfer techniques.	

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ECM-321**
Title of the Course : **Information Theory and Coding**
Number of Credits : **03**
Effective From AY : **2024-2025**

Pre-requisites for the Course:	Applied Mathematics	
Course Objectives	**This course will enable students to:** 1. Understand the fundamental concepts of information theory such as entropy, information rate, mutual information, and channel capacity. 2. Compute and evaluate parameters to determine the channel capacity of communication channels. 3. Understand the principles of source and channel coding to achieve reliable and efficient data transmission. 4. Implement error control coding techniques for error detection and correction in communication systems.	
Contents		**No. of Hours**
Unit 1	**Information Theory:** Information content, unit of information, Entropy, Entropy of binary source, rate of information, Joint entropy and conditional entropy. **Mutual Information and channel capacity:** Noise free channel, Channel with independent input and output, Symmetric channel, Binary symmetric channel (BSC), Binary erasure channel (BEC), cascaded channels.	**12**
Unit 2	**Shannon's Theorem, Capacity of Gaussian channel:** Shannon-Hartley Theorem, Bandwidth –S/N Trade-off, Shannon's Limit. **Source Coding:** Coding Efficiency, Shannon –Fano coding, Huffman coding. Lempel-Ziv (LZ) coding.	**10**
Unit 3	**Error Control Coding:** Types of codes, Parity Check bit for error detection, Hamming distance. **Linear Block Codes:** Syndrome and error detection, standard array and syndrome decoding for error correction. Single Parity Check Bit Code, Repeated Codes, Hadamard Code, Hamming Codes, Reed Muller codes, dual codes. **Cyclic Codes:** Encoding and Decoding of Cyclic Codes	**12**
Unit 4	**Burst Error Correction:** Block Interleaving, Convolutional Interleaving, Reed Solomon (RS) Code, Concatenated Codes.	**11**

	Convolutional Coding: Code Generation, Generator Matrix, State and Trellis Diagrams for Convolutional codes, Code tree, Sequential decoding, Viterbi algorithm. **Turbo Codes:** Encoding and Decoding of Turbo Codes.	
Pedagogy	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References	**TEXTBOOKS:** 1. Herbert Taub, Donald Schilling, Goutam Saha, Principles of Communication Systems, 4th Edition, Tata McGraw Hill, 2017, ISBN: 9781259029851 2. Ranjan Bose, Information Theory Coding and Cryptography, 2nd Edition, Tata McGraw Hill Publishing Company Limited, 2008, ISBN: 9780070151512 3. Salvatore Gravano, Information Theory Coding and Cryptography, 2nd Edition, Tata McGraw Hill, 2008, ISBN: 9789385880568 **REFERENCE BOOKS:** 1. Bernard Sklar, Pabitra Kumar Ray, Digital Communications Fundamentals and Applications, 3rd Edition, Pearson Education, 2021, ISBN: 9780134724052 2. J Das, S K Mullick, P K Chatterjee, Principles of Digital Communication, John Wiley, 1986, ISBN: 9780470202401 3. R P Singh, S Sapre, Communication Systems Analog and Digital, 3rd Edition, Tata McGraw Hill, 2012, ISBN: 9781259004605	
Course Outcome	**After going through this course the student will be able to:** **CO1:** Explain the fundamental concepts of information theory. **CO2:** Apply Shannon's theorems and evaluate the capacity of various channels to understand bandwidth–S/N trade-offs **CO3:** Design efficient source coding schemes and algorithms to achieve efficiency and data compression. **CO4:** Evaluate error control coding techniques for burst and random error correction	

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ECM-322**
Title of the Course : **Information Theory and Coding Lab**
Number of Credits : **01**
Effective From AY : **2024-2025**

Pre-requisites for the Course:	Knowledge of applied Mathematics	
Course Objectives	**This course will enable students to:** 1. Describe key information theory measures such as entropy, mutual information, and channel capacity. 2. Implement and examine source coding algorithms, including Shannon–Fano and Huffman codes, for data compression. 3. Design and simulate channel coding schemes for error detection and correction. 4. Apply advanced error control coding techniques such as Convolutional, Reed–Solomon, and Turbo codes to enhance reliable communication.	
Contents	**List of Experiments**	**No. of Hours**
	1. Computation of Joint, Marginal and Conditional Entropy. 2. Computation of Mutual information. 3. Computation of Channel capacity of BSC, BEC and Cascaded BSC channel 4. Generation of Shannon -Fano code 5. Generation of Huffman code 6. Generation of Linear block encoding and decoding 7. Implementation of Cyclic Redundancy encoding and decoding 8. Implementation of Reed-Solomon encoding and decoding 9. Implementation of Convolutional encoding and decoding 10. Implementation of Turbo encoding and decoding	**30**
Pedagogy	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References	**TEXT BOOKS:** 1. Herbert Taub, Donald Schilling, Goutam Saha, Principles of Communication Systems, 4th Edition, Tata McGraw Hill, 2017, ISBN: 9781259029851 2. Ranjan Bose, Information Theory Coding and Cryptography, 2nd Edition, Tata McGraw Hill Publishing Company Limited, 2008, ISBN: 9780070151512	

	3. Salvatore Gravano, Information Theory Coding and Cryptography, 2nd Edition, Tata McGraw Hill, 2008, ISBN: 9789385880568 **REFERENCE BOOKS:** 1. Holly Moore, MATLAB for Engineers, 7th Edition, Pearson Education, 2023, ISBN: 9780137874663 2. John G Proakis, Masoud Salehi, Gerhard Bauch, Contemporary Communication Systems Using MATLAB, 3rd Edition, Cengage Learning, 2013, ISBN: 9781133936510
Course Outcome	**After going through this course, the student will be able to:** **CO1:** Compute entropies, mutual information and channel capacities for given data sets and channels. **CO2:** Implement source coding algorithms for efficient data compression. **CO3:** Implement and test block and cyclic coding schemes for error detection and correction. **CO4:** Simulate convolutional and turbo coding schemes and evaluate their performance in noisy channel conditions.

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ECM 323**
Title of the Course : **ARM Microcontroller and Embedded Systems**
Number of Credits : **3**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Basic structure of Computers and microcontrollers	
Course Objectives	**This course will enable students to:** 1. Understand processor architecture fundamentals, including instruction set concepts, processor organization, memory systems, and pipelines. 2. Acquire knowledge of processor programming models and assembly-level instruction execution, with emphasis on addressing and operating modes. 3. Gain exposure to system-level mechanisms such as exception handling, interrupts, and memory hierarchy. 4. Develop insight into embedded system integration concepts, including coprocessor interfacing, on-chip bus architectures, and development methodologies.	
Contents		**No. of Hours**
Unit 1	**ARM architecture and Processor fundamentals:** Types of computer Architectures, ISA's and ARM History, RISC and ARM Design, architectural inheritance, ARM Programmer's model, memory system, memory formats and data types, ARM core data flow model, Processor modes, registers: General purpose and Program status, flags, Overview of Endianness, unaligned access support. **Pipelines**: ARM 3 and 5 stage Pipeline, hazards, efficiency, ARM family attribute comparison. Exceptions, interrupts and vector table, Core extensions, Jazelle extension ARM Development tools. ARM7TDMI block, core and functional diagrams, memory interface, bus Interface signals and bus cycle types.	**11**
Unit 2	**ARM7TDMI assembly instructions and modes:** Conditional execution, addressing modes: data processing operands, memory access operands, Load and store operands, Stack operations, Shift Operations. **ARM Instruction set:** Branch, data processing, comparison, SIMD, Multiply, miscellaneous data processing, status register transfer, load store, coprocessor, exception-generating instructions. Elementary assembly level programs. **Thumb** state**:** Thumb Programmers model, Thumb exceptions, Implementation and applications. Thumb Instruction set in	**12**

	brief.	
Unit 3	**Exception handling:** ARM processor exceptions and modes, vector table, exception priorities, link offset registers. Interrupt handling: Assigning interrupts, interrupt latency, IRQ and FIQ exceptions, basic interrupt stack design, Interrupt handling schemes: non-nested, nested, reentrant and prioritized simple interrupt handler. ARM7TDMI Exception and abort model, instructions to improve exception handling. **Caches:** Memory hierarchy and cache memory, caches and memory management units, basic architecture of cache memory, set associativity. Relationship between cache and main memory, Cache policy.	**11**
Unit 4	**ARM Coprocessor Interface:** Coprocessor availability, interface signals, handshaking, connecting coprocessors. **Vector Floating Point Processor (VFP) architecture:** Overview, floating point model, registers, floating-point exceptions, compliance with IEEE 754 standard, VFP and ARM interactions. **Advanced Microcontroller Bus Architecture (AMBA):** Overview, Typical AMBA Based Microcontroller, AHB bus features, components, bus interconnection, AHB Bus transfers, APB bus transfers, APB Bridge	**11**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**TEXTBOOKS** 1. Andrew N Sloss, Dominic Symes, Chris Wright, ARM System Developer's Guide Designing and Optimizing System Software, 1st Edition, Morgan Kaufmann, 2004, ISBN: 9781558608740 2. Steve Furber, ARM System-on-Chip Architecture, 2nd Edition, Addison Wesley, 2000, ISBN: 9780201675191 3. William Hohl, Christopher Hinds, ARM Assembly Language Fundamentals and Techniques, 2nd Edition, Routledge, 2017, ISBN: 9781482229851 **REFERENCES** 1. Sarah Harris, David Harris, Digital Design and Computer Architecture ARM Edition, 3rd Edition, Morgan Kaufmann Elsevier, 2022, ISBN: 9780128200643	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Explain processor architecture concepts including instruction sets, pipelines, memory organization, and processor operating modes. **CO2:** Apply assembly-level programming techniques using appropriate addressing modes and instruction execution mechanisms. **CO3:** Analyze exception and interrupt handling mechanisms and	

	evaluate their impact on processor performance and system reliability. **CO4:** Interpret system-level integration features such as cache organization, coprocessor interaction, and on-chip bus architectures in embedded systems.

Name of the Programme : **B.E in Electronics and Computer Engineering**
Course Code : **ECM-324**
Title of the Course : **ARM Microcontroller and Embedded systems Lab**
Number of Credits : **01**
Effective from AY : **2024-2025**

Prerequisites for the Course:	Basic Knowledge of Microcontroller and Programming	
Course Objectives:	**This course will enable students to:**: 1. Understand assembly-level programming concepts for data transfer, arithmetic, logical, and control operations in embedded processors. 2. Comprehend embedded processor architecture, including registers, status flags, and instruction execution flow. 3. Develop basic embedded applications using processor-based platforms and standard interfacing techniques. 4. Configure and interface external peripherals for embedded system applications	
Content:	**List of programs/ Experiments**	**No. of Hours**
	1. Write programs to demonstrate basic data transfer instructions. 2. Write programs to perform arithmetic operations on registers and memory. 3. Write programs to implement logical and bitwise operations. 4. Write programs to illustrate conditional execution and flag updates. 5. Write programs to implement loops and iterative operations. 6. Interface LEDs and switches with the ARM development board. 7. Implement pulse width modulation (PWM) using the ARM development board. 8. Interface an alphanumeric LCD with the ARM development board. 9. Interface a seven-segment display with the ARM development board. 10. Interface a DC motor with the ARM development board	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	

References/ Readings:	**TEXTBOOKS:** 1. Andrew N Sloss, Dominic Symes, Chris Wright, ARM System Developer's Guide Designing and Optimizing System Software, 1st Edition, Morgan Kaufmann, 2004, ISBN: 9781558608740 2. Steve Furber, ARM System-on-Chip Architecture, 2nd Edition, Addison Wesley, 2000, ISBN: 9780201675191 3. William Hohl, Christopher Hinds, ARM Assembly Language Fundamentals and Techniques, 2nd Edition, Routledge, 2017, ISBN: 9781482229851 **REFERENCES** 1. Sarah Harris, David Harris, Digital Design and Computer Architecture ARM Edition, 3rd Edition, Morgan Kaufmann Elsevier, 2022, ISBN: 9780128200643
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Write and execute ARM assembly programs to perform data transfer, arithmetic, logical, and conditional operations. **CO2:** Perform loops, iterative operations, and algorithmic solutions using assembly programming. **CO3:** Interface peripheral devices such as LEDs, switches, LCDs, seven-segment displays, and motors with an ARM development board. **CO4:** Apply pulse width modulation (PWM) and basic control techniques on an ARM development board.

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ECM-325**
Title of the Course : **Digital Signal Processing**
Number of Credits : **3**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Applied mathematics	
Course Objectives:	**This course will enable students to:** 1. Understand the fundamental concepts and classifications of different types of signals and systems. 2. Explain the properties and responses of Linear Time-Invariant (LTI) systems in both time and frequency domains. 3. Introduce Fourier analysis, sampling theory, and Discrete Fourier Transform (DFT) for signal representation and analysis. 4. Design and implement digital filters (IIR and FIR) using standard design techniques**.**	
Contents:		**No of Hours**
UNIT- 1	**Introduction**: Definitions and concept of different types of signals **Classification of signals**: continuous time and discrete time signals; Causal and Non-Causal, Periodic and Non-periodic signals, Signal Energy and Power, Even and Odd Signals. **Basic signal types:** Exponential and Sinusoidal signal; Unit impulse and Unit step, Unit Ramp functions, Sinc function. **Systems:** Continuous time and Discrete time system. **Linear time invariant (LTI) systems:** Introduction, Continuous time LTI system, the convolution integral, Discrete time LTI system, the convolution sum, Impulse Response of LTI system.	**12**
UNIT-2	**Continuous-Time Fourier Transform:** Representation of aperiodic signals: Fourier transform of aperiodic signals. **Discrete-Time Fourier Transform**: Representation of aperiodic signals; Fourier transform of aperiodic signals. Properties. **Sampling of continuous time signals:** Periodic sampling, Frequency domain representation of sampling, Reconstruction of a Band limited Signal from its samples.	**11**
UNIT -3	**The Discrete Fourier transform (DFT):** Introduction, Properties of Discrete Fourier Transform:- Linearity, periodicity, Complex Conjugate, Time shift, Frequency shift, Duality and	**11**

	Convolution in time domain is multiplication in frequency domain. Circular convolution using the DFT. Computation of the Discrete Fourier transform. **Fast Fourier Transform:** Efficient computation of DFT, Decimation- in- time FFT (in-place computations), Decimation-in-Frequency FFT (in-place computations) (Radix – 2 only) (upto 8 point DFT only)	
UNIT -4	**IIR Filters:** IIR Filter design techniques: IIR Filter design by impulse invariant method and bilinear transformation. **Design of IIR Filters:** Butterworth low pass filter design using impulse invariance and bilinear transformation. **FIR Filters:** Magnitude and phase response of digital filters, frequency response of linear phase FIR filters. **Design techniques for FIR filters**: Window techniques (Rectangular, Hamming)**.**	**11**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**TEXTBOOKS:** 1. A V Oppenheim, Alan S Willsky, S Hamid Nawab, Signals and Systems, 2nd Edition, Pearson, 1997, ISBN: 9780138147570 2. A V Oppenheim, Ronald W Schafer, Discrete-Time Signal Processing, 3rd Edition, Pearson, 2009, ISBN: 9780131988422 3. John G Proakis, Dimitris G Manolakis, Digital Signal Processing Principles Algorithms and Applications, 4th Edition, Pearson, 2007, ISBN: 9788131708244 4. S Salivahanan, Digital Signal Processing, 3rd Edition, McGraw Hill Education, 2014, ISBN: 9780070702535 **REFERENCE BOOKS:** 1. Sanjit K Mitra, Digital Signal Processing A Computer-Based Approach, 4th Edition, McGraw Hill Education, 2011, ISBN: 9780073380490 2. Simon Haykin, Barry Van Veen, Signals and Systems, 3rd Edition, Wiley, 2021, ISBN: 9781119783121 3. V Krishnaveni, A Rajeshwari, Signals and Systems, 1st Edition, Wiley India, 2013, ISBN: 9788126542338.	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Classify different types of signals and systems and describe their basic properties. **CO2:** Analyze the behavior of Linear Time-Invariant (LTI) systems using convolution and Fourier techniques.	

	CO3: Examine the sampling process and frequency-domain representations of discrete-time signals. **CO4:** Design and evaluate digital filters (IIR and FIR) using impulse invariant, bilinear, and window methods.

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ECM-326**
Title of the Course : **Digital Signal Processing Lab**
Number of Credits : **1**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Basic Programming skills	
Course Objectives:	**This course will enable students to:** 1. Understand the fundamental concepts of discrete-time signals and systems and their basic operations. 2. Analyze the behavior of Linear Time-Invariant (LTI) systems using convolution and frequency-domain techniques. 3. Apply Fourier Transform, DFT, and FFT algorithms for spectral analysis of discrete-time signals. 4. Design and implement digital filters (IIR and FIR) to meet given frequency specifications using standard techniques.	
Contents:	**List of Programs /Experiments**	**No. of Hours**
	1. Program to generate the following discrete time signals a. Unit Impulse Signal b. Unit Step Signal c. Unit Ramp Signal d. Exponential Signal e. Sinusoidal Signal 2. Program to perform the following operations on signals. a. Signal Addition b. Signal Multiplication c. Scaling d. Shifting e. Folding 3. Program to perform convolution sum for discrete-time LTI systems and verify linearity and time-invariance properties. 4. Demonstrate the effect of sampling a continuous-time signal and reconstruct the signal. Observe aliasing effects for different sampling rates. 5. Computation of N point DFT of a given sequence (without using built-in function) and to plot the magnitude and phase spectrum.	**30**

	6. Using the DFT and IDFT, compute the following for any two given sequences a) Circular convolution b) Linear convolution 7. Develop decimation in time radix-2 FFT algorithm without using built-in functions. 8. Design and implementation of digital IIR Butterworth low pass filter to meet the given specifications. Plot magnitude and phase response. 9. Design and implementation of digital IIR Butterworth high pass filter to meet the given specifications. Plot magnitude and phase response. 10. Design and implementation of digital low pass and high pass FIR filter using a Hamming window to meet the given specifications. Plot magnitude and phase response.	
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**TEXTBOOKS:** 1. A V Oppenheim, Alan S Willsky, S Hamid Nawab, Signals and Systems, 2nd Edition, Pearson, 1997, ISBN: 9780138147570 2. A V Oppenheim, Ronald W Schafer, Discrete-Time Signal Processing, 3rd Edition, Pearson, 2009, ISBN: 9780131988422 3. John G Proakis, Dimitris G Manolakis, Digital Signal Processing Principles Algorithms and Applications, 4th Edition, Pearson, 2007, ISBN: 9788131708244 4. S Salivahanan, Digital Signal Processing, 3rd Edition, McGraw Hill Education, 2014, ISBN: 9780070702535 5. Vinay K Ingle, John G Proakis, Digital Signal Processing Using MATLAB A Problem Solving Companion, 2nd Edition, Cengage Learning, 2012, ISBN: 9781133596400, Cengage Learning, 2018, ISBN: 9386668114 / 9789386668118 **REFERENCE BOOKS:** 1. Sanjit K Mitra, Digital Signal Processing A Computer-Based Approach, 4th Edition, McGraw Hill Education, 2011, ISBN: 9780073380490 2. Simon Haykin, Barry Van Veen, Signals and Systems, 3rd Edition, Wiley, 2021, ISBN: 9781119783121 3. V Krishnaveni, A Rajeshwari, Signals and Systems, 1st Edition, Wiley India, 2013, ISBN: 9788126542338	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Generate and perform basic operations on discrete-time signals. **CO2:** Analyze and verify the properties of Linear Time-Invariant (LTI) systems using convolution.	

	CO3: Compute and interpret the DFT, IDFT, and FFT for discrete-time signals and perform convolution using frequency-domain methods. **CO4:** Design and implement IIR and FIR digital filters and evaluate their frequency responses.

SEMESTER VI

Major Courses

Name of the Programme : B.E. in Electronics and Computer Engineering
Course Code : ECM-308
Title of the Course : Digital System Design using HDL
Number of Credits : 03
Effective from AY : 2024-2025

Pre-requisites for the Course:	Logic Design	
Course Objectives:	**This course will enable students to:** 1. Understand the syntax and semantics of HDL. 2. Write programs for designing combinational circuits. 3. Write programs for designing sequential circuits. 4. Gain knowledge of finite state machines, switch-level modeling, and user-defined primitives.	
Content:		**No. of Hours**
UNIT- 1	Emergence of HDLs, Design Flow using HDLs, Importance of HDLs. Hierarchical Modeling Concepts: Modules, Instances. **Data Types:** Nets, Registers, Vectors, Integer, Real, and Time, Arrays, Memories, Parameters, Strings. Modules and Ports. **Gate Level Modeling:** Design of 4bit Adder cum subtractor, multiplexer, Demultiplexer, Decoder, Encoder. Test benches to verify the Functionality.	**11**
UNIT-2	**Dataflow Modeling:** Continuous assignment statements, **Delays:** Regular assignment delay, implicit assignment delay, and net declaration delay. Define expressions, operators, and operands. **Operator types:** for all possible operations—arithmetic, logical, relational, equality, bitwise, reduction, shift, concatenation, Replication and conditional. 4:1 Multiplexer, Demultiplexer, Decoder, Encoder, Full Adder	**11**
UNIT -3	**Behavioral Modeling: Structured procedures:** always and initial. Blocking and non-blocking procedural assignments. Conditional statements using if and else. Multiway branching using case, casex, and casez statements, looping statements using while, for, repeat, and forever. Definition of sequential and parallel blocks. Generate Blocks (Loop, Conditional, Case)	**11**

	4:1 Multiplexer, comparator, Flip Flops, shift registers, counters, ALU, Memories.	
UNIT -4	**Tasks and functions:** Differences between Tasks and Functions in Verilog, Task and Function declaration and invocation. Examples. **Finite State Machine:** Examples of design using HDL. **Switch Level Modeling:** Switch Level Elements- MOS, CMOS, Bidirectional and Resistive Switches, Power and Ground. Examples- CMOS Inverter, NAND, NOR, 2:1 Multiplexer. **User-Defined Primitives (UDPs):** Introduction to UDPs, Syntax and Structure of UDPs Examples- basic gates, 2:1 multiplexer, D and JK flip flops	**12**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**TEXTBOOKS:** 1. Charles H Roth Jr, Lizy Kurian John, Byeong Kil Lee, Digital Systems Design Using Verilog, 2nd Edition, Cengage Learning, 2020, ISBN: 9780357113355 2. J Bhasker, Verilog HDL Synthesis A Practical Primer, 1st Edition, Star Galaxy Publishing, 1998, ISBN: 9780965039154 3. Samir Palnitkar, Verilog HDL A Guide to Digital Design and Synthesis, 2nd Edition, Pearson Education, 2003, ISBN: 9788177589184 **REFERENCE BOOKS:** 1. IEEE, IEEE Standard 1364-1995: Verilog Hardware Description Language – Language Reference Manual, First Edition, IEEE Press, New York, USA, 1995.	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Understand the syntax, semantics, and hierarchical modelling concepts of a hardware description language. **CO2:** Write HDL programs for combinational circuits and verify their functionality. **CO3:** Apply tasks, functions, switch-level modelling, and user-defined primitives for hardware design. **CO4:** Design sequential circuits, including flip-flops, counters, shift registers, and finite state machines.	

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ECM-309**
Title of the Course : **Digital System Design using HDL LAB**
Number of Credits : **01**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Digital System Design	
Course Objectives:	**The course aims to provide the student with:** 1. proficiency in HDL for modeling, synthesizing, and verifying digital circuits, with emphasis on gate-level and data-flow modeling techniques. 2. Ability to write efficient HDL programs using behavioral modeling to describe and implement complex digital logic. 3. Practical experience in the design and implementation of sequential circuits. 4. Ability to build complete system by integrating multiple digital components, and perform thorough functional verification and testing using HDL.	
Content:	**List of experiments**	**No. of Hours**
	1. Write programs using gate-level modeling, along with test benches, to implement: a. 1-bit Full Adder using Half Adders b. 2-to-4 Decoder using NAND gates only (structural model) 2. Write a program to implement a 4-bit Adder–Subtractor using 1-bit Full Adders and verify the design using a test bench. 3. Write programs using dataflow modeling, along with test benches, to implement: a. 4-to-1 Multiplexer b. Half Subtractor and Full Subtractor 4. Write programs to convert: a. 4-bit Binary code to Gray code b. Gray code to 4-bit Binary code Verify the designs using test benches. 5. Write programs using if and case statements to implement: a. 4-to-1 Multiplexer b. 8-to-3 Encoder c. 3-to-8 Decode Verify each using appropriate test benches.	**30**

	6. Write a program to implement a Comparator and verify its operation using a test bench. 7. Write a program to implement a Memory Unit and verify its functionality using a test bench. 8. Write programs for D Flip-Flop and JK Flip-Flop, along with test benches to verify their operation. 9. Write a program to implement a 4-bit UP/DOWN Counter and verify the design using a test bench. 10. Write programs to demonstrate combinational and sequential circuit modelling techniques, and verify their functionality using test benches.	
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**TEXTBOOKS:** 1. J Bhasker, Verilog HDL Synthesis A Practical Primer, 1st Edition, Star Galaxy Publishing, 1998, ISBN: 9780965039154 2. Samir Palnitkar, Verilog HDL A Guide to Digital Design and Synthesis, 2nd Edition, Pearson Education, 2003, ISBN: 9788177589184 **REFERENCE BOOKS:** 1. Charles H Roth Jr, Lizy Kurian John, Byeong Kil Lee, Digital Systems Design Using Verilog, 2nd Edition, Cengage Learning, 2020, ISBN: 9780357113355 2. IEEE, IEEE Standard 1364-2005 Verilog Hardware Description Language (HDL) Language Reference Manual, IEEE Press, 2005	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Write and verify HDL programs for combinational circuits using gate-level and dataflow modelling. **CO2:** Design sequential circuits including flip-flops, counters, and memory units using behavioural modelling. **CO3:** Implement code conversion, multiplexers, encoders, decoders, and comparators using conditional and case statements. **CO4:** Apply combinational and sequential modelling techniques to integrate and verify complete digital systems.	

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ITH310**
Title of the Course : **Augmented Reality and Virtual Reality**
Number of Credits : **02**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Engineering Mathematics	
Course Objectives:	**This course will enable students to:** 1. Understand the basic concepts of Virtual Reality (VR) and Augmented Reality (AR). 2. Know input and output devices of VR and AR. 3. Understand the interaction techniques of VR and AR. 4. Illustrate the role of Virtual Reality (VR) and Augmented Reality (AR) in addressing real-world industrial applications.	
Contents:		**No. of Hours.**
UNIT- 1	**Introduction to Virtual Reality** Defining Virtual Reality, History of VR , Introduction top VR- Four Key Elements of Virtual Reality Experience, , Human Physiology and Perception, Key Elements of Virtual Reality Experience, Virtual Reality System, Interface to the Virtual World-Input & output- Visual, Aural & Haptic Displays, Applications of Virtual Reality. **Representing the Virtual World** Representation of the Virtual World, Visual Representation in VR, Aural Representation in VR and Haptic Representation in VR	**07**
UNIT- 2	**The Geometry of Virtual Worlds & The Physiology of Human Vision** Geometric Models, Changing Position and Orientation, Axis-Angle Representations of Rotation, Viewing Transformations, Chaining the Transformations, Human Eye, eye movements & implications for VR. **Visual Rendering, Perception, and Interactive Technique Rendering-** Visual Rendering Systems, Aural Rendering Systems, Haptic Rendering Systems **Interacting with the Virtual World-** Manipulating a Virtual World: Manipulation Methods-Direct user control, physical control, and virtual control. Navigating in a Virtual World: Wayfinding	**08**
UNIT- 3	**Augmented Reality (AR):** Taxonomy, Technology and Features of Augmented Reality, AR Vs VR, Challenges with AR, AR systems and functionality, Augmented Reality Methods, Visualization Techniques for Augmented Reality, Enhancing interactivity in AR Environments, Evaluating AR systems **Computer Vision for Augmented Reality**	**08**

	Computer Vision for Augmented Reality - Marker Tracking, Multiple-Camera Infrared Tracking, Natural Feature Tracking by Detection, Simultaneous Localization and Mapping, Outdoor Tracking	
UNIT- 4	**AR Techniques- Marker based & Marker less tracking Marker-based approach-** Introduction to marker-based tracking, types of markers, marker camera pose and identification, visual tracking, mathematical representation of matrix multiplication Marker types- Template markers, 2D barcode markers, imperceptible markers. **Marker-less approach**- Localization based augmentation, real world examples Tracking methods- Visual tracking, feature based tracking, hybrid tracking, and initialization and recovery.	**07**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
Reference/ Readings:	**TEXTBOOKS:** 1. William R Sherman, Alan B Craig, Understanding Virtual Reality, Interface, Application and Design, (The Morgan Kaufmann Series in Computer Graphics), Morgan Kaufmann Publishers, San Francisco, CA, 2002, ISBN-13: 978-1558603530. 2. Dieter Schmalstieg, Tobias Hollerer, "Augmented Reality: Principles & Practice", Pearson Education India, First Edition, 2016, ISBN-10: 9332578494. 3. Sanni Siltanen, "Theory and Applications of Marker-Based Augmented Reality", Julkaisija – Utgivare Publisher, 2012, ISBN: 978-951-38-7449-0. **REFERENCE BOOKS:** 1. Burdea, Grigore C and Philippe Coiffet, "Virtual Reality Technology", Wiley Inter science, India, 2003. 2. Alan B Craig, William R Sherman, Jeffrey D Will, "Developing Virtual Reality Applications: Foundations of Effective Design", Morgan Kaufmann Publishers, 2009. 3. Alan B Craig, "Understanding Augmented Reality: Concepts and Applications", Morgan Kaufmann, 2013.	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Understand the fundamental concepts of AR and VR and describe the input and output devices used. **CO2:** Apply the various representations and techniques in AR and VR. **CO3:** Analyze the methods required for hardware and software in AR and VR **CO4:** Design and implement the real-world examples of marker-based and marker less augmented reality applications.	

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ITH-311**
Title of the Course : **Augmented Reality and Virtual Reality Lab**
Number of Credits : **02**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Basics of programming, Engineering Mathematics	
Course Objectives:	**This course will enable students to:** 1. The fundamental tools, workflows, and development environments used in AR and VR application development using Unity. 2. Applying 3D modelling, lighting, texturing, and interaction techniques to build realistic and engaging AR/VR experiences. 3. Interactive AR applications using AR Foundation and mobile-based deployment workflows. 4. VR interactive techniques using XR Interaction Toolkit and device simulation to prototype virtual environments.	
Contents:	**List of Programs /Experiments**	**No of Hours**
	1. Installation of Unity and Visual Studio, setting up Unity for VR development, understanding documentation of the same. 2. Develop a scene in Unity that includes: cube, plane and sphere, apply transformations on the 3 game objects. 3. Create and manipulate 3D primitive objects and apply camera projections to understand scene composition. 4. Import 3D models from Unity Asset Store and apply lighting, materials, and visual effects to enhance realism. 5. Model simple 3D objects using Blender and import them into Unity; apply textures, materials, and shaders. 6. Create an interactive AR application using AR Foundation with plane detection and virtual object placement. 7. Add audio, text, and particle effects to AR objects to create enhanced AR experiences. 8. Develop VR-enabled applications using Unity XR Interaction Toolkit with simulated controllers. 9. Create AR-based visualization applications such as human anatomy, architecture preview, or science models (DNA/RNA, solar system, etc.). 10. Implement AR image tracking using AR Foundation to display 3D models upon scanning predefined images. 11. Develop an AR application using face tracking or body tracking to overlay virtual masks, filters, or accessories.	**60**

	12. Build an AR navigation prototype using anchored waypoints and directional indicators for indoor guidance. 13. Create a VR environment with physics-based interactions such as grabbing, throwing, and colliding objects. 14. Design a VR user interface (VR canvas) with interactive buttons, sliders, and menus using XR Interaction Toolkit. 15. Develop a VR locomotion system (ray-based teleportation or joystick-based movement) without requiring a VR headset. 16. Implement object snapping, scaling, and rotation interactions in VR for assembly or puzzle-type tasks. 17. Build an AR shopping/preview application where users place and resize virtual furniture in real-world space. 18. Develop a VR simulation prototype with environmental effects such as fog, lighting transitions, and spatial audio. 19. Create a mixed-reality storytelling experience combining AR world elements and VR-style UI interactions. 20. Create an immersive environment (living room/ battlefield/ tennis court) with only static game objects. 3D game objects can be created using Blender or use available 3D	
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**TEXTBOOKS:** 1. William R Sherman and Alan B Craig, "Understanding Virtual Reality, Interface, Application and Design",(The Morgan Kaufmann Series in Computer Graphics)", Morgan Kaufmann Publishers, San Francisco, CA, 2002, ISBN-13: 978-1558603530. 2. Schmalstieg / Hollerer, "Augmented Reality: Principles & Practice" by Pearson Education India; First edition (12 October 2016), ISBN-10: 9332578494. 3. Sanni Siltanen, Theory and applications of marker-based augmented reality. Julkaisija – Utgivare Publisher.2012. ISBN 978-951-38-7449-0 **REFERENCE BOOKS:** 1. Burdea, Grigore C and Philippe Coiffet, "Virtual Reality Technology", Wiley Inter science, India, 2003. 2. Alan B Craig, William R Sherman, Jeffrey D Will, "Developing Virtual Reality Applications: Foundations of Effective Design", Morgan Kaufmann Publishers, 2009. 3. Alan B. Craig, Understanding Augmented Reality, Concepts and Applications, Morgan Kaufmann, 2013.	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Build functioning AR applications capable of plane detection, object placement, and basic interactive features. **CO2:** Create VR prototypes using Unity's XR tools, demonstrating interaction techniques like grabbing, teleportation, and UI interaction.	

	CO3: Design and integrate 3D models, materials, lighting, audio, and effects to enhance the quality and realism of immersive experiences. **CO4:** Deploy AR/VR applications on mobile and simulator-based platforms and evaluate their performance and usability.

Name of the Programme : **B.E. in Electronics and Computer Science**
Course Code : **ECS-311**
Title of the Course : **Computer Networks**
Number of Credits : **03**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Mathematics, Basic computer fundamentals	
Course Objectives:	**The course aims to provide the student with**: 1. An introduction to the concept of OSI model, TCP/IP, identifying different network topologies and protocols. 2. An understanding of Data Link layer protocols and technologies. 3. Application of fundamental concepts of network layer operations, routing algorithms, and Internet protocols (IPv4/IPv6), including addressing, subnetting, VLANs, NAT. 4. A comprehensive understanding of transport-layer services and mechanisms, along with key application-layer protocols and architectures such as DNS, E-mail, and the World Wide Web	
Contents:		**No of Hours**
UNIT- 1	Reference Models: Layered architecture of OSI Model, TCP/IP architecture. **Line coding:** NRZ, RZ, Biphase (Manchester and Differential Manchester), AMI, HDB3, B8ZS, 2BIQ, 8B6T. **LAN**: **LAN topologies**: Bus, Ring, Tree, Star, Ethernet: Types, Contention Access, CSMA, CSMA/CD, Token ring, fast ethernet The Physical Layer: Interface RS-232, DTE-DCE interface.	**10**
UNIT-2	**Data Link Layer:** Frame design consideration, flow control, error control (stop- and-wait mechanism, sliding window), sequence numbering of piggybacking acknowledgement. **Data Link protocols:** BISYNC- transmission frames, protocol operation, HDLC- flow and error control, framing, transparency, protocol operation, Comparison of BISYNC and HDLC. **Switching:** Switching networks, circuit switching, space division switching, Time division switching, packet switching (datagram and virtual circuit –SVC and PVC)	**12**
UNIT -3	**Networking Devices:** Repeaters, Bridges, Hub, Switch, Routers, Gateways, Firewall. **Network Layer:** Services, virtual circuits and datagram subnet, routing algorithms (shortest path, flooding, flow based, distance vector, link state), congestion control, choke packets, load	**11**

	shedding, jitter control, flow specifications, traffic shaping (leaky bucket and token bucket algorithm) **Internet Protocols:** IPv4 (Classful and Classless Addressing) Sub-netting, VLAN, NAT, IPv6, Address Mapping: ARP, DHCP, ICMP	
UNIT -4	**Transport Protocols:** Transport service: services provided to the upper layer, connection establishment, connection release, multiplexing, flow control and buffering **Application Layer:** Domain Name System: DNS Name Space, Resource Records, Name Servers; Electronic Mail: Architecture and Services, The User Agent, Message Formats, Message Transfer, Final Delivery; World Wide Web: Architectural Overview, Client and Server Side, URLs, Cookies; HTTP: Connections, methods, message headers, caching	**12**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**TEXTBOOKS:** 1. Andrew S Tanenbaum, Nick Feamster, David J Wetherall, Computer Networks, 6th Edition, Pearson, 2021, ISBN: 9780136681557 2. Behrouz A Forouzan, Data Communications and Networking, 5th Edition, McGraw Hill Education, 2017, ISBN: 9781259096468 3. Prakash C Gupta, Data Communications and Computer Networks, 2nd Edition, PHI Learning, 2014, ISBN: 9788120348646 **REFERENCE BOOKS:** 1. Achyut S Godbole, Data Communications and Networks, 2nd Edition, McGraw Hill Education, 2017, ISBN: 9780071077705 2. William Stallings, Data and Computer Communications, 10th Edition, Pearson, 2013, ISBN: 9780132126953	
Course Outcomes:	**Upon completion of the course, students will be able to:** CO 1. Explain layered communication models, data transmission techniques, line coding schemes, LAN architectures, and physical layer interfaces. CO 2. Explain the transport-layer services including connection management and multiplexing, and describe application-layer protocols such as DNS, Email, WWW, and HTTP. CO 3. Analyze data link layer protocols, framing, flow/error control mechanisms, frame sequencing, and compare switching techniques including circuit, packet, and message switching. CO 4. Evaluate the functions of networking devices, apply routing algorithms and demonstrate understanding of IPv4/IPv6 addressing and address mapping protocols.	

Name of the Programme : **B.E. in Electronics and Computer Science**
Course Code : **ECS-312**
Title of the Course : **Computer Networks Lab**
Number of Credits : **01**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Fundamental Networking Concepts and Programming Skills	
Course Objectives:	**The course aims to provide the student with:** 1. An understanding of the various line coding schemes in communication networks. 2. An understanding of the working principle of various communication protocols with respect to the OSI model. 3. Analysis of the various data communication algorithms. 4. An understanding of the concept of data transfer between nodes in a network.	
Contents:	**List of Programs /Experiments**	**No of Hours**
	Line Encoding (Using OCTAVE/PYTHON/MATLAB) 1. Implement Unipolar NRZ & NRZ-L and NRZ-I encoding techniques 2. Implement RZ and Bi-phase (Manchester and Differential Manchester) encoding techniques 3. Implement Bipolar Encoding schemes (AMI, Pseudo ternary, B8ZS and HDB3) **Networking Configuration: (Using Network Simulator/Cisco Packet Tracer)** 4. Setting up a network to study various Topologies 5. Configuration of LAN 6. Configuring Networking and Internetworking Devices such as Hub, Switch, Router, Bridge 7. Inter-Network Data Exchange Setup 8. VLAN Creation and Verification and Switch Connection for VLAN Expansion 9. Configuration of Routing Information protocol 10. Configuration of Open shortest Path First (OSPF) Algorithm 11. Network Configuration with Static & Default Routing	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, Collaborative, experiential, and problem-solving approaches.	
Instructions:	List of experiments to be performed from above list.	
References/ Readings:	**TEXTBOOKS:** 1. Andrew S Tanenbaum, Nick Feamster, David J Wetherall, Computer Networks, 6th Edition, Pearson, 2021, ISBN: 9780136681557	

	2. Behrouz A Forouzan, Data Communications and Networking, 5th Edition, McGraw Hill Education, 2017, ISBN: 9781259096468 3. Prakash C Gupta, Data Communications and Computer Networks, 2nd Edition, PHI Learning, 2014, ISBN: 9788120348646 **REFERENCE BOOKS:** 1. Achyut S Godbole, Data Communications and Networks, 2nd Edition, McGraw Hill Education, 2017, ISBN: 9780071077705 2. William Stallings, Data and Computer Communications, 10th Edition, Pearson, 2013, ISBN: 9780132126953
Course Outcomes:	**After going through this course, the students will be able to**: CO 1. Analyse various data communication protocols CO 2. Implement various topologies in a computer network CO 3. Implement line coding techniques for computer networks CO 4. Configure networking devices to set up communication between the same.

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ECM-310**
Title of the Course : **Design and Analysis of Algorithms**
Number of Credits : **3**
Effective from AY : **2024-2025**

Pre- requisites for the Course:	Basic knowledge of programming, understanding of fundamental data structures.	
Course Objectives:	**This course will enable students to:** 1. Understand the fundamental concepts of algorithm design and performance analysis. 2. Apply mathematical methods to analyze algorithm efficiency using asymptotic notations and recurrence relations. 3. Develop efficient algorithms using design paradigms such as divide and conquer, greedy, and dynamic programming. 4. Explore advanced problem-solving techniques including backtracking, branch and bound, and handling NP-hard problems.	
Contents:		**No. of Hours**
UNIT-I	**Algorithm Analysis & Complexity:** Algorithm Definition and Specification, Performance analysis (Space complexity, Time complexity, Asymptotic Notations), Solving Recurrence – Iteration, recursion tree and master method. **Divide and Conquer:** General method, Binary Search, Merge sort, Quick sort, Finding Min-Max, Finding kth smallest element, Strassen's matrix multiplication	**12**
UNIT-2	**Greedy Method:** General Method, Knapsack Problem, Minimum cost Spanning tree, Single source shortest path. **Dynamic Programming:** General Method, Multistage Graphs, All pair shortest paths, Single source shortest path with General weights, Optimal Binary Search Tree, 0/1 knapsack problem, Travelling salesperson problem.	**12**
UNIT-3	**Backtracking:** General Method, 8-queens problem, Sum of subsets problem, graph colouring, Hamiltonian Cycles, knapsack problem. **Branch-and-Bound:** General Method, 0/1 knapsack, Travelling salesperson problem	**11**
UNIT-4	**String and Pattern Matching Algorithms**: Brute Force, KMP, Boyer Moore, Tries. **Text compression:** Huffman Coding. **Text similarity testing:** LCS.	**10**

	NP-Hard and NP-Complete Problems: Basic concepts, NP-Hard Graph Problems: Clique Decision Problem, Randomized, Probabilistic and Approximation Algorithms.	
Pedagogy:	Interactive, reflective, and inquiry-based learning with an emphasis on problem-solving, algorithm tracing, and case-based analysis.	
References/ Readings:	**TEXTBOOKS:** 1. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, 2nd Edition, Galgotia Publications, 2008, ISBN: 978-8175152571. 2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, 3rd Edition, McGraw-Hill, 2009, ISBN: 978-0262033848. **REFERENCE BOOKS:** 1. Gilles Brassard, Paul Bratley, Fundamentals of Algorithmics, Prentice Hall India, 1996, ISBN: 9780133350689 2. Michael T Goodrich, Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, 3rd Edition, Wiley, 2013, ISBN: 978-1118033161 3. Jon Kleinberg, Éva Tardos, Algorithm Design, 1st Edition, Pearson Education, 2005, ISBN: 9780321295354	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Explain the fundamentals of algorithm analysis and various complexity measures. **CO2:** Apply divide and conquer, greedy, and dynamic programming approaches to solve computational problems. **CO3:** Analyze and evaluate algorithmic solutions for optimization using backtracking and branch-and-bound methods. **CO4:** Design algorithms for NP-hard problems using heuristic, randomized, or approximation.	

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ECM-311**
Title of the Course : **Design and Analysis of Algorithms Lab**
Number of Credits : **1**
Effective from AY : **2024-2025**

Pre- requisites for the Course:	Basic knowledge of programming and fundamental of data structures,	
Course Objectives:	**This course will enable students to:** 1. Implement and analyze basic algorithms. 2. Apply various algorithm design paradigms such as Divide and Conquer, Greedy, Dynamic Programming, Backtracking, and Branch and Bound. 3. Compare algorithmic approaches based on computational efficiency and problem characteristics. 4. Develop programming skills to design and evaluate algorithms for complex computational problems.	
Contents:	**List of Programs /Experiments**	**No. of Hours**
	1. Write a program to implement binary search using divide and conquer. 2. Write a program to implement merge sort using divide and conquer. 3. Write a program to implement quick sort using divide and conquer. 4. Write a program to implement minimum cost spanning trees using the greedy approach. 5. Write a program to implement single source shortest path algorithm using the greedy approach. 6. Write a program to implement 0/1 knapsack problem using dynamic programming. 7. Write a program to implement optimal binary search tree (OBST) using dynamic programming. 8. Write a program to implement single source shortest path algorithm using dynamic programming. 9. Write a program to implement sum of subset problem using backtracking. 10. Write a program to implement graph colouring problem using backtracking. 11. Write a program to implement pattern matching algorithms. 12. Write a program to implement text compression and text similarity testing.	30
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	

References/ Readings:	**TEXTBOOKS** 1. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, 2nd Edition, Galgotia Publications, 2008, ISBN: 978-8175152571. 2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, 3rd Edition, McGraw-Hill, 2009, ISBN: 978-0262033848. **REFERENCE BOOKS:** 1. Gilles Brassard, Paul Bratley, Fundamentals of Algorithmics, Prentice Hall India, 1996, ISBN: 9780133350689 2. Michael T Goodrich, Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, 3rd Edition, Wiley, 2013, ISBN: 978-1118033161 3. Jon Kleinberg, Éva Tardos, Algorithm Design, 1st Edition, Pearson Education, 2005, ISBN: 978-0321295354.
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Implement and test basic algorithms using different design techniques in a programming environment. **CO2:** Apply divide and conquer, greedy, and dynamic programming paradigms to solve computational problems experimentally. **CO3:** Evaluate and compare algorithm performance using time and space complexity measures. **CO4:** Design and implement heuristic, randomized, or approximation-based algorithmic solutions.

Name of the Program: B.E. in Electronics & Computer Engineering

Course Code : ECM 327

Title of the Course : Soft Computing Techniques

Number of Credits : 3

Effective from AY : 2024-2025

Pre-requisites for the Course:	Knowledge of Applied Mathematics. Algorithms. Programming	
Course Objectives:	**This course aims to provide the students with:** 1. Understanding of Neural Networks and its training methodologies. 2. Ability to explain Fuzzy Logic and Fuzzy Inference Systems. 3. Ability to write Genetic Algorithms and Evolutionary Algorithms. 4. In-sight into Deep Learning, Expert Systems & Hybrid Systems.	
Content:		**No of hours**
UNIT - 1	**Introduction to Soft Computing:** Soft Computing versus Hard Computing, Soft-Computing Techniques: Artificial Neural Networks, Fuzzy Systems, Evolutionary Algorithm, Hybrid Systems. Types of Problems: Classification, Functional Approximation, Optimization. **Introduction to Neural Networks:** Mc-Culloch Pitts neuron model, Activation functions & basic gates. Training algorithms-Hebbian learning rule, perceptron learning rule, Delta learning rule, Widrow-Hoff learning rule and related problems. **Multilayer Networks:** Error Back propagation training algorithm	**12**
UNIT - 2	**Fuzzy Logic:** Introduction, Classical Set Theory (Crisp Set): Operations & Properties, Fuzzy Set Theory: Operations & Properties, Membership Functions and types, Fuzzy v/s Crisp Sets, Classical relations (Cartesian product) and Fuzzy relations: Cardinality, Operations, Properties and Composition. Crisp Logic vs Fuzzy logic, Fuzzy logic operations: AND, OR, NOT, Implication, Aggregation and Defuzzification, Lambda-cuts or for fuzzy sets & relations, Defuzzification methods, The Fuzzy Process.	**11**
UNIT -3	**Genetic Algorithms:** Concept, Solution, Initial Population, Genetic Operators, Fitness Function, Stopping Condition. Fitness Scaling, Selection, Mutation, Crossover. **Other Evolutionary Algorithms:** Particle Swarm Optimization, Ant Colony Optimization.	**11**

UNIT - 4	**Hybrid Systems:** Sequential , Auxiliary and Embedded Hybrid Systems, Types of Hybrid Systems: Neuro-Fuzzy, Neuro-Genetic, Fuzzy Genetic Hybrid Systems- Advantages and Applications **Deep Neural Networks:** What is deep learning? Convolutional neural networks: Architecture, Convolution operation, Motivation and Pooling. **Expert Systems:** Expert System Architecture.	**11**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**TEXT BOOKS:** 1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, ISBN: 978-0262035613 2. Kishan Mehrotra, Chilukuri Mohan, Sanjay Ranka, Elements of Artificial Neural Network, Penram Publications, ISBN: 978-8187972092 3. Anupam Shukla, Ritu Tiwari, Rahul Kala, Real Life Applications of Soft Computing, CRC Press, 2010, ISBN: 978-1439801911 4. S N Sivanandan, S N Deepa, Principles of Soft Computing, 3rd Edition, Wiley India, ISBN: 978-8126518074 5. J Zurada, Introduction to Artificial Neural Network, 1st Edition, Jaico Publications, 2012, ISBN: 978-8179929025	
	REFERENCE BOOKS: 1. Charu C Aggarwal, Neural Networks and Deep Learning, Springer, 1st Edition, 2018. 2. Rajasekaran, G A Vijayalakshmi Pai, Neural Networks, Fuzzy Logic and Genetic Algorithm, PHI Learning Pvt Ltd, 1st Edition, 2013, ISBN-10: 8120321863. 3. Josh Patterson, Adam Gibson, Deep Learning: A Practitioner's Approach, O'Reilly Media, 1st Edition, 2017. 4. Timothy J Ross, Fuzzy Logic with Engineering Applications, 3rd Edition, Wiley India, 3rd Edition, 2010, ISBN: 978-8126531264.	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Explain different types of Soft Computing techniques, Expert Systems and their applications. **CO2:** Apply Evolutionary Algorithms to optimization problems. **CO3:** Design Neural Networks for logic functions and understand deep neural networks and its applications. **CO4:** Design Fuzzy Inference Systems to solve Real-Life Problems.	

Name of the Programme : **B.E. in Electronics & Computer Engineering**
Course Code : **ECM-328**
Title of the Course : **Soft Computing Lab**
Number of Credits : **1**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Basics of Mathematics and Python or MATLAB.	
Course Objectives:	**This course will enable the students to:** 1. Develop proficiency in implementing various neural network models and training algorithms. 2. Develop fuzzy inference systems for real world examples. 3. Apply genetic algorithm techniques to efficiently solve optimization problems. 4. Competent in implementing deep learning models such as CNN & auto encoders.	
Content:	**List of Experiments/Programs**	**No of Hours**
	1. Implement various activation functions for neural networks. 2. Implement Perceptron Network for basic logic Gates. 3. Implement various training algorithms such as Hebbian learning rule, Perceptron learning rule, Delta learning rule, Widrow-Hoff learning rule. 4. Implement Error backpropagation Network. 5. Design and implement various fuzzy membership functions (Triangular, Trapezoidal, Gaussian) for fuzzy sets. 6. Implementation of basic Fuzzy Logic Operations. 7. Develop a Fuzzy Inference System (FIS) for a given problem. 8. Implement simple genetic algorithm to solve an optimization problem. 9. Implement Particle Swarm Optimization Algorithm/Ant Colony Optimization Algorithm to solve a simple optimization problem. 10. Implement a Convolutional Neural Network or Basic Autoencoder.	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**TEXT BOOKS:** 1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 1st Edition, 2016, ISBN: 978-0262035613 2. Kishan Mehrotra, Chilukuri Mohan, Sanjay Ranka, Elements of	

	Artificial Neural Networks, Penram Publications, 1st Edition, 1996, ISBN: 978-8187972092 3. Anupam Shukla, Ritu Tiwari, Rahul Kala, Real Life Applications of Soft Computing, CRC Press, Special Indian Edition, 2010, ISBN: 978-1439801911 4. S N Sivanandan, S N Deepa, Principles of Soft Computing, 3rd Edition, Wiley India, 2013, ISBN: 978-8126518074 5. J Zurada, Introduction to Artificial Neural Networks, 1st Edition, Jaico Publications, 2012, ISBN: 978-8179929025 **REFERENCE BOOKS:** 1. Charu C Aggarwal, Neural Networks and Deep Learning, Springer, 2018, ISBN: 978-3319944623 2. Rajasekaran, G A Vijayalakshmi Pai, Neural Networks, Fuzzy Logic and Genetic Algorithm, PHI Learning Pvt Ltd, June 2013, ISBN: 978-8120341224 3. Josh Patterson, Adam Gibson, Deep Learning: A Practitioner's Approach, O'Reilly Media, 2017, ISBN: 978-1491914250 4. Timothy J Ross, Fuzzy Logic with Engineering Applications, 3rd Edition, Wiley India, 2010, ISBN: 978-0470743768
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Analyse and apply genetic algorithm techniques for solving optimization problems efficiently. **CO2:** Implement neural network models and training algorithms. **CO3:** Develop fuzzy inference systems for real-world decision-making tasks. **CO4:** Implement advanced soft computing techniques like CNN & auto encoders.

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course code : **ECM -329**
Title of the course : **Theory of Computations**
Number of Credits : **3**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Knowledge of Discrete Mathematics, Data structures, Logic Design	
Course Objectives:	**This course will enable students to:** 1. Understand the fundamental concepts of automata theory, formal languages, and computational models. 2. Analyze and construct finite automata, regular expressions, and regular grammars for pattern recognition problems. 3. Design and evaluate context-free grammars, pushdown automata, and their applications to language processing. 4. Explore Turing machines, linear-bounded automata, and concepts of computability, undecidability, and reducibility.	
Contents:		**No of Hours**
UNIT-1	**Basic Terminology:** Basics of String, Set Theory, Relation on Set, Graph & Tree, Digital Circuit, Basics of Automata Theory and Theory of Computation, History of Automata Theory, Use of Automata. Uses of Automata, Characteristics of Automaton, Finite Automata (FA), Graphical and Tabular Representation of FA, Transitional System, Deterministic finite automata (DFA) & Non-deterministic finite automata (NFA), Conversion of an NFA to a DFA, Equivalence of DFA and NFA, Dead State, Minimization of Finite Automata, Myhill–Nerode Theorem, Two-way Finite Automata, Applications & Limitations of Finite Automata.	**10**
UNIT-2	Finite Automata with Output, Conversion of One Machine to Another. (**Language and Grammar:** Grammar, The Chomsky Hierarchy. **Regular Expression:** Basics, Operations, Identities, The Arden's Theorem, Construction of Finite Automata from a Regular Expression, Equivalence of Two Finite Automata, Equivalence of Two Regular Expressions, Construction of Regular Grammar from an RE, Constructing FA from Regular Grammar. Pumping Lemma for Regular Expression, Applications of Regular Expression.	**12**
UNIT-3	**Context-free Grammar:** Definition, Derivation and Parse Tree, Ambiguity, Left Recursion and Left Factoring, Simplification, Linear Grammar, Normal Form, Closure Properties and Pumping Lemma for CFL, Applications of CFL. **Pushdown Automata:** Basics, Acceptance, Deterministic & Non-	**12**

	deterministic pushdown automata (DPDA and NPDA), Construction of PDA from CFG, Construction of CFG Equivalent to PDA, Graphical Notation for PDA, Two-stack PDA.	
UNIT-4	**Turing Machine:** Basics, Transitional Representation, Non-deterministic Turing Machine, Conversion of Regular Expression to Turing Machine, Two-stack PDA and Turing Machine, Variations of the Turing Machine, Turing Machine as an Integer Function, Universal Turing Machine, Linear-Bounded Automata (LBA), Post Machine, Church's Thesis. **Computability and Undecidability:** TM Languages, Unrestricted Grammar, Modified Chomsky Hierarchy, Properties of Recursive and Recursively Enumerable Language, Undecidability, Reducibility.	**11**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**TEXTBOOKS:** 1. John C Martin, Introduction to Languages and the Theory of Computation, 4th Edition, Tata McGraw-Hill, 2011, ISBN: 978-0070684553 2. John E Hopcroft, Rajeev Motwani, Jeffrey D Ullman, Introduction to Automata Theory, Languages and Computation, 3rd Edition, Narosa Publishing House (Indian Edition), 2007, ISBN: 978-8173195678 3. Shyamalendu Kandar, Introduction to Automata Theory, Formal Languages and Computation, 1st Edition, Pearson, 2013, ISBN: 978-9332522055 **REFERENCE BOOKS:** 1. K L P Mishra, N Chandrasekaran, Theory of Computer Science (Automata, Languages and Computation), 3rd Edition, 2006, PHI Learning Pvt Ltd, ISBN: 978-8120331325 2. Michael Sipser, Introduction to the Theory of Computation, 3rd Edition, 2012, Cengage Learning India Pvt Ltd, ISBN: 978-8131525293Elaine Rich, Automata, Computability and Complexity, Indian Edition, Pearson Education India, 2009, ISBN: 978-8131715045.	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Explain the basic concepts of deterministic and non-deterministic finite automata. **CO2:** Analyze the formal relationships among machines, languages and grammars **CO3:** Design a finite automaton to recognize a given regular language. **CO4:** Construct minimalist automata for recursively enumerable languages	

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course code : **ECM -330**
Title of the course : **Theory of Computations Lab**
Number of Credits : **1**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Knowledge of Discrete Mathematics, Data structures, Logic Design, Computer programming	
Course Objectives:	**This course will enable students to:** 1. Understand the fundamental concepts of automata, formal languages, and grammars. 2. Design and simulate Finite Automata, Pushdown Automata, and Turing Machines for solving computational problems. 3. Analyze language recognition and computational processes using appropriate automata models. 4. Develop analytical and problem-solving skills related to computation theory and formal language processing.	
Contents:	**List of Programs /Experiments**	**No of Hours**
	1. Design a Finite State Machine (FSM) that accepts all strings over input symbols {0, 1} having three consecutive 1's as a substring. 2. Design a Finite State Machine (FSM) that accepts all strings over input symbols {0, 1} which are divisible by 3. 3. Design a Finite State Machine (FSM) that accepts all decimal string which are divisible by 3. 4. Design a Push Down Automata (PDA) that accepts all string having equal number of 0's and 1's over input symbol {0, 1} for a language 0^n1^n where n >= 1. 5. Design a Program to create PDA machine that accept the well-formed parenthesis. 6. Design a PDA to accept WCW^R where w is any binary string and W^R is reverse of that string and C is a special symbol. 7. Design a Turing Machine that calculate 2's complement of given binary string. 8. Design a Turing Machine which will increment the given binary number by 1. 9. Design a Turing Machine that's accepts the following language $a^nb^nc^n$ where n>0. 10. Design a Turing Machine to accept W^R where w is any binary string and W^R is reverse of that string.	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based,	

	collaborative, experiential, and problem-solving approaches.
References/ Readings:	**TEXTBOOKS:** 1. John C Martin, Introduction to Languages and the Theory of Computation, 4th Edition, 2011, Tata McGraw-Hill, ISBN: 978-0070684553 (Latest edition available) 2. John E Hopcroft, Rajeev Motwani, Jeffrey D Ullman, Introduction to Automata Theory, Languages and Computation, 3rd Edition, 2007, Narosa Publishing House (Indian Edition), ISBN: 978-8173195678 (Latest edition available) 3. Shyamalendu Kandar, Introduction to Automata Theory, Formal Languages and Computation, 1st Edition, 2013, Pearson, ISBN: 978-9332522055 (No newer edition available) **REFERENCE BOOKS:** 1. Michael Sipser, Introduction to the Theory of Computation, Indian Edition, Cengage Learning India Pvt. Ltd., 2013, ISBN: 978-8131525293. 2. K.L.P. Mishra and N. Chandrasekaran, Theory of Computer Science (Automata, Languages and Computation), 3rd Edition, PHI Learning Pvt. Ltd., New Delhi, 2006, ISBN: 978-8120331325
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Design finite automata (DFA/NFA) for a given formal languages and identify their acceptance criteria. **CO2:** Construct and simulate a pushdown automata (PDA) for context-free languages. **CO3:** Design and analyze a Turing Machines for various computational problems. **CO4:** Design and Analyze different computational models (FA, PDA, TM) to language classes in the Chomsky hierarchy.

Name of the Programme : **B.E. in Electronics and Computer Engineering**
Course Code : **ECM331**
Title of the Course : **Software Engineering and Testing**
Number of Credits : **3**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Knowledge of Programming, Data Structures and Algorithms	
Course Objectives:	**This course will enable students to:** 1. Understand software engineering principles for building reliable systems. 2. Acquire knowledge of software development methodologies from requirements analysis to deployment. 3. Gain insight into software testing techniques and tools. 4. Develop project management skills for effective and successful software development.	
Contents:		**No of Hours**
UNIT- 1	**Software Life-Cycle Models:** Code-and-Fix Life-Cycle Model, Waterfall Life- Cycle Model, Rapid-Prototyping Life-Cycle Model, Open Source Life-Cycle Model, Agile Processes, Synchronize-and-Stabilize Life-Cycle Model, Spiral Life- Cycle Model. **The Tools of the Trade:** CASE, Taxonomy of CASE, Scope of CASE. **More on UML:** Class Diagrams, Use-Case Diagrams, Interaction Diagrams, State Charts, Activity Diagrams. **Requirements:** Determining what the Client Needs, Overview of the Requirements Workflow, Understanding the domain, The Business Model, Initial Requirements, Metrics and Challenges for Requirement Workflow.	**12**
UNIT-2	**Testing:** Non-Execution-Based Testing, Execution-Based Testing, Planning and Estimating: Planning and the Software Process, Estimating Duration and Cost. **Testing Techniques:** Test Case Selection, Black-Box Unit-Testing Techniques, Glass-Box Unit-Testing Techniques, Code Walkthroughs and Inspections, Integration Testing, Product Testing, Acceptance Testing. **Post-delivery Maintenance:** Development and Maintenance, Management of Post-delivery Maintenance. Reverse Engineering, Testing during Post-delivery Maintenance, Metrics	**11**

	and Challenges for Post-delivery Maintenance.	
UNIT -3	**Defect Prevention and Process improvement:** Basic concepts and generic approaches, Root cause analysis for defect prevention, Training for defect prevention, Defect prevention techniques. **Control Flow, Data Dependency:** Basic Control Flow Testing, Loop Testing, CFT Usage, and Other Issues, Data Dependency and Data Flow Testing.	**11**
UNIT -4	**Software testing tools and overview:** Need for automated testing tools, Taxonomy of testing tools, Functional/Regression testing tools, Performance testing tools, Testing management tools, Source code testing tools, Selection of testing tools. **Case study:** Overview of Win Runner, Silk Test, SQA Robot, Load runner.	**11**
Pedagogy:	Inquiry based learning, Integrative, Reflective Learning, Constructive learning and Collaborative learning	
References/ Readings:	**TEXTBOOKS:** 1. Jeff Tian, Software Quality Engineering Testing Quality Assurance and Quantifiable Improvement, 2006, Wiley, ISBN -9780471713462 2. K V K K Prasad, Software Testing Tools, Dreamtech Press India Pvt Ltd, 2004, ISBN -9788177226140 3. Stephen R. Schach, Object-Oriented and Classical Software Engineering, 8th Edition,2010, Tata McGraw Hill ISBN 9780073376189. **REFERENCE BOOKS:** 1. Kshirasagar Naik ,Priyadarshi Tripathy, Software Testing and Quality Assurance Theory and Practice, Wiley, 2018, ISBN -9781119493761 2. Roger Pressman Software Engineering A Practitioner's Approach 7th Edition , McGraw Hill, 2019, ISBN- 9781259872976	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Understand different software testing methods and tools **CO2:** Apply proper test plans and procedures in software development. **CO3:** Design quality software systems that are modular and reusable **CO4:** Develop, test, and maintain reliable software systems.	

Name of the Programme : **Electronics and Computer Engineering**
Course Code : **ECM332**
Title of the Course : **Software Engineering and Testing Lab**
Number of Credits : **1**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Knowledge of Programming, Data Structures and Algorithms	
Course Objectives:	**This course will enable students to:** 1. Understand software engineering principles for building reliable systems. 2. Develop software solutions for specific requirements and deploy them effectively. 3. Gain insight into software testing techniques and tools. 4. Acquire project management skills for successful software development.	
Contents:	**List of Experiments/Programs**	**No of Hours**
	1. **Software Life Cycle Models** *Objective:* Compare Waterfall, Spiral, Agile, and Prototype models based on their features and applications. 2. **Preparation of Software Requirement Specification (SRS)** *Objective:* Prepare an SRS document for a mini project (e.g., Library or Hostel Management System). 3. **Drawing UML Diagrams using a CASE Tool** *Objective:* Create Use Case, Class, and Activity diagrams for the selected project using a CASE tool like StarUML or Draw.io. 4. **Software Project Planning and Estimation** *Objective:* Prepare a project plan including timeline, cost, and resource estimation using basic techniques (e.g., LOC or Function Point). 5. **Preparation of Test Plan and Test Case Design** *Objective:* Design a sample test plan and prepare test cases using black-box and boundary value analysis. 6. **Code Walkthrough and Inspection (Document-Based)** *Objective:* Review a sample program or pseudocode and identify possible logical or design defects. 7. **Software Quality Metrics Study** *Objective:* Identify and explain important software	**30**

	quality metrics such as reliability, maintainability, and efficiency. 8. **Study of Software Testing Tools (Demonstration)** *Objective:* Observe and document the working of tools like Selenium, JMeter, or LoadRunner (theoretical demo or video-based). 9. **Defect Prevention and Root Cause Analysis (RCA)** *Objective:* Perform RCA for a sample software defect and suggest preventive actions. 10. **Software Maintenance and Change Management** *Objective:* Prepare a maintenance plan describing types of maintenance and change control process.	
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**TEXTBOOKS:** 1. Jeff Tian, Software Quality Engineering Testing Quality Assurance and Quantifiable Improvement, 2006, Wiley, ISBN -9780471713462 2. K V K K Prasad, Software Testing Tools, Dreamtech Press India Pvt Ltd, 2004, ISBN -9788177226140 3. Stephen R. Schach, Object-Oriented and Classical Software Engineering, 8th Edition,2010, Tata McGraw Hill ISBN 9780073376189. **REFERENCE BOOKS:** 1. Kshirasagar Naik, Priyadarshi Tripathy, Software Testing and Quality Assurance Theory and Practice, Wiley, 2018, ISBN -9781119493761 2. Roger Pressman Software Engineering A Practitioner's Approach 7th Edition , McGraw Hill, 2019, ISBN- 9781259872976	
Course Outcomes:	**After going through this course, the students will be able to:** **CO1:** Apply software engineering concepts and life-cycle models. **CO2:** Design software systems using suitable methods and standards. **CO3:** Use testing tools and techniques to ensure software quality. **CO4:** Plan and manage software projects effectively.	