CMR TECHNICAL CAMPUS
UGC AUTONOMOUS
M. Tech Syllabus (w. e. f. A.Y. 2025-26)

EMBEDDED SYSTEMS

I SEMESTER

Course Code	Course Title	L	T	P	Credits
25ES101PC	FPGA Based System Design	3	0	0	3
25ES102PC	Embedded System Design	3	0	0	3
25ES111PE 25ES112PE 25ES113PE	1. CMOS VLSI Design 2. Wireless Sensor Networks 3. Advanced Computer Architecture	3	0	0	3
25ES121PE 25ES122PE 25ES123PE	1. Machine Learning and Deep Learning 2. Advanced RISC Architectures 3. Automotive Embedded Systems	3	0	0	3
25ES103PC	FPGA Based System Design Laboratory	0	0	4	2
25ES104PC	Embedded System Design Laboratory	0	0	4	2
25ES105PC	Research Methodology & IPR	2	0	0	2
25ES---AC	Audit Course–I	2	0	0	0
	Total Credits	**16**	**0**	**8**	**18**

II SEMESTER

Course Code	Course Title	L	T	P	Credits
25ES201PC	Embedded Real Time Operating Systems	3	0	0	3
25ES202PC	IoT System Design	3	0	0	3
25ES231PE 25ES232PE 25ES233PE	1. GPU Architectures 2. VLSI Test and Testability 3. Hardware Software Co-Design	3	0	0	3
25ES241PE 25ES242PE 25ES243PE	1. Hardware Security in VLSI Design 2. Hardware Accelerators for Machine Learning Models 3. Image and Video Processing	3	0	0	3
25ES203PC	Embedded Real Time Operating System Laboratory	0	0	4	2
25ES204PC	IoT System Design Laboratory	0	0	4	2
25ES205PC	Mini Project with Seminar	0	0	4	2
25ES---AC	Audit Course-II	2	0	0	0
	Total Credits	**14**	**0**	**12**	**18**

III SEMESTER

Course Code	Course Title	L	T	P	Credits
25ES351PE 25ES352PE 25ES353PE	1. Machine Learning for Robotics 2. Edge Computing 3. Embedded Biomedical Applications	3	0	0	3
25ES311OE 25ES312OE 25ES313OE	1. Business Analytics 2. Operations Research 3. Industrial Safety	3	0	0	3
25ES301PR	Dissertation Work Review–II	0	0	18	6
	Total Credits	**6**	**0**	**18**	**12**

IV SEMESTER

Course Code	Course Title	L	T	P	Credits
25ES401PR	Dissertation Work Review – III	0	0	18	06
25ES402PR	Dissertation Viva-Voce	0	0	42	14
	Total	**0**	**0**	**60**	**20**

Audit Course I & II:

25ES001AC	English for Research Paper Writing
25ES002AC	Disaster Management
25ES003AC	Sanskrit for Technical Knowledge
25ES004AC	Value Education
25ES005AC	Constitution of India
25ES006AC	Pedagogy Studies
25ES007AC	Stress Management by Yoga
25ES008AC	Personality Development Through Life Enlightenment Skills

25ES101PC: FPGA BASED SYSTEM DESIGN (PC – I)

M.Tech. I Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Pre-Requisite: Switching Theory and Logic Design

Course Overview

This course introduces the principles and practices of FPGA-based system design, with a strong focus on digital logic implementation, system architecture, and modern design methodologies. Students will gain hands-on experience with hardware description languages, combinational and sequential logic design, and explore advanced topics such as platform FPGAs and multi-FPGA systems. The course equips learners with foundational knowledge and practical skills to architect, design, and optimize embedded digital systems using FPGAs. Suitable for postgraduate students in electronics, embedded systems, or VLSI design, this course bridges the gap between traditional digital logic design and modern reconfigurable computing.

Course Outcomes:

1. Explain the architectural features and functional components of FPGA-based systems and describe their advantages in digital design applications.
2. Design combinational logic circuits using Verilog HDL and evaluate them for delay, power, and resource utilization on FPGA platforms.
3. Develop and simulate sequential logic circuits and finite state machines using appropriate design styles and clocking rules.
4. Apply behavioral design methodologies and demonstrate system modeling using HDLs in real- world case studies.
5. Analyze and propose solutions for large-scale FPGA-based systems involving busses, platform FPGAs, and multi-FPGA architectures.

UNIT – I **8L**

Introduction to FPGA-Based Systems: Introduction, basic concepts, digital design and FPGAs, FPGA-based system design, FPGA architectures, SRAM-based FPGAs, permanently programmed FPGAs, chip I/O, circuit design of FPGA fabrics, architecture of FPGA fabrics.

UNIT - II **8L**

Combinational Logic Design in FPGAs: Introduction, the logic design process, hardware description languages -modeling with HDLs, Verilog, Combinational network delay, power and energy optimization, arithmetic logic, logic implementation for FPGAs, physical design for FPGAs, the logic design process revisited

UNIT - III **8L**

Sequential Logic and State Machines: Introduction, the sequential machine design process, sequential design styles, rules for clocking, performance analysis, power optimization

UNIT - IV **8L**

Design Methodologies and Behavioral Architecture: Introduction, behavioral design, design methodologies, design example

UNIT - V **8L**

System-Level Design and Advanced FPGA Applications: Introduction, buses, platform FPGAs, multi-FPGA systems, Novel architectures

TEXT BOOKS

1. Wolf, Wayne. *FPGA-Based System Design*. Pearson Education India, 2005.

REFERENCE BOOKS

1. Maxfield, Clive. *The Design Warrior's Guide to FPGAs: Devices, Tools and Flows*. Newnes, 1st ed., 2004.
2. Trimberger, Stephen M. *Field-Programmable Gate Array Technology*. Springer Science & Business Media, 2012.
3. Kuon, Ian, Russell Tessier, and Jonathan Rose. *FPGA Architecture: Survey and Challenges*. Now Publishers Inc., 2007

25ES102PC: EMBEDDED SYSTEM DESIGN (PC – II)

M.Tech. I Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course introduces the fundamental concepts and design methodologies of embedded systems. It begins by distinguishing embedded systems from general-purpose computing systems, highlighting their specialized nature and application-specific design. Students will explore the essential building blocks of embedded systems, including hardware components, microcontrollers, and interfacing techniques. Emphasis is placed on the development and role of embedded firmware, including how firmware interacts with hardware through APIs. By the end of the course, students will gain a foundational understanding required to design, develop, and analyze embedded systems used in modern electronic devices and real-time applications.

Course Outcomes: At the end of this course, students will be able to

1. Expected to differentiate the design requirements between General Purpose and Embedded Systems.
2. Expected to acquire the knowledge of firmware design principles.
3. Expected to understand the role of Real Time Operating System in Embedded Design.
4. To acquire the knowledge and experience of task level Communication in any Embedded System.

UNIT- I **8L**

Introduction to Embedded Systems: Definition of Embedded System, Embedded Systems Vs General Computing Systems, History of Embedded Systems, Classification, Major Application Areas, Purpose of Embedded Systems, Characteristics and Quality Attributes of Embedded Systems.

UNIT- II **8L**

Typical Embedded System: Core of the Embedded System: General Purpose and Domain Specific Processors, ASICs, PLDs, Commercial Off-The-Shelf Components (COTS), Memory: ROM, RAM, Memory according to the type of Interface, Memory Shadowing, Memory selection for Embedded Systems, Sensors and Actuators, Communication Interface: Onboard and External Communication Interfaces

UNIT- III **8L**

Embedded Firmware: Reset Circuit, Brown-out Protection Circuit, Oscillator Unit, Real Time Clock, Watchdog Timer, Embedded Firmware Design Approaches and Development Languages.

UNIT- IV **8L**

RTOS Based Embedded System Design: Operating System Basics, Types of Operating Systems, Tasks, Process and Threads, Multiprocessing and Multitasking, Task Scheduling.

UNIT- V **8L**

Task Communication: Shared Memory, Message Passing, Remote Procedure Call and Sockets, Task Synchronization: Task Communication/Synchronization Issues, Task Synchronization Techniques, Device Drivers, How to Choose an RTOS

TEXT BOOKS:

1. *Introduction to Embedded Systems*-Shibu K.V, McGrawHill.

REFERENCE BOOKS:

1. *Embedded Systems-Raj Kamal, TMH.*
2. *Embedded System Design-Frank Vahid, Tony Givargis, John Wiley.*
3. *Embedded Systems–Lyla, Pearson,2013*
4. *An Embedded Software Primer-David E. Simon, Pearson Education.*

25ES111PE:CMOS VLSI DESIGN (PE - I)

M.Tech. I Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course provides an in-depth study of CMOS (Complementary Metal-Oxide-Semiconductor) technology and its application in Very Large Scale Integration (VLSI) circuit design. It begins with the fundamentals of MOS device operation and transient response, laying the groundwork for designing efficient digital circuits. Students will explore the design methodologies of both combinational and sequential MOS logic circuits. The course also covers advanced topics such as dynamic logic techniques and memory cell design, enabling students to analyze and implement complex integrated circuits. By the end of the course, learners will be equipped with the theoretical knowledge and practical skills needed to design and optimize CMOS-based VLSI systems.

Course Objectives:

1. To understand the concepts of MOS Design and transient response
2. To know the design of combinational MOS logic circuits
3. To know the design of sequential MOS logic circuits
4. To understand the dynamic logic and also memory designing

Course Outcomes: Students will be able to:

1. Design of combinational MOS logic and sequential MOS logic circuits
2. Design of different Memories using MOS transistors
3. Design a circuits based on dynamic logic
4. Use CMOS transmission gates in various applications

UNIT – I **8L**

MOS Design

Pseudo NMOS logic- Inverter, Inverter threshold voltage, output high voltage, Output low voltage, gain at gate threshold voltage, transient response, rise time, fall time, pseudo NMOS logic gates, transistor equivalency, CMOS inverter logic.

UNIT - II **8L**

Combinational MOS logic circuits

MOS logic circuits with NMOS loads, Primitive CMOS logic gates- NOR and NAND gates, Complex logic circuits design- realizing Boolean expressions using NMOS gates and CMOS gates, AOI and OIA gates, CMOS full-adder, CMOS transmission gates, designing with transmission gates.

UNIT - III **8L**

Sequential MOS logic circuits: Behavior of bistable elements, SR Latch, Clocked Latch and Flip-flop circuits, CMOS D Latch and edge triggered flip-flop.

UNIT - IV **8L**

Dynamic Logic Circuits: Basic principle, Voltage Bootstrapping, Synchronous dynamic pass transistor circuits, Dynamic CMOS transmission gate logic, high performance dynamic CMOS circuits.

UNIT - V **8L**

Semiconductor Memories: Types, RAM array Organization, DRAM- types, operation, leakage currentsin DRAM cell and refresh operation, SRAM - operation, leakage currents in SRAM cells, Flash memory-NOR flash and NAND flash.

TEXT BOOKS:

1. *Digital Integrated Circuit Design- Ken Martin, Oxford University Press, 2011.*
2. *CMOS Digital Integrated Circuit Analysis and Design – Sung Mo Kang, Yusuf Leblebici, TMH, 3rd Ed., 2011.*

REFERENCE BOOKS:

3. *Introduction to VLSI Systems :A Logic, Circuit and System Perspective*-Ming BoL in, CRC Press, 2011.
4. *Digital Integrated Circuits: A Designs Perspective* -Jan M.Rabaey, Anantha Chandrakasan, Borivoje Nikolic, 2nd Ed., PHI.

25ES112PE: WIRELESS SENSOR NETWORKS (PE – I)

M.Tech. I Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course offers a thorough understanding of Wireless Sensor Networks (WSNs), focusing on their architecture, applications, and design challenges. Students will explore the fundamental concepts, including network structures, node architectures, and communication protocols. The course covers key topics such as routing and MAC protocols, data gathering, and data dissemination techniques essential for efficient sensor network operation. Emphasis is also placed on emerging technologies, hardware platforms, software tools, and design principles required for developing real-world WSN applications. By the end of the course, students will be equipped with the knowledge and skills to design, deploy, and manage wireless sensor networks across various domains.

Course Objectives

1. To acquire the knowledge about various architectures and applications of Sensor Networks
2. To understand issues, challenges and emerging technologies for wireless sensor networks
3. To learn about various routing protocols and MAC Protocols
4. To understand various data gathering and data dissemination methods
5. To Study about design principals, node architectures, hardware and software required for implementation of wireless sensor networks.

Course Outcomes: Upon completion of the course, the student will be able to:

1. Analyze and compare various architectures of Wireless Sensor Networks
2. Understand Design issues and challenges in wireless sensor networks
3. Analyze and compare various data gathering and data dissemination methods.
4. Design, Simulate and Compare the performance of various routing and MAC protocol

UNIT-I **8L**

Introduction to Sensor Networks, unique constraints and challenges, Advantage of Sensor Networks, Applications of Sensor Networks, Types of wireless sensor networks.

UNIT-II **8L**

Mobile Ad-hoc Networks (MANETs) and Wireless Sensor Networks, Enabling technologies for Wireless Sensor Networks. Issues and challenges in wireless sensor networks.

UNIT-III **8L**

Routing protocols, MAC protocols: Classification of MAC Protocols, S-MAC Protocol, B- MAC protocol, IEEE 802.15.4 standard and ZigBee.

UNIT-IV **8L**

Dissemination protocol for large sensor network. Data dissemination, data gathering, and data fusion; Quality of a sensor network; Real-time traffic support and security protocols.

UNIT-V **8L**

Design Principles for WSNs, Gateway Concepts Need for gateway, WSN to Internet Communication, and Internet to WSN Communication. Single-node architecture, Hardware components & design constraints, Operating systems and execution environments, introduction to TinyOS and nesC.

TEXTBOOKS:

1. *Ad-Hoc Wireless Sensor Networks*- C. Siva Ram Murthy,B. S. Manoj, Pearson
2. *Principles of Wireless Networks* – Kaveh Pah Laven and P. Krishna Murthy, 2002, PE

REFERENCEBOOKS:

1. *Wireless Digital Communications* – Kamilo Feher, 1999, PHI.
2. *Wireless Communications*-Andrea Goldsmith, 2005 Cambridge University Press.
3. *Mobile Cellular Communication* – Gottapu Sasi bhushana Rao, Pearson Education, 2012.
4. *Wireless Communication and Networking* – William Stallings, 2003, PHI.

25ES113PE: ADVANCED COMPUTER ARCHITECTURE (PE - I)

M.Tech. I Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course delves into the advanced concepts of computer system design and architecture. It begins with the fundamentals of computer design principles and progresses to explore instruction-level and thread-level parallelism. Key topics include pipeline processing, superscalar architectures, and performance optimization techniques. The course also addresses the design challenges and solutions related to interconnection networks in parallel computing environments. By the end of the course, students will gain a strong understanding of high-performance architecture design and the ability to analyze and evaluate complex computing systems.

Course Objectives:

1. To understand the fundamental of computer design
2. To know the pipelines and parallelism concepts
3. To know the issues in interconnect networks

Course Outcomes: At the end of the course, students will be able to:

1. Familiarize the instruction set, memory addressing of Computer
2. Handle the issues in pipelining and parallelism
3. Familiarize the practical issues in inter network

UNIT - I

Fundamentals of Computer Design: Fundamentals of Computer design, Changing faces of computing and task of computer designer, Technology trends, Cost price and their trends, measuring and reporting performance, quantitative principles of computer design, Amdahl’s law. Instruction set principles and examples- Introduction, classifying instruction set- memory addressing- type and size of operands, operations in the instruction set.

UNIT - II

Pipelines: Introduction, basic RISC instruction set, Simple implementation of RISC instruction set, Classic five stage pipe line for RISC processor, Basic performance issues in pipelining, Pipeline hazards, Reducing pipeline branch penalties.

Memory Hierarchy Design: Introduction, review of ABC of cache, Cache performance, Reducing cache miss penalty, Virtual memory.

UNIT - III

Instruction Level Parallelism the Hardware Approach: Instruction-Level parallelism, Dynamic scheduling, Dynamic scheduling using Tomasulo’s approach, Branch prediction, high performance instruction delivery- hardware based speculation.

ILP Software Approach: Basic compiler level techniques, static branch prediction, VLIW approach, Exploiting ILP, Parallelism at compile time, Cross cutting issues -Hardware verses Software.

UNIT - IV

Multi Processors and Thread Level Parallelism: Multi Processors and Thread level Parallelism- Introduction, Characteristics of application domain, Systematic shared memory architecture, Distributed shared – memory architecture, Synchronization.

UNIT - V

Inter Connection and Networks: Introduction, Interconnection network media, Practical issues in interconnecting networks, Examples of inter connection, Cluster, Designing of clusters.

Intel Architecture: Intel IA- 64 ILP in embedded and mobile markets Fallacies and pit falls

TEXT BOOK:

1. John L. Hennessy, David A. Patterson, “Computer Architecture: A Quantitative Approach”, 3rd Edition, Elsevier.

REFERENCE BOOKS:

1. John P. Shen and Miikko H. Lipasti, “Modern Processor Design: Fundamentals of Super Scalar Processors”, 2002, Beta Edition, McGraw-Hill
2. Kai Hwang, Faye A.Brigs., “Computer Architecture and Parallel Processing”, Mc Graw Hill.
3. Dezso Sima, Terence Fountain, Peter Kacsuk, “Advanced Computer Architecture - A Design Space Approach”, Pearson Education.

25ES121PE: MACHINE LEARNING AND DEEP LEARNING (PE - II)

M.Tech. I Sem.	L	T	P	C
	3	0	0	3

Course Overview

This course introduces the principles and practices of **Machine Learning and Deep Learning**, focusing on both theoretical understanding and practical implementation. It begins with the fundamentals of machine learning, including learning paradigms, model evaluation, and the challenges that have led to deep learning advances. The course then explores deep neural architectures, including feedforward networks, convolutional and recurrent networks, alongside critical concepts such as regularization and optimization. Finally, students are equipped with practical methodologies for building robust models and exposed to real-world applications across computer vision, speech recognition, and natural language processing.

Course Outcomes: At the end of the course, students will be able to:

1. Analyse machine learning algorithms and evaluate model performance using concepts like overfitting, bias-variance, and likelihood-based estimation.
2. Design deep feedforward neural networks using appropriate architectures, activation functions, and backpropagation techniques.
3. Apply regularization and optimization techniques to improve generalization and training efficiency of deep neural networks.
4. Implement convolutional and recurrent neural network models for processing spatial and sequential data.
5. Evaluate model performance and deployment strategies in real-world applications such as computer vision, NLP, and speech recognition.

Unit-I:

Machine Learning Basics: Learning algorithms, Capacity, overfitting and underfitting, estimators, bias and variance, maximum likelihood estimation, Bayesian statistics, Supervised and unsupervised learning algorithms, building a machine learning algorithm, challenges motivating deep learning.

Unit-II:

Deep Feedforward Networks: Gradient-based learning, hidden units, architecture design, back-propagation and other differentiation algorithms.

Unit-III:

Regularization for Deep Learning: Norm penalties, Dataset augmentation, multi-task learning, early stopping, sparse representations, ensemble methods, dropout. **Optimization:** Optimization for Training Deep Models, challenges in neural network optimization, basic algorithms, parameter initialization strategies, algorithms with adaptive learning rates.

Unit-IV:

Convolutional Neural Networks: The convolution operation, motivation, pooling, convolution and pooling as an infinitely strong prior, variants of the basic convolution function, structures outputs, data types, efficient convolution algorithms, random or unsupervised features.

Sequence Modeling:

Recurrent and Recursive Nets, Recurrent Neural Networks, Recursive Neural Networks, Long Short-Term Memory, optimization for long-term dependencies.

Unit V:

Practical Methodology: Performance metrics, selecting hyperparameters, debugging strategies. **Applications:** Large-scale deep learning, computer vision, speech recognition, natural language processing, other applications.

Text Book

1. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Reference Books

1. Géron, Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2nd ed., O'Reilly Media, 2019.
2. Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer, 2006.
3. Chollet, François. *Deep Learning with Python*. 2nd ed., Manning Publications, 2021.

25ES122PE: ADVANCED RISC ARCHITECTURES (PE – II)

M.Tech. I Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course introduces students to the fundamentals of processor design with a focus on instruction pipelining, memory organization, and performance optimization techniques. Emphasis is placed on the RISC-V Instruction Set Architecture (ISA), enabling students to understand and implement efficient IC designs. The course also explores pipeline hazards, cache optimization strategies, and real-world design challenges in modern processors. Students will gain hands-on experience by developing applications using the indigenous VEGA THEJAS32 microcontroller, bridging theoretical knowledge with practical system-level design and embedded programming

Course Outcomes: At the end of the course, students will be able to:

1. Apply the Instruction pipeline concept in IC design
2. Analyze the Hazards and Performance issues in Pipelining
3. Evaluate the Optimization techniques in Cache memory
4. Create IC designs using RISC-V Instruction set Architecture
5. Create an application using VEGA THEJAS32 Microcontroller

Unit

I:

Fundamental techniques of computer design: RISC and CISC architectures – Computer arithmetic – Comparison of RISC and CISC architectures. Verilog: Introduction and review of basic designs using Verilog. MIPS processor: Introduction to MIPS features – MIPS instruction set – Logical design of MIPS data path – Control unit and instruction decoder.

Unit II:

Processor Pipelining: Basics of Pipelining, Classic five stage pipelining in RISC processor,Performance issues in pipelining, Pipeline Hazards (Structural hazards, Data Hazards, Control Hazards), Data forwarding and bypassing techniques, Branch prediction technique: Static and Dynamic branch prediction.

Unit III:

Memory hierarchy: Memory hierarchy, Locality of References, Cache memory principles, Types of caches (Virtual and Physical cache), Cache architecture, Direct mapped, set associative and fully associative caches, Block Replacement Techniques and Write Strategy, Design Concepts in Cache Memory. Basic and Advanced Optimization Techniques in Cache Memory.

Unit IV:

RISC-V Architecture: RISC-V Instruction Set Architecture, Registers – General Purpose Registers, Control and Status Registers, Operating Modes, Programmers' Model for Base Integer ISA, Base Instruction Formats, Exceptions, Traps, and Interrupts, Machine-Level CSRs misa, mhartid, mstatus, mtvecmedeleg and mideleg, mip and mie, mepc, mcause, mtval.

Unit V:

VEGA THEJAS32 Microcontroller: Functional Bock diagram, CPU, Memory Mapped input output and Interrupts Project using ARIES Development board.

TEXTBOOKS:

1. Computer Architecture, A quantitative approach by John L Hennessy and David A Patterson Fifth Edition.
2. Computer organization and architecture, designing for performance, William Stallings Eight Edition.

REFERENCEBOOKS:

1. Georg Hager, Gerhard Wellein, Introduction to High Performance Computing for Scientists and Engineers, Chapman & Hall / CRC Computational Science series,2011.
2. The RISC-V Reader by David A Patterson and Andrew Waterman First Edition.
3. The RISC-V Instruction Set Manual Volume I: User-Level ISA Version 2.1

25ES123PE: AUTOMOTIVE EMBEDDED SYSTEMS (PE -II)

M.Tech. I Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course offers an in-depth exploration of embedded systems in modern vehicles, covering functional domains, standardized architectures like AUTOSAR, and in-vehicle communication protocols such as CAN and FlexRay. Students will learn about intelligent vehicle technologies, model-based development, and software reuse in the automotive domain. Emphasis is placed on system reliability, safety, and the integration of emerging technologies. The course bridges theoretical foundations with practical applications, preparing students for advanced roles in automotive software and embedded systems engineering.

Course Outcomes: After studying the course, each student is expected to be able to

1. Understand and analyze the structure, requirements, and standardized architecture of in-vehicle embedded systems including AUTOSAR.
2. Explain the role of intelligent technologies and communication protocols in modern vehicles.
3. Explain and Analyze FlexRay and CAN protocols.
4. Apply principles of software product-line engineering and variability modeling.
5. Demonstrate an understanding of automotive architecture description languages and model-based development techniques.

Unit I

Vehicle Functional Domains and Their Requirements: General Context, Functional Domains: Power Train Domain, Chassis, Domain, Body Domain, Multimedia, Telematic, and HMI, Active/Passive Safety, Diagnostic. Standardized Components, Models, and Processes: In-Vehicle Networks and Protocols, Operating Systems, Middleware, Architecture Description Languages for Automotive Applications. Certification Issue of Safety-Critical In-Vehicle Embedded Systems.

Application of the AUTOSAR Standard: Motivation: Shortcomings in Former Software Structures, Setting up AUTOSAR, Main Objectives of AUTOSAR, Working Methods in AUTOSAR. Mainstay of AUTOSAR-AUTOSAR Architecture: AUTOSAR Concept, Layered Software Architecture. Main Areas of AUTOSAR Standardization: BSW and RTE: BSW, BSW Conformance Classes, RTE. Main Areas of AUTOSAR Standardization-Methodology and Templates: Objectives of the Methodology, Description of the Methodology, AUTOSAR Models, Templates, and Exchange Formats, System Configuration, ECU Configuration, Implementation to Existing Development Processes and Tooling. AUTOSAR in Practice-Conformance Testing. AUTOSAR in Practice-Migration to AUTOSAR ECU. AUTOSAR in Practice-Application of OEM–Supplier Collaboration. AUTOSAR in Practice: Demonstration of AUTOSAR-Compliant ECUs: Description of the Demonstrator, Concepts Shown by the Demonstrator.

Unit II:

Intelligent Vehicle Technologies: Introduction: Road Transport and Its Evolution: Such a Wonderful Product, Safety Problems, Congestion Problem, Energy and Emissions. New Technologies: Sensor Technologies, Sensor Fusion, Wireless Network Technologies, Intelligent Control Applications, Latest Driving Assistance. Dependability Issues: Introduction, Fail-Safe Automotive Transportation Systems, Intelligent Auto diagnostic. Fully Autonomous Car-Dream or Reality?: Automated Road Vehicles, Automated Road Network, Automated Road Management, Deployment Paths.

Embedded Automotive Protocols: Automotive Communication Systems-Characteristics and Constraints: From Point-to-Point to Multiplexed Communications, Car Domains and Their Evolution, Different Networks for Different Requirements, Event-Triggered versus Time-Triggered. In-Car Embedded Networks: Priority Buses, TT Networks, Low-Cost Automotive Networks, Multimedia Networks. Middleware Layer: Rationale for a Middleware, Automotive MWs Prior to AUTOSAR, AUTOSAR. Open Issues for Automotive Communication Systems: Optimized Networking Architectures, System Engineering.

Unit III:

FlexRay Protocol: Introduction: Event-Driven versus Time-Driven Communication, Objectives of FlexRay, History of FlexRay. FlexRay Communication: Frame Format, Communication Cycle, Static Segment, Dynamic Segment. FlexRay Protocol: Protocol Architecture, Protocol Wakeup and Startup, Wakeup, Clock Synchronization, Fault-Tolerance Mechanisms. FlexRay Application: FlexRay Implementation, FlexRay Tool Support. Impact on Development, Verification of FlexRay.

Dependable Automotive CAN Networks: Introduction: Main Requirements of Automotive Networking, Networking Technologies, CAN Features and Limitations. Data Consistency Issues: Management of Transient Channel Faults in CAN, Impairments to Data Consistency, On the Probability of the Data Inconsistency Scenarios, Solutions to Really Achieve Data, Consistency over CAN. CANcentrate and ReCANcentrate-Star Topologies for CAN: Rationale, CANcentrate and ReCANcentrate Basics, Other Considerations. CANEL: Clock Synchronization, Data Consistency, Error Containment, Support for Fault Tolerance, CANELy Limitations. FTT-CAN-Flexible Time-Triggered Communication on CAN: FTT System Architecture, Dual-Phase Elementary Cycle, SRDB, Main Temporal Parameters within the EC, Fault-Tolerance Features, Accessing the Communication Services. FlexCAN-A Deterministic, Flexible, and Dependable Architecture for Automotive Networks: Control System Transactions, FlexCAN Architecture, How FlexCAN Addresses CAN Limitations, FlexCAN Applications and Summary. Other Approaches to Dependability in CAN: TTCAN, Fault-Tolerant Time-Triggered Communication Using CAN, TCAN, ServerCAN, Fault-Tolerant Clock Synchronization Over CAN.

Unit IV:

Product Lines in Automotive Electronics: Characteristics of Automotive Product Lines: Basic Concepts of Software Product Lines, Characteristics and Needs of Automotive Electronics with Respect to Product-Line Engineering. Basic Terminology: Software Product Lines, Variability, Feature Modeling as a Form of Variability Modeling, Discussion-Feature Modeling for the Automotive Domain. Global Coordination of Automotive Product-Line Variability: Coordination of Small- to Medium-Sized Product Lines, Coordination of Highly Complex Product Lines. Artifact-Level Variability: Basic Approach, Difficulties Related to Artifact-Local Variability, Representing Variability in ECU, Requirements Specifications, Evaluation of Representations, Mapping Representations to a Common Basis.

Reuse of Software in Automotive Electronics: Reuse of Software-A Challenge for Automotive OEMs. Requirements for the Reuse of Software in the Automotive Domain. Supporting the Reuse of Application Software Components in Cars: Processes, Development of Modularized Automotive Software Components, Function Repository, Development of an In-Vehicle Embedded System. Application Example.

Unit V:

Automotive Architecture Description Languages: Introduction. Engineering Information Challenges: Reducing Cost and Lead Time, Development Organization and Information Exchange, Product Complexity, Quality and Safety, Concurrent Engineering, Reuse and Product Line Architectures, Analysis and Synthesis, Prototyping. State of Practice: Model-Based Design, Tools, Problems beyond Model-Based Design. ADL as a Solution: General Aspects on an Automotive ADL, What Needs to Be Modeled. Existing ADL Approaches: Forsoft Automotive, SysML, Architecture and Analysis Description Language, Modeling and Analysis of Real-Time and Embedded Systems, AUTOSAR Modeling, EAST-ADL.

Model-Based Development of Automotive Embedded Systems:

Introduction: What Is MBD? Motivating MBD for Automotive Embedded Systems: Role of MBD in Automotive Embedded Systems Development, MBD Means, Driving Factors for MBD, Potential Benefits of MBD Approaches. Context, Concerns, and Requirements: Contextual Requirements on MBD, Product Concerns Addressed by MBD Efforts. MBD Technology: Modeling Languages: Abstractions, Relations, and Behavior, Analysis Techniques, Synthesis Techniques, Tools. State of the Art and Practice: Automotive State of Practices, Research and Related Standardization Efforts. Guidelines for Adopting MBD in Industry: Strategic Issues, Adopting MBD: Process and Organizational Considerations, Desired Properties of MBD Technologies, Common Arguments against MBD and Pitfalls.

Text Books:

1. N. Navet and F. Simonot-Lion, Eds., Automotive Embedded Systems Handbook. Boca Raton, FL, USA: CRC Press, 2008.
2. William B Ribbens, Understanding Automotive Electronics: An Engineering Perspective, Seventh Edition, Elsevier, 2012.
3. Feiler, P. H., Gluch, D. P., & Hudak, J. J. (2012). Model-Based Engineering with AADL: An Introduction to the SAE Architecture Analysis & Design Language. Addison-Wesley/SAE International.

Reference Books:

1. Ola Larses, Architecting and Modeling Automotive Embedded Systems, KTH, 2005
2. AUTOSAR. (2025). Standards of AUTOSAR. Retrieved from https://www.autosar.org/standards
3. Vlacic, L., Parent, M., & Harashima, F. (2001). Intelligent Vehicle Technologies: Theory and Applications. Elsevier Science.

25ES103PC: FPGA BASED SYSTEM DESIGN LABORATORY (Lab – I)

M.Tech. I Sem.	**L**	**T**	**P**	**C**
	0	**0**	**4**	**2**

Course Overview

This laboratory course focuses on practical hands-on experience with FPGA design and implementation using Hardware Description Languages (HDLs) such as Verilog or VHDL. Students will learn to design, simulate, and implement fundamental digital building blocks, combinational and sequential circuits, and complex systems like finite state machines and communication modules on FPGA platforms. The course covers a range of topics from basic logic gates and arithmetic units to real-time systems and digital signal processing applications. Emphasis is placed on simulation, verification, and hardware validation to bridge the gap between theoretical concepts and real-world digital system design.

Course Outcomes: After studying the course, each student is expected to be able to

1. Write and simulate HDL code for basic combinational and sequential digital circuits using Verilog HDL.
2. Design and implement complex digital systems such as FSMs, arithmetic logic units, and communication modules on FPGA platforms.
3. Develop and verify interfacing protocols for external hardware devices like seven-segment displays, keypads, and UART communication.
4. Apply design methodologies for timing, synchronization, and resource optimization in FPGA-based digital system design.
5. Demonstrate the ability to deploy and debug digital designs on FPGA development boards, validating functionality through hardware testing

List of Experiments (Any 12 Experiments)

1. **LED Blinking Using Verilog:** Implement a basic LED blink circuit to understand FPGA programming and output pin control.
2. **Design and Simulation of Basic Logic Gates using Verilog HDL:** To write HDL code for basic logic gates (AND, OR, NOT, XOR), simulate their functionality, and verify the outputs using a waveform viewer.
3. **Implementation of 4-bit Adder/Subtractor on FPGA:** To design and implement a 4-bit adder/subtractor using Verilog, simulate and verify functionality on an FPGA development board.
4. **Design of a 4x1 Multiplexer and 1x4 Demultiplexer:** To write and simulate Verilog code for a 4x1 MUX and 1x4 DEMUX and validate outputs through FPGA implementation.
5. **Comparator Design (2-bit or 4-bit):** Design a simple digital comparator circuit that compares two binary inputs.
6. **Design and Implementation of a 4-bit Synchronous Counter:** To implement a 4-bit up/down synchronous counter using HDL, simulate for timing and logic correctness, and deploy it on FPGA hardware.
7. **Finite State Machine (FSM) Design:** Sequence Detector-To design a Moore or Mealy FSM for sequence detection, simulate the state transitions, and implement the design on an FPGA board.
8. **Design and Implementation of an ALU Supporting Basic Operations:** To design an Arithmetic Logic Unit that performs basic arithmetic and logic functions (ADD, SUB, AND, OR, NOT) and test it on FPGA.
9. **Interfacing Seven Segment Display with FPGA:** To write HDL code to drive a seven-segment display with binary or BCD inputs and implement the interface on FPGA hardware.

10. **Shift Register (Left/Right):** Design a shift register that shifts input bits left or right on each clock pulse.
11. **Implementation of Traffic Light Controller using FSM on FPGA:** To design a traffic light controller using finite state machines, simulate its time sequence, and implement it on FPGA hardware.
12. **PWM Signal Generation using FPGA:** To implement a pulse-width modulation (PWM) generator using Verilog HDL and demonstrate its use in applications like LED brightness control.
13. **Real-Time Clock Design and Display using FPGA:** To implement a real-time clock on FPGA with hour-minute-second display and use multiplexed seven-segment display for output.
14. **Serial Input with UART Receiver (Basic):** Implement a simple UART receiver to receive serial data and blink an LED.
15. **Implementing Digital Filters (FIR/IIR) on FPGA:** To implement a basic Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) filter in HDL and simulate its output for given inputs.

25ES104PC: EMBEDDED SYSTEM DESIGN LABORATORY (Lab – II)

M.Tech. I Sem.	**L**	**T**	**P**	**C**
	0	**0**	**4**	**2**

Course Overview

The Embedded System Design Laboratory focuses on the practical aspects of designing, developing, and testing embedded systems using real-time hardware platforms. Through a series of hands-on experiments, students will learn to configure embedded devices, interface sensors and actuators, perform GPIO programming, and implement embedded applications for automation and control. The course also explores advanced topics such as wireless communication, web hosting, media streaming, and custom firmware porting. Emphasis is placed on understanding system-level integration, debugging, and real-world deployment of embedded solutions.

Course Outcomes: After studying the course, each student is expected to be able to

1. Set up embedded development environments by flashing operating systems and configuring device interfaces.
2. Interface and program embedded peripherals such as sensors, actuators, and displays using GPIO.
3. Design and implement embedded applications for monitoring, automation, and communication.
4. Deploy embedded systems with network capabilities for web hosting, media transmission, and real-time data access.

List of Experiments:

1. **Functional Testing Of Devices:** Flashing the OS on to the device into a stable functional state by porting desktop environment with necessary packages.
2. **Exporting Display On To Other Systems:** Making use of available laptop/desktop displays as a display for the device using SSH client & X11 display server.
3. **GPIO Programming:** Programming of available GPIO pins of the corresponding device using native programming language. Interfacing of I/O devices like LED/Switch etc., and testing the functionality.
4. **Interfacing Chronos eZ430:** Chronos device is a programmable texas instruments watch which can be used for multiple purposes like PPT control, Mouse operations etc., Exploit the features of the device by interfacing with devices.
5. **ON/OFF Control Based On Light Intensity:** Using the light sensors, monitor the surrounding light intensity & automatically turn ON/OFF the high intensity LED's by taking some pre-defined threshold light intensity value.
6. **Battery Voltage Range Indicator:** Monitor the voltage level of the battery and indicating the same using multiple LED's (for ex: for 3V battery and 3 LED's, turn on 3 LED's for 2-3V, 2 LED's for1-2V, 1 LED' for 0.1-1V & turn off all for 0V)
7. **Dice Game Simulation:** Instead of using the conventional dice, generate a random value similar to dice value and display the same using a 16X2 LCD. A possible extension could be to provide the user with option of selecting single or double dice game.
8. **Displaying RSS News Feed On Display Interface:** Displaying the RSS news feed headlines on a LCD display connected to device. This can be adapted to other websites like twitter or other information websites. Python can be used to acquire data from the internet.

9. **Porting Open wrt To the Device:** Attempt to use the device while connecting to a wifi network using a USB dongle and at the same time providing a wireless access point to the dongle.
10. **Hosting a website on Board:** Building and hosting a simple website(static/dynamic) on the device and make it accessible online. There is a need to install server (eg: Apache) and there by host the website.
11. **Webcam Server:** Interfacing the regular USB webcam with the device and turn it into fully functional IP webcam & test the functionality.
12. **FM Transmission:** Transforming the device into a regular FM transmitter capable of transmitting audio at desired frequency (generally 88-108 Mhz)

Note: Devices mentioned in the above lists include Arduino, Raspbery Pi, Beaglebone

25ES105PC:RESEARCH METHODOLOGY AND IPR

M.Tech. I Sem.	**L**	**T**	**P**	**C**
	2	**0**	**0**	**2**

Course Objectives:

1. To understand the research problem
2. To know the literature studies, plagiarism and ethics
3. To get the knowledge about technical writing
4. To analyze the nature of intellectual property rights and new developments
5. To know the patent rights

Course Outcomes: At the end of this course, students will be able to

1. Understand research problem formulation.
2. Analyze research related information
3. Follow research ethics
4. Understand that today’s world is controlled by Computer, Information Technology, but tomorrow world will be ruled by ideas, concept, and creativity.
5. Understanding that when IPR would take such important place in growth of individuals & nation, it is needless to emphasis the need of information about Intellectual Property Right to be promoted among students in general & engineering in particular.
6. Understand that IPR protection provides an incentive to inventors for further research work and investment in R & D, which leads to creation of new and better products, and in turn brings about, economic growth and social benefits.

UNIT- I:

Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem.

Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations

UNIT- II:

Effective literature studies approaches, analysis, Plagiarism, Research ethics

UNIT- III:

Effective technical writing, how to write report, Paper Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee

UNIT- IV:

Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.

UNIT- V:

Patent Rights: Scope of Patent Rights. Licensing and transfer of technology. Patent information and databases. Geographical Indications. New Developments in IPR: Administration of Patent System. New developments in IPR; IPR of Biological Systems, Computer Software etc.

Traditional knowledge Case Studies, IPR and IITs.

TEXT BOOKS:

1. Stuart Melville and Wayne Goddard, "Research methodology: an introduction for science &engineering students'"
2. Wayne Goddard and Stuart Melville, "Research Methodology: An Introduction"

REFERENCE BOOKS:

1. Ranjit Kumar, 2nd Edition, "Research Methodology: A Step by Step Guide for beginners"
2. Halbert, "Resisting Intellectual Property", Taylor & Francis Ltd ,2007.
3. Mayall, "Industrial Design", McGraw Hill, 1992.
4. Niebel, "Product Design", McGraw Hill, 1974.
5. Asimov, "Introduction to Design", Prentice Hall, 1962.
6. Robert P. Merges, Peter S. Menell, Mark A. Lemley, "Intellectual Property in New Technological Age", 2016.
7. T. Ramappa, "Intellectual Property Rights Under WTO", S. Chand, 2008

25ES201PC: EMBEDDED REAL TIME OPERATING SYSTEM (PC -III)

M.Tech. II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course provides a comprehensive introduction to Real-Time Operating Systems (RTOS) and their applications in embedded systems. Students will explore the fundamental concepts of real-time embedded systems, understand task management, synchronization mechanisms like semaphores and message queues, and delve into exception handling and timer services. The course also covers memory management strategies and practical techniques for modularizing applications to meet real-time constraints. Additionally, it addresses common design challenges such as deadlocks and priority inversion, preparing students to design, analyse, and implement robust real-time embedded applications.

Course Outcomes: After completing this course the student will be able to:

1. Analyse key concepts and roles of real-time operating systems in embedded system design.
2. Evaluate task scheduling and synchronization methods in real-time systems.
3. Design interrupt and timer handling mechanisms for embedded real-time environments.
4. Apply real-time memory management techniques in embedded applications.
5. Create modular real-time applications to resolve deadlocks and priority inversion.

Unit-I:

Introduction to RTOS: Real life examples of embedded systems, real-time embedded systems, the future of embedded systems, a brief history of operating systems, defining an RTOS, the scheduler, objects, services, key characteristics of an RTOS.

Unit-II:

Tasks: Introduction, defining a task, task states and scheduling, typical task operations, typical task structure, synchronization, communication, and concurrency.

Semaphores: Introduction, defining semaphores, typical semaphore operations, typical semaphore use.

Message Queues: Introduction, defining message queues, message queue states, message queue content, message queue storage, typical message queue operations, and typical message queue use.

Unit-III:

Exceptions and Interrupts: Introduction, what are exceptions and interrupts, a closer look at exceptions and interrupts, processing general exceptions, the nature of spurious interrupts.

Timer and Timer Services: Introduction, real-time clocks and system clocks, programmable interval timers, timer interrupt service routines, a model for implementing the soft-timer handling facility, timing wheels, soft timers and timer related operations.

Unit-IV:

Memory Management: Introduction, dynamic memory allocation in embedded systems, fixed-size memory management in embedded systems, blocking vs. non-blocking memory functions, hardware memory management units.

Unit-V:

Modularizing Applications: Introduction, an outside-in approach to decomposing applications, guidelines and recommendations for identifying concurrency, schedulability analysis – rate monotonic analysis.

Common Design Problems: Introduction, resource classification, deadlocks, priority inversion.

Text Books:

1. Qing Li, and Caroline Yao. *Real-Time Concepts for Embedded Systems*. CMP Books, 2003.

Reference Books:

1. Prasad, K. V. K. K. *Embedded Real-Time Systems: Concepts, Design & Programming*. Dream Tech Press, 2005.
2. Simon, David E. *An Embedded Software Primer*. 1st ed., 5th impression, Addison-Wesley Professional, 2007.
3. Singh, Rajib Mall. *Real-Time Systems: Theory and Practice*. Pearson Education India, 2006

.

25ES202PC: IOT SYSTEM DESIGN (PC-IV)

M.Tech. II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course introduces the foundational principles and system-level design of the Internet of Things (IoT). It covers the basic concepts, definitions, and significance of IoT in modern connected systems. Students will learn about the interaction between IoT and Machine-to-Machine (M2M) communication, and how they integrate to form intelligent networks. The course provides a detailed understanding of IoT architecture, including device-level design, network protocols, and data management. By the end of the course, students will have a comprehensive view of how to design and develop scalable, interoperable, and secure IoT systems.

Course Outcomes: At the end of this course, students will be able to

1. Integrate the sensors and actuator depending on the applications
2. Interface the IoT and M2M with value chains
3. Write Python programming for Arduino, Raspberry Pi devices
4. Design IoT based systems such as Agricultural IoT, Vehicular IoT etc.,

Unit I:
IoT introduction: Introduction and definition of IoT, Evolution of IoT, IoT growth, Application areas of IoT, Characteristics of IoT, IoT stack, Enabling technologies, IoT levels, IoT sensing and actuation, Sensing types, Actuator types.

Unit II:
IoT and M2M: M2M to IoT – A Basic Perspective– Introduction, Differences and similarities between M2M and IoT, SDN and NFV for IoT.M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT, The international driven global value chain and global information monopolies.

Unit III:
IoT Hands-on: Introduction to Arduino Programming, Integration of Sensors and Actuators with Arduino. Introduction to Python programming, Introduction to Raspberry Pi, Interfacing Raspberry Pi with basic peripherals, Implementation of IoT with Raspberry Pi.

Unit IV:
IoT Architecture: IoT Architecture components, Comparing IoT architectures, A simplified IoT architecture, The core IoT functional stack, IoT data management and compute stack.

Unit V:
IoT System design: Challenges associated with IoT, Emerging pillors of IoT, Agricultural IoT, Vehicular IoT, Healthcare IoT, Smart cities, Transportation and logistics.

Text Books:

1. Sudip Misra, Anandarup Mukherjee, Arijit Roy "*Introduction to IOT*", Cambridge University Press.
2. David Hanes, Gonzalo salgueiro, Patrick Grossetete, Rob barton, Jerome henry "*IoT Fundamentals Networking technologies, protocols, and use cases for IoT*", Cisco Press

Reference Books:

1. Cuno pfister, "*Getting started with the internet of things*", O Reilly Media, 2011
2. Francis daCosta, "*Rethinking the Internet of Things: A Scalable Approach to Connecting Everything*", 1 st Edition, Apress Publications.
3. "*Internet of Things concepts and applications*", Wiley
4. Arshdeep Bahga,Vijay Madisetti "*Internet of Things A Hands on approach*", Universities Press
5. Shriram K Vasudevan, RMD Sundaram, Abhishek S Nagarajan, "*Internet of things*" John Wiley and Sons.
6. Massimo Banzi, Michael Shiloh Make*: Getting Started with the Arduino,* Shroff Publisher/Maker Media Publishers.

25ES231PE: GPU ARCHITECTURES (PE– III)

M.Tech. II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course introduces students to the principles and practices of GPU computing, with a focus on CUDA programming and parallel algorithm design. It covers the architectural foundations of modern GPUs, emphasizing how they differ from traditional CPUs in terms of memory hierarchy, thread scheduling, and computational throughput. Students will gain hands-on experience in writing and optimizing GPU code, debugging parallel programs, and applying GPU-specific optimization strategies such as memory coalescing and shared memory utilization. The course also includes practical exposure to applying GPU acceleration in areas such as scientific computing and deep learning, particularly on embedded platforms.

Course Outcomes: At the end of this course**, s**tudents will be able to:

1. Understand working proficiency with CUDA, algorithmic GPU programming and parallel computing
2. Comprehend with classic scientific computing algorithms and problems
3. Optimize GPU code and debug GPU code
4. Analyze architecture specific details like memory access coalescing ,shared memory usage, GPU thread scheduling
5. Apply deep learning algorithms on embedded GPUs

Unit I:
GPU architectures: Introduction to the ideas of parallelism and the GPU programming model CPU vs GPU Parallelizing algorithms on paper, First CUDA program**.**

Unit II:
CUDA programming: Hardware of Graphics Processing Units and parallel communication patterns, Brief on GPU architecture, Basics of CUDA C, Floating point precision and support on GPUs.

Unit III:
Parallel primitives and algorithms on GPU: The CUDA programming language will be mastered while learning how to implement these algorithms., Matrix Operations, Stencil – Image Blurring, Filters, Gauss Jacobi-Finite difference updates for PDEs, Histogram, binning 1, Reduce – Maximum and Minimum – Summation, Prefix-sum (Scan) Algorithm – Radix Sort, Generating Cumulative Distributions, Complex algorithms – N-body solutions.

Unit IV:
Optimizing GPU Applications: Coalesced Memory Transactions, Grid Blocks, Thread Blocks, domain decomposition, Asynchronous Kernels and Multistreaming Possible Items: Libraries on GPU, cuBLAS Thrust, cuFFT, cuRAND, Multi-node GPU processing, Multi-GPU per node processing, CUDA in other languages (Python/Fortran), Scaling.

Unit V:
Deep learning on GPUs: Deep learning on GPUs, Combining graphics and compute, Display the results of computations– Interactive systems, Collision detection with voxelized solid (Gargoyle), Ray tracing in CUDA kernels, or ray tracing cores, Microsoft DXR (DX12 API), Vulkan, NVIDIA OptiX / RTX, NVIDIA Turing: “World‘s First Ray Tracing GPU”- Quadro RTX, Geforce RTX

Reference Books:

1. GPUs for Graphics: OpenGL 4.0 Shading Language Cookbook, 2nd Edition
2. Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Publisher: Addison-Wesley Professional,2013, 3rd Edition.

25ES232PE: VLSI TEST AND TESTABILITY (PE – III)

M.Tech. II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course introduces students to digital system testing techniques and fault diagnosis in VLSI circuits. It covers fault models, test generation methods for combinational and sequential circuits, and various design-for-testability (DFT) strategies. Students will explore scan-based designs, Built-In Self- Test (BIST) techniques, memory testing algorithms, and fault diagnosis methods. Emphasis is placed on practical testing challenges in digital and embedded systems to ensure reliability and fault tolerance.

Course Outcomes: At the end of this course, students will be able to

1. Identify different types of faults in digital circuits and explain their corresponding logical fault models.
2. Apply test generation techniques to detect faults in combinational and sequential digital circuits.
3. Design scan-based and ad-hoc DFT architectures for improving circuit testability.
4. Implement Built-In Self-Test (BIST) strategies for digital systems and memory blocks.
5. Analyse fault diagnosis techniques to locate and interpret faults in logic-level digital circuits.

Unit-I:
Basics of Testing and Fault Modeling: Introduction to Testing - Faults in digital circuits - Modeling of faults - Logical Fault Models - Fault detection - Fault location - Fault dominance - Logic Simulation - Types of simulation - Delay models - Gate level Event-driven simulation.

Unit-II:
Test Generation for Combinational and Sequential Circuits: Test generation for combinational logic circuits - Testable combinational logic circuit design - Test generation for sequential circuits - design of testable sequential circuits. Design For Testability

Unit-III:
Design for Testability: Ad-hoc design - Generic scan-based design - Classical scan-based design – System level DFT approaches.

Unit-IV:
Self-Test and Test Algorithms: Built-In Self-Test - Test pattern generation for BIST - Circular BIST- BIST Architectures - Testable Memory Design - Test algorithms - Test generation for Embedded RAMs.

Unit-V:
Fault Diagnosis: Logic Level Diagnosis - Diagnosis by UUT reduction - Fault Diagnosis for Combinational Circuits - Self-checking

Text Books:

1. Abramovici, M., M. A. Breuer, and A. D. Friedman. *Digital Systems Testing and Testable Design*. 1st ed., Jaico Publishing House, 2002.
2. Bushnell, M. L., and V. D. Agrawal. *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Illustrated ed., Springer Science & Business Media, 2006

Reference Books

1. Lala, P. K. *Digital Circuit Testing and Testability*. Academic Press, 2002.
2. Crouch, A. L. *Design-for-Test for Digital IC's and Embedded Core Systems*. 1st ed., Prentice Hall International, 1999.

25ES233PE: HARDWARE SOFTWARE CO-DESIGN (PE – III)

M.Tech. II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course focuses on the integrated development of hardware and software components in embedded system design. It introduces students to key co-design issues, including system partitioning, prototyping, and emulation techniques. The course explores architecture-specific optimization strategies that balance performance, power, and flexibility. Additionally, students will gain practical knowledge of industry-standard tools used in hardware/software co-design. By the end of the course, learners will be able to effectively design, implement, and evaluate embedded systems through a unified hardware-software development approach.

Course Outcomes: At the end of the course, students will be able to:

1. Acquire the knowledge on various models of Co-design.
2. Explore the interrelationship between Hardware and software in a embedded system
3. Acquire the knowledge of firmware development process and tools during Co-design.
4. Implement validation methods and adaptability.

Unit-I:

Co-Design Issues: Co- Design Models, Architectures, Languages, A Generic Co-design
Methodology. Co-Synthesis Algorithms: Hardware software synthesis algorithms: hardware – software partitioning distributed system co-synthesis.

Unit-II:

Prototyping and Emulation: Prototyping and emulation techniques, prototyping and emulation environments, future developments in emulation and prototyping architecture specialization techniques, system communication infrastructure**.**

Target Architectures: Architecture Specialization techniques, System Communication infrastructure, Target Architecture and Application System classes, Architecture for control dominated systems (8051- Architectures for High performance control), Architecture for Data dominated systems (ADSP21060, TMS320C60), Mixed Systems**.**

Unit-III:

Compilation Techniques and Tools for Embedded Processor Architectures: Modern embedded architectures, embedded software development needs, compilation technologies, practical consideration in a compiler development environment**.**

Unit-IV:

Design Specification and Verification: Design, co-design, the co-design computational model, concurrency coordinating concurrent computations, interfacing components, design verification, implementation verification, verification tools, interface verification**.**

Unit-V:

Languages for System – Level Specification and Design-I: System – level specification, design representation for system level synthesis, system level specification languages**,**

Languages for System – Level Specification and Design-II: Heterogeneous specifications and multi- language co-simulation, the cosyma system and lycos system**.**

Text Books:

1. *Hardware / Software Co- Design Principles and Practice* – Jorgen Staunstrup, Wayne Wolf – Springer, 2009.

Reference Books:

1. *Hardware / Software Co- Design* - Giovanni De Micheli, Mariagiovanna Sami, Kluwer Academic Publishers, 2002.
2. *A Practical Introduction to Hardware/Software Co-design* -Patrick R. Schaumont, Springer, 2010

25ES241PE: HARDWARE SECURITY IN VLSI DESIGN (PE – IV)

M.Tech. II Sem.	L	T	P	C
	3	0	0	3

Course Overview

This course provides an in-depth understanding of security challenges and vulnerabilities in modern VLSI systems, particularly within SoC and PCB designs. Students will learn about hardware threats such as Trojans, reverse engineering, IP piracy, side-channel attacks, test-oriented and physical attacks, and how to design secure hardware using primitives like PUFs and TRNGs. Through a mix of theoretical foundations and hands-on experimentation, students will be equipped to identify, analyse, and mitigate hardware security threats in real-world VLSI systems.

Course Outcomes: At the end of the course, **s**tudents will be able to:

1. Identify and explain the key layers and components of VLSI systems vulnerable to hardware attacks.
2. Analyse different types of hardware Trojans and IP piracy techniques and suggest appropriate countermeasures.
3. Evaluate side-channel and test-oriented attacks on SoC and FPGA-based systems using suitable models.
4. Demonstrate an understanding of physical attacks and PCB vulnerabilities through attack modeling and analysis.
5. Design and implement hardware security primitives such as PUFs and TRNGs to protect VLSI designs from cloning and counterfeiting.

Unit I

Introduction to Hardware Security: Overview of a computing system, layers of a computing system, hardware security vs. hardware trust, attacks, vulnerabilities and countermeasures, conflict between security and test/debug, evolution of hardware security, overview of electronic hardware – nanoscale technologies, ASICs and FPGAs, printed circuit board, embedded systems, hardware-firmware-software interaction.

Unit II:

Hardware Trojans: Introduction, SoC design flow, hardware trojans, hardware trojans in FPGA designs, hardware trojans taxonomy, trust benchmarks, countermeasures against hardware trojans, hardware trojan attacks. **Hardware IP Piracy and Reverse Engineering:** Introduction, hardware intellectual property (IP), security issues in IP-based SoC design, security issues in FPGA, reverse engineering and tampering.

Unit III:

Side-Channel Attacks: Introduction, taxonomy of side-channel attacks, uncommon side-channel attacks, power analysis attacks, electromagnetic (EM) side-channel attacks, fault-injection attacks, timing attacks.**Test-Oriented Attacks:** Introduction, scan-based attacks, JTAG-based attacks.

Unit IV:

Physical Attacks and Countermeasures: Introduction, reverse engineering, probing attack.**Attacks on PCB:** Security challenges, attack models, bus snooping attack.

Unit-V:

Hardware Security Primitives: Introduction, physical unclonable functions (PUFs), true random number generator (TRNG), design of anti-counterfeit, existing challenges and attacks. **Security and Trust Assessment:** Security assets and attack models, pre-silicon and post-silicon security and trust assessment.

Text Book

1. Bhunia, Swarup, and Mark M. Tehranipoor. *Hardware Security: A Hands-on Learning Approach*. 1st ed., Academic Press, 2019.

Reference Books

1. Tehranipoor, Mark, and Cliff Wang, editors. *Introduction to Hardware Security and Trust*. Springer, 2012.
2. Bhunia, Swarup, and Sandip Ray. *Fundamentals of IP and SoC Security: Design, Verification, and Debug*. Springer, 2017.
3. Wolf, Marilyn. *Embedded System Interfacing: Design for the Internet-of-Things (IoT)*. Morgan Kaufmann, 2019.
4. Skorobogatov, Sergei. *Semi-Invasive Attacks: A New Approach to Hardware Security Analysis*. Springer, 2007.

25ES242PE: HARDWARE ACCELERATORS FOR MACHINE LEARNING MODELS (PE -IV)

M.Tech. II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course provides a comprehensive understanding of designing hardware accelerators tailored for Deep Neural Network (DNN) models. It begins with an exploration of various DNN architectures and their development environments, followed by strategies for efficient hardware implementation. Students will learn key techniques for optimizing memory usage and computational performance in DNN execution. The course emphasizes near-data processing paradigms and performance benchmarking of DNN models on custom hardware platforms. Finally, students will gain practical experience in designing and implementing accelerator logic using contemporary hardware description tools and platforms, preparing them for real-world applications in edge AI and high- performance computing.

Course Outcomes: At the end of the course, students will be able to:

1. Explore the various DNN models and development resources.
2. Understand the hardware implementation strategies for DNN.
3. Develop the memory optimization and computational optimization techniques to DNN.
4. Perform the near-data processing and benchmark evaluation to DNN models.
5. Design and implement the accelerator logic to DNN models.

Unit I

Overview of DNNs: Convolutional Neural Networks (CNNs), Popular DNN Models, DNN development resources: Frameworks, Models, Popular Data Sets for Classification, And Data Sets for Other Tasks

Evolution of hardware platforms for Deep Learning: CPUs, GPUs, FPGAs, DSPs, accelerators; Hardware considerations in inference and training, Accelerate Kernel Computations on CPU and GPU Platforms, Energy-Efficient Dataflow for Accelerators, DNN data handling characteristics, Weight Stationary (WS), Output Stationary (OS), No Local Reuse (NLR), Row Stationary (RS). Accelerating the convolution operation: Algorithms, Data flow patterns, Memory reuse Case-study on writing a custom GPU

Unit II:

Kernel for accelerating convolution, optimizing networks: Weight quantization, network compression, sparse operations, zero forwarding, learning with hardware in the loop, learning and inference on low-memory devices.

Unit III:

Memory and compute: Optimizations to CNNs such as tiling, loop optimizations, batching, quantization, pruning, Cache Blocking, four convolution strategies (Direct, GEMM, FFT and Winograd), Model-size aware and system-aware pruning of CNNs, MLPerf Benchmark for evaluating DNN accelerators.

Unit IV:

Near-data processing: DRAM, SRAM, Non-volatile Resistive Memories, Sensors, co-design of DNN models and hardware: Reduce Precision, Reduced Number of Operations and Model Size. Benchmarking metrics DNN evaluation and comparison, Metrics for DNN Models, Metrics for DNN Hardware.

Unit V:

Deep Learning on Systolic Array and Tensor Processing Unit (TPU) v1 to v4, Distinct Characteristics of Training and Inference, Architectures of TPU v1, v2, v3 and v4; comparison between their architectures, Comparison of CPU, TPU and GPU, Deep Learning on FPGA and Microsoft’s Brainwave Architecture, Deep Learning techniques on FPGA; efficacy of FPGAs for binarized neural networks (BNNs).

Text Books:

1. Shiho Kim, Ganesh Chandra Deka,*Hardware Accelerator Systems for Artificial Intelligence and Machine Learning*, Volume 122 - March 28, 2021, 1st Edition,

Reference Books:

1. V. Sze, Y. -H. Chen, T. -J. Yang and J. S. Emer, "*Efficient Processing of Deep Neural Networks: A Tutorial and Survey*," in Proceedings of the IEEE, vol. 105, no. 12, pp. 2295-2329, Dec. 2017, doi:10.1109/JPROC.2017.2761740.
2. https://docs.amd.com/r/2022.2-English/ug896-vivado-ip/Xilinx-Resources.

25ES243PE: IMAGE AND VIDEO PROCESSING (PE -IV)

M.Tech. II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course provides a comprehensive introduction to the fundamentals of image and video processing, highlighting the distinct characteristics and challenges associated with each domain. It begins by exploring the key differences between image and video data in terms of structure, temporal aspects, and processing techniques. The course then delves into essential filtering operations used in both image and video processing, enabling students to enhance, restore, and extract relevant information from visual data. In the latter part of the course, students are introduced to the principles of image and video compression, covering both lossless and lossy techniques. Through theoretical concepts and practical applications, the course aims to equip students with the foundational knowledge and tools necessary to analyze and process visual information effectively.

Course Outcomes: At the end of the course, students will be able to:

1. Understand the fundamentals of digital image processing
2. Appreciate the advantages of compression in image /video processing
3. Understand the concepts of video formation, sampling and representation
4. Understand the principles of motion estimation in a video
5. Analyze the principles of multi-dimensional estimation with reference to a video signal.

Unit I

Fundamentals of Image Processing: Basic steps of Image processing system sampling and quantization of an Image – Basic relationship between pixels

Image Transforms: 2 – D Discrete Fourier Transform, Discrete Cosine Transform (DCT), Introduction to wavelet Transform, Continuous wavelet Transform, Discrete wavelet Transform, Filter banks

Unit II:

Image Enhancement:

Spatial Domain Methods: Histogram Processing, Fundamentals of Spatial Filtering, Smoothing Spatial filters, Sharpening Spatial filters

Frequency Domain Methods: Basics of filtering in frequency domain, image smoothing, image sharpening, selective filtering

Unit III:

Segmentation: Segmentation concepts, Point, Line and Edge Detection, Edge Linking using Hough Transform, Thresholding, Region Based segmentation.

Morphological Image Processing

Dilation and Erosion, Opening and closing, the hit or miss Transformation, Overview of Digital Image Watermarking Methods

Unit IV:

Image Compression: Image compression fundamentals – Coding Redundancy, Spatial and Temporal Redundancy. Compression Models: Lossy and Lossless, Huffmann Coding, Arithmetic Coding, LZW Coding, Run Length Coding, Bit Plane Coding, Transform Coding, Predictive Coding, Wavelet Coding, Wavelet Based Image Compression, JPEG standards.

Image Restoration: Degradation Models, PSF, Circulant And Block - Circulant Matrices, Deconvolution, Restoration Using Inverse Filtering, Wiener Filtering.

Unit V:

Basic Steps of Video Processing: Analog video, Digital Video, Time varying Image FormationModels: 3D Motion Models, Geometric Image Formation, Photometric Image Formation, Sampling of Video Signals, Filtering Operations

2-D Motion Estimation: Optical Flow, General Methodologies, Pixel Based Motion Estimation, Block Matching Algorithm, Mesh based Motion Estimation, Global Motion Estimation, Region based Motion Estimation, Multi Resolution Motion Estimation. Waveform based Coding, Block based Transform Coding, Predictive Coding, Application of Motion Estimation in video Coding. Overview of motion compensated hybrid coding (MPEG & H-264)

Text Books:

1. *Digital Image Processing*, Gonzaleze and Woods, 3rd Edition, Pearson
2. *Video Processing and Communication*, Yao Wang, Joern Ostermann and Ya-Qin Zhang, 1st Edition, Prentice Hall
3. *Digital Video Processing*, M. Tekalp, Prentice Hall International

Reference Books:

1. *Digital Signal Processing: Principles, Algorithms & Applications*, J. G. Proakis& D. G. Manolokis, 4th Edition, PHI, 2001
2. *Adaptive Filter Theory*, S. Haykin Pearson, 2003
3. *DSP–A Practical Approach*, Emmanuel C. I. Feacher, Barrie W. Jervis, 2nd Edition, Pearson Education, 2008
4. *Modern Spectral Estimation: Theory & Application*, S. M. Kay, 1988, PHI
5. *H-264 & MPEG-4 Video Compression, Video Coding for Next Generation multimedia*, I. E. Richardson, John Wiley & Sons, 2009

25ES203PC: EMBEDDED REAL TIME OPERATING SYSTEM LABORATORY (Lab – III)

M.Tech. II Sem.	**L**	**T**	**P**	**C**
	0	**0**	**4**	**2**

Course Overview

The Embedded RTOS Laboratory course is designed to provide hands-on experience and practical understanding of Real-Time Operating System (RTOS) concepts as applied in embedded systems. The course focuses on the simulation-based design and analysis of core operating system functionalities, enabling students to explore process management, inter-process communication (IPC), and I/O handling within an RTOS environment. Students will gain proficiency in using RTOS commands and system call interfaces essential for embedded application development. Through a series of structured laboratory experiments and mini-projects, the course also introduces key concepts in real-time embedded software design, task scheduling, synchronization, and resource management. By the end of the course, students will be equipped to develop and test real-time applications using industry-standard RTOS platforms.

Course Outcomes: At the end of the laboratory work, students will be able to:

1. Apply various real time concepts in building embedded systems
2. Implement the RTOS development tools in building real time embedded Systems
3. Appreciate the necessity of Inter Process Communication and Synchronization mechanisms
4. Apply the concept of RTOS in the designing of real time systems

List of Assignments:

Develop program in Linux for the following:

1. Develop a program utilizing command line arguments (argc and argv)
2. Create new process using fork( ).
3. Communicate between parent and child process using pipes.
4. Communicate between parent and child process using FIFOs
5. Develop a program to communicate between processes using message queue.
6. Develop a program using system calls that is similar to ‘cp’ command.
7. Create a new thread using POSIX Thread library
8. Develop a program to demonstrate the use of synchronizing access to shared resource using semaphores. (POSIX Thread based)
9. Develop a program to demonstrate the use of synchronizing access to shared resource using mutex. (POSIX Thread based)
10. Develop a program to demonstrate the use of signaling semaphore for sending event from one thread to another. (POSIX Thread based)

25ES204PC: IOT SYSTEM DESIGN LABORATORY (Lab – IV)

M.Tech. II Sem.	**L**	**T**	**P**	**C**
	0	**0**	**4**	**2**

Course Overview

This Internet of Things: Systems Design' course provides a structured series of lab sessions, each with defined objectives, aimed at reinforcing key IoT concepts through hands-on experience. While some foundational knowledge in networking and basic familiarity with programming (especially C/C++) is beneficial, the lab book accommodates learners of all levels by offering clear instructions, explanations, and external resources. The labs are intended to be completed within two hours and cover essential IoT practices without delving deeply into programming languages, encouraging students to independently explore and bridge any knowledge gaps.

Course Outcomes: At the end of the laboratory work, students will be able to:

1. Program microcontrollers and single-board computers for sensor and actuator interfacing.
2. Build and deploy complete IoT systems with cloud connectivity.
3. Use tools and protocols for IoT communication, including BLE, RFID, and NFC.
4. Simulate, debug, and enhance IoT applications using Raspberry Pi and related tools.

Note: Minimum of 10 Experiments have to be conducted

List of Experiments:

1. Micro-Controller Programming
2. Single-board Computer Programming
3. Posting Data to an IoT Cloud Platform
4. Connecting an IoT Gateway to an IoT Cloud
5. Connecting a Sensor Node to IoT Gateway
6. End to End Full Stack IoT Development
7. Introduction to Wireshark on Raspberry Pi
8. Programming Arduino with Blockly
9. Programming Raspberry Pi with Python
10. Bluetooth Low Energy (BLE) Based Systems
11. RFID and NFC Based Tracking
12. Multimedia Communication
13. Microcontroller Programming Simulator
14. Advance Sensors, Actuators, Components
15. 3D Objects Designing and Printing
16. Getting Started with Raspberry Pi Camera
17. Debugging the Raspberry Pi

25ES351PE: MACHINE LEARNING FOR ROBOTICS (PE – V)

M.Tech. III Sem.	L	T	P	C
	3	0	0	3

Course Overview

This course bridges the gap between machine learning techniques and robotic systems, equipping students with the knowledge and skills to develop intelligent, adaptive robots. It covers the fundamentals of robot hardware and software, data acquisition, and preprocessing for robotic applications. Students will explore core machine learning algorithms tailored to robotic data, including supervised, unsupervised, and reinforcement learning. The course also delves into training controllers through reinforcement learning and imitation learning, enabling autonomous behavior in dynamic environments. By the end of the course, students will be able to integrate machine learning models into robotic platforms for perception, control, and decision-making tasks.

Course Outcomes: Student will be able to

1. Understand and work with robot hardware and software
2. Implement, plan machine learning analysis on suitable forms of computer and roboticsdata
3. Find out more about training controllers via reinforcement learning and imitation learning
4. Integrate machine learning applications into a practical understanding of robotic models

Unit I

Introduction to Robotics: Types and Classification of robots; Science and Technology of Robots Rigid Body Transformation: Overview of Rigid Body Kinematics; Homogeneous Transformation; Link Transformation Matrices, Forward and Inverse Kinematics & Dynamics of Robots, Planning and Control of Robots,Robotic vision sensors and their interfacing

Unit II:

Fundamentals of Computer Vision: Image acquisition and representation, image transformation, filtering, restoration, morphing, Camera Models, Calibration, Single view geometry, Multiple view geometry, Epipolar geometry, RANSAC

Unit III:

Position and Orientation: Feature-based alignment; Pose estimation; Time-varying pose and trajectories, Structure from motion, dense Motion Estimation, Visual Odometry (Semi-direct VO, direct sparse odometry), Bundle Assignment

Unit IV:

Localization and Mapping: Initialization, Tracking, Mapping, geometric SLAM formulations (indirect vs. direct error formulation, geometry parameterization, sparse vs. dense model, optimization approach), Relocalization and map Optimization, Visual SLAM, Examples: Indirect (Feature based) methods (MonoSLAM, PTAM, ORB-SLAM), Direct methods (DTAM, LSD- SLAM), Sensor combinations (IMU, mono vs. Stereo, RGB-Depth), Analysis and parameter studies.

Unit V:

Recognition and Interpretations: Concepts of machine learning and deep learning, sequence modeling, Learning for robotic vision

Text Books:

1. Fu. K.S., Gonzalez R.C. and Lee C.S.G., *Robotics: Control, Sensing, Vision and Intelligence,* Tata McGraw Hill, 2008,ISE Edition.
2. Ghosal A. *Robotics: Fundamental Concepts and Analysis*, Oxford University Press, 2006, 1st Edition.

Reference Books:

1. H. R. Everett, *Sensors for Mobile Robots: Theory and Application*, A K Peters/CRC Press, 1995, 1st Edition.
2. 2. Dahiya, Ravinder S., Valle, Maurizio, *Robotic Tactile Sensing*, Springer, 2013.

25ES352PE: EDGE COMPUTING (PE-V)

M.Tech. III Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Overview

This course introduces the fundamentals of Edge Computing and its role in real-time, low-latency applications. It explores the shift from cloud to edge, emphasizing scenarios like autonomous vehicles and predictive maintenance. Students will learn key technologies such as Docker, Kubernetes, MQTT, and distributed analytics on edge data centers. The course includes hands-on exposure to frameworks for edge-cloud collaboration and time-critical data processing. It also covers AI, machine learning, and deep learning applications tailored for edge environments.

Course Outcomes: At the completion of this course, each student will be able to

1. Understand the fundamentals of edge computing, its architecture, and its role in low-latency, real-time applications
2. Analyze and implement distributed data analytics techniques and edge-cloud frameworks for intelligent data processing
3. Gain hands-on experience with edge computing technologies such as Docker, Kubernetes, MQTT, Kafka, and time synchronization
4. Apply machine learning and deep learning models in edge scenarios such as autonomous systems, predictive maintenance, and reinforcement learning

Unit I

Introduction to Cloud and its limitations to support low latency and RTT. From Cloud to Edge computing: Waves of innovation, Introduction to Edge Computing Architectures

Unit II:

Edge Computing to support User Applications (5G-Slicing, self-driving cars and more). Concepts of distributed systems in edge computing such as time ordering and clock synchronization, distributed snapshot, etc.

Unit III:

Introduction to Edge Data Center, Lightweight Edge Clouds and its services provided by different service providers. Introduction to Dockercontainers and Kubernetes in edge computing. Design of edge storage systems like key-value stores

Unit IV:

Introduction to MQTT and Kafka for an end-to-end edge pipeline. Edge analytics topologies for M2M and WSN network (MQTT).

Unit V:

Use cases of machine learning for edge sensor data in predictive maintenance, image classifiers and self-driving cars. Deep Learning On-Device inference at the edge to support latency-based applications.

Text books & Reference books:

1. *Cloud Computing: Principles and Paradigms*", Editors: Rajkumar Buyya, James Brob
2. "*Fog and Edge Computing: Principles and Paradigms*", Rajkumar Buyya (Editor), Satish Narayana Srirama (Editor), Wiley, 2019.
3. "*Cloud and Distributed Computing: Algorithms and Systems*", Rajiv Misra, Yashwant Patel, Wiley 2020.

25ES353PE: EMBEDDED BIOMEDICAL APPLICATION (PE -V)

M.Tech. III Sem.	L	T	P	C
	3	0	0	3

Course Overview

This course provides an interdisciplinary introduction to the field of biomedical engineering with a focus on embedded system applications. It begins by exploring the fundamentals of biomedical engineering, followed by an in-depth study of wearable health monitoring devices and their embedded architectures. Students will gain insights into the hardware requirements for medical image processing and the role of embedded systems in diagnostic and therapeutic applications. The course also covers various assistive and support devices commonly used in clinical settings. By the end of the course, learners will be equipped with the foundational knowledge and practical skills to develop and analyze embedded solutions for modern biomedical challenges.

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

1. Demonstrate the fundamental art of biomedical engineering.
2. Illustrate about wearable health devices and its importance.
3. Implement image processing applications using software and hardware.
4. Compare various embedded diagnostic applications.
5. Build and analyze of some biomedical equipment.

Unit I

INTRODUCTION TO BIOMEDICAL ENGINEERING: Origin of bio potential and its propagation- Resting and Action Potential – Bio signals characteristics Types of electrodes - Types of transducers and applications-Bio-amplifiers- Types of recorders, components of a biomedical system.

Unit II:

WEARABLE HEALTH DEVICES: Concepts of wearable technology in health care-Components of wearable devices- Biosensors- Blood glucose sensors - Head worn- Hand worn- Body worn- pulse oxymeter- Cardiac pacemakers – Hearing aids and its recent advancements-wearable artificial kidney.

Unit III:

EMBEDDED SYSTEM FOR MEDICAL IMAGE PROCESSING: Introduction to embedded image processing. ASIC vs FPGA - memory requirement-, power consumption-parallelism - Design issues in VLSI implementation of Image processing algorithms - interfacing. Hardware implementation of image processing algorithms: Segmentation and compression

Unit IV:

EMBEDDED SYSTEM FOR DIAGNOSTIC APPLICATIONS: ICCU patient monitoring system – ECG-EEG-EMG acquisition system-MRI scanner - CT scanner Sonography

Unit V:

CASE STUDY: Respiratory measurement using spirometer- IPPB unit for monitoring respiratory parameters - ventilators- -Defibrillator- Glucometer-Heart- Lung machine.

Reference Books:

1. Leslie Cromwell, *"Biomedical Instrumentation and Measurement",* Prentice Hall of India New Delhi, 2007.
2. John G.Webster, *"Medical Instrumentation Application and Design",* 3rd Edition, Wiley India Edition, 2007.
3. Khandpur R.S, *Handbook of Biomedical Instrumentation*, Tata McGraw Hill, New Delhi, 3rd Edition, 2014.
4. L.A Geddes and L.E.Baker, *Principles of Applied Biomedical Instrumentation,* 3rd Edition John Wiley and Sons, Reprint 2008.
5. Richard S.Cobbold, *Transducers for Biomedical Measurements; Principle and applications*John Wiley and sons, 1992.

25ES001AC:ENGLISH FOR RESEARCH PAPER WRITING
(Audit Course - I & II)

Prerequisite: None

Course objectives: Students will be able to:

- Understand that how to improve your writing skills and level of readability
- Learn about what to write in each section
- Understand the skills needed when writing a Title Ensure the good quality of paper at very first-time submission

UNIT-I:
Planning and Preparation, Word Order, Breaking up long sentences, Structuring Paragraphs and Sentences, Being Concise and Removing Redundancy, Avoiding Ambiguity and Vagueness

UNIT-II:
Clarifying Who Did What, Highlighting Your Findings, Hedging and Criticizing, Paraphrasing and Plagiarism, Sections of a Paper, Abstracts. Introduction

UNIT-III:
Review of the Literature, Methods, Results, Discussion, Conclusions, The Final Check.

UNIT-IV:
key skills are needed when writing a Title, key skills are needed when writing an Abstract, key skills are needed when writing an Introduction, skills needed when writing a Review of the Literature,

UNIT-V:
skills are needed when writing the Methods, skills needed when writing the Results, skills are needed when writing the Discussion, skills are needed when writing the Conclusions. useful phrases, how to ensure paper is as good as it could possibly be the first- time submission

TEXT BOOKS/ REFERENCES:

1. Goldbort R (2006) Writing for Science, Yale University Press (available on Google Books)
2. Day R (2006) How to Write and Publish a Scientific Paper, Cambridge University Press
3. Highman N (1998), Handbook of Writing for the Mathematical Sciences, SIAM. Highman'sbook.
4. Adrian Wallwork, English for Writing Research Papers, Springer New York DordrechtHeidelberg London, 2011

25ES002AC:DISASTER MANAGEMENT (Audit Course - I & II)

Prerequisite: None

Course Objectives: Students will be able to

- learn to demonstrate a critical understanding of key concepts in disaster risk reduction and humanitarian response.
- critically evaluate disaster risk reduction and humanitarian response policy and practice from multiple perspectives.
- develop an understanding of standards of humanitarian response and practical relevance in specific types of disasters and conflict situations.
- critically understand the strengths and weaknesses of disaster management approaches,
- planning and programming in different countries, particularly their home country or the countries they work in

UNIT-I:
Introduction:
Disaster: Definition, Factors and Significance; Difference Between Hazard and Disaster; Natural and Manmade Disasters: Difference, Nature, Types and Magnitude.
Disaster Prone Areas in India:
Study of Seismic Zones; Areas Prone to Floods and Droughts, Landslides and Avalanches; Areas Prone to Cyclonic and Coastal Hazards with Special Reference to Tsunami; Post-Disaster Diseases and Epidemics

UNIT-II:
Repercussions of Disasters and Hazards:
Economic Damage, Loss of Human and Animal Life, Destruction of Ecosystem. Natural Disasters: Earthquakes, Volcanisms, Cyclones, Tsunamis, Floods, Droughts and Famines, Landslides and Avalanches, Man-made disaster: Nuclear Reactor Meltdown, Industrial Accidents, Oil Slicks and Spills, Outbreaks of Disease and Epidemics, War and Conflicts.

UNIT-III:
Disaster Preparedness and Management:
Preparedness: Monitoring of Phenomena Triggering A Disaster or Hazard; Evaluation of Risk: Application of Remote Sensing, Data from Meteorological and Other Agencies, Media Reports: Governmental and Community Preparedness.

UNIT-IV:
Risk Assessment Disaster Risk:
Concept and Elements, Disaster Risk Reduction, Global and National Disaster Risk Situation. Techniques of Risk Assessment, Global Co-Operation in Risk Assessment and Warning, People's Participation in Risk Assessment. Strategies for Survival.

UNIT-V:
Disaster Mitigation:
Meaning, Concept and Strategies of Disaster Mitigation, Emerging Trends In Mitigation. Structural Mitigation and Non-Structural Mitigation, Programs of Disaster Mitigation in India.

TEXT BOOKS/ REFERENCES:

1. R. Nishith, Singh AK, “Disaster Management in India: Perspectives, issues and strategies “’New Royal book Company.
2. Sahni, Pardeep Et. Al. (Eds.),” Disaster Mitigation Experiences and Reflections”, Prentice Hall of India, New Delhi.
3. Goel S. L., Disaster Administration and Management Text and Case Studies”, Deep &Deep Publication Pvt. Ltd., New Delhi.

25ES003AC:SANSKRIT FOR TECHNICAL KNOWLEDGE
(Audit Course - I & II)

Prerequisite: None

Course Objectives:

1. To get a working knowledge in illustrious Sanskrit, the scientific language in the world
2. Learning of Sanskrit to improve brain functioning
3. Learning of Sanskrit to develop the logic in mathematics, science & other subjects enhancing the memory power
4. The engineering scholars equipped with Sanskrit will be able to explore the huge knowledge from ancient literature

Course Outcomes: Students will be able to

1. Understanding basic Sanskrit language
2. Ancient Sanskrit literature about science & technology can be understood
3. Being a logical language will help to develop logic in students

UNIT-I:

Alphabets in Sanskrit,

UNIT-II:

Past/Present/Future Tense, Simple Sentences

UNIT-III:

Order, Introduction of roots,

UNIT-IV:

Technical information about Sanskrit Literature

UNIT-V:

Technical concepts of Engineering-Electrical, Mechanical, Architecture, Mathematics

TEXT BOOKS/ REFERENCES:

1. “Abhyaspustakam” – Dr. Vishwas, Samskrita-Bharti Publication, New Delhi
2. “Teach Yourself Sanskrit” Prathama Deeksha-Vempati Kutumbshastri, Rashtriya Sanskrit Sansthanam, New Delhi Publication
3. “India’s Glorious Scientific Tradition” Suresh Soni, Ocean books (P) Ltd., New Delhi.

25ES004AC:VALUE EDUCATION
(Audit Course - I & II)

Prerequisite: None

Course Objectives: Students will be able to

4. Understand value of education and self- development
5. Imbibe good values in students
6. Let the should know about the importance of character

Course outcomes: Students will be able to

7. Knowledge of self-development
8. Learn the importance of Human values
9. Developing the overall personality

UNIT-I:

Values and self-development –Social values and individual attitudes. Work ethics, Indian vision of humanism. Moral and non- moral valuation. Standards and principles. Value judgements

UNIT-II:

Importance of cultivation of values. Sense of duty. Devotion, Self-reliance. Confidence, Concentration. Truthfulness, Cleanliness. Honesty, Humanity. Power of faith, National Unity. Patriotism. Love for nature, Discipline

UNIT-III:

Personality and Behavior Development - Soul and Scientific attitude. Positive Thinking. Integrity and discipline, Punctuality, Love and Kindness.

UNIT-IV:

Avoid fault Thinking. Free from anger, Dignity of labour. Universal brotherhood and religious tolerance. True friendship. Happiness Vs suffering, love for truth. Aware of self-destructive habits. Association and Cooperation. Doing best for saving nature

UNIT-V:

Character and Competence –Holy books vs Blind faith. Self-management and Good health. Science of reincarnation, Equality, Nonviolence, Humility, Role of Women. All religions and same message. Mindy our Mind, Self-control. Honesty, Studying effectively

TEXT BOOKS/ REFERENCES:

1. Chakroborty, S.K. “Values and Ethics for organizations Theory and practice”, Oxford University Press, New Delhi

25ES005AC:CONSTITUTION OF INDIA
(Audit Course - I & II)

Prerequisite: None

Course Objectives: Students will be able to:

- Understand the premises informing the twin themes of liberty and freedom from a civil rights perspective.
- To address the growth of Indian opinion regarding modern Indian intellectuals' constitutional role and entitlement to civil and economic rights as well as the emergence of nationhood in theearly years of Indian nationalism.
- To address the role of socialism in India after the commencement of the Bolshevik Revolution in 1917 and its impact on the initial drafting of the Indian Constitution.

Course Outcomes: Students will be able to:

- Discuss the growth of the demand for civil rights in India for the bulk of Indians before the arrivalof Gandhi in Indian politics.
- Discuss the intellectual origins of the framework of argument that informed the conceptualization of social reforms leading to revolution in India.
- Discuss the circumstances surrounding the foundation of the Congress Socialist Party [CSP] under the leadership of Jawaharlal Nehru and the eventual failure of the proposal of direct elections through adult suffrage in the Indian Constitution.
- Discuss the passage of the Hindu Code Bill of 1956.

UNIT-I:

History of Making of the Indian Constitution: History Drafting Committee, (Composition & Working),

Philosophy of the Indian Constitution: Preamble**,** Salient Features.

UNIT-II:

Contours of Constitutional Rights & Duties: Fundamental Rights Right to Equality**,** Right to Freedom**,** Right against Exploitation**,** Right to Freedom of Religion**,** Cultural and Educational Rights**,** Right to Constitutional Remedies, Directive Principles of State Policy**,** Fundamental Duties.

UNIT-III:

Organs of Governance: Parliament**,** Composition**,** Qualifications and Disqualifications**,** Powers and Functions**,** Executive**,** President**,** Governor**,** Council of Ministers, Judiciary, Appointment and Transfer of Judges, Qualification, Powers and Functions.

UNIT-IV:

Local Administration: District's Administration head: Role and Importance, Municipalities: Introduction, Mayor and role of Elected Representative, CEO of Municipal Corporation. Pachayati raj: Introduction, PRI: Zila Pachayat. Elected officials and their roles, CEO Zila Pachayat: Position and role. Block level: Organizational Hierarchy (Different departments), Village level: Role of Elected and Appointed officials, Importance of grass root democracy.

UNIT-V:

Election Commission: Election Commission: Role and Functioning. Chief Election Commissioner and Election Commissioners. State ElectionCommission: Role and Functioning. Institute and Bodies for the welfare of SC/ST/OBC and women.

TEXT BOOKS/ REFERENCES:

1. The Constitution of India, 1950 (Bare Act), Government Publication.
2. Dr. S. N. Busi, Dr. B. R. Ambedkar framing of Indian Constitution, 1st Edition, 2015.
3. M. P. Jain, Indian Constitution Law, 7th Edn., Lexis Nexis, 2014.
4. D.D. Basu, Introduction to the Constitution of India, Lexis Nexis, 2015.

25ES006AC:PEDAGOGY STUDIES
(Audit Course - I & II)

Prerequisite: None

Course Objectives: Students will be able to:

- Review existing evidence on the review topic to inform programme design and policy making undertaken by the DfID, other agencies and researchers.
- Identify critical evidence gaps to guide the development.

Course Outcomes: Students will be able to understand:

- What pedagogical practices are being used by teachers in formal and informal classrooms in developing countries?
- What is the evidence on the effectiveness of these pedagogical practices, in what conditions, and with what population of learners?
- How can teacher education (curriculum and practicum) and the school curriculum and guidance materials best support effective pedagogy?

UNIT-I:

Introduction and Methodology: Aims and rationale, Policy background, Conceptual framework and terminology Theories of learning, Curriculum, Teacher education. Conceptual framework, Research questions**.** Overview of methodology and Searching.

UNIT-II:

Thematic overview: Pedagogical practices are being used by teachers in formal and informal classrooms in developing countries. Curriculum, Teacher education.

UNIT-III:

Evidence on the effectiveness of pedagogical practices, Methodology for the indepth stage: quality assessment of included studies. How can teacher education (curriculum and practicum) and the scho curriculum and guidance materials best support effective pedagogy? Theory of change. Strength and nature of the body of evidence for effective pedagogical practices. Pedagogic theory and pedagogical approaches. Teachers' attitudes and beliefs and Pedagogic strategies.

UNIT-IV:

Professional development: alignment with classroom practices and follow-up support, Peer support, Support from the head teacher and the community. Curriculum and assessment, Barriers to learning: limited resources and large class sizes

UNIT-V:

Research gaps and future directions: Research design**,** Contexts**,** Pedagogy**,** Teacher education**,** Curriculum and assessment**,** Dissemination and research impact.

TEXT BOOKS/ REFERENCES:

1. Ackers J, Hardman F (2001) Classroom interaction in Kenyan primary schools, Compare, 31(2): 245-261.
2. Agrawal M (2004) Curricular reform in schools: The importance of evaluation, Journal of Curriculum Studies, 36 (3): 361-379.
3. Akyeampong K (2003) Teacher training in Ghana - does it count? Multi-site teacher

Education research project (MUSTER) country report 1. London: DFID.

4. Akyeampong K, Lussier K, Pryor J, Westbrook J (2013) Improving teaching and learning of basic maths and reading in Africa: Does teacher preparation count? International Journal Educational Development, 33 (3): 272–282.
5. Alexander RJ (2001) Culture and pedagogy: International comparisons in primary education. Oxford and Boston: Blackwell.
6. Chavan M (2003) Read India: A mass scale, rapid, ‘learning to read’ campaign.
7. www.pratham.org/images/resource%20working%20paper%202.pdf.

25ES007AC STRESS MANAGEMENT BY YOGA
(Audit Course - I & II)

Prerequisite: None

Course Objectives:

- To achieve overall health of body and mind
- To overcome stress

Course Outcomes: Students will be able to:

- Develop healthy mind in a healthy body thus improving social health also
- Improve efficiency

UNIT-I:
Definitions of Eight parts of yog. (Ashtanga)

UNIT-II:
Yam and Niyam.

UNIT-III:
Do`s and Don't's in life.
i) Ahinsa, satya, astheya, bramhacharya and aparigraha
ii) Shaucha, santosh, tapa, swadhyay, ishwarpranidhan

UNIT-IV:
Asan and Pranayam

UNIT-V:
i) Various yog poses and their benefits for mind & body
ii) Regularization of breathing techniques and its effects-Types of pranayam

TEXT BOOKS/ REFERENCES:

1. 'Yogic Asanas for Group Tarining-Part-I": Janardan Swami Yogabhyasi Mandal, Nagpur
2. "Rajayoga or conquering the Internal Nature" by Swami Vivekananda, Advaita Ashrama (Publication Department), Kolkata

25ES008AC:PERSONALITY DEVELOPMENT THROUGH LIFE ENLIGHTENMENT SKILLS
(Audit Course - I & II)

Prerequisite: None

Course Objectives:

1. To learn to achieve the highest goal happily
2. To become a person with stable mind, pleasing personality and determination
3. To awaken wisdom in students

Course Outcomes: Students will be able to

1. Study of Shrimad-Bhagwad-Geeta will help the student in developing his personality and achieve the highest goal in life
2. The person who has studied Geeta will lead the nation and mankind to peace and prosperity
3. Study of Neetishatakam will help in developing versatile personality of students

UNIT-I:

Neetisatakam-Holistic development of personality

1. Verses- 19,20,21,22 (wisdom)
2. Verses- 29,31,32 (pride & heroism)
3. Verses- 26,28,63,65 (virtue)

UNIT-II:

Neetisatakam-Holistic development of personality

1. Verses- 52,53,59 (dont's)
2. Verses- 71,73,75,78 (do's)

UNIT-III:

Approach to day to day work and duties.

1. Shrimad Bhagwad Geeta: Chapter 2-Verses 41, 47,48,
2. Chapter 3-Verses 13, 21, 27, 35, Chapter 6-Verses 5,13,17, 23, 35,
3. Chapter 18-Verses 45, 46, 48.

UNIT-IV:

Statements of basic knowledge.

1. Shrimad Bhagwad Geeta: Chapter2-Verses 56, 62, 68
2. Chapter 12 -Verses 13, 14, 15, 16,17, 18
3. Personality of Role model. Shrimad Bhagwad Geeta:

UNIT-V:

1. Chapter2-Verses 17, Chapter 3-Verses 36,37,42,
2. Chapter 4-Verses 18, 38,39
3. Chapter18 – Verses 37,38,63

TEXT BOOKS/ REFERENCES:

1. "Srimad Bhagavad Gita" by Swami Swarupananda Advaita Ashram (Publication Department),Kolkata.
2. Bhartrihari's Three Satakam (Niti-sringar-vairagya) by P.Gopinath, Rashtriya Sanskrit Sansthanam, New Delhi.