MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act,1956

M.TECH in EMBEDDED SYSTEMS

Course Structure (MLRS-MT25)

Applicable From 2025-26 Admitted Batch

S. No	Category	Breakup of credits (Total 68 credits)
1	Programme Core	20
2	Professional Elective	15
3	Open Elective	03
4	Mandatory Credit Course	02
5	Project Work	28
6	Audit Course	-
	Total Credits	**68**

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act,1956

M.TECH in EMBEDDED SYSTEMS

Course Structure (MLRS-MT25)

Applicable From 2025-26 Admitted Batch

I YEAR I SEMESTER

S. No.	Course Code	Course Name	Course Area	Periods per week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIE)	External (SEE)	Total
1	2515501	FPGA Based System Design	PC	3	0	0	3	40	60	100
2	2515502	Embedded System Design	PC	3	0	0	3	40	60	100
3		Professional Elective - I	PE	3	0	0	3	40	60	100
4		Professional Elective - II	PE	3	0	0	3	40	60	100
5	2515571	FPGA Based System Design Laboratory	PC	0	0	4	2	40	60	100
6	2515572	Embedded System Design Laboratory	PC	0	0	4	2	40	60	100
7	2511345	Research Methodology & IPR	MC	2	0	0	2	40	60	100
8		Audit Course - I	AC	2	0	0	0	100	-	100
TOTAL				**16**	**0**	**8**	**18**	**380**	**420**	**800**

I YEAR II SEMESTER

S. No.	Course Code	Course Name	Course Area	Periods per week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIE)	External (SEE)	Total
1	2525503	Embedded Real Time Operating Systems	PC	3	0	0	3	40	60	100
2	2525504	IoT System Design	PC	3	0	0	3	40	60	100
3		Professional Elective - III	PE	3	0	0	3	40	60	100
4		Professional Elective - IV	PE	3	0	0	3	40	60	100
5	2525573	Embedded Real Time Operating System Laboratory	PC	0	0	4	2	40	60	100
6	2525574	IoT System Design Laboratory	PC	0	0	4	2	40	60	100
7	2525525	Mini Project with Seminar	PS	0	0	4	2	100	-	100
8		Audit Course - II	AC	2	0	0	0	100	-	100
TOTAL				**14**	**0**	**12**	**18**	**440**	**360**	**800**

II YEAR I SEMESTER

S. No.	Course Code	Course Name	Course Area	Periods per week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIE)	External (SEE)	Total
1		Professional Elective - V	PE	3	0	0	3	40	60	100
2		Open Elective	OE	3	0	0	3	40	60	100
3	2535526	Dissertation Work Review-I	PS	0	0	18	6	100	0	100
TOTAL				**6**	**0**	**18**	**12**	**180**	**120**	**300**

II YEAR II SEMESTER

S. No.	Course Code	Course Name	Course Area	Periods per week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIE)	External (SEE)	Total
1	2535527	Dissertation Work Review-II	PS	0	0	18	6	100	0	100
2	2535528	Dissertation Viva - Voce	PS	0	0	42	14	0	100	100
TOTAL				**0**	**0**	**40**	**20**	**100**	**100**	**200**

Professional Elective (PE) Courses

Professional Elective-I

S. No.	Course Code	Course Title
1	2515505	CMOS VLSI Design
2	2515506	Wireless Sensor Networks
3	2515507	Advanced Computer Architecture

Professional Elective-II

S. No.	Course Code	Course Title
1	2515508	Machine Learning and Deep Learning
2	2515509	Advanced RISC Architectures
3	2515510	Automotive Embedded Systems

Professional Elective-III

S. No.	Course Code	Course Title
1	2525511	GPU Architectures
2	2525512	VLSI Test and Testability
3	2525513	Hardware Software Co-Design

Professional Elective-IV

S. No.	Course Code	Course Title
1	2525514	Hardware Security in VLSI Design
2	2525515	Hardware Accelerators for Machine Learning Models
3	2525516	Image and Video Processing

Professional Elective-V

S. No.	Course Code	Course Title
1	2535517	Machine Learning for Robotics
2	2535518	Edge Computing
3	2535519	Embedded Biomedical Applications

Audit Course – I		
S. No.	**Course Code**	**Course Title**
1	2511346	English for Research Paper Writing
2	2511347	Disaster Management
3	2511348	Sanskrit for Technical Knowledge
4	2511349	Value Education

Audit Course – II		
S. No.	**Course Code**	**Course Title**
1	2521350	Constitution of India
2	2521351	Pedagogy Studies
3	2521352	Stress Management by Yoga
4	2521353	Personality Development Through Life Enlightenment Skills

Open Elective (OE) Course

S. No.	Course Code	Course Title
1	2535520	Business Analytics
2	2535521	Industrial Safety
3	2535522	Operations Research
4	2535523	Cost Management of Engineering Projects
5	2535524	Composite Materials

***Open Elective** – Students should take Open Electives from list of Open Electives offered by other Departments/Branches Only.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

I-I

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2515501: FPGA Based System Design

I Year M.Tech. ES I – Sem. **L T P C**

3 0 0 3

Couse Overview:
This course covers FPGA-based system design with emphasis on digital logic, system architecture, and modern methodologies. Students gain hands-on experience with HDLs, combinational and sequential logic, and advanced topics like platform and multi-FPGA systems. It equips learners with the knowledge and skills to design and optimize embedded digital systems, bridging traditional logic design and reconfigurable computing.

Pre-requisites: Digital System Design.

Course Objectives:
The students will try to learn
- FPGA architectures and their role in modern digital system design
- Hardware description languages (Verilog HDL) for modeling and implementing combinational and sequential circuits
- Skills in applying design methodologies for behavioral modeling, simulation, and verification.
- Advanced FPGA design concepts including finite state machines, platform FPGAs, and multi-FPGA systems
- Enable learners to analyze performance trade-offs in terms of delay, power, and resource utilization for FPGA-based designs

Course Outcomes:
After successful completion of the course, students shall be able to
- Explain the fundamental concepts, architectures, and design principles of FPGA-based systems, including FPGA fabrics, programming technologies, and chip I/O structures.
- Apply combinational logic design techniques using Hardware Description Languages (Verilog) for implementing optimized arithmetic and logic circuits on FPGAs.
- Analyze sequential logic circuits and finite state machines in FPGAs, considering clocking rules, performance, and power optimization.
- Assess different FPGA design methodologies and behavioral architectures for selecting suitable design approaches based on given application requirements.
- Formulate system-level FPGA solutions incorporating buses, platform FPGAs, multi-FPGA systems, and novel architectures for advanced applications.

Module – I: Introduction to FPGA-Based Systems **10L**

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.

Module – II: Combinational Logic Design in FPGAs **9L**

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.

Module – III: Sequential Logic and State Machines **7L**

Introduction, the sequential machine design process, sequential design styles, rules for clocking,

performance analysis, power optimization

Module – IV: Design Methodologies and Behavioral Architecture **7L**

Introduction, behavioral design, design, methodologies, design example

Module – V: System-Level Design and Advanced FPGA Applications **6L**

Introduction, buses, platform FPGAs, multi-FPGA systems, Novel architectures

TEXT BOOKS:

1. Wolf, Wayne. FPGA-Based System Design. Pearson Education India, 2005. 2009.

REFERENCES:

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

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2515502: EMBEDDED SYSTEM DESIGN

I Year M.Tech. ES I – Sem. **L T P C**

3 0 0 3

Couse Overview:
This course covers the fundamentals of Embedded Systems, covering their architecture, components, and applications. It explores firmware design, RTOS concepts, task scheduling, communication, and synchronization. Students will gain insights into hardware-software integration and design techniques for real-time embedded applications.

Pre-requisites: Knowledge on computer architecture, microprocessors and microcontrollers, digital electronics, and operating system fundamentals.

Course Objectives:
The students will try to learn
- The fundamentals, characteristics, and applications of embedded systems
- Understand core components like processors, memory, sensors, and interfaces
- Explore embedded firmware design and supporting hardware circuits
- Study RTOS concepts including tasks, threads, and scheduling
- Learn task communication, synchronization, device drivers, and RTOS selection

Course Outcomes:
By the end of this course, students will be able to
- Explain the fundamentals, characteristics, and applications of embedded systems.
- Apply knowledge of processors, memory, sensors, and communication interfaces in embedded systems.
- Interpret embedded firmware components and associated development approaches.
- Evaluate RTOS concepts, task models, and scheduling techniques.
- Design task communication and synchronization mechanisms using appropriate RTOS features.

Module I: Introduction to Embedded Systems **10L**
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.

Module II: Typical Embedded System: **10L**
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.

Module III: Embedded Firmware: **7L**
Reset Circuit, Brown-out Protection Circuit, Oscillator Unit, Real Time Clock, Watchdog Timer, Embedded Firmware Design Approaches and Development Languages.

Module IV: RTOS Based Embedded System Design: **7L**
Operating System Basics, Types of Operating Systems, Tasks, Process and Threads, Multiprocessing and Multitasking, Task Scheduling.

Module –V: Task Communication: **8L**

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.
2. Embedded Systems-RajKamal, TMH.

REFERENCES:

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

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2515505: CMOS VLSI Design
(Professional Elective - I)

I Year M.Tech. ES I – Sem. **L T P C**

3 0 0 3

Couse Overview:

Introduces MOS transistors, CMOS technology, and fabrication basics. Covers digital logic design uing CMOS gates and circuits. Explains layout, architecture, and hierarchy in VLSI systems. Focuses on performance trade-offs like power, delay, and area. Highlights CAD tools, testing, and applications in modern ICs.

Course Objectives:

The students will try to learn

- The CMOS transistor operation and fabrication process
- The combinational and sequential digital circuits using CMOS logic
- Layout techniques and VLSI system architectures
- The power, speed, and area trade-offs in circuit design
- The CAD tools for VLSI design, verification, and testing

Course Outcomes:

After successful completion of the course, students shall be able to

- Explain MOS inverter characteristics and pseudo-NMOS and CMOS logic behavior.
- Construct combinational MOS logic circuits comprising CMOS gates, adders, transmission-gate–based logic.
- Evaluate the operation of sequential MOS circuits based on latches and flip-flops.
- Evaluate dynamic logic circuit principles and high-performance CMOS designs.
- Explain the organization, operation, and leakage mechanisms of semiconductor memory technologies.

Module – I: MOS Design **10L**

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.

Module – II: Combinational MOS logic circuits **10L**

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.

Module – III: Sequential MOS logic circuits **7L**

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

Module – IV: Dynamic Logic Circuits **8L**

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

Module – V: Semiconductor Memories **8L**

Types, RAM array Organization, DRAM types, operation, leakage currents in 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 MoKang, Yusuf Leblebici, TMH, 3rdEd., 2011.

REFERENCES:

1. Introduction to VLSI Systems: A Logic, Circuit and System Perspective- Ming Bol in, CRCPress,2011.
2. Digital Integrated Circuits: A Designs Perspective- Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolic, 2nd Ed., PHI.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2515506: WIRELESS SENSOR NETWORKS
(Professional Elective - I)

I Year M.Tech. ES I – Sem. **L T P C**

3 0 0 3

Couse Overview:

The course Wireless Sensor Networks (WSN) covers the design, development, and applications of sensor networks. It introduces fundamental concepts, protocols, architectures, and algorithms. Students learn to analyze and solve real-world problems in IoT, smart cities, and automation. The course blends theory with practical implementation, focusing on energy efficiency, communication, and data processing. It also highlights current trends and research challenges for advanced study and industry applications.

Course Objectives:

The students will try to learn

- About 3D vector co-ordinate systems and electromagnetic field concepts
- Skills in selecting appropriate Maxwell's equations in electromagnetic theory for a given application and analyze the problem
- The propagation characteristics of electromagnetic waves at boundary of different media
- The ability to compute various parameters for transmission lines using smith chart and classical theory
- Various line parameters by conventional and graphical methods

Course Outcomes:

After successful completion of the course, students shall be able to

- Explain the fundamentals, advantages, types, and applications of wireless sensor networks.
- Analyze enabling technologies, issues, and challenges in wireless sensor networks and MANETs.
- Implement routing and MAC protocols for wireless sensor networks using IEEE 802.15.4 and ZigBee standards.
- Evaluate data dissemination, real-time support, and security mechanisms in sensor networks.
- Develop WSN architectures with gateways, node hardware, and operating environments using TinyOS.

Module – I: Electrostatics **8L**

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

Module – II: **8L**

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

Module – III: **7L**

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

Module – IV: **7L**

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

Module – V: **10L**

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.

TEXT BOOKS:

1. E.C. Jordan, K.G. Balmain, “Electromagnetic waves and Radiating Systems,” PHI 2nd Edition, 2000.
2. Matthew N.O. Sadiku, “Elements of Electromagnetics,” Oxford University Press, 4th Edition, 2009.

REFERENCES:

1. William H. Hayt Jr., John A. Buck, “Engineering electromagnetic,” Tata McGraw Hill, 7th Edition, 2006.
2. Nathan Ida, “Engineering Electromagnetic,’’ Springer (India) Pvt. Ltd, 2nd Edition, 2005
3. G. Sashibushana Rao, “Electromagnetic field theory and Transmission lines,’’ Wiley (India) 1st Edition, 2013.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2515507: ADVANCED COMPUTER ARCHITECTURE
(Professional Elective - I)

I Year M.Tech. ES I – Sem. **L T P C**

3 0 0 3

Couse 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:

The students will try to learn

- Fundamentals of computer design, technology trends, performance metrics, and instruction set principles
- Pipelining concepts, hazards, and memory hierarchy designs for improving processor performance
- Instruction-level parallelism using both hardware and software approaches for efficient execution
- Multiprocessor architectures and thread-level parallelism for high-performance computing
- Interconnection networks, cluster design, and Intel IA-64 architecture while identifying common fallacies and pitfalls

Course Outcomes:

After successful completion of the course, students shall be able to

- Explain fundamental principles of computer design, instruction set architecture, and performance evaluation.
- Analyze pipelined processor design and memory hierarchy techniques for performance improvement.
- Apply hardware and software approaches to exploit instruction-level parallelism.
- Evaluate multiprocessor and thread-level parallel architectures and synchronization mechanisms.
- Design interconnection networks and clustered systems based on modern Intel architectures.

Module – I: Fundamentals of Computer Design **10L**

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.

Module – II: Pipelines, Memory Hierarchy Design **8L**

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.

Module – III: Instruction Level Parallelism the Hardware Approach, ILP Software Approach 7L

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.

Module – IV: Multi Processors and Thread Level Parallelism 7L

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.

Module – V: Inter Connection and Networks, Intel Architecture 6L

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 BOOKS:

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

REFERENCES:

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

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2515508: MACHINE LEARNING AND DEEP LEARNING
(Professional Elective - II)

I Year M.Tech. ES I – Sem. **L T P C**

3 0 0 3

Couse 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.

Course Objectives:

The students will try to learn

- Fundamentals of machine learning, including learning paradigms, bias–variance tradeoff, and probabilistic approaches
- Designing and training of deep feedforward neural networks using gradient-based methods and backpropagation
- Regularization and optimization techniques for improving deep learning performance and generalization
- The CNNs, RNNs, LSTMs, and other architectures for image, sequence, and structured data modeling
- Deep learning methodologies to real-world applications with effective tuning, evaluation, and debugging strategies

Course Outcomes:

After successful completion of the course, students shall be able to

- Explain fundamental machine learning concepts, learning algorithms, and challenges motivating deep learning.
- Apply deep feedforward network architectures and backpropagation for model training.
- Analyze regularization and optimization techniques for training deep neural networks.
- Design convolutional and sequence models such as CNNs and RNN/LSTM for structured data.
- Evaluate deep learning models using appropriate performance metrics and apply them to real-world applications.

Module – I: Machine Learning Basics **10L**

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.

Module – II: Deep Feedforward Networks **7L**

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

Module – III: Regularization for Deep Learning, Optimization **10L**

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.

Module – IV: Convolutional Neural Networks, Sequence Modeling 10L
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.

Module – V: Practical Methodology , Applications 7L
Practical Methodology: Performance metrics, selecting hyperparameters, debugging strategies. Applications: Large-scale deep learning, computer vision, speech recognition, natural language processing, other applications.

TEXT BOOKS:

1. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.
2. Neural Networks and Deep Learning Michael A. Nielsen – 2015.

REFERENCES:

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.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2515509: ADVANCED RISC ARCHITECTURES
(Professional Elective - II)

I Year M.Tech. ES I – Sem. **L T P C**

3 0 0 3

Couse 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 Objectives:

The students will try to learn

- Computer architectures, including RISC, CISC, and MIPS, along with basic Verilog-based hardware design concepts
- Processor pipelining, including pipeline stages, hazards, performance issues, and branch prediction techniques
- Memory hierarchy design, cache architectures, mapping techniques, replacement policies, and optimization strategies
- The RISC-V Instruction Set Architecture, registers, instruction formats, exceptions, interrupts, and control/status registers
- The VEGA THEJAS32 microcontroller, its functional blocks, memory-mapped I/O, and interrupt handling using the ARIES development board

Course Outcomes:

After successful completion of the course, students shall be able to

- Explain fundamental computer design techniques, RISC–CISC architectures, and MIPS processor datapath and control.
- Apply pipelining concepts and hazard mitigation techniques to improve processor performance.
- Analyze memory hierarchy design and cache optimization techniques.
- Explain RISC-V instruction set architecture, programming model, and machine-level control registers.
- Design and implement embedded applications using the VEGA THEJAS32 microcontroller and ARIES development board.

Module – I: Fundamental techniques of computer design, MIPS processor **10L**

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.

Module – II: Processor Pipelining **10L**

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.

Module – III: Memory hierarchy **10L**

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.

Module – IV: RISC-V Architecture **9L**

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.

Module – V: VEGA THEJAS32 Microcontroller **7L**

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

TEXT BOOKS:

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.

REFERENCES:

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

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2515510: AUTOMOTIVE EMBEDDED SYSTEMS
(Professional Elective - II)

I Year M.Tech. ES I – Sem. **L T P C**

3 0 0 3

Couse 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.

Pre-requisites: Basic Electronics and Electrical Engineering, Computer Architecture and Embedded Systems.

Course Objectives:

The students will try to learn

- Functional Domains and Standards in Automotive Systems
- Intelligent Vehicle Technologies and Embedded Communication Protocols
- Dependable Automotive Networks
- Software product-line concepts, manage variability, and support the reuse of software across multiple automotive platforms.
- Model and Develop Embedded Automotive Architectures

Course Outcomes:

After successful completion of the course, students shall be able to

- Explain vehicle functional domains, safety requirements, and the role of AUTOSAR in automotive embedded systems.
- Analyze intelligent vehicle technologies, automotive communication protocols, and middleware solutions.
- Apply dependable automotive networking protocols such as CAN and FlexRay for fault-tolerant communication.
- Evaluate software product line engineering and software reuse strategies in automotive electronics.
- Design automotive embedded systems using architecture description languages and model-based development approaches.

Module – I: Vehicle Functional Domains and Their Requirements **10L**

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.

Module – II: **11L**

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.

Module – III: **9L**

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.

Module – IV: **9L**

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 Terminolog: 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.

Module – V: **9L**

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.

REFERENCES:

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.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2515571: FPGA BASED SYSTEM DESIGN LABORATORY

I Year M.Tech. ES I – Sem. **L T P C**

0 0 4 2

Couse Overview:

This lab course provides hands-on experience with FPGA design using Verilog/VHDL. Students will design, simulate, and implement digital building blocks, sequential circuits, FSMs, and communication modules on FPGA platforms. The course emphasizes simulation, verification, and hardware validation to connect theory with real-world digital system design.

Pre-requisites: Basic concepts of microprocessors and microcontrollers**,** VLSI.

Course Objectives:

The students will try to learn

- Writing and simulating HDL code for basic combinational and sequential digital circuits
- Designing and implementing complex digital systems such as FSMs, ALUs, and communication modules on FPGA platforms
- Developing and verifying interfacing protocols for external hardware devices including displays, keypads, and UARTs
- Applying design methodologies for timing analysis, synchronization, and resource optimization in FPGA-based systems
- How to Deploying, debugging, and validating digital designs through hardware testing on FPGA development boards

Course Outcomes:

After successful completion of the course, students shall be able to

- Apply Verilog HDL to design, simulate, and implement basic combinational and sequential digital circuits on FPGA.
- Analyze the functionality and timing behavior of arithmetic, logic, and control circuits using simulation tools and FPGA hardware.
- Design and implement finite state machine–based control systems for real-time digital applications.
- Interface FPGA-based designs with external peripherals such as LEDs, seven-segment displays, UART, and PWM-controlled devices.
- Develop and verify FPGA-based systems for data processing and real-time applications, including counters, filters, and clocks.

List of 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. 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.

NOTE: Minimum of 10 experiments to be conducted.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2515572: EMBEDDED SYSTEM DESIGN LABORATORY

I Year M.Tech. ES I – Sem.

L	T	P	C
0	0	4	2

Couse Overview:
The Embedded System Design Laboratory provides hands-on experience in configuring and testing embedded devices. Students learn to program GPIOs, interface sensors and actuators, and develop applications for automation, monitoring, and control. The course also introduces advanced concepts like wireless communication, web hosting, media streaming, and firmware customization, emphasizing practical system integration and debugging skills.

Pre-requisites: Basic knowledge of embedded systems, microcontrollers, and programming concepts.

Course Objectives:
The students will try to learn:

- Flashing and testing embedded devices to achieve stable functional states
- Programming and interfacing GPIO, LEDs, switches, and other peripherals
- Developing small automation and simulation projects like dice games, light-based control, and battery monitoring
- Networking, remote display, and web-based application implementation on embedded devices
- Exploring advanced applications such as webcam server, and device integration

Course Outcomes:
By the end of this course, students Shall be able to:

- Perform functional testing and operating system deployment on embedded devices to achieve stable operation.
- Configure remote access and display sharing using networking tools such as SSH and X11.
- Develop GPIO-based applications to interface sensors, LEDs, switches, and peripheral devices.
- Design embedded applications using sensors and actuators for automation and monitoring tasks.
- Implement network-enabled applications including web hosting, multimedia streaming, and wireless communication on embedded platforms.

List of Experiments:

1. **Functional Testing of Devices:**
   Flashing the OS onto the device raspberry-pi 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 raspberry-pi using native programming language. Interfacing of I/O devices like LED/Switch etc., and testing the functionality.
4. **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.

5. **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 for 1–2V, 1 LED for 0.1–1V & turn off all for 0V).
6. **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.
7. **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.
8. **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 thereby host the website.
9. **Webcam Server:**
   Interfacing the regular USB webcam with the device and turn it into fully functional IP webcam & test the functionality**.**
10. **FM Transmission:**
    Transforming the device into a regular fm transmitter capable of transmitting audio at desired frequency.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

I-II

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2525503: EMBEDDED REAL TIME OPERATING SYSTEM

I Year M.Tech. ES II – Sem. **L T P C**

3 0 0 3

Couse Overview:
This course provides an in-depth understanding of real-time operating systems used in embedded systems. It covers RTOS architecture, scheduling, task management, synchronization, and memory handling. The course also explores practical design challenges like deadlocks, priority inversion.

Course Objectives:
The students will try to learn
- Fundamentals and architecture of Real-Time Operating Systems
- Task scheduling, management, and inter-task communication
- Synchronization techniques using semaphores and message queues
- Timer services, memory management, and interrupt handling
- How to address common design challenges in RTOS-based applications

Course Outcomes:
After successful completion of the course, students shall be able to
- Explain the fundamentals of real-time operating systems, their components, and scheduling mechanisms.
- Apply RTOS primitives in task management, synchronization, and inter-task communication.
- Analyze exception handling, interrupt mechanisms, and timer services in real-time systems.
- Evaluate memory management strategies and the role of MMUs in RTOS-based systems.
- Design real-time applications addressing concurrency, scheduling analysis, and resource management issues.

Module – I: **9L**
Introduction to RTOS, Embedded systems, real-time systems, RTOS components, history, scheduler, and objects.

Module – II: **8L**
Tasks, Semaphores, and Message Queues, Task definitions, states, scheduling, synchronization, semaphores, message queues.

Module – III **7L**
Exceptions, Interrupts, and Timer Services Exception handling, interrupts, timers, programmable interval timers, soft timers.

Module – IV: **6L**
Memory Management Dynamic and fixed-size memory allocation, blocking/non-blocking functions, MMUs.

Module – V: **6L**
Application Design and Design Problems Modularization, concurrency, rate-monotonic analysis, deadlocks, resource management, priority inversion.

TEXT BOOKS:

1. Jean J. Labrosse – MicroC/OS-II: The Real-Time Kernel, CMP Books.
2. Raj Kamal – Embedded Systems: Architecture, Programming and Design, McGraw-Hill.

REFERENCES:

1. Krishna & Shin – Real-Time Systems, McGraw-Hill.
2. David E. Simon – An Embedded Software Primer, Addison-Wesley.
3. Tanennbaum A. S. – Modern Operating Systems, Pearson.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2525504: IoT SYSTEM DESIGN

I Year M.Tech. ES II – Sem. **L T P C**

3 0 0 3

Couse 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.

Course Objectives:
The students will try to learn
- Fundamental concepts, evolution, applications, and enabling technologies of the Internet of Things.
- Differences, similarities, and integration aspects between Machine-to-Machine communication and IoT, along with value chains and global perspectives.
- Basics of Arduino and Raspberry Pi platforms for integrating sensors, actuators, and programming in IoT applications.
- Architectural components, layers, and functional stacks of IoT systems, including data management and computation.
- Design challenges, emerging pillars, and domain-specific applications of IoT such as agriculture, healthcare, transportation, and smart cities.

Course Outcomes:
After successful completion of the course, students shall be able to
- Explain fundamental concepts, characteristics, and enabling technologies of the Internet of Things.
- Analyze the relationship between M2M and IoT, covering value chains and supporting technologies.
- Develop basic IoT applications using Arduino, Raspberry Pi, and sensor–actuator integration.
- Compare IoT architectures and data management frameworks.
- Design IoT systems for real-world applications such as agriculture, healthcare, and smart cities.

Module – I: IoT introduction **7L**
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.

Module – II: IoT and M2M **8L**
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.

Module – III: IoT Hands-on **8L**
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.

Module – IV: IoT Architecture **7L**

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

Module – V: IoT System design **7L**

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

REFERENCES:

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

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2525511: GPU Architectures
(Professional Elective-III)

I Year M.Tech. ES II – Sem. **L T P C**

3 0 0 3

Couse 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

Pre-requisites: C/C++ programming, Basic Computer Architecture

Course Objectives:

The students will try to learn

- Comparison of CPU and GPU architectures and explain the principles of parallelism in GPU computing
- GPU-based algorithms such as matrix operations, histo-gramming, reductions, scans, and N-body simulations
- Optimization techniques like memory coalescing, thread/block tuning, and multi-streaming to improve performance
- CUDA libraries (cuBLAS, cuFFT, cuRAND, Thrust) and tools for efficient GPU programming and multi-GPU scaling
- Deep learning models and graphics applications (e.g., ray tracing, collision detection) using GPU computing platforms

Course Outcomes:

After successful completion of the course, students shall be able to

- Explain GPU architectures, parallelism concepts, and the GPU programming model.
- Apply CUDA programming techniques in developing parallel programs on GPUs.
- Implement parallel primitives and algorithms with CUDA for scientific and data-intensive applications.
- Optimize GPU applications using memory, execution, and scaling techniques and GPU libraries.
- Develop GPU-accelerated applications for deep learning, graphics, and advanced visualization.

Module I: GPU architectures **7L**

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

Module II: CUDA programming **8L**

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

Module III: Parallel primitives and algorithms on GPU **9L**

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.

Module IV: Optimizing GPU Applications **9L**

Coalesced Memory Transactions, Grid Blocks, Thread Blocks, domain decomposition, Asynchronous Kernels and Multi-streaming 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.

Module V: Deep learning on GPUs **10L**

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.

TEXTBOOKS:

1. **"**Programming Massively Parallel Processors: A Hands-on Approach" David B. Kirk, Wen-mei W. Hwu, 4th Edition (2022).
2. "CUDA by Example: An Introduction to General-Purpose GPU Programming" Jason Sanders, Edward Kandrot, 1st Edition (2010).

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.
3. "GPU Parallel Program Development Using CUDA" Tolga Soyata, 1st Edition (2018).

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2525512: VLSI TEST AND TESTABILITY
(Professional Elective-III)

I Year M. Tech. ECE II – Sem. **L T P C**

3 0 0 3

Couse Overview:
This course provides an in-depth understanding of testing and testability techniques in VLSI circuit design. It covers fault modeling, test generation algorithms, design-for-testability (DFT) methods, and advanced topics such as built-in self-test (BIST) and boundary scan. Students will learn how to identify, model, and detect faults in digital integrated circuits and apply practical DFT strategies to improve the quality and reliability of VLSI systems.

Pre-requisites: Digital Logic Design (combinational & sequential circuits), VLSI Design Fundamentals (CMOS logic, design flow).

Course Objectives:

The students will try to learn

- Basic concepts of fault modeling and testing in VLSI circuits
- Test pattern generation techniques for different fault models
- Design-for-testability (DFT) methods for improved test coverage
- Knowledge on BIST, boundary scan, and test compression
- Analysing and design testable VLSI systems

Course Outcomes:

- Explain fault models, testing principles, and simulation techniques for digital circuits.
- Apply test generation methods for combinational and sequential circuits with design-for-testability concepts.
- Analyze scan-based and system-level design-for-testability approaches.
- Implement self-test architectures and test algorithms for logic and memory circuits.
- Evaluate fault diagnosis techniques and self-checking mechanisms in digital systems.

Module – I: Basics of Testing and Fault Modeling: **9L**
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.

Module – II: **Test Generation for Combinational and Sequential Circuits:** **8L**
Test generation for combinational logic circuits - Testable combinational logic circuit design - Test generation for sequential circuits - design of testable sequential circuits. Design For Testability.

Module – III: Design for Testability **10L**
Ad-hoc design - Generic scan-based design - Classical scan-based design – System level DFT approaches.

Module – IV: Self-Test and Test Algorithms **9L**
Built-In Self-Test - Test pattern generation for BIST - Circular BIST - BIST Architectures - Testable Memory Design - Test algorithms - Test generation for Embedded RAMs.

Module – V: Fault Diagnosis **10L**
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
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

REFERENCES:

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.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2525513: HARDWARE SOFTWARE CO-DESIGN
(Professional Elective-III)

I Year M.Tech. ECE II – Sem. **L T P C**

3 0 0 3

Couse Overview:
This course provides a comprehensive understanding of Hardware–Software Co-Design principles used in embedded system and System-on-Chip (SoC) development. It covers methodologies for partitioning system functionality between hardware and software, exploring trade-offs in performance, cost, and power. Students will learn co-design techniques, system specification models, co-simulation, co-verification, and modern design flows used in real-time embedded applications.

Pre-requisites: Digital System Design, Microprocessors/Microcontrollers

Course Objectives:
The students will try to learn
- Hardware–software co-design concepts, methodologies, and design flow
- Specification models, system-level design, and hardware–software partitioning techniques
- Expertise knowledge on co-simulation, co-verification, and co-synthesis tools
- Trade-offs between hardware and software implementation in embedded systems
- Designing of efficient, optimized, and cost-effective SoC solutions

Course Outcomes:
After successful completion of the course, students shall be able to
- Explain hardware–software co-design models, architectures, and co-synthesis methodologies.
- Apply prototyping, emulation, and architecture specialization techniques for embedded systems.
- Analyze compilation techniques and toolchains for modern embedded processor architectures.
- Evaluate design specification, co-design verification, and interface verification methods.
- Apply system-level specification languages and co-simulation techniques for heterogeneous systems.

Module – I: Co-Design Issues **10L**
Co- Design Models, Architectures, Languages, A Generic Co-design Methodology. Co-Synthesis Algorithms: Hardware software synthesis algorithms: hardware – software partitioning distributed system co-synthesis.

Module – II: Prototyping and Emulation **11L**
Prototyping and emulation techniques, prototyping and emulation environments, future developments in emulation and prototyping architecture specialization techniques, system communication infrastructure. 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.

Module – III: Compilation Techniques and Tools for Embedded Processor Architectures **9L**
Modern embedded architectures, embedded software development needs, compilation technologies, practical consideration in a compiler development environment.

Module – IV: Design Specification and Verification **8L**

Design, co-design, the co-design computational model, concurrency coordinating concurrent computations, interfacing components, design verification, implementation verification, verification tools, interface verification.

Module – V: Languages for System **7L**

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.

REFERENCES:

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

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2525514: HARDWARE SECURITY IN VLSI DESIGN
(Professional Elective-IV)

I Year M.Tech. ECE II – Sem. **L T P C**

3 0 0 3

Couse 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 Objectives:

The students will try to learn

- Comprehensive understanding of computing system layers and their security challenges, including the distinction between hardware security and trust, vulnerabilities, attacks, and relevant countermeasures.
- Hardware Trojans, IP piracy, and reverse engineering threats in SoC and FPGA design, while exploring benchmarks and mitigation strategies
- Side-channel and test-oriented attacks, such as power, electromagnetic, fault-injection, timing, scan-based, and JTAG-based attacks, and evaluate their impact on hardware trust.
- Physical attacks and PCB-level threats, including reverse engineering, probing, and bus snooping attacks, and study practical countermeasures for safeguarding electronic hardware.
- Hardware security primitives and trust assessment techniques, including the design and application of PUFs, TRNGs, anti-counterfeit mechanisms, and both pre-silicon and post-silicon security evaluation models

Course Outcomes:

After successful completion of the course, students shall be able to

- Explain fundamental concepts of hardware security, trust, threats, and system-level interactions.
- Analyze hardware trojans, IP piracy, and reverse engineering attacks along with countermeasures.
- Evaluate side-channel and test-oriented attacks and their impact on secure hardware design.
- Analyze physical attacks on chips and PCBs and corresponding protection techniques.
- Apply hardware security primitives and perform security and trust assessment of hardware systems.

Module – I: Introduction to Hardware Security **10L**

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.

Module – II: Hardware Trojans: **9L**

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.

Module – III Side-Channel Attacks: **9L**

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.

Module – IV: Physical Attacks and Countermeasures **8L**

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

Module – V: Hardware Security Primitives, Security and Trust Assessment **7L**

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 BOOKS:

1. Bhunia, Swarup, and Mark M. Tehranipoor. Hardware Security: A Hands-on Learning Approach. 1st ed., Academic Press, 2019.
2. Tehranipoor, Mark, and Cliff Wang, editors. Introduction to Hardware Security and Trust. Springer, 2012.

REFERENCES:

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

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2525515: HARDWARE ACCELERATORS FOR MACHINE LEARNING MODELS (Professional Elective-IV)

I Year M.Tech. ECE II – Sem. **L T P C**

3 0 0 3

Couse 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 Objectives:
The students will try to learn

- Comprehensive understanding of deep neural networks (DNNs) and their hardware requirements, covering popular architectures, frameworks, datasets, and the evolution of hardware platforms for training and inference
- Hardware-aware optimization techniques for DNNs, including weight quantization, compression, sparsity exploitation, zero forwarding, and learning with constrained resources
- Memory–compute interactions and optimization strategies, such as tiling, loop transformations, batching, pruning, and convolution acceleration algorithms (Direct, GEMM, FFT, Winograd)
- Emerging hardware solutions and co-design methodologies, including near-data processing with advanced memories, reduced precision models, benchmarking metrics for DNNs, and evaluation of accelerator performance
- Advanced architectures for deep learning acceleration, including systolic arrays, Tensor Processing Units (TPUs v1–v4), FPGAs, and Brainwave systems, and compare their efficacy for training, inference, and binarized neural networks (BNNs)

Course Outcomes:
After successful completion of the course, students shall be able to

- Explain deep neural network models and the evolution of hardware platforms for deep learning.
- Apply kernel optimization, quantization, and compression techniques for efficient DNN acceleration.
- Analyze memory–compute trade-offs and optimization strategies for CNN acceleration on modern hardware.
- Evaluate near-data processing architectures and benchmarking metrics for DNN models and hardware.
- Compare and apply DNN acceleration techniques on GPUs, TPUs, and FPGAs for training and inference.

Module – I: Overview of DNNs **10L**

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

Module – II: Kernel for accelerating convolution, optimizing networks **9L**

Weight quantization, network compression, sparse operations, zero forwarding, learning with hardware in the loop, learning and inference on low-memory devices.

Module – III Memory and compute **9L**

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.

Module – IV: Near-data processing **8L**

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.

Module – V: **8L**

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.

REFERENCES:

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.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2525516: IMAGE AND VIDEO PROCESSING
(Professional Elective-IV)

I Year M.Tech. ECE II – Sem. **L T P C**

3 0 0 3

Couse 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 Objectives:

The students will try to learn

- Fundamentals of digital image processing, including image formation, sampling, quantization, pixel relationships, and image transforms such as Fourier, Cosine, and Wavelet transforms.
- Image enhancement techniques in both spatial and frequency domains, focusing on filtering, smoothing, sharpening, and selective filtering operations.
- Image segmentation and morphological processing techniques, including edge detection, thresholding, region-based methods, and applications such as image watermarking.
- Image compression and restoration methods, covering coding redundancies, compression models and standards (JPEG, wavelet coding), and restoration techniques using inverse, Wiener, and deconvolution filtering.
- The principles of video processing and motion estimation, including video formation models, motion estimation methods, transform and predictive coding, and standards like MPEG

Course Outcomes:

After successful completion of the course, students shall be able to

- Explain fundamental concepts of digital image processing and image transform techniques.
- Apply spatial and frequency domain methods for image enhancement.
- Analyze image segmentation and morphological processing techniques.
- Evaluate image compression and restoration methods for efficient image representation.
- Apply motion estimation techniques and video processing principles for video coding applications.

Module – I: Fundamentals of Image Processing, Image Transforms **10L**

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

Module – II: Image Enhancement **9L**

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

Module – III: Segmentation, Morphological Image Processing 9L

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

Module – IV: Image Compression 8L

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.

Module – V: Basic Steps of Video Processing, 2-D Motion Estimation 7L

Basic Steps of Video Processing: Analog video, Digital Video, Time varying Image Formation Models: 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

REFERENCES:

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

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2525573: EMBEDDED REAL TIME OPERATING SYSTEM LABORATORY

I Year M.Tech. ES II – Sem.

L	T	P	C
0	0	4	2

Couse 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.

Course Objectives:

The students will try to learn:

- Practical skills in process management by creating and controlling processes using system calls such as fork () and handling inter-process communication (IPC)
- IPC mechanisms including pipes, FIFOs, and message queues for efficient data exchange between processes
- Usage of system calls for file handling and process control, by implementing programs similar to basic UNIX utilities (e.g., cp)
- Multithreading concepts using POSIX threads (pthreads) through thread creation, synchronization, and inter-thread communication
- Synchronization techniques using semaphores and mutexes for safe access to shared resources and signaling mechanisms between concurrent threads

Course Outcomes:

After successful completion of the course, students shall be able to

- Develop system-level programs using command-line arguments and core UNIX system call
- Execute process creation and management along with inter-process communication mechanisms through pipes, FIFOs, and message queues.
- Implement file manipulation utilities through low-level system calls.
- Use POSIX threads in the development of multithreaded applications.
- Apply synchronization mechanisms, semaphores and mutexes, ensuring safe management of shared resources.

List of Experiments:

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)

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2525574: IoT SYSTEM DESIGN LABORATORY

I Year M.Tech. ES II – Sem. **L T P C**

0 0 4 2

Couse 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++/Python) 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 Objectives:

The students will try to learn

- Understand the fundamentals of Arduino and Raspberry Pi for IoT applications
- Interface sensors, actuators, and communication devices with microcontroller/microprocessors
- Design and implement basic automation systems using Arduino and Raspberry Pi
- Analyze sensor data and control real-time operations in embedded IoT applications
- Apply programming concepts to develop smart prototypes for domains such as health, automation, and traffic control

Course Outcomes:

After successful completion of the course, students shall be able to

- Develop programs for microcontrollers and single-board computers with appropriate development environments.
- Implement IoT applications by posting sensor data on cloud platforms and monitoring network traffic.
- Apply programming tools and simulators for debugging and testing embedded systems.
- Interface sensors, actuators, and peripheral modules with Arduino and Raspberry Pi.
- Design and implement wireless and sensor-based embedded applications for real-world use cases.

List of Experiments:

1. Micro-Controller Programming
2. Single-board Computer Programming
3. Posting Data to an IoT Cloud Platform
4. Introduction to Wireshark on Raspberry Pi
5. Programming Arduino with Blockly
6. Programming Raspberry Pi with Python
7. Microcontroller Programming Simulator
8. Advance Sensors, Actuators, Components
9. Debugging the Raspberry Pi
10. Interfacing LED using Arduino & Raspberry Pi.
11. Interfacing DHT11 sensor using Arduino & Raspberry Pi
12. Design of Motion Sensor Alarm using PIR Sensor
13. Wi-Fi-Module Interfacing with Arduino (ESP8266)
14. Bluetooth Module Interfacing with Arduino (HC-05).
15. Driving LED Array ULN2003 Driver.

NOTE: Minimum of 10 experiments to be conducted

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

II-I

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2535517: MACHINE LEARNING FOR ROBOTICS
(Professional Elective-V)

II Year M.Tech. ES I – Sem. **L T P C**

3 0 0 3

Course Overview

This course introduces the integration of machine learning with robotic systems, covering robot hardware, software, and data acquisition. It equips students with knowledge of supervised, unsupervised, and reinforcement learning for robotic perception and control. By the end, students will be able to design intelligent, adaptive robots capable of autonomous decision-making in dynamic environments.

Pre-requisites: Basic understanding of robotics concepts with desirable awareness of machine learning fundamentals and data processing techniques.

Course Objectives:

The students will try to learn

- The fundamentals of robotics, including classification, kinematics, dynamics, planning, control, and robotic vision sensors
- Core computer vision concepts such as image representation, transformations, camera models, calibration, and multi-view geometry
- How to analyze and estimate robot position, orientation, motion trajectories, and visual odometry techniques
- Localization and mapping approaches, including SLAM formulations, relocalization, optimization, and sensor fusion methods.
- To integrate machine learning and deep learning methods for robotic vision, recognition, and interpretation tasks

Course Outcomes:

After successful completion of the course, students shall be able to

- Explain the motivation for edge computing and its architectural evolution from cloud computing.
- Analyze distributed system concepts and application support in edge computing environments.
- Apply containerization and orchestration technologies for deploying services in edge data centers.
- Implement end-to-end edge data pipelines using messaging systems and edge analytics frameworks.
- Evaluate machine learning and deep learning applications for low-latency inference at the edge.

Module-I: **Introduction to Robotics** **7L**

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

Module-II: **Fundamentals of Computer Vision** **7L**

Image acquisition and representation, image transformation, filtering, restoration, morphing, Camera Models, Calibration, Single view geometry, Multiple view geometry, Epipolar geometry, RANSAC

Module-III: **Position and Orientation** **7L**

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.

Module-IV: **Localization and Mapping** **10L**

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, LSDSLAM), Sensor combinations (IMU, mono vs. Stereo, RGB-Depth), Analysis and parameter studies.

Module V: Recognition and Interpretations **5L**

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

References:

1. H. R. Everett, Sensors for Mobile Robots: Theory and Application, A K Peters/CRC Press, 1995, 1st Edition.
2. Dahiya, Ravinder S., Valle, Maurizio, Robotic Tactile Sensing, Springer, 2013.
3. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2nd Edition, 2022.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2535518: EDGE COMPUTING
(Professional Elective-V)

II Year M.Tech. ES I – Sem.

L	T	P	C
3	0	0	3

Couse Overview:
This course introduces the fundamentals of Edge Computing and its role in low-latency, real-time applications. Students will learn key technologies such as Docker, Kubernetes, MQTT, and distributed analytics on edge data centers. It also covers AI, machine learning, and deep learning applications for time-critical edge scenarios like autonomous vehicles and predictive maintenance.

Pre-requisites: Fundamental understanding of computer networks, cloud computing, and distributed systems, programming skills, familiarity with operating systems and containerization, and awareness of machine learning concepts.

Course Objectives:
The students will try to learn

- Fundamentals of Edge Computing and its significance in low-latency, real-time applications
- The use of technologies such as Docker, Kubernetes, MQTT, and distributed analytics for edge deployment
- The edge data center architectures, lightweight clouds, and edge storage systems
- The edge-cloud collaboration frameworks and time-critical data processing pipelines
- The AI, machine learning, and deep learning methods for edge-based applications such as predictive maintenance and autonomous systems

Course Outcomes:
After successful completion of the course, students shall be able to

- Explain fundamental concepts of robotics, kinematics, dynamics, and robot control.
- Apply computer vision techniques for image acquisition, transformation, and geometric analysis.
- Analyze pose estimation, motion estimation, and visual odometry for robotic perception.
- Evaluate techniques for position estimation and environment mapping.
- Apply machine learning and deep learning methods for recognition and interpretation in robotic vision.

Module I: **8L**
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.

Module II: **9L**
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.

Module III: **9L**
Introduction to Edge Data Center, Lightweight Edge Clouds and its services provided by different service providers. Introduction to Docker containers and Kubernetes in edge computing. Design of edge storage systems like key-value stores

Module IV: **8L**

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

Module V: **8L**

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.

TEXTBOOKS:

1. Rajkumar Buyya, Satish Narayana Srirama (Eds.), Fog and Edge Computing: Principles and Paradigms, Wiley, 2019.
2. Rajiv Misra, Yashwant Patel, Cloud and Distributed Computing: Algorithms and Systems, Wiley, 2020

REFERENCES:

1. Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Cloud Computing: Principles and Paradigms, Wiley, 2011.
2. Kai Hwang, Jack Dongarra, Geoffrey Fox, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann, 2012.
3. Flavio Bonomi, Rodolfo Milito, Jiang Zhu, Sateesh Addepalli, Fog Computing and Its Role in the Internet of Things, Springer, 2014.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2535519: EMBEDDED BIOMEDICAL APPLICATIONS
(Professional Elective-V)

II M.Tech. I Semester

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 Objectives:
The students will try to learn
- Fundamentals of bio-potentials, bio-signals, electrodes, transducers, and biomedical systems
- Wearable health devices, biosensors, and their importance in modern healthcare
- Embedded system concepts for medical image processing and hardware design
- Embedded diagnostic systems such as ECG, EEG, MRI, CT, and sonography
- Biomedical equipment through case studies like ventilators, defibrillators, and heart-lung machines

Course Outcomes:
By the end of this course, students will be able to:
- Explain fundamental biomedical engineering concepts, bio-signals, and biomedical instrumentation.
- Examine the design principles and operational mechanisms of wearable health devices and biosensors used in healthcare applications
- Apply embedded system concepts for medical image processing and hardware implementation of algorithms.
- Discuss embedded diagnostic systems used for physiological signal acquisition and medical imaging.
- Evaluate biomedical devices and systems through practical case studies in healthcare applications.

Module- I: Introduction to biomedical engineering **9L**
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.

Module- II: Wearable health devices **9L**
Wearable health devices: Concepts of wearable technology in Healthcare-Components of wearable devices- Biosensors- Blood glucose sensors - Head worn- Hand worn- Body worn pulse oximeter-Cardiac pacemakers – Hearing aids and its recent advancements-wearable artificial kidney.

Module- III: **Embedded system for medical image processing** **10L**

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

Module- IV: Embedded system for diagnostic applications **8L**

Embedded system for diagnostic applications: ICCU patient monitoring system – ECG-EEG-EMG acquisition system-MRI scanner - CT scanner Sonography.

Module- V: Case study **8L**

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

TEXT 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.

REFERENCES BOOKS

1. Khandpur R.S, Handbook of Biomedical Instrumentation, Tata McGraw Hill, New Delhi, 3rd Edition, 2014.
2. L.A Geddes and L.E. Baker, Principles of Applied Biomedical Instrumentation, 3rd Edition John Wiley and Sons, Reprint 2008.
3. Richard S. Cobbold, Transducers for Biomedical Measurements; Principle and applications John Wiley and sons, 1992.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2511345: RESEARCH METHODOLOGY & IPR

I Year M.Tech. ES I – Sem. **L T P C**

2 0 0 2

Couse Overview:
This course equips students with the knowledge and skills to plan, conduct, and present scientific research responsibly. It covers research design, data collection methods, and statistical analysis for accurate interpretation and reporting. Learners gain insight into intellectual property rights, including patents, copyrights, trademarks, and technology transfer, along with practical guidance on patent filing processes. Emphasis on research ethics, plagiarism prevention, and responsible authorship prepares students for high-quality, ethically sound scholarly and industrial research.

Course Objectives:
The students will try to learn
- The fundamentals of research, its types, and the process of formulating a research problem with an effective literature review
- Effective research designs and the application of suitable sampling techniques and data-collection methods with accuracy and reliability
- The statistical analysis, result interpretation, and presentation of findings in a structured research report
- The importance of Intellectual Property Rights and their role in protecting and transferring innovations
- Patent filing process and practice ethical standards in research and publication.

Course Outcomes:
After successful completion of the course, students shall be able to
- Describe the meaning, sources, and key characteristics of a good research problem and common errors in its selection.
- Conduct a comprehensive literature review and demonstrate ethical standards in research for the avoidance of plagiarism.
- Write structured research reports and proposals effectively and present research ideas before review panels.
- Explain forms of intellectual property with national, international patenting and IPR procedures.
- Analyze the scope and rights associated with patents, and explore recent trends in IPR across domains like biotechnology, software, and traditional knowledge.

Module – I: Introduction to Research Methodology **9L**
Introduction to Research Methodology: Meaning and objectives of research, Types of research: basic, applied, qualitative, quantitative, and mixed methods, Research process and design, Formulation of research problem, Literature survey and review techniques.

Module – II: Research Design and Data Collection Methods **9L**
Research Design and Data Collection Methods: Research design types: descriptive, experimental, exploratory, Sampling methods and techniques, Primary and secondary data sources, Data collection tools: surveys, interviews, observations, Validity and reliability of data.

Module – III: Data Analysis and Interpretation **8L**

Data Analysis and Interpretation: Statistical tools and software, Descriptive and inferential statistics, Hypothesis testing and ANOVA, Regression and correlation analysis, Interpretation and presentation of data, Report writing and referencing styles.

Module – IV: Intellectual Property Rights (IPR) **8L**

Intellectual Property Rights (IPR): Overview and importance of IPR, Types of IPR: patents, trademarks, copyrights, industrial designs, Trade secrets and geographical indications, IPR and technology transfer, Role of IPR in research and innovation.

Module – V: Patent Filing and Research Ethics **8L**

Patent Filing and Research Ethics: Patent filing process: national and international (PCT), Patent search and analysis tools, Research ethics and scientific misconduct, Plagiarism and tools for plagiarism detection, Ethical issues in publication and authorship

TEXT BOOKS:

1. Kothari, C. R., and Gaurav Garg. Research Methodology: Methods and Techniques. 4th ed., New Age International, 2019.
2. Nejakar, Santosh M. Research Methodology and Intellectual Property Rights. 1st ed., Techno Science Publications, 2021.

REFERENCES:

1. Leedy, Paul D., and Jeanne Ellis Ormrod. Practical Research: Planning and Design. 12th ed., Pearson, 2019.
2. Creswell, John W., and J. David Creswell. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. 5th ed., Sage Publications, 2018.
3. Day, Robert A., and Barbara Gastel. How to Write and Publish a Scientific Paper. 8th ed., Cambridge University Press, 2016.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2511346: ENGLISH FOR RESEARCH PAPER WRITING
(Audit Course – I)

I Year M.Tech. ES I – Sem. **L T P C**

2 0 0 0

Couse Overview:
This course provides a comprehensive introduction to the interdisciplinary field of disaster and risk management. It explores the fundamental concepts, classification, impacts, and management strategies associated with natural and human-made disasters. Emphasis is placed on understanding vulnerabilities, institutional frameworks, risk assessment tools, and approaches to resilience and recovery.

Course Objectives:
The students will try to learn

- How to improve your writing skills and level of readability
- The process of identifying and applying suitable content in each section of a research paper
- The skills needed for writing a proper title and for maintaining the quality of the paper during its first submission
- An introduction to the research process through engagement with literature writing
- The evaluation of effective university-level essays by applying strategies to improve content, organization, word choice, grammar, and mechanics
- The evaluation of effective university-level essays by applying strategies to improve content, organization, word choice, grammar, and mechanics

Course Outcomes:
After successful completion of the course, students shall be able to

- Describe effective writing skills for planning, sentence structuring, clarity, and conciseness.
- Explain methods of academic writing for presenting results, paraphrasing, avoiding plagiarism, and organizing abstracts and introductions.
- Write a research paper using principles of literature review, methods, results, discussion, conclusions, and final checks.
- Provide constructive feedback on draft titles, abstracts, introductions, and literature reviews for clarity and impact.
- Evaluate draft manuscripts for coherence, logical flow, and readiness for journal submission.

Module – I: **6L**
Planning and Preparation, Word Order, breaking up long sentences, Structuring Paragraphs and Sentences, Being Concise and Removing Redundancy, Avoiding Ambiguity and Vagueness.

Module – II: **7L**
Clarifying Who Did What, Highlighting Your Findings, Hedging and Criticizing, Paraphrasing and Plagiarism, Sections of a Paper, Abstracts. Introduction.

Module – III: **6L**
Review of the Literature, Methods, Results, Discussion, Conclusions, The Final Check.

Module – IV: **7L**
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,

Module – V: **6L**

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:

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's book.
4. Adrian Wallwork, English for Writing Research Papers, Springer New York Dordrecht Heidelberg London, 2011

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2511346: DISATER MANAGRMENT
(Audit Course – I)

I Year M.Tech. ES I – Sem.

L	T	P	C
2	0	0	0

Couse Overview:
This course provides a comprehensive introduction to the interdisciplinary field of disaster and risk management. It explores the fundamental concepts, classification, impacts, and management strategies associated with natural and human-made disasters. Emphasis is placed on understanding vulnerabilities, institutional frameworks, risk assessment tools, and approaches to resilience and recovery.

Course Objectives:
The students will try to learn

- The definition and differentiation of key concepts such as hazard, disaster, risk, vulnerability, resilience, and capacity
- The classification of different types of disasters (natural, technological, biological, hybrid) with examples
- Analysing the causes, characteristics, and dynamics of various hazards
- The identification and assessment of different dimensions of vulnerability (social, economic, environmental, physical)
- The explanation of the disaster risk equation and the application of disaster risk frameworks to real-world scenarios

Course Outcomes:
After successful completion of the course, students shall be able to

- Explain fundamental concepts of hazards, disasters, risk, vulnerability, and resilience.
- Analyze disaster impacts, trends, and regional risk profiles using real-world case studies.
- Apply disaster management cycle principles and institutional frameworks for effective response and recovery.
- Evaluate risk assessment, modelling techniques, and tools such as GIS and simulation methods.
- Design mitigation and resilience strategies incorporating technology and community-based approaches.

Module – I: Fundamentals of Disaster & Risk Concepts **8L**
Definitions: hazard, disaster, risk, vulnerability, resilience, capacity Classification of disasters: natural, technological, biological, hybrid Causes, characteristics & dynamics of hazards Vulnerability analysis: social, economic, environmental, physical factors Disaster risk equation and framework.

Module – II: Disaster Impacts, Trends & Profiles **7L**
Impacts: human, economic, environmental, infrastructural, ecological Trends in global and regional disasters Disaster profile of India / region: seismic zones, flood zones, cyclonic regions, landslides, coastal hazards Case studies of past disasters.

Module – III: Disaster Management Cycle & Institutional Frameworks 8L

Phases: prevention, mitigation, preparedness, response, recovery Structural vs. non-structural measures Early warning systems, communications, decision support Institutional setup: NDMA, SDMA, DDMA, local bodies, NGOs, international agencies Policies, legal frameworks, standard operating procedures (SOPs).

Module – IV: Risk Assessment, Modelling & Tools 8L

Hazard mapping, risk & vulnerability assessment methods GIS, remote sensing, spatial analysis Simulation models, scenario analysis, probabilistic risk modelling Monitoring & early warning technologies Data sources, indicators, metrics.

Module – V: Disaster Mitigation, Resilience & Applications 9L

Mitigation strategies (engineering, ecosystem-based, community based) Resilience building & adaptation Disaster risk reduction (DRR) strategies post-disaster recovery, rehabilitation, reconstruction Role of technology (IoT, mobile, social media, GIS) Community engagement, capacity building, educational & social approaches.

TEXT BOOKS:

1. R. Subramanian, “Disaster management,” Indian textbook aligning with UGC/AICTE syllabus.
2. Rajendra K. Pandey, “Disaster Management in India,” Focuses on policies, institutions, planning in India.

REFERENCES:

1. Damon P. Coppola, “Introduction to International Disaster Management,” Butterworth-Heinemann, 4th Edition (2020).
2. Asian Development Bank (ADB), “Disaster Management: A Disaster Manager's Handbook’’, 2nd Edition, 1991.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2511348: SANSKRIT FOR TECHNICAL KNOWLEDGE
(Audit Course – I)

I Year M.Tech. ES I – Sem. **L T P C**

2 0 0 0

Couse Overview:
This course introduces students to the foundational aspects of Sanskrit with a focus on its relevance to technical and scientific literature. It emphasizes the clarity, structure, and precision of Sanskrit as a language, exploring its potential as a tool for technical communication and ancient scientific wisdom.

Course Objectives:
The students will try to learn

- Basics of the Sanskrit language, including Sandhi, Samasa, Karaka, and sentence construction.
- Logical structure of Sanskrit and its relevance to computational linguistics and artificial intelligence.
- Reading and interpretation of classical Sanskrit texts on subjects such as mathematics, architecture, metallurgy, medicine, and astronomy
- Identification and translation of relevant verses (shlokas) from ancient manuscripts containing practical scientific or engineering knowledge
- Exploration of connections between Sanskrit texts and current scientific principles, encouraging holistic thinking and innovation
- Cultivation of appreciation for traditional Indian knowledge systems and their potential modern applications

Course Outcomes:
After successful completion of the course, students shall be able to

- Understand basic Sanskrit grammar, sentence structure, and verb–noun usage for simple communication.
- Identify and apply technical roots, prefixes, and compound formations used in Sanskrit scientific texts.
- Interpret and translate basic engineering and technical statements between Sanskrit and modern languages.
- Explain mathematical, geometrical, and architectural concepts as expressed in classical Sanskrit sources.
- Read and comprehend technical sutras, kārikās, and their commentaries for applied understanding.

Module – I: Basics & Grammar Refresher **8L**

Sanskrit alphabet (varṇamālā), vowels, consonants, sandhi rules— Declensions (vibhaktis) & noun/adjective agreement Verb roots, tenses (past, present, future), moods, voice (parasmaipada/ātmanepada) Basic sentence structure (subject–verb–object), simple sentences.

Module – II:Technical Vocabulary & Roots **7L**

Introduction to roots (dhātu) used in technical/śāstra texts— Prefixes, affixes used in technical formation Compound words (samāsa) commonly used in scientific/technical contexts.

Module – III: Sanskrit in Engineering / Applied Fields **8L**

Sanskrit equivalents / expressions for basic technical engineering terms (e.g. in Electrical, Mechanical) Reading short Sanskrit technical statements / descriptions Translating simple modern technical statements into Sanskrit.

Module – IV: Mathematics, Architecture, Geometry in Sanskrit 7L

Terms and expressions in mathematical texts (e.g. geometry, algebra) Architectural/engineering technical terms (e.g. measurements, proportions, structural terms) in Sanskrit Excerpts from classical technical treatises.

Module – V: Sutra / Kārikā / Bhāṣya Reading

Structure of technical sutra/karikā texts Reading & comprehending short sutras / verses Use of commentaries / bhāṣya / vyākhyāna to understand meaning — Introduction to Navya-Nyāya style language (if relevant)

TEXT BOOKS:

1. S. Venkita subramonia Iyer, Technical Literature in Sanskrit .
2. Scientific Knowledge in Sanskrit Literature.

REFERENCES:

1. Kshitish Chandra Chatterji, “Technical Terms and Technique of Sanskrit Grammar,” A classic work explaining Sanskrit grammatical technical terms and methods
2. Bijoya Goswami, “A Dictionary of Technical Terms in Kautilya’s Arthaśāstra ’’, This is a more modern collection of scholarship connecting Sanskrit with scientific and technological themes.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2511349: VALUE EDUCATION
(Audit Course – I)

I Year M.Tech. ES I – Sem. **L T P C**

2 0 0 0

Couse Overview:

This course introduces the fundamental principles of human values, ethics, and moral reasoning essential for responsible personal and professional conduct. It emphasizes character building, harmony in relationships, social responsibility, and self-awareness. The course prepares students to apply value-based decision-making in academic, personal, and societal contexts, thereby nurturing them into holistic and responsible professionals.

Course Objectives:

The students will try to learn

- The importance of value education for personal growth and societal well-being
- About human values, moral development, and ethics in professional and social life
- The significance of self-awareness, self-esteem, and emotional intelligence
- Principles of harmony in individual, family, society, and nature
- To apply values in real-life situations, including conflict resolution, sustainable living, and professional ethics

Course Outcomes:

After successful completion of the course, students shall be able to

- Explain the concept, purpose, and significance of value education in personal and professional life.
- Demonstrate self-awareness, emotional intelligence, and ethical values in interpersonal relationships.
- Apply social, family, and professional responsibilities with sensitivity to equality and justice.
- Analyze ethical issues and decision-making processes in professional and organizational contexts.
- Integrate human values with sustainability, technology, and environmental responsibility.

Module – I: Introduction to Value Education

- Definition, purpose, and importance of value education.
- Human values – classification, role of values in personality development.
- Need for moral values in technical and professional education.

Module – II: Self and Relationships

- Self-awareness and self-esteem.
- Emotional intelligence and empathy.
- Values in relationships – trust, respect, responsibility, and love.

Module – III: Harmony in Family and Society:

- Family values, duties, and responsibilities.
- Social ethics and responsibilities of engineers.
- Importance of gender equality and social justice.

Module – IV: Professional Ethics and Human Excellence

- Professional and organizational ethics.

- Ethical dilemmas and decision-making.
- Role of values in achieving excellence at workplace.

Module – V: Values for Sustainable Development

- Harmony with nature, sustainable lifestyles.
- Science, technology, and value integration.
- Case studies on environmental ethics, social responsibility, and Gandhian principles.

TEXT BOOKS:

1. R. R. Gaur, R. Sangal, G. P. Bagaria, A Foundation Course in Human Values and Professional Ethics, Excel Books, 2010.
2. S. B. Gogate, Value Education: Philosophical Foundations, Principles and Practice, Himalaya Publishing House, 2012.

REFERENCES:

1. M. Govindraj Reddy, Professional Ethics and Human Values, Maruthi Publications, 2016.
2. A. N. Tripathy, Human Values, New Age International Publishers, 2003.
3. Kalam, A. P. J. Abdul, Ignited Minds: Unleashing the Power Within India, Penguin Books, 2002.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2521350: CONSTITUTION OF INDIA
(Audit Course – II)

I Year M.Tech. ES II – Sem. **L T P C**

2 0 0 0

Couse Overview:

This course provides a comprehensive understanding of the Constitution of India, its philosophy, structure, and functioning. It introduces students to the fundamental rights, duties, directive principles, and the values enshrined in the Constitution. The course emphasizes the role of citizens in a democratic system, functioning of government institutions, and constitutional provisions for social justice and equality. It aims to develop responsible citizenship, awareness of constitutional morality, and respect for democratic values.

Course Objectives:

The students will try to learn

- The historical background and making of the Indian Constitution
- Fundamental rights, directive principles, and fundamental duties of Indian citizens
- The structure, powers, and functions of the Union and State governments
- The role of the judiciary in safeguarding constitutional provisions
- The importance of constitutional values in promoting justice, liberty, equality, and fraternity

Course Outcomes:

After successful completion of the course, students shall be able to

- Understand the historical background, philosophy, and core values of the Indian Constitution.
- Explain Fundamental Rights, Directive Principles, and Fundamental Duties and their role in governance.
- Describe the structure, powers, and functioning of the Union Government and Parliament.
- Analyze the roles of State Government institutions and the judiciary in upholding constitutional governance.
- Appreciate local self-governance, democratic processes, and constitutional values, namely justice, equality, and secularism.

Module – I: Introduction to the Constitution:

- Historical background and framing of the Constitution.
- Salient features and philosophy of the Constitution.
- Preamble: Meaning, significance, and values.

Module – II: Fundamental Rights and Duties:

- Fundamental Rights: scope, restrictions, and significance.
- Directive Principles of State Policy: role and importance.
- Fundamental Duties of citizens.

Module – III: Union Government:

- President: powers and functions.
- Prime Minister and Council of Ministers.
- Parliament: structure, powers, and legislative process.

Module – IV: State Government and Judiciary

- Governor and Chief Minister.

- State Legislature: composition and powers.
- Judiciary: Supreme Court and High Courts – powers, functions, and judicial review.

Module – V: Local Governance and Constitutional Values

- Panchayati Raj and Municipalities.
- Election Commission and democratic processes.
- Constitutional provisions for social justice, equality, and secularism.
- Role of youth in upholding constitutional values.

TEXT BOOKS:

1. D. D. Basu, Introduction to the Constitution of India, Lexis Nexis, 24th ed., 2019.
2. J. C. Johari, Indian Government and Politics, Sterling Publishers, 6th ed., 2017.

REFERENCES:

1. P. M. Bakshi, The Constitution of India, Universal Law Publishing, 16th ed., 2020.
2. Granville Austin, The Indian Constitution: Cornerstone of a Nation, Oxford University Press, 2008.
3. Subhash Kashyap, Our Constitution: An Introduction to India's Constitution and Constitutional Law, National Book Trust, 2011.

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AUTONOMOUS)

2521351: PEDAGOGY STUDIES
(Audit Course – II)

I Year M.Tech. ES II – Sem. **L T P C**

2 0 0 0

Couse Overview:

The Pedagogy Studies course is designed to prepare postgraduate engineering students to effectively teach, design, and evaluate curriculum in higher education, particularly in the fields of engineering and technology. It equips learners with theoretical foundations, teaching strategies, assessment techniques, and research skills necessary for fostering effective learning environments. The course bridges the gap between technical expertise and teaching competency, ensuring that M. Tech graduates not only excel as engineers but also as educators and researchers in academia or training organizations.

Course Objectives:

The students will try to learn

- The principles and practices of pedagogy in engineering and technology education
- Designing effective teaching-learning strategies using modern tools and digital resources
- Educational psychology and student learning behavior for better curriculum planning
- Communication and presentation skills for effective knowledge dissemination
- Sustainability and ethics into teaching practices
- Curriculum design, assessment methods, and research-based pedagogy

Course Outcomes:

After successful completion of the course, students shall be able to

- Explain key concepts related to curriculum, teacher education, and learning theories.
- Outline various pedagogical practices used by teachers in formal and informal classrooms in developing countries.
- Analyze the strengths of evidence supporting effective pedagogical strategies and teacher attitudes impacting classroom learning.
- Assess the alignment of professional development programs with classroom practices, including barriers such as resources and class size.
- Apply knowledge of research design and pedagogy to identify future directions and gaps in teacher education and curriculum research.

Module – I: **7L**

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.

Module – II: **6L**

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

Module – III: **8L**

Evidence on the effectiveness of pedagogical practices, Methodology for the stage, quality assessment of included studies. How can teacher education (curriculum and practicum) and the school 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.

Module – IV: **7L**

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

Module – V: **6L**

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

TEXT BOOKS:

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.

REFERENCES:

1. 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.
2. Alexander RJ (2001) Culture and pedagogy: International comparisons in primary education. Oxford and Boston: Blackwell.
3. Chavan M (2003) Read India: A mass scale, rapid, ‘learning to read’ campaign.
f