NIRMA UNIVERSITY

INSTITUTE OF TECHNOLOGY

M. Tech. Electronics & Communication Engineering

M.Tech (EC-Embedded Systems)

Syllabus

SEMESTER- I

Course Code	**3EC3108**
Course Title	**Advanced Digital System Design**

Course Learning Outcomes (CLO):

At the end of the course, students will be able to -

1. comprehend digital circuits, systems, design using LSI - MSI Logics and architecture of the different reconfigurable devices
2. ddesign of synchronous system, asynchronous systems, data path and control path using Finite State Machine and analyse the timing issues in the systems
3. implement systems using algorithmic state machine chart and Microprogramming through state machine charts
4. iimplement the modern digital systems through EDA tools for Simulation & Synthesis of hardware description language

Unit No.	Syllabus	Teaching hours:45
I.	Introduction to Digital Systems Design: overview of digital fundamentals, design with LSI and MSI Logics, Advanced Boolean algebra and advanced K Map for optimization, Hazards in combinational networks	07
II.	Synchronous Sequential Logic: Analysis and design procedures of Sequential circuits, Concept of State reduction and state assignment, Shift registers, Counters, HDL for Sequential Circuits, Timing and Triggering Consideration, Clock Skew	10
III.	Hardware Description Languages: Fundamentals of HDL, Overview of Digital Design with HDL, Hierarchical Modeling Concepts: Top-down and bottom-up design methodology, Modules and Port, Gate-Level Modeling, Dataflow Modeling, Behavioral Modeling, Tasks and Functions, RTL Description of Simple Machine and Design from RTL description, Test Bench, Synthesis Issues	10
IV.	Asynchronous Sequential Logic: Analysis and design of asynchronous sequential circuits, primitive flow table, reduction of state and flow tables, concept of race, race free state assignment, critical race and hazards, design issues like metastability, synchronizers, clock skew and timing considerations	05
V.	SM Charts and Microprogramming: State machine charts, derivation of SM charts, realization of SM charts, Linked microprogramming, Hierarchical & Concurrent FSM (HCFSM).	05
VI.	Programmable Logic Devices: Introduction to PLDs, PROM based design, Design and Implementation using PLD	03
VII.	FPGA & CPLD Architectures: RAM based FPGAs, Antifuse FPGAs, Selecting PGAs based on CLBs, Input/output Blocks, Programmable Interconnect, study of Xilinx ,Altera FPGAs	05

Suggested Readings:

1. Charles H Roth Jr. and John Lizy Kurian, “Digital Systems Design using VHDL”, Cengage Learning India.
2. Samir Palnitkar, Verilog® HDL: A Guide to Digital Design and Synthesis, Second Edition, Prentice Hall PTR.
3. Charles H.Roth, Jr. “Fundamentals of Logic Design”, Jaico Publishing
4. Michael D. Ciletti, Advanced Digital Design with the Verilog HDL, PHI
5. Joseph Cavanagh, Digital Design and Verilog HDL Fundamentals, CRC Press
6. Chan and Mourad, Digital Design using Field Programmable Gate Arrays, Prentice Hall of India
7. William I. Fletcher, An Engineering Approach to Digital Design, Prentice Hall of India
8. J. Bhaskar, VHDL Synthesis: A Practical Primer, Star Galaxy Publishing.
9. Pong P. Chu, FPGA Prototyping by Verilog Examples, Wiley, 2008.

L= Lecture, T= Tutorial, P= Practical, C= Credit

3EC3102 Processor Architecture & Design [4 0 0 4]

Course Learning Outcome:

After successful completion of the course, student will be able to

1. Understand basic architecture of Reduced Instruction Set Computing.
2. Apply basic concept of Parallelism and Memory Hierarchy to improve processor performance
3. Analyze embedded processor peripherals and bus standards
4. Evaluate advanced processor platforms

Syllabus:

Processor Architecture Fundamentals: Measuring and reporting performance, Instruction set principles, Instruction level parallelism-Pipeline concepts, Data Path, Memory Hierarchy Design, VLIW, SIMD architecture and performance, Thread level Parallelism

ARM Microcontroller Architecture: ARM Microcontroller architecture : Block Diagram, Features, Memory Mapping Memory Controller (MC), Memory Controller Block Diagram, Address Decoder, External Memory Areas, Internal Memory Mapping, External Bus Interface (EBI), Organization of the External Bus Interface, EBI Connections to Memory Devices, External Memory Interface, Write Access, Read Access, Wait State Management, , Memory Management Units, details of the ARM MMU, ARM Instruction Set, Thumb Instruction Set and Interrupt Handling.Embedded Processor Peripherals: Interrupt Controller - Normal Interrupt, Fast Interrupt, AIC, System Timer (ST)-Period Interval Timer (PIT), Watchdog Timer (WDT), Real-time Timer (RTT), Real Time Clock (RTC) and Parallel Input/output Controller (PIO) Timers/Conters,SPI,PWM,WDT,I2C,CAN, Emerging Bus Standards (USB,PCI), PCIe, CPCI, VPX.

Case Study: Embedded Processor Architecture, ARM Processors - OMAP, TI MSP-430 RISC/ARM Design Philosophy, ARM CORTEX Architecture and Programming.

Self Study:

The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

References:

1. John L. Hennessy, David A. Patterson, Computer Architecture: A Quantitative Approach, Stanford University, Elsevier
2. Steve Furber, ARM System- On- Chip Architecture, Pearson Education Asia
3. Daniel Tabak, Advanced Microprocessor, Tata McGraw Hill
4. Andrew N Sloss, Dominic Symes, Chris Wright, ARM System Developer's Guide - Designing and Optimizing System Software, Elsevier

3EC3103 Digital Signal Processing and Applications [3 0 0 3]

Course Learning Outcome:

After successful completion of the course, student will be able to

1. Understand the concepts of FS, DFS, FT, DTFT, DFT and Z transforms
2. Understand the basics of FIR, IIR and Adaptive filters
3. Design and Implement various kinds of FIR, IIR and Adaptive filters
4. Design System based on Multirate signal processing
5. Understand Wavelet Transforms and Filter bank implementation
6. Apply DSP processor architectures for Codecs in Embedded Systems

Syllabus:

Fundamentals of Digital Signal Processing ,DFT, IDFT, FFT, Convolution, FIR and IIR Filter Design, Algorithm implementation using DSP, Digital Signal Processor Architecture, DSP based software development tools, Introduction to Multirate Singal Processing, Adaptive Filters, CIC filters, DSP applications, Introduction to Codecs.

Self Study:

The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

References:

1. Li Tan, DSP Fundamentals & Applications, Elsevier
2. Sanjit Mitra, DSP A Computer Based Approach , Tata McGraw Hill
3. Sen M Kuo, Woon Seng Gan, Digital Signal Processors, Architectural Implementations and Applications, Pearson Education
4. Fredric J. Harris, Multirate Digital Signal Processing for Communication Systems, Prentice Hall India

3EC3104 Embedded Programming [3 0 0 3]

Course Learning Outcome:
After successful completion of the course, student will be able to
1. Understand the Software Architectures for Embedded Systems
2. Program Embedded Systems using programming languages
3. Understand issues related to Programming, Run-time Environment and Memory management in Embedded Systems
4. Analyze the embedded systems' specification and develop software programs
5. Evaluate the requirements of programming Embedded Systems, related software architectures and tool chain for Embedded Systems

Syllabus:
Software Architectures: Requirements of Embedded Software, Interrupts- Basics, shared data problem, latency, scheduling policy: Round Robin, Round robin with interrupt, function queue scheduling.
Programming Languages: Desired Language Characteristics, Introduction to Object Oriented Programming concepts, C for Programming embedded systems, Object Oriented Programming for Embedded Systems in C++ , Use of Java for Embedded Systems , Run-time Exception handling
Programming and Run-time Environment: Mixing C and assembly, Compiling, Assembling, Linking, Debugging, Basic Compilation Techniques, Profiling: Analysis and Optimization of Execution Time, Analysis and Optimization of Energy and Power, Analysis and Optimization of Program Size, Program Validation and Testing
I/O Programming & Memory management: Synchronization, transfer rate, latency, Polled waiting loops, interrupt driven I/O, DMA, Storage class, scope and lifetime of variables, static and dynamic allocation, shared memory, recursive/reentrant functions
Introduction to Boot Loader, Embedded File Systems, Embedded Database and Data Structure

Self Study:
The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

References:
1. David Simon, Embedded Software Premier, Pearson Education
2. Wayne Wolf, Computers as Components, Morgan Kaufmann
3. Sri Ram Iyer ,Pankaj Gupta, Embedded Real-time Systems Programming, Tata McGraw Hill
4. Daniel W lewis, Fundamentals of Embedded Software: where C and Assembly meet, Prentice Hall
5. Shibu K V, Introduction to Embedded Systems, Tata McGraw Hill

3EC3202 Electronic System Design [3 0 0 3]

Course Learning Outcome:

After successful completion of the course, student will be able to

1. Understand the design issues in analog and mixed signal circuit design
2. Solve design & speed issues in ADC- DAC, power supply and learn the techniques to overcome them
3. Understand the benefits and issues on migration of 5-volt and 3.3 volt logic to lower voltage supplies
4. Analyze design for testability and estimating digital system reliability
5. Understand and identify design issues in digital circuit layout and PCB and learn techniques to mitigate them
6. Understand the concept of electromagnetic interference, electrostatic discharge and techniques to reduce them in electronic systems
7. Understand influence of packaging and enclosures for electronic system

Syllabus:

Practical Analog and Mixed Signal Circuit Design Issues and Techniques:

Passive components: Understanding and interpreting data sheets and specifications of various passive and active components, non-ideal behaviour of passive components.

Op amps: DC performance of op amps: Bias, offset and drift, AC Performance of operational amplifiers: band width, slew rate and noise, Properties of a high quality instrumentation amplifier, Design issues affecting dc accuracy & error budget analysis in instrumentation amplifier applications, Isolation amplifier basics and Active filers: design of low pass, high pass and band pass filters.

ADCs and DACs: Characteristics and performance parameters of ADC & DAC, interfacing to microcontrollers, selecting proper ADC and DAC.

Power supplies: Characteristics, design of full wave bridge regulated power supply. Circuit layout and grounding in mixed signal system.

Practical Logic Circuit Design Issues and Techniques: Understanding and interpreting data sheets, specifications of various CMOS & BiCMOS family Logic devices, Electrical behaviour (steady state & dynamic) of CMOS& BiCMOS family logic devices, Benefits and issues on migration of 5-volt and 3.3 volt logic to lower voltage supplies. CMOS/TTL Interfacing Basic design considerations for live insertion. JTAG/IEEE 1149.1 design considerations, Design for testability, estimating digital system reliability, Digital circuit layout and grounding, PCB design guidelines for reduced EMI.

Electromagnetic Compatibility: Designing for EMC, EMC regulations, typical noise path, methods of noise coupling and methods of reducing interference in electronic systems.

Cabling of Electronic Systems: Capacitive coupling, effect of shield on capacitive coupling, inductive coupling, effect of shield on inductive coupling, effect of shield on magnetic coupling, magnetic coupling between shield and inner conductor, shielding to prevent magnetic radiation, shielding a receptor against magnetic fields, coaxial cable versus shielded twisted pair, ribbon cables.

Grounding of Electronic Systems: Safety grounds, signal grounds, single-point ground systems, multipoint-point ground systems, hybrid grounds, functional ground layout, practical low frequency grounding, hardware grounds, grounding of cable shields ground loops, shield grounding at high frequencies Power line filtering, power supply decoupling, decoupling filters, high frequency filtering and system bandwidth.

Protection Against Electrostatic Discharges (ESD): Static generation, human body model, static discharge, ESD protection in equipment design, software and ESD protection, ESD versus EMC.

Packaging & Enclosures of Electronic System: Effect of environmental factors on electronic system, nature of environment and safety measure, Packaging's influence and its factors.

Self Study:

The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

References:

1. Kim R. Fowler, Electronic Instrument Design, Oxford University Press.
2. Henry Wott, Noise Reduction Techniques in Electronic Systems, John Wiley & Sons.
3. John F. Wakerly, Digital Design Principles& Practices, Prentice Hall

4. Robert F. Coughlin, Operational Amplifiers and Linear Integrated Circuits, Prentice Hall
5. Mark.T Thompson, Intuitive Analog circuit design, Elsevier

3EC3106 Design Technologies Lab –I [0 0 2 2]

Course Learning Outcome:

After successful completion of the course, student will be able to

1. Design the Digital Systems using reconfigurable devices with the help of Hardware Description Languages and IP cores
2. Perform the Synthesis, Simulation and Implementation of System on reconfigurable device
3. Implement Digital Systems using EDA tools
4. Implement Embedded Systems using soft core processors on reconfigurable devices

Syllabus:

Lab Experiments based on Digital System Design using VHDL/Verilog, Synthesis, Simulation and FPGA Implementation

Case Studies: Embedded System Design using FPGA, PSoC. DSP Design using FPGA, PSoC

Design of a real-time data acquisition & processing system using the FPGA based soft processor platforms

Self Study:

The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

Laboratory Work:

Laboratory work will be based on above syllabus with minimum 10 experiments to be incorporated.

3EC3107 Embedded Systems Lab [0 0 2 2]

Course Learning Outcome:

1. After successful completion of the course, student will be able to
2. Understand hardware design requirements of embedded systems
3. Understand software design requirements of embedded systems
4. Select appropriate software tool chain, hardware and technique for embedded system design
5. Design a system with the desired design metrics under realistic constraints
6. Experiment and test the embedded system hardware and software design

Syllabus:

Lab Experiments based Advanced Controllers: ARM Processors, Cortex Core, TI MSP430, PSoC, ATMEL, AVR, PIC. Linux ,Embedded OS, Embedded Programming of Controllers using Assemblers, Compilers, Linkers, Loaders, Debuggers ,Embedded In-Circuit Emulators and JTAG, Profilers and Test Coverage Tools ,Build Tools for Embedded Systems ,Configuring and Building GNU Cross-Toolchain. Design of a real-time data acquisition & control system using the controller based Platforms

Self Study:

The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

Laboratory Work:

Laboratory work will be based on above syllabus with minimum 10 experiments to be incorporated.

L	T	P	C
-	-	2	1

Course Code	**3EC3114**
Course Title	**Embedded Networking Laboratory**

Course Outcomes (COs):

At the end of the course, students will be able to -

1. Interface microcontroller/microprocessor using popular buses
2. Experiment with network of devices using popular network standards.

Lab Experiments based on serial/parallel port communication, IP based network programing, bus protocols like SPI, I2C, I2S. Wireless network programming using WiFi, Bluetooth and Zigbee standards, Use of Python and Scripting Language for programming.

L = Lecture, T = Tutorial, P = Practical, C = Credit

SEMESTER- II

3EC3201 High Performance Computing [3 0 0 3]

Course Learning Outcome:
After successful completion of the course, student will be able to

1. Apply the basic knowledge of state of art general purpose processor architecture and understand the issues related with the High Performance Computing
2. Apply fundamentals of programming and write programs for various High performance Computing
3. Program using various computing platform the technology associated with it
4. Understand the different techniques of programming for high performance computing for various domain of engineering
5. Devise new approaches of high performance computing for engineering applications

Syllabus:
Parallel Processing Concepts (Quick Overview) : Levels of parallelism (instruction, transaction, task, thread, memory, function), Models (SIMD, MIMD, SIMT, SPMD, Dataflow Models, Demand-driven Computation etc), Architectures: N-wide superscalar architectures, multi-core, multi-threaded
Parallel Programming with CUDA: Processor Architecture, Interconnect, Communication, Memory Organization, and Programming Models in high performance computing architectures: (Examples: IBM CELL BE, Nvidia Tesla GPU, Intel Larrabee Microarchitecture and Intel Nehalem microarchitecture), Memory hierarchy and transaction specific memory design , Thread Organization
Fundamental Design Issues in Parallel Computing: Synchronization, Scheduling, Job Allocation, Job Partitioning, Dependency Analysis, Mapping Parallel Algorithms onto Parallel Architectures, and Performance Analysis of Parallel Algorithms
Fundamental Limitations Facing Parallel Computing: Bandwidth Limitations, Latency Limitations, Latency Hiding/Tolerating Techniques and their limitations
Power-Aware Computing and Communication: Power-aware Processing Techniques, Power-aware Memory Design, Power-aware Interconnect Design, Software Power Management
Advanced Topics:, Petascale Computing, Optics in Parallel Computing, Quantum Computers, Recent developments in Nanotechnology and its impact on HPC

Self Study:
The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

References:

1. George S. Almasi , Alan Gottlieb, Highly Parallel Computing, Benjamin Cummings
2. Kai Hwang, Naresh Jotwani, Advanced Computer Architecture: Parallelism, Scalability, Programmability, Tata McGraw Hill
3. David Culler, Jaswinder Pal Singh, Parallel Computer Architecture: A hardware/Software Approach, Morgan Kaufmann
4. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, Addison-Welsey.
5. David A. Bader (Ed.), Chapman & Hall, Petascale Computing: Algorithms and Applications, CRC Computational Science Series

3EC3203 Embedded Operating System [3 0 0 3]

Course Learning Outcome:
After successful completion of the course, student will be able to
1. Understand the fundamental concepts of Real Time Systems
2. Acquire specification, verification and practical knowledge of Real Time Operating Systems
3. Understand the use of multitasking techniques in Real Time Systems
4. Understand the features and structures of practical Real Time System implementations
5. Analyze the system's safety criticality requirement and their impact on real-time system design
6. Select tools and techniques for real time system design and implement with embedded operating systems

Syllabus:
Introduction: Overview of OS: Multithread Systems, Processes and Threads, Context Switching, Multi tasking, Cooperative Multi-tasking, Pre-emptive Operating Systems structure, Operating system function, Timing requirements on processes, Features of an Operating System
Real Time Task Scheduling: Process state and scheduling, Clock driven and Event driven scheduling, Rate-Monotonic Scheduling, Earliest-Deadline First Scheduling, Fault-Tolerant Scheduling
Inter-process Communication: Signals, Shared Memory Communication, Message-Based Communication
Real-time Memory Management: Process Stack Management, Dynamic Allocation
I/O Operations: Synchronous and Asynchronous I/O , Interrupt Handling, Device Drivers
Handling resource sharing and dependencies among real time tasks: Resource sharing Protocols: Priority Inheritance Protocol, Highest locker protocol, priority ceiling protocol, Priority Inversion, Issues in resource sharing protocols
Example of Real-time OS: Features of RTOS, POSIX, Case studies: FreeRTOS. µCOS, RTx51 TinyOS, Building RTOS/EoS image for Target Hardware, Benchmarking RTOS, VxWorks, CMX
Example of Embedded OS: Embedded Linux, Android, WinCE, Symbian
Evaluating and Optimizing Operating System Performance: Effects of scheduling, Response-time Calculation, Interrupt latency, Time-loading, Memory Loading, Power Optimization Strategies for Processes, Advanced Configuration and Power Interface

References:

1. Wayne Wolf, Computers as Components Principles of Embedded Computing System Design, Morgan Kaufman
2. Rajib Mall, Real Times Systems Theory and Practice, Pearson Education
3. Peter Marvedel, Embedded System Design, Springer
4. Krisha & Shin, Real-Time Systems, McGraw Hill

ELECTIVE- I

3EC3125 Multimedia Systems & Applications [3 0 0 3]

Course Learning Outcome:

After successful completion of the course, student will be able to

1. Understand different lossy and lossless compression algorithms
2. Evaluate various lossy and lossless compression algorithms for variety of source data
3. Understand the various transforms required to compress images and critically analyze their advantages and limitations
4. Analyze different audio, image and video compression standards and their advance features
5. Understand different protocols of multimedia communication networking and their applications
6. Analyze various case studies on Audio, Image or Video compression algorithms and evaluate their ability for various design aspects of compression algorithm

Syllabus:

Multimedia information system : Digital multimedia system, hardware, algorithm and standards, multimedia compression techniques and standards, multimedia communication and synchronization, multimedia storage and retrieval, media I/O server and synchronization mechanism, multimedia data transfer service using internet express, conferencing, multimedia standards, tools and applications.

Self Study:

The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

References:

1. Li & Drew, Fundamentals of Multimedia, Prentice Hall India
2. Kalid Sayood, Data Compression, Morgan Kauffman
3. Saloman, Data Compression Handbook, Springer
4. Halsall, Multimedia Communications & Networking, Person Education Asia
5. Steiner, Multimedia Computing, Person Education Asia

L	T	P	C
3	-	-	3

3EC3135 Design of Integrated Circuits [3 0 0 3]

Course Learning Outcome:
After successful completion of the course, student will be able to
1. Understand the fundamentals of MOS Transistor, static CMOS, Dynamic CMOS based design and Advanced MOS Modeling
2. Analyze the problems associated with CMOS layout and the fabrication processing
3. Design custom IC using layout, design rules and Stick diagrams methods
4. Understand the various analog and mixed signal circuit component and its characteristics
5. Understand basic blocks of the real word interfacing with digital system like ADC and DAC

Syllabus:
Introduction: MOS Transistor, Advanced MOS Modeling, IC system design options, CMOS processing, layout and design rules, Stick diagrams, Digital CMOS layout design, Analog CMOS Layout Considerations
CMOS Digital Circuit Design: Inverter transfer characteristics, noise margins, Transient response and transistor sizing, Circuit Power Consumption, Capacitance estimation, buffer design, Speed-power-area trade-off, Static complementary gates, Transmission gates and tristate circuits, Storage elements, Pass transistor logic, Dynamic logic
CMOS Analog Circuit Design: Current Mirrors and Single Stage Amplifiers, Op-AMP Design and Compensation, Comparators,
CMOS Mixed Circuit Design: A/D Converter, D/A Converter, PLL, Memory

Self Study:
The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

References:

1. S. M. Kang ,Y. Leblebici, CMOS Digital Integrated Circuits: Analysis and Design, McGraw Hill.
2. David A Johns, Ken Martin, Analog Integrated Circuit Design, John Wiley & Sons Inc.
3. B. Razavi, Design of Analog CMOS Integrated Circuits, McGraw- Hill.
4. Philip E. Allen, Douglas R.Holberg, CMOS Analog Circuit Design, Oxford University Press.

Course Code	**3EC3165**
Course Title	**Virtual Reality Engineering**

Course Learning Outcomes (CLO):
At the end of the course, students will be able to -

1. comprehand the Basics of Virtual Reality, and its technology
2. analyze available Virtual reality implementation fundamentals/systems
3. design Virtual Reality Application
4. apply Virtual Reality tools/technology/standards in Real World Design Constraints

Unit No.	**Syllabus**	**Teaching hours: 45**
I.	Introduction : Goals, Definitions, Software, Hardware, Sensation, perception	03
II.	Geometry of Virtual Worlds and Study of Perception and Sensation Perceptual Engineering, Importance in Virtual Reality	06
III.	Light and Optics, Human Optical System Human Visual Physiology	06
IV.	Human Visual Physiology Visual Perception, Depth and Motion Perception	05
V.	Tracking Systems, Pose tracking, technologies for pose tracking	03
VI.	Visual Rendering, Rastering, Shading, CUDA programming	06
VII.	Auditory Sensation and Perception, Rendering Audio, 3D audio	05
VIII.	Haptic Sensation and Perception, Rendering Haptics, Stereognosis, Sensation and Perception of Other Senses, Rendering other senses	05
IX.	Interfaces for Virtual Reality, Locomotion, Manipulation, Social Interaction, Applications and Challenges in VR, Evaluation of VR systems	06

Suggested Readings:

1. Steve Lavalle, Virtual Reality by, online open book
2. George Mather, Foundations of Sensation and Perception: Psychology Press
3. Kelly S. Hale, Kay M. Stanney, Handbook of Virtual Environments: Design, Implementation, and Applications, CRC Press.
4. Peter Shirley, Michael Ashikhmin, and Steve Marschner, Fundamentals of Computer Graphics, A K Peters/CRC Press
5. Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev, 3D User Interfaces, Addison-Wesley.
6. K. S. Hale and K. M. Stanney, Handbook on Virtual Environments, CRC Press.
7. Bureda Gridore, Coiffet Phillipe, Virtual Reality Technology Wiley Interscience

L= Lecture, T= Tutorial, P= Practical, C= Credit

ELECTIVE – II & III

3EC3214 Sensor Networks [3 0 0 3]

Course Learning Outcome:

After successful completion of the course, student will be able to

1. Understand wireless sensor networks concepts, principles and applications
2. Understand communication protocols and standards utilized in Wireless Sensor Networks
3. Analyze protocols used in various types of Wireless Sensor Networks
4. Understand appropriate techniques, standards and tools for Wireless Sensor Network design, sensor node embedded system design and implementation
5. Understand tools and techniques for Sensor Network management
6. Design, implement and maintain wireless sensor networks

Syllabus:

Introduction: Challenges and Hurdles in Sensor network design, Radio-frequency identification (RFID)
Applications of Sensor Networks :Disaster relief applications, Environment control and biodiversity mapping, Intelligent buildings Facility management, Machine surveillance and preventive maintenance, Precision agriculture Medicine and health care, Telematics, Logistics
Single-node architecture: Hardware components, Energy consumption of sensor nodes, Operating systems and execution environments, Physical layer and transceiver design considerations in WSNs
Network architecture: Sensor network scenarios - single hop and multi hop, network, multiple sink/sources, Optimization goals and figures of merit - QoS, energy efficiency, scalability, robustness, Design principles for WSNs, Service interfaces of WSNs, Gateway concepts
Time synchronization, Localization and positioning: Time synchronization problem, Protocols based on sender/receiver synchronization, Protocols based on receiver/receiver synchronization, Properties of localization and positioning procedures, Localization approaches- Proximity, Trilateration and triangulation, Single-hop and Multi hop localization
Topology control: Aspects of topology-control algorithms, Controlling topology in flat networks and Hierarchical networks, Hierarchical networks by clustering, Hierarchical topologies and power control
Link-layer protocols: Error control, Framing, Link management
Medium Access Control : Fundamentals of MAC Protocols, Types of MAC protocols - Schedule-Based and Random Access- Based Protocols, Case Study- Sensor-MAC
Routing : Problems in routing, Gossiping and agent-based unicast forwarding, Energy-efficient unicast, broadcast and multicast techniques, Geographic routing, Mobility support
Transport layer and quality of service: Coverage and deployment, Reliable data transport, Single packet delivery, Block delivery, Congestion control and rate control
Operating system for Sensor Nodes: Embedded operating systems, Programming paradigms and application programming interfaces, Structure of operating system and protocol stack, Case Study: TinyOS and nesC

Self Study:

The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

References:

1. Holger Karl, Andreas Willig, Protocols and Architectures for Wireless Sensor Networks, John Wiley & Sons
2. Kazem Sohraby, Daniel Minoli, Taieb Znati, Wireless Sensor Networks, Technology, Protocols, and Applications, John Wiley & Sons
3. Ananthram Swami, Qing Zhao, Yao-Win Hong, Lang Tong, Wireless Sensor Networks, Signal Processing and Communications Perspectives, John Wiley Publications
4. C. S. Raghavendra, Krishna M. Sivalingam, Taieb Znati, Wireless Sensor Networks, Kluwer Academic Publishers
5. Bhaskar Krishnamachari, Networking Wireless Sensors, Cambridge University Press

3EC3254 Software Engineering [3 0 0 3]

Course Learning Outcome:

After successful completion of the course, student will be able to

1. Understand the software models
2. Understand the software engineering process in terms of requirements, design, and implementation
3. Understand software project management, risk management and change management
4. Apply a software engineering process to a large, term-long embedded software project
5. Produce a software design based on requirements, a software prototype to explore a particular design, and conduct Verification, Validation and documentation
6. Work effectively in a team and deliver software product (demo) and experiences (presentations)

Syllabus:

Introduction: Software products, the software process, Software models: Waterfall Model, Incremental Model, Evolutionary Model, Boehm's spiral model, Process visibility, professional responsibility, computer based system engineering.

Requirements & Specification: Requirements engineering, analysis, system model, software prototyping, formal specification, algebraic specification, model based specification.

Project Management: Introduction to Project Management; Project Planning, Project Scheduling and Tracking, Software Metrics and measurement

Risk Management: S/W Risk, Risk Identification, Risk Projection, RMM.

Configuration Management: Introduction to Configuration management, versioning of software, Change Control, Software release, SCM standards, SCM Audit.

Design Concept And Methods: Design process, Architectural design, Object Oriented design, function-oriented design, real-time system design, user interface design.

Software Quality Assurance: Quality Models, SQA, S/W Reviews, statistical Quality Assurance

Change Request management: Requirements of software changes, change request management lifecycle, change request form, change request analysis and implementation.

Verification and Validation: Unit Testing, Integration Testing, Validation Testing.

Case Tool: Computer Aided software engineering, CASE workbenches, integrated CASE environments, Introduction to Rational Unified process and Rational Tools

Maintenance and Evolution: Client/Server software engineering, software maintenance, configuration management, software re-engineering, software reverse-engineering.

Maturity Models of Software Industry (CMM, 6sigma, PCMM, ISO 9001)

Self Study:

The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

References:

1. Roger S. Pressman, Software Engineering, McGraw-Hill International
2. Ian Sommerville, Software Engineering, Addison Wesley

3EC3264 Mobile Programming [3 0 0 3]

Course Learning Outcome:

After successful completion of the course, student will be able to

1. Understand the Mobile Programming Languages
2. Design and Develop Mobile User Interfaces
3. Understand Wireless connectivity, Mobile security and mobile development process
4. Select the technology for Mobile applications
5. Debug the mobile application implementation

Syllabus:

Mobile Programming Languages, XML, UML, Mobile Development Platforms (Android, Symbian, JavaME), Design & Development of Mobile User Interfaces ,Mobile Application Development & Debugging, Additional Dimensions of Mobile Application Development: Mobile agents and peer-to-peer architectures for mobile computing ,Wireless connectivity and mobile applications, Synchronization and replication of mobile data, Mobility and location information, Active transactions Mobile security, The mobile development process Architecture, design, and technology selection Mobile application implementation hurdles, Testing

Self Study:

The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

References:

1. Shambhu Upadhyaya, Abhijit Chaudhury, Kevin Kwiat and Mark Weiser Upadhyaya, Shambhu Chaudhury, Abhijit Kwiat, Kevin Weiser, Mark Mobile Computing: Implementing Pervasive Information and Communications Technologies, Springer
2. Rajkamal, Mobile Computing, Oxford University Press
3. Asoke K Talukder, Hasan Ahmed, Roopa Yavagal, Mobile Computing, Tata McGraw Hall
4. Reza B'Far, Mobile Computing Principles :Designing and Developing Mobile Applications withUML and XML, Cambridge University Press

3EC3274 Speech and Image Processing [3 0 0 3]

Course Learning Outcome:
After successful completion of the course, student will be able to

1. Understand the fundamentals of speech and Image processing
2. Understand various speech processing feature extraction methods and basic speaker recognition methods
3. Apply the knowledge of spatial domain and frequency domain filtering to solve variety of application
4. Understand various multi-resolution transforms and apply it to extract various features from a given image
5. Critically analyze various standard image restoration, segmentation, color and morphological image processing algorithms
6. Understand and critically analyze various recent year case study papers on speech, image or video processing algorithms and proposed novel solutions for case study problems

Syllabus:
Speech Processing: Production and Classification of Speech sounds: Introduction, Anatomy and physiology of speech production, categorization of speech sounds, Prodoy, speech perception, analysis and synthesis of Pole Zero speech models, STFT analysis and synthesis, wiener Filter based speech enhancement techniques, selected methods of speech enhancement. Speech Analysis and Synthesis.
Speaker Recognition: Spectral features for speaker recognition, speaker recognition algorithms, non spectral features in speaker recognition.
Digital Image Processing: Image Fundamentals, Image Enhancement;
Multiresolution Transform based Image Processing: Singular Value Decomposition, Discrete Wavelet Transform, Contourlet Transform, Statistical description of Images, PCA Transform; Image Segmentation: Detection of Discontinuities, Point Detection, Line Detection, Edge Detection, Edge Linking and Boundary Detection, Thresholding, Region-Based Segmentation.; Color Image Processing, Morphological Image Processing: Object Recognition: Object Representation, boundary descriptors, regional descriptors, regional descriptors, Parametric and Nonparametric Object Recognition Techniques, Pattern recognition Techniques and special topics on object recognition.

Self Study:
The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

References:

1. R. C. Gonzalez, R. E. Woods, Digital Image Processing, Addison Wesley.
2. Maria Petrou, Costas Petrou, Image Processing: The Fundamentals, John Wiley & Sons
3. K. Jain, Fundamentals of digital Image Processing , Prentice-Hall.
4. Thomos F. Quatieri, Discrete Time Speech Signal Processing Principles & Practice, Pearson Education

Course Code	**3EC3294**
Course Title	**Machine Learning for Embedded Systems**

L	T	P	C
3	-	-	3

Course Learning Outcomes (CLO):

At the end of the course, students will be able to -

1. comprehend various types of machine learning concepts and range of problems that can be handled by machine learning
2. analyze different machine learning algorithms
3. port machine learning algorithms on embedded platform
4. implement machine learning applications on various framework

Unit No.	Syllabus	Teaching hours: 45
I.	Machine Learning Introduction: Introduction, Concept of learning, designing a learning system, perspective and issues in machine learning, classification, regression, clustering, supervised and unsupervised learning, applications of machine learning and ML in embedded systems	03
II.	Regression Techniques Regression, Linear models for regression, Gradient Descent and Normal Equations Method, Multiple Linear Regression, Evaluation Measures for Regression Analysis	06
III.	Supervised Learning & Unsupervised learning Decision Trees, Bayesian Decision Theory, Parametric Methods, Dimensionality Reduction algorithms, kernel methods and reinforcement learning Unsupervised Learning :Clustering, k- means Algorithm, Linear models for classification, Expectation Maximization, Mixture of Gaussians	13
IV.	Ensemble Learning Techniques for generating base classifiers; techniques for combining classifiers, bootstrap, bagging, random forest, AdaBoost	06
V.	Neural Networks Introduction, Biological motivation, NN representation and learning, Perceptron, multi-layer networks and back propagation, introduction to Convolutional Neural Networks and Deep Learning	06
VI.	Machine Learning Hardware and Embedded Applications Machine Learning Hardware Tensor Flow TPU, machine learning algorithm implementation framework (open-source software libraries - Caffe, Torch, Theano), machine learning algorithms on hardware like GPU, CPU and FPGA. Machine Learning algorithms and applications on embedded systems (machine learning applications to IoT and cloud computing)	06
VII	Embedded Machine Learning Applications Machine Learning algorithms and applications on embedded systems (machine learning applications to IoT and cloud computing)	05

Suggested Readings:

1. T.M. Mitchell, Machine Learning, McGraw-Hill.
2. Chris Bishop, Pattern Recognition and Machine Learning, Springer.
3. Ethern Alpaydin, Introduction to Machine Learning, MIT Press.
4. Duda R.O. and Hart P.E., Pattern Classification and Scene Analysis, John Wiley & Sons, NewYork.
5. Ian Goodfellow et al, Deep Learning (Adaptive Computation and Machine Learning series),MIT Press
6. Brahmbhatt Samarth ,Practical OpenCV Implementation, APress
7. https://developer.arm.com/technologies/machine-learning-on-arm(accessed on Jan 22,2018)

L= Lecture, T= Tutorial, P= Practical, C= Credit

Course Code	3EC32104
Course Title	Internet of Things

Course Learning Outcomes (CLO):
At the end of the course, students will be able to –

1. analyze IOT needs and define business process of it
2. analyze available Hardware, Software and Data Management tools/technology/standards available for IoT design
3. design architecture for an IoT Application
4. apply IoT hardware and software tools/technology/standards in Real World Design Constraints

Unit No.	Syllabus	Teaching hours: 45
I.	IoT Introduction IoT Overview; Internet in general and Internet of Things: layers, protocols, packets, services, performance parameters, Applications, Potential and Challenges, Design principles, Need Capabilities, M2M and IoT fundamentals, Business Processes in IoT, Everything as a Service (XaaS), M2M and IoT Analytics	05
II.	IoT Architecture Reference Model and Architecture, IoT Reference Model: Functional View, Information View, Deployment and Operational View, Real-World Design Constraints, Data representation and Visualization, Interaction and remote control	06
III.	IoT Hardware Local and wide area networking standards, Networking Devices, DNS, NAT, Routers, Sensor Nodes, COTS, Industrial sensors, Actuators, Energy Storage Module, Power Management Module, RF Module, General Purpose Design boards, Collection and storage of sensor data on cloud	09
IV.	IoT Protocols PHY/MAC Layer, Wireless HART, Z-Wave, Bluetooth Low Energy, Zigbee Smart Energy, Network Layer- TCP, UDP, Socket Programming, IoT Application layer Protocols : HTTP, CoAP, XMPP, AMQP, MQTT protocols	10
V.	Service Layer Protocols and Security in IoT Applications	09
VI.	Case studies: Sensor body-Area-Network, Cloud of Things, Commercial Building Automation	06

Suggested Readings:

1. Pethuru Raj and Anupama C. Raman, The Internet of Things: Enabling Technologies, Platforms, and Use Cases, CRC Press.
2. Arshdeep Bahga and Vijay Madisetti, Internet of Things: A Hands-on Approach, Universities Press.
3. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, Academic Press.
4. Peter Waher, Learning Internet of Things, PACKT publishing.
5. Bernd Scholz-Reiter, Florian Michahelles, Architecting the Internet of Things, Springer.
6. Daniel Minoli, Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications, Willy Publications.
7. Peter Waher, Learning Internet of Things, Packt Publishing, 2015.
8. Editors Ovidiu Vermesan Peter Friess, Internet of Things – From Research and Innovation to Market 4. Deployment, River Publishers, 2014.
9. N. Ida, Sensors, Actuators and Their Interfaces, Scitech Publishers, 2014.

L= Lecture, T= Tutorial, P= Practical, C= Credit

3EC3206 Design Technologies Lab- II [0 0 2 2]

Course Learning Outcome:

After successful completion of the course, student will be able to

1. Select appropriate programming tool and technique for real time system design
2. Design, build, and debug Real Time Operating Systems program
3. Devise and conduct experiment to test Real Time Operating Systems program
4. Acquire good real time systems programming skills and understood good programming practices
5. Solve practical real time operating system problems using C programs with multiple processes (and/or multiple threads of execution)

Syllabus:

Lab Experiments based on DSP Processors, along with Porting RTOS and Embedded Operating Systems, Introduction to Bootloaders and Board Support Packages , Embedded File Systems ,Building RTOS / EOS Image for Target Hardware ,Time, Space and Power aware Programming,Design of a real-time data acquisition, processing & control system using the DSP Processor based Platforms

Self Study:

The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

Laboratory Work:

Laboratory work will be based on above syllabus with minimum 10 experiments to be incorporated.

3EC3207 High Performance Computing Lab [0 0 2 2]

Course Learning Outcome:

After successful completion of the course, student will be able to

1. Understand the concepts of basic and advance Progamming techniques using programming languages
2. Identify basic architectural elements of GPU hardware and acquire an appreciation of the importance of GPU for a broad class of engineering applications
3. Understand the basic procedure to do programming in CUDA and test various example projects on GPGPU Processors using CUDA and OPENCL
4. Simulate and analyze FFT, filtering, adaptive filters, basic transforms based algorithms and coding theory and their applications to engineering and embedded systems using appropriate simulation tools and implementations on GPGPU processors
5. Evaluate the trade-offs necessary in algorithm design for real-time DSP implementation

Syllabus:

Lab Experiments based on Graphics Processor Unit, OMAP, Networking for high performance computing

Case Studies : NVidia GPUs, Porting of Algorithms on Parallel Processing using parallel programming lanuauges (like OpenCL ,CUDA etc.), Programming of Muti core Systems

Self Study:

The self study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self study contents.

Laboratory Work:

Laboratory work will be based on above syllabus with minimum 10 experiments to be incorporated.

3EC3208 Seminar/Case Studies [0 0 1 1]

Course Learning Outcome:
After successful completion of the course, student will be able to

1. Carry out intense study on a specific topic related to current development in their field of specialization
2. Collect, interpret and analyze the information
3. Compare and evaluate the existing solutions for a specific cases study
4. Develop skills of presentation and report writing
5. Develop a skill to work in a team

A student is required to select advance topic relevant to subject studies. The student should refer and review literature from IEEE and equivalent journals/proceedings, prepare and present report based on this.Each student will be assign at least one case study for each will make a presentation and submit a report based on this.

M.Tech. Semester - II

L	T	P	C
-	-	10	5

Course Code	**3EC3210**
Course Title	**Minor Project**

Course Outcomes (COs):
At the end of the course, students will be able to -

1. Identify the issues related to the recent trends in the field of embedded systems.
2. Formulate the problem definition, analyse and do functional simulation of the same.
3. Design, implement, test and verify the engineering solution related to the problem definition.
4. Compile, comprehend and present the work carried out.

A student is required to carry out project work in the relevant area of post-graduate study. The project work may include design / simulation / synthesis / development of a system / fabrication of prototype, etc. At the end of the semester, a student has to submit a detailed report incorporating literature survey, problem formulation, clear problem statement, research methods, result analysis, conclusion, etc. It is expected that the student should defend his/her work before the jury / panel of examiners.

L = Lecture, T = Tutorial, P = Practical, C = Credit

SEMESTER-III

3EC3301 Major Project Part –I (Full Time) [0 0 0 15]

Course Learning Outcome:

After successful completion of the course, student will be able to

1. Understand the issues related with the recent trends in the field of engineering and its applications.
2. Formulate the problem definition, analyze and do functional simulation of the same
3. Design, Implement, test and verify the engineering solution related to problem definition.
4. Compile, Comprehend and Present the work carried out
5. Manage Project

A student is required to carry out elaborated project work. The project may be either design and fabrication work or a simulation and synthesis of a problem/system, develop algorithms and verify the feasibility on a computer. At the end of the semester student will be required to submit a detailed report of literature survey, design problem formulation, analysis, functional simulation and synthesis, work plan and work done and will defend his/her work carried out before examiners.

SEMESTER – IV

3EC3401 Major Project Part –II (Full Time) [0 0 0 13]

Course Learning Outcome:

After successful completion of the course, student will be able to

1. Understand the issues related with the recent trends in the field of engineering and its applications
2. Formulate the problem definition, analyze and do functional simulation of the same
3. Design, Implement, test and verify the engineering solution related to problem definition
4. Compile, Comprehend and Present the work carried out
5. Manage Project

A student is required to carry out elaborated project work. The project may be either design and fabrication work or a simulation and synthesis of a problem/system, develop algorithms and verify the feasibility on a computer. At the end of the semester student will be required to submit a detailed report of literature survey, design problem formulation, analysis, functional simulation and synthesis, work plan and work done and will defend his/her work carried out before examiners.

3EC3402 Comprehensive / Viva Voce [0 0 0 2]

Course Learning Outcome:

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

1. Exhibit the strength and grip on the fundamentals of the subjects studied in the previous semesters
2. Comprehend for all the courses studied in the entire programme

Each student will prepare for the subjects studied in the previous semesters and present it. the comprehensive presentation consists of the fundamentals of the subjects, their relevance in further study and in industry.