# College of Engineering, Pune

**(An Autonomous Institute of Govt. of Maharashtra, Permanently Affiliated to S.P. Pune University)**

## Department of Electronics and Telecommunication

Curriculum Structure & Detailed Syllabus (PG Program)

## M. Tech. – Electronics (VLSI and Embedded Systems)

**(Effective from: A.Y. 2019-20)**

**PG Program [M. Tech. Electronics – VLSI and Embedded Systems] Curriculum Structure**

**W. e. f AY 2019-20 and Applicable for batches admitted from AY 2019-20 to 2022-23**

## List of Abbreviations

| Abbreviation | Title | No of courses | Credits | % of Credits |
|---|---|---|---|---|
| PSMC | Program Specific Mathematics Course | 1 | 4 | 5.88 % |
| PSBC | Program Specific Bridge Course | 1 | 4 | 5.88 % |
| DEC | Department Elective Course | 3 | 9 | 13.24 % |
| MLC | Mandatory Learning Course | 2 | -- | -- |
| PCC | Program Core Course | 5 | 16 | 23.53 % |
| LC | Laboratory Course | 5 | 7 | 10.29 % |
| IOC | Interdisciplinary Open Course | 1 | 3 | 4.41 % |
| LLC | Liberal Learning Course | 1 | 1 | 1.47 % |
| SLC | Self Learning Course | 2 | 6 | 8.82 % |
| SBC | Skill Based Course | 2 | 18 | 26.47 % |
| Total | | **25** | **68** | **100 %** |

## PG Program [M. Tech. Electronics – VLSI and Embedded Systems] Curriculum Structure

### Semester I

| Sr. No. | Course Type | Course Code | Course Name | Teaching Scheme | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| 1. | PSMC | ETC-19002 | Probability, Graph and Field Theory | 3 | 1 | -- | 4 |
| 2. | PSBC | EVE-19001 | Programming Languages for Embedded Software | 3 | -- | 2 | 4 |
| 3. | DEC | <br>ETC(DE)19009<br>ETC(DE)19004<br>EVE(DE)19002<br>ETC(DE)19005 | Department Elective –I<br>a) Digital Signal and Image Processing<br>b) Machine Learning<br>c) Communication Busses and Standards<br>d) Automotive Embedded Product Development * | 3 | -- | -- | 3 |
| 4. | MLC | ML-19011 | Research Methodology and Intellectual Property Rights | 2 | -- | -- | -- |
| 5. | MLC | ML-19012 | Effective Technical Communication | 1 | -- | -- | -- |
| 6. | PCC | EVE-19002 | RTL Simulation and Synthesis | 3 | -- | -- | 3 |
| 7. | PCC | EVE-19003 | Microcontrollers: Architecture and Programming | 3 | -- | -- | 3 |
| 8. | LC | EVE-19004 | RTL Simulation and Synthesis Lab | -- | 1 | 2 | 2 |
| 9. | LC | EVE-19005 | Microcontrollers Programming Lab | -- | 1 | 2 | 2 |
| 10. | LC | EVE-19006 | Seminar | -- | -- | 2 | 1 |
| | | | **Total** | **18** | **3** | **8** | **22** |

_Note: ‘*’ Elective Courses to be offered to only Hella selected students._

## PG Program [M. Tech. Electronics – VLSI and Embedded Systems] Curriculum Structure

Interdisciplinary Open Course (IOC): Every department shall offer one IOC course (in Engineering/Science/Technology). A student can opt for an IOC course offered by a department except the one offered by his /her department.

### Semester II

| Sr. No. | Course Type | Course Code | Course Name | Teaching Scheme | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| 1. | IOC | ETC-19004 | Interdisciplinary Open Course | 3 | -- | -- | 3 |
| 2. | DEC | <br>ETC(DE)-19006<br>ETC(DE)-19007<br>EVE(DE)-19003<br>ETC(DE)-19001 | Department Elective –II<br>a) Internet of Things<br>b) Automotive Electronics<br>c) Embedded Systems Security<br>d) Automotive Embedded Hardware Development * | 3 | -- | -- | 3 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 3. | DEC | <br>EVE(DE)-19004<br>EVE(DE)-19005<br>ETC(DE)-19008<br>ETC(DE)-19002 | Department Elective –III<br>a) Analog CMOS VLSI Design<br>b) Memory Technologies<br>c) Low Power VLSI Design<br>d) Automotive Embedded Software Development * | 3 | -- | -- | 3 |
| 4. | LLC | LL-19001 | Liberal Learning Course | 1 | -- | -- | 1 |
| 6. | PCC | EVE-19007 | Digital CMOS VLSI Design | 3 | -- | -- | 3 |
| 7. | PCC | EVE-19008 | Linux in Embedded Systems | 3 | -- | -- | 3 |
| 8. | PCC | EVE-19009 | VLSI Design Verification and Testing | 2 | 1 | 2 | 4 |
| 9. | LC | EVE-19010 | CMOS VLSI Design Lab | -- | -- | 2 | 1 |
| 10. | LC | EVE-19011 | Linux in Embedded Systems Lab | -- | -- | 2 | 1 |
| | | | **Total** | **18** | **1** | **6** | **22** |

*Note: '*' Elective Courses to be offered to only Hella selected students.*

Interdisciplinary Open Course on "Embedded System Design" is offered to students of other departments.

## PG Program [M. Tech. Electronics – VLSI and Embedded Systems] Curriculum Structure

### Semester-III

| Sr. No. | Course Type | Course Code | Course Name | Teaching Scheme | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| 1. | SBC | EVE-20001 | Dissertation Phase – I | -- | -- | 18 | 9 |
| 2. | SLC | EVE-20002 | Massive Open Online Course –I | 3 | -- | -- | 3 |
| | | | **Total** | **3** | **--** | **18** | **12** |

### Semester-IV

| Sr. No. | Course Code | Course Code | Course Name | Teaching Scheme | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| 1. | SBC | EVE-20003 | Dissertation Phase – II | -- | -- | 18 | 9 |
| 2. | SLC | EVE-20004 | Massive Open Online Course –II | 3 | -- | -- | 3 |
| | | | **Total** | **3** | **--** | **18** | **12** |

MOOC Courses Identified:

- Real Time Embedded Systems
- VLSI design for Fault Tolerance and Testability
- Parallel Computing
- Advanced IOT Applications

**Semester I**

## (PSMC) [ETC-19002] Probability, Graph and Field Theory

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60 |
| Tutorials: 1 hr/week | Credits: 04 |

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Grasp and apply Graph theory for engineering problem solving and CAD tools developments
2. Culminate operations in groups, rings and field theory towards applications in digital electronic systems.
3. Characterize random variables and its functions with probability distributions and cumulative distributions

### Syllabus Contents:

**Graph Theory:**

Basic concepts of Graph Theory, Digraphs, Paths and Circuits, Reachability and Connectedness, Matrix representation of graphs, Subgraphs & Quotient Graphs, Isomorphic digraphs & Transitive Closure digraph, Euler's Path & Circuit- definitions and examples, Connected graphs and shortest paths: Walks, trails, paths, connected graphs, distance, cut-vertices, cut-edges, blocks, connectivity, weighted graphs, shortest path algorithms, Special classes of graphs: Bipartite graphs, line graphs, Chordal graphs, Planar graphs

**Group, Rings & Fields:**

Set theory Relations and Functions, Semigroups, Monoids, Subsemigroup, Submonoid, Isomorphism & Homomorphism, Fields, Integral Domain, Ring Homomorphism, Ring homomorphisms and their kernels, Ideals in commutative rings, First Isomorphism Theorem for commutative rings; the example of integers mod n, **Ab**elian groups, Quotient Groups, fields of characteristic p > 0, algebraic field extension, algebraic closure, perfect field, Galois groups of polynomials, Galois groups over rationals,, Field automorphisms. Automorphism group, decomposition of a permutation into a product of disjoint cycles, Homological algebra, polynomials, Algebraic extensions, Galois extensions, Galois group of a polynomial. Galois correspondence, Calculating Galois groups. Cyclotomic extensions, Applications to Cryptography, Basic definitions of cellular automata and symbolic dynamics

**Probability and Statistics:**

Definitions, conditional probability, Bayes Theorem and independence. - Random Variables: Discrete, continuous and mixed random variables, probability mass, probability density and cumulative distribution functions, mathematical expectation, moments, moment generating function, Chebyshev inequality. - Special Distributions: Discrete uniform, Binomial, Geometric, Poisson, Exponential, Gamma, Normal distributions. - Pseudo random sequence generation with given distribution, Functions of a Random Variable - Joint Distributions: Joint, marginal and conditional distributions, product moments, correlation, independence of random variables, bi-variate normal distribution. - Stochastic Processes: Definition and classification of stochastic processes, Poisson process - Norms, Statistical methods for ranking data

## References:

1. Kolman, Bernard, Robert C. Busby, and Sharon Cutler Ross, “Discrete mathematical structure” Prentice-Hall, Inc., 2003.
2. Dummit, David Steven, and Richard M. Foote, “Abstract algebra” Vol. 3. Hoboken: Wiley, 2004.
3. McIntosh, Harold V, “One dimensional Cellular Automata”, Luniver Press, 2009.
4. Hoekstra, Alfons G., Jiri Kroc, and Peter MA Sloot, eds., “Simulating complex systems by Cellular Automata”, Springer, 2010.
5. Steeb, Willi-Hans, “The nonlinear workbook: Chaos, fractals, cellular automata, genetic algorithms, gene expression programming, support vector machine, wavelets, hidden Markov models, fuzzy logic with C++”, World Scientific Publishing Company, 2014.
6. Bondy, John Adrian, and Uppaluri Siva Ramachandra Murty, “Graph theory with applications”, Vol. 290. London: Macmillan, 1976.
7. Baron, Michael, “Probability and statistics for computer scientists”, Chapman and Hall/CRC, 2013.
8. Web Resources http://mathworld.wolfram.com/ElementaryCellularAutomaton.html

## (PSBC) [EVE-19001] Programming Languages for Embedded Software

| Teaching Scheme | Examination Scheme |
|---|---|
| Lectures: 3 hrs/week<br>Practical: 2 hrs/week | T1 - 20, Assignments – 30, End-Sem Exam – 50 marks<br>Credits: 04 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Write an embedded C application of moderate complexity.
2. Develop and analyze algorithms in C++.
3. Differentiate interpreted languages from compiled languages.
4. Acquire programming skills in core Python
5. Develop applications using scripting languages such as Python.

### Syllabus Contents:

**Embedded ‘C’ Programming:**

- Bitwise operations, Dynamic memory allocation, OS services
- Linked stack and queue, Sparse matrices, Binary tree
- Interrupt handling in C, Code optimization issues
- Writing LCD drives, LED drivers, Drivers for serial port communication

- Embedded Software Development Cycle and Methods (Waterfall, Agile)

**Assignments:**

1. Develop a simple embedded application using C to be deployed on any hardware platform.
2. Write an LCD/LED/Serial port device driver in C.

**Object Oriented Programming with Embedded C++:**

- Differences between C and C++, Fundamentals of object oriented programming; OOP vs. Procedure oriented programming.
- OOP concepts: classes, objects, abstraction, polymorphism, inheritance, data binding and encapsulation.
- Basics of C++: features of C++, data types, standard I/O, arrays and strings in C++.
- Classes in C++, instantiation, creating objects and object scope, data abstraction, data encapsulation, constructors and destructors, methods and access modifiers, function and operator overloading
- Inheritance-Base and Derived classes, Inheritance types, Scope Resolution operator; polymorphism and virtual functions, exception handling

**Assignments:**

1. Implement a C++ program which reverses the words in a given sentence.
2. Write a complex application in C++ (e.g. CRC implementation) using Object Orientated design principles.

**Scripting Languages:**

- Overview of Scripting Languages – Python, Overview, History and Features
- Python: Variables, Operators, Decision Making, Loops, Strings, Lists, Tuples, Functions, Modules, Packages, Exceptions, Classes/Objects, Regular Expressions, CGI Programming, Database Access, Networking, Sending Email, Multithreading, XML Processing, GUI programming.

**Assignments:**

1. Write a program for Encryption and Decryption of messages in Python.
2. Write a Python program to read, write and append contents to the text and binary file
3. Write a simple Python script that serves a simple HTTP Response and a simple HTML Page.
4. Develop GUI for an Expression Calculator using ‘tk’.

## References:

1. Michael J. Pont , “Embedded C”, Pearson Education, 2nd Edition, 2008
2. Martin C. Brown,” Python: The Complete Reference”, McGraw-Hill Education, 2001
3. A. Michael Berman, “Data structures via C++”, Oxford University Press, 2002
4. Robert Sedgewick, “Algorithms in C++”, Addison Wesley Publishing Company, 1999
5. Abraham Silberschatz, Peter B, Greg Gagne, “Operating System Concepts”, John Willey & Sons, 2005
6. E. Balaguruswami, “Object-Oriented Programming With C++”, McGraw-Hill Education, 7th Edition, 2017

## (DEC-I) (a) [ETC(DE)-19009] Digital Signal and Image Processing

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60<br>Credits:03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Understand and apply knowledge of various transforms and probability theory in signal and image processing
2. Perform fundamental operations in digital image processing like enhancement, encoding, feature extraction, and segmentation.
3. Analyze, apply and critically evaluate various signal/image processing algorithms appropriate for practical applications.

### Syllabus Contents:

**Review of Discrete Time signals and systems**:

Characterization in time and Z and Fourier – domain, Fast Fourier Transform algorithms – In-place computations, Butterfly computations, bit reversals.

**Digital Filter design:**

FIR and IIR filters–Impulse invariance, bilinear transformation

**Representation of Signal Processing Algorithms:**

Signal flow, Data flow and Dependence Graphs. Iteration Bound, algorithms for computing iteration bound, Pipelining, Parallel processing.

**Introduction to Image Processing:**

Applications and fields of image processing, Fundamental steps in Digital image processing, Elements of visual perception, Image sensing and acquisition, Basic Concepts in Sampling and Quantization, representing digital images.

**Image Enhancement:**

Some basic gray level transformations, Histogram Processing, Sharpening Spatial filters, Image Enhancement in the spatial and Frequency domain, Pseudo-colouring

**Segmentation:**

Some Basic Relationships between pixels, point, Edge based segmentation, Boundary detection, extraction and representation, Threshold based segmentation, Region based segmentation, Texture based segmentation. Morphological operations

**Image Compression:**

Data redundancies Variable-length coding, Predictive coding, Transform coding, Image compression standards.

**Case studies:**

VLSI architectures for implementation of Image Processing algorithms

## References:

1. J.G. Proakis, Manolakis "Digital Signal Processing", Pearson, 4th Edition
2. Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Pearson , 4rd Edition
3. Keshab Parhi, "VLSI Digital Signal Processing Systems – Design and Implementation", Wiley India

## (DEC-I) (b) [ETC(DE)-19004] Machine Learning

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60 |
| | Credits: 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Grasp and develop machine learning algorithms namely linear, logistic and multivariate regression
2. Design and implement machine learning solutions to classification and clustering problems
3. Evaluate and interpret the results of the machine learning algorithms.

### Syllabus Contents:

Review of Probability Theory and Linear algebra, Convex Optimization, Introduction to Statistical Decision Theory, Regression: Linear Regression, Multivariate Regression, Subset Selection, Shrinkage Methods, Principal Component Regression, Logistic Regression, Partial Least Squares Classification: Linear Classification, LDA

Introduction to Perceptron and SVM, Neural Networks: Introduction, Early Models, Perceptron Learning, Back-propagation, Initialization of neural network, Training and Validation, Parameter Estimation

Decision Trees - Stopping Criterion and Pruning, Loss function, Categorical Attributes, Multiway Splits, Missing values, Instability, Regression Trees. Bootstrapping and Cross Validation, Class Evaluation, Measures, ROC curve, MDL, Ensemble methods, Committee Machines and Stacking.

Gradient Boosting, Random Forests, Multi-class Classification, Naïve Bayes, Bayesian Networks, Undirected Graphical Models, HMM, Variable elimination, Belief Propagation, Partitional clustering, Hierarchical Clustering, Birch Algorithm, CURE Algorithm, Density- Based Clustering, Gaussian Mixture Models, Expectation Maximization, Learning Theory, Re-enforcement Learning

### References:

1. Ethem Alpaydin, "Introduction to Machine Learning", PHI, 2005
2. Bishop Christopher, "Neural Networks for Pattern Recognition", New York, NY: Oxford University Press, ISBN: 9780198538646
3. Mitchell Tom, "Machine learning", New York, NY: McGraw-Hill, ISBN:9780070428072
4. Hastie, T. R. Tibshirani, and J. G. Friedman, "The Elements of Statistical Learning: Data

Mining, Inference and Prediction", New York, NY: Springer, ISBN:9780387952840

5. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani "Introduction to Statistical Learning", Springer, 2013.

## (DEC-I) (c) [EVE(DE)-19002] Communication Busses and Standards

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60<br>Credits: 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Select a particular serial bus suitable for a particular application.
2. Develop APIs for configuration, reading and writing data onto serial bus.
3. Design and develop peripheral interfaces attached to desired serial bus.
4. Observe and draw conclusions of TCP/IP protocols utilized in uni-cast and multicast packet transfers.

### Syllabus Contents:

**Serial Busses:**
Physical interface, Data and Control signals, features, limitations and applications of RS232, RS485, I$^2$C, SPI

**CAN:** Architecture, Data transmission, Layers, Frame formats, applications

**PCIe:** Revisions, Configuration space, Hardware protocols, applications

**USB**: Transfer types, enumeration, Descriptor types and contents, Device driver

**Data Streaming Serial Communication Protocol:**
Serial Front Panel Data Port (SFPDP) using fiber optic and copper cable

**TCP/IP:**
Bridges, Routers, LANs, Static and Dynamic routing, Multicast and Realtime service, Network Management and Security

### References:

1. Jan Axelson, "Serial Port Complete - COM Ports, USB Virtual Com Ports, and Ports for Embedded Systems ", Lakeview Research, 2nd Edition
2. Jan Axelson, "USB Complete", Penram Publications
3. Mike Jackson, Ravi Budruk, "PCI Express Technology", Mindshare Press
4. Wilfried Voss, "A Comprehensible Guide to Controller Area Network", Copperhill Media Corporation, 2nd Edition, 2005.
5. Serial Front Panel Draft Standard VITA 17.1 – 200x
6. Shivendra S. Panwar, Shiwen Mao, Jeong-dong Ryoo, Yihan Li, "TCP/IP Essentials", Cambridge University Press

7. Technical references on www.can-cia.org, www.pcisig.com, www.usb.org

## (DEC-I) (d) [ETC(DE)-19005] Automotive Embedded Product Development

| Teaching Scheme | Examination Scheme |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60<br>Credits: 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Acquire automotive product development understanding
2. Learn project management concept
3. Apply processes, methods and tools to demonstrate learning

### Syllabus Contents:

**Automotive System Overview & Product development:**

Major Automotive trends (e-mobility, Autonomous Driving, Comfort & Connected Cars), Vehicle EE architecture, Products. Integration of Mechanical, Software, Hardware domains and their interdependences, Design for x Abilities (manufacturability, testability, serviceability, maintainability), Overview of Design guidelines.

**Process, Methods & Tools:**

Requirements engineering and version control tools: DOORS, PTC, V model, Product Engineering Process, Automotive Spice, TS 16949, Key Performance Indicators for development.

**Product Reliability, Safety and Quality:**

DFMEA, PFMES, Warranty, Design Validations, Process Validations, Customer Line Return, Non Quality Expenses, First Pass Yield, Statistical tools, ASIL levels, Safety Goals, Safety Measures, HARA, FMEDA, ISO 26262

**Project Management and Organization:**

Matrix Organization, Line responsibilities, Functional responsibility, Team work, Leadership, Scope management, Scheduling, Cost, Monitoring & Tracking, Engineering Change Management, Milestones.

### References:

1. Online resources
2. Reference manuals from Hella-India

## (MLC) [ML-19011] Research Methodology and Intellectual Property Rights

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 2 hrs/week | Continuous evaluation:<br>Assignments/Presentations/Quizzes/Tests |

**Course Outcomes:**

At the end of the course, students will demonstrate the ability to:

1. Understand research problem formulation and approaches of investigation of solutions for research problems
2. Learn ethical practices to be followed in research and apply research methodology in case studies and acquire skills required for presentation of research outcomes
3. Discover how IPR is regarded as a source of national wealth and mark of an economic leadership in context of global market scenario
4. Summarize that it is an incentive for further research work and investment in R & D, leading to creation of new and better products and generation of economic and social benefits

**Syllabus Contents:**

**Unit 1:** [5 Hrs]

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

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

**Unit 2:** [5 Hrs]

Effective literature studies approaches, analysis

Use Design of Experiments /Taguchi Method to plan a set of experiments or simulations or build prototype

Analyze your results and draw conclusions or Build Prototype, Test and Redesign

**Unit 3:** [5 Hrs]

Plagiarism, Research ethics

Effective technical writing, how to write report, Paper.

Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee

**Unit 4 :** [4 Hrs]

Introduction to the concepts Property and Intellectual Property, Nature and Importance of Intellectual Property Rights, Objectives and Importance of understanding Intellectual Property Rights

**Unit 5 :** [7 Hrs]

Understanding the types of Intellectual Property Rights: -Patents-Indian Patent Office and its Administration, Administration of Patent System – Patenting under Indian Patent Act, Patent Rights and its Scope, Licensing and transfer of technology, Patent information and database. Provisional and Non Provisional Patent Application and Specification, Plant Patenting, Idea Patenting

Integrated Circuits, Industrial Designs, Trademarks (Registered and unregistered trademarks), Copyrights, Traditional Knowledge, Geographical Indications, Trade Secrets, Case Studies

**Unit 6 :** [4 Hrs]

New Developments in IPR, Process of Patenting and Development: technological research, innovation, patenting, development

International Scenario: WIPO, TRIPs, Patenting under PCT

### References:

1. Aswani Kumar Bansal, "Law of Trademarks in India"
2. B L Wadehr, "Law Relating to Patents, Trademarks, Copyright, Designs and Geographical Indications"
3. G.V.G Krishnamurthy, "The Law of Trademarks, Copyright, Patents and Design"
4. Satyawrat Ponkse, "The Management of Intellectual Property"
5. S K Roy Chaudhary & H K Saharay, "The Law of Trademarks, Copyright, Patents"
6. T. Ramappa, S. Chand, "Intellectual Property Rights under WTO"
7. Manual of "Patent Office Practice and Procedure"
8. WIPO: "WIPO Guide To Using Patent Information"
9. Halbert , "Resisting Intellectual Property", Taylor & Francis
10. Mayall , "Industrial Design", Mc Graw Hill
11. Niebel, "Product Design", Mc Graw Hill
12. Asimov , "Introduction to Design", Prentice Hall
13. Robert P. Merges, Peter S. Menell, Mark A. Lemley, "Intellectual Property in New Technological Age"

## (MLC) [ML-19012] Effective Technical Communication Skills

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 1 hr/week | Marks: 100 (4 Assignments - 25 Marks each)<br>Credits: -- |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Produce effective dialogue for business related situations
2. Use listening, speaking, reading and writing skills for communication purposes and attempt tasks by using functional grammar and vocabulary effectively
3. Analyze critically different concepts / principles of communication skills
4. Demonstrate productive skills and have a knack for structured conversations
5. Appreciate, analyze, evaluate business reports and research papers

### Syllabus Contents:

**Unit 1: Fundamentals of Communication** [4 Hrs]

7 Cs of communication, common errors in English, enriching vocabulary, styles and registers

**Unit 2: Aural-Oral Communication** [4 Hrs]

The art of listening, stress and intonation, group discussion, oral presentation skills

**Unit 3: Reading and Writing** [4 Hrs]

Types of reading, effective writing, business correspondence, interpretation of technical reports and research papers

### References:

1. Raman Sharma, “Technical Communication”, Oxford University Press.
2. Raymond Murphy, “Essential English Grammar” (Elementary & Intermediate) Cambridge University Press
3. Mark Hancock, “English Pronunciation in Use”, Cambridge University Press
4. Shirley Taylor, “Model Business Letters, Emails and Other Business Documents”, Prentice Hall, Seventh Edition
5. Thomas Huckin, Leslie Olsen, “Technical writing and Professional Communications for Non-native speakers of English”, McGraw Hill

## (PCC) [EVE-19002] RTL Simulation and Synthesis

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60<br>Credits: 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Design Finite state machines and perform timing analysis
2. Model, simulate, verify the synchronous digital design with Verilog HDL
3. Design, Analyze and Verify the synchronous digital design on FPGAs
4. Understand the concept of CPLD, FPGA, SoC and IP

### Syllabus Contents:

Top down approach, Hardware modelling of combinational and sequential circuits with verilog HDL, writing a test bench

Design of finite state machines (Synchronous and asynchronous), system design using ASM chart, Static Timing analysis, Meta-stability, clock issues, Need and design strategies for Multi-clock Domain designs.

Data path and Control path design, Arithmetic implementation strategies for data path design, Processor Design, Micro-programmed control design, Single cycle MMIPS

Programmable Logic Devices: Fine grained and coarse grained FPGA , Xilinx series

ASIC Design flow- Introduction to ASIC Design Flow, SOC, Floor planning, Placement, Clock tree synthesis, Routing, Physical verification, Power analysis, Design for performance, Low power VLSI design techniques, Technology Challenges

IP and Prototyping.

### References:

1. Stephen Brown and ZvonkoVranesic, “Fundamentals of Digital logic with Verilog Design”,

Mc-GrawHill, 3rd edition

2. Donald D Givone, “Digital principles and Design”, Tata Mc-Graw Hill, 2003
3. David Harris and Sarah Harris, “Digital Design and Computer Architecture”, Morgan Kaufmann, 2nd edition, 2012
4. Doug Amos, Austin Lesea, Rene Richter, “FPGA based prototyping methodology manual”, Xilinx
5. Douglas Smith, “HDL Chip Design: A Practical Guide for Designing, Synthesizing & Simulating ASICs & FPGAs Using VHDL or Verilog”, Doone Publications.
6. IEEE standard HDL based on Verilog HDL, published by IEEE.
7. Ben Cohen, “Real Chip design and Verification using Verilog and VHDL”, Vhdl Cohen Publishing, 2002

## (PCC) [EVE-19003] Microcontrollers: Architecture and Programming

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60 |
| | Credits: 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Examine and develop sample assembly language codes involving ARM processor core features.
2. Design and Develop code having sections with different processing priority using nested vectored interrupt controller features.
3. Utilize signal processing capability native to ARM processor core in development of small applications.
4. Develop small applications by utilizing the ARM processor core based SoC and development platform.

### Syllabus Contents:

**ARM Cortex-M3/M4 processor:**

Applications, Programming model – Registers, Operation modes, Exceptions and Interrupts, Reset Sequence, Introduction to the Tiva microcontrollers, Tiva block diagram, System Clocks and Control, Hibernation Module on Tiva

Instruction Set, Unified Assembler Language, Memory Maps, Memory Access Attributes, Permissions, Bit-Band Operations, Unaligned and Exclusive Transfers. Coretx-M4 specific instructions, Barrel shifter, Pipeline, Bus Interfaces

Exceptions, Types, Priority, Vector Tables, Interrupt Inputs and Pending behavior, Fault Exceptions, Supervisor and Pendable Service Call, Nested Vectored Interrupt Controller, Basic Configuration, SYSTICK Timer, Interrupt Sequences, Exits, Tail Chaining, Interrupt Latency

Floating point operations, Floating point unit (FPU), Registers, Lazy stacking

**Assembly and Mixed Language Applications:**
Assembly code in applications, Interaction between C and assembly

**TM4C123x microcontroller**:
Internal memory, GPIOs, Timers, ADC, UART, SPI and USB and other serial interfaces, PWM, RTC, WDT

**ARM Cortex M4 – DSP Support:**
DSP on a microcontroller, Architecture of traditional DSP processor, DSP instructions – fractional arithmetic, SIMD data, load store instructions, arithmetic instructions, optimization strategies, DSP applications

**Processor benchmarking**

## References:

1. Joseph Yiu, "The definitive guide to ARM Cortex-M3 and Cortex-M4 Processors", Elsevier, 3rd Edition
2. Jonathan W Valvano, "Embedded Systems: Real-Time Interfacing to ARM Cortex-M Microcontrollers", Createspace publications ISBN: 978-1463590154, 2014.
3. Sloss Andrew N, Symes Dominic, Wright Chris, "ARM System Developer's Guide: Designing and Optimizing", Morgan Kaufman Publication
4. Steve furber, "ARM System-on-Chip Architecture", Pearson Education
5. Venkatramani B. and Bhaskar M. "Digital Signal Processors: Architecture, Programming and Applications" , TMH , 2nd Edition
6. Frank Vahid and Tony Givargis, "Embedded System Design", Wiley
7. Technical references and user manuals on www.arm.com, www.ti.com

## (LC) [EVE-19004] RTL Simulation and Synthesis Lab

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Tutorials: 1 hr/week | Marks- 100 |
| Practical: 2 hrs/week | Credits: 02 |

### Laboratory Outcomes:

At the end of the laboratory work, students will demonstrate the ability to:

1. Model, simulate, verify, synthesize and implement RTL design with Verilog on FPGA
2. Identify, formulate, solve and implement problems in domains of signal processing, communication systems etc. using RTL design tools.
3. Use EDA tools like Cadence, Mentor Graphics and Xilinx.

### List of Experiments:

Verilog implementation of:

1. MUX/DEMUX, Full Adder, 8-bit magnitude comparator, encoder/ decoder, priority encoder, D FF, 4 bit Shift registers (SISO, SIPO, PISO, bidirectional), Synchronous Counters, binary to gray converter, parity generator.
2. Sequence generator / detectors, Synchronous FSM – Mealy and Moore machines.
3. Vending machines - Traffic Light controller, ATM, elevator control.
4. Interfacing with peripherals like seven segment display, LED, UART, Custom GPIO, Temp sensor, Ethernet, DDR.
5. SPI, I2C, PCI protocols, Bus Arbiter, UART
6. Single port SRAM, Synchronous and Asynchronous FIFO
7. MIPS ( Microprocessor without interlock pipeline stages) design
8. Lab based on physical design (STA) using Encounter tool

***Note:*** Journal Submission is in the form of CD. It should contain HDL codes, snapshot of results, synthesis reports, RTL view. In case the programs are downloaded on FPGA / SoC, the pre & post synthesis and implementation reports should also be submitted.

## (LC) [EVE-19005] Microcontrollers Programming Lab

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Tutorials: 1 hr/week | Marks- 100 |
| Practical: 3 hrs/week | Credits: 02 |

### Laboratory Outcomes:

At the end of the laboratory work, students will demonstrate the ability to:

1. Develop prototype codes using commonly available on and off chip peripherals with and without interrupts on Cortex M3/M4 development boards
2. Develop small DSP applications using CMSIS–DSP library routines with floating and fixed point versions and analyse the performance.

### List of Experiments:

Experiments to be carried out on Tiva Launch-pads:

1. Blink an LED with software delay, delay generated using the SysTick timer.
2. System clock real time alteration using the PLL modules.
3. Control intensity of an LED using PWM implemented in software and hardware.
4. Control an LED using switch by polling method, by interrupt method and flash the LED once every five switch presses.
5. Key matrix and alphanumeric LCD interfacing and programming.
6. UART programming with accessing TX ad RX buffers directly and using DMA.
7. Recording of analog readings at the output of rotary potentiometer connected to ADC channel.
8. Programming (ISL 29023) Ambient and Infrared Light sensor available on Sensor Hub Booster Pack using $I^2C$ interface.
9. Calling C functions from assembly programs and vice versa.
10. To develop an assembly code and C code to compute Euclidian distance between any two points
11. To develop assembly and C code for implementation of convolution operation
12. To design and implement filters in C to enhance the features of given input sequence/signal

## (LC) [EVE-19006] Seminar

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Practical: 2 hrs/week | Marks- 100 |
| | Credits: 01 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Identify contemporary topics/concepts pertaining to VLSI and Embedded Systems and prepare documentation with improved substance.
2. Present the selected topic with superiority demonstrating good communication

skills.

## Guidelines:

**Selection of Topic:**

- Select a topic relevant to the stream of study with content suitable for M. Tech. level presentation. For selection topics refer internationally reputed journals. The primary reference should be published during the last two or three years.
- Some of the journals/publications suitable for reference are: IEEE/the IET/IETE/Springer/Science Direct/ACM journals in the areas of Computing and Processing (Hardware and Software), Circuits-Devices and Systems, Communication - Networking and Security, Robotics and Control Systems, Signal Processing and Analysis, Machine Learning, IoT and any other related domain
- Get the topic approved by the seminar guide well in advance.

**Preparation of Presentation and Report:**

- In slides, list out key point only. You may include figures, charts equations tables etc. but not running paragraphs. Font size used should be at least 20.
- Figures should be very clear and possibly drawn by you using suitable software tools. There should be a slide on “Conclusion”.
- A report of the seminar should be prepared which should contain the following.
  - Title of the seminar.
  - Name and other details of presenter and the guide.
  - Abstract of the topic.
  - Contents such as Introduction, Theory to elaborate the concept, Implementation if carried out by the presenter/or fellow researcher/s, Comparison with other relevant techniques, Conclusion etc.
  - List of references strictly in IEEE format.

**Oral Presentation:**

- Student needs to orally present the topic for 20 minutes with good voice projection and with modest pace

**Answering Queries:**

- Student needs to answer queries raised by the audience and evaluators. This session shall be restricted to 5 minutes. In case of more queries, student is supposed to solve the queries offline.

**Semester II**

## (IOC) [ETC-19004] Embedded System Design

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60<br>Credits-03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Implement simple sketches on the Arduino boards involving several peripherals
2. Identify, design and implement applications on the Arduino boards producing custom shields.
3. Deploy low end applications using low and high level languages on microcontroller platform.

### Syllabus Contents:

**Introduction to processors:**
Introduction of Microprocessors and Microcontrollers, Introduction of Arduino Microcontrollers.

**Introduction to architecture**:
Atmega328: Basics and Architecture, Instruction Set

**Arduino programming:**
Arduino programming basics, Analog/Digital components and its application with Arduino, IDE for Arduino.

**Other utilities in Arduino:**
Timers, Analog comparators and hardware interrupts

**Interfacing with peripherals:**
Communication buses, Interfacing of I/O devices

**Case studies:**
Case studies of a few projects using Arduino boards and Shields

### References:

1. Brian Evans, “Beginning Arduino Programming”, Springer, 2011
2. Michael J. Pont , “Embedded C”, Pearson Education, 2nd Edition, 2008
3. Raj Kamal, “ Embedded Systems – Architecture: Programming and Design”, TMH
4. Frank Vahid and Tony Givargis, “Embedded System Design”, Wiley

## (DEC-II) (a) [ETC(DE)-19006] Internet of Things

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60<br>Credits: 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Identify IoT technologies and their types that are in use and appreciate requirements in typical scenarios.
2. Apply available technologies to tackle scenarios in terms of using experimental platforms for implementing prototypes and testing them as running applications.
3. Analyze and evaluate the data received through sensors in IoT.

### Syllabus Contents:

Introduction to IoT, Sensing, Actuation, Sensor Networks, Wireless sensor networks: introduction, Edge resource pooling and caching, client side control, and configuration.

Basics of Networking, Communication Protocols: IPV6, 6LowPAN, CoAP, MQTT, Machine-to-Machine Communications.

Operating system requirement for IoT, Smart objects as building blocks for IoT, Embedded systems platforms for IoT, IO drivers.

Software Defined Networks (SDN), From Cloud to Fog and MIST networking for IoT Communications, Principles of Edge/P2P networking, Cloud and Fog Ecosystem for IoT Review of architecture,

OLAP and OLTP, NoSQL databases, Row and column Oriented databases, Introduction to Columnar DBMS CStore, Run: Length and Bit vector Encoding, Integrating Compression, and Query Execution in Columnar databases.

Industrial IoT, Case Study: Agriculture, Healthcare, Activity Monitoring.

### References:

1. A Bahaga, V. Madisetti, “Internet of Things- Hands on approach”, VPT publisher, 2014.
2. A. McEwen, H. Cassimally, “Designing the Internet of Things”, Wiley, 2013.
3. Cuno Pfister, “Getting started with Internet of Things”, Maker Media, 1st edition, 2011.
4. Samuel Greenguard, “Internet of things”, MIT Press.
5. Big-Data Analytics for Cloud, IoT and Cognitive Computing- Kai Hwang, Min Chen
6. http://www.datamation.com/open-source/35-open-source-tools-for-the-internet-ofthings-1.html
7. https://developer.mbed.org/handbook/AnalogIn
8. http://www.libelium.com/50_sensor_applications M2MLabs Mainspring
9. http://www.m2mlabs.com/framework Node-RED
10. http://nodered.org

## (DEC-II) (b) [ETC(DE)-19007] Automotive Electronics

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60<br>Credits: 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Determine the function and operation of various modules and sensors and their utilization in the management of the vehicle control.
2. Recognise and understand the different wiring diagrams used in manufacturer's workshop manuals.
3. Realize importance of comfort, safety and interference related issues in an automotive.

### Syllabus Contents:

**Introduction:**

Microprocessor and micro Computer applications in automobiles; components for engine management system, chassis system, motion control; electronic panel meters.

**Sensors & Actuators:**

Different types of Sensors such as oxygen sensors, crank angle position sensors, fuel metering/vehicle speed sensors and detonation sensors, altitude sensors, flow Sensors, throttle position sensors, Solenoids, stepper motors, relays.

**CAN:**

Architecture, Data transmission, Layers, Frame formats, applications

**Fuel Injection & Ignition System:**

Feedback carburettor system; throttle body injection and multi point fuel injection System; injection system controls; electronic spark timing.

**Engine Control System:**

Open loop and closed loop control system; engine cooling and warm-up control; acceleration, deceleration and idle speed control; integrated engine control system; exhaust emission control engineering; on-board diagnostics

**Automotive Electrical:**

Batteries; starter motor & drive mechanism; D.C. generator and alternator; lighting design; dashboard instruments; horn, warning system and safety devices.

**Comfort & Safety:**

Seats, mirrors and sun roofs; central locking and electronic windows; cruise control; in-car multimedia; security; airbag and belt tensioners

Electromagnetic Interference Suppression, Electromagnetic compatibility

### References:

1. Ronald K. Jurgen, "Automotive Electronics Handbook", MGH

2. Al Santini , “Automotive Electricity and Electronics”, Delmar Publishers, NY
3. Young, Griffitns, “Automobile Electrical & Electronic Equipments”, Butterworth Publication, London.
4. Bechfold, “Understanding Automotive Electronics”, SAE, 1998.

## (DEC-II) (c) [EVE(DE)-19003] Embedded Systems Security

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60 |
| | Credits: 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Recognize vulnerabilities, attacks and need of protection mechanisms for embedded systems
2. Analyze and evaluate software vulnerabilities and attacks on operating systems
3. Identify terms/concepts relevant to embedded cryptography.
4. Develop and deploy solutions for security of embedded software and data protection.

### Syllabus Contents:

**Introduction to Embedded Systems Security:** Trends, Policies, Threats

**Systems Software Considerations:**
Role of the Operating System, Multiple Independent Levels of Security, Microkernel versus Monolith, Core Embedded Operating System Security Requirements, Access Control and Capabilities, Hypervisors and System Virtualization, I/O Virtualization, Remote Management, Assuring Integrity of the TCB

**Secure Embedded Software Development:**
PHASE—Principles of High-Assurance Software Engineering, Minimal Implementation, Component Architecture, Least Privilege, Secure Development Process, Independent Expert Validation, Case Study: HAWS—High-Assurance Web Server, Model-Driven Design

**Embedded Cryptography:**
Cryptographic Modes, Block Ciphers, Authenticated Encryption, Public Key Cryptography, Key Agreement, Public Key Authentication, Elliptic Curve Cryptography, Cryptographic Hashes, Message Authentication Codes, Random Number Generation, Key Management for Embedded Systems, Cryptographic Certifications

**Data Protection Protocols for Embedded Systems:**

Data-in-Motion Protocols, Data-at-Rest Protocols

**Emerging Applications:** Embedded Network Transactions, Automotive Security, Secure Android,

Next-Generation Software-Defined Radio

### References:

1. David Kleidermacher and Mike Kleidermacher, “Embedded Systems Security”, Elsevier
2. Gebotys, Catherine H., “Security in Embedded Devices”, Springer
3. Stapko T., “Practical Embedded Security”, Elsevier/Newnes

## (DEC-II) (d) [ETC(DE)-19001] Automotive Embedded Hardware Development

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 30 marks each, End-Sem Exam – 40 |
| | Credits: 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Acquire automotive specific hardware design skills
2. Understand concepts such as DFM, DFT, EMC, DFMEA
3. Apply processes, methods and tools to demonstrate design skills

### Syllabus Contents:

**Low Power Domain:**

16/32 bit controllers, Hardware-Software interfaces, Communication interfaces – CAN, LIN, SPI, Wireless interfaces – Bluetooth, ISM band applications, I/O interfaces – digital, analog signal conditioning, switches, relays, high side, low side drivers, Introduction to design tools (Microcap, Cadence Concept HDL and Allegro)

**High Power Domain:**

Selection of power switches devices – MOSFETs/IGBTs/SiC/GaNFETs, Gate driver design, power loss calculations, thermal management, and Design considerations for High Voltage applications.

**Electromagnetic Compatibility:**

Introduction to various regulatory requirements and International electrical and EMC standards, Understanding origin of pulses, disturbances, circuit and PCB layout design techniques to meet EMC.

**Design for Manufacturability and Testability :**

PCB Layout considerations, Manufacturing interfaces and process flow, ICT, AOI and EOL testing.

### References:

1. Online resources
2. Reference manuals from Hella-India

*Note:* Complementary lab session will be organized to ensure hands-on learning.

## (DEC-III) (a) [EVE(DE)-19004] Analog CMOS VLSI Design

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60<br>Credits: 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Understand and apply design metric concepts to simple and complex analog circuits using CMOS.
2. Analyze and design the simple and complex circuits to create the analog building blocks of a system.
3. Simulate complex digital circuits using EDA tools like Cadence, Mentor Graphics and other open source software tools like NGSPICE.

### Syllabus Contents:

**Foundation**:

Analog Landscape, Right design Thinking, Techniques for intuitive and graphical understanding, Understanding building blocks of analog circuits viz R, L, C and dependent V and I sources and their combinations, Large Signal Models of MOS Transistors: I-V Characteristics, Early Effect, Channel Length Modulation, Back Gate Effect and other Second-Order Effects.

**Analog Sub-circuits**:

Single Stage Amplifier: CS stage with resistance load, Divide connected load, Current source load, Triode load, CS stage with source degeneration, Source follower, Common-gate stage, Cascade stage, Choice of device models.

**Frequency response**:

CS stage, Source follower, Common gate stage, Cascade stage and difference pair, Noise analysis

**Differential Amplifiers**:

Basic difference pair, Common mode response, Differential pair with MOS loads, Gilbert cell. Cascade current mirrors, Active current mirrors.

**Operational amplifiers design**:

One stage OPAMP, Two stage OPAMP, Gain boosting, Slew rate, PSRR. Design equations and procedure

**Basic Compensation techniques**:

Compensation of 2 stage OPAMP, Other compensation techniques.

**Advanced OPAMP:**

Cascode and folded cascode op-amps, common mode feedback techniques

**Voltage References:**

Basic Design and Evaluation of Band Gap Reference, and CMOS Band Gap References MOS

**Voltage Comparators**:
Various Configurations and Offset Cancellation Techniques

**Switched Capacitor circuits**:
Basic Switched Capacitor Integrators, Z-transforms, Switched Capacitor Filter Design, MOSFET-C Filters

**Introduction to data converters**:
Digital-to-Analog and analog to digital converters specifications understanding, Current scaling DAC, Voltage scaling DAC charge scaling DAC, Extending resolution of parallel DAC, similar scaled DACs High speed ADCS, parallel or flash ADCS, interpolating ADCS, folding ADCS, Multi-bit pipeline ADCS delta sigma modular, Decimators filters

### References:

1. Behzad Razavi,"Design of Analog CMOS Integrated Circuits", Tata Mc Graw-Hill.
2. Allen Holberg,"CMOS analog Circuit Design", Oxford University Press.
3. Mohammed Ismail, Terri Fiez, " Analog VLSI Signal and Information Processing",Mc Graw Hill International Editions.
4. Roubik Gregorian, Gabor C. Temes, "Analog MOS Integrated Circuits for Signal Processing", Wiley series on filters.
5. Gray Hurst Lewis," Analysis and Design of Analog Integrated Circuits", Fourth Edition

## (DEC-III) (b) [EVE(DE)-19005] Memory Technologies

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60<br>Credits: 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Select architecture and design semiconductor memory circuits and subsystems.
2. Identify reliability and radiation effects on memory
3. Work with the state-of-the-art memory chip design

### Syllabus Contents:

**Random Access Memory Technologies:**
Static Random Access Memories (SRAMs), SRAM Cell Structures, MOS SRAM Architecture, MOS SRAM Cell and Peripheral Circuit, Bipolar SRAM, Advanced SRAM Architectures, Application Specific SRAMs

DRAMs, MOS DRAM Cell, BiCMOS DRAM, Error Failures in DRAM, Advanced DRAM Design and Architecture, Application Specific DRAMs.

SRAM and DRAM Memory controllers

**Non-Volatile Memories:**

Masked ROMs, PROMs, Bipolar & CMOS PROM, EEPROMs, Floating Gate EPROM Cell, OTP EPROM, EEPROMs, Nonvolatile SRAM, Flash Memories

**Semiconductor Memory Reliability and Radiation Effects:**

General Reliability Issues, RAM Failure Modes and Mechanism, Nonvolatile Memory, Radiation Effects, SEP, Radiation Hardening Techniques. Process and Design Issues, Radiation Hardened Memory Characteristics, Radiation Hardness Assurance and Testing.

**Advanced Memory Technologies and High-density Memory Packing Technologies:**

Ferroelectric Random Access Memories (FRAMs), Gallium Arsenide (GaAs) FRAMs, Analog Memories, Magneto Resistive Random Access Memories (MRAMs), Experimental Memory Devices. Memory Hybrids (2D & 3D), Memory Stacks, Memory Testing and Reliability Issues, Memory Cards, High Density Memory Packaging, Introduction to digital tablet PC, LCD, DVD

## References:

1. Ashok K Sharma," Semiconductor Memories: Technology, Testing and Reliability , PHI
2. Ashok K Sharma, "Advanced Semiconductor Memories: Architectures, Designs and Applications", Wiley Interscience
3. Luecke Mize Care, "Semiconductor Memory design & application", Mc-Graw Hill.
4. Kiyoo Itoh, "VLSI memory chip design", Springer International Edition

## (DEC-III) (c) [ETC(DE)-19008] Low Power VLSI Design

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60 |
| | Credits: 03 |

## Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Identify the sources of power dissipation in digital IC systems & understand the impact of power on system performance and reliability. Understand leakage sources and reduction techniques.
2. Characterize and model power consumption & understand the basic analysis methods.
3. Use EDA tools like Cadence, Mentor Graphics and other open source software tools like NGSPICE.

## Syllabus Contents:

**Power Estimation and Control in CMOS VLSI Circuits:**

System performance, growing technology challenges, existing low power design techniques, realistic estimation expectations, low power – the need, issues in low power design, design transformations, low power arithmetic components- circuit design styles, adders, multipliers.

**Low Power design Techniques:**
Major component of power dissipation, Threshold voltage scaling, Impact of scaling on dynamic power, circuit topologies for low power, Leakage power contributors and remedies, Low power design technique, DVFS, pipelining, parallel processing.

**Low Power Memory Design:**
Transistor Scaling and leakage, leakage control, Circuit level leakage reduction techniques, MTCMOS, SCCMOS, VTCMOS, DTMOS, Cache Memories leakage, Noise margin, Parametric Failures in SRAM, SRAM leakage reduction schemes, Supply gating, Iddq test, Stuck at fault tests, test power, test coverage, gated de-cap.

**Low Power Clock Distribution:**
Power dissipation in clock distribution, single driver versus distributed buffers, buffers & device sizing under process variations, zero skew Vs. tolerable skew, chip & package co-design of clock network.

**Low Power Microprocessor Design System:**
Power management support, architectural trade-offs for power, choosing the supply voltage, low-power clocking, implementation problem for low power, comparison of microprocessors for power & performance.

### References:

1. P. Rashinkar, Paterson and L. Singh, "Low Power Design Methodologies", Kluwer Academic, 2002.
2. Kaushik Roy, Sharat Prasad, "Low power CMOS VLSI circuit design", John Wiley sons Inc.,2000.
3. J.B.Kulo and J.H Lou, "Low voltage CMOS VLSI Circuits", Wiley, 1999.
4. A. P. Chandrasekaran and R. W. Broadersen, "Low power digital CMOS design", Kluwer, 1995.
5. Gary Yeap, "Practical low power digital VLSI design", Kluwer, 1998.

## (DEC-III) (d) [ETC(DE)-19002] Automotive Embedded Software Development

| Teaching Scheme | Examination Scheme |
|---|---|
| Lectures: 3 hrs/week | Assignments- 60 marks, End-Sem Exam – 40<br>Credits: 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Acquire automotive specific software design skills
2. Understand concepts such as AUTOSAR, MATLAB, Communication Protocol
3. Apply processes, methods and tools to demonstrate design skills

## Syllabus Contents:

**Software Architecture:**

Classical architecture, Layered Architecture (AUTOSAR), All layer information (e.g. RTE, BSW, Application) Tool: Davinci developer, Configurator, Raphsody.

**Communication Protocols:**

CAN, LIN, Automotive Ethernet, RF, Bluetooth, Wi-Fi, Diagnostic Protocols: UDS, Tools: CANoe, Vehicle spy, CAPEL, TAE scripting.

**Model Based Development:**

Algorithm/application development using Simulink, Stateflow, Code generator.

**Embedded C:**

Concept of C (structure, union, pointer, bitwise operator), Logic building according to requirement, MISRA C guidelines.

**Software Testing:**

Unit testing, Model in loop (MIL) testing, Module testing, Integration testing, Software in loop (SIL) testing, Hardware in loop (HIL) testing, Tools: Tessy, Polyspace, TPT, Winidea, QAC, HIL test setup.

## References:

1. Online resources
2. Reference manuals from Hella-India

*Note:* Complementary lab session will be organized to ensure hands-on learning.

## (LLC) [] Liberal Learning Course

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Contact Period: 1 hr/week | T1, T2 – 20 marks each, End-Sem Exam – 60<br>Credits: 01 |

## Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Learn new topics from various disciplines without any structured teaching or tutoring
2. Understand qualitative attributes of a good learner
3. Understand quantitative measurements of learning approaches and learning styles
4. Understand various sources and avenues to harvest/gather information
5. Assess yourself at various stages of learning

## Course Features:

- 10 Areas, Sub areas in each
- Voluntary selection
- Areas (Sub areas):

1. Agriculture (Landscaping, Farming, etc.)
2. Business (Management, Entrepreneurship, etc.)

3. Defense (Study about functioning of Armed Forces)
4. Education (Education system, Policies, Importance, etc.)
5. Fine Arts (Painting, Sculpting, Sketching, etc.)
6. Linguistics
7. Medicine and Health (Diseases, Remedies, Nutrition, Dietetics, etc.)
8. Performing Arts (Music, Dance, Instruments, Drama, etc.)
9. Philosophy
10. Social Sciences (History, Political Science, Archaeology, Geography, Civics, Economics, etc.)

### Evaluation:

- **T1:** A brief format about your reason for selecting the area, sub area, topic and a list of 5 questions (20 marks)
- **T2:** Identify and meet an expert (in or outside college) in your choice of topic and give a write up about their ideas regarding your topic (video /audio recording of your conversation permitted (20 marks)
- **ESE:** Presentation in the form of PPT, demonstration, performance, charts, etc. in front of everyone involved in your sub area and one external expert (60 marks)

### Resources:

- Expert (s), Books, Texts, Newspaper, Magazines, Research Papers, Journal, Discussion with peers or faculty, Internet, etc.

## (PCC) [EVE-19007] Digital CMOS VLSI Design

| Teaching Scheme | Examination Scheme |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 20 marks each, End-Sem Exam – 60 |
| | Credits: 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Understand and apply design metric concepts to simple and complex digital circuits using CMOS.
2. Analyze and design the simple and complex circuits to create the building blocks of a system.
3. Simulate complex digital circuits using EDA tools like Cadence, Mentor Graphics and other open source software tools like NGSPICE.

### Syllabus Contents:

Introduction to VLSI, Manufacturing process of CMOS integrated circuits, CMOS n-well process design rules, packaging integrated circuits, trends in process technology. MOS transistor, Energy band diagram of MOS system, Threshold voltage equation, Secondary effects in MOSFETS.

Technology Scaling and Road map, Scaling issues, Standard 4 mask NMOS Fabrication process. MOS capacitances, Modeling of MOS transistors using SPICE level I and II equations, capacitance models. Electrical wire models, SPICE wire models.

**Quality metrics of a digital design:**
Cost, Functionality, Robustness, Power, and Delay. CMOS inverter: Switching Threshold, Noise Margin, Dynamic behavior of CMOS inverter, computing capacitances, propagation delay, Dynamic power consumption, static power consumption, energy, and energy delay product calculations, stick diagram, I C layout design and tools.

**Combinational logic:**
Static CMOS design, Logic effort, Ratioed logic, Pass transistor logic, Dynamic logic, Speed and power dissipation in dynamic logic, Cascading dynamic gates, CMOS transmission gate logic.

**Sequential logic:**
Static latches and registers, Bi-stability principle, MUX based latches, Static SR flip-flops, Master-slave edge-triggered register, Dynamic latches and registers, Concept of pipelining, Pulse registers, non-bistable sequential circuit.

**Advanced technologies:**
Giga-scale dilemma, Short channel effects, High–k, Metal Gate Technology, FinFET, TFET etc.

**Physical design flow:**
Floor planning, Placement, Routing, CTS, Power analysis and IR drop estimation - static and dynamic, ESD protection-human body model, Machine model.

## References:

1. Jan M Rabaey, AnanthaChadrakasan, Borivoje Nikolic, “Digital integrated circuits a design perspective”, Pearson education.
2. Sung Mo Kang, Yusuf Leblebici, “CMOS digital integrated circuits”, TataMcGraw Hill Publication.
3. Neil E Weste, Kamran Eshraghian, “Principle of CMOS VLSI Design”, Pearsoneducation.
4. Baker Li Boyce, “CMOS Circuit Design, Layout, and Simulation”, Wiley, 2nd Edition.
5. Pucknell, D.A. and Eshraghian, K., “Basic VLSI Design”, PHI, 3rd Edition.

## (PCC) [EVE-19008] Linux in Embedded Systems

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Lectures: 3 hrs/week | T1, T2 – 30 marks each, End-Sem Exam – 40<br>Credits - 03 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Create complex applications with multiple processes and threads incorporating

synchronization and inter-process communication features

2. Recognize the standard Linux and Embedded file systems and emulate simple tasks based on file system.
3. Develop pseudo and specific Linux device drivers in order to support standard and custom resources
4. Understand of the embedded Linux development model.

## Syllabus Contents:

**Linux OS Introduction:**
User/Kernel Model, Processes, Daemons, Threads, System Calls, Shell, Shell, Virtual Memory. Executable file layout

**User Level Programming:**
Creating Processes, Linking/Loading, Signals, Shared Library, Threads and multithreaded program, Semaphores, Mutex, IPC mechanism Pipes, Shared memory.

**Kernel Internals Basics:**
Process Internal representation, Linux File System Abstraction, Virtual File system, iNodes, files, /proc, Kernel Queue Data Structure, Memory Allocation (buddy system, slab cache), Embedded File systems

**Working with Kernel Artifacts:**
Kernel Layers, Basic Driver Architecture, Block & character driver distinction, Low level drivers, OS drivers etc, Device major, minor number, Interfaces to driver read, write, ioctl etc, Blocking and non-blocking calls, Semaphores, Mutex Multi core Synchronization, and spin locks, Proc & Sysfs interfaces, Block driver examples

**Interrupt Management:**
Interrupt Handling in Normal Processor, Traditional ARM7 multi mode interrupts, Interrupt Control Mechanism, Interrupts and bottom halves, Writing interrupt driven drivers, Implementing bottom halves, Kernel Threads & Work Queues, Kernel timer, Jiffies, Timer interrupts

**Linux Control Groups and TCP/IP Networking:**
Resource limiting, Prioritization, Accounting and Control**,** Sockets APIs, Client and Server design, Remote Procedure Call

**Embedded Linux Specific:**
Boot sequence, I2C, SPI driver structure and application, Study of Simulated PCI driver, Linux Kernel Structure, BSP.

## References:

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, “Operating System Concepts”, Wiley, Ninth Edition
2. Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman, “Linux Device Drivers” , O’Reilly, Third Edition
3. Karim Yaghmour, “Building Embededd Linux Systems”, O'Reilly & Associates
4. Christopher Hallinan, “Embedded Linux Primer: A Practical Real World Approach”, Prentice Hall, 2nd Edition, 2010.

5. Derek Molloy, "Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux", Wiley, 1st Edition, 2014.

## (PCC) [EVE-19009] VLSI Design Verification and Testing

| Teaching Scheme | Examination Scheme |
|---|---|
| Lectures: 2 hrs/week | T1 - 20, Assignments – 30 , End-Sem |
| Tutorials: 1 hr/week | Exam – 50 marks |
| Practical: 2 hrs/week | Credits - 04 |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Learn verification techniques and create reusable verification environment.
2. Verify increasingly complex designs more efficiently.
3. Apply the concepts of testing to get a better yield in IC design
4. Use EDA tools like Cadence, Mentor Graphics.

### Syllabus Contents:

**Verification guidelines**:
Basic Testbench functionality, Directed testing, Verification Process, Testbench components, Constrained-Random stimulus, Functional coverage, Building layered testbench, Simulation environment phases, Maximum code reuse, Testbench performance.

**System Verilog Constructs:**
Data types: Built-in data types, Fixed-size arrays, Dynamic arrays, Queues, Associative arrays, Array methods, Choosing a storage type, Creating new types with typedef , Type conversion, Enumerated types, Constants strings, Expression width.

Procedural Blocks: Fork Join, Tasks and Functions, Inter Process Communication(Semaphore mailbox), Interface: ports, clocking blocks, virtual interface, Top-level Program – Module interactions.

Basic OOPs: Inheritance, Polymorphism

Randomization: Randomization and constrained Randomization pre_randomize and post_randomize functions, Random number generators

Functional coverage: Cover group, cover points, Bins

System Verilog Assertions: Properties, Boolean expressions

**Basics of Testing**:
Fault models, Combinational logic and fault simulation, Test generation for Combinational Circuits, test pattern generation methods and algorithm

**Design for Testability**:
Scan design, use of scan chains, boundary scan architecture, Built-in self-test

**Assignments:**

Apply the concepts of system Verilog constructs to verify DUTs of:

1. ALU
2. FIFO
3. Ethernet
4. AMBA protocol

**References:**

1. Chris Spears, “ System Verilog for Verification”, Springer, 2nd Edition
2. M. Bushnell and V. D. Agrawal, "Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits", Kluwer Academic Publishers
5. IEEE 1800-2009 standard (IEEE Standard for SystemVerilog— Unified Hardware Design, Specification and Verification Language).
6. System Verilog website – www.systemverilog.org
7. http://www.sunburst-design.com/papers/CummingsSNUG2006Boston_SystemVerilog_Events.pdf
8. General reuse information and resources www.design-reuse.com
9. OVM, UVM(on top of SV) www.verificationacademy.com
10. Verification IP resources http://www.cadence.com/products/fv/verification_ip/pages/default.aspx
11. http://www.synopsys.com/Tools/Verification/FunctionalVerification/VerificationIP/Pages/default.aspx

## (LC) [EVE-19010] CMOS VLSI Design Lab

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Practical: 2 hrs/week | Marks- 100 |
| | Credits- 01 |

### Laboratory Outcomes:

At the end of the laboratory work, students will demonstrate the ability to:

1. Understand Digital and Analog Circuit design using CMOS.
2. Build blocks of a system to solve engineering problems.
3. Use EDA tools like Cadence, Mentor Graphics and other open source software tools like NGSPICE through lab exercises.

### List of Experiments:

1. DC and Transient analysis of NMOS and PMOS Transistor using NGSPICE.
2. DC, Transient analysis of CMOS Inverter using NGSPICE.
3. Design of 5 stage ring oscillator using NGSPICE.
4. DC, Transient analysis of CMOS Inverter using Cadence EDA Tool.
5. Schematic to Symbol generation using Cadence EDA Tool.
6. Schematic to Layout of CMOS Inverter using Cadence EDA Tool.
7. Post Layout simulation of CMOS Inverter and Parasitic Extraction.
8. Design of all basic gates and /or combinatorial circuits using Cadence EDA Tool.
9. Design of a 6 Transistor SRAM cell using Cadence EDA Tool.
10. Design of Sequential circuits using Cadence EDA Tool.
11. AC analysis of NMOS and PMOS Transistor using Cadence EDA Tool.
12. AC analysis of Common Source Amplifier using Cadence EDA Tool.
13. AC analysis of Common Gate and Common Drain Amplifier using Cadence EDA Tool.
14. AC analysis of Current Source using Cadence EDA Tool.
15. DC and AC analysis of two stage Differential Amplifier using Cadence EDA Tool.
16. Noise and Frequency analysis of Amplifiers using Cadence EDA Tool.
17. Design of Band Gap reference circuits using Cadence EDA Tool.
18. Design of D to A converters using Cadence EDA Tool.
19. Design of A to D converters using Cadence EDA Tool.
20. Design of basic switched capacitor filters using Cadence EDA Tool.

## (LC) [EVE-19011] Linux in Embedded Systems Lab

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Practical: 2 hrs/week | Marks- 100 |
| | Credits- 01 |

### Laboratory Outcomes:

At the end of the laboratory work, students will demonstrate the ability to:

1. Develop multithreaded applications, libraries and device drivers for Linux OS
2. Configure, compile and load the embedded Linux kernel on to target platform.

### List of Experiments:

1. Linux commands and shell scripting
2. Applications with multiple threads, processes, synchronisation and inter-process communications mechanisms
3. Building, utilizing static and dynamic libraries in applications
4. Linux file systems and emulating several commands related to file systems such as ls, pwd
5. Develop and use pseudo and serial communication Linux device drivers
6. Design and implement custom network applications using socket programming
7. Compile and install bootloader and use basic commands of bootloader
8. Making a tiny embedded system with busy box
9. Configure and boot an embedded Linux relying on block storage

**Semester III and IV**

## (SBC) [EVE-20001 & EVE-20003] Dissertation Phase – I and II

| Teaching Scheme | Examination Scheme |
|---|---|
| -- | Marks: For both phase I and II |
| | Mid-sem: 30 Marks |
| | End-sem: 70 Marks |

### Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Conceive a problem statement either from rigorous literature survey or from the requirements raised by external entity.
2. Design, implement and test the prototype/algorithm in order to solve the conceived problem.
3. Publish the research work in journals/conferences of repute contributing to growth of technology in the domain.

### Guidelines:

As per the AICTE directives, the dissertation is a yearlong activity, to be carried out and evaluated in two phases i.e. Phase – I: July to December and Phase – II: January to June.

The dissertation may be carried out preferably in-house i.e. department's laboratories and centers OR in industry allotted through department's T & P coordinator.

After multiple interactions with guide and based on comprehensive literature survey, the student shall identify the domain and define dissertation objectives. The referred literature should preferably include IEEE/IET/IETE/Springer/Science Direct/ACM journals in the areas of Computing and Processing (Hardware and Software), Circuits-Devices and Systems, Communication-Networking and Security, Robotics and Control Systems, Signal Processing and Analysis, Machine Learning, IoT and any other related domain. In case of Industry sponsored projects, the relevant application notes, white papers, product catalogues should be referred and reported.

Student is expected to detail out specifications, methodology, resources required, critical issues involved in design and implementation and phase wise work distribution, and submit the proposal within a month from the date of registration.

**Phase – I deliverables**:

A document report comprising of summary of literature survey, detailed objectives, project specifications, paper and/or computer aided design, proof of concept/functionality, part results, A record of continuous progress.

**Phase – I evaluation**:

A committee comprising of guides of respective specialization shall assess the progress/performance of the student based on report, presentation and Q & A. In case of unsatisfactory performance, committee may recommend repeating the phase-I work.

During phase – II, student is expected to exert on design, development and testing of the proposed work as per the schedule. Accomplished results/contributions/innovations should be published in terms of research papers in reputed journals and reviewed focused conferences OR IP/Patents.

**Phase – II deliverables**:
A dissertation report as per the specified format, developed system in the form of hardware and/or software, A record of continuous progress.

**Phase – II evaluation**:
Guide along with appointed external examiner shall assess the progress/performance of the student based on report, presentation and Q & A. In case of unsatisfactory performance, committee may recommend for extension or repeating the work.

***