M. Tech. (VLSI & Embedded Systems Design)
PG Course Structure-R25

I-SEMESTER

S. No	Course Code	Course Title	Contact hours/week			Credits	Scheme of Valuation		
			L	T	P		CIE	SEE	Total
THEORY COURSES									
1	25VE01	Advanced Digital Systems Design	3	1	0	4	40	60	100
2	25VE02	Embedded Hardware Platforms and Programming	3	1	0	4	40	60	100
3	25VE03	FPGA Design	3	1	0	4	40	60	100
4	**PROGRAM ELECTIVE – I**		3	0	0	3	40	60	100
5	**PROGRAM ELECTIVE – II**		3	0	0	3	40	60	100
LABORATORY COURSES									
6	25VE51	Advanced Digital Systems Design Lab	0	1	2	2	40	60	100
7	25VE52	Embedded Systems Lab	0	1	2	2	40	60	100
8	25PI01	Seminar – I	0	0	2	1	100	--	100
Total			**15**	**5**	**6**	**23**	**380**	**420**	**800**

List of Professional Elective Courses in I Semester (Electives – I & II)

Course Code	Course Title
25VE04	Scripting Languages for VLSI
25VE05	VLSI Architectures
25VE06	VLSI System Design
25VE07	VLSI Testing & Testability
25VE08	System on Programming Chip Design
25VE09	Embedded System Design using FPGA
25VE10	ARM Microcontroller based Design
25VE11	Cryptography and Network Security

@ Minimum 2/3 themes per elective

II-SEMESTER

S. No	Course Code	Course Title	Contact hours/week			Credits	Scheme of Valuation		
			L	T	P		CIE	SEE	Total
THEORY COURSES									
1	25VE12	Digital CMOS Circuit Design	3	1	0	4	40	60	100
2	25VE13	System Design with Embedded Linux	3	1	0	4	40	60	100
3	25VE14	Embedded Real Time Operating Systems (ERTOS)	3	1	0	4	40	60	100
4	**PROGRAM ELECTIVE – III**		3	0	0	3	40	60	100
5	**PROGRAM ELECTIVE – IV**		3	0	0	3	40	60	100
LABORATORY COURSES									
6	25VE53	Digital CMOS Circuit Design Lab	0	1	2	2	40	60	100
7	25VE54	System Design with Embedded Linux Lab	0	1	2	2	40	60	100
8	25PI02	Seminar – II	0	0	2	1	100	--	100
Total			**15**	**5**	**6**	**23**	**380**	**420**	**800**

List of Professional Elective Courses in II Semester (Electives III & IV)

S.No.	Course Title
25VE15	VLSI Signal Processing
25VE16	Advanced VLSI Interconnects
25VE17	Quantum Computing
25VE18	Digital VLSI System Design
25VE19	System Design using Embedded Processors
25VE20	Architectures for DSP
25VE21	Internet of Things
25VE22	Embedded Network and Protocols

@ Minimum 2/3 themes per elective

III-SEMESTER

S. No	Course Code	Course Title	Contact hours/week			Credits	Scheme of Valuation		
			L	T	P		CIE	SEE	Total
THEORY COURSES									
1	25RM01	Research Methodology and IPR / Swayam 12-Weeks MOOC Course – RM & IPR	3	0	0	3	40	60	100
LABORATORY COURSES									
2	25PI03	Summer Internship/Industrial Training (8-10 Weeks)*	-	-	-	3	100	-	100
3	25PI04	Comprehensive Viva#	-	-	-	2	100	-	100
4	25PI05	Dissertation Part-A$	-	-	20	10	100	-	100
Total			**3**	**0**	**20**	**18**	**340**	**60**	**400**

*** Student attended during summer/year break and assessment will be done in 3rd Semester.**
 # Comprehensive Viva can be conducted courses completed up to 2nd Semester.
$ Dissertation Part-A: Internal Assessment.

IV-SEMESTER

S. No	Course Code	Course Title	Contact hours/week			Credits	Scheme of Valuation		
			L	T	P		CIE	SEE	Total
1	25PI06	Dissertation Part-B%	-	-	32	16	-	100	100
Total			**-**	**-**	**32**	**16**	**-**	**100**	**100**

% External Assessment

M. Tech. (VLSI& Embedded Systems Design)
I-Semester
Detailed Syllabus

25VE01 – Advanced Digital Systems Design

M. Tech. (I-Sem.)

L	T	P	Cr.
3	1	0	4

Pre-requisites: Switching Theory and Logic Design

Course Educational Objective: This course covers the architectures of FPGA and CPLD, advanced concepts on designing of Combinational and Sequential circuits with various fault modelling techniques.

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

CO1: Expose the design approaches using FPGAs. **(Understand-L2)**

CO2: Provide in depth understanding of Fault models. **(Understand-L2)**

CO3: Understands test pattern generation techniques for fault detection. **(Understand-L2)**

CO4: Design fault diagnosis in sequential circuits. **(Apply-L3)**

CO5: Provide understanding in the design of flow using case studies. **(Understand-L2)**

UNIT-I:

Programmable Logic Devices: The concept of programmable Logic Devices, SPLDs, PAL devices, PLA devices, GAL devices, CPLD-Architecture, FPGAs-FPGA technology, architecture, virtex CLB and slice, FPGA Programming Technologies, Xilinx XC2000, XC3000, XC4000 Architectures, Actel ACTI,ACT2 and ACT3 Architectures.

UNIT-II:

Analysis and derivation of clocked sequential circuits with state graphs and tables: A sequential parity checker, Analysis by signal tracing and timing charts-state tables and graphs-general models for sequential circuits, Design of a sequence detector, More Complex design problems, Guidelines for construction of state graphs, serial data conversion, Alphanumeric state graph notation. Need and Design strategies for multi-clock sequential circuits.

UNIT-III:

Sequential circuit Design: Design procedure for sequential circuits-design example, Code converter, Design of Iterative circuits, Design of a comparator, Controller (FSM) Metastability, Synchronization, FSM Issues, Pipelining resources sharing, Sequential circuit design using FPGAs, Simulation and testing of Sequential circuits, Overview of computer Aided Design.

UNIT-IV:

Fault Modeling and Test Pattern Generation: Logic Fault Model, Fault detection & redundancy, Fault equivalence and fault location, Fault dominance, Single stuck at fault model, multiple Stuck at Fault models, Bridging Fault model.

Fault diagnosis of combinational circuits by conventional methods, path sensitization techniques, Boolean difference method, KOHAVI algorithm, Test algorithms-D algorithm, Random testing transition count testing, signature analysis and test bridging faults.

UNIT-V:

Fault Diagnosis in sequential circuits: Circuit Test Approach, Transition check Approach, State identification and fault detection experiment, Machine identification, Design of fault detection experiment.

TEXT BOOKS:

1. Digital Electronics and design with VHDL- Volnei A. Pedroni, Elsevier Publications.
2. Fundamentals of Logic Design-Charles H. Roth, Jr. -5th Ed., Cengage Learning.
3. Digital Circuits and Logic Design-Samuel C.LEE, PHI, 2008.

REFERENCE BOOKS:

1. Logic Design Theory-N.N. Biswas, PHI.
2. Digital System Design using programmable logic devices- Parag K. Lala, BS publications.
3. Switching and Finite Automata Theory -Zvi Kohavi & Niraj K. Jha, 3rd Edition, Cambridge,2010

M. Tech. (I-Sem.)

25VE02 – Embedded Hardware Platforms and Programming

L	T	P	Cr.
3	1	0	4

Pre-requisites: Microprocessors and Microcontrollers

Course Educational Objective:

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

CO1: Identify the functioning of embedded systems for different applications. **(Understand-L2)**

CO2: Develop embedded system programming skills. **(Apply-L3)**

CO3: Design, implement and test an embedded system. **(Apply-L3)**

CO4: Identify the unique characteristics of real-time embedded systems. **(Understand-L2)**

UNIT-I:

Introduction to Embedded Computing: Embedded systems Overview, Characteristics of embedded computing applications, Design Challenges, Common Design Metrics, Processor Technology, IC Technology, Tradeoffs.

UNIT-II:

Process of Embedded System Development: The development process, Requirements, Specification, Architecture Design, Designing Hardware and Software components, system Integration and Testing.

UNIT-III:

Hardware platforms: Types of Hardware Platforms, Single board computers, PC Add-on cards, custom-built hardware platforms, ARM Processor, CPU performance, CPU power consumption, Bus-based computer systems, Memory devices, I/O devices, component interfacing, Designing with microprocessors, system level performance analysis.

UNIT-IV:

Program Design and Analysis: components for Embedded programs, Models of programs, Assembly, Linking, and loading, basic compilation techniques, software performance optimization, program level energy and Power analysis, Program validation and Testing.

UNIT-V:

Real-Time Operating Systems: Architecture of the kernel, Tasks and Task Scheduler, Scheduling algorithms, Interrupt Service Routines, Semaphores, Mutex, Mailboxes, Message queues, Event Registers, Pipes, Signals, Timers, Memory management, Priority Inversion problem. Overview of off-the-shelf operating systems - MicroC/OS II, Vxworks, RT Linux.

TEXT BOOKS:

1. Wayne Wolf: Computers as Components-Principles of Embedded Computer
2. System Design, Morgan Kaufmann Publisher-2006, 2ndEdition
3. David E-Simon: An Embedded software Primer, Pearson Education, 2007, 1st Edition
4. K. V. K. K. Prasad Real-Time Systems: Concepts Design and Programming, Dreamtech Press, 2005.

25VE03 – FPGA Design

M. Tech. (I-Sem.)

L	T	P	Cr.
3	1	0	4

Pre-requisites: Switching Theory and Logic Design

Course Educational Objective:

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

CO1: Understand FPGA design flow. **(Understand-L2)**

CO2: Identify the building blocks of commercially available FPGA/CPLDs. **(Understand-L2)**

CO3: Develop VHDL/Verilog models and synthesize targeting for Vertex, Spartan FPGAs. **(Apply-L3)**

CO4: Develop parameterized library cells and implement system designs using parameterized. **(Apply-L3)**

UNIT-I:

INTRODUCTION TO FPGAs: Evolution of programmable devices, FPGA Design flow, Applications of FPGA.

DESIGN EXAMPLES USING PLDs: Design of Universal block, Memory, Floating point multiplier, Barrel shifter.

UNIT-II:

FPGAs/CPLDs: Programming Technologies, Commercially available FPGAs, Xilinx's Vertex and Spartan, Actel's FPGA, Altera's FPGA/CPLD.

UNIT-III:

Building blocks of FPGAs/CPLDs: Configurable Logic block functionality, Routing structures, Input/output Block, Impact of logic block functionality on FPGA performance, Model for measuring delay.

UNIT-IV:

Routing Architectures: Routing terminology, general strategy for routing in FPGAs, routing for row – based FPGAs, introduction to segmented channel routing, routing for symmetrical FPGAs, example of routing in a symmetrical FPGA, general approach to routing in symmetrical FPGAs, independence from FPGA routing architectures, FPGA routing structures.

UNIT-V:

FPGA architectural assumptions, the logic block, the connection block, connection block topology, the switch block, switch block topology, architectural assumptions for the FPGA CASE STUDY – Applications using Kintex-7, Virtex-7, Artix-7.

TEXT BOOKS:

1. John V. Old Field, Richrad C. Dorf, Field Programmable Gate Arrays, Wiley, 2008.
2. Data sheets of Artix-7, Kintex-7, Virtex-7
3. Stephen D. Brown, Robert J. Francis, Jonathan Rose, Zvonko G. Vranesic, Field Programmable Gate Arrays, 2nd Edition, Springer, 1992.

25VE04 – Scripting Languages for VLSI

M. Tech. (I-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: C Language

Course Educational Objective:

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

CO1: Gain fluency in programming with scripting languages. **(Understand-L2)**

CO2: Create and run scripts using PERL/TCL/PYTHON in CAD Tools. **(Apply-L3)**

CO3: Demonstrate the use of PERL/PYTHON/ TCL in developing system and web applications. **(Understand-L2)**

CO4: Develop a real time project using PERL/PYTHON. **(Apply-L3)**

UNIT-I:

Introduction to Scripts and Scripting: Basics of Linux, Origin of Scripting languages, scripting today, Characteristics and uses of scripting languages.

UNIT-II:

PERL: Introduction to PERL, Names and values, Variables and assignment, Scalar expressions, Control structures, Built-in functions, Collections of Data, Working with arrays, Lists and hashes, Simple input and output, Strings, Patterns and regular expressions, Subroutines, Scripts with arguments.

UNIT-III:

Advanced PERL: Finer points of Looping, Subroutines, Using Pack and Unpack, Working with files, Type globs, Eval, References, Data structures, Packages, Libraries and modules, Objects and modules in action, Tied variables, interfacing to the operating systems, Security issues.

UNIT-IV:

TCL: The TCL phenomena, Philosophy, Structure, Syntax, Parser, Variables and data in TCL, Control flow, Data structures, Simple input/output, Procedures, Working with Strings, Patterns, Files and Pipes, Example code.

UNIT-V:

Advanced TCL: The eval, source, exec and up-level commands, Libraries and packages, Namespaces, Trapping errors, Event-driven programs, Making applications 'Internet-aware', 'Nuts-and-bolts' internet programming, Security issues, TCL and TK integration.

PYTHON: Introduction to PYTHON language, PYTHON-syntax, statements, functions, Built-in functions and Methods, Modules in PYTHON, Exception Handling.

TEXT BOOKS:

1. The World of Scripting Languages- David Barron, Wiley Student Edition, 2010.
2. PYTHON Web Programming, Steve Holden and David Beazley, New Riders Publications 12.

REFERENCE BOOKS:

1. TCL/TK: A Developer's Guide- ClifFlynt, 2003, Morgan Kaufmann Series.
2. Core PYTHON Programming, Chun, Pearson Education, 2006.
3. Learning Perl, Randal L. Schwartz, O' Reilly publications 6th Edition, 2011.
4. Linux: The Complete Reference", Richard Peterson McGraw Hill Publications, 6th Edition, 2008

25VE05 – VLSI Architectures

M. Tech. (I-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: Computer Architecture, Microprocessors and Microcontrollers

Course Educational Objective:

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

CO1: Design RISC architecture and control units for a given instruction set. **(Apply-L3)**

CO2: Improve the performance of RISC processors by applying pipelining techniques. **(Apply-L3)**

CO3: Translate DSP algorithms into efficient hardware architectures and design associated building blocks. **(Apply-L3)**

CO4: Analyze the impact of retiming, unfolding, and folding on the performance of DSP architectures. **(Analyze-L4)**

UNIT-I:

Instruction Set Architectures and CPU Performance: Overview of Instruction Set Architectures – CISC, RISC, and DSP Processors, CPU Performance and Its Factors, Evaluating Performance Metrics.

UNIT-II:

Design of RISC Processor: Designing the Datapath and Control Unit for a RISC Processor, Multicycle Implementation of RISC Architecture.

UNIT-III:

Enhancing Performance with Pipelining: Overview of Pipelining, Pipelined Datapath, Pipelined Control Unit, Pipeline Hazards – Data, Control, and Structural Hazards, Techniques for Hazard-Free Pipelined RISC Implementation.

UNIT-IV:

Multiprocessors and DSP Algorithm Representation: Introduction to Multiprocessors, Multiprocessors Connected by a Single Bus and Network, Network Topologies, Evolution vs. Revolution in Computer Architecture, DSP Algorithm Representation – Data Flow Graphs, Loop Bound and Iteration Bound, Algorithms for Computing Iteration Bound.

UNIT-V:

Pipelining, Parallel Processing, and VLSI Performance Techniques: Introduction to Pipelining and Parallel Processing, FIR Filter Pipelining, Parallel Processing Techniques, Pipelining and Parallel Processing for Low Power, VLSI Architecture Optimization Techniques – Retiming, Unfolding, and Folding.

TEXT BOOKS:

1. D. A, Patterson And J. L. Hennessy, Computer Organization and Design: Hardware / Software Interface, Elsevier, 2011, 4th Edition
2. Keshab Parhi, VLSI digital signal processing systems design and implementations, Wiley 1999.

25VE06 – VLSI System Design

M. Tech. (I-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: VLSI Design

Course Educational Objective:

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

CO1: Model the behaviour of a MOS Transistor. **(Apply-L3)**

CO2: Understanding CMOS Inverter. **(Understand-L2)**

CO3: Design combinational and sequential circuits using CMOS gates. **(Apply-L3)**

CO4: Identify the sources of power dissipation in a CMOS circuit. **(Understand-L2)**

CO5: Analyze SRAM cell and memory arrays. **(Analuze-L4)**

UNIT-I:

MOS Transistors, CMOS Logic, CMOS Fabrication and Layout, Design Partitioning, Fabrication, Packaging,and Testing, MOS transistor Theory, Long Channel I-V Characteristics, C-V Characteristics, Non-Ideal I-V Effects, DC Transfer Characteristics. The CMOS Inverter: The Static CMOS Inverter -An Intuitive Perspective, Evaluating the Robustness of the CMOS Inverter: The Static Behavior, Performance of CMOS Inverter: The Dynamic Behavior.

UNIT-II:

CMOS Processing Technology, CMOS Technologies, Layout Design Rules, CMOS Process Enhancements, Technology-Related CAD Issues, Manufacturing Issues, Circuit Simulation- A SPICE Tutorial, Device Models, Device Characterization, Circuit Characterization, Interconnect Simulation. Combinational Circuit Design, Circuit Families, Silicon-On-Insulator Circuit Design, Sub Threshold Circuit Design, Sequential Circuit Design, Circuit Design of Latches and Flip-Flops, Static Sequencing Element Methodology, Sequencing Dynamic Circuits, Synchronizers, Wave Pipelining.

UNIT-III:

Power, Sources of Power Dissipation, Dynamic Power, Static Power, Energy-Delay Optimization, Low Power Architectures, Robustness, Variability, Reliability, Scaling, Statistical Analysis of Variability, Variation Tolerant Design. Delay, Transient Response, RC Delay Model, Linear Delay Model, Logical Effort of Paths, Timing Analysis Delay Models, Datapath Subsystems, Addition/Subtraction, One/Zero Detectors, Comparators, Counters, Boolean Logical Operations, Coding, Shifters, Multiplication.

UNIT-IV:

Array Subsystems, SRAM, DRAM, Read-Only Memory, Serial Access Memories, Content-Addressable Memory, Programmable Logic Arrays, Robust Memory Design, Special-Purpose Subsystems.

UNIT-V:

CMOS Testing-The need for testing, Manufacturing test principles, Design strategies for test, Chip level test techniques, System level test techniques, Layout design for improved testability.

TEXT BOOKS:

1. Neil H.E. Weste, David Harris, Ayan Banerjee, CMOS VLSI Design – A Circuits and Systems Perspective, Pearson Education, 2006, 3rd Edition.
2. Neil H. E. Weste Kamran Eshraghian, Principles of CMOS VLSI DESIGN: A Systems Perspective, Pearson Education, 2006, 2nd Edition.

REFERENCE BOOKS:

1. Jan M RABAEY, Digital Integrated Circuits, Pearson Education, 2003, 2nd Edition.
2. Douglas A. Pucknell, Kamran Eshraghian, Basic VLSI Design, PHI, 1994, 3rd Edition.

25VE07 – VLSI Testing & Testability

M. Tech. (I-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: VLSI Design

Course Educational Objective:

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

CO1: Identify the significance of testable design. **(Understand-L2)**

CO2: Understand the concept of yield and identify the parameters influencing the same. **(Understand-L2)**

CO3: Specify fabrication defects, errors, and faults. **(Understand-L2)**

CO4: Implement combinational and sequential circuit test generation algorithms. **(Apply-L3)**

CO5: Identify techniques to improve fault coverage. **(Understand-L2)**

UNIT-I:

Role of Testing in VLSI Design Flow, Testing at Different Levels of Abstraction, Fault, Error, Defect, Diagnosis, Yield. Types of Testing, Rule of Ten, Defects in VLSI Chip. Modelling Basic Concepts, Functional Modelling at Logic Level and Register Level, Structure Models, Logic Simulation, Delay Models. Various Types of Faults, Fault Equivalence and Fault Dominance in Combinational and Sequential Circuits.

UNIT-II:

Fault Simulation Applications, General Fault Simulation Algorithms: Serial and Parallel, Deductive Fault Simulation Algorithms.

UNIT-III:

Combinational Circuit Test Generation, Structural Vs Functional Test, ATPG, Path Sensitization Methods. Difference Between Combinational and Sequential Circuit Testing, Five and Eight Valued Algebra, Scan Chain-Based Testing Method.

UNIT-IV:

D-Algorithm Procedure, Problems. PODEM Algorithm, Problems on PODEM Algorithm. FAN Algorithm, Problems on FAN Algorithm. Comparison of D, FAN and PODEM Algorithms. Design for Testability, Ad-Hoc Design, Generic Scan-Based Design.

UNIT-V:

Classical Scan-Based Design, System Level DFT Approaches. Test Pattern Generation for BIST, Circular BIST, BIST Architectures. Testable Memory Design: Test Algorithms, Test Generation for Embedded RAMs.

TEXT BOOKS:

1. M. Abramovici, M. Breuer, and A. Friedman, "Digital Systems Testing and Testable Design, IEEE Press, 1990.
2. M. Bushnell and V. Agrawal, "Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits", Kluwer Academic Publishers, 2000.

REFERENCE BOOKS:

1. Stroud, "A Designer's Guide to Built-in Self-Test", Kluwer Academic Publishers, 2002
2. V. Agrawal and S. C. Seth, Test Generation for VLSI Chips, Computer Society Press. 1989.

OTHER SUGGESTED READINGS:

1. NPTEL Courses (https://archive.nptel.ac.in/courses/117/105/117105137/)

25VE08 – System on Programming Chip Design

M. Tech. (I-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: VLSI Design

Course Educational Objective:

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

CO1: Understand the fundamental concepts and components of System-on-Chip (SoC) design, including design flow, hardware/software partitioning, and applications. **(Understand-L2)**

CO2: Analyze and compare processor architectures such as RISC, CISC, VLIW, Superscalar, and soft/firm/custom processors, with emphasis on instruction handling and memory integration. **(Analyze-L4)**

CO3: Evaluate various interconnection mechanisms like on-chip buses (AMBA, Core Connect, Wishbone, Avalon) and Network-on-Chip (NoC) architectures including topologies, routing algorithms, and QoS strategies. **(Apply-L3)**

CO4: Apply IP-based design methodologies in SoC development, including IP classification, reuse, lifecycle, integration, and implementation using FPGA prototypes. **(Apply-L3)**

CO5: Design and assess SoC implementations and testing techniques, including IP integration, RTOS, EDA tools, test automation strategies, and P1500 wrapper standardization. **(Apply-L3)**

UNIT-I:

Introduction: Driving Forces for SoC - Components of SoC - Design flow of SoC Hardware/Software nature of SoC - Design Trade-offs – SoC Applications. System-level Design: Processor selection-Concepts in Processor Architecture: Instruction set architecture (ISA), elements in Instruction Handing-Robust processors: Vector processor, VLIW, Superscalar, CISC, RISC—Processor evolution: Soft and Firm processors, Custom-Designed processors- on-chip memory.

UNIT-II:

Interconnection: On-chip Buses: basic architecture, topologies, arbitration and protocols, Bus standards: AMBA, Core-Connect, Wishbone, Avalon - Network-on-chip: Architecture- topologies-switching strategies – routing algorithms flow control, Quality-of-Service- Re- configurability in communication architectures.

UNIT-III:

IP based system design: Introduction to IP Based design, Types of IP, IP across design hierarchy, IP life cycle, Creating and using IP – Technical concerns on IP reuse – IP integration - IP evaluation on FPGA prototypes.

UNIT-IV:

SOC implementation: Study of processor IP, Memory IP, wrapper Design - Real-time operating system (RTOS), Peripheral interface and components, High-density FPGAs - EDA tools used for SOC design.

UNIT-V:

SOC testing: Manufacturing test of SoC: Core layer, system layer, application layer - P1500 Wrapper Standardization - SoC Test Automation (STAT).

TEXT BOOKS:

1. Michael J. Flynn, Wayne Luk,"Computer system Design: Systemon- Chip", Wiley- India, 2012.
2. Sudeep Pasricha, Nikil Dutt, "On Chip Communication Architectures: System on Chip Interconnect", Morgan Kaufmann Publishers, 2008.
3. 3. W. H. Wolf, "Computers as Components: Principles of Embedded Computing System Design", Elsevier, 2008.

REFERENCE BOOKS:

1. Patrick Schaumont "A Practical Introduction to Hardware/Software Co-design", 2nd Edition, Springer, 2012.
2. Lin, Y-L.S. (ed.), "Essential issues in SOC design: designing complex systems-onchip. Springer, 2006.
3. Wayne Wolf, "Modern VLSI Design: IP Based Design", Prentice-Hall India, Fourth Edition, 2009.

25VE09 – Embedded System Design using FPGA

M. Tech. (I-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: VLSI Design

Course Educational Objective:

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

CO1: Explain the architecture of embedded systems and identify the role of FPGAs and SoCs in modern VLSI-based platforms. **(Understand-L2)**

CO2: Develop and simulate digital circuits using VHDL/Verilog and design high quality modular systems based on control flow graphs and abstraction principles. **(Apply-L3)**

CO3: Demonstrate the ability to select and integrate system software, cross development tools, boot-loaders, and monitors in FPGA-based embedded platforms. **(Understand-L2)**

CO4: Analyze partitioning strategies and communication mechanisms to optimize performance, resource usage, and system scalability. **(Analyze-L4)**

CO5: Apply principles of spatial parallelism and identify contemporary design issues to build efficient, high-performance FPGA-based solutions. **(Apply-L3)**

UNIT-I:

Introduction to Embedded Systems and FPGA Platforms: Embedded System Overview: H/W-FPGA-Embedded SoC, Use of VLSI circuit technology, Platform FPGAs – Altera Cyclone, FPGA Platform, Components of platform FPGA systems, Adding custom compute cores, assembling platform-based systems.

UNIT-II:

Hardware Description and System Design: Hardware Description Languages: VHDL, Verilog, Other High-Level HDLs, HDL to Configuration Bit-stream generation.

System Design using FPGA: Principles of system design, Design quality, Modules and interfaces, Abstraction and state, Cohesion and coupling, Design reuse strategies, Control flow graph, Origins of platform FPGA designs.

UNIT-III:

Software Design for FPGA Systems: Software Design Considerations: System Software Options, Root File System, Cross-Development Tools for Embedded Applications.

Monitors and Boot-loaders: Role in platform-based development, Integration techniques.

UNIT-IV:

Partitioning and Communication: Partitioning Overview: Partitioning Problem, Basic definitions, Expected performance gain, Resource considerations in partitioning, Analytical Approach to Partitioning.

Scheduling and Communication: Invocation and coordination mechanisms, Transfer of state, Practical Issues in Profiling, Data structure design, Feature size manipulation.

UNIT-V:

Parallelism and Contemporary Issues: Spatial Design Concepts: Principles of parallelism, Identifying parallelism in applications.

Spatial Parallelism with Platform FPGAs: Within FPGA hardware cores, Across FPGA designs.

Contemporary Issues in Embedded FPGA System Design: Trends, challenges, and emerging technologies.

TEXT BOOKS:

1. Ron Sass, Andrew G. Schmidt, Embedded Systems Design with Platform FPGAs: Principles and Practices, First Edition, Tata McGraw Hill, India, 2011.

REFERENCE BOOKS:

1. Charles H. Roth Jr., Digital Systems Design Using VHDL, Reprint Edition, PWS Publishing Company (Thomson Books), USA, 2012.
2. V. A. Padroni, Circuit Design with VHDL, First Edition, MIT Press, Cambridge, England, 2011.
3. Wayne Wolf, FPGA Based System Design, First Edition, Prentice Hall, Modern Semiconductor Design Series, USA, 2011.

25VE10 – ARM Microcontroller based Design

M. Tech. (I-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: Microprocessors and Microcontrollers

Course Educational Objective:

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

CO1: Explore the selection criteria of ARM processors by understanding the functional level tradeoff issues. **(Understand-L2)**

CO2: Implementations on ARM developments towards the functional capabilities. **(Apply-L3)**

CO3: Work with ASM level program using the instruction set. **(Apply-L3)**

CO4: Programming the ARM Cortex M. **(Apply-L3)**

CO5: Discuss about Floating Point Operations. **(Understand-L2)**

UNIT-I:

ARM Embedded Systems: RISC design philosophy, ARM design philosophy, Embedded system hardware, Embedded system software.
ARM Processor Fundamentals: Registers, CPSR, Pipeline, Exceptions, Interrupts and Vector Table, Core Extensions, Architecture Revisions, ARM Processor Families.
Architecture of ARM Processors: Programmer's model, modes and states, special and floating-point registers, APSR, Memory system, MPU, Exceptions, NVIC, vector table, Fault handling, SCB, Debug, Reset sequence.

UNIT-II:

ARM Instruction Set: Data processing, branch, load-store, software interrupt, and program status register instructions, loading constants, ARMv5E extensions, Conditional execution.
Thumb Instruction Set: Thumb Register Usage, ARM-Thumb Interworking, Branch, Data Processing, Load-Store, Stack, and Software Interrupt Instructions.

UNIT-III:

Technical Details of Cortex M Processors: Overview of Cortex-M3 and M4: architecture, instruction set, block diagram, memory system, exception and interrupt support.
Features: Performance, code density, low power, MPU, OS support, Cortex-M4-specific DSP features, Debug support, Scalability, Compatibility.

UNIT-IV:

Instruction Set of Cortex M: Instruction set background, comparison across Cortex-M processors, UAL syntax, instruction suffixes, Cortex-M4-specific instructions, Barrel shifter, Special instructions and register access.

UNIT-V:

Floating Point Operations: Floating point data and FPU overview (CPACR, FP registers, FPSCR, FPCCR, FPCAR, FPDSCR, MVFR0, MVFR1).
DSP Applications: Dot product, Biquad filter, FIR, FFT and optimized DSP code writing for Cortex-M4.

TEXT BOOKS:

1. Andrew N. SLOSS, Dominic SYMES, Chris WRIGHT-ARM System Developer's Guide Designing and Optimizing System Software, Elsevier Publications, 2004.
2. Joseph Yiu, The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors by Elsevier Publications, 3rd Edition.

REFERENCE BOOKS:

1. Steve Furber-Arm System on Chip Architectures–Edison Wesley, 2000.
2. David Seal-ARM Architecture Reference Manual, Edison Wesley, 2000.

25VE11 – Cryptography and Network Security

M. Tech. (I-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: Computer Networks

Course Educational Objective:

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

CO1: Identify and utilize different forms of cryptography techniques. **(Understand-L2)**

CO2: Incorporate authentication and security in the network applications. **(Apply-L3)**

CO3: Distinguish among different types of threats to the system and handle the same. **(Understand-L2)**

CO4: Analyze Public-Key (Asymmetric) Cryptography and message digest algorithms. **(Analyze-L4)**

CO5: Discuss about Authentication and System Security. **(Understand-L2)**

UNIT-I:

Security: Need, security services, Attacks, OSI Security Architecture, one time passwords, Model for Network security, Classical Encryption Techniques like substitution ciphers, Transposition ciphers, Cryptanalysis of Classical Encryption Techniques.

UNIT-II:

Number Theory: Introduction, Fermat's and Euler's Theorem, The Chinese Remainder Theorem, Euclidean Algorithm, Extended Euclidean Algorithm, and Modular Arithmetic.

UNIT-III:

Private-Key (Symmetric) Cryptography: Block Ciphers, Stream Ciphers, RC4 Stream cipher, Data Encryption Standard (DES), Advanced Encryption Standard (AES), Triple DES, RC5, IDEA, Linear and Differential Cryptanalysis.

UNIT-IV:

Public-Key (Asymmetric) Cryptography: RSA, Key Distribution and Management, Diffie-Hellman Key Exchange, Elliptic Curve Cryptography, Message Authentication Code, hash functions, message digest algorithms: MD4, MD5, Secure Hash algorithm, RIPEMD-160, HMAC.

UNIT-V:

Authentication and System Security: IP and Web Security, Digital Signatures, Digital Signature Standards, Authentication Protocols, Kerberos, IP security Architecture, Encapsulating Security Payload, Key Management, Web Security Considerations, Secure Socket Layer, Secure Electronic Transaction, Intruders, Intrusion Detection, Password Management, Worms, viruses, Trojans, Virus Countermeasures, Firewalls, Trusted Systems.

TEXT BOOKS:

1. William Stallings, "Cryptography and Network Security, Principles and Practices", Pearson Education, 3rd Edition.
2. Charlie Kaufman, Radia Perlman and Mike Speciner, "Network Security, Private Communication in a Public World", Prentice Hall, 2ndEdition.

REFERENCE BOOKS:

1. Christopher M. King, Ertem Osmanoglu, Curtis Dalton, "Security Architecture, Design Deployment and Operations", RSA Press,
2. Stephen Northcutt, Leny Zeltser, Scott Winters, Karen Kent, and Ronald W. Ritchey, "Inside Network Perimeter Security", Pearson Education, 2ndEdition
3. Richard Bejtlich, "The Practice of Network Security Monitoring: Understanding Incident Detection and Response", William Pollock Publisher, 2013.

25VE51 – Advanced Digital Systems Design Lab

M. Tech. (I-Sem.)

L	T	P	Cr.
0	1	2	2

Pre-requisites: VLSI Design

Course Educational Objective:

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

CO1: Design and simulate basic memory systems such as RAM and ROM using HDL. **(Apply-L3)**

CO2: Design and implement control units and data path logic for processor-like architectures. **(Apply-L3)**

CO3: Apply coding techniques like Hamming Code for error detection and correction. **(Apply-L3)**

CO4: Design and implement sequential digital systems using Finite State Machines (Mealy and Moore models). **(Apply-L3)**

CO5: Develop and simulate real-time digital systems including UART communication, PWM generation, and digital clocks. **(Apply-L3)**

CO6: Design and simulate application-oriented digital systems like vending machines, home alarm systems, and traffic controllers. **(Apply-L3)**

Programming can be done using either VHDL /verilog HDL. Download the programs on FPGA boards and performance testing may be done using pattern generator (32 channels) and logic analyzer apart from verification by simulation with any of the front-end tools and implement all the Designs in FPGA Kits.

List of Experiments:

1. Design of Memory (RAM and ROM).
2. Design of Control Unit and Data Processor Logic Design
3. Design and implementation of Hamming Code.
4. Design of Sequence Detector (Finite State Machine- Mealy and Moore Machines).
5. Design of DNA Sequence Detector
6. Design of Pulse Width Modulation
7. Design of UART Transmitter and Receiver Module
8. Design of Seven Segment Display
9. Design of Traffic Light Controller
10. Design and simulation of Home Alarm System.
11. Design and simulation of Digital Clock.
12. Design and simulation of Vending Machine.

Lab Requirements:

Software required: Xilinx Vivado Tool.

Hardware required: Personal Computer, FPGA Development Board.

25VE52 – Embedded Systems Lab

M. Tech. (I-Sem.)

L	T	P	Cr.
0	1	2	2

Pre-requisites: C Language, Microprocessors and Microcontrollers

Course Educational Objective:

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

CO1: Demonstrate the ability to write and execute basic Embedded C programs on microcontroller platforms. **(Understand-L2)**

CO2: Apply digital I/O interfacing techniques by programming ports to control and monitor external hardware. **(Apply-L3)**

CO3: Implement timing-based operations using software and hardware delays, including loops and timers. **(Apply-L3)**

CO4: Design embedded applications for real-time control scenarios such as traffic lights and alarms. **(Apply-L3)**

CO5: Interface serial communication peripherals and measure real-time data over communication links. **(Apply-L3)**

CO6: Develop embedded software solutions for domain-specific applications such as industrial automation. **(Apply-L3)**

CO7: Demonstrate the use of port headers and external devices (like LCDs and keypads) in an embedded system. **(Understand-L2)**

List of Experiments:

1. Write a simple program to print " Hello World"
2. Write a simple program to show a delay
3. Write a loop application to copy values from P1 to P2.
4. Write a C program for counting the no of times that a switch is pressed & released.
5. Write a simple program to create a portable hardware delay.
6. Write a C program to test loop time outs.
7. Write a C program to test hardware based timeouts loops.
8. Illustrate the use of port header file (PORT M) using an interface consisting of a keyword and Liquid crystal display.
9. Develop a simple EOS showing traffic light sequencing.
10. Write a program to display elapsed time over RS-232 Link.
11. Write a program to drive SEOS Using Timer 0.
12. Develop software for milk pasteurization system.
13. Develop & implement a program for intruder alarm system.

M. Tech. (VLSI& Embedded Systems Design)
II-Semester
Detailed Syllabus

25VE12 – Digital CMOS Circuit Design

M. Tech. (II-Sem.)

L	T	P	Cr.
3	1	0	4

Pre-requisites: VLSI Design

Course Educational Objective:

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

CO1: Analyze MOSFET behavior and CMOS inverter characteristics under static and dynamic conditions. **(Analyze-L4)**

CO2: Design various combinational and sequential logic blocks using CMOS technology. **(Apply-L3)**

CO3: Optimize data path elements such as adders, multipliers, and barrel shifters. **(Analyze-L4)**

CO4: Design and evaluate memory architectures including SRAM and ROM cells. **(Apply-L3)**

CO5: Interpret and implement circuit layouts using stick diagrams and layout rules. **(Apply-L3)**

UNIT-I:

MOS Transistor Principles and CMOS Inverter: MOSFET characteristics under Static and Dynamic Conditions, MOS Transistor Secondary Effects, CMOS Inverter – Static Characteristic, Dynamic Characteristic, Power, Energy, and Energy Delay Parameters, Stick Diagram and Layout Diagrams.

UNIT-II:

Combinational Logic Circuits: Static CMOS Design, Different Styles of Logic Circuits, Logical Effort of Complex Gates, Static and Dynamic Properties of Complex Gates, Interconnect Delay, Dynamic Logic Gates.

UNIT-III:

Sequential Logic Circuits: Static Latches and Registers, Dynamic Latches and Registers, Timing Issues, Pipelines, Non-Bistable Sequential Circuits.

UNIT-IV:

Arithmetic Building Blocks: Data Path Circuits, Architectures for Adders, Accumulators, Multipliers, Barrel Shifters, Speed and Area Tradeoffs..

UNIT-V:

Memory Architectures: Memory Architectures and Memory Control Circuits: Read-Only Memories, ROM Cells, Read-Write Memories (RAM), Dynamic Memory Design, 6-Transistor SRAM Cell, Sense Amplifiers.

TEXT BOOKS:

1. Jan Rabaey, Anantha Chandrakasan, B Nikolic,"Digital Integrated Circuits: A Design Perspective", Prentice Hall of India, 2nd Edition, Feb 2003
2. N. Weste, K. Eshraghian,"Principles of CMOS VLSI Design", Addision Wesley, 2nd Edition, 1993.

REFERENCE BOOKS:

1. M J Smith, “Application Specific Integrated Circuits”, Addisson Wesley, 1997
2. Sung-Mo Kang &Yusuf Leblebici, “CMOS Digital Integrated Circuits Analysis and Design”, McGraw-Hill, 1998.

25VE13 – System Design with Embedded Linux

M. Tech. (II-Sem.)

L	T	P	Cr.
3	1	0	4

Pre-requisites: Operating System, Embedded System

Course Educational Objective:

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

CO1: Execute Linux and File I/O Commands. **(Apply-L3)**

CO2: Analyze Kernel Architecture and Scheduler Features. . **(Analyze-L4)**

CO3: Develop Device Drivers for various peripherals. . **(Apply-L3)**

CO4: Explore Linux Root File System and concepts of Embedded Linux. . **(Apply-L3)**

CO5: Analyze RT Linux Basics and OS Safety. **(Analyze-L4)**

UNIT-I:

Overview of LINUX: Introduction to UNIX/LINUX, LINUX Commands, File I/O (open, create, close, lseek, read, write), Process Control (fork, vfork, exit, wait, waitpid, exec), Embedded LINUX Vs Desktop LINUX, Embedded LINUX Distributions.

UNIT-II:

Linux Kernel: Embedded Linux Architecture, Kernel Architecture, Hardware Abstraction Layer, Memory Manager, Scheduler, File System, I/O and Networking Subsystem, Inter Process Communication, User Space, and Start-up Sequence.

UNIT-III:

Embedded Drivers: Board Support Package: Embedded Storage, Memory Technology Devices (MTD), Embedded Drivers: Serial, I2C, USB, Ethernet, Timer, Kernel Modules, and Embedded File System..

UNIT-IV:

Building and Debugging: Kernel, Root File System, Case Studies: RTL LINUX, Micro C/OS-II, VxWorks, Embedded Linux, and Tiny OS.

UNIT-V:

Linux Tasks: Porting Applications, Real-Time Linux Basics, Kernel Priority, Task Creation, Print Commands, Compilation, Safety-Critical Features, Components, Programs.

TEXT BOOKS:

1. Chris Simmonds, "Mastering Embedded Linux Programming" - Second Edition, PACKT Publications Limited.
2. Karim Yaghmour, "Building Embedded Linux Systems", O'Reilly & Associates
3. P Raghvan, Amol Lad, Sriram Neelakandan, "Embedded Linux System Design and Development", Auerbach Publications.

REFERENCE BOOKS:

1. Christopher Hallinan, "Embedded Linux Primer: A Practical Real-World Approach", Prentice Hall, 2nd Edition, 2010.
2. Derek Molloy, "Exploring Beagle Bone: Tools and Techniques for Building with Embedded Linux", Wiley, 1st Edition, 2014

M. Tech. (II-Sem.)

25VE14 – Embedded Real Time Operating Systems (ERTOS)

L	T	P	Cr.
3	1	0	4

Pre-requisites: Operating Systems

Course Educational Objective:

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

CO1: Illustrate real time programming concepts. **(Understand-L2)**

CO2: Apply RTOS functions to implement embedded applications. **(Apply-L3)**

CO3: Understand fundamentals of design consideration for embedded applications. **(Understand-L2)**

CO4: Describe about the memory units and real time memory applications. **(Understand-L2)**

CO5: Discuss communication Common Design Problems. **(Understand-L2)**

UNIT-I:

Introduction to Real-Time Operating Systems: Defining an RTOS, The scheduler, Kernel Objects and services, Key characteristics of an RTOS. Task: Defining a Task, Task States and Scheduling, Typical Task Operations, Typical Task Structure, Synchronization, Communication and Concurrency.

UNIT-II:

Semaphores: Defining Semaphores, Typical Semaphore Operations, Typical Semaphore Use. Message Queues: Defining Message Queues, Message Queue States, Message Queue Content, Message Queue Storage, Typical Message Queue Operations, Typical Message Queue Use. Pipes, Event Registers, Signals and Condition Variables.

UNIT-III:

Exceptions and Interrupts: Exceptions and Interrupts, Applications of Exceptions and Interrupts, Closer look at exceptions and interrupts, Processing General Exceptions, Nature of Spurious Interrupts.
Timer and Timer Services: Real-Time Clocks and System Clocks, Programmable Interval Timers, Timer Interrupt Service Routines.
Exceptions and Interrupts: Exceptions and Interrupts, Applications of Exceptions and Interrupts, Closer look at exceptions and interrupts, Processing General Exceptions, Nature of Spurious Interrupts.
Timer and Timer Services: Real-Time Clocks and System Clocks, Programmable Interval Timers, Timer Interrupt Service Routines.

UNIT-IV:

Memory Management: Dynamic Memory Allocation in Embedded Systems, Fixed-Size Memory Management in Embedded Systems, Blocking vs. Non- Blocking Memory Functions, Hardware Memory Management Units. Modularizing an Application for Concurrency: An Outside-In Approach to Decompose Applications, Guidelines and Recommendations for Identifying Concurrency, Schedulability Analysis.

UNIT-V:

Synchronization and Communication: Synchronization, Communication, Resource Synchronization Methods, Critical Section, Common Practical Design Patterns, Specific Solution Design Patterns.
Common Design Problems: Resource Classification, Deadlocks, Priority Inversion.

TEXT BOOKS:

1. Qing Li, Caroline Yao (2003), "Real-Time Concepts for Embedded Systems", CMP Books.

REFERENCE BOOKS:

1. Albert Cheng, (2002), "Real-Time Systems: Scheduling, Analysis and Verification", Wiley Interscience.
2. Hermann Kopetz, (1997), "Real-Time Systems: Design Principles for Distributed Embedded Applications", Kluwer.
3. Insup Lee, Joseph Leung, and Sang Son, (2008) "Handbook of Real-Time Systems", Chapman and Hall.Krishna and Kang G Shin, (2001), "Real-Time Systems", McGraw Hill.

25VE15 – VLSI Signal Processing

M. Tech. (II-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: Digital Signal Processing, VLSI Design

Course Educational Objective:

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

CO1: Understand the fundamentals of DSP systems, data flow modeling, and techniques like pipelining and parallel processing for FIR filters. **(Understand-L2)**

CO2: Apply retiming, unfolding, and algorithmic strength reduction techniques to optimize DSP architectures. **(Apply-L3)**

CO3: Analyze and implement pipelined and parallel processing architectures for IIR filters and fast convolution methods. **(Analyze-L4)**

CO4: Design and evaluate bit-level arithmetic structures such as multipliers, FIR filters, and distributed arithmetic implementations. **(Apply-L3)**

CO5: Explore synchronous, wave, and asynchronous pipelining techniques and apply numerical strength reduction methods in DSP systems. **(Apply-L3)**

UNIT-I:

Introduction to DSP: Typical DSP Algorithms, Benefits of DSP Algorithms, Representation of DSP Algorithms. Pipelining and Parallel Processing: Introduction, Pipelining of FIR Digital Filters, Parallel Processing, Pipelining and Parallel Processing for Low Power.

Retiming: Introduction, Definitions and Properties, Solving System of Inequalities, Retiming Techniques.

UNIT-II:

Folding: Introduction, Folding Transform, Register Minimization Techniques, Register Minimization in Folded Architectures, Folding of Multirate Systems. Unfolding: Introduction, Algorithm for Unfolding. Properties of Unfolding, Critical Path, Unfolding and Retiming, Applications of Unfolding

UNIT-III:

Systolic Architecture Design: Introduction, Systolic Array Design Methodology, FIR Systolic Arrays, Selection of Scheduling Vector, Matrix Multiplication and 2D Systolic Array Design, Systolic Design for Space Representations Containing Delays.

UNIT-IV:

Fast Convolution: Introduction, Cook-Toom Algorithm, Winograd Algorithm, Iterated Convolution, Cyclic Convolution, Design of Fast Convolution Algorithm by Inspection.

UNIT-V:

Low Power Design: Scaling Vs Power Consumption, Power Analysis, Power Reduction Techniques, Power Estimation Approaches. Programmable DSP: Evaluation of Programmable DSPs, DSP Processors for Mobile and Wireless Communications, Processors for Multimedia Signal Processing.

TEXT BOOKS:

1. VLSI Digital Signal Processing- System Design and Implementation – Keshab K. Parhi, 1998, Wiley Inter Science.
2. VLSI and Modern Signal Processing – Kung S. Y, H. J. While House, T. Kailath, 1985, Prentice Hall.

REFERENCE BOOKS:

1. Design of Analog – Digital VLSI Circuits for Telecommunications and Signal Processing – Jose E. France, YannisTsividis, 1994, Prentice Hall.
2. VLSI Digital Signal Processing – Medisetti V. K, 1995, IEEE Press (NY), USA.

25VE16 – Advanced VLSI Interconnects

M. Tech. (II-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: VLSI Design

Course Educational Objective:

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

CO1: Gain insight into transmission line parameters of VLSI interconnects. **(Understand-L2)**

CO2: Understand novel and emerging solutions for future VLSI interconnect technologies. **(Understand-L2)**

CO3: Analyze the impact of inductive effects in high-speed interconnects. **(Analyze-L4)**

CO4: Examine the influence of quantum effects in nanoscale interconnects. **(Analyze-L4)**

UNIT-I:

Introduction: Introduction to VLSI Interconnects, The Distributed RC Interconnect Model, Elmore Delay in Interconnects, Scaling Effects in Interconnects, Simulation and Delay Mitigation in RC Interconnects.

UNIT-II:

Inductive Effects: Inductive Effects in Interconnects, Distributed RLC Interconnect Model, Transmission Line Equations, When to Consider the Inductive Effects? Equivalent Elmore Model for RLC Interconnects, Two- Pole Model of RLC Interconnects from ABCD Parameters, RLC Interconnect Simulation.

UNIT-III:

Skin Effect and Electromigration: Origin of the Skin Effect, Effective Resistance at High Frequencies, Power Dissipation due to Interconnects, Electromigration in Interconnects, Mitigation of Electromigration.

UNIT-IV:

Crosstalk: Capacitive Coupling in Interconnects, Crosstalk Effects in Two Identical Interconnects, Mitigation Techniques, Analysis and Simulation of Coupled Interconnects. Extraction of Capacitance, Extraction of Inductance, Estimation of Interconnect Parameters from S-parameters.

UNIT-V:

Quantum Effects: Quantum Conductance, Quantum Capacitance, Kinetic Inductance, Graphene Nanoribbon Interconnects, Analysis and Simulation of Interconnect Considering Quantum Effects.

TEXT BOOKS:

1. Ashok K. Goel, High-SpeedVLSIInterconnects,2007.
2. Y. S. Diamand, Advanced Nanoscale ULSI Interconnects: Fundamentals and Applications, 2009.

REFERENCE BOOKS:

1. H. S Philip Wong and Deji Akinwande, Carbon nanotube and Graphene Device Physics, 2011.

Other Suggested Readings:

1. NPTELCourses(https://onlinecourses.nptel.ac.in/noc22_ee125/preview

25VE17 – Quantum Computing

M. Tech. (II-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: Quantum Mechanics

Course Educational Objective:

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

CO1: Understand the fundamental principles of quantum computation and the concept of qubits. **(Understand-L2)**

CO2: Analyze multi-qubit systems and quantum communication protocols. **(Analyze-L4)**

CO3: Analyze multi-qubit systems and quantum communication protocols. **(Analyze-L4)**

CO4: Design and implement basic quantum algorithms and quantum circuits. **(Applye-L3)**

UNIT-I:

Review of Quantum Mechanics and Motivation for Quantum Computation. Qubit: The Qubit State - Matrix and Bloch Sphere Representation - Computational Basis - Unitary Evolution.

UNIT-II:

Multi-Qubit States: No-Cloning Theorem, Superdense Coding, Pure States to Bell States, Bell Inequalities. Protocols with Multi-Qubits: Swapping, Teleportation. Gates: CNOT, Toffoli Gate, NAND, FANOUT, Walsh-Hadamard.

UNIT-III:

Measurement: Projective Operators - General, Projective and POVM Measurement. Ensemble: Density Operators - Pure and Mixed Ensemble - Time Evolution - Post Measurement Density Operator. **Composite Systems:** Partial Trace, Reduced Density Operator, Schmidt Decomposition, Purification, Bipartite Entanglement.

UNIT-IV:

Quantum Computing: Classical Computing Using Qubits, Quantum Parallelism, Deutsch's Algorithm, Deutsch-Jozsa Algorithm.

UNIT-V:

Quantum Circuits: Basic Gates, ABC Decomposition, Gray Codes, Universal Gates, Principle of Deferred and Implicit Measurements.

Quantum Fourier Transform and Applications: Phase Estimation, Order Finding, Factoring, Discrete Logarithm, Hidden Subgroup Problems. Role of Prime Factoring in Classical Cryptography. Search Algorithms, Quantum Error Correcting Codes, Physical Realization of Qubits.

TEXT BOOKS:

1. M.A. Nielsen and I.L Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2010, 10th Anniversary Edition
2. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press, 2019.
3. Ray La Pierre, Introduction to Quantum Computing, Springer, 2021.

REFERENCE BOOKS:

1. Quantum Theory: Concepts and Methods, Asher Peres, Kluwer Academic Publishers, 1993.
2. Venkateswaran Kasirajan, Fundamentals of Quantum Computing: Theory and Practice, Springer, 2021.

Other Suggested Readings:

1. NPTEL Courses(https://nptel.ac.in/courses/106106232

25VE18 – Digital VLSI System Design

M. Tech. (II-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-Requisites: STLD

COURSE EDUCATIONAL OBJECTIVES:
In this course student will learn about the design of various combinational and sequential circuits using verilog HDL and VHDL, write the verilog tasks, functions, and various digital modules.

COURSE OUTCOMES: At the end of this course student will be able to
CO1: Design Combinational and Sequential circuits. **(Apply-L3)**
CO2: Understand Digital System Design flow using Verilog HDL. **(Understand-L2)**
CO3: Model Digital System Using Verilog HDL. **(Apply-L3)**
CO4: Write Verilog Tasks, Functions, UDPs for Digital modules. **(Apply-L3)**

UNIT-I: REVIEW OF LOGIC DESIGN FUNDAMENTALS
Combinational Logic. Boolean Algebra and Algebraic Simplification. Karnaugh Maps. Designing with NAND and NOR Gates. Hazards in Combinational Circuits. Flip-Flops and Latches. Mealy Sequential Circuit Design. Design of a Moore Sequential Circuit. Equivalent States and Reduction of State Tables. Sequential Circuit Timing. Tristate Logic and Busses.

UNIT-II: INTRODUCTION TO VERILOG
Computer-Aided Design. Hardware Description Languages. Verilog Description of Combinational Circuits. Verilog Modules. Verilog Assignments. Procedural Assignments. Modeling Flip-Flops Using Always Block. Always Blocks Using Event Control Statements. Delays in Verilog. Compilation, Simulation, and Synthesis of Verilog Code. Verilog Data Types and Operators. Simple Synthesis Examples. Verilog Models for Multiplexers. Modeling Registers and Counters Using Verilog Always Statements. Behavioral and Structural Verilog. Constants. Arrays. Loops in Verilog. Testing Verilog Model. A Few Things to Remember.

UNIT-III: DESIGN EXAMPLES
BCD to 7-Segment Display Decoder. A BCD Adder.32-Bit Adders. Traffic Light Controller. State Graphs for Control Circuits. Scoreboard and Controller. Synchronization and Debouncing. A Shift-and-Add Multiplier. Array Multiplier. A Signed Integer/Fraction Multiplier. Keypad Scanner. Binary Dividers.

UNIT-IV: SM CHARTS AND MIRCOPROGRAMMING
State Machine Charts. Derivation of SM Charts. Realization of SM Charts. Implementation of the Dice Game. Microprogramming. Linked State Machines.

UNIT-V: ADDITIONAL TOPICS IN VERILOG & DESIGN OF A RISC MICROPROCESSOR
Verilog Functions. Verilog Tasks. Multi-Valued Logic and Signal Resolution. Built-in Primitives. User Defined Primitives. SRAM Model. Model for SRAM Read/Write System. Rise and Fall Delays of Gates. Named Association. Generate Statements. System Functions. Compiler Directives. File I/O Functions. Timing Check. The RISC Philosophy. The MIPS ISA. MIPS Instruction Encoding. Implementation of a MIPS Subset. Verilog Model.

TEXT BOOK:
1. Digital Systems Design using VERILOG Charles H. Roth, Lizykurian John.

REFERENCE:

1. Principles of CMOS VLSI Design - N. H. E. Weste, K. Eshraghian, 2nd Ed., Adisson Wesley.
2. CMOS VLSI Design, A Circuits and Systems Perspective, Neil H. E. Weste, Third Edition
3. Introduction to VLSI Systems - A Logic Circuit and System Perspective - Ming Bo, Liu, CRC Press, 1st Edition, 2011.

25VE19 – System Design using Embedded Processors

M. Tech. (II-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: ARM Controllers

Course Educational Objective:

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

CO1: Understand the fundamental concepts, architecture, and application areas of embedded systems along with development tools. **(Understand-L2)**

CO2: Explain the ARM Cortex-M3 architecture, its instruction sets, and internal registers relevant to embedded system programming. **(Understand-L2)**

CO3: Analyze exception handling mechanisms, Nested Vectored Interrupt Controller (NVIC), and interrupt behavior in Cortex-M3. **(Analyze-L4)**

CO4: Develop embedded programs using C and assembly language with CMSIS support, including interrupt and memory protection handling. **(Apply-L3)**

CO5: Apply knowledge of STM32L15xxx microcontroller architecture and peripherals in designing, debugging, and implementing embedded system applications. **(Apply-L3)**

UNIT-I:

Embedded Concepts: Introduction to embedded systems, Application Areas, Categories of embedded systems, Overview of embedded system architecture, Specialties of embedded systems, Recent trends in embedded systems, Architecture of embedded systems, Hardware architecture, Software architecture, Application Software, Communication Software, Development and debugging Tools.
ARM Architecture: Background of ARM Architecture, Architecture Versions, Processor Naming, Instruction Set Development, Thumb-2 and Instruction Set Architecture.

UNIT-II:

Overview of Cortex-M3: Cortex-M3 Basics: Registers, General Purpose Registers, Stack Pointer, Link Register, Program Counter, Special Registers, Operation Mode, Exceptions and Interrupts, Vector Tables, Stack Memory Operations, Reset Sequence.
Instruction Sets: Assembly Basics, Instruction List, Instruction Descriptions.
Cortex-M3 Implementation Overview: Pipeline, Block Diagram, Bus Interfaces on Cortex-M3, I-Code Bus, D-Code Bus, System Bus, External PPB and DAP Bus.

UNIT-III:

Exceptions: Exception Types, Priority, Vector Tables, Interrupt Inputs and Pending Behavior, Fault Exceptions, Supervisor Call and Pendable Service Call.
NVIC: Nested Vectored Interrupt Controller Overview, Basic Interrupt Configuration, Software Interrupts and SYSTICK Timer.
Interrupt Behavior: Interrupt/Exception Sequences, Exception Exits, Nested Interrupts, Tail-Chaining Interrupts, Late Arrivals and Interrupt Latency.

UNIT-IV:

Cortex-M3/M4 Programming: Overview, Typical Development Flow, Using C, CMSIS (Cortex Microcontroller Software Interface Standard), Using Assembly.

Exception Programming: Using Interrupts, Exception/Interrupt Handlers, Software Interrupts, Vector Table Relocation. Memory Protection Unit and Other Cortex-M3 Features: MPU Registers, Setting Up the MPU, Power Management, Multiprocessor Communication.

UNIT-V:

Cortex-M3/M4 Microcontroller: STM32L15xxx ARM Cortex M3/M4 Microcontroller: Memory and Bus Architecture, Power Control, Reset and Clock Control,STM32L15xxx Peripherals: GPIOs, System Configuration Controller, NVIC, ADC, Comparators, GP Timers, USART.

Development and Debugging Tools: Software and Hardware tools like Cross Assembler, Compiler, Debugger, Simulator, In-Circuit Emulator (ICE), Logic Analyzer etc.

TEXT BOOKS:

1. The Definitive Guide to the ARM Cortex-M3, Joseph Yiu, Second Edition, Elsevier Inc. 2010.
2. Embedded/Real Time Systems Concepts, Design and Programming Black Book, Prasad, KVK.
3. David Seal "ARM Architecture Reference Manual", 2001 Addison Wesley, England; Morgan Kaufmann Publishers.

REFERENCE BOOKS:

1. Steve Furber, "ARM System-on-Chip Architecture", 2nd Edition
2. Cortex-M series-ARM Reference Manual
3. Cortex-M3 Technical Reference Manual (TRM)

25VE20 – Architectures for DSP

M. Tech. (II-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: Digital Signal Processors

Course Educational Objective:

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

CO1: Understand programmable DSP architectures and system-level design approaches. **(Understand-L2)**

CO2: Analyze memory organization, instruction sets, and superscalar SISC processors. **(Analyze-L4)**

CO3: Design and implement efficient data paths and pipelined logic structures. **(Apply-L3)**

CO4: Apply high-level synthesis techniques and low-power design strategies. **(Apply-L3)**

CO5: Utilize HDLs and prototyping tools for real-time DSP system development. **(Apply-L3)**

UNIT-I:

Digital Signal Processors: The Programmable DSP Architecture, Top-Down Design of Dedicated DSPs, A Library- Based Systems Design Environment. Classification of Architectures: An Abstract Computing Machine, Optimization of performance, Interconnection between Functional Units.

UNIT-II:

A Multi-level Classification, Data and Instruction Memories: SISC Architectures, Addressing Modes, External Interface Units. VLSI SISC Processors: The SISC Processor, Pipeline Controlin SISCs, Superscalar Processors.

UNIT-III:

Data Path Logic Design: Introduction, Synchronous Data Path Design, Monolithic Arithmetic Circuits, Implementation of Pipeline.

UNIT-IV:

High level Synthesis (HLS) of Data Path, Low power Data Design, Floating Point Arithmetic. Rapid Prototyping: Introduction, High Level Languages (HLLs) in DSP.

UNIT-V:

Hardware Description Languages (HDLs), Optimizing Compilers, DSP Prototyping Environment, Real-Time SISC Prototyping.

TEXT BOOKS:

1. Vijay. K. Madisetti,―VLSI Digital Signal Processors-An Introduction to Rapid Prototyping and Design Synthesis‖, IEEE Press, 1999.
2. Richard J. Higgins,― Digital Signal Processing in VLSI ‖,Prentice Hall,1990.
3. B. Venkata Ramani and M. Bhaskar, Digital Signal Processors, Architecture, Programming and Applications –TMH, 2004.

REFERENCE BOOKS:

1. Jonatham Stein, Digital Signal Processing ,John Wiley,2005.
2. Avtar Singhand S.Srinivasan, Digital Signal Processing–Thomson Publications

25VE21 – Internet of Things

M. Tech. (II-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites:

Course Educational Objective:

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

CO1: Analyze and compare various IoT hardware platforms and networking components including Linux-based configurations. **(Analyze-L4)**

CO2: Understand the fundamentals of networking, OSI model, and data communication concepts essential for IoT systems. **(Understand-L2)**

CO3: Explain IoT architecture, communication patterns, and protocol stacks such as 6LoWPAN with security considerations. **(Understand-L2)**

CO4: Develop IoT applications using web technologies, databases, and mobile development tools with attention to data privacy. **(Apply-L3)**

CO5: Evaluate advanced IoT use cases, sensor node integration, and the role of big data and Industry 4.0 in smart systems. **(Apply-L3)**

UNIT-I:

The IoT Networking Core: Technologies involved in IoT Development: Internet/Web and Networking Basics, OSI Model, Data transfer referred with OSI Model, IP Addressing, Point to Point Data transfer, Point to Multi Point Data transfer & Network Topologies, Sub-netting, Network Topologies referred with Web, Introduction to Web Servers, Introduction to Cloud Computing.

UNIT-II:

IoT Platform Overview: Overview of IoT supported Hardware platforms such as Raspberry Pi, ARM Cortex Processors, Arduino and Intel Galileo boards.

Network Fundamentals: Overview and working principle of Wired Networking equipment – Routers, Switches; Overview and working principle of Wireless Networking equipment – Access Points, Hubs etc. Linux Network Configuration Concepts: Networking configurations in Linux, Accessing Hardware & Device Files interactions.

UNIT-III:

IoT Architecture: History of IoT, M2M – Machine to Machine, Web of Things, IoT protocols. Applications: Remote Monitoring & Sensing, Remote Controlling, Performance Analysis.

The Architecture: The Layering concepts, IoT Communication Pattern, IoT Protocol Architecture, The 6LoWPAN. Security aspects in IoT.

UNIT-IV:

IoT Application Development: Application Protocols. Back-end Application Designing: Apache for handling HTTP Requests, PHP & MySQL for data processing, MongoDB Object type Database, HTML, CSS & jQuery for UI Designing, JSON library for data processing, Security & Privacy during development. Application Development for Mobile Platforms: Overview of Android / iOS App Development tools..

UNIT-V:

Case Study & Advanced IoT Applications: IoT applications in home, infrastructures, buildings, security, industries, home appliances, and other IoT electronic equipment. Use of Big Data and Visualization in IoT, Industry 4.0 concepts. Sensors and Sensor Nodes and interfacing using any embedded target boards (Raspberry Pi / Intel Galileo / ARM Cortex / Arduino).

TEXT BOOKS:

1. 6LoWPAN: The Wireless Embedded Internet, Zach Shelby, Carsten Bormann, Wiley
2. Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, Dr. Ovidiu Vermesan, Dr. Peter Friess, River Publishers
3. Interconnecting Smart Objects with IP: The Next Internet, Jean-Philippe Vasseur, Adam Dunkels, Morgan Kuffmann.

REFERENCE BOOKS:

1. The Internet of Things: From RFID to the Next-Generation Pervasive Network ed Lu Yan, Yan Zhang, Laurence T. Yang, Huansheng Ning
2. Internet of Things (A Hands-on-Approach), Vijay Madisetti, Arshdeep Bahga
3. Designing the Internet of Things, Adrian Mc Ewen (Author), Hakim Cassimally
4. Asoke K Talukder and Roopa R Yavagal, “Mobile Computing,” Tata Mc Graw Hill, 2010

25VE22 – Embedded Network and Protocols

M. Tech. (II-Sem.)

L	T	P	Cr.
3	0	0	3

Pre-requisites: ARM Controllers

Course Educational Objective:

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

CO1: Acquire knowledge on communication protocols of connecting Embedded Systems. **(Understand-L2)**

CO2: Master the design level parameters of USB and CAN bus protocols. **(Apply-L3)**

CO3: Design Ethernet in Embedded networks considering different issues. **(Apply-L3)**

CO4: Acquire the knowledge of wireless protocols in Embedded domain. **(Understand-L2)**

UNIT-I:

Embedded Networking: Introduction – Serial/Parallel Communication – Serial communication protocols - RS232 standard – RS485 – Synchronous Serial Protocols - Serial Peripheral Interface (SPI) – Inter Integrated Circuits (I2C) – PC Parallel port programming - ISA/PCI Bus protocols – Firewire.

UNIT-II:

USB bus – Introduction – Speed Identification on the bus – USB States – USB bus communication Packets – Data flow types – Enumeration – Descriptors – PIC18 Microcontroller USB Interface – C Programs – CAN Bus – Introduction - Frames – Bit stuffing – Types of errors – Nominal Bit Timing – PIC microcontroller CAN Interface – A simple application with CAN.

UNIT-III:

Elements of a network – Inside Ethernet – Building a Network: Hardware options – Cables, Connections and network speed – Design choices: Selecting components – Ethernet Controllers – Using the internet in local and internet communications – Inside the Internet protocol.

UNIT-IV:

Exchanging messages using UDP and TCP – Serving web pages with Dynamic Data – Serving web pages that respond to user Input – Email for Embedded Systems – Using FTP – Keeping Devices and Network secure.

UNIT-V:

Wireless sensor networks – Introduction – Applications – Network Topology – Localization – Time Synchronization – Energy efficient MAC protocols – SMAC – Energy efficient and robust routing – Data Centric routing.

TEXT BOOKS:

1. Embedded Systems Design: A Unified Hardware/Software Introduction-Frank Vahid, Tony Givargis, John & Wiley Publications, 2002
2. Parallel Port Complete: Programming, interfacing and using the PCs parallel printer port- Jan Axelson, Penram Publications, 1996

REFERENCE BOOKS:

1. Advanced PIC microcontroller projects in C: from USB to RTOS with the PIC18F series – Dogan Ibrahim, Elsevier 2008.
2. Embedded Ethernet and Internet Complete-Jan Axelson, Penram publications, 2003.
3. Networking Wireless Sensors-Bhaskar Krishnama chari ,Cambridge press2005

25VE53 – Digital CMOS Circuit Design Lab

M. Tech. (II-Sem.)

L	T	P	Cr.
0	1	2	2

Pre-requisites:

Course Educational Objective:

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

CO1: Have the ability to explain the VLSI Design Methodologies using Mentor Graphics Tools. **(Understand-L2)**

CO2: Grasp the significance of various design logic Circuits in full-custom IC Design. **(Apply-L3)**

CO3: Have the ability to explain the Physical Verification in Layout Extraction. **(Understand-L2)**

CO4: Fully Appreciate the design and analyze of CMOS Digital Circuits. **(Apply-L3)**

CO5: Grasp the Significance of Pre-Layout Simulation and Post-Layout Simulation. **(Apply-L3)**

List of Experiments:

1. Inverter Characteristics.
2. NAND and NOR Gate
3. XOR and XNOR Gate
4. 2:1 Multiplexer
5. Full Adder
6. RS-Latch
7. Clock Divider
8. JK-Flip Flop
9. Synchronous Counter
10. Asynchronous Counter
11. Static RAM Cell
12. Dynamic Logic Circuits
13. Linear Feedback Shift Register

Lab Requirements:

Software:

1. Mentor Graphics Tool/Cadence/ Synopsys/Industry Equivalent Standard Software

Hardware:

1. Personal Computer with necessary peripherals, configuration and operating System.

25VE54 – System Design with Embedded Linux Lab

M. Tech. (II-Sem.)

L	T	P	Cr.
0	1	2	2

Pre-requisites:

Course Educational Objective:

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

CO1: Demonstrate the ability to interface sensors and actuators with microcontroller boards. **(Understand-L2)**

CO2: Develop applications using Raspberry Pi for real-time control of output devices and sensor monitoring. **(Apply-L3)**

CO3: Design embedded systems using BeagleBone board for basic input/output operations and display interfacing. **(Apply-L3)**

CO4: Interface input devices and sensors with Embedded Linux boards and develop basic human-machine interaction applications. **(Apply-L3)**

CO5: Integrate sensors, actuators, and communication interfaces to build real-time embedded applications. **(Apply-L3)**

CO6: Demonstrate debugging and testing skills for verifying sensor data, controlling actuators, and troubleshooting embedded systems. **(Understand-L2)**

Using Ardiuno Board

1. Temperature and Humidity sensor
2. Soil moisture
3. Ultra sonic sound sensor to measure distance
4. IR Sensor

Using Raspberry PI

1. Servo motor
2. MQ2 Gas sensor
3. LCD
4. Relay

Using beagle bone boards

1. Led blinking
2. Seven segment display
3. LCD
4. Switch(buzzer)

Using embedded Linux Board

1. 4×4Matrix
2. 2. Light dependent resistor

M. Tech. (VLSI& Embedded Systems Design)
III-Semester
Detailed Syllabus

25RM01 – Research Methodology and IPR

M. Tech. (III-Sem.)

L	T	P	Cr.
3	0	0	3

Course Outcomes: At the end of the course, the student will be able to:
CO1: Understand research problem formulation. **(Understand-L2)**
CO2: Analyze research related information, Follow research ethics **(Understand-L2)**
CO3: Understand that today's world is controlled by Computer, Information Technology, but tomorrow world will be ruled by ideas, concept, and creativity. **(Understand-L2)**
CO4: Understanding that when IPR would take such important place in growth of individuals & nation, it is needless to emphasise the need of information about Intellectual Property Right to be promoted among students in general & engineering in particular. **(Understand-L2)**
CO5: Understand that IPR protection provides an incentive to inventors for further research work and investment in R & D, which leads to creation of new and better products, and in turn brings about, economic growth and social benefits. **(Understand-L2)**

UNIT-I:
Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem, Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations

UNIT-Il:
Effective literature studies approaches, analysis 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-Ill:
Nature of Intellectual Property: Patents, Designs, Trademarks and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT

UNIT-IV:
Patent Rights: Scope of Patent Rights, Licensing and transfer of technology, Patent information and databases, Geographical Indications.

UNIT-V:
New Developments in IPR: Administration of Patent System. New developments in IPR; IPR of Biological Systems, Computer Software etc, Traditional knowledge Case Studies, IPR and IITs

TEXT BOOKS

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

REFERENCE BOOKS

1. Ranjit Kumar, 2nd Edition , "Research Methodology: A Step by Step Guide for beginners"
2. Halbert, "Resisting Intellectual Property", Taylor & Francis Ltd, 2007.
3. Mayall , "Industrial Design", McGraw Hill,1992.