Curriculum
2021

M. Tech.
VLSI and Embedded system Design

(Duration of Study : 2 years)

GMRIT
Training Tomorrow's Engineers Today

Department of Electronics and Communication Engineering
GMR Institute of Technology
Rajam, Andhra Pradesh
(An Autonomous Institute Affiliated to JNTU Kakinada, AP)
NBA Accredited and NAAC Accredited

Department of Electronics and Communication Engineering VLSI and Embedded system Design

[Minimum Credits to be earned: 68]

No	Course Code	Course	Periods			
			L	T	P	C
First Semester						
1	21MEX101	Advanced Optimization Techniques	4	-	-	4
2	21VLS101	VLSI Technology and Design	4	-	-	4
3		Elective I	4	-	-	4
4		Elective II	4	-	-	4
5		Elective III	4	-	-	4
6	21VLS102	HDL Programming Laboratory	-	-	3	1.5
7	21VLS103	Term Paper	-	-	3	1.5
		Total	20	-	6	23
Second Semester						
1	21VLS201	Embedded and Real Time Systems	4	-	-	4
2	21VLS202	Algorithms for VLSI Design Automation	4	-	-	4
3		Elective IV	4	-	-	4
4		Elective V	4	-	-	4
5		Elective VI	4	-	-	4
6	21VLS203	Embedded systems Laboratory	-	-	3	1.5
7	21VLS204	IC Implementation Laboratory	-	-	3	1.5
		Total	20	-	6	23

No	Course Code	Course	Periods			
			L	T	P	C
Third Semester						
1	21VLS301	Internship	-	-	-	4
2	21VLS302	Project	-	-	-	-
3		Research Methodology and IPR (Audit Course)	-	-	-	0
		Total	-	-	-	4
Fourth Semester						
1	21VLS302	Project	-	-	-	18

List of Elective Courses

Elective I						
No	Course Code	Course	Periods			
			L	T	P	C
1	21VLS001	Analog and Digital IC Design	4	-	-	4
2	21VLS002	Digital Design through VERILOG	4	-	-	4
3	21VLS003	Embedded Software Design	4	-	-	4
Elective II						
1	21VLS004	Advanced Microcontrollers and Processors	4	-	-	4
2	21VLS005	DSP Processors and Architectures	4	-	-	4
3	21VLS006	VLSI Signal Processing	4	-	-	4
Elective III						
No	Course Code	Course	Periods			
			L	T	P	C
1	21VLS007	Embedded System Design	4	-	-	4
2	21VLS008	System Verilog Programming for Verification	4	-	-	4
3	21VLS009	IoT system Design	4	-	-	4
Elective IV						
1	21VLS010	Design of Fault Tolerant System	4	-	-	4
2	21VLS011	Embedded Networking	4	-	-	4
3	21VLS012	Low Power VLSI Design	4	-	-	4
Elective V						
1	21VLS013	CPLD and FPGA Architectures & Applications	4	-	-	4
2	21VLS014	Hardware Software Co-design	4	-	-	4
3	21VLS015	System Modeling & Simulation	4	-	-	4
Elective VI						
1	21CSE203	Soft Computing Techniques	4	-	-	4
2	21VLS016	Memory Architectures	4	-	-	4
3	21VLS017	Programming Languages for Embedded Systems	4	-	-	4

21MEX101 Advanced Optimization Techniques

4 0 0 4

Course Outcomes

1. Design of mechanical systems and interdisciplinary engineering applications and business solutions using suitable optimization technique.
2. Apply numerical or iterative techniques in power systems for optimal power flow solutions.
3. Optimize the parameters in control systems for desired steady state or transient response.
4. Optimize the cost function in deciding economic factors of power systems.
5. Design of electrical systems optimally using suitable techniques like univariate method, steepest descent method etc.
6. Design of electrical systems optimally using, steepest and descent method etc

Unit I
Linear programming and Assignment Problem

Linear programming-Two-phase simplex method, Big-M method, duality, interpretation, applications, Assignment problem- Hungarian's algorithm, Degeneracy, applications, unbalanced problems, traveling salesman problem

Applications of assignment problems **11+4 Hours**

Unit II
Classical and Numerical Optimization Techniques

Classical optimization techniques-Single variable optimization with and without constraints, multi–variable optimization without constraints, multi–variable optimization with constraints–method of Lagrange multipliers, Kuhn-Tucker conditions.
Numerical methods for optimization-Nelder Mead's Simplex search method, Gradient of a function, Steepest descent method, Newton's method, types of penalty methods for handling constraints

Exterior penalty function method for handling constraint **11+4 Hours**

Unit III
Genetic algorithm and Programming

Genetic algorithm (GA) -Differences and similarities between conventional and evolutionary algorithms, working principle, reproduction, crossover, mutation, termination criteria, different reproduction and crossover operators, GA for constrained optimization, draw backs of GA.
Genetic Programming (GP)-Principles of genetic programming, terminal sets, functional sets, differences between GA & GP, random population generation, solving differential equations using GP

Solving differential equations using GP **12+4Hours**

Unit IV
Multi-Objective GA

Multi-ObjPareto's analysis, Non-dominated front, multi–objective GA, Non-dominated sorted GA, convergence criterion, applications of multi-objective problems
Basic Problem solving using Genetic algorithm, Genetic Programming & Multi Objective GA and simple applications of optimization for engineering systems

Simple applications of optimization for engineering systems

11+3 Hours
Total: 45+15 Hours

Textbook (s)

1. J. S. Arora, Introduction to Optimum Design, McGraw Hill International Ed., NY, 1989
2. K. Deb, Optimization for Engineering Design: Algorithms and Examples, 2nd Ed., PHI, 1995
3. S. S. Rao, Engineering Optimization: Theory and Practice, New Age International (P) Ltd., 2001

Reference (s)

1. D. E. Goldberg, Genetic Algorithms in Search and Optimization, Pearson publication, 1990
2. J. R. Koza, Genetic Programming, MIT Press, 1993
3. K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley, 2001

21VLS101 VLSI Technology and Design

4 0 0 4

Course Outcomes

1. Identify the issues related to the IC Fabrication Process.
2. Explain the design of better devices with IC technology
3. Make use of circuit characterization and performance estimation.
4. Apply the concepts of Logic Gates, Combinational and Sequential networks to design the better.
5. Apply design rules for better layouts
6. Illustrate concepts of floor planning and Routing.

Unit I
Introduction to MOS Technologies & MOS Circuits

MOS, CMOS, Bi CMOS Technology trends and projections. MOS Fabrication Processes.Ids-Vds relationships, Threshold voltage Vt, Gm, Gds and Wo, Pass Transistor, MOS, CMOS& Bi CMOS Inverters, Zpu/Zpd, MOS Transistor circuit model, Latch-up in CMOS circuits.

Channel length modulation, mobility degradation **11+3 Hours**

Unit II
Circuit Characterisation and Performance Estimation & Logic Gates

Resistance estimation - Capacitance estimation, Inductance - Switching characteristics ,Transistor sizing - Power dissipation and design margining ,Charge sharing - Scaling.
Static complementary gates, switch logic-pass transistor and transmission gate logic, Alternative gate circuits.

Energy Delay optimization **12+4 Hours**

Unit III
Combinational Logic Networks & Sequential Systems

Layouts, Simulation, Network delay, interconnect design, power optimization, Switch logic networks, Gate and Network testing.
Memory cells and Arrays, clocking disciplines, Design, power optimization, Design validation and testing.

True Single Phase Clock (TSPC) Latches and Flip-Flops **11+4 Hours**

Unit IV
Layout Design, Floor Planning & Routing

Stick Diagrams, Scalable Design rules, Layout Design tools. Floor Planning concepts, Shape functions and Floorplan sizing, Types of Local Routing problems ,Area routing , Channel routing , Global routing , Algorithms for Global Routing

Micron based design rules **11+4 Hours**

Total: 45+15 Hours

Textbook (s)

1. K. Eshraghian et. al ,Essentials of VLSI Circuits and Systems, PHI of India Ltd., 2005
2. Wayne Wolf, Modern VLSI Design, 3rd Edition, Pearson Education, fifth Indian Reprint, 2005.

Reference (s)

1. N.H.E Weste, K.Eshraghian, Principals of CMOS Design,Adison Wesley, 2nd Edition,1993
2. Fabricius,Introduction to VLSI Design,MGH International Edition, 1990.
3. Baker, Li Boyce, CMOS Circuit Design, Layout and Simulation, PHI, 2004.
4. D. Shugard J. Fishburn and K. Keutzer, Algorithms and Techniques for VLSI Layout Synthesis,Kluwer Academic Publishers, 1989.

21VLS102 HDL Programming Laboratory

0 0 3 1.5

Course Outcomes

1. Design and implement the fundamental digital logic circuits using Verilog HDL.
2. Make use of system level design.
3. Create the technology schematics based on the design.
4. Illustrate t design rule checks and timing parameters.
5. Explain the resources consumed by the design on FPGA.
6. Illusrtae the Interconnections during place and route.

List of Experiments

1. Digital Circuits Design (Sub-System Level) Description using Verilog and VHDL.
2. Verification of the Functionality of Designed Circuits using functional Simulator.
3. Timing simulation for critical path time calculation.
4. Synthesis Reports of top-order designed Digital circuits.
5. Place and Route techniques for major FPGA vendors such as Xilinx, Altera and Actel etc.
6. Implementation of Designed Digital Circuits using FPGA and CPLD devices

List of Mini-Projects[1]

1. Signed and Unsigned Multiplier Design
2. ALU Design
3. MAC Design

Reading Material (s)

1. T.R. Padmanabhan and B. Bala Tripura Sundari, Design through Verilog HDL, WSE, 2004.
2. Stephen. Brown and ZvonkoVranesic, Fundamentals of Logic Design with Verilog, TMH, 2005.
3. Michael D. Ciletti,Advanced Digital Design with Verilog HDL, PHI, 2005.
4. J. Bhaskar, A Verilog Primier, BSP, 2003.

[1] Students shall opt any one of the Mini–Projects in addition to the regular experiments

21VLS201 Embedded and Real time systems

4 0 0 4

Course Outcomes

1. Define many issues involved with embedded systems.
2. Classify different Real-Time Operating System concepts.
3. Apply Program using system calls in ID Environment.
4. Build an embedded system with tasks and executive.
5. Make use of tools to build an embedded real-time system.
6. Outline the implementation of a small embedded system effectively .

Unit I
Introduction

Embedded systems over view, design challenges, processor technology, Design technology, Trade-offs. Single purpose processors, RT-level combinational logic, sequential logic (RTlevel),custom purpose processor design(RT -level), optimizing custom single purpose
processors.General Purpose Processors - Basic architecture, operations, programmer's view, development environment, Application specific Instruction –Set processors (ASIPs)-Micro controllers and Digital signal processsors.

Application of Embedded systems design.

11+3 Hours

Unit II
State Machine And Concurrent Process Models And Communication Processes

Introduction, models Vs Languages, finite state machines with data path model (FSMD), program state machine model(PSM, concurrent process model, concurrent processes, communication among processes, synchronization among processes, Implementation, data flow model, real-time systems.Communication Processes – Need for communication interfaces, RS232/UART, RS422/RS485, USB, Infrared, IEEE1394 Firewire, Ethernet, IEEE 802.11, Blue tooth.

Existed embedded project implementation with the models.

10+4 Hours

Unit III
Introduction To Real Time Systems And Programming Languages And Tools

Introduction, Issues in Real Time Computing, Structure of a Real Time System, Task classes, Performance Measures for Real Time Systems, Estimating Program Run Times. Task Assignment and Scheduling – Classical uniprocessor scheduling algorithms, Uniprocessor scheduling of IRIS tasks, Task assignment, Mode changes, and Fault Tolerant Scheduling.
Programming Languages and Tools – Desired language characteristics, Data typing, Control structures, Facilitating Hierarchical Decomposition, Packages, Run – time (Exception) Error handling, Overloading and Generics, Multitasking, Low level programming, Task Scheduling, Timing Specifications, Programming Environments, Run – time support**.**

C and C++ Programming with Visual C basic concepts.

15+5 Hours

Unit IV
Real Time Databases And Design Technology

Real time Database, Basic Definition, Real time Vs General Purpose Databases, Main Memory Databases, Transaction priorities, Transaction Aborts, Concurrency control issues, Disk Scheduling Algorithms, Two – phase Approach to improve Predictability, Maintaining Serialization Consistency, and Databases for Hard Real Time Systems. Automation, Synthesis, parallel evolution of compilation and synthesis, Logic synthesis, RT synthesis, Behavioral Synthesis, Systems Synthesis and Hard ware/Software Co-Design, Verification, Hardware/Software co-simulation, Reuse of intellectual property codes.

DBMS concept , Simulation tools

9+3 Hours
Total: 45+15 Hours

Textbook (s)

1. Frank Vahid, Tony D.Givargis,John Wiley & Sons,Embedded System Design-A Unified Hardware/Software Introduction, Wiley,Inc.2002.
2. KVKK prased ,Embedded/Real Time Systems, Dreamtech press,2005.
3. Krishna. C. M, Kang. G, Shin, Real Time Systems, McGraw Hill, 2003.

Reference (s)

1. Herma. K, Real Time Systems–Design for distributed Embedded Applications, Kluwer Academic, 2002.
2. Charles Crowley, Operating Systems-A Design Oriented approach,McGraw Hill, 2004.
3. Raymond J.A.Bhur, Donald L.Bailey, “An Introduction to Real Time Systems”, PHI 2002.

21VLS202 Algorithms for VLSI Design Automation

4 0 0 4

Course Outcomes

1. Define tractable and intractable methods for VLSI design
2. Apply Backtracking, Branch and Bound Methods for combinational optimization
3. Apply Genetic Algorithm for combinational optimization
4. Apply the concept of routing and placement for layout compaction
5. Analyze logic synthesis and verification of the design of digital circuits
6. Analyze the physical design flow of FPGA's and MCM's.

Unit I
Design Methodologies & Combinational Optimization

Introduction to Design Methodologies, Design Automation tools, Algorithmic Graph Theory, Computational Complexity, Tractable and Intractable Problems.
Backtracking, Branch and Bound, Dynamic Programming, Integer Linear Programming, Local Search, Simulated Annealing, Tabu search, Genetic Algorithms.

Graph Algorithms: Depth-First Search, Breadth-First Search **12+4 Hours**

Unit II
Layout & Simulation

Layout Compaction, Placement, Floorplanning and Routing Problems, Concepts and Algorithms.Gate Level Modelling and Simulation, Switch level modeling and simulation

Partitioning, Kernighan-Lin Partitioning Algorithm **11+4 Hours**

Unit III
Logic Synthesis and High Level Synthesis

Basic issues and Terminology, Binary ,Decision diagram, Two- Level Logic Synthesis.
Hardware Models, Internal representation of the input algorithm, Allocation, Assignment and Scheduling, Some Scheduling Algorithms, Some aspects of Assignment problem, High – level Transformations.

Heuristic based on ROBDDDs **11+4 Hours**

Unit IV
Physical Design Automation

FPGA technologies, Physical Design cycle for FPGA's partitioning and Routing for segmented and staggered models.
MCM technologies, MCM physical design cycle, Partitioning, Placement – Chip array based and full custom approaches, Routing – Maze routing, Multiple stage routing, Topologic routing, Integrated Pin – Distribution and routing

Routing in Programmable Multichip Modules **11+3 Hours**

Total:45+15 Hours

Textbook (s)

1. S. H. Gerez ,Algorithms for VLSI Design Automation, WILEY student Edition, John Wiley & Sons (Asia) Pvt. Ltd. 1999.
2. Naveed Sherwani, Algorithms for VLSI Physical Design Automation, 3rd Edition, Springer International Edition, 2005.

Reference (s)

1. Fredrick J. Hill and Gerald R. Peterson, Computer Aided Logical Design with Emphasis on VLSI, Wiley Publishers, 4th Edition, 1993.
2. Wayne Wolf, Modern VLSI Design: Systems on silicon, Pearson Education Asia, 2nd Edition, 1998.

21VLS203 Embedded systems laboratory

0 0 3 1.5

Course Outcomes

1. Design prototype embedded projects using ARM9
2. Program and test different software and hardware modules
3. Program the I2C Serial Communication Bus.
4. Test and identify the real time bugs with master and slave devices using I2C
5. Capture the real time data transfer properties
6. Implement Automatic Speed Detection digital I/O for synchronizing etc using USB Analyzer

List of Experiments

1. Basic data streaming with LEDs.
2. Interfacing of Seven Segment Display.
3. Interfacing of Matrix Key Pad.
4. Serial communication (USART).
5. Simulate an I2C master or slave device.
6. Program and verify I2C-based memory device.
7. Passively monitor an I2C bus in real-time.
8. Real-time capture and delayed-download capture
9. High-Speed USB Chirp Detection
10. Automatic Speed Detection
11. Digital I/O for synchronizing

List of Mini-Projects[2]

1. Implement Traffic light

Reading Material(s)

1. Rajkamal, Embedded Systems: Architecture, Programming and Design, TMH Publications, Second Edition, 2008

[2] Students shall opt any one of the Mini–Projects in addition to the regular experiments

21VLS204 IC Implementation Laboratory

0 0 3 1.5

Course Outcomes

1. Carry out the simulations of MOSFET circuits
2. Implement the layout diagrams for CMOS circuits
3. Carry out the DRC and LVS in a given circuit implementation
4. Compute the effects of parasitic present in the various layers of the layouts
5. Carry out the simulations of basic analog circuits
6. Demonstrate the ASIC implementation flow

List of Experiments

1. Verify the functionality of a CMOS inverter
2. Verify the functionality of a CMOS AND gate
3. Design the layout of a CMOS OR gate
4. Perform the LVS and DRC for the CMOS OR gate Layout
5. Perform the RC extraction for the CMOS inverter
6. Compute the delay of the CMOS inverter after the RC extraction
7. Simulate a basic Common Source amplifier
8. Simulate a basic Common Drain amplifier
9. Design and simulate the 2:1 MUX using transmission gate logic
10. Perform the design and ASIC Implementation of an 8-bit ripple carry adder

List of Mini-Projects[2]

1. Carry out the ASIC implementation of a universal shift register

2. Carry out the full custom IC implementation of a ALU performing four arithmetic and four logical operations.

Reading Material(s)

1. Kamran Eshraghian, Douglas ,A. Pucknell And Sholeh Eshraghian, Essentials of VLSI Circuits and Systems, , Prentice-Hall of India Private Limited, 2005 Edition.
2. Behzad Razavi, Design of Analog CMOS Integrated Circuits, McGraw Hill, 2003
3. Jan M. Rabaey, Anantha Chandrakasan and Borivoje Nikolic, Digital Integrated Circuits, Pearson Education, 2nd edition,2016.
4. Weste and Eshraghian, Principles of CMOS VLSI Design, Pearson Education, 3rdEdition, 1999
5. Michael john Sebastian smith, Application specification integrated circuits, Addition Wesley,1st
6. edition,1997

[2] Students shall opt any one of the Mini–Projects in addition to the regular experiments

Research Methodology and IPR

0 0 0 0

Unit I

Research Problem And Scope For Solution

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

Unit II

Format

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 III

Process and Development

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

UNIT IV

Patent Rights and IPR

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

Textbook (s)

1. Goddard, Wayne, and Stuart Melville. Research methodology: An introduction. Juta and Company Ltd, 2004.
2. Kumar, Ranjit. Research methodology: A step-by-step guide for beginners. Sage, 2018.

Reference (s)

1. Halbert, "Resist ing Intellectual Property", Taylor & Francis Ltd ,2007.
2. Mayall, "Industrial Design", McGraw Hill, 1992.
3. Niebel, "Product Design", McGraw Hill, 1974.
4. Asimov, "Introduction to Design", Prent ice Hall, 1962.
5. Robert P. Merges, Peter S. Menell, Mark A. Lemley, " Intellectual Property in New Technological Age", 2016.
6. T. Ramappa, "Intellectual Property Rights Under WTO", S. Chand, 2008

21VLS001 Analog and Digital IC design (Elective – I)

4 0 0 4

Course Outcomes

1. Design and implement the fundamental analog IC blocks
2. Demonstrate the internal circuits and topologies of Opamp
3. Analyze the PLL and switched capacitors
4. Demonstrate the Verilog models for combinational and sequential circuits
5. Demonstrate digital integrated circuits building blocks
6. Explain the internal structures of ADC and DAC

Unit I

Current Mirrors, Single Stage Amplifiers and Operational Amplifiers

Simple COMS, BJT current mirror, Cascode, Wilson and Widlar current mirrors. Common Source amplifier source follower, common gate amplifier.General considerations one – state op-amps, two stage opamps-gains boosting stage- comparison I/P range limitations slew rate**.**

Folded cascode opamp

11+4 Hours

Unit II

Comporators And Swithched Capacitors Circuits

Using an op-amp for a comparator, charge injection error- latched Comparator,Basic Building blocks op-amps capacitors switches, non-over lapping clocks, Basic operations and analysis-resistor equivalence of la switched capacitor- parasitic sensitive integrator parasitic insensitive integrators signal flow graph analysis-First order filters- switch sharing fully differential filters.

Examples of Bipolar comparators **11+4 Hours**

Unit III

Combinational and Sequential Ic Design By Using Verilog, Logic Families

VERILOG modeling for decoders, encoders, multiplexers, adders and subtractors. VERILOG modeling for latches, flip flops, counters, shift registers, FSMs.

COMS, TTL, ECL, logic families COMS / TTL- interfacing and comparison of logic families.

Low power issues in BiCMOS logic families

11+4 Hours

Unit IV

Digitalintegradedsystembuilding Blocks, DAC and ADCs

Multiplexers, decoders, barrel shifters, counters and digital single bit adders.Decoder based converter resistor storing converters folded resistor string converter, Binary scale converters ,Binary weighted resistor converters – Reduced resistance ratio ladders, R-2R based converters, Thermometer code current mode D/A converters.Integrating converters, successive approximation converters. DAC based successive approximation, flash converters time interleaved A/D converters.

Hybrid Converters

12+3 Hours
Total: 45+15 Hours

Textbook (s)

1. David A Johns, Ken Martin, Analog Integrated circuit Design, John Wiley & Sons.1996.
2. Behzad Razavi, Design of Analog CMOS Integrated Circuits, TMH, 2003
3. Ken Martin, Digital Integrated Circuit Design, Oxford University, 2000.
4. John F Wakerly, Digital Design Principles & Practices ,Pearson Education & Xilinx Design Series, 3rd Edition, 2002.

Reference (s)

1. Ken Martin Digital Integrated Circuit Design, Oxford University, 2000.
2. John F Wakerly, Digital Design Principles & Practices, Pearson Education & Xilinx Design Series, 3rd Edition, 2002.
3. Samir Palnitkar, Verylog HDL-A Guide to Digital Design and Synthesis, Prentice Hall India, 2002.

21VLS002 Digital design through VERILOG (Elective – I)

4 0 0 4

Course Outcomes

1. Illustrate the importance of EDA tools and VLSI designs
2. Design and implement the fundamental digital logic circuits using Verilog HDL
3. Explain the system level design and related concepts.
4. Design and Implement the digital designs against timing parameters
5. Apply knowledge of drawing SM charts
6. Create the basic awareness on FPGA and CPLD architectures.

Unit I

Introduction to Verilog, Language Constructs & Conventions and Gate Level Modeling

Verilog as HDL, Levels of Design Description, Concurrency, Simulation and Synthesis, Functional Verification, System Tasks, Programming Language Interface (PLI), Module, Simulation and Synthesis Tools, Test Benches.
Introduction, Keywords, Identifiers, White Space Characters, Comments, Numbers, Strings, Logic Values, Strengths, Data Types, Scalars and Vectors, Parameters, Memory, Operators, System Tasks, Exercises.
Introduction, AND Gate Primitive, Module Structure, Other Gate Primitives, Illustrative Examples, Tri-State Gates, Array of Instances of Primitives, Additional Examples, Design of Flip flops with Gate Primitives, Delays, Strengths and Contention Resolution, Net Types, Design of Basic Circuits, Exercises.

VLSI Design, ASIC Design Flow

12+4 Hours

Unit II

Behavioral Modeling, Modeling at Data Flow Level and Switch Level Modeling

Introduction, Operations and Assignments, Functional Bifurcation, Initial Construct, Always Construct, Examples, Assignments with Delays, Wait construct, Multiple Always Blocks, Designs at Behavioral Level, Blocking and Non blocking Assignments, The case statement, Simulation Flow. i*f* and i*f*-else constructs, assign-deassign construct, repeat construct, for loop, the disable construct, while loop, forever loop, parallel blocks, force-release construct, Event.
Introduction, Continuous Assignment Structures, Delays and Continuous Assignments, Assignment to Vectors, Operators.
Introduction, Basic Transistor Switches, CMOS Switch, Bi-directional Gates, Time Delays with Switch Primitives, Instantiations with Strengths and Delays, Strength Contention with Trireg Nets, Exercises.

Fork-Join construct

11+3 Hours

Unit III

System Tasks, Functions, Compiler Directives, Functions & Tasks, User-Defined Primitives and Digital Design with SM Charts

Introduction, Parameters, Path Delays, Module Parameters, System Tasks and Functions, File-Based Tasks and Functions, Compiler Directives, Hierarchical Access, General Observations, Exercises,
Introduction, Function, Tasks, User- Defined Primitives (UDP), FSM Design (Moore and Mealy Machines)
State Machine Charts, Derivation of SM Charts, Realization of SM Charts, Implementation of the Dice Game, Alternative realizations for SM Charts using Microprogramming, Linked State Machines.

Design of Finite State Machines **11+4 Hours**

Unit IV

Designing with Programmable Gate Arrays and Complex Programmable Logic Devices, Verilog Models
Xilinx 3000 Series FPGAs, Designing with FPGAs, Using a One-Hot State Assignment, Altera Complex Programmable Logic Devices (CPLDs), Altera FLEX 10K Series CPLDs.Static RAM Memory, A simplified 486 Bus Model, Interfacing Memory to a Microprocessor Bus, UART Design, Design of Microcontroller CPU.

Xilinx 4000 Series **11+4 Hours**

Total: 45+15 Hours

Textbook (s)

1. T.R. Padmanabhan and B. Bala Tripura Sundari, Design through Verilog HDL, WSE, 2004.
2. J. Bhaskar, A Verilog Primier, BSP, 2003.

Reference (s)

1. Stephen. Brown and Zvonko Vranesic, Fundamentals of Logic Design with Verilog, TMH, 2005.
2. Charles H Roth, Digital Systems Design using VHDL, Thomson Publications, 2004.
3. Michael D. Ciletti, Advanced Digital Design with Verilog HDL, PHI, 2005.

21VLS003 Embedded Software Design (Elective – I)

4 0 0 4

Course Outcomes

1. Design and analyze an embedded system.
2. Identify the various steps involved in embedded Software development.
3. Analyze the various intermediate results during debug process to re assemble the system components based on requirement.
4. Illustrate current software porting technologies and problems.
5. Test for the software of an embedded system to justify the system design.
6. Analyze the complete embedded life cycle in designing of embedded systems

Unit I
Embedded Design Life Cycle

Introduction, Product Specification, Hardware/software partitioning, Iteration and Implementation, Detailed hardware and software design, Hardware/Software integration, Product Testing and Release, Maintaining and upgrading existing products. Selection Process: Packaging the Silicon, Adequate Performance, RTOS Availability, Tool chain Availability, Other issues in the Selection process.

Small Embedded system Design **11+4 Hours**

Unit II
Partitioning Decision

Hardware/Software Duality, Hardware Trends, ASICs and Revision Costs.Development Environment: The Execution Environment, Memory Organization, System Startup. Special Software Techniques: Manipulating the Hardware, Interrupts and Interrupt service Routines (ISRs), Watchdog Times, Flash Memory, Design Methodology. Basic Tool Set: Host – Based Debugging, Remote Debuggers and Debug Kernels, ROM Emulator, Logic Analyzer.

Introduction to Interrupts **11+4 Hours**

Unit III
Background Debug Mode

Background Debug Mode, Joint Test Action Group (JTAG) and Nexus. ICE – Integrated Solution: Bullet Proof Run Control, Real time track, Hardware Break points, Overlay memory, Timing Constrains, Usage Issue, Setting the Trigger. Testing: Why Test? When to Test? Which Test? When to Stop? Choosing Test cases, Testing Embedded Software, Performance Testing Maintenance and Testing, The Future.

Introduction to debugging concept **11+4 Hours**

Unit IV
Writing Software for Embedded Systems

The compilation Process, Native Versus Cross-Compilers, Runtime Libraries, Writing a Library, Using alternative Libraries, using a standard Library, Porting Kernels, C extensions for Embedded Systems, Downloading. Emulation and debugging techniques; Buffering and Other Data Structures: What is a buffer? Linear Buffers, Directional Buffers, Double Buffering, Buffer Exchange, Linked Lists, FIFOs, Circular Buffers, Buffer Under run and Overrun, Allocating Buffer Memory, Memory Leakage. Memory and Performance Trade-offs.

Advantages of buffers **12+3hours**

Total: 45+15 Hours

Textbook (s)

1. Arnold Berger, Embedded Systems-An introduction to processes, tools and techniques, Elsevier, New Delhi, 2010.
2. Barry B. Brey, The Intel Microprocessors, 4th Edition, PHI, 1997.

Reference (s)

1. Frank Vahid, Tony D. Givargis, Embedded system Design: A Unified Hardware/Software Introduction, John Wiley & Sons Inc.2002.
2. Rajkamal, Embedded Systems: Architecture, Programming and Design, TMH Publications, 2nd Ed, 2008.

21VLS004 Advanced Microcontrollers and Processors (Elective – II)

4 0 0 4

Course Outcomes

1. Design the home appliances and toys using Microcontroller chips.
2. Design computers like desktops, laptops using various processors.
3. Design the high speed communication circuits using serial bus connection.
4. Apply commercial CPU(s) as realistic vehicles to demonstrate these concepts by introducing CPU instructions and internal register structures.
5. Analyze full internal workings of a typical simple CPU including the utilization of the various hardware resources during the execution of instructions.
6. Design different types of embedded systems

Unit I
General Microcontrollers

Introduction to the 8051& 8052 microcontrollers, features, architecture, memory organization, addressing modes, instruction set, assembly programming, software development tools, parallel I/O ports, interrupts, timers/counters, serial communication, data and control transfer operations, serial data transmissions, programming and interfacing using 8051.Introduction to Atmel microcontrollers (89CXX and 89C20XX), Architectural overview of Atmel 89C51 and Atmel 89C2051,pin descriptions of Atmel microcontrollers, using flash memory devices Atmel 89XX and Atmel 89C20XX, Applications of MCS-51 and Atmel 89C51 AND 89C2051 microcontrollers.

Digital clock design **12+4 Hours**

Unit II
PIC and AVR Microcontrollers

An introduction to PIC microcontrollers, PIC 8 Series and PIC 16 series microcontrollers and PIC family of microcontrollers (16C6X/7X,16F84A, 12F50X and 16F8XX), architecture, instruction set , programming using assembly and c languages of the PIC microcontrollers, interfacing PIC Microcontroller to other devices, applications of PIC microcontrollers.Introduction to AVR microcontroller, AVR RISC microcontroller architecture, AVR instructions set, AVR hardware design issues, hardware and software interfacing with AVR, communications links for the AVR, AVR system development tools.

Integrated development environment tools **11+4 Hours**

Unit III
ATMEGA and ARM Processors

Introduction to AVR family of microcontrollers, Introduction to ATMEGA 8 and AT90S1200 microcontrollers, architecture and pin diagram of the ATMEGA 8 and AT90S1200 microcontrollers, programming of ATMEGA 8 using c and assembly languages, interfacing of ATMEGA8 to other modules.An introduction to ARM processors, ARM architecture, ARM Instructions set, thumb instructions set, design issues, c and assembly programming in ARM, architectural support for system development, optimized primitives, exception and interrupt handling, caches memory protecting units, memory management units, embedded operating system using in the ARM

Introduction to Smart phone **11+4 Hours**

Unit IV
ARM Processor Cores

Introduction to ARM processors cores, embedded ARM applications, architecture, instruction set, programming using assembly and c languages of ARM 7TDMI and ARM9TDMI processors, interfacing ARM 7TDMI and ARM9TDMI processors to other devices, applications of ARM 7TDMI and ARM9TDMI processors.Introduction to embedded processors, ISA architecture models, internal processor design, processor performance, configurability features, processor architecture, instruction set, programming of embedded processors (power PC processor, Micro blaze processor and Nios processor) and interfacing to other modules.

Basic concepts of Embedded boards **11+3 Hours**

Total: 45+15 Hours

Textbook (s)

1. Ajay V Deshmukh, Microcontrollers-Theory and Applications, TMH Publication, 2005.
2. Dhananjay V Gadre, Programming and customising the AVR microcontroller, TMH Publications, 2000.
3. Stephen B Furber, ARM system – on chip Architecture, Pearson Publishers second edition, 2000.
4. Tammy Noergaard, Embedded systems architecture, Elsevier Publications, 2006.

Reference (s)

1. V. Udayashankara, M. S. Mallikarjuna Swamy 8051 Microcontroller-Hardware, Software and applications, TMH Publications, 2005.
2. lucio Bi Jasio ,PIC microcontrollers, Newnes Publishers, 2008.

21VLS005 DSP Processors and Architectures (Elective – II)

4 0 0 4

Course Outcomes

1. Apply DFT and FFT algorithms for DSP application
2. Apply the number format, dynamic range and various sources of errors in DSP system
3. Implement application programs on a DSP processor
4. Make use of TMS processors to implement various DSP algorithms
5. Utilize TMS320C54XX DSP device to implement FFT algorithms
6. Explain Interfaceing of various peripherals devices with TMS processors

Unit I
Introduction to Digital Signal Procesing

Introduction, The sampling process, discrete time sequences. Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), linear time invariant systems, Decimation and interpolation. Number formats for signals and coefficients in DSP systems, Dynamic Range and Precision, Sources of error in DSP implementations, DSP Computational errors, Compensating filter.

Finite wordlength effects in digital filters

10+3 Hours

Unit II
Architectural features of DSP devices

Basic Architectural features, DSP Computational Building Blocks, Bus Architecture and Memory, Data Addressing Capabilities, Address Generation Unit, Programmability and Program Execution, Speed Issues, Features for External interfacing. Hardware looping, Interrupts, Stacks, Relative Branch support, Pipelining and Performance, Pipeline Depth, Interlocking, Branching effects, Interrupt effects and Pipeline Programming models.

Multiple acess memory, multiported memory

12+4 Hours

Unit III
TMS320C54XX Digital Signal Processors

Data Addressing modes of TMS320C54XX DSPs, Data Addressing modes of TMS320C54XX Processors, Memory space of TMS320C54XX Processors, Program Control, TMS320C54XX instructions and Programming, On-Chip Peripherals. Introduction, Types of interrupts, Pipeline Operation of TMS320C54XX Processors.

TMS320C54X Assembly language instructions

13+4 Hours

Unit IV
Implementation of FFT Algorithms and Interfacing techniques

An FFT Algorithm for DFT Computation, A Butterfly Computation, Overflow and scaling, Bit-Reversed index generation, An 8-Point FFT implementation on the TMS320C54XX, Computation of the signal spectrum. Memory space organization, External bus interfacing signals, Memory interface, Parallel I/O interface, Programmed I/O, Interrupts and I/O, Direct memory access (DMA).

A CODEC Interface circuit **10+4 hours**

Total: 45+15 Hours

Textbook (s)

1. Avtar Singh and S. Srinivasan, Digital Signal Processing, Thomson Publications, 2004.
2. Lapsley, DSP Processor Fundamentals, Architectures & Features, S. Chand & Co, 2000.

Reference (s)

1. B. Venkata Ramani and M. Bhaskar, Digital Signal Processors, Architecture, Programming and Applications, TMH, 2004.
2. Jonatham Stein, Digital Signal Processing, John Wiley, 2005.

21VLS006 VLSI Signal processing (Elective – II)

4 0 0 4

Course Outcomes

1. Design parallel processors in VLSI systems
2. Illustrate the register minimization using the retiming, unfolding & folding concepts.
3. Design systolic architecture using canonical mapping and generalized mapping
4. Design parallel bit circuits
5. Apply FFT for signal processing applications
6. Demonstrate the DSP processors

Unit I
Introduction to the VLSI Signal Processing, Pipelining and Parallel Processing

Typical Signal Processing Algorithms, Overview of VLSI Architectures, Representations of DSP Algorithms. Introduction, Data Flow Graph Representation, Loop bound and Iteration Bound, Algorithms for computing Iteration bound, Pipelining of FIR filters, Parallel Processing,

Pipelining and parallel processing for low power

11+4 Hours

Unit II
Retiming, Unfolding and Folding

Definitions and Properties, Solving systems of inequalities, Retiming techniques. Unfolding Algorithm, Properties of unfolding, Critical Path, Unfolding and Retiming, Folding Transformation, Register Minimization techniques. Register Minimization in folded architectures.

Folding of multirate systems

11+4 Hours

Unit III
Systolic Architecture Designand Arithmetic Components

Matrix Operations and 2D Systolic Array Design, Parallel Algorithm Expressions, Canonical Mapping Methodology, Generalized Mapping.Parallel bit circuits: Carry-Look ahead addition, Prefix Computations, Carry-Save Addition, Multiplication.

Systolic design for space representations containing delays **11+4 Hours**

Unit IV
Fast Convolution and Programmable Digital Signal Processors

Introduction, Cook-Toom algorithm, Winogard algorithm, Iterated Convolution and Cyclic convolution. Important Features, DSP Processors for Mobile and Wireless Communications, Processors for Multidimensional Signal Processing.

Design of Fast convolution algorithm by inspection

12+3 Hours
Total: 45+15 Hours

Textbook (s)

1. K. K. Parhi, VLSI Digital Signal Processing Systems, Design and Implementation, John Wiley, 1999.

Reference (s)

1. S.Y.Kung, VLSI Array Processors, Prentice-Hall, 1988

21VLS007 Embedded System Design (Elective – III)

4 0 0 4

Course Outcomes

1. Design of Embedded system and they can implement in various real time applications.
2. Analyze the complete embedded life cycle in designing of embedded systems (hardware and software).
3. Explain various external devices to embedded systems.
4. Identify the proper driver configuration for embedded system in various applications like automated washing machines.
5. Select suitable operating system for real time embedded systems based on application requirement.
6. Build a new proposed system and test the design.

Unit I
Introduction

An Embedded System-Definition, Examples, Current Technologies, Integration in system Design, Embedded system design flow, hardware design concepts, software development, processor in an embedded system and other hardware units, introduction to processor based embedded system design concepts.

Design metrices of Embedded systems design. **6+2 Hours**

Unit II
Embedded Hardware

Embedded hardware building blocks, Embedded Processors – ISA architecture models, Internal processor design, processor performance, Board Memor, ROM, RAM, Auxiliary Memory, Memory Management of External Memory, Board Memory and performance. Embedded board Input / output – Serial versus Parallel I/O, interfacing the I/O components, I/O components and performance, Board buses – Bus arbitration and timing, Integrating the Bus with other board components, Bus performance.

External communication interface. **15+5 Hours**

Unit III
Embedded Software

Device drivers, Device Drivers for interrupt-Handling, Memory device drivers, On-board bus device drivers, Board I/O drivers, Explanation about above drivers with suitable examples.Embedded operating systems – Multitasking and process Management, Memory Management, I/O and file system management, OS standards example – POSIX, OS performance guidelines, Board support packages, Middleware and Application Software
– Middle ware, Middleware examples, Application layer software examples.

High level language programming **12+4 Hours**

Unit IV
Embedded System Design, Development, Implementation and Testing

Embedded system design and development lifecycle model, creating an embedded system architecture, introduction to embedded software development process and tools- Host and Target machines, linking and locating software, Getting embedded software into the target system, issues in Hardware-Software design and co-design. Implementing the design-The main software utility tool, CAD and the hardware, Translation tools, Debugging tools, testing on host machine, simulators, Laboratory tools, System Boot-Up.

Design aided tools **12+4 Hours**

Total: 45+15 Hours

Textbook (s)

1. Tammy Noergaard,Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers, Elsevier(Singapore) Pte.Ltd. Publications, 2005.

Reference (s)

1. Frank Vahid, Tony D. Givargis, Embedded system Design: A Unified Hardware/Software Introduction,John Wily & Sons Inc., 2002.
2. Peter Marwedel,Embedded System Design, Science Publishers, 2007.
3. Arnold S Burger, Embedded System Design, CMP.booksusa 2002.
4. Rajkamal,Embedded Systems: Architecture, Programming and Design, TMH Publications, Second Edition, 2008.

21VLS008 System Verilog Programming for Verification (Elective – III)

4 0 0 4

Course Outcomes

1. Interpret the verification guidelines and data types
2. Execute the programs using Procedural Statements and Routines.
3. Demonstrate the System Verilog constructs through simulations
4. Explain the basic OOPs concepts.
5. Organize the design modules in SV test bench.
6. Execute the ASIC verification using SV with help of EDA tools

Unit I

Introduction to Verification and Data Types

Introduction, The Verification Process, The Verification Plan, White-Box Verification, Grey-Box Verification, Black-Box Verification, The Verification Methodology, Basic Test-bench Functionality, Directed Testing, Methodology Basics, Constrained-Random Stimulus, Functional Coverage, Test-bench Components, Layered Test-bench, Simulation Environment Phases, Maximum Code Reuse, Introduction to data types, Built-in Data Types, Fixed-Size Arrays, Dynamic Arrays, Queues, Associative Arrays, Linked Lists, Array Methods, Creating New Types with typedef, Creating User-Defined Structures, Enumerated Types, Constants, Strings, Net Types.

String methods, operators

12+4 Hours

Unit II

Tasks and Functions

Introduction, Procedural Statements, Task and Function Overview, Tasks, Functions, and Void Functions, Routine Arguments, Returning from a Routine, Local Data Storage, Time Values.

Enumerated types, repeat and forever

11+3 Hours

Unit III

Basics of OOP for System Verilog

Introduction, OOP Terminology, Creating New Objects, Object Deallocation, Using Objects, Static Variables vs. Global Variables, Class Routines, Defining Routines Outside of the Class, Scoping Rules, Using One-Class Inside Another, Understanding Dynamic Objects, Copying Objects, Building a Test-bench.

Nested–Inner Class & Anonymous Classes–Generic Class Types

11+4 Hours

Unit IV

Integration of the Test bench and Design

Introduction, Separating the Test-bench and Design, The Interface Construct, Stimulus Timing, Interface Driving and Sampling, Top-Level Scope, Program – Module Interactions, System Verilog Assertions, The Four-Port ATM Router.

Current Trends in Testing and Verification: Advanced verification methodologies: UVM basic levels. EDA Development Environments – case study.

Interprocess communication: semaphore, mailboxes, and event

11+4 Hours

Total: 45+30 Hours

Textbook (s)

1. Chris Spear, "System Verilog for Verification: A Guide to Learning the Test bench Language Features", Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2006
2. Donald Thomas, "Logic Design and Verification Using System Verilog", Create Space Independent Publishing Platform, 2014.

3. Bergeron, J. "Writing Testbenches Using SystemVerilog Springer." 0-387-29221-7, BusinessMedia (2006).

Reference (s)

1. Language Reference Manual for System Verilog

21VLS009 IoT system Design (Elective – III)

4 0 0 4

Course Outcomes

1. Understand the different target boards
2. Summarize the values chains Perspective of M2M to IoT
3. Implement the state of the Architecture of an IoT
4. Demonstrate knowledge and understanding the security and ethical issues of an IOT
5. Compare IOT Applications in Industrial & real world
6. Demonstrate knowledge and understanding the security and ethical issues of an IoT

Unit I
Fundamentals of IoT
Evolution of Internet of Things, Enabling Technologies, IoT Architectures,oneM2M, IoT World Forum (IoTWF) and Alternative IoT models, Simplified IoT Architecture and Core IoT Functional Stack, Fog, Edge and Cloud in IoT, Functional blocks of an IoT ecosystem, Sensors, Actuators, Smart Objects and Connecting Smart Objects. IoT Platform overview: Overview of IoT supported Hardware platforms such as: Raspberry pi, ARM Cortex Processors,
Arduino and Intel Galileo boards **12+4 Hours**

Unit II
IoT Protocols
IT Access Technologies: Physical and MAC layers, topology and Security of IEEE 802.15.4, 802.15.4g, 802.15.4e, 1901.2a, 802.11ah and Lora WAN, Network Layer: IP versions, Constrained Nodes and Constrained Networks, Optimizing IP for IoT: From 6LoWPAN to 6Lo, Routing over Low Power and Lossy Networks, Application Transport Methods: Supervisory Control and Data Acquisition, Application Layer Protocols: CoAP and MQTT
LoRa Wan and 6lowpan **11+3 Hours**

Unit III
Design and Development
Design Methodology, Embedded computing logic, Microcontroller, System on Chips, IoT system building blocks, Arduino, Board details, IDE programming, Raspberry Pi, Interfaces and Raspberry Pi with Python Programming, Security and Privacy Requirements
Atmega 328, load store instructions of ARM processor **12+4 Hours**

Unit IV
Data Analytics and Supporting Services
Structured Vs Unstructured Data and Data in Motion Vs Data in Rest, Role of Machine Learning – No SQL Databases, Hadoop Ecosystem, Apache Kafka, Apache Spark, Edge Streaming Analytics and Network Analytics, Xively Cloud for IoT, Python Web Application Framework, Django, AWS for IoT, System Management with NETCONF-YANG
IoT for smart grid, smart healthcare **10+4 Hours**
Total: 45+15 Hours

Textbook (s)

1. IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, Cisco Press, 2017
2. Rajkamal, Internet of things: Architecture and design principles, TMH Publication, 2017

Reference (s)

1. Internet of Things – A hands-on approach, ArshdeepBahga, Vijay Madisetti, Universities Press, 2015
2. The Internet of Things – Key applications and Protocols, Olivier Hersent, David Boswarthick, OmarElloumi and Wiley, 2012.
3. "From Machine-to-Machine to the Internet of Things – Introduction to a New Age of Intelligence",Jan Ho¨ ller, VlasiosTsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand. David Boyle and Elsevier, 2014.

21VLS010 Design of Fault Tolerant System (Elective – IV)

4 0 0 4

Course Outcomes

1. Illustrate fault tolerant digital system
2. Design and minimize the error for a given system
3. Explain the self-checking circuits.
4. Design and test combinational circuits.
5. Design and test a sequential circuit.
6. Design an automatic testing system using BIST for basic sequential circuits

Unit I
Basic Concepts and Fault Tolerant Design

Reliability concepts, Failure & Faults, Reliability and failure rate, Relation between reliability and meantime between failure, Maintainability and Availability, Reliability of series, Parallel and Parallel-Series combinational circuits.'

Basic concepts – Static, dynamic, hybrid, Triple Modular Redundant System, Self purging redundancy, Sift out redundancy (SMR), SMR Configuration, Use of error correcting code, Time redundancy and software redundancy.

Fault Modelling **12+3 Hours**

Unit II
Self Checking Circuits and Fail Safe Design

Basic concepts of Self checking circuits, Design of Totally Self Checking checker, Checkers using m out of n codes, Berger code, Low cost residue code.

Strongly fault secure circuits, fail-safe design of sequential circuits using partition theory and Berger code, totally self-checking PLA design.

Error Correcting Codes **11+4 Hours**

Unit III
Design for Testability for Combinational Circuits

Basic concepts of testability, controllability and observability, the Reed Muller's expansion technique, OR-AND-OR design, use of control and syndrome testable design. Theory and operation of LFSR, LFSR as Signature analyzer, Multiple-input Signature Register.

Testing for Single Stuck Faults. PLA Testing **11+4 Hours**

Unit IV
Design for Testability for Sequential Circuits and Built in Self Test

Controllability and observability by means of scan register, Storage cells for scan design, classic scan design, Level Sensitive Scan Design (LSSD).

BIST concepts, Test pattern generation for BIST exhaustive testing, pseudorandom testing, pseudo exhaustive testing, constant weight patterns, Generic offline BIST architecture.

Logic-Level Diagnosis and System-Level Diagnosis **11+4**

Hours Total: 45+15 Hours

Textbook (s)

1. Parag K. Lala, Fault Tolerant & Fault Testable Hardware Design, PHI, 1985
2. M. Abramovili, M. A. Breues, A. D. Friedman, Digital Systems Testing and Testable Design, Jaico publications, 1997.

Reference (s)

1. Israel Koren, C. Manikrishna, Morgan Kaufmann, Fault tolerant systems, 2007.

21VLS011 Embedded Networking (Elective – IV)

4 0 0 4

Course Outcomes

1. Analyze embedded Networking applications
2. Apply the concept of CAN in embedded applications like automotives
3. Test for the designed embedded networks
4. Design embedded application using CAN open standard
5. Make use of micro CAN open for designing networking applications
6. Explain different wireless communication systems

Unit I
Embedded Networking

Code requirements, Communication requirements, Introduction to CAN open, CAN open standard.Object directory, Electronic Data Sheets & Device ,Configuration files, Service Data Objectives, Network management CAN open messages, Device profile encoder.

Psedo code of CAN

11+4 Hours

Unit II
CAN Open Configuration

Evaluating system requirements choosing devices and tools, Configuring single devices. Overall network configuration, Network simulation, Network Commissioning , Advanced features and testing.

Simple application using CAN

12+4 Hours

Unit III
Controller Area Network

Underlying Technology CAN Overview, Selecting a CAN Controller, CAN development tools. Implementing CAN open Communication layout and requirements – Comparison of implementation methods.

Simulation using CAN

11+4 Hours

Unit IV
Micro CAN open

CAN open source code, Conformance test, Entire design life cycle.Physical layer, Data types, Object dictionary, Communication object identifiers, Emerging objects, Node states.

Exploring micro CAN open **11+3hours**

Total 45+15 hours

Textbook (s)

1. Glaf P. Feiffer, Andrew Ayre and Christian Keyold, Embedded Networking with CAN and CAN open Embedded System Academy 2005.

Reference (s)

1. Gregory J. Pottie, William J. Kaiser, Principles of Embedded Networked Systems Design, Cambridge University Press, 2nd Edition, 2005.

21VLS012 Low power VLSI design (Elective – IV)

4 0 0 4

Course Outcomes

1. Design and estima1te the power of different circuits
2. Apply Bi-CMOS and advanced Bi-CMOS circuits for different applications
3. Apply different isolation techniques in chip design
4. Design low power memory elements
5. Demonstrate the Latches and Flip-Flops in detail
6. Demonstrate ESD free BiCMOS circuits

Unit I
Introduction to Low Power Low Volatge

Low- voltage low power design, limitations, Silicon-on-Insulator. Bi CMOS processes. Integration and Isolation considerations, Integrated Analog/Digital CMOS Process.

Future trends and directions of CMOS/BICMOS processes **11+4 Hours**

Unit II
MOS/BiCMOS Process Technology and Integration

Conventional CMOS and BiCMOS logic gates. Performance evaluation. Comparison of advanced BiCMOS Digital circuits. ESD-free Bi CMOS Digital circuit.

Bootstrapped –Type Bicmos **11+4 Hours**

Unit III
Design and Test of Low Voltage CMOS Circuits and Low power static RAM architectures

Introduction , circuit design style , Leakage current in deep sub micrometer transistors , Deep sub micrometer device design issues, Low voltage circuit design issues.Introduction , organization of a static RAM , MOS static RAM memory cell , banked organization of SRAMs, Reducing voltage swings on bit lines, reducing power in the write driver circuits , Reducing power in sense amplifier circuits.

Testing deep sub micrometer ICs with elevated intrinsic leakage **11+4 Hours**

Unit IV
Evolution of latches and Low energy computing using energy recovery techniques

Evolution of Latches and Flip flops-quality measures for latches and Flip flops, Design perspective. Energy dissipation in transistor channel using an RC model, Energy recovery circuit design.

Quality measures of latches and flipflops **12+3 Hours**

Total:45+15 Hours

Textbook (s)

1. Yeo Rofail/ Gohl (3 Authors), CMOS/BiCMOS ULSI low voltage, low power, Pearson Education Asia 1st Indian reprint, 2002
2. Kaushik Roy, Sharat Prasad, Low Power CMOS VLSI Circuit Design, Wiley publishers 2000.

Reference (s)

1. Gary Yeap, Practical VLSI Design, Springer publishers,1997.

21VLS013 CPLD and FPGA Architectures & Applications (Elective – V)

4 0 0 4

Course Outcomes

1. Make use of the routing architecture, design flow, technology mapping for FPGAs
2. Design Xilinx XC4000 & ALTERA's FLEX 8000/10000 AT &T ORCA's in FPGA through mapping.
3. Illustrate state machine charts using PAL in FSMs.
4. Design Controller, data path design, Functional partition using digital design tools.
5. Design the system level design for various architectures using mentor graphics EDA tools
6. Demonstrate different case studies of CPLDs and FPGAs.

Unit I
PLDs and CPLDs

ROM, PLA, PAL, CPLD, FPGA – Features, Architectures, Programming,Applications and mplementation of MSI circuits using Programmable logic Devices.Complex Programmable Logic Devices: Altera series – Max 000/7000 series and Altera FLEX logic-10000 series CPLD, AMD's- CPLD (Mach 1 to 5), Cypress FLASH
370 Device technology, Lattice pLSI's architectures – 3000 series – Speed performance and in system programmability.

Programming technologies

11+4 Hours

Unit II
FPGAs and Finite State Machines (FSM)

Field Programmable Gate Arrays- Logic blocks, routing architecture, design flow, technology mapping for FPGAs, Case studies Xilinx XC4000 & ALTERA's FLEX 8000/10000 AT &T ORCA's(Optimized Reconfigurable Cell Array): ACTEL's ACT-1,2,3.Top Down Design, State Transition Table, State assignments for FPGAs, Realization of state machine charts using PAL, Alternative realization for state machine charts using microprogramming, linked state machine

Linked state machines

11+4 Hours

Unit III
FSM Architectures and Design Methods

Architectures Centered around non registered PLDs, Design of state machines centered around shift registers, One_Hot state machine, Petrinets for state machines-Basic concepts and properties.One –hot design method, Use of ASMs in one-hot design method, Applications of one-hot design method, Extended Petri-nets for parallel controllers, Meta Stability, Synchronization, Complex design using shift registers.

Alternative Realization for State Machine Chart using Microprogramming.

11+4 hours

Unit IV
System Level Design and Case studies

Controller, data path designing, Functional partition, Digital front end digital design tools for FPGAs & ASICs, System level design using mentor graphics EDA tool Design considerations using CPLDs and FPGAs of parallel adder cell, parallel adder sequential circuits, counters, multiplexers, parallel controllers.

Extended Petri-nets for parallel controllers

12+3 Hours
Total 45+15 Hours

Textbook (s)

1. S. Trimberger, Edr,Field Programmable Gate Array Technology , Kluwer Academic Publications,1994.
2. Richard F. Tinder, Engineering Digital Design, 2nd Edition, Academic press., 2011
3. Fundamentals of logic design-Charles H. Roth, 4th Edition Jaico Publishing House, 2004

Reference (s)

1. P. K. Chan & S. Mourad, Digital Design Using Field Programmable Gate Array, Prentice Hall, 1994.
2. S. Brown, R. J. Francis, J. Rose, Z.G.Vranesic, Field programmable gate array, BSP, 2007.

21VLS014 Hardware Software Co-design (Elective – V)

4 0 0 4

Course Outcomes

1. Analyze any embedded system's hardware and software design issues
2. Choose different Co-design Models, Algorithms and methodology etc., for embedded system design
3. Apply Embedded Software Development tools, Compilation Techniques for embedded applications
4. Test for the hardware and software individually
5. Explain System-level performance modeling, low-level performance modeling and High-level synthesis
6. Illustrate integration process of hardware and firmware

Unit I
Introduction to Co Design

Co- Design Models, Architectures, Languages, A Generic Co-Design Methodology, Hardware – Software Synthesis Algorithms : Hardware – Software Partitioning, Distributed System Co-Synthesis.Prototyping and Emulation techniques, Prototyping and Emulation Environments, Future Developments in Emulation and Prototyping, Architecture Specialization Techniques.

Waterfall model of co Design **11+4 Hours**

Unit II
System Design

System Communication infrastructure, Target Architectures and Application System Classes, Architectures for Control Dominated System and Data – Dominated Systems.Modern Embedded Architectures, Embedded Software Development needs, Compilation Technologies, Practical Consideration in a compiler Development Environment. Design specification and verification: Design, Co- Design, The Co- Design Computational Model, Concurrency, coordinating Concurrent Computations, interfacing components

Integrated Development Environment **12+4 Hours**

Unit III
Design Tools

Design Verification, Implementation Verification, Verification Tools, Interface Verification.System – Level Specification, Design representation for system level synthesis, System level specification Languages, Heterogeneous Specifications and Multi Language Co – Simulation. The cosyma system and Lycos system.

Programming languages

11+3 Hours

Unit IV
Firmware Design

System-level performance modeling vs. low-level performance modeling - Modeling of execution speed (system latency) and energy consumption for hardware and software - Estimation of memory requirements. High-level synthesis - behavioral specification of hardware, module set allocation, resource binding, operation scheduling, controller synthesis.

Cross Compiler

11+4 Hours
Total: 45+15 Hours

Textbook (s)

1. Kluwer, Hardware /Software Co–Design Principles and Practice, Academic Publishers, 2nd printing 2000.

Reference (s)

1. Arnold S. Berger, Embedded System Design, CMP books, USA 2002.
2. Wayne Wolf, Computers as Components: Principles of Embedded Computer Systems Design, Morgan Kaufman Publishers, 2005.

21VLS015 System Modeling & Simulation (Elective – V)

4 0 0 4

Course Outcomes

1. Analyze the given system or problem
2. Design a model to represent the system or problem
3. Make use of simulation for the designed model
4. Develop simulation models for time and event driven systems
5. Design simulation models for given system using petri nets
6. Analyze the queuing systems and Optimize the model to get optimum performance

Unit I
Introduction to Simulation

Basic Simulation Modeling, Systems, Models and Simulation, Alternative approach to modeling and simulation.Guidelines for determining levels of model detail, Techniques for increasing model validity and credibility.Classification of Simulation Software,Desirable Software features, General purpose simulation packages – Arena, Extend and others,Object Oriented Simulation.

Modern Simulation software's features and analysis

12+3 Hours

Unit II
Discrete Event Simulation

Discrete Event Simulation, Simulation of Single server queingsystem, Simulation diagrams, Queing theory, simulating queing systems, Types of Queues, Multiple servers,Simulation of Inventory System.

The OMNeT++ discrete event simulation system

11+4 Hours

Unit III
Building Simulation Models

Modeling input signals, delays, System Integration, Linear Systems, Motion Control models, numerical experimentation,System identification, Searches, Alpha/beta trackers, multidimensional optimization and modeling and simulation methodology.

Advanced multidimensional optimization and modeling

11+4 Hours

Unit IV
State Machines Modeing and Simulation

Disturbance signals, state machines, petri nets & analysis, System encapsulation,Probabilistic systems, Discrete Time Markov processes, Random walks, Poisson processes, the exponential distribution, simulating a poison process, Continuous – Time Markov processes.

Petri nets Design and Simulation

11+4 Hours
Total: 45+15 Hours

Textbook (s)

1. Frank L. Severance, System Modeling & Simulation, An introduction, John Wiley & Sons, 2001.
2. Averill M. Law, W. David Kelton, Simulation Modeling and Analysis, TMH, 3rd Edition, 2003.

Reference (s)

1. Geoffery Gordon, Systems Simulation, PHI, 1978.

21CSE203 Soft Computing Techniques (Elective – VI)

4 0 0 4

Course Outcomes

1. Identify and describe soft computing techniques and their roles in building intelligent machines.
2. Identify and apply supervised learning methods to different neural networks models for pattern classification and regression problems.
3. Evaluate and apply unsupervised learning methods to different neural networks models for storing and pattern classification.
4. Evaluate and implement the Support Vector Machines to classify objects in real time applications.
5. Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems.
6. Evaluate and compare solutions by various soft computing approaches for a given problem.

Unit I
Introduction to Soft Computing and Artificial Neural Networks

Introduction to soft computing, Fuzzy logic, Neural Networks and Evolutionary Computing, Approximations of Multivariate functions, Non – linear Error surface and optimization.

Introduction to ANN, Basic models of ANN, important terminologies, Basic Learning Laws, Supervised Learning Networks, Perceptron Networks, Adaptive Linear Neuron, Back propagation Network Radial basis function network and Hopfield Networks.

Bi-directional associative memories. **11+3 Hours**

Unit II
Unsupervised Learning Networks and Introduction to Classical Sets and Fuzzy Sets

Introduction, Fixed Weight Competitive Nets, Maxnet, Hamming Network, Kohonen Self-Organizing Feature Maps, Learning Vector Quantization, Counter Propagation Networks, Adaptive Resonance Theory Networks. Special Networks - Introduction to various networks.

Crisp Sets and Fuzzy Sets - operations. Classical Relations and Fuzzy Relations - Cardinality, Properties and composition. Tolerance and equivalence relations. Membership functions- Features, Fuzzification, membership value assignments, Defuzzification.

Simulated annealing network. **11+4 Hours**

Unit III
Fuzzy Logic and Genetic Algorithm

Classical& Fuzzy logic, Operations, Boolean Logic, Multivalued Logics, Fuzzy Rule Base and Approximate Reasoning ,Fuzzy Decision making ,Fuzzy Logic Control Systems.

Introduction to GA, Traditional Optimization and search techniques, Search space, Operators: Encoding, Selection, Crossover and Mutation. Stopping Condition of GA.

Fuzzy arithmetic and Fuzzy measures **11+4 Hours**

Unit IV
Support Vector Machine and Applications of Soft Computing

Introduction, optimal hyper plane for linearly separable pattern, linear classifier, nonlinear classifier problem, optimal plane for non-separable pattern, example XOR problem, and support vector machine for non-linear regression, summary and discussion.

A fusion Approach of Multispectral Images with SAR Image for flood area analysis, Optimization of TSP using GA Approach and GA-Fuzzy system for Control of flexible Robots.

Hybrid soft computing techniques **12+4 Hours**

Total: 45+15 Hours

Textbook (s)

1. S N Sivanandam, S N Deepa, Principles of Soft Computing, Wiley India, 2011
2. V. Kecman, Learning and Soft computing, Pearson Education, India

Reference (s)

1. Fakhreddine O Karray, Clarence D Silva, Soft Computing and Intelligent System Design, Pearson Edition, 2004.
2. Guanrong Chen, Trung Tat Pham, Chapman & Hall/CRC, Introduction to Fuzzy Systems, 2009.
3. S. Haykins, Neural networks: a comprehensive foundation, Pearson Education, India.

21VLS016 Memory Architectures (Elective – VI)

4 0 0 4

Course Outcomes

1. Illustrate the RAM technologies, the architecture of SRAM
2. Interpret the architectures of DRAM, DRAM controllers, and their applications
3. Demonstrate the applications of SRAMs and DRAMs
4. Demonstrate the operation of Non-Volatile Memories
5. Abstract the Semiconductor Memory Reliability and Radiation Effects
6. Abstract the process and design issues involved in Semiconductor Memories

Unit 1

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

12+4 Hours

Unit 2 DRAM Technologies

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

12+4 Hours

Unit 3

Non-Volatile Memories

Masked ROMs, PROMs, Bipolar & CMOS PROM, EEPROMs, Floating Gate EPROM Cell, OTP EPROM, EEPROMs, Non-volatile SRAM, Flash Memories.

12+4 Hours

Unit 4

Semiconductor Memory Reliability and Radiation Effects

Semiconductor Memory Reliability and Radiation Effects: General Reliability Issues,
RAM Failure Modes and Mechanism, Non-volatile Memory, Radiation Effects, SEP, Radiation Hardening Techniques. Process and Design Issues, Radiation Hardened Memory Characteristics, Radiation Hardness Assurance, Testing, Memory repair

12+4 Hours

Total:45+15 Hours

Text Books:

1. Ashok K Sharma, "Advanced Semiconductor Memories: Architectures, Designs and Applications", Wiley Inter science
2. KiyooItoh, "VLSI memory chip design", Springer International Edition
3. Wang, Laung-Terng, Cheng-Wen Wu, and Xiaoqing Wen. "VLSI test principles and architectures: Design for testability", Elsevier, 2006.

Reference Books:

1. Ashok K Sharma," Semiconductor Memories: Technology, Testing and Reliability , PHI

21VLS017 Programming Languages for Embedded Systems (Elective – VI)

4 0 0 4

Course Outcomes

1. Understand the embedded C application of moderate complexity.
2. Summarize the features of Object Oriented Programming.
3. Develop and analyze algorithms in C++.
4. Outline Overloading and Inheritance concepts in embedded firmware development.
5. Differentiate interpreted languages from compiled languages.
6. Demonstrate the significance of Scripting languages in firmware development.

Unit I

Fundamentals of Programming in Embedded C

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.,

Embedded Software Development Cycle and Methods (Waterfall, Agile) **10+4 Hours**

Unit II

Object Oriented Programming

Introduction to procedural, modular, object-oriented and generic programming techniques, Limitations of procedural programming, objects, classes, data members, methods, data encapsulation, data abstraction and information hiding, inheritance, polymorphism, CPP Programming: „cin", „cout", formatting and I/O manipulators, new and delete operators, Defining a class, data members and methods, „this" pointer, constructors, destructors, friend function,

Dynamic memory allocation **11+3 Hours**

Unit III

Overloading and Inheritance

Need of operator overloading, overloading the assignment, Overloading using friends, type conversions, single inheritance, base and derived classes, friend Classes, types of inheritance, hybrid inheritance, multiple inheritance, virtual base class.

Polymorphism, virtual functions. **12+4 Hours**

Unit IV

Templates, Exception Handling and Scripting Languages

Function template and class template, member function templates and template arguments, syntax for exception handling code: try-catch- throw, Multiple Exceptions. Overview of Scripting Languages – PERL, CGI, VB Script, Java Script. PERL: Operators, Statements Pattern Matching etc. Data Structures, Modules, Objects, Tied Variables, Inter process Communication Threads.

Compilation & Line Interfacing **12+4 Hours**

Total: 45+15 Hours

Textbook (s)

1. Michael J. Pont , "Embedded C", Pearson Education, 2nd Edition, 2008
2. Randal L. Schwartz, "Learning Perl", O"Reilly Publications, 6th Edition 2011

Reference (s)

1. A. Michael Berman, "Data structures via C++", Oxford University Press, 2002
2. Robert Sedgewick, "Algorithms in C++", Addison Wesley Publishing Company, 1999
3. Abraham Silberschatz, Peter B, Greg Gagne, "Operating System Concepts", John Willey & Sons, 2005.