SCHEME OF TEACHING AND EXAMINATION
M.TECH. – ELECTRONICS

II SEMESTER

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Subject Code</th>
<th>Name of the Subject</th>
<th>Teaching hours/week</th>
<th>Duration of Exam in Hours</th>
<th>Marks for</th>
<th>Total Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lecture</td>
<td>Practical</td>
<td>Tutorial</td>
<td>I.A.</td>
</tr>
<tr>
<td>10LEL21</td>
<td>10EC009</td>
<td>Advances in VLSI Design</td>
<td>4</td>
<td>-</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>10LEL22</td>
<td>10EC054</td>
<td>Nanoelectronics</td>
<td>4</td>
<td>-</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>10LEL23</td>
<td>10EC077</td>
<td>Synthesis &amp; Optimization of Digital Circuits</td>
<td>4</td>
<td>2</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>10LEL24</td>
<td>10EC118</td>
<td>Advanced Embedded System</td>
<td>4</td>
<td>2</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>10LEL25</td>
<td>10ECxxx</td>
<td>Elective-II (10LEL25x)</td>
<td>4</td>
<td>-</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>10LEL26</td>
<td>10EC0921</td>
<td>Mini – Project/Seminar</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td>50$</td>
</tr>
</tbody>
</table>

Project Phase I (6 week Duration) should start between II Semester and III Semester, after availing a vacation of 2 weeks. This will be evaluated during III semester.

Total 20 07 06 15 300 500 800

Practical will be evaluated 25 marks and internal assessment for 25 marks. Lab journals should be maintained.

Assignments/seminar will be evaluated for 25 marks and internal assessment for 25 marks. Record of Assignments/seminar should be maintained.

Mini projet should be done individually and is assessed for 25 marks. Seminar on Miniproject will be assessed for 25 marks.

ELECTIVE – II

| Course Code | Subject Code | Subject Name         | | | |
|-------------|--------------|----------------------| | | |
| 10LEL251    | 10EC012      | ASIC Design          | | | |
| 10LEL252    | 10EC075      | Speech & Audio Processing | | | |
| 10LEL253    | 10EC126      | Real Time Operating System | | | |

SEMINER II

ADVANCES IN VLSI DESIGN

Subject Code : 10EC009  IA Marks : 50
No. of Lecture Hours /week : 04  Exam Hours : 03
Total no. of Lecture Hours : 52  Exam Marks : 100

Review of MOS Circuits: MOS and CMOS static plots, switches, comparison between CMOS and BI - CMOS.

MESFETS: MESFET and MODFET operations, quantitative description of MESFETS.

MIS Structures and MOSFETS: MIS systems in equilibrium, under bias, small signal operation of MESFETS and MOSFETS.

Short Channel Effects and Challenges to CMOS: Short channel effects, scaling theory, processing challenges to further CMOS miniaturization

Beyond CMOS: Evolutionary advances beyond CMOS, carbon Nano tubes, conventional vs. tactile computing, computing, molecular and biological computing Mole electronics-molecular Diode and diode- diode logic .Defect tolerant computing,
Super Buffers, Bi-CMOS and Steering Logic: Introduction, RC delay lines, super buffers- An NMOS super buffer, tri state super buffer and pad drivers, CMOS super buffers, Dynamic ratio less inverters, large capacitive loads, pass logic, designing of transistor logic, General functional blocks - NMOS and CMOS functional blocks.


System Design: CMOS design methods, structured design methods, Strategies encompassing hierarchy, regularity, modularity & locality, CMOS Chip design Options, programmable logic, Programmable inter connect, programmable structure, Gate arrays standard cell approach, Full custom Design.

REFERENCE BOOKS:


NANOELECTRONICS

Subject Code : 10EC054
IA Marks : 50
No. of Lecture Hours /week : 04
Exam Hours : 03
Total no. of Lecture Hours : 52
Exam Marks : 100

Shrink-down Approaches: Introduction, CMOS Scaling, The nanoscale MOSFET, Finfets, Vertical MOSFETs, limits to scaling, system integration limits (interconnect issues etc.), Resonant Tunneling Transistors, Single electron transistors, new storage, optoelectronic, and spintronics devices.

Atoms-up Approaches: Molecular electronics involving single molecules as electronic devices, transport in molecular structures, molecular systems as alternatives to conventional electronics, molecular interconnects; Carbon nanotube electronics, band structure & transport, devices, applications.

REFERENCE BOOKS:


SYNTHESIS AND OPTIMIZATION OF DIGITAL CIRCUITS

Subject Code : 10EC077
IA Marks : 50
No. of Lecture Hours /week : 04
Exam Hours : 03
Total no. of Lecture Hours : 52
Exam Marks : 100

Introduction: Microelectronics, semiconductor technologies and circuit taxonomy, Microelectronic design styles, computer aided synthesis and optimization.
Graphs: Notation, undirected graphs, directed graphs, combinatorial optimization, Algorithms, tractable and intractable problems, algorithms for linear and integer programs, graph optimization problems and algorithms, Boolean algebra and Applications.

Hardware Modeling: Hardware Modeling Languages, distinctive features, structural hardware language, Behavioral hardware language, HDLs used in synthesis, abstract models, structures logic networks, state diagrams, data flow and sequencing graphs, compilation and optimization techniques.

Two Level Combinational Logic Optimization: Logic optimization, principles, operation on two level logic covers, algorithms for logic minimization, symbolic minimization and encoding property, minimization of Boolean relations.

Multiple Level Combinational Optimizations: Models and transformations for combinational networks, algebraic model, Synthesis of testable network, algorithm for delay evaluation and optimization, rule based system for logic optimization.

Sequential Circuit Optimization: Sequential circuit optimization using state based models, sequential circuit optimization using network models.

Schedule Algorithms: A model for scheduling problems, Scheduling with resource and without resource constraints, Scheduling algorithms for extended sequencing models, Scheduling Pipe lined circuits.

Cell Library Binding: Problem formulation and analysis, algorithms for library binding, specific problems and algorithms for library binding (lookup table F.P.G.As and Antifuse based F.P.G.As), rule based library binding.


REFERENCE BOOKS:

ADVANCED EMBEDDED SYSTEM

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>No. of Lecture Hours /week</th>
<th>Exam Hours</th>
<th>Total no. of Lecture Hours</th>
<th>Exam Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>10EC118</td>
<td>50</td>
<td>04</td>
<td>03</td>
<td>52</td>
<td>100</td>
</tr>
</tbody>
</table>

Typical Embedded System: Core of the Embedded System, Memory, Sensors and Actuators, Communication Interface, Embedded Firmware, Other System Components Characteristics and Quality Attributes of Embedded Systems


Embedded Firmware Design and Development: Embedded Firmware Design Approaches, Embedded Firmware Development Languages
Real-Time Operating System (RTOS) based Embedded System Design.
Operating System Basics, Types of OS, Tasks, Process and Threads, Multiprocessing and Multitasking, Task Scheduling, Threads, Processes and Scheduling: Putting them altogether, Task Communication, Task Synchronization, Device Drivers, How to Choose an RTOS

The Embedded System Development Environment: The Integrated Development Environment (IDE), Types of Files Generated on Cross-compilation, Disassembler/Decompiler, Simulators, Emulators and Debugging, Target Hardware Debugging, Boundary Scan.


Reference Books:

Advanced Embedded System Lab:
Lab Experiments : -
1. 1. Use the EDA (Electronic Design Automation) tools to learn the
2. Embedded Hardware Design and for PCB design.
3. 2. Familiarize the different entities for the circuit diagram design.
    3. Familiarize with the layout design tool, building blocks, component placement, routings, design rule checking etc.

Embedded Programming Concepts (RTOS):

4. Create ‘n’ number of child threads. Each thread prints the message ‘I’m in thread number …’ and sleeps for 50 ms and then quits. The main thread waits for complete execution of all the child threads and then quits. Compile and execute in Linux.
5. 5. Implement the multithread application satisfying the following :
   1. Two child threads are created with normal priority.
   2. Thread 1 receives and prints its priority and sleeps for 50ms and then quits.
   3. Thread 2 prints the priority of the thread 1 and raises its priority to above normal and retrieves the new priority of thread 1, prints it and then quits.
   4. The main thread waits for the child thread to complete its job and quits.
6. 6. Implement the usage of anonymous pipe with 512 bytes for data sharing between parent and child processes using handle inheritance mechanism.
7. 7. Test the program below using multithread application-
   1. The main thread creates a child thread with default stack size and name ‘Child_Thread’.
   2. The main thread sends user defined messages and the message ‘WM_QUIT’ randomly to the child thread.
   3. The child thread processes the message posted by the main thread and quits when it receives the ‘WM_QUIT’ message.
   4. The main thread checks the termination of the child thread and quits when the child thread complete its execution.
   5. The main thread continues sending the random messages to the child thread till the ‘WM_QUIT’ message is sent to child thread.
   6. The messaging mechanism between the main thread and child thread is synchronous.
8. Test the program application for creating an anonymous pipe with 512 bytes of size and pass the ‘Read Handle’ of the pipe to a second process using memory mapped object. The first process writes a message ‘Hi from Pipe Server’. The 2nd process reads the data written by the pipe server to the pipe and displays it on the console. Use event object for indicating the availability of data on the pipe and mutex objects for synchronizing the access in the pipe.
9. Create a POSIX based message queue for communicating between two tasks as per the requirements given below:
   a. Use a named message queue with name ‘MyQueue’.
   b. Create two tasks(Task1 & Task2) with stack size 4000 & priorities 99 & 100 respectively.
   c. Task 1 creates the specified message queue as Read Write and reads the message present, if any, from the message queue and prints it on the console.
   d. Task2 open the message queue and posts the message ‘Hi from Task2’.
   e. Handle all possible error scenarios appropriately.

Any other experiments can be included to support the theory.

ELECTIVE II

ASIC DESIGN

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>10EC012</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No. of Lecture Hours /week</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>04</td>
<td>03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total no. of Lecture Hours</th>
<th>Exam Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>52</td>
<td>100</td>
</tr>
</tbody>
</table>

Note: All Designs Will Be Based On VHDL

Introduction: Full Custom with ASIC, Semi custom ASICS, Standard Cell based ASIC, Gate array based ASIC, Channeled gate array, Channel less gate array, structured get array, Programmable logic device, FPGA design flow, ASIC cell libraries

Data Logic Cells: Data Path Elements, Adders, Multiplier, Arithmetic Operator, I/O cell, Cell Compilers

ASIC Library Design: Logical effort: practicing delay, logical area and logical efficiency logical paths, multi stage cells, optimum delay, optimum no. of stages, library cell design.

Low-level Design Entry: Schematic Entry: Hierarchical design. The cell library, Names, Schematic, Icons & Symbols, Nets, schematic entry for ASIC’S, connections, vectored instances and buses, Edit in place attributes, Netlist, screener, Back annotation

Programmable ASIC: programmable ASIC logic cell, ASIC I/O cell
A Brief Introduction to Low Level Design Language: an introduction to EDIF, PLA Tools, an introduction to CFI designs representation. Half gate ASIC. Introduction to Synthesis and Simulation;


REFERENCE BOOKS:

SPEECH AND AUDIO PROCESSING

Time Domain Models for Speech Processing: Time dependent processing of speech, Short time energy and average magnitude, Short time average zero crossing rate, Speech vs silence discrimination using energy & zero crossings, Pitch period estimation, Short time autocorrelation function, Short time average magnitude difference function, Pitch period estimation using autocorrelation function, Median smoothing.


Short time Fourier analysis: Linear Filtering interpretation, Filter bank summation method, Overlap addition method, Design of digital filter banks, Implementation using FFT, Spectrographic displays, Pitch detection, Analysis by synthesis, Analysis synthesis systems.

Homomorphic Speech Processing: Homomorphic systems for convolution, Complex cepstrum, Pitch detection, Formant estimation, Homomorphic vocoder.

Linear Predictive Coding Of Speech: Basic principles of linear predictive analysis, Solution of LPC equations, Prediction error signal, Frequency domain interpretation, Relation between the various speech parameters, Synthesis of speech from linear predictive parameters, Applications.


Automatic Speech Recognition: Introduction, Speech recognition vs. Speaker recognition, Signal processing and analysis methods, Pattern comparison techniques, Hidden Markov Models, Artificial Neural Networks.

Audio Processing: Auditory perception and psychoacoustics - Masking, frequency and loudness perception, spatial perception, Digital Audio, Audio Coding - High quality, low-bit-rate audio coding standards, MPEG, AC-3, Multichannel audio - Stereo, 3D binaural and Multichannel surround sound.

REFERENCE BOOKS:


REAL TIME OPERATING SYSTEMS


Device Drivers, Interrupt Servicing Mechanism & Interrupt Latency.

**Real Time Operating System:** Fundamental Requirements of RTOS, Real Time Kernel Types, Schedulers, Various Scheduling modules with examples, Latency (Interrupt Latency, Scheduling Latency and Context Switching Latency), Tasks, State Transition Diagram, Task Control Block. Inter-task communication and synchronization of tasks.

**Memory and File Management:** Pipelining and Cache Memories, Paging and Segmentation, Fragmentation, Address Translation.

**Case Study:** Introduction to VX Works/Mucos/pSOS; Example systems.

**Development and Verification of Real Time Software:** Building Real Time applications; Considerations such as double buffing.

**REFERENCE BOOKS:**