ADVANCED DIGITAL DESIGN
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

SEMESTER – 1

Subject Code 16SCE11 IA Marks 20
Number of Lecture Hours/Week 04 Exam Marks 80
Total Number of Lecture Hours 50 Exam Hours 03

CREDITS – 04

Course objectives: This course will enable students to
- Explain various IC technology options
- Demonstrate Logic simulation, Design verification, Verilog.
- Illustrate behavioral modeling, Boolean-Equation, Flip-Flops and Latches; multiplexers, encoders, and decoders, synchronizers for asynchronous signals.
- Demonstrate combinational logic; three-state devices and bus interfaces; Registered logic; registers and counters; Resets; Divide and conquer; Partitioning a design.
- Define basics of PLA; PAL; Programmability of PLDs; CPLDs; FPGAs;

Module 1
Introduction: Design methodology – An introduction; IC technology options 10 Hours

Module 2
Logic Design with Verilog: Structural models of combinational logic; Logic simulation, Design verification, and Test methodology; Propagation delay; Truth-Table models of Combinational and sequential logic with Verilog. 10 Hours

Module 3
Logic Design with Behavioral Models: Behavioral modeling; A brief look at data types for behavioral modeling; Boolean-Equation – Based behavioral models of combinational logic; Propagation delay and continuous assignments; Latches and Level – Sensitive circuits in Verilog; Cyclic behavioral models of Flip-Flops and Latches; Cyclic behavior and edge detection; A comparison of styles for behavioral modeling; Behavioral models of multiplexers, encoders, and decoders; Dataflow models of a Linear-Feedback Shift Register; Modeling digital machines with repetitive algorithms; Machines with multicycle operations; Design documentation with functions and tasks; Algorithmic state machine charts for behavioral modeling; ASMD charts; Behavioral models of counters, shift registers and register files; Switch debounce, meta-stability and synchronizers for asynchronous signals; Design example 10 Hours

Module 4
Synthesis of Combinational and Sequential Logic: Introduction to synthesis; Synthesis of combinational logic; Synthesis of sequential logic with latches; Synthesis of three-state devices and bus interfaces; Synthesis of sequential logic with flip-flops; Synthesis of explicit state machines; Registered logic; State encoding; Synthesis of implicit state machines, registers and counters; Resets; Synthesis of gated clocks and clock enables; Anticipating the results of synthesis; Synthesis of loops; Design traps to avoid; Divide and conquer: Partitioning a design. 10 Hours

Module 5
Programmable Logic and Storage Devices: Programmable logic devices; storage devices; PLA; PAL; Programmability of PLDs; CPLDs; FPGAs; Verilog-Based design flows for FPGAs; Synthesis with FPGAs. 10 Hours

Course Outcomes
The students should be able to:
- Work on various IC technology options.
- Demonstrate logic simulation, Design verification, Verilog.
- Work on Flip-Flops and Latches; multiplexers, encoders, and decoders, synchronizers for
- Design and implement circuits on combinational logic; Registered logic; registers and counters; Resets; Divide and conquer: Partitioning a design.

**Question paper pattern:**
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

**Reference Books:**

---

### CLOUD COMPUTING

[As per Choice Based Credit System (CBCS) scheme]  
(Effective from the academic year 2016 -2017)

#### SEMESTER – 1

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>16SCS12/16SCE12</td>
<td>16SIT22/16SSE254</td>
<td>16SCN22/16LNI151</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>80</th>
</tr>
</thead>
<tbody>
<tr>
<td>04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
<th>03</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CREDITS – 04**

**Course objectives:** This course will enable students to
- Define and Cloud, models and Services.
- Compare and contrast programming for cloud and their applications
- Explain virtualization, Task Scheduling algorithms.
- Apply ZooKeeper, Map-Reduce concept to applications.

<table>
<thead>
<tr>
<th>Module 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction, Cloud Infrastructure:</strong> Cloud computing, Cloud computing delivery models and services, Ethical issues, Cloud vulnerabilities, Cloud computing at Amazon, Cloud computing the Google perspective, Microsoft Windows Azure and online services, Open-source software platforms for private clouds, Cloud storage diversity and vendor lock-in, Energy use and ecological impact, Service level agreements, User experience and software licensing. Exercises and problems.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cloud Computing: Application Paradigms.:</strong> Challenges of cloud computing, Architectural styles of cloud computing, Workflows: Coordination of multiple activities, Coordination based on a state machine model: The Zookeeper, The Map Reduce programming model, A case study: The Gre The Web application, Cloud for science and engineering, High-performance computing on a cloud, Cloud computing for Biology research, Social computing, digital content and cloud computing.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cloud Resource Virtualization:</strong> Virtualization, Layering and virtualization, Virtual machine monitors, Virtual Machines, Performance and Security Isolation, Full virtualization and paravirtualization, Hardware support for virtualization, Case Study: Xen a VMM based paravirtualization, Optimization of network virtualization, vBlades</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>
Performance comparison of virtual machines, The dark side of virtualization, Exercises and problems

<table>
<thead>
<tr>
<th>Module 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cloud Resource Management and Scheduling:</strong> Policies and mechanisms for resource management, Application of control theory to task scheduling on a cloud, Stability of a two-level resource allocation architecture, Feedback control based on dynamic thresholds, Coordination of specialized autonomic performance managers, A utility-based model for cloud-based Web services, Resourcing bundling: Combinatorial auctions for cloud resources, Scheduling algorithms for computing clouds, Fair queuing, Start-time fair queuing, Borrowed virtual time, Cloud scheduling subject to deadlines, Scheduling MapReduce applications subject to deadlines, Resource management and dynamic scaling, Exercises and problems.</td>
</tr>
<tr>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cloud Security, Cloud Application Development:</strong> Cloud security risks, Security: The top concern for cloud users, Privacy and privacy impact assessment, Trust, Operating system security, Virtual machine Security, Security of virtualization, Security risks posed by shared images, Security risks posed by a management OS, A trusted virtual machine monitor, Amazon web services: EC2 instances, Connecting clients to cloud instances through firewalls, Security rules for application and transport layer protocols in EC2, How to launch an EC2 Linux instance and connect to it, How to use S3 in java, Cloud-based simulation of a distributed trust algorithm, A trust management service, A cloud service for adaptive data streaming, Cloud based optimal FPGA synthesis, Exercises and problems.</td>
</tr>
<tr>
<td>10 Hours</td>
</tr>
</tbody>
</table>

**Course Outcomes**

The students should be able to:

- Compare the strengths and limitations of cloud computing
- Identify the architecture, infrastructure and delivery models of cloud computing
- Apply suitable virtualization concept.
- Choose the appropriate cloud player
- Address the core issues of cloud computing such as security, privacy and interoperability
- Design Cloud Services
- Set a private cloud

**Question paper pattern:**

The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

**EMBEDDED COMPUTING SYSTEMS**

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2016 -2017)

**SEMESTER – 1**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>16SCE13 /16SCS152</th>
<th>IA Marks</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>04</td>
<td>Exam Marks</td>
<td>80</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>50</td>
<td>Exam Hours</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 04**

**Course objectives:** This course will enable students to

- Explain a general overview of Embedded Systems
- Show current statistics of Embedded Systems
- Examine a complete microprocessor-based hardware system
- Design, code, compile, and test real-time software
- Integrate a fully functional system including hardware and software
- Make intelligent choices between hardware/software tradeoffs

<table>
<thead>
<tr>
<th>Module 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to embedded systems: Embedded systems, Processor embedded into a system, Embedded hardware units and device in a system, Embedded software in a system, Examples of embedded systems, Design process in embedded system, Formalization of system design, Design process and design examples, Classification of embedded systems, skills required for an embedded system designer.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Devices and communication buses for devices network: IO types and example, Serial communication devices, Parallel device ports, Sophisticated interfacing features in device ports, Wireless devices, Timer and counting devices, Watchdog timer, Real time clock, Networked embedded systems, Serial bus communication protocols, Parallel bus device protocols-parallel communication internet using ISA, PCI, PCI-X and advanced buses, Internet enabled systems-network protocols, Wireless and mobile system protocols.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device drivers and interrupts and service mechanism: Programming-I/O busy-wait approach without interrupt service mechanism, ISR concept, Interrupt sources, Interrupt servicing (Handling) Mechanism, Multiple interrupts, Context and the periods for context switching, interrupt latency and deadline, Classification of processors interrupt service mechanism from Context-saving angle, Direct memory access, Device driver programming.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 4</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter process communication and synchronization of processes, Threads and tasks: Multiple process in an application, Multiple threads in an application, Tasks, Task states, Task and Data, Clear-cut distinction between functions. ISRS and tasks by their characteristics, concept and semaphores, Shared data, Inter-process communication, Signal function, Semaphore functions, Message Queue functions, Mailbox functions, Pipe functions, Socket functions, RPC functions.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 5</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-time operating systems: OS Services, Process management, Timer functions, Event functions, Memory management, Device, file and IO subsystems management, Interrupt routines in RTOS environment and handling of interrupt source calls, Real-time operating systems, Basic design using an RTOS, RTOS task scheduling models, interrupt latency and response of the tasks as performance metrics, OS security issues.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>
Introduction to embedded software development process and tools, Host and target machines, Linking and location software.

**Course Outcomes**

The students should be able to:

- Distinguish the characteristics of embedded computer systems.
- Examine the various vulnerabilities of embedded computer systems.
- Design an embedded system.
- Design and develop modules using RTOS.
- Implement RPC, threads and tasks.

**Question paper pattern:**

The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**


---

**PROBABILITY STATISTICS AND QUEUING THEORY**

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2016 -2017)

**SEMESTER – I**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>16LNI14 / 16SCN14/16SCS14/16SSE14 / 16SIT14 /16SCE14 / 16SFC14</td>
<td>20</td>
<td>80</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 04**

**Course objectives:** This course will enable students to

- Develop analytical capability and to impart knowledge of Probability, Statistics and Queuing.
- Apply above concepts in Engineering and Technology.
- Acquire knowledge of Hypothesis testing and Queuing methods and their applications so as to enable them to apply them for solving real world problems.

**Module 1**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axioms of probability, Conditional probability, Total probability, Baye’s theorem, Discrete Random variable, Probability mass function, Continuous Random variable. Probability density function, Cumulative Distribution Function, and its properties, Two-dimensional Random variables, Joint pdf / cdf and their properties</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

**Module 2**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability Distributions / Discrete distributions: Binomial, Poisson Geometric and Hyper-geometric distributions and their properties. Continuous distributions: Uniform, Normal, exponential distributions and their properties.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

**Module 3**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Processes: Classification, Methods of description, Special classes, Average values of Random Processes, Analytical representation of Random Process,</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

| Module 4 | Testing Hypothesis: Testing of Hypothesis: Formulation of Null hypothesis, critical region, level of significance, errors in testing, Tests of significance for Large and Small Samples, t-distribution, its properties and uses, F-distribution, its properties and uses, Chi-square distribution, its properties and uses, $\chi^2$ – test for goodness of fit, $\chi^2$ test for Independence | 10 Hours |

| Module 5 | Symbolic Representation of a Queuing Model, Poisson Queue system, Little Law, Types of Stochastic Processes, Birth-Death Process, The M/M/1 Queuing System, The M/M/s Queuing System, The M/M/s Queuing with Finite buffers. | 10 Hours |

**Course Outcomes**

The students should be able to:

- Demonstrate use of probability and characterize probability models using probability mass (density) functions & cumulative distribution functions.
- Explain the techniques of developing discrete & continuous probability distributions and its applications.
- Describe a random process in terms of its mean and correlation functions.
- Outline methods of Hypothesis testing for goodness of fit.
- Define the terminology & nomenclature appropriate queuing theory and also distinguish various queuing models.

**Question paper pattern:**

The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

## COMPUTE R SYSTEMS PERFORMANCE ANALYSIS

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2016 -2017)

### SEMESTER – I

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>16SCE151</td>
<td>20</td>
<td>80</td>
</tr>
</tbody>
</table>

| Number of Lecture Hours/Week | 03 | Exam Hours | 03 |
|------------------------------|----|------------|
| Total Number of Lecture Hours| 40 |            |

### CREDITS – 03

#### Course objectives:
This course will enable students to
- Discuss mathematical foundations needed for performance evaluation of computer systems
- Illustrate metrics used for performance evaluation
- Develop the analytical modeling of computer systems
- Develop new queuing analysis for both simple and complex systems
- Analyze techniques for evaluating scheduling policies

<table>
<thead>
<tr>
<th>Module 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 1</td>
<td>Introduction: The art of Performance Evaluation; Common Mistakes in Performance Evaluation, A Systematic Approach to Performance Evaluation, Selecting an Evaluation Technique, Selecting Performance Metrics, Commonly used Performance Metrics, Utility Classification of Performance Metrics, Setting Performance Requirements.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 2</td>
<td>Workloads, Workload Selection and Characterization: Types of Workloads, addition instructions, Instruction mixes, Kernels; Synthetic programs, Application benchmarks, popular benchmarks. Work load Selection: Services exercised, level of detail; Representativeness; Timeliness, Other considerations in workload selection. Work load characterization Techniques: Terminology; Averaging, Specifying dispersion, Single Parameter Histograms, Multi Parameter Histograms, Principle Component Analysis, Markov Models, Clustering.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 3</td>
<td>Monitors, Program Execution Monitors and Accounting Logs: Monitors: Terminology and classification; Software and hardware monitors, Software versus hardware monitors, Firmware and hybrid monitors, Distributed System Monitors, Program Execution Monitors and Accounting Logs, Program Execution Monitors, Techniques for Improving Program Performance, Accounting Logs, Analysis and Interpretation of Accounting log data, Using accounting logs to answer commonly asked questions.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 4</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 4</td>
<td>Capacity Planning and Benchmarking: Steps in capacity planning and management; Problems in Capacity Planning; Common Mistakes in Benchmarking; Benchmarking Games; Load Drivers; Remote- Terminal Emulation; Components of an RTE; Limitations of RTEs. Experimental Design and Analysis: Introduction: Terminology, Common mistakes in experiments, Types of experimental designs, 2k Factorial Designs, Concepts, Computation of effects, Sign table method for computing effects; Allocation of variance; General 2k Factorial Designs, General full factorial designs with k factors: Model, Analysis of a General Design, Informal Methods.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 5</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 5</td>
<td>Queuing Models: Introduction: Queuing Notation; Rules for all Queues; Little’s Law, Types of Stochastic Process. Analysis of Single Queue: Birth-Death Processes; M/M/1 Queue; M/M/m Queue; M/M/m/B Queue with finite buffers; Results for other M/M/1 Queuing Systems. Queuing Networks: Open and Closed Queuing Networks; Product form networks, queuing Network models of Computer Systems. Operational Laws:</td>
</tr>
</tbody>
</table>
Utilization Law; Forced Flow Law; Little's Law; General Response Time Law; Interactive Response Time Law; Bottleneck Analysis; Mean Value Analysis and Related Techniques; Analysis of Open Queuing Networks; Mean Value Analysis; Approximate MVA; Balanced Job Bounds; Convolution Algorithm, Distribution of Jobs in a System, Convolution Algorithm for Computing $G(N)$, Computing Performance using $G(N)$, Timesharing Systems, Hierarchical Decomposition of Large Queuing Networks; Load Dependent Service Centers, Hierarchical Decomposition, Limitations of Queuing Theory.

**Course Outcomes**

The students should be able to:
- Identify the need for performance evaluation and the metrics used for it
- Implement Little's law and other operational laws
- Apply the operational laws to open and closed systems
- Use discrete-time and continuous-time Markov chains to model real world systems
- Develop analytical techniques for evaluating scheduling policies

**Question paper pattern:**

The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

DISTRIBUTED OPERATING SYSTEM
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016-2017)
SEMESTER – I

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>16SCE152/ 16SIT154 / 16SSE152</th>
<th>IA Marks</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>03</td>
<td>Exam Marks</td>
<td>80</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>40</td>
<td>Exam Hours</td>
<td>03</td>
</tr>
</tbody>
</table>

CREDITS – 03

Course objectives: This course will enable students to
- Explain distributed systems principles associated with communication, naming, synchronization, distributed file systems, system design, distributed scheduling, and several case studies
- Extend foundational concepts and as well as practical deployments.
- Recall distributed operating system concepts that includes architecture, Mutual exclusion algorithms, Deadlock detection algorithms and agreement protocols
- Explain the distributed resource management components viz. the algorithms for implementation of distributed shared memory, recovery and commit protocols

Module 1


Module 2

**Remote Procedure Calls:** Introduction, The RPC Model, Transparency of RPC, Implementing RPC Mechanism, Stub Generation, RPC Messages, Marshaling Arguments and Results, Server Management, Parameter-Passing Semantics, Call Semantics, Communication Protocols for RPCs, Complicated RPCs, Client-Server Binding, Exception Handling, Security, Some Special Types of RPCs, RPC in Heterogeneous Environments, Lightweight RPC, Optimization for Better Performance, Case Studies: Sun RPC.

Module 3

**Distributed Shared Memory:** Introduction, General Architecture of DSM Systems, Design and Implementation Issues of DSM, Granularity, Structure of Shared Memory Space, Consistency Models, Replacement Strategy, Thrashing, Other approaches to DSM, Heterogeneous DSM, Advantages of DSM. **Synchronization:** Introduction, Clock Synchronization, Event Ordering, Mutual Exclusion, Dead Lock, Election Algorithms.

Module 4


Module 5

Course Outcomes
The students should be able to:

- The concepts underlying distributed systems
- Demonstrate an ability to apply theory and techniques to unseen problems.
- Demonstrate the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
- Explore the various resource management techniques for distributed systems.

Question paper pattern:
The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

Reference Books:

SOFTWARE AGENTS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

SEMESTER – I

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>16LNI423 / 16SCE153</td>
<td></td>
<td>80</td>
<td>03</td>
</tr>
</tbody>
</table>

IA Marks: 20

Course objectives: This course will enable students to
- Explain the principles and fundamentals of designing agents
- Define the architecture design of different agents.
- Demonstrate design of the agents
- Illustrate user interaction with agents
- Discover the role of agents in assisting the users in day to day activities

Module -1

Module -2
Electronic Messages, Semiformal Systems and Radical Tailorability, Oval: A Radically Tailorable Tool for Information Management and Cooperative Work, Examples of Application and Agents in Oval, Conclusions: An Addendum: The Relationship between Oval and Objects Lens

**Module – 3**

Agents that Reduce Work and Information Overload Introduction, Approaches to Building Agents, Training a Personal Digital Assistant, Some Example of Existing Agents, Electronic Mail Agents, Meeting Scheduling Agent, News Filtering Agent, Entertainment Selection Agent, Discussion, Acknowledgements Software Agents for Cooperative Learning: Computer-Supported Cooperative Learning, Examples of Software Agents for Cooperative Learning, Examples of Software Agents for Cooperative Learning, Developing an Example, Discussion and Perspectives.

| 8 Hours |

**Module-4**


| 8 Hours |

**Module-5**

Agent for Information Gathering: Agent Organization, The Knowledge of an Agent, The Domain Model of an Agent, Modeling other Agent, communication language and protocol, query processing, an information goal, information source selection, generating a query access plan, interleaving planning and execution, semantic query optimization, learning, caching retrieved data, related work, discursion, acknowledgement. Mobile Agents: Enabling Mobile Agents, Programming Mobile Agents, Using Mobile Agents.

| 8 Hours |

**Course outcomes:**

The students should be able to:

- Identify and explore the advantages of agents and design the architecture for an agent
- Analyze the agent in details in a view for the implementation
- Analyze communicative actions with agents.
- Analyze typical agents using a tool for different types of applications.

**Question paper pattern:**

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

### Course Objectives

This course will enable students to:

- Discover recent trends in the field of Computer Architecture and identify performance related parameters
- Explain pipelining, thread-level parallelism and Memory hierarchy design

### Module 1

Data-Level Parallelism in vector, SIMD, and GPU Architectures: Introduction, Vector Architecture, SIMD Instructions Set Extensions for Multimedia, Graphics Processing Units, Detecting and Enhancing Loop-level Parallelism, Crosscutting Issues, Putting it All Together: Mobile versus Server GPUs and Tesla versus Core i7, Fallacies and Pitfalls, Concluding Remarks, Historical Perspective and References Case Study and Exercises by Jason D. Bakos.

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

### Module 2


<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

### Module 3


<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

### Module 4


<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

### Module 5


<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

### Course Outcomes

The students should be able to:
- Implement Pipelining concepts
- Identify the limitations of ILP
- Demonstrate an ability to apply theory and techniques to unseen problems.
- Interpret the thread-level parallelism concepts.
- Explain concepts of vector process super computers and Cray X1.

**Question paper pattern:**
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

**Reference Books:**

---

**DIGITAL DESIGN AND ECS LABORATORY**

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

**SEMESTER – 1**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>16SCE16</td>
<td></td>
<td>80</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 02**

**Course objectives:** This course will enable students to
- Implement Behavioral modeling, few combinational and sequential digital circuits.
- Implement basic network error correction schemes on Verilog;
- Design a complete microprocessor-based hardware system
- Design, code, compile, and test real-time software
- Integrate a fully functional system including hardware and software
- Gain the ability to make intelligent choices between hardware/software tradeoffs

**PART – A: Digital Design Lab Work**

**Experiment List:**
**Note:** Use appropriate tools/language to implement the following experiment:

1. Design, develop, and verify a Verilog module that implements a JK Edge-Triggered Flip-Flop with Active-Low Preset ad Clear Inputs.

2. Design, develop, and verify a Verilog module that produces a 4-bit output indicating the number of 1s in an 8-bit input word.

3. Design, develop, and verify a Verilog module that implements a Universal Shift Register and then implement a bidirectional ring counter capable of counting in either direction, beginning with first active clock edge after reset.

4. Design, develop, and verify a Verilog module that implements a counter whose modulus value n ≤ 10.

5. Design, develop, and verify a Verilog module that implements a Hamming Encoder that produces a 7-bit Hamming code given a 4-bit input word.
6. Design, develop, and verify a Verilog module that implements a 16 bit cyclic redundancy check (CRC-16) algorithm, which shall encode 14 bits of message with a 3-bit CRC.

Note: Student can verify the verilog module output using Xilinx or equivalent simulator and FPGA Kit

PART – B: Embedded Computing Lab Work

Experiment List:
To get in touch with development tool/environment for ATMEL microcontroller program and architecture. To know the overview of Kiel software and an introduction to ATMEL 8051 architecture.
1. Write an embedded C program to add subtract multiply divide 16 bit data by ATMEL microcontroller. Write a separate module for each of the arithmetic module and bind it under a single module.
2. Write embedded c program to generate 10 KHz frequency using interrupts on P1.2 ant to view it on the CRO.
3. Write a program to interface 16X2 LCD to ATMEL microcontroller and use port P0 for interfacing it and use port P1 to interface key board.
4. Write a program to control DC motor using PWM method. To monitor the PWM status and control the speed of DC motor in 100% and 25% duty cycle pulse.
5. Transmission and reception of data. The module has to be designed to have a clear understanding of how serial and parallel interface devices are controlled and interfaced with microcontroller.

NOTE: Use AT89C52 microcontroller as main kit with peripherals and Keil µVision 4/ Equivalent tool.

Course Outcomes
The students should be able to:
- Develop basic building blocks of digital circuits using Verilog
- FPGA programming and implementation of Verilog codes on them.
- To get in touch with development tool/environment for ATMEL microcontroller program and architecture.
- To know the overview of Kiel software and an introduction to ATMEL 8051 architecture.
- Design and develop modules using RTOS

Conduction of Practical Examination:
1. All laboratory experiments must be included for practical examination.
2. Students are allowed to pick one experiment from each part and execute both.
3. Strictly follow the instructions as printed on the cover page of answer script and breakup of marks
4. PART –A: Procedure + Conduction + Viva: 10 + 20 +10 (40)
5. PART –B: Procedure + Conduction + Viva: 10 + 20 +10 (40)
6. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.
## SEMINAR

[As per Choice Based Credit System (CBCS) scheme]  
(Effective from the academic year 2016 -2017)

### SEMESTER – I

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>16SCE17 / 16SCN17 / 16LNI17 / 16SIT17 / 16SSE17 / 16SCS17 / 16SIT17 / 16SFC17</td>
<td>IA Marks</td>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Lecture Hours</td>
<td>Exam Hours</td>
<td>-</td>
</tr>
</tbody>
</table>

### CREDITS – 01

**Course objectives:** This course will enable students to

- Motivate the students to read technical article
- Discover recent technology developments

**Descriptions**

The students should read a recent technical article (try to narrow down the topic as much as possible) from any of the leading reputed and refereed journals like:

1. IEEE Transactions, journals, magazines, etc.
2. ACM Transactions, journals, magazines, SIG series, etc.
3. Springer
4. Elsevier publications etc

In the area of (to name few and not limited to)

- Web Technology
- Cloud Computing
- Artificial Intelligent
- Networking
- Security
- Data mining

**Course Outcomes**

The students should be able to:

- Conduct survey on recent technologies
- Infer and interpret the information from the survey conducted
- Motivated towards research

**Conduction:**

The students have to present at least ONE technical seminar on the selected topic and submit a report for internal evaluation.

**Marks Distribution:** Literature Survey + Presentation (PPT) + Report + Question & Answer + Paper: 20 + 30 + 30 + 20 (100).