**COURSE:** IV B.Tech I SEM **REGULATION:** R-16

**BRANCH:** ECE  **SUBJECT:** System Design through Verilog

**Course Objective:** The student will be able to

* Understand the basic elements of verilog language
* Understand the design of basic circuits in verilog in different levels of modeling.
* Understand the synthesis of combinational and sequential logic using verilog
* Understand the design of verilog models for microprocessor and microcontrollers

**Course Outcomes:** After the completion of the course the student will be able to

* **CO1:** To **Understand** language constructs and conventions to write verilog program
* **CO2:** To **design** basic circuits with gate level modeling
* **CO3:** To **write** verilog programin behavioral modeling withdifferent design constructs
* **CO4:** To **construct** verilog program in data flow level and switch level modeling
* **CO5:** To **synthesize** verilog for combinational and sequential logics
* **CO6:** To **Develop** verilog modules for microprocessor and microcontrollers

**LESSON PLAN**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UNIT-IINTRODUCTION TO VERILOG: 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. LANGUAGE CONSTRUCTS AND CONVENTIONS: Introduction, keywords, identifiers, whitespace characters, comments, numbers, strings, logic values, data types, scalars and vectors, parameters, memory, operators, system tasks. | | | | | | | | |
| S.NO | | TOPIC | | NO.HOURS | | COs | | POs |
| 1 | | Verilog as HDL, Levels of design description, concurrency, simulation and synthesis | | 2 | | CO1 | |  |
| 2 | | functional verification, system tasks, programming language interface(PLI) | | 2 | | CO1 | |  |
| 3 | | module, simulation and synthesis tools, test benches | | 2 | | CO1 | |  |
| 4 | | Language constructs and conventions | | 1 | | CO1 | |  |
| 5 | | keywords, identifiers, whitespace characters, comments, numbers, strings | | 2 | | CO1 | |  |
| 6 | | logic values, data types, scalars and vectors, parameters, memory, operators, system tasks. | | 2 | | CO1 | |  |
| 7 | | Extra topics | | 1 | |  | |  |
| TOTAL ----  **12** | | | | | | | | |
| UNIT-II GATE LEVEL MODELLING: Introduction, AND gate primitive, module structure, other gate primitives, illustrative examples, tristate gates, array of instances of primitives, design of Flip flops with gate primitives, delays, strengths and contention resolution, net types, design of basic circuits. | | | | | | | | |
| 8 | | Introduction to gate level modelling | | 1 | | CO2 | |  |
| 9 | | AND gate primitive, module structure, other gate primitives, illustrative examples | | 2 | | CO2 | |  |
| 10 | | tristate gates, array of instances of primitives | | 2 | | CO2 | |  |
| 11 | | design of Flip flops with gate primitives. | | 2 | | CO2 | |  |
| 12 | | delays, strengths and contention resolution, net types | | 1 | | CO2 | |  |
| 13 | | design of basic circuits. | | 1 | | CO2 | |  |
| 14 | | Extra topics | | **1** | |  | |  |
| TOTAL ---- **10** | | | | | | | | |
| UNIT-III BEHAVIORAL MODELLING: Introduction, operations and assignments, functional Bifurcation, initial construct, always construct, examples, assignments with delays, wait construct, multiple always blocks, designs at behavioural level, blocking and non blocking assignments, the case statement, simulation flow, if and if else constructs, assign-De assign construct, repeat construct, FOR loop, the disable construct, While loop, Forever loop, parallel blocks, force-release construct, event. | | | | | | | | |
| 15 | Introduction to Behavioral Modeling | | 1 | | CO3 | |  | |
| 16 | operations and assignments, functional Bifurcation, initial construct, always construct, examples | | 2 | | CO3 | |  | |
| 17 | assignments with delays, wait construct, multiple always blocks, designs at behavioral level, blocking and non blocking assignments | | 2 | | CO3 | |  | |
| 18 | the case statement, simulation flow, if and if else constructs, assign-De assign construct, repeat construct | | 2 | | CO3 | |  | |
| 19 | FOR loop, the disable construct, While loop, Forever loop, parallel blocks, force-release construct, event. | | 2 | | CO3 | |  | |
| 20 | Extra Programms | | 1 | |  | |  | |
| TOTAL ---- **10** | | | | | | | | |
| UNIT-IV DATAFLOW LEVEL AND SWITCH LEVEL MODELLING: Introduction, continuous assignment structures, delays and continuous assignments, assignment to vectors, basic transistor switches, CMOS switch, Bidirectional gates and time delays with switch primitives, instantiations with strengths and delays, strength contention with trireg nets. | | | | | | | | |
| 21 | Introduction, continuous assignment structures, delays and continuous assignments | | 2 | | CO4 | |  | |
| 22 | assignment to vectors, basic transistor switches, CMOS switch | | 2 | | CO4 | |  | |
| 23 | Bidirectional gates and time delays with switch primitives | | 2 | | CO4 | |  | |
| 24 | instantiations with strengths and delays | | 1 | | CO4 | |  | |
| 25 | strength contention with trireg nets | | 1 | | CO4 | |  | |
| 26 | Extra Programms. | | 2 | | CO4 | |  | |
| TOTAL ---- **10** | | | | | | | | |
| UNIT-V SYNTHSIS OF COMBINATIONAL AND SEQUENTIAL LOGIC USING VERILOG: Synthesis of combinational logic: Net list of structured primitives, a set of continuous assignment statements and level sensitive cyclic behavior with examples, Synthesis of priority structures, Exploiting logic don’t care conditions. Synthesis of sequential logic with latches: Accidental synthesis of latches and Intentional synthesis of latches, Synthesis of sequential logic with flip-flops, Synthesis of explicit state machines. | | | | | | | | |
| 27 | Synthesis of combinational logic: Net list of structured primitives | | 2 | | CO5 | |  | |
| 28 | a set of continuous assignment statements and level sensitive cyclic behavior with examples | | 2 | | CO5 | |  | |
| 29 | Synthesis of priority structures, Exploiting logic don’t care conditions. | | 3 | | CO5 | |  | |
| 30 | Synthesis of sequential logic with latches | | 3 | | CO5 | |  | |
| 31 | Accidental synthesis of latches and Intentional synthesis of latches | | 3 | | CO5 | |  | |
| 32 | Synthesis of sequential logic with flip-flops, Synthesis of explicit state machines | | 2 | | CO5 | |  | |
| TOTAL ---- **15** | | | | | | | | |
| UNIT-VI VERILOG MODELS: Static RAM Memory, A simplified 486 Bus Model, Interfacing Memory to a Microprocessor Bus, UART Design and Design of Microcontroller CPU. | | | | | | | | |
| 40 | Static RAM Memory | | 2 | | CO6 | |  | |
| 41 | A simplified 486 Bus Model | | 1 | | CO6 | |  | |
| 42 | Interfacing Memory to a Microprocessor Bus | | 2 | | CO6 | |  | |
| 43 | UART Design and Design of Microcontroller CPU | | 1 | | CO6 | |  | |
| TOTAL ---- **6** | | | | | | | | |

TEXT BOOKS:1. Design through Verilog HDL – T.R. Padmanabhan and B. Bala Tripura Sundari, WSE, IEEE Press, 2004.  
2. Advanced Digital Design with Verilog HDL – Michael D. Ciletti, PHI, 2005.  
REFERENCES:1. Fundamentals of Logic Design with Verilog – Stephen. Brown and Zvonko Vranesic, TMH, 2005.  
2. A Verilog Primier – J. Bhasker, BSP, 2003.

**Expected Total number of teaching Hours: 63**

**FACULTY HOD-ECE**

1. **Dr. Sk. Enaul Haq)**
2. **Shaik. Riyazuddien)**