INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad, Telangana

COURSE CONTENT

DESIGN AND VERIFICATION USING SYSTEM VERILOG								
I Semester: ES								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
BESE18	**Elective**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	0	0	3	40	60	100
Contact Classes: 45	**Tutorial Classes: Nil**	**Practical Classes: Nil**				**Total Classes: 45**		
Prerequisite: Digital Electronics, Basic HDL Programming.								

I. COURSE OVERVIEW:

This course introduces students to SystemVerilog, a modern hardware description and verification language widely used in digital design. It focuses on both RTL design and verification methodologies, covering synthesizable constructs, testbench creation, assertions, functional coverage, constrained random testing, and object-oriented verification features. The course bridges the gap between hardware design and verification by providing hands-on experience with industry-standard practices, including the Universal Verification Methodology (UVM).

II. COURSES OBJECTIVES:

The students will try to learn

I. Fundamentals of SystemVerilog language constructs for modeling and design.

II. Development of structured testbenches for digital system verification. Application of object-oriented programming (OOP) features in SystemVerilog for reusable verification environments.

III. Basics of UVM and its role in large-scale, reusable, and scalable verification.

III. COURSE OUTCOMES:

At the end of the course students should be able to:

CO 1 Apply SystemVerilog constructs to design synthesizable RTL modules.

CO 2 Develop effective SystemVerilog testbenches for digital circuits.

CO 3 Use assertions to verify correctness and improve design reliability.

CO 4 Implement constrained random verification and functional coverage.

CO 5 Employ OOP concepts in SystemVerilog for modular and reusable verification components.

CO 6 Demonstrate familiarity with UVM methodology for industry-standard verification practices.

IV. COURSE CONTENT:

MODULE – I: VERIFICATION METHODOLOGIES: (9)

The Verification Process, The Verification Plan, The Verification Methodology Manual, Basic Testbench Functionality, Directed Testing, Methodology Basics, Constrained-Random Stimulus, What Should You Randomize, Functional Coverage, Testbench Components, Layered Testbench, Building a Layered Testbench, Simulation Environment Phases, Maximum Code Reuse, Testbench Performance, Conclusion.

MODULE – II: FUNDAMENTALS OF SYSTEM VERILOG (9)

DATA TYPES, Built-in Data Types, Fixed-Size Arrays, Dynamic Arrays, Queues, Associative Arrays, Linked Lists, Array Methods, Choosing a Storage Type, Creating New Types with typedef, Creating User-Defined Structures, Enumerated Types, Constants, Strings, Expression Width, Net Types, Conclusion, PROCEDURAL STATEMENTS AND ROUTINES, Introduction Procedural Statements, Tasks, Functions, and Void Functions, Task and Function Overview, Routine Arguments, Returning from a Routine, Local Data Storage, Time Values.

MODULE – III: OBJECT ORIENTED CONCEPTS FOR VERIFICATION (9)

Think of Nouns, not Verbs, Your First Class, Where to Define a Class, 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, Public vs. Private, Straying Off Course, Building a Testbench.

MODULE –IV: RANDOMIZATION TECHNIQUES FOR VERIFICATION (9)

What to Randomize, Randomization in SystemVerilog, Constraint Details, Solution Probabilities, Controlling Multiple Constraint Blocks, Valid Constraints, In-line Constraints, The pre_randomize and post_randomize Functions, Constraints Tips and Techniques, Common Randomization Problems, Iterative and Array Constraints, Atomic Stimulus Generation vs. Scenario Generation, Random Control, Random Generators, Random Device Configuration.

MODULE – V: CONNECTING THE TESTBENCH AND DESIGN (9)

Separating the Testbench and Design, The Interface Construct, Stimulus Timing, Interface Driving and Sampling, Connecting It All Together, Top-Level Scope, Program – Module Interactions, SystemVerilog Assertions, The Four-Port

V. TEXT BOOKS:

1. CHRIS SPEAR Synopsys, Inc. “SYSTEMVERILOG FOR VERIFICATION A Guide to Learning the Testbench Language Features” Springer.
2. Jonathan W. Valvano, *Embedded Systems: Introduction to Arm® Cortex™-M Microcontrollers*. VI.

VI. REFERENCE BOOKS:

1. **Jonathan W. Valvano**, *Embedded Microcomputer Systems: Real-Time Interfacing*, Brooks/Cole, 1999.
2. **M. A. Mazidi**, *The AVR Microcontroller and Embedded Systems: Using Assembly and C*, 2nd Edition, Pearson Education Limited, 2011.
3. **William Hohl**, *ARM Assembly Language*, CRC Press, ISBN: 978-81-89643-04-1.

VII. MATERIALS ONLINE

1. Course template
2. Tutorial question bank
3. Assignments
4. Model question paper - I
5. Model question paper - II
6. Lecture notes
7. Power point presentations
8. Early Lecture Readiness Videos