GUJARAT TECHNOLOGICAL UNIVERSITY

Master of Engineering Syllabus

Subject Code : 3720403

Subject Name : VLSI Design Verification Techniques

WEF Academic Year :	2023 – 24
Semester :	2
Category of the Course :	Program Elective

Type of course : Analysis and Design of CMOS based Digital Circuits.

Prerequisite : Basic Digital Electronics and CMOS based circuits.

Rationale : This course will provide an opportunity to the students to learn about various topics of VLSI such as design of digital circuits using MOSFET device as well as using hardware description language (HDL). In laboratory part of this course, students will be given exposure to HDL for automated design of digital circuits. This subject is very important for the students who would like to pursue their career in VLSI domain.

Course Scheme :

Teaching Scheme			Total Credits	Assessment Pattern and Marks				Total Marks
				Theory		Practical		
L	T	PR	C	ESE (E)	PA(M)	ESE (V)	PA (I)	
3	0	2	4	70	30	30	20	150

Course Content :

Sr. No.	Course Content	No. of Hours	% of Weightage
1	**The Verification Plan** The Role of the Verification Plan, Levels of Verification, Verification Strategies, From Specification to Features, From Features to test cases, From Test cases to Test benches.	7	10
2	**Verification Tools** Linting Tools, Simulators, Third-Party Models, Waveform Viewers, Code Coverage, Verification Languages, Revision Control, Issue Tracking, and Metrics.	8	15

3	**Functional Verification Methods and Tools** Concept, Test bench architecture, Simulation tools, Functional and code coverage.	8	15
4	**Formal Verification Methods** Binary Decision Diagram, Equivalence Checking, Assertion based verification, Emulation.	7	15
5	**Introduction to System Verilog** System Verilog modeling concepts, Data types, declarations, syntax rules, Procedural blocks and assignments, Programming statements, Operators and operation rules, Compound data types and packages, System Verilog interface ports, Verification constructs and testbench interfaces, Verification timing using clocking blocks, Object Oriented testbench, Dynamic arrays and scoreboards, Constrained random stimulus generation, Functional coverage, Synchronization (events, mailboxes, semaphores), Assertions overview.	10	25
6	**Universal Verification Methodology (UVM)** Introduction, Overview of UVM, Object Oriented Programming, UVM library Basics, Interface Universal Verification Component.	8	20
	Total	**48**	**100**

Reference Books :

1. Introduction to Formal Hardware Verification, T.Kropf, Springer Verlag, 2000.
2. System-on-a-Chip Verification- Methodology and Techniques, P. Rashinkar, Paterson and L. Singh, Kluwer Academic Publishers, 2001.
3. Janick Bergeron, “Writing test benches functional verification of HDL models” Kluwer Academic Publishers, New York, Boston, Dordrecht, London, Moscow, 2002.
4. Janick Bergeron, Writing Testbenches using SystemVerilog, Springer.
5. Srikanth Vijayaraghavan, Meyyappan Ramanathan - A Practical Guide for SystemVerilog AssertionsSpringer.
6. Ray Salemi - The UVM Primer_ An Introduction to the Universal Verification Methodology-Boston Light Press.
7. Spear, Chris, Tumbush, Greg ,System Verilog for Verification- A guide to learning the testbench language features, Springer.

Course Outcome :

CO1 : To help the engineers to design the system with Hardware Description Language and system Verilog

CO2 : To practice for writing synthesizable RTL models that work correctly in both simulation and synthesis.

CO3 : Understand the features and capabilities of the UVM class library for system Verilog.

CO4 : Combine multiple UVCs into a complete verification environment.

CO5 : Create and configure reusable, scalable, and robust UVM verification components (UVCs).

Suggested List of Experiments :

1) Introduction to Verilog and System Verilog
2) Running simulator and debug tools
3) Experiment with 2 state and 4 state data types 4) Implementation of counters and testing.
5) Implementation of Mealy and Moore FSM and testing.
6) Experiment with blocking and non-blocking assignments
7) Model and verify simple ALU
8) Model and verify an Instruction stack
9) Use an interface between testbench and DUT
10) Developing a test program
11) Create a simple and advanced OO testbench
12) Use mailboxes for verification
13) Generate constrained random test values
14) Using coverage with constrained random tests
15) Simulate a simple UVM testbench and DUT
16) Examining the UVM testbench
17) Design and simulate sequence items and sequence
18) Design and simulate a UVM driver and sequencer
19) Design and simulating UVM monitor and agent
20) Design, simulate and examine coverage
21) Design and simulate a configurable UVM test environment

****