CND 212
Digital Testing and Verification
2023

Course Description

The main objective of this course is to cover the verification tools and methods, with emphasis on SystemVerilog and UVM, for digital circuits and systems with a focus on Hardware verification techniques. The course will cover SystemVerilog concepts such as Data types, Function, Tasks, threads, interprocess communication, Interfaces, randomization, Code Coverage, Functional Coverage and Assertion Based Verification. In addition to OOP and Hierarchal Testbenchs. Also, writing UVM verification projects including components such as transaction, generator, configuration, sequencer, driver, monitor, scoreboard, agent, environment, test. In addition, faults in digital systems, test generation and testable systems, and pattern generation and comparator circuits of the built-in-self-test.

Contact Hours

Credit Hours	Lecture Hours	Lab Hours	Student work	Total
6	24 (1.25x2)/week	21 (3x1)/week	48	93

Prerequisites

Introduction to Digital Design

Learning Outcomes

After successful completion of this course, the student will be able to:

1. Understand the main techniques of functional verification of digital systems.
2. Use SystemVerilog to describe and verify digital systems
3. Be able to analyze source codes and outputs of verification tools
4. Master creating verification environments in SystemVerilog language according to UVM verification methodology.
5. Define the relationship between fault models and physical failures,
6. Design efficient logic and fault simulators,
7. Generate relevant test patterns using different techniques
8. Distinguish various design-for-testability methods,

Course Materials

- Ashok Mehta, “ASIC/SoC Functional Design Verification A Comprehensive Guide to Technologies and Methodologies”, Springer.
- Chris Spear, Greg Tumbush, “SystemVerilog for Verification: A Guide to Learning the Testbench Language Features”, 3rd Edition, Springer.
- Ray Salemi, “The UVM Primer: A Step-by-Step Introduction to the Universal Verification Methodology”, Boston Light Press.
- Vanessa R. Copper, “Getting Started with UVM: A Beginner’s Guide”, Verilab Publishing.
- J. Aynsley, D. Long, D. Smith, M. Smith, B. Jensen, M. Jones, “Doulos UVM Golden Reference Guide”, Doulos Ltd.
- Lun Li, Mitchell A. Thornton, Digital System Verification, A Combined Formal Methods and Simulation Framework, Springer
- Mark Zwolinski, Digital System Design with SystemVerilog, Prentice Hall.
- N. K. Jha, S. Gupta, Testing of Digital Systems, Cambridge University Press
- M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*, Springer.
- L. Wang, C. Wu, X. Wen, *VLSI Test Principles and Architectures: Design for Testability*, Morgan Kaufmann Publishers.

CAD Tools: Synopsys flow including

- Synopsys VCS,
- Synopsys Verdi

Course Project: By the end of this course the students are required to deliver a complete project (chosen from variety of proposals) assigned by the industry experts and university professors instructors.

Course Schedule

Week	Lecture	Lab
1	Introduction, Motivation and history of verification	Introduction to systemverilog, Data types (Basic, advanced) Enum’s, typedefs, union
2	Introduction to Verification Process and methodologies (Directed Vs Random, Functional verification process, Stimulus Generation, Bus functional model, Monitors and reference models, Coverage Driven Verification, Verification Planning and management)	Arrays, queues, associative arrays

3	Introduction to SystemVerilog (New Data types, Tasks and Functions, Interfaces, Clocking blocks)	SystemVerilog Functions/Tasks, Fork/join, semaphore,
4	Object Oriented Programming and Randomization (OOP Basics, Classes - Objects and handles, Polymorphism and Inheritance, Randomization, Constraints)	OOP concepts, mailbox, classes (types), Randomization
5	Threads and Virtual Interfaces (Fork Join, Event controls, Mailboxes and semaphores, Virtual Interfaces, Transactors, Building verification environment, Test cases, callbacks)	Interface, packages, mod ports
6	Functional Coverage (Coverage models, Coverpoints and bins, Cross coverage, Regression testing)	SV Assertions
7	Universal Verification Methodology-1 (UVM overview and UVM testbench in details)	Code and Functional Coverage
8	Universal Verification Methodology-2 (UVM TLM: tlm ports,tlm Fifo, analysis ports and fifos…)	Randomization
9	Universal Verification Methodology-3 (UVM use case for complete testbench implementation: from building to simulation)	UVM 1 (introduction, UVM base class library, hierarchal testbench components
10	Approaches to Testing (Fault Simulation, Analysis of a Faulted Circuit, The Stuck-At Fault Model)	UVM 2 (Configuration, database, factory, Phasing, Reporting, objections)
11	Design for Testability: scan chain, controllability and observability, techniques and Automatic Test Pattern Generation, testability trade-offs, board-level and system-level approaches, Boundary Scan standards	UVM 3 (Transaction level modeling)