1. **MODULE AIM**

This module prepares the students to design and develop functional verification environments for multimillion gate designs. The students will be introduced to object oriented programming, constrained random generation and assertions based on Verification Methodology Manual (VMM). Design and development of full chip functional verification environment for multimillion gate designs will be taught with relevant case studies. Students will be trained on the use of industry standard Synopsys and Cadence tools for full chip functional verification.

2. **TEACHING AND LEARNING**

**Intended Module Learning Outcomes**

After undergoing this module students will be able to:

1. Explain the functional verification methodologies for multimillion gate designs
2. Design and develop full chip functional verification environment using object oriented programming, constrained random generation and assertions based on VMM
3. Apply the various verification methodologies developed for verifying multimillion gate designs
4. Proficiently use industry standard Synopsys and Cadence tools for full chip functional verification

**Module Content**

1. **HVL for functional Verification:** Introduction to Design Verification-Need for verification, verification process, Challenges in verification, cost involved, time involved, Methodologies and techniques, mission and goals of verification, plans, verification flow, verification guidelines, HDL, HVL, HDVL, and System. Verilog for design and verification, verification methodology manual, Low power verification techniques. Introduction to System Verilog - System Verilog enhancements to Verilog 2001, Generations of System verilog standard and enhancements for hardware design, System Verilog dot name and dot star enhancements, Case study on instantiation example for complex ALU, System Verilog Verification Features, Built-in data types and operators, user defined types, enumerated data types, System Verilog 2, state and 4 state data types, static and automatic variables, verification , advantages of 2 state date types, synthesis guidelines, structure declarations, assigning values, passing structures through ports, synthesis guidelines, packed and unpacked arrays, unions and packed unions, dynamic & associative arrays, Case study on verification of memory, SystemVerilog Procedural and specialised procedural Blocks, sequential logic procedural blocks, task and function enhancements, passing task/function, System Verilog procedural statements for design verification, Enhanced for loops, do while loops, continue and break, unique if else and priority if else, Case studies on complex FSM implementations

2. **Advanced HVL for functional Verification:** Object Oriented Programming for Verification, OOP terminologies, local and global variables, scoping rules, System Verilog class data type, Defining class objects, public Vs private Class methods inheritance, Data hiding, building an object oriented test bench, Case studies on OOP System Verilog assertions and interfaces, Assertions in System Verilog, assertion concepts, immediate and concurrent assertions, Case studies on assertion based verification, System Verilog interfaces, verification with interfaces, Case study on ATM router interface, System Verilog randomization, Verification strategy using VMM, constraint details, random control, Random generators, random device configuration, agent, scoreboard, checker, driver, monitor and other functional layers, building a complete verification environment, Case study, SystemVerilog functional coverage,Introduction, coverage types, functional, coverage strategies, simple functional coverage example, cover group, coverage options, analysing coverage data, measuring coverage statistics during simulation, Case study on functional coverage’s
3. MODULE RESOURCES

Essential Reading

1. Module Notes

Recommended Reading

Books


Journals

IEEE Transactions on Computer-Aided Design of Integrated circuits and systems

Internet Sites

1. wwwverificationguild.com (accessed on 18th January 2012)
2. www.systemverilog.org (accessed on 18th January 2012)
3. www.electrosofts.com (accessed on 18th January 2012)

Software Tool: Synopsis Tools (ICC Compiler), Cadence Tools (RTL Encounter)