20VL018 - VERIFICATION METHODOLOGIES

Course Learning Objective

- To expose the students to all aspects of functional verification of digital systems
- To introduce verification of hardware designs.
- To provide a practical approach for verification of designs.
- To give an introduction to FPGA based verification and Emulation of VLSI systems.
- To study the basic concepts of system verilog.
- To Study the basic concepts of OOPs

Course outcomes:

Upon successful completion of this course, students should be able to:

CO1: To learn about the C++ and testing environment of digital systems

CO2: To create test benches for digital systems

CO3: Model a scenario for Verification of a DUT in System Verilog

CO4: Understand the concept of randomization and its importance in verification coverage in a bigger design.

CO5: To develop Register -Register Model Integration, Registers–Memory Allocation Manager
CO6: Implementation of UVM was derived to enable better verification platform.

UNIT-1

Getting Started with C++, Setting Out to C++, dealing with Data, Compound Types, Loops and Relational Expressions, Branching Statements and Logical Operators, Functions: C++'s Programming Modules, Adventures in Functions, Memory Models and Namespaces, Objects and Classes, working with Classes, Classes and Dynamic Memory Allocation, Class Inheritance, Reusing Code in C++, Friends, Exceptions, The string Class and the Standard Template Library, Input, Output, and Files.

UNIT-2

System Verilog (SV) - Data Types, Arrays, Structures, Unions, Procedural Blocks, Tasks & Functions, Procedural Statements, Interfaces, Basic OOPs, Randomization, Threads & Inter Process Communication, Advanced OOPs & Test bench guidelines, Advanced Interfaces. A Complete System Verilog Test Bench (SVTB), Functional Coverage in System Verilog, Interfacing with C, FSM Modeling with SV, Connecting Test bench & Design, Behavioral& Transaction Level Modeling with SV.

UNIT-3

System Verilog Assertions (SVA) – Introduction to SVA, Building blocks, Properties, Boolean expressions, Sequence, Single & Multiple Clock definitions, Implication operators (Overlapping & Non-overlapping), Repeatition operators, Built-in System functions ($past, $stable, $onehot, $onehot0, $isunknown), Constructs (ended, and, intersect, or, first_match, throughout, within, disableiff, expect, matched, if –else), assertion directives, nested implication, formal arguments in property.

UNIT-4

UVM overview, OOP, UVM Library Basics, Interface UVCs, Automating UVC creation, Simple Test Bench Integration, Stimulus Generation Topics.

UNIT-5

Register and Memory Package, System UVCs and Test Bench integration,

Text Books

1. SystemVerilog for design: a guide to using SystemVerilog for hardware design and modeling By Stuart Sutherland, Simon Davidmann, Peter Flake Edition: illustrated Published by Springer, 2004 ISBN 1402075308, 9781402075308
2. Stephen Prata ,“C++ Primer Plus”, 2012 Pearson Education, Inc.
3. System Verilog for Verification: A Guide to Learning the Test bench Language Features By Chris Spear Edition: 2, Published by Springer, 2008 .
4. A Practical guide for System Verilog Assertions By Srikanth Vijayaraghavan&MeyyappanRamanathan Published by Springer, 2005.
5. Sharon Rosenberg, Kathleen A Meade “A practical guide to adopting the Universal Verification Methodology”,2010.

Reference Books

1. Writing testbenches using System Verilog By Janick Bergeron Edition: illustrated Published by Birkhäuser, 2006 .
2. SystemVerilog Assertions Handbook: --for Formal and Dynamic Verification - Ben Cohen, cohen, Venkataramanan, Kumari, Srinivasan Venkataramanan, AjeethaKumari - Published by vhdlcohen publishing, 2005 .
3. An Integrated Formal Verification solution DSM sign-off market trends, www.cadence.com.
4. Recent literature in Functional Verification using Hardware Verification Languages.