ECE6024	VLSI Verification Methodologies	L	T	P	J	C
		2	0	0	4	3
Pre-requisite	ECE5017 Digital Design with FPGA	Syllabus version				
		1.0				

Course Objectives:

1. To introduce various verification techniques.
2. To write Test bench using System Verilog.
3. To develop UVM test bench environment

Expected Course Outcome:

The students will be able to :
1. Comprehend the VLSI verification techniques.
2. Define classes and create objects.
3. Develop System Verilog modules.
4. Make Verification environment using System Verilog.
5. Cognize the UVM Verification environment.
6. Create reusable verification environment using UVM.

Student Learning Outcomes (SLO):	1,14,17

1. Having an ability to apply mathematics and science in engineering applications
14. Having an ability to design and conduct experiments, as well as to analyze and interpret data
17. Having an ability to use techniques, skills and modern engineering tools necessary for engineering practice.

Module:1	Verification Techniques	4 hours

Introduction to Verification - Testing Vs Verification - Verification Technologies - Functional Verification- Code coverage – Functional coverage. Testbench – Linear Testbench - Linear Random Testbench - Self-checking Testbench – Regression - RTL Formal Verification.

Module:2	Basic OOP	3 hours

OOP Terminology, Creating Object, object deallocation, copying objects, static variables, Global variables, Inheritance, Polymorphism

Module:3	System Verilog – Data Types & Procedural statements	6 hours

Introduction to System Verilog – Literal values-data Types – Arrays – Array methods – Creating new types with typedef – user defined structures – Enumerated types – attributes - operators – expressions - Procedural statements and control flow - Processes in System Verilog – Task and functions – Routine arguments – Returning from a routine

Module:4	Connecting Testbench and Design	3 hours

Program, Interface, Stimulus timing, Module interactions, Connecting together, Development of self-checking test environment – Generator, Transactor, Driver, Monitor, Checker, Scoreboard

Module:5	Randomization, Assertion and Coverage	3 hours

Randomization in system Verilog, Constraints, Functional coverage, cross coverage, cover groups, Assertions

Module:6	Universal Verification Methodology	4 hours

Introduction to UVM - Verification components - Transaction level modeling

Module:7	**UVM – Verification Environments**	**5 hours**
Developing reusable verification components - Using Verification components – Developing reusable verification environment – Register classes.		
Module:8	**Contemporary issues:**	**2 hours**
	Total Lecture hours:	**30 hours**

Reference Books:

1. Vanessa R. Copper, "Getting started with UVM: A Beginner's Guide", Verilab Publishing, First Edition, 2013.
2. Ray Salmei, "The UVM Primer: A Step-by-Step Introduction to the Universal Verification Methodology" Boston Light Press; First edition, 2013.
3. Christian B Spear, "System Verilog for Verification: A guide to learning the Testbench language features", Springer publications, Third Edition, 2012.
4. Janick Bergeron, "Writing Testbenches using System Verilog" Synopsys Inc., Springer Publications, 2006.

Mode of Evaluation: CAT / Assignment / Quiz / FAT / Project / Seminar

List of Challenging Projects (Indicative) **CO: 4, 6**

1. Develop a system Verilog testbench to verify your DUT by following the steps given below.
   i) Write the following blocks in system Verilog to verify your design
   a. Program Block
   b. Interface Block with clocking block and modport
   c. Top Level Harness file which has the instance of your DUT, test program and the interface.
   ii) Develop the Generator, Transactor and Driver components for your DUT
   iii) Develop the self-checking feature by writing the receiver, monitor and checker components for your DUT.
   iv) Simulate and verify the output.

2. Define a packet class to encapsulate the packet information and create random packet objects in the generator then send, receive and check the correctness of the DUT using the packet objects for the given router IP. Follow the instructions given in the lab to complete the task. Simulate and verify the output for the good RTL code and the faulty code. Include covergroups and check the functional coverage is greater than 90%.

Mode of evaluation:

Recommended by Board of Studies	05-03-2019		
Approved by Academic Council	No. 54	Date	14.03.2019