Subject Code	Subject Name	Teaching Scheme			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/Practical	Tutorial	Total
EXC7053	ASIC Verification	04	--	--	04	--	--	04

Subject Code	Subject Name	Examination Scheme							
		Theory Marks				Term Work	Practical	Oral	Total
		Internal assessment			End Sem. Exam				
		Test 1	Test 2	Avg. of Test 1 and Test 2					
EXC7053	ASIC Verification	20	20	20	80	--	--	--	100

Course Pre-requisite:

- EXL304: Object Oriented Programming Methodology Laboratory
- EXC303: Digital Circuits and Design

Course Objectives:

1. To teach ASIC Verification fundamentals
2. To highlight the significance of verification in VLSI industry

Course Outcomes:

After successful completion of the course student will be able to

1. demonstrate an understanding of programmable devices and languages
2. demonstrate an understanding of verification process in VLSI systems
3. write system verilog code for VLSI systems
4. carry out verification of design successfully using simulators

Module No.	Unit No.	Topics	Hrs.
1		**Programmable Devices and Verilog**	08
	1.1	**Programmable Devices:** Architecture of FPGA, CPLD with an example of Virtex-7 and Spartan -6 family devices	
	1.2	**Verilog HDL:** Data types, expressions, assignments, behavioral, gate and switch level modeling, tasks and functions	
	1.3	**Verification Basics:** Technology challenges, Verification methodology options, Verification methodology, Testbench creation, testbench migration, Verification languages, Verification IP reuse, Verification approaches, Verification and device test, Verification plans, reference design of Bluetooth SoC, Verification Guidelines	
2		**Data types, procedural statements and testbench**	08
	2.1	**Data Types:** Built in, Fixed size array, dynamic array, queues, associative array, linked list, array methods, choosing a storage type, creating new types with typedef, creating user-defined structures, type conversion, enumerated types, constants, strings, expression width	
	2.2	**Procedural Statements and Routines:** Procedural statements, tasks, functions and void functions, task and function overview, routine arguments, returning from a routine, local data storage, time values	
	2.3	**Connecting the Testbench and Design:** Separating the testbench and design, the interface construct, stimulus timing, interface driving and sampling, connecting it all together, top-level scope, program-module interactions, system verilog assertions, the four port ATM router, the ref port direction, the end of simulation, directed test for the LC3 fetch block	
3		**OOP and Randomization**	10
	3.1	**Basic OOP:** Class, Creating new objects, Object deallocation, using objects, variables, class methods, defining methods outside class, scoping rules, using one class inside another, understanding dynamic objects, copying objects, public vs. local, building a testbench	
	3.2	**Randomization:** Randomization in system Verilog, constraint details, solution probabilities, controlling multiple constraint blocks, valid constraints, In-line constraints, The pre-randomize and post-randomize functions, Random number functions, Constraints tips and techniques, common randomization problems, Iterative and array constraints, Atomic stimulus generation vs. scenario generation, random control, random number generators, random device configuration	
4		**IPC and advanced OOP**	08
	4.1	**Threads and Interprocess Communication:** working with threads, disabling threads, interprocess communication, events, semaphores, mailboxes, building a testbench with threads and IPC	
	4.2	**Advanced OOP and Testbench Guidelines:** Inheritance, Blueprint pattern, downcasting and virtual methods, composition, inheritance and alternatives, copying an object, abstract classes and pure virtual methods, callbacks, parameterized classes	
5		**Assertions and Functional Coverage**	12
	5.1	**System Verilog Assertions:** Assertions in verification methodology, Understanding sequences and properties, SystemVerilog Assertions in the Design Process, Formal Verification Using Assertions and SystemVerilog Assertions Guidelines	
	5.2	**Functional Coverage:** Coverage types, strategies, examples, anatomy of a cover group, triggering a cover group, data sampling, cross coverage, generic cover groups, coverage options, analyzing coverage data, measuring coverage statistics during simulation	
6		**Advanced interfaces and interfacing with C**	6
	6.1	**Advanced Interfaces:** Virtual interfaces with the ATM router, Connecting to multiple design configurations, procedural code in an interface	
	6.2	**A complete System Verilog Testbench:** Design blocks, testbench blocks, alternate tests	
	6.3	**Interfacing with C:** Passing simple values, connecting to a simple C routine, connecting to C++, simple array sharing, open arrays, sharing composite types, pure and context imported methods, communicating from C to system verilog, connecting other languages	
		Total	**52**

Recommended Books:

1. Chris Spear, “*System Verilog for Verification: A guide to learning the testbench language features*”, Springer, 2nd Edition
2. Stuart Sutherland, Simon Davidmann, and Peter Flake, “*System Verilog for Design: A guide to using system verilog for hardware design and modeling*”, Springer, 2nd Edition.
3. Ben Cohen, Srinivasan Venkataramanan, Ajeetha Kumari and Lisa Piper, “*SystemVerilog Assertions Handbook*”, VhdlCohen Publishing, 3rd edition
4. *System Verilog Language* Reference manual
5. S Prakash Rashinkar, Peter Paterson and Leena Singh, “*System on Chip Verification Methodologies and Techniques*”, Kluwer Academic, 1st Edition.
6. *Spartan and Virtex family* user manuals from Xilinx
7. *Verilog Language* Reference manual

Internal Assessment (IA):

Two tests must be conducted which should cover at least 80% of syllabus. The average marks of both the test will be considered as final IA marks

End Semester Examination:

1. Question paper will comprise of 6 questions, each of 20 marks.
2. Total 4 questions need to be solved.
3. Question No.1 will be compulsory and based on entire syllabus wherein sub questions of 2 to 5 marks will be asked.
4. Remaining question will be selected from all the modules.