Technical Examinations Board, Gujarat State, Gandhinagar

System Verilog with Project

Title	SEMI104: System Verilog with Project
Level	Certificate Course
Course Duration	90 Hrs (30 Hrs Theory + 20 Hrs Practical + 40 Hrs Project) Ten Weeks (Part time) Three Weeks (Full Time)
Entry Qualification	B.E./B.Tech. Sem-3 onwards (EC / IC / CE /Electrical /Mechanical/ IT / ICT or Similar Branch) / B.Sc./M.Sc. (with Physics/IT/ Electronics) / BCA/MCA / Diploma

Teaching Scheme:

Sub Code	Subject Name	Teaching Scheme		Examination Scheme				Term Work Marks	Total Marks
		Theory	Practical	Theory Marks	Hrs.	Practical Marks	Hrs.		
SEMI104	System Verilog with Project	3	6 (With Project)	50	2	100	3	25	175

Total Week	= 10	Theory	= 1 hour slot
Total Teaching slot/Week	= 06	Practical	= 2 hour slot
Theory Periods	= 30	Total teaching 9 hours/week (Part-time) 6 hours/day (Full time)	
Practical Periods	= 30		

System Verilog

Verification is the process of ensuring that a given hardware design works as expected specifcations. Chip design is a very extensive and time-consuming process and costs millions to fabricate. Functional defects in the design if caught at an earlier stage in the design process will help save costs. System Verilog offers more compact code compared to Verilog, has functionality like structures, enumerated types for better scalability, and Interfaces for higher level of abstraction.

This course gives you an in-depth introduction to the main SystemVerilog enhancements to the Verilog hardware description language (HDL), discusses the benefits of the new features, and demonstrates how design and verification can be more efficient and effective when using System Verilog constructs. This course contains data types, procedural block, Object Oriented Programming, random constraint, assertion etc.

Course Objectives:

The goal of this course is to make student familiar with the language and the verification processes carried out in industry. This course makes student write SystemVerilog code to describe practical digital logic functions, intuitively and concisely. It makes student utilize new syntax like typedef, struct, enum, etc. to customize the code to application-specific chip architectures or data-packet formats. It prepares student add assert statements to check key design properties, and to develop reusable testbench code for simulating logic functions or bus operations, including defining a class of objects, calling its methods, constraining random stimuli, and using interface connections. Student will have exposure to simulation tools being used in industry, will get hands on experience and will be prepared for industry for verification in VLSI.

SEMI104: System Verilog with Project	
Unit - 1	**Introduction to System Verilog (SV)**
1.1	ASIC Design System Level Languages
1.2	Why SV?
1.3	SV Methodology Overview and LRM detail
1.4	SV a supper set of Verilog-2001
1.5	SV Verification flow overview, TB Architecture, TB Components (Driver, Monitor and Scoreboard)
Unit - 2	**SV Data Types and usage in Design and Verification**
2.1	2/4 state data types
2.2	Typedef, unresolved and resolved
2.3	SV Array and union
Unit - 3	**Overview of Procedural Block and Control operational flow**
3.1	Procedural Block Flow, Fork Join inter process synchronization
3.2	Control Block Flow
Unit - 4	**Overview of Object-Oriented Programming**
4.1	Object Orientation and SV
4.2	Syntax and basic example with SV object, Class coding and constructor and, Object (Class instance and usage) with "this", Parameterised class, Simple OOPs based Example
Unit - 5	**SV Class type and extension (Inheritance) with example**
5.1	Static Class and Method
5.2	Extend Class – purpose and usage
5.3	Scope Resolution Operators
Unit - 6	**SV Polymorphism and other OOPS features**
6.1	Virtual function in base class
6.2	Class extended for Polymorphism with example
6.3	Override members i.e. override object of sub-class with parent class
6.4	Supper, Casting and chaining Data encapsulation etc.
6.5	Constant and Abstract class, Call Backs (Inserting Call backs, Registering Call backs)
Unit - 7	**SV Random Constraint**
7.1	Constraint Block, Randomise methods, Disable Random constraint
7.2	In line constraint random variable
7.3	Dynamic Constraints
7.4	Scope variable and randomization within class
7.5	Random Number generation (urandom, urandom_range, etc.)
7.6	Seeding and weighed case
7.7	Random Sequence
Unit – 8	**SV Interface and Clocking Block**

8.1	interface, modports, Specify block, Parameterised and Virtual interface, Interface object, Example with standard clock generation and data checks
8.2	Clocking block, Clocking block basic and input sampling, Example of Clocking blocks with program and interface
Unit – 9	**SV Assertions**
9.1	Immediate/ In-line assertions
9.2	Concurrent assertions
9.3	Constant Expressions and sequences
9.4	System Functions $onehot, $isunknown, etc.
9.5	Declare Property
9.6	Expect sequence
Unit - 10	**Complete SV Verification example code**
10.1	Explaining one code of complete verification using SV
Project (40 Hrs)	

Suggested List of Practical's

Sr. No	Practical Name
1	Write example code using each of the System Verilog Data Types
2	Write example code for a multi-dimensional array, Packed & Unpacked array, dynamic array and associative array
3	Write example code for Queue, structure
4	Design class for Half adder, Full adder. Make use of class object
5	Other based on syllabus & student

Reference books:

- Chris Spears, System Verilog for Verification, 2nd Edition, Springer, 2008.
- Vijayaraghavan, Srikanth, and Meyyappan Ramanathan. A practical guide for SystemVerilog assertions, Springer Science & Business Media, 2006.
- Bergeron, Janick. Writing testbenches using SystemVerilog, 1st Edition, Springer Science & Business Media, 2007.

Software/Tool list:

- Xilinx Vivado
- EDA Playground

Subject Course Committee

Prof. C. H. Vithalani, Prof. P. J. Brahmbhatt, Prof. T. P. Chanpura, Prof. M. S. Dave, Prof. P. B. Bhatt, Prof. J. A. Dhumale, Prof. A. S. Patel, Prof. A. K. Konkani