Technical Examinations Board, Gujarat State, Gandhinagar

System Verilog

Title	SEMI103: System Verilog
Level	Certificate Course
Course Duration	90 Hrs (60 Hrs Theory + 30 Hrs Practical) Fifteen 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.		
SEMI103	System Verilog	4	2	100	3	50	2	25	175

Total Week	= 15	Theory	= 1 hour slot
Total Teaching slot/Week	= 05	Practical	= 2 hour slot
Theory Periods	= 60	Total teaching 6 hours/week (Part-time) 6 hours/day (Full time)	
Practical Periods	= 15		

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:

This course provides working knowledge of the Verilog. It explains design, test and implementation of digital hardware, the hierarchy and modelling of structures. 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.

SEMI103: System Verilog	
Unit - 1	**Introduction to VLSI**
1.1	Introduction to VLSI Design, Overview of Very Large-Scale Integration, Importance & applications of VLSI technology, Challenges and trends in VLSI design
1.2	Introduction to Verilog, History and evolution of Verilog, modules, ports, & data types, Verilog design flow
1.3	What is HDL?
1.4	Most Commonly used HDLs
1.5	What is Verilog?
1.6	Difference between VHDL and Verilog
1.7	What is usage of Verilog?
1.8	Application Specific IC (ASIC) vs Field Programmable Gate Array (FPGA)
1.9	RTL Design Methodologies
Unit - 2	**Overview of Digital Design with Verilog HDL**
2.1	Hierarchical Modelling Concepts, Top-Down Design, Bottom Down Design, Example, Four Bit Adder (Ripple Carry Counter), Verilog Modules, Instance, Testing Block (Stimulus), Levels of Abstraction
2.2	Basic Concepts, Lexical Conventions, Whitespace, Comments, Operators, Number Specification, Sized numbers, Unsized numbers, X or Z values, Negative Numbers, Identifiers and Keywords, Strings, Identifiers and key words
2.3	Data Types, Value Set, Nets
2.4	Registers, Vectors, Vector Part Select, Variable Vector Part Select
2.5	Integer, Real, and Time Register Data Types, Integer, Real, Time, Arrays, Memories, Parameters, Strings
2.6	System Tasks, displaying information, monitoring information, Stopping and finishing in a simulation
2.7	Compiler Directives, Define, include
Unit - 3	**Modules and Ports**
3.1	Modules
3.2	Ports, List of Ports, Port Declaration
3.3	Port Connection Rules, Inputs, Outputs, Inouts, Width matching, Unconnected ports, Example of illegal port connection
3.4	Connecting Ports to External Signals, connecting by ordered list, Connecting ports by Name
3.5	Hierarchical Names
3.6	Gate-Level Modelling, Gate Types, And / Or Gates, Buf / Not Gates, Bufif / notif
3.7	Array of Instances, Examples
3.8	Gate Delays, Rise, Fall, and Tum-off Delays, Rise delay, Fall delay, Tum-off delay
3.9	Min/Typ/Max Values, Min value, Typ val, Max value
Unit - 4	**Data Flow Modelling**
4.1	Continuous Assignments, Implicit Continuous Assignment, Implicit Net Declaration
4.2	Delays, Regular assignment delay, Implicit continuous Assignment delay, Net declaration delay
4.3	Expressions, Operators and Operands, Expressions, Operators, Operands

4.4	Operator types, Arithmetic Operators, Binary operators, Unary operators, Logical operators, Relational operators, Equality operators, Bitwise operators, Reduction operators, Shift operators, Concatenation operators, Replication operators, Condition operators, Operator precedence
Unit - 5	**Behavioural Modelling**
5.1	Structure procedures, Initial statement, Combined variable declaration & initialization, Combined port/data declaration & initialization, always Statement
5.2	Procedural Assignments, Blocking Assignments, Non-blocking Assignments, Application of non-blocking assignments
5.3	Timing controls and event scheduling, Delay-Based Timing Control, Regular delay control, Intra-assignment delay control Zero delay control, zero delay control, Event-Based Timing Control, Regular event control, named event control, Event OR Control, Level-Sensitive Timing Control
5.4	Conditional Statements, Multiway Branching, Case statement, Casex, casez, keywords
5.5	Loops, While loop, For loop, Repeat loop, forever loop
5.6	Sequential and parallel Blocks, Block types, Sequential blocks, Parallel blocks
5.7	Special feature of blocks, Nested blocks, Named blocks, Disabling named blocks
5.8	Generate blocks, Generate loop, Generate conditional, Generate case
Unit - 6	**Tasks and Functions**
6.1	Differences between Tasks and Functions, Task Declaration & Invocation
6.2	Task Examples, Use of input and output arguments, Asymmetric Sequence Generator
6.3	Functions, Function Declaration & Invocation, Function Examples, Parity calculation, Left/right shifter
6.4	Automatic (Recursive) Functions, Constant Functions, Signed Functions
6.5	Useful modelling technique, Procedural Continuous Assignments, assign and de-assign, force and release, force and release on registers, force and release on nets
6.6	Overriding Parameters, defparam Statement, Module Instance Parameter Value
6.7	Conditional Compilation and Execution, Conditional Compilation, Conditional Execution
6.8	Useful system tasks, File output, Opening a file, Writing to files, Closing files
Unit - 7	**FSM, Advanced Verilog & Coding Style**
7.1	Basic FSM structure, Moore Vs Mealy, Common FSM coding styles, Registered outputs
7.2	Advanced Verilog for Verification, System Tasks, Compiler directives, Internal variable monitoring, File input and output
7.3	Synthesis Coding Style, Unwanted latches, Synthesizable operators, RTL coding styles, Synthesis errors
Unit - 8	**Coverage**
8.1	What is Coverage in Verification?
8.2	Why is Coverage Important?
8.3	Code Coverage
8.4	Different types of code coverage, Statement/Branch Coverage, Expression/Condition Coverage, Path Coverage, Toggle Coverage, FSM Coverage

8.5	Functional Coverage
8.6	Exploring Functional Coverage Metrics
8.7	Coverage Analysis
8.8	Merits & Demerits of CC & FC
8.9	Assertion-Based Verification
8.10	Constrained-Random Testing
8.11	Formal Verification
Unit - 9	**Verilog for Synthesis and Verification**
9.1	Verilog Synthesis and Optimization, Overview of synthesis process, Writing synthesizable Verilog code, Optimizing Verilog code for area, speed, and power
9.2	Verification Techniques in Verilog, Introduction to verification methodologies
9.3	Case Studies and Projects, Real-world VLSI design projects using Verilog
Unit - 10	**Introduction to System Verilog (SV)**
10.1	ASIC Design System Level Languages
10.2	Why SV?
10.3	SV Methodology Overview and LRM detail
10.4	SV a supper set of Verilog-2001
10.5	SV Verification flow overview, TB Architecture, TB Components (Driver, Monitor and Scoreboard)
Unit - 11	**SV Data Types and usage in Design and Verification**
11.1	2/4 state data types
11.2	Typedef, unresolved and resolved
11.3	SV Array and union
Unit - 12	**Overview of Procedural Block and Control operational flow**
12.1	Procedural Block Flow, Fork Join inter process synchronization
12.2	Control Block Flow
Unit - 13	**Overview of Object-Oriented Programming**
13.1	Object Orientation and SV
13.2	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 - 14	**SV Class type and extension (Inheritance) with example**
14.1	Static Class and Method
14.2	Extend Class – purpose and usage
14.3	Scope Resolution Operators
Unit - 15	**SV Polymorphism and other OOPS features**
15.1	Virtual function in base class
15.2	Class extended for Polymorphism with example
15.3	Override members i.e. override object of sub-class with parent class
15.4	Supper, Casting and chaining Data encapsulation etc.
15.5	Constant and Abstract class, Call Backs (Inserting Call backs, Registering Call backs)

Unit - 16	SV Random Constraint
16.1	Constraint Block, Randomise methods, Disable Random constraint
16.2	In line constraint random variable
16.3	Dynamic Constraints
16.4	Scope variable and randomization within class
16.5	Random Number generation (urandom, urandom_range, etc.)
16.6	Seeding and weighed case
16.7	Random Sequence
Unit - 17	**SV Interface and Clocking Block**
17.1	interface, modports, Specify block, Parameterised and Virtual interface, Interface object, Example with standard clock generation and data checks
17.2	Clocking block, Clocking block basic and input sampling, Example of Clocking blocks with program and interface
Unit - 18	**SV Assertions**
18.1	Immediate/ In-line assertions
18.2	Concurrent assertions
18.3	Constant Expressions and sequences
18.4	System Functions $onehot, $isunknown, etc.
18.5	Declare Property
18.6	Expect sequence
Unit - 19	**Complete SV Verification example code**
19.1	Explaining one code of complete verification using SV

Suggested List of Practicals

Sr. No	Practical Name
1	Write down a Verilog code example using data types like reg, wire, integer, time, real, arrays, strings. Understand the difference between scalar and vector
2	Write down a Verilog code for 4-bit ripple carry adder using FA and HA with 3 different modelling techniques.
3	Write down a Verilog code using various operators like Arithmetic, Relational, Equality, Logical, Bitwise, Shift Operators.
4	Write example code using each of the System Verilog Data Types
5	Write example code for a multi-dimensional array, Packed & Unpacked array, dynamic array and associative array
6	Write example code for Queue, structure
7	Design class for Half adder, Full adder. Make use of class object
8	Other based on syllabus & student

Reference books:

- Samir Palnitkar, Verilog® HDL: A Guide to Digital Design and Synthesis, Prentice Hall PTR.
- Charles H. Roth, Jr. Lizy Kurian and John Byeong Kil Lee, Digital Systems Design using Verilog, Cengage Learning India, 2016.
- Frank Vahid and Roman Lysecky, Digital Design with RTL Design, Verilog and VHDL

- 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