Technical Examinations Board, Gujarat State, Gandhinagar

VLSI Verification

Title	SEMI106: VLSI Verification
Level	Certificate Course
Course Duration	140 Hrs (100 Hrs Theory + 40 Hrs Practical) Twenty Weeks (Part time) Seven 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.		
SEMI106	VLSI Verification	5	2	100	3	50	2	25	175

Total Week	= 20	Theory	= 1 hour slot
Total Teaching slot/Week	= 06	Practical	= 2 hour slot
Theory Periods	= 100	Total teaching 7 hours/week (Part-time) 4 hours/day (Full time)	
Practical Periods	= 20		

VLSI Verification

It is standard verification course which includes basic fundamentals of Electronics and Digital Design. The VLSI verification process is a meticulous examination of a chip's design before it's physically manufactured. It's like a rigorous final exam, ensuring the millions of transistors on the chip collaborate flawlessly. This multi-step process involves creating a virtual model of the chip and feeding it various test scenarios. Engineers analyze the model's behaviour to identify and rectify any errors or bugs in the design logic. Verification techniques like simulations and formal proofs guarantee the chip functions according to its intended specifications, catching potential issues early on to prevent costly mistakes and ensure a reliable final product. This course contains introduction to VLSI, concepts of Basic electronics, Digital electronics and Verilog and verification techniques using System Verilog and UVM.

Course Objectives:

This course aims to providing detailed knowledge in functional verification process starting from Basic Electronics, Digital Design, Hardware Descriptive Language, RTL, Synthesis & Simulation, Verification, TestBench, literals, tasks & functions, interfaces, etc. In this process the student will understand the entire logic process and will be able to take on the challenges posed by the even demanding VLSI verification industry. 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.

SEMI106: VLSI Verification	
Unit - 1	**Introduction to VLSI?**
1.1	What is VLSI?, Moore's law, Scaling, ITRS, Silicon
Unit - 2	**Semiconductors**
2.1	Insulators, Conductors, Semiconductors, Silicon Crystal, Intrinsic Semiconductors, Two Types of Flow, Doping a semiconductor, Two type of Extrinsic Semiconductors, Unbiased Diode, Forward Bias, Reverse Bias, Breakdown
Unit - 3	**Bipolar Junction Transistor (BJT)**
3.1	BJT Operation, BJT voltages and currents, CE, CB and CC characteristics, DC load line and bias point, Base bias
Unit - 4	**Transistor as switch**
Unit - 5	**FET and its biasing**
5.1	JFET, Comparison of BJT and JFET, JFET Characteristics, FET, Biasing in ohmic and active region, Transconductance, Amplification and switching, MOSFETs, CMOS introduction
Unit - 6	**MOSFET Operation and Characteristics**
6.1	MOS Cap Operation, MOSFET, NMOS/PMOS, IV Char, Non ideal Effects, Threshold Voltage, Switch, Capacitance in MOSFET
Unit - 7	**CMOS inverter**
7.1	Inverter, Layout, Noise Margin, RC Eq of NMOS/PMOS, CMOS Logic gates
Unit - 8	**Gates**
8.1	CMOS Logic gates, Logical Effort, Pass Transistor Logic
Unit - 9	**What is Digital and Analog Systems?**
9.1	Introduction to Digital Electronics and Applications, Number System and Code Conversion, BCD, Excess - 3, Gray Code, Complements, Logic Gates, Boolean Algebra, SOP, POS, K – Maps
Unit - 10	**Combinational Logic Design**
10.1	Adders, Subtractors, Comparator, Multiplexers, De-multiplexers, Encoder, Decoder, Parity Checker
Unit - 11	**Sequential Logic Design**
11.1	Difference between Combinational and Sequential circuits, Flip – Flop & Latch, Trigger Types, Level Triggered, Edge Triggered, Setup Hold Time, Max Delay, Min Delay, Clocked S – R Flip Flops, J – K Flip Flops, D – Flip Flops, T – Flip Flops, Master Slave Flip Flops
Unit - 12	**Shift Registers**
12.1	SISO, SIPO, PISO, PIPO, USR
Unit - 13	**Counters**
13.1	Asynchronous, Synchronous, Mod, Ring counter, Johnson (Twisted Ring counter)
Unit - 14	**FSM – Finite State Machine**
14.1	Mealy & Moore, Overlapping and Non-Overlapping
Unit - 15	**Memory**
15.1	6T RAM, FIFO, Read, Write, PLA/FPGA
Unit - 16	**Overview of Digital Design with Verilog**

16.1	Hierarchical Modelling Concepts, Top-Down Design, Bottom Down Design, Verilog Modules, Instance, Testing Block (Stimulus), Levels of Abstraction, Data Types, Value Set, Nets, Registers, Vectors, Vector Part Select, Variable Vector Part Select, Integer, Real, and Time Register Data Types, Integer, Real, Time, Arrays, Memories, Parameters, Strings, System Tasks, displaying information, monitoring information, Stopping and finishing in a simulation, Compiler Directives, Define, include
Unit - 17	**Modules and Ports**
17.1	Modules, Ports, List of Ports, Port Declaration, Port Connection Rules, Inputs, Outputs, Inouts, Width matching, Unconnected ports, Connecting Ports to External Signals, connecting by ordered list, Connecting ports by name, Hierarchical Names
Unit - 18	**Gate-Level Modelling**
18.1	Gate Types, And / Or Gates, Buf / Not Gates, Bufif / notif, Array of Instances, Gate Delays, Rise, Fall, and Tum-off Delays, Rise delay, Fall delay, Tum-off delay, Min/Typ/Max Values, Min value, Typ val, Max value
Unit - 19	**Data Flow Modelling**
19.1	Continuous Assignments, Implicit Continuous Assignment, Implicit Net Declaration Delays, Regular assignment delay, Implicit continuous Assignment delay, Net declaration delay, Expressions, Operators and Operands, Expressions, Operators, Operands, Operator precedence
Unit - 20	**Behavioural Modelling**
20.1	Structure procedures, Initial statement, Combined variable declaration & initialization, Combined port/data declaration & initialization, always Statement, Procedural Assignments, Blocking Assignments, Non-blocking Assignments, Conditional Statements, Loops, Generate blocks
Unit - 21	**Tasks and Functions**
21.1	Differences between Tasks and Functions, Task Declaration, Function Declaration, Parity calculation, Left/right shifter, Useful modelling technique, Procedural Continuous Assignments, assign and deassign, force and release, force and release on registers, force and release on nets, Overriding Parameters, defparam Statement, Module_Instance Parameter Value, Useful system tasks, File output, Opening a file, Writing to files, Closing Files
Unit - 22	**FSM, Advanced Verilog & Coding Style**
22.1	Basic FSM structure, Moore Vs Mealy, Common FSM coding styles, Advanced Verilog for Verification, System Tasks, Compiler directives, Synthesis Coding Style, Unwanted latches, Synthesizable operators, RTL coding styles, Synthesis errors
Unit - 23	**Coverage**
23.1	What is Coverage?, Why is Coverage Important?, Different types of code coverage, Statement/Branch Coverage, Expression/Condition Coverage, Path Coverage, Toggle Coverage, FSM Coverage, Functional Coverage, Exploring Functional Coverage Metrics, Coverage Analysis, Assertion-Based Verification, Constrained-Random Testing, Formal Verification
Unit - 24	**Introduction to System Verilog**
24.1	ASIC Design System Level Languages, Why SV?, SV Methodology Overview and LRM detail, SV a supper set of Verilog-2001, SV Verification flow overview, TB Architecture, TB Components (Driver, Monitor and Scoreboard)
Unit - 25	**SV Data Types and usage in Design and Verification**
25.1	2/4 state data types, Typedef, unresolved and resolved, SV Array and union

Unit - 26	**Overview of Procedural Block and Control operational flow**
26.1	Procedural Block Flow, Fork Join inter process synchronization, Control Block Flow
Unit - 27	**Overview of Object-Oriented Programming**
27.1	Object Orientation and SV, 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 - 28	**SV Class type and extension (Inheritance) with example**
28.1	Static Class and Method, Extend Class – purpose and usage, Scope Resolution Operators
Unit - 29	**SV Polymorphism and other OOPS features**
29.1	Virtual function in base class, Class extended for Polymorphism with example, Override members i.e. override object of sub-class with parent class, Supper, Casting and chaining Data encapsulation etc., Constant and Abstract class, Call Backs (Inserting Call backs, Registering Call backs)
Unit - 30	**SV Random Constraint**
30.1	Constraint Block, Randomise methods, Disable Random constraint, In line constraint random variable, Dynamic Constraints, Scope variable and randomization within
30.2	class, Random Number generation (urandom, urandom_range, etc.), Seeding and weighed case, Random Sequence
Unit - 31	**SV Interface and Clocking Block**
31.1	interface, modports, Specify block, Parameterised and Virtual interface, Interface object, Example with standard clock generation and data checks, Clocking block, Clocking block basic and input sampling, Example of Clocking blocks with program and interface
Unit - 32	**SV Assertions**
32.1	Immediate/ In-line assertions, Concurrent assertions, Constant Expressions and sequences, System Functions $onehot, $isunknown, etc., Declare Property, Expect sequence
Unit - 33	**Complete SV Verification example code**
33.1	Explaining one code of complete verification using SV
Unit - 34	**UVM Overview**
34.1	UVM hierarchy, UVM Packages, MACRO, Include and SV Interface, Example, Component and environment and test, Binding/Combining UVM Objects
Unit - 35	**UVM Transaction, Analysis and Reporting**
35.1	Transactions, Analysis, Reporting
Unit - 36	**UVM Sequences**
36.1	Basic Sequence, Connecting Sequence to test environment
Unit - 38	**Overview of UVM Test bench Architecture**
Unit - 39	**Advance UVM sequences**
39.1	Case and feeding of sequences, Layered sequence and other sequence
Unit - 40	**Factory and configuration**
Unit - 41	**Introduction to TLM**

Unit - 42	**Writing and adding multiple tests**
Unit - 43	**Register Modelling**

Suggested List of Practical's

Sr. No	Practical Name
1	Implementation of Boolean function using logic gates
2	Design and implementation of half and full adders and subtractors
3	Design and implementation of multiplexer and Boolean function using multiplexer
4	Design and implementation of an encoder and decoder
5	Write down a Verilog code example using data types like reg, wire, integer, time, real, arrays, strings. Understand the difference between scalar and vector
6	Write down a Verilog code for 4-bit ripple carry adder using FA and HA with 3 different modelling techniques.
7	Write down a Verilog code using various operators like Arithmetic, Relational, Equality, Logical, Bitwise, Shift Operators.
8	SV Write example code using each of the System Verilog Data Types
9	SV Write example code for a multi-dimensional array, Packed & Unpacked array, dynamic array and associative array
10	SV Write example code for Queue, structure
11	SV Design class for Half adder, Full adder. Make use of class object
12	UVM Simple Memory Testbench
13	UVM sequencer - Driver
14	U VM - DUT Interface, Generator, Driver, Monitor & Scoreboard
15	UVM - Coverage Example
16	Other based on syllabus

Reference books:

- Millman & Halkias, Electronic Devices and Circuits, McGraw Hill
- M. Morris Mano, Digital logic and Computer Design, Pearson Education India.
- Anand Kumar, Fundamentals of Digital Circuits, Prentice Hall India.
- 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.
- 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.
- SystemVerilog For Verification: A Guide to Learning the Testbench Language Features by Chris Spear & Greg Tumbush (3rd Edition)
- A Practical Guide For SystemVerilog Assertions by Srikanth Vijayaraghavan & Meyyappan Ramanathan SystemVerilog Assertions and Functional Coverage: Guide to Language, Methodology and Applications by Ashok B. Mehta
- A Practical Guide to Adopting Universal Verification Methodology (UVM) by Sharon Rosenberg & Kathleen A Meade (2nd Edition)
- The UVM Primer: A Step-by-Step Introduction to the Universal Verification Methodology by Ray Salemi Getting Started with UVM: A Beginner's Guide by Vanessa R. Copper

Software/Tool list:

- Xilinx Vivado
- EDA Playground
- Cadence xcelium

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