iMCT

iMobile Computing Technology

Incubated by Science and Technology Park, University of Pune, DST

Verilog HDL for IC Design and Verification

Course	**:**	**Verilog HDL for IC Design and Verification**
Duration	**:**	3-4 days (5-7hrs/day)
Dates	**:**	As per mutual agreement.
Time	**:**	**5 Hr/Day (Minimum)**
Eligibility	**:**	BE/BTech/ME/MTech (CSE / E&TC / Electronics), Third year, Final year students.

Objectives :

- Learn fundamental of digital and about IC design flow.
- Learn about Verilog language.
- Learn Modeling, Design, Verification concepts.

Results:

- At the end of training session you will be able to write your own Design and Verification environment in Verilog, Verilog is commonly used in most of the IC design industry compared with other HDL like VHDL. Students will get benefit of early knowledge before going to Industry.

Skills developed:

- Basics of Digital electronics
- IC design flow.
- Deign and Verification knowledge.
- Verilog as language.

Quick review of Digital Electronics

- Basic Semiconductor Theory
- Fabrication process – CMOS
- CMOS Transistors basics
- Digital Logic

Experiment : Design circuit using CMOS

- Boolean algebra
- Sequential and combinational logic
- IC design flow introduction

Verilog language detail study

- What is HDL?
- What is LRM
- Verilog HDL design flow.
- HDL modelling techniques
- Verilog History and lexical conventions
- Simulation and synthesis ?
- Verilog modelling structure
- Verilog models, Verification structure Specification, top level view of DUT and TB

LAB 1: How to interpret SPECIFICATION

Lexical Conventions

- Language essential (White space, identifiers, number, System task and function, compiler directive)

LAB 2 : `timescale

Basic code structure

- Module, port, instantiation
- Parameterise modules.
- 2001 and 95 style of module declaration.

LAB 3 : Module declaration

Verilog data types

- Net and Variable
- Resolution type
- Multidimensional arrays

LAB 4 : 2D array memory model

Procedural assignment and statements

- Procedural block overview
- initial block
- always block and sensitivity list
- assign statement
- Delay timing control

How simulator works.

- Event Queue and Delta
- Procedural block and synthesis issues.
- Blocking and non-blocking assignments with event queue
- Modelling delays in HDL (inertial and transport)

LAB 5 : and gate with inertial delay

LAB 6 : and gate with transport delay

Procedural statement

- If..else, case, forever, repeat, while, disable and synthesis issues.

LAB 7: 8 bit adder design and synthesis

- How to write combinational logic with always block and synthesis issues.

LAB8 - Design MUX using always with continuous assignment

LAB9 - Design MUX using assign

- Verilog gate level primitive.
- Function and task in combinational logic.
- Zero delay loops
- Latches in synthesis

LAB10 - Experience latches

- Example of async. Adder and subtracter

How to write sequential logic

- Types of sequential logic
- Basic DFF design with synchronous reset.
- DFF with a-synchronous reset.

LAB 11 : DFF with synchronous reset

LAB 12 : DFF with asynchronous reset

- Level sensitive models
- Shift registers and synthesis with blocking and non-blocking assignments.
- Example of synchronous adder and subtracter

LAB 13 : 8 bit counter design

Procedural calls : task and function

- User defines calls
- User define function, scope and synthesis issues.

LAB 14 : 7 segment display

- User define task
- Task and testbench example of adder.

Simulation timings

- Type of delays, flop-flop, combinational, interconnect delays.
- Timings at RTL modeling.
- Timings after synthesis
- Cell library design and specify block
- Netlist and SDF
- Post layout timings

LAB 15 : Counter synthesis and SDF annotation

Bi-directional and tri-state buffers

- tri in verilog
- SoC interconnecting bus architecture.
- tri-state implementation with simple bus example
- tri-state with assign statement.
- tri-state with always block

LAB 16 : Memory design with inout data bus

How to write FSM

- FSM types
- Mealy and Moore machine.
- Example of sequence detection circuit
- Design state machine.
- Write FSM (Sequential and combinational block of state mechine)
- Writing one-hot example of FSM

LAB 16: Sequence detector FSM design

How to write test benches

- What is TB
- Verification flow review.
- Advance verification architecture
- Step by step sequence detector testbench
- TB clock generation.
- TB reset generation.
- TB data generation and dummy model.
- TB comparator or scoreboard.
- Compiler directives used in TB
- Memory initialization used in TB
- $monitor and $strobe used in TB
- force release used in TB
- File IO in TB

LAB 17 – Write TB for FIFO

CPLD and FPGA

- Target technology
- Why programmable logic is required
- programmable logic fundamental understanding
- PAL, CPLD, FPGA architecture
- Spartan3an architecture overview
- Examples to use Sparten3an FPGA board.

LAB 18 – LED, LCD, Keyboard or switch control. Etc.

Advantages of training to college students, and Industry people.

1. It is necessary for all engineers to have knowledge about IC design and Verification flow, iMCT will insist students to take insight of VLSI world and learn most popular HDL language in industry.

2. This will be a common training activity for CSE and electronics students. It will be added advantage for their curriculum and they could get good opportunities in the best organizations to work with. Though this activity will be added advantage, it is not out of syllabus and scope for students.

3. Students can develop on chip advanced protocols and implement more design and verification specific methodologies.

4. iMCT is the 1st organization who is delivering such training programs based on industry projects and not available in the educational system till the date with such a low cost solutions.

5. iMCT will help in setting up lab IC design and this will help students to do experiments in house rather then going out for some institute.

6. Training will be given by iMCT's highly experience engineering resource.

Prerequisite:

- Machines installed with Window OS for students.
- ModelSim or QuestaSim, and any synthesis tool available in collage Lab.

NOTE:

- The training will be conducted by iMCT’s expertise and delivered to only college students and staff.
- Time for training can be extended if required.
- Training will be conducted in the premises of collage/University, it is expected that collage will provide required logistics like classroom, lab room with all prerequisite like computer, software, LCD projector.
- If students want to take training in iMCT/STP premises need to intimate 15 days before the training schedule.
- ***We have other courses RTOS with ARM 7, Mobile Computing Technology and mini OS, Advance topics like System-Verilog for chip Verification, Open Source Verification Technology for IC design. Please contact us for more details.***

Mrs. Prajakta Pathak
Head, iMCT, Pune
Mobile: 09611561828
Mail-id: prajakta@imct.in