

Accelerated Verilog Primer

Overview

Accelerated Verilog Primer is a 2-day workshop that provides a practical overview of the language is presented, with a focus on how Verilog is used for modeling hardware and for design verification using digital logic simulation. This workshop is a abridged version of our 4-day *Comprehensive Verilog and SystemVerilog for Design and Synthesis* workshop. The same general topics are covered, but this accelerated version does not go into as much depth on synthesis rules and the labs are shorter. In this accelerated workshop, engineers learn how to read and understand Verilog models, and how to use Verilog to verify correct model functionality. Several labs reinforce the principles presented, with about 25% of the class time devoted to lab work. and how to write small hardware models or to read and understand larger, more complex models. The workshop materials include a complete student guide and a useful “*Verilog HDL Quick Reference Guide*” (\$15 value).

Course Objectives

At the conclusion of this workshop, students will be able to read and understand Verilog code, and be able to write and small digital hardware models and testbenches. Students will know proper modeling styles and recognize subtle modeling style errors that can lead to simulation race conditions and poor synthesis results.

Intended Audience

This workshop is intended for verification engineers, engineering managers, technical writers and others who wish to understand how Verilog is used in today’s design flows. This accelerated workshop is *not* a replacement for—or a prerequisite for—Sutherland HDL’s 4-day *Comprehensive Verilog and SystemVerilog for Design and Synthesis* workshop. Engineers who will be creating Verilog models of complex hardware should attend the 4-day workshop instead of this accelerated workshop.

Prerequisites (essential)

Knowledge of digital design engineering is required. Without this background, students cannot fully benefit from this course. Labs include writing models of basic digital circuits such as shift registers and arithmetic logic units.

Presented By

Stuart Sutherland or Don Mills. Mr. Sutherland is an expert consultant in Verilog design, with more than fourteen years experience using the Verilog language and Verilog software products. Mr. Mills is an independent consultant with extensive experience in the top-down design, synthesis and verification of ASICs and FPGAs.

Software Tools Used

Students will use Verilog simulators for labs, such as Cadence *NC-Verilog*[™], Synopsys *VCST*[™], or Mentor Graphics *ModelSim*[™].

Quotes From Students

“I would strongly recommend this course to anyone that wants to learn more about Verilog, simulation, and modern design practices.”

Workshop Locations

This workshop can be presented on-site, at your facilities. We also offer several open-enrollment workshops throughout the year. For more information, please refer to our web page, www.sutherland-hdl.com, or call us at +1-503-692-0898.

(company names and product names are trademarks or registered trademarks of their respective companies)

Syllabus — Accelerated Verilog Primer Syllabus

Day One

Introduction to Verilog and SystemVerilog

- Concepts of top-down design
- Overview of RTL models
- Overview of gate/switch models

Design Verification Using Simulation

- Writing verification testbenches in Verilog
- Running your preferred Verilog simulator
- Debugging designs with simulation
- Lab: running your simulator and debug tools

Verilog HDL Syntax and Semantics

- Identifier names
- Logic values and numbers
- Data types
- SystemVerilog extensions to Verilog data types
- Exercise: selecting the correct data types

Procedures, Programming Statements and Operators

- Procedural blocks
- SystemVerilog enhanced procedural blocks
- Tasks and functions
- Continuous assignments
- Programming statements and operators
- Exercise: programming statement subtleties (“gotchas”)
- Lab: modeling a simple ALU and testbench

Day Two

Modeling Structural Netlists—After Synthesis

- Design hierarchy
- Module instantiation
- Generating arrays of instances
- Parameterized models and redefining parameters
- Verilog constructs used in ASIC/FPGA libraries
- Delay calculation and backannotation
- SDF files
- Lab: model and verify a hierarchical design

Modeling RAMs and ROMs

- Modeling memories
- Modeling bi-directional ports
- Testing bi-directional ports
- Timing constraints
- Lab: model and verify a dual-port RAM

Design Verification with Verilog

- Design documentation
- Using Verilog as a verification language
- Structured tests
- Configurable test benches
- Writing output files
- Reading test vector files
- Adding built-in error checking
- Lab: verify a design using test vectors