SUTHERLAND HDL

22805 SW 92nd Place., Tualatin, Oregon 97062 • Phone: +1-503-692-0898 • URL: www.sutherland-hdl.com

Verilog and SystemVerilog Language Foundations

Overview

Verilog and SystemVerilog Language Foundations is a fast-paced workshop designed to help engineers read, understand, and maintain digital hardware models and conventional verification testbenches written in Verilog and SystemVerilog. Critical language foundations are presented, including the proper use of data types, 2-state versus 4-state modeling, enumerated types, structures, and SystemVerilog's versatile set of programming statements. Hands-on labs reinforce the concepts discussed during the class lectures. This workshop provides a ***critical language foundation*** for more advanced training on SystemVerilog, including writing object-oriented verification testbenches, and coding SystemVerilog Assertions. ***Not just for beginners!*** Engineers who have years of experience with Verilog often comment regarding how much they learn and benefit from the details provided in this workshop.

Course Length: 2-days on-site, 2-days *eTutored™ live*, 2 to 30 days *eTutored™ self-paced*.

Intended Audience and Objectives

This workshop is ideal — and important — for three primary groups:

1. Engineers and managers who need to read and understand Verilog/SystemVerilog hardware models or verification testbenches, but who will not be writing large amounts of code.
2. Engineers who will be attending advanced training on verification. The knowledge gained in this workshop provides the prerequisite foundation for Sutherland HDL's *"SystemVerilog Object Oriented Verification"* and "SystemVerilog Assertions for Design Engineers and Verification Engineers" workshops.
3. Verilog users who need to learn the important enhancements SystemVerilog added to traditional Verilog.

(The information presented in this workshop can be combined with other workshops, and is integrated into Sutherland HDL's *"Verilog/SystemVerilog for Design and Synthesis"* workshop.)

Prerequisite Knowledge (essential)

Attendees should be familiar with general digital hardware engineering concepts such as the difference between combinational logic and sequential logic, binary and hexadecimal number systems, and binary operations.

Included Materials

- Full-color training binder with copies of all lecture slides, lab instructions, and supplemental information. (*eTutored™ self-paced* courses include an eBook instead of a training binder.)
- Lab files, including example solutions that illustrate proper and efficient coding styles.

Software Tools Used

Engineers will use both Verilog simulation and synthesis tools during class labs. The tools used can be the Aldec *Riviera-Pro*™, Cadence *NC-Verilog*™, Synopsys *VCS*™, or Mentor Graphics *ModelSim*™ simulator, and the Cadence *Encounter RTL Compiler*™, Synopsys *DC*™ or *SynplifyPro*™, or Mentor *Precision*™ synthesis compiler.

Comments From Students

"Excellent, wonderful class. Definitely worth the time. The instructor was extremely well prepared. He kept my interest the entire time."

Workshop Locations

This workshop can be presented on-site at your facilities or as an *eTutored™ live* online class. We also offer several public *eTutored™ live* workshops throughout the year. For more information, please visit *www.sutherland-hdl.com*.

Licensed Training Materials

Sutherland HDL's training materials can be licensed for use in internal training programs. Licensed materials include presentation PowerPoint files, printable PDF files, and lab files. Train-the-trainer services are also provided.

Syllabus — Verilog and SystemVerilog Language Foundations

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
- Lab: modeling a simple ALU and testbench

SystemVerilog User-defined Types and Packages

- User-defined types and enumerated types
- Structures and unions
- Casting
- Packages and $unit
- Lab: Model and verify an Instruction Stack

SystemVerilog Interfaces

- Using interfaces to simplify inter-module connections
- Specifying interface views (modports)
- Using tasks and functions in interfaces
- Interfaces as a verification construct
- Lab: model and verify a master/slave interface bus

"The SystemVerilog standard enables modeling larger, more complex designs, and, at the same time, simplifies writing models that synthesize correctly. Every design engineer learning Verilog should also learn the synthesizable portions of SystemVerilog, and know how to take full advantage of these powerful extensions to Verilog."

Stuart Sutherland, President of Sutherland HDL, Inc.