

SystemVerilog Assertions for Design and Verification Engineers

Overview

SystemVerilog Assertions for Design and Verification Engineers is an advanced workshop covering the IEEE 1800 SystemVerilog Assertions (SVA). SVA enables engineers to verify extremely complex logic using a concise, portable methodology. SystemVerilog Assertions offer improvements at every stage of design and verification process. This workshop provides a thorough examination of SVA and assertion-based verification methodologies. Both immediate and concurrent assertions are presented, with discussion on the appropriate usage of each type of assertion. SVA sequence and property blocks are covered in great detail, with a focus on the semantics and proper usage of the many sequence and property operators. The presentation materials are laden with practical examples of writing assertions for various types of hardware logic. Topics presented in this comprehensive study on SVA include the use of local variables, property and sequence arguments, multiple thread termination and uniqueness, assertion-based system functions, and using assertions with multi-clock designs. Several labs reinforce the principles presented, with forty percent of the class time devoted to hands-on experience.

Course length: 2-days on-site. 3-days *eTutored™ live*. 2 to 60 days *eTutored™ self-paced*.

Intended Audience and Objectives

This workshop is targeted towards both digital design engineers and digital verification engineers. The workshop is for experienced Verilog engineers. This workshop will enable design and verification engineers to immediately be productive with assertion-based verification methodologies and to write assertions and sequences that describe and verify complex design functionality.

Prerequisites (essential)

A working knowledge of SystemVerilog is essential in order to fully benefit from this workshop. In order to fully understand and utilize the concepts presented in this course, students should have first completed the Sutherland HDL *SystemVerilog Verification Foundations* course or equivalent.

Included Materials

- Comprehensive binder with all PowerPoint slides (printed in color), lab instructions, and supplemental white papers. (*eTutored™ self-paced* courses use an eBook and other online materials instead of a training binder.)
- The reference book “*SystemVerilog Assertions Handbook*” by Ben Cohen, et al. (\$150 value).
- A handy “*Verilog HDL Quick Reference Guide*” (\$15 value).
- All lab files, including example solutions that illustrate proper and efficient coding styles.

Software Tools Used

The Cadence *Incisive™*, Synopsys *VCS™* or the Mentor Graphics *Questa™* simulator will be used for labs.

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, or call us at +1-503-692-0898.

Syllabus — SystemVerilog Assertions for Design and Verification Engineers

Day One

Introduction to SystemVerilog Assertions (SVA)

- A quick look at SystemVerilog Assertions
- Software tools supporting SVA
- Lab: Running simulations with SVA

Assertion Based Verification (ABV) Methodologies

- The traditional design process
- Using SVA in the definition of designs
- Using SVA in the definition of verification
- Using SVA with reusable IP
- Using SVA to facilitate coverage metrics
- Case study: a synchronous FIFO

Overview of the SystemVerilog language

- Design hierarchy: modules, interfaces, packages
- Program blocks and clocking blocks
- Procedural blocks
- Enhanced tasks and functions
- SystemVerilog data types
- Enumerated types
- Lab

Overview of SVA Properties and Sequences

- Immediate and concurrent assertions
- The SVA property construct
- The SVA sequence construct
- When to use properties versus sequences
- Antecedent, consequent and threads
- Assertion, assumption and verification directives
- Lab

Understanding Properties

- Property declaration syntax
- Using formal arguments
- Local variables in properties
- Clocking events
- Disabling condition
- Property expressions
- Property operators
- Lab

Understanding Sequences

- Sequence operators and built-in functions
- Capturing temporal behavior
- Implication operators
- First match operator
- Repetition operators
- Sequence composition operators
- Sequence methods
- Lab

Day Two

Advanced Properties and Sequences

- Data types in properties and sequences
- Proper use of assertion overlapping
- Multiple thread termination
- Unbounded ranges in properties
- Emulating PSL-like constructs in SVA
- Lab

SVA System Functions and System Tasks

- Using the \$sampled system function
- Using the \$past, \$fell and \$stable system functions
- Vector analysis system functions
- Severity level system functions
- Assertion control system tasks
- Lab

Clocked and Multi-clocked Assertions

- Clock specification for properties and sequences
- Clock resolution
- Multiple clocked sequences
- Multiple clocked properties
- Lab

Binding SVA to Design Blocks

- The SVA bind construct
- Binding to all instances of a module or interface
- Binding to a single instance of a module or interface
- Verifying VHDL models using SVA
- Lab

Verification Directives & Verification-based Coverage

- The assert, assume and cover directives
- SVA coverage
- Coverage metrics
- Lab

Formal Verification Using SVA

- Formal verification methodology
- SVA in formal specifications
- Formal verification coverage metrics
- Formal verification versus dynamic simulation with SVA
- Case study: a traffic light controller

SVA Coding Guidelines

- Naming conventions
- Where to write properties and assertions
- Proper usage of the property negation operator
- Proper usage of local variables (write before read)
- Proper usage of unbounded ranges
- Using a default clock
- Using dynamic data types with assertions