**World Class Verilog & SystemVerilog Training**

# *Sunburst Design - Introduction to Verilog-2001 & Best Coding Practices*

by Recognized Verilog & SystemVerilog Guru, Cliff Cummings of Sunburst Design, Inc.

*Cliff Cummings is the only Verilog & SystemVerilog Trainer who helped develop every IEEE & Accellera Verilog, Verilog Synthesis and SystemVerilog Standard.*

**2 Days**
**60% Lecture, 40% Lab**
**Basic - Intermediate Level**

## Course Objective

Simply stated, to give engineers *world class* Verilog training using award winning materials developed by renowned Verilog & SystemVerilog Guru, Cliff Cummings

Upon completion of this course, students will:

- have a working understanding of the Verilog-2001 language with a focus on design, synthesis and verification
- understand keywords, syntax and semantics of Verilog-2001
- be able to write simple hardware models and testbenches and understand more complex models
- learn to use ASIC, FPGA and other vendor simulation libraries

## Course Overview

*Sunburst Design - Introduction to Verilog-2001 & Best Coding Practices* is a 2-day, fast-paced, introductory course on the IEEE 1364-2001 Verilog Hardware Description Language. The course presents a useful overview of the language with a focus on using Verilog for design, synthesis and simulation. A comprehensive 400+ page student guide and 49-page Verilog-2001 HDL Quick Reference Guide supplement the lecture and provide excellent resources for after-class reference. Several exercises and basic labs reinforce the principles presented.

This basic-level course helps students understand the Verilog-2001 language, Verilog netlists and Verilog software tools. Students will learn the keywords, syntax and semantics of the Verilog-2001 language. Students will also learn to write small hardware models and understand larger, more complex models. This course does not go into detail on all of the Verilog language subtleties or software tool specifics that are covered in the 4-day *Sunburst Design - Comprehensive Verilog-2001 Design & Best Coding Practices* training course. The

*Comprehensive* course provides an in-depth study of the Verilog language, simulation and fundamental synthesis coding styles.

### Target Audience

*Sunburst Design - Introduction to Verilog-2001 & Best Coding Practices* is intended for VHDL engineers and new and self-taught Verilog design and verification engineers that require a rapid introduction to the capabilities of Verilog-2001. This course also fulfills the prerequisites for Advanced and Expert Sunburst Design Verilog & SystemVerilog training courses and is usually offered as an optional first day for multi-day advanced or expert training.

### Prerequisites (mandatory)

A knowledge of digital design engineering. The ability to create files and efficiently use the editors on the operating system used in labs. Without the above skills, students cannot fully benefit from this course. Students will be writing Verilog models for basic and advanced digital circuits such as adders, multiplexers, flip-flops, and shift registers, barrel shifters and a simple DSP processor.

### The Sunburst Design - Advantage

Who is teaching your "expert" and "advanced" classes? Most companies will not tell you because their instructors might not have much design experience or may never have participated on any of the Verilog Standards groups or presented at industry recognized conferences. Go to our web site and read about the Sunburst Design - Instructors - they are simply the best at what they do and they have the experience and qualifications to offer best-in-class training.

### Sunburst Design Courses:

- *Sunburst Design - Advanced SystemVerilog for Design & Verification* - 4 days
    - *Sunburst Design - Advanced SystemVerilog for Design* - 3 days
    - *Sunburst Design - Advanced SystemVerilog for Verification* - 3 days
- *Sunburst Design - Expert Verilog-2001 for Synthesis & Verification* - 4 days
    - *Sunburst Design - Expert Verilog-2001 & Coding for RTL Design & Synthesis* - 2 days
    - *Sunburst Design - Expert Verilog-2001 Design, RTL Synthesis & Verification Techniques* - 2 days
- *Sunburst Design - Comprehensive Verilog-2001 Design & Best Coding Practices* - 4 days
    - *Sunburst Design - Introduction to Verilog-2001 & Best Coding Practices* - 2 days
    - *Sunburst Design - Advanced Verilog-2001 Knowledge & Design Practices* - 2 days
    - *Sunburst Design - Accelerated Introduction to Verilog-2001 & Best Known Coding Practices* - 1 day
- *Advanced Verilog PLI Courses* - (taught by a Sunburst Design training partner)

**Course Customization?** - Sunburst Design courses can be customized to include ***your*** company's coding guidelines or to modify the course for a different audience. Sections can be added or deleted from a course to meet you company's needs.

## Course Syllabus

### Day One

### Overview of Verilog Resources

### Introduction to Verilog Modeling

- An introduction and overview of major Verilog-2001 modeling basics.

- Modules
- Port and net declarations
- V2K1 ANSI-C style module headers
- Instantiation with positional and named ports
- Procedural blocks: initial & always
- Hierarchy
- Introduction to synthesis design flows
- Power-user guidelines (presented in section 1 - detailed in later sections)

### Verilog HDL Syntax & Semantics

- Detailed instruction of important Verilog-2001 (V2K1) language syntax.

- Good formatting = fewer bugs & better documentation
- Comments
- V2K1 attributes
- Identifier names
- Name scopes
- Language tokens
- Numbers and logic values
- Net & variable types
- Scalars & vectors
- Multi-dimensional arrays
- Port declaration styles
- Parameters
- Verilog's 4+ logic values

**Design Verification & Running Simulations**
- An introduction to writing Verilog testbenches and running Verilog simulations.

- Strength basics
- Simulation times, delays, timescales and time formatting
- Writing Verilog testbenches
- Repeat-loop & forever loop
- Important compiler directives
- Display and formatting commands
- System tasks for simulation control
- Using multiple Verilog source files & Verilog command files
- Running a Verilog simulation*
- Lab: basic testbench development

* Course notes are printed with detailed instructions on how to use the major Verilog simulators (NC Verilog, VCS, ModelSim, Questa - note: per Cadence, Verilog-XL does not and will not support Verilog-2001 enhancements).

**Continuous Assignments & Operators**
- Detailed discussion of continuous assignments with design examples, followed by an overview of Verilog-2001 operators, also with examples.

- Continuous assignments
- Procedural continuous assignments (do not use these!)
- Required net declarations
- ?: conditional operator

**Programming Statements & Timescales**
- Detailed discussion of blocking and nonblocking assignments, followed by an overview of Verilog-2001 programming statements with examples. This section concludes with a discussion of Verilog timescales and their impact on simulation efficiency.

- Verilog operators - arithmetic, bitwise, logical, unary reduction, more
- Sequential & parallel statement groups
- Blocking & nonblocking assignments (introduced)
- Time delays
- Level-sensitive timing controls
- Edge-sensitive timing controls
- Sensitivity lists (V2K1)
- If-else & case statements
- For, while, repeat & forever loops
- Tasks, functions and automatic (V2K1)
- Rise, fall, min, max delays
- `timescale & $timeformat
- Lab: (optional) `timescale & $timeformat capability and efficiency

**Day Two**

**Combinational Logic Modeling**
- Behavioral & synthesizable coding styles for modeling combinational logic. Includes multiple Verilog-2001 enhancements.

- Sensitivity lists
- Continuous assignments
- Procedural combinational blocks
- V2K1 comma-separated and @* sensitivity lists
- Inertial & transport delays
- Correct methods for adding behavioral timing delays

**Sequential Logic Modeling**
- Behavioral & synthesizable coding styles for modeling sequential logic

- Sensitivity lists
- Flip-flops and latches
- Synchronous and asynchronous inputs
- Where to add timing delays
- Lab: model and verify an 8-bit ALU
- Lab: model and verify a pipeline
- Lab: (optional) extra labs

**RAM and ROM Modeling**
- Behavioral coding styles for modeling RAMs and ROMs

- Modeling memories
- Array declarations
- System tasks to load memories
- Preferred coding style for read operations
- Preferred coding style for write operations
- Testing bi-directional ports
- Basic timing constraints
- Lab: model and verify a RAM

**Structural Netlists**
- How to construct a hierarchical Verilog netlist.

- Design hierarchy
- Module instantiation
- Port & net mismatches
- Parameterized models
- Redefining parameters
- V2K1 Multi-dimensional arrays
- Arrays of Instance
- Generate statements
- Lab: model and verify a hierarchical design

**Classroom Details**
Training is generally conducted at your facilities. For maximum effectiveness, we recommend having one workstation or PC for every two students, with licenses for your preferred Verilog simulator (we often can help provide the simulator and temporary training licenses).

## For more information, contact:

Cliff Cummings - cliffc@sunburst-design.com - Sunburst Design, Inc. - 503-641-8446