World Class Verilog & SystemVerilog Training

Sunburst Design - Advanced Verilog-2001 Knowledge & Design 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
40% Lecture, 60% Lab
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 full understanding of the Verilog-2001 language
- understand the necessary constructs for
  - design, synthesis and verification
  - simulation of gate-level netlists
- know how to write and simulate
  - complex hardware models
  - simple synthesizable models
  - self-checking testbenches
- know how to run efficient Verilog simulations
- be immediately productive at modeling and simulating complex Verilog designs

Course Overview

Sunburst Design - Advanced Verilog-2001 Knowledge & Design Practices is a 2-day, fast-paced, intensive course on advanced IEEE 1364-2001 Verilog Hardware Description Language topics and its usage for hardware design and verification. This course is designed to emphasize important RTL modeling and efficient testbench techniques and to allow additional lab time to put these techniques into practice.

This course includes 40+ slides and a copy of Cliff's first award winning paper on nonblocking assignments to offer a comprehensive introduction to blocking vs. nonblocking assignments along with important coding guidelines related to design styles aimed at preventing Verilog simulation race conditions.

Another 40+ slides help to demonstrate multiple efficient Finite State Machine (FSM) coding styles. An EISA bus arbiter lab is designed to reinforce FSM concepts developed in class (no silly traffic-light controllers or soda pop change machines in this course!)

A detailed 350+ page student guide and 49-page Verilog-2001 HDL Quick Reference Guide supplement the lecture and provide excellent resources for after-class reference. Numerous exercises and labs reinforce the principles presented, with about 60% of class time devoted to lab work, including a 5-hour final DSP design lab on the last day of class. The final lab utilizes all of the techniques learned in earlier labs and reinforces how all aspects of Verilog are used in a large design project.

Target Audience

Sunburst Design - Advanced Verilog-2001 Knowledge & Design Practices is intended for new and self-taught Verilog design and verification engineers that require a full knowledge of the capabilities of the Verilog-2001 language. This is both a design and language class. This course also fulfills all of the prerequisites for the Sunburst Design - Advanced Verilog-2001 for Synthesis & Verification course; a course that focuses specifically on Verilog-2001 techniques for design, synthesis and advanced verification.

Prerequisites (mandatory)

This course assumes that students have a practical working knowledge of Verilog HDL or have completed introductory Verilog HDL training. Engineers with VHDL synthesis experience and some Verilog exposure will do well in this class.

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

Blocking vs. Nonblocking Assignments
- Detailed instruction about Verilog blocking and nonblocking assignments. Detailed description of how the Verilog event queue works. Coding guidelines using blocking & nonblocking assignments. After using blocking and nonblocking assignments for two days, now we are ready to give the details for the guidelines already presented.

- Different types of blocking assignments
- Different types of nonblocking assignments
- Explanation of Verilog event scheduling
- Assignment execution order
- Pipeline examples
- Delay line modeling
- Common misconceptions about nonblocking assignments
- Blocking & nonblocking assignment guidelines

State Machine Design
- Detailed description and guidelines for coding Verilog state machines.

- Moore, Mealy, binary & onehot state machines
- State machine coding style guidelines
- Parameter states vs. `define states (do not use `define)
- Two always block state machine coding style
- One always block state machine coding style
- Three always block, registered output, coding style
- Lab: EISA bus arbiter state machine design

File I/O & Testbench Development
- Description of Verilog file I/O commands and usage. Fundamentals of using Verilog file I/O within test environments. Fundamentals of developing self-checking testbenches. Includes a detailed description of stimulus and verification timing.

- Verilog-2001 File I/O commands
- basic testbench structures
- testbench template file
- Self-checking testbench techniques
- When to apply stimulus vectors
- When to verify output vectors
- Common testbench strategies for RTL & gate-level verification
- Lab: Self-checking testbench
- Lab: (optional) Verilog-2001 File I/O experimentation

Verilog Synthesis Design Flows

- This section details how designs are verified, both functionally and for correct timing. Synthesis design flows are also shown for ASICs and FPGAs.

- Functional verification design flows
- Timing verification design flows
- ASIC synthesis design flow
- FPGA synthesis design flow
- Improving synthesis results
- Lab: Introduction to synthesis* *(if synthesis software and licenses are available)

Behavioral Commands & Verilog Strengths

- This section details advanced behavioral commands that are sometimes used and abused in models and testbenches. Correct and incorrect usage examples are included.

- Named blocks
- Disabling blocks & tasks
- Force-release
- Continuous vs. procedural assign
- Assign-deassign
- Named event declaration and usage
- Fork-join
- Verilog strengths

Gate & User Defined Primitives (UDPs)

- For comprehensive coverage of the Verilog language, gate primitives and UDPs are taught in this section. Many vendors use these two primitive types to accurately model their library primitives.

- Gate primitives
- Gate instantiation
- Delays & strengths
- User Defined Primitives (UDPs)
- Combinational UDPs
- Latch UDPs
- Sequential UDPs
- UDP issues
- Lab: (optional) sequential UDP
- Lab: (optional) debugging UDP table entry bugs

Day Two

Specify Blocks

- What all engineers should know to examine and use ASIC, FPGA and other vendor libraries. How path delays and timing checks are built into vendor models.

- Specify block basics
- 1, 2, 3, 6 or 12 timing transitions
- Handling "X" transitions
- Timing check tasks
- Delay calculation and SDF backannotation
- Lab: (optional) simulate with SDF delay backannotation

Switch Primitives

- For comprehensive coverage of the Verilog language, switch and strength reducing primitives are taught in this section. This section also details how switch primitives and strengths can be used to model passive devices, such as resistors and power supplies for board-level simulation.

- Switch primitives
- Verilog strengths
- Modeling passive devices

Verilog Wizardry (Lab Intensive Project)

- This is the final project for the course. Students will use all aspects of the Verilog language in an actual design flow. Students will draw upon what they have learned in the previous three days to model, simulate and verify a complete Digital Signal Processor design.

- Lab: model and verify a synthesizable Digital Signal Processor

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