
**World Class SystemVerilog & UVM Training**

# ***Sunburst Design - Comprehensive SystemVerilog Design & Synthesis***

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.*

**4 Days**
**70% Lecture, 30% Lab**
**Advanced Level**

**Course Selection**
There are two versions of this course. If engineers already know SystemVerilog, they should choose the 2-day Sunburst Design - Expert SystemVerilog Design & Synthesis course.

This course combines the content of the 2-day SystemVerilog Fundamentals training and the 2-day and the 2-day Expert SystemVerilog Design & Synthesis course, and the 1-day CDC & FIFO Design course into a single continuous flow over four days.

**Course Objective**
Simply stated, to give engineers *world class* SystemVerilog language, advanced design & synthesis training using award winning materials developed by renowned Verilog & SystemVerilog Guru, Cliff Cummings

**Upon completion of this course, students will:**

- Write efficient synthesizable SystemVerilog RTL models
  - includes new SystemVerilog data types and capabilities
  - includes new SystemVerilog RTL and abstraction capabilities
  - includes six different FSM coding styles
  - includes multi-clock and FIFO design techniques
- Gain exposure to new SystemVerilog modeling capabilities
  - includes use of dynamic types and arrays for behavioral modeling

**Course Overview**
*Sunburst Design - Comprehensive SystemVerilog Design & Synthesis* is a 4-day fast-paced intensive course that includes all of the content of the 2-day SystemVerilog Fundamentals training, as well as all of the content of the Expert SystemVerilog Design & Synthesis training. This training focuses on proven and new SystemVerilog features for design and synthesis. Efficient and proven coding styles are combined with frequent exercises and insightful labs to demonstrate the capabilities of new SystemVerilog features. You will discover that SystemVerilog capabilities are fully backward compatible with Verilog-2001 designs.

**For more information, contact:**
Cliff Cummings - cliffc@sunburst-design.com - Sunburst Design, Inc. - 801-960-1996

This SystemVerilog training was developed and is frequently updated by the renowned SystemVerilog guru and IEEE SystemVerilog committee member, Cliff Cummings. Cliff has presented at numerous SystemVerilog seminars and training classes world-wide, including the 2003-2004 SystemVerilog NOW! Seminars, the 2010 ModelSim SystemVerilog Assertion Based Verification Seminars, and multiple Verification Academy DAC seminars.

The 1000+ page binder and 140+ page lab guide for this 3-day course covers all of the important SystemVerilog coding styles for RTL & behavioral design. These materials are constantly being updated with the latest clarifications and corrections passed by the IEEE SystemVerilog committee, of which Cliff is an active participant. Numerous proven usage guidelines are taught and explained.

**Target Audience**
*Sunburst Design - Comprehensive SystemVerilog Design & Synthesis* is intended for design engineers who require in-depth knowledge on the IEEE SystemVerilog standard with an emphasis on the new RTL, design & synthesis capabilities.

**Prerequisites (mandatory)**
***This is a very advanced SystemVerilog class that assumes engineers already have a good working knowledge of the Verilog language.***

This course assumes that students have a practical working knowledge of Verilog HDL or have completed Verilog HDL training. Engineers with VHDL synthesis experience and some Verilog exposure will do well in this class. Engineers with no prior HDL training or experience will struggle in this class. Engineers with weak Verilog knowledge or experience should consider adding the 1-day, *Sunburst Design - Accelerated Introduction to Verilog-2001 & Best Known Coding Practices* course to fully prepare for advanced SystemVerilog training.

**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 or SystemVerilog Standards groups or presented at industry recognized conferences. Go to our web site, download the award-winning papers and read about the Sunburst Design - Instructors - they are the best and they have the experience and qualifications to offer best-in-class training.

**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.

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

**For more information, contact:**
Cliff Cummings - cliffc@sunburst-design.com - Sunburst Design, Inc. - 801-960-1996

## Course Syllabus

### Day One

**SystemVerilog Enhancements & Methodology Overview**
- Includes a quick review of SystemVerilog resources available to design & verification engineers.

- Verilog & SystemVerilog Keywords
- SystemVerilog Books & Resources
- SystemVerilog Enhancements Strategy & High-Level Methodology

**Configurations** ***(Optional - included for engineering teams that want to include advanced configuration training)***
- Important Verilog basics that are not generally understood by most Verilog users. Configurations introduced by Verilog-2001. Common useful command lines switches, project directory structures, and commands for conditionally compiled SystemVerilog designs and testbenches.

- Essential Verilog basics
- Working with project directory structures
- Verilog/SystemVerilog command switches
- Conditional compilation for design and verification
- Configuration files

**Data Types & Typedefs** ***(Included in SystemVerilog Fundamentals Training)***
- Includes data types, enumerated types, compilation units, packages, casting and randomization functions.

- Nets & Variables Fundamentals & Guidelines
- Blocking & Nonblocking Assignment Fundamentals & Guidelines

- SystemVerilog data types
- Enhanced literal numbers syntax
- Resolved & Unresolved types
- 4-state & 2-state types
- Typedefs
- Near-Universal types
- SystemVerilog type usage guidelines
- Enumerated types
- Struct data type intro
- Type parameters
- Intro to the SystemVerilog program construct (just 1 slide)
- $unit & $root
- Compilation units & separate compilation
- Packages & :: (package scope operator)

**For more information, contact:**
Cliff Cummings - cliffc@sunburst-design.com - Sunburst Design, Inc. - 801-960-1996

- SystemVerilog package strategies
- Strings (just 1 slide)
- Static & dynamic type-casting
- Random number generation: $random -vs- $urandom -vs- $urandom_range
- Simulation command aliases & switch definitions
- LABS: Multiple SystemVerilog types, typedefs, type-casting and logic labs

**SystemVerilog Operators, Loops, Jumps. New Logic-Specific Processes. Enhanced functions & tasks. New timing controls** ***(Included in SystemVerilog Fundamentals Training)***
- New operator and enhanced loop control have been added for improved design & verification. New always_type blocks show design intent and help ensure construction of proper hardware designs. Enhancements were added to tasks and functions to increase their capabilities. All of these features, and more, are described in this section.

- New SystemVerilog operators
- Enhanced loops & jumping statements
- Logic specific processes (always_type blocks) document designer intent
- always_comb
- always_latch
- always_ff
- Added design checks using always_type blocks
- always @* -vs- always_comb
- void functions
- always_comb & void functions
- Combinational sensitivity
- Design encapsulation through void functions
- SystemVerilog enhancements to tasks & functions
- SystemVerilog priority & unique - modifiers for case- & if-statements
- `timescale directive
- SystemVerilog timeunit & timeprecision

**For more information, contact:**
Cliff Cummings - cliffc@sunburst-design.com - Sunburst Design, Inc. - 801-960-1996

**Latches & Priority Encoders. Unique & Priority - full_case & parallel_case.**
- Introduction to Verilog synthesis design flows. Detailed description of two synthesis problem areas: latches and priority encoders. Detailed description of the synthesis directives "full_case" and "parallel_case", and why they should generally be avoided. This section also details how unique and priority are new SystemVerilog replacements for the dangerous "Evil Twins," full_case parallel_case.

- always_latch
- always blocks & sensitivity lists
- Generating latches
- Generating priority encoders
- Latch & priority encoder guidelines
- full_case parallel_case, "the Evil Twins"
- What is full_case?
- What is parallel_case?
- unique & priority case
- unique & priority if
- unique0 (SystemVerilog-2009 enhancement)
- One example using case modifiers (+ 2 more reference examples)
- LABS: simple SystemVerilog combinational and sequential logic labs

**Implicit .* and .name Port Instantiation** ***(Included in SystemVerilog Fundamentals Training)***
- Implicit port connections can reduce top-level ASIC and FPGA coding efforts by more than 70% and simultaneously enforce greater port type checking.

- Verilog-2001 positional & named ports
- SystemVerilog .* implicit ports
- SystemVerilog .name implicit ports
- Implicit port connection rules & comparisons - includes IEEE 1800 latest updates
- Strong port-type checking
- New debugging techniques - automatic expansion of .* ports
- Block-level testbenches with implicit ports
- Advantages & disadvantages
- LABS: implicit port instantiation labs

**For more information, contact:**
Cliff Cummings - cliffc@sunburst-design.com - Sunburst Design, Inc. - 801-960-1996

**Day Two**

**Nonblocking Assignments, Race Conditions & SystemVerilog Event Scheduling**
- SystemVerilog is fully backward compatible with Verilog-2001 (it is also fully race backward compatible!) This section describes in detail how the new SystemVerilog event scheduling works and how it will reduce race conditions between RTL designs and verification suites.

- Verillog-2001 Event Scheduling
- 8 guidelines for RTL coding & nonblocking assignments
- SystemVerilog enhanced scheduling - includes IEEE 1800 latest updates
- Verilog -vs- SystemVerilog race conditions
- Scheduling of new SystemVerilog commands
- * Blocking & Nonblocking Assignment Details
- * Mixed RTL & Gate simulations

**Structs, Unions, Packed & Unpacked Arrays** ***(Included in SystemVerilog Fundamentals Training)***
- Packed & unpacked arrays, unions and structs allow greater abstraction and more concise coding. The new dynamic array types facilitate behavioral modeling and assist in the development of verification environments.

- Structs & assignment patterns
- Packed & unpacked arrays
- Array indexing
- Structs & packed structs
- Unions & packed unions

**Interfaces** ***(Included in SystemVerilog Fundamentals Training)***
- Interfaces are a powerful new form of abstraction and this section details how they work for design and verification. This section also discusses when and when not to use interfaces.

- Interface usage overview
- Introduction to generic interfaces
- Interfaces -vs- records
- How interfaces work
- 4 requirements for good interface usage
- Interfaces - legal & illegal usage
- Interface constructs
- Interface modports
- LABS: multiple interface and interface-protocol labs

**For more information, contact:**
Cliff Cummings - cliffc@sunburst-design.com - Sunburst Design, Inc. - 801-960-1996

**Combinational Logic I & II**
- RTL coding styles for combinational logic, including problems and inefficiencies that arise from poor coding styles. Includes Verilog-2001 (V2K1) combinational logic enhancements. Numerous combinational labs demonstrate many potential problem areas related to common combinational coding styles. Additional RTL coding styles for combinational logic, including more problems and inefficiencies that occur from poor coding styles. Verilog-2001 (V2K1) enhancements are discussed including reasons to avoid over-usage of generate statements. Example of poor usage includes I/O pad instantiation (use the more concise and better supported Array of Instance). More combinational labs demonstrate many potential problem areas related to common combinational coding styles.

- always_comb
- Continuous assignments
- Always blocks
- V2K1 @* and comma-separated sensitivity lists
- Instantiated library elements
- Instantiated primitives
- Synthesizable and non-synthesizable Verilog constructs
- Bitwise -vs- logical operators
- Tasks & functions
- Tri-state drivers
- Bi-directional busses
- Instantiating ASIC/FPGA library primitives
- Guidelines
- LABS: Combinational logic labs I & II

**SDF Backannotation** ***(Optional - included for engineering teams that want to learn SDF timing-based simulation techniques)***
- This section details SDF (Standard Delay Format) file generation, writing Verilog gate-level netlists, gate-level simulations and gate-level simulations using SDF timing (may be important for simulation of full-board and system designs where Static Timing Analysis is not generally practical)

- SDF file basics
- Gate-level netlists
- SDF back annotation of gate-level designs
- Directory structures and command files for gate-level simulation
- Simulating with SDF files
- Conditional compilation of SDF files
- LAB: Post-synthesis gate-level netlist generation and simulation with SDF timing

**For more information, contact:**
Cliff Cummings - cliffc@sunburst-design.com - Sunburst Design, Inc. - 801-960-1996

**Day Three**

**Combinational Labs Review**
- An in-depth review of all the coding styles used in the combinational labs and the synthesized results. Conclusions are drawn about which coding styles infer the most efficient logic implementations.

- Review of Day-2 labs

**Sequential Logic**
- This section covers coding styles for sequential logic. Inferring efficient designs using adders and other large resources is also detailed. Also discusses and includes advantages and disadvantages of instantiation.

- always_ff
- Edge-sensitive sensitivity list
- Basic asynchronous & synchronous resets
- Additional flip-flop coding styles
- Simulation/synthesis differences
- Simulation efficiency
- Register banks
- Memories
- Instantiating Blocks
- Resource sharing
- LABS: Sequential logic labs

**Synchronous & Asynchronous Reset Design**
- Detailed material for selection and usage of synchronous and asynchronous reset design taken from actual design experiences.

- Synchronous vs. asynchronous resets
- Reset removal metastability
- Asynchronous reset synchronizer circuitry
- Reset distribution trees and techniques
- Multiple clock domains reset synchronization

**For more information, contact:**
Cliff Cummings - cliffc@sunburst-design.com - Sunburst Design, Inc. - 801-960-1996

**IEEE Verilog 2001 Enhancements**
- A concise summary of the important IEEE Verilog-2001 enhancements for both design and verification.

- The V2K1 top-5 enhancement requests
- V2K1 Multi-dimensional arrays
- Array of Instances (AOI) & V2K1 generate statements
- V2K1 reentrant tasks and functions
- V2K1 configurations
- New V2K1 port and parameter styles
- V2K1 sensitivity lists
- V2K1 RTL enhancements
- Guidelines
- Labs: Combinational labs II

**SVA - SystemVerilog Assertions** ***(Included in SystemVerilog Fundamentals Training)***
- This section details how the SystemVerilog Assertion (SVA) syntax works and how assertions can be used for design and verification. Special macro-techniques are shown to reduce assertion coding effort by up to 80%.

- What is an assertion? / Who should add assertions?
- Assertion benefits - bug detection efficiency
- SystemVerilog assertion types
- SystemVerilog immediate assertions
- SystemVerilog concurrent assertions
- Assert & cover properties & labels
- Properties and assert property
- Overlapping & non-overlapping implications
- Edge testing functions
- Sequences
- Vacuous success
- Property styles
- Reduced assertion coding effort using macros
- Macros with default arguments (SystemVerilog-2009 update)
- Assertion coding style efficiency benchmarks
- SystemVerilog assertion system functions
- Sampled value functions
- Assertion severity tasks
- Assertion and coverage example of an FSM design
- Binding SVA to an existing model
- Bind command details and guidelines
- LABS: SystemVerilog Assertions with synchronous FIFO design

**For more information, contact:**
Cliff Cummings - cliffc@sunburst-design.com - Sunburst Design, Inc. - 801-960-1996

**Day Four**

**SystemVerilog FSM Design Techniques** ***(Mostly included in SystemVerilog Fundamentals Training)***

- Six different FSM coding styles, enhanced with new SystemVerilog constructs, are detailed and compared for coding and synthesis efficiency. Multiple FSM designs are benchmarked for coding style efficiency.

- FSM coding goals
- Moore & Mealy
- Binary & Onehot
- ASIC -vs- FPGA FSM design
- Review proven FSM coding styles
- One always block - avoid this
- Two always blocks - recommended
- Three always blocks - recommended
- Onehot case(1'b1) - recommended
- Onehot parameters - avoid this
- Output encoded - recommended
- Coding & synthesis efficiency
- Verilog-2001 FSM enhancements
- SystemVerilog FSM enhancements
- Advanced enumerated types
- LABS: SystemVerilog FSM design labs

**Multi-clock Clock Domain Crossing (CDC) & FIFO Design Techniques using SystemVerilog**

- Very advanced design techniques from Cliff's award-winning presentations on the efficient implementation of multi-clock CDC & FIFO designs. These materials are not specific to SystemVerilog but solutions are shown using SystemVerilog syntax (advanced techniques that all design engineers should know - the stuff you did not learn in college).

- (1) Metastability & synchronizers - synchronizing 1-bit signals
- (2) Passing multiple control signals - synchronizing multi-bit signals or busses
    a. Consolidation
    b. Controlled synchronization - multicycle path formulations (MCP)
    c. FIFO synchronizer
    d. Gray codes & Gray code counters
- (3) Design partitioning - design & synthesis techniques
    a. Naming conventions
    b. Synthesis scripting & timing analysis issues
- (4) Simulation issues
    a. X-propagation issues
    b. Synopsys command for SDF files
    c. Multi-SDF files
    d. ASIC/FPGA vendor cells and models

**For more information, contact:**
Cliff Cummings - cliffc@sunburst-design.com - Sunburst Design, Inc. - 801-960-1996

e. Simulation model to expose synchronization problems

- (5) Multi-clock FIFO design - large section on design and FIFO issues
- LAB: MCP controlled synchronization lab
- LAB: 2-Register synchronization lab
- LAB: 2-clock FIFO lab

**For more information, contact:**
Cliff Cummings - cliffc@sunburst-design.com - Sunburst Design, Inc. - 801-960-1996