

## Course Description

One of the problems facing system designers today is that standard HDL languages such as VHDL and Verilog, while very adequate for RTL descriptions, do not have a high-enough level of abstraction to describe models efficiently. They also lack the advanced verification structures that are needed with today's more complex hardware.

SystemVerilog is built on top of Verilog 2001. It improves the productivity, readability, and reusability of Verilog-based code. The language enhancements in SystemVerilog provide more concise hardware descriptions, while still providing an easy route with existing tools into current hardware implementation flows. The enhancements take on the form of programming structures that allow more robust modeling of hardware constructs, constrained randomization of data values for the creation of test vectors, and assertion-based testing. Here, the rules are established for the values of signals (related to each other) and these rules (assertions) are tested as the simulation evolves.

**Level** – Intermediate

**Course Duration** – 2 days

**Price** – \$1600 or 16 Training Credits

**Course Part Number** – HDT-SYSVER-200-ILT

**Who Should Attend?** – HDL Designers and Verification Engineers who want to keep current with the latest trends in ASIC/FPGA Verification Methodology, specifically for creating more elaborate tests.

**Prerequisites**

- Basic HDL (VHDL or Verilog) knowledge

**Software Tools**

- ModelSim

**Hardware**

- None

## Course Outline

### Day 1

- Advanced Data Types (2-value vs. 4-value logic, Packed & Unpacked Arrays, Structures & Unions, Enumerated Data Types, Queues)
- SystemVerilog (Definitions, Inheritance)
- Associative Arrays (Inserting vs. Extracting values)
- **Lab 1** – Comparison of HDL and SystemVerilog
- **Lab 2** – SystemVerilog Classes

### Day 2

- Random Data Generation (Generating Random Numbers, Random Constraints)
- Assertions (Immediate vs. Concurrent)
- Basic Interfaces
- Synthesis of SystemVerilog Structures
- **Lab 3** – Constrained Randomization
- **Lab 4** – Assertions

### Lab Descriptions

- **Lab 1** – Comparison of HDL and SystemVerilog - A comparison is made between a conventional HDL and SystemVerilog. This lab will illustrate the design structure of using SystemVerilog in a conventional HDL design flow.
- **Lab 2** – SystemVerilog Classes - Students will use SystemVerilog to create a complex model of a data structure using the various data structures available. A testbench will be written in order to compare the model and the Synthesizable HDL implementation of the structure.
- **Lab 3** – Constrained Randomization - Creating randomly constrained data inputs to a structure. This will demonstrate how large scale testing can be applied in order to verify operation, including finding corner cases.
- **Lab 4** – Assertions - Illustrates the use of assertions in order to verify the design's coding. The use of assertions verifies that the RTL code complies with the actual specification of the design.

## Register Today

Hardent offers courses to help designers produce fast, predictable and efficient designs. For a detailed list, visit [www.hardent.com/training](http://www.hardent.com/training) or contact Hardent's Training Coordinator for additional information, to register for a class or to schedule a private course.

**Contact:** Chris Green  
**Email:** [training@hardent.com](mailto:training@hardent.com)  
**Telephone:** 514-999-3880