LOGTEL
It's more than knowledge

Authorized Training Provider
XILINX®

Verification with SystemVerilog

Course Description

This comprehensive course is a thorough introduction to SystemVerilog constructs for verification. This class addresses writing testbenches to verify your design under test (DUT) utilizing the new constructs available in SystemVerilog. Object-oriented modeling, new data types, re-usable tasks and functions, randomization, code coverage, assertions, and the Direct Programming Interface (DPI) are all covered. The information gained can be applied to any digital design verification approach. This course combines insightful lectures with practical lab exercises to reinforce key concepts.
In this two-day course, you will gain valuable hands-on experience. Incoming students with a Verilog background will finish this course empowered with the ability to more efficiently verify designs.

Level: FPGA 1

Training Duration: 2 days

Who Should Attend?

Hardware and verification engineers

Prerequisites:

- Verilog design experience or completion of the *Designing with Verilog* course

Recommended

- Designing with SystemVerilog course

Software Tools:

- Questa Sim Prime Simulator 10.4a
- Vivado® Design or System Edition 2015.3

Hardware:

- Architecture: N/A*
- Demo board: None*

Skills Gained: After completing this training, you will be able to:

- Describe the advantages and enhancements to SystemVerilog to support verification
- Define the new data types available in SystemVerilog
- Analyze and use the improvements to tasks and functions
- Discuss and use the various new verification building blocks available in SystemVerilog
- Describe object-oriented programming and create a class-based verification environment
- Explain the various methods for creating random data
- Create and utilize random data for generating stimulus to a DUT
- Identify how SystemVerilog enhances functional coverage for simulation verification
- Utilize assertions to quickly identify correct behavior in simulation
- Identify how the direct programming interface can be used with C/C++ in a verification environment

Course Outline

1. Introduction to SystemVerilog for Verification
2. Data Types
3. Tasks and Functions
4. Lab 1: Implementing Tasks and Functions
5. SystemVerilog Verification Building Blocks
6. Lab 2: Connecting the Testbench to the DUT
7. Object-Oriented Modeling
8. Lab 3: Object-Oriented Modeling
9. Randomization
10. Lab 4: Randomization
11. Coverage
12. Lab 5: Coverage
13. Assertions
14. Lab 6: Assertions
15. Direct Programming Interface
16. Demo: Direct Programming Interface

Lab Description

Lab 1: Implementing Tasks and Functions – Use a task and function to provide input data for a DUT and perform simulation.
Lab 2: Connecting the Testbench to the DUT – Utilize new SystemVerilog verification building blocks to connect the input data to the DUT.
Lab 3: Object-Oriented Modeling – Use object-oriented programming concepts to create a class for enhancing the verification of the DUT.
Lab 4: Randomization – Create random data as input into the DUT to fully validate the design.
Lab 5: Coverage – Create and use a coverage group to validate the code coverage for the DUT. Make adjustments and again validate the coverage.
Lab 6: Assertions – Create an assertion to validate all possible conditions are verified for the DUT.