Verification with SystemVerilog

FPGA 1

LANG-SVVER (v1.0)

Course Specification

Course Description

This course provides an introduction to SystemVerilog constructs for verification.

This emphasis is on:

- Writing testbenches to verify a design under test (DUT) utilizing the constructs available in SystemVerilog
- Reviewing object-oriented modeling, data types, reusable tasks and functions, randomization, code coverage, assertions, and the Direct Programming Interface (DPI)

What's New for 2021.1

- All labs have been updated to the latest software versions

Level – FPGA 1

Course Details

- 2 days
- 10 lectures
- 5 labs

Price –

Course Part Number – LANG-SVVER

Who Should Attend? – Hardware and verification engineers

Prerequisites

- Experienced Verilog user or completion of the *Designing with Verilog* course

Recommended

- *Designing with SystemVerilog* course

Software Tools

- Vivado® Design Suite 2021.1

Hardware

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

* This course does not focus on any particular architecture. Check with your local Authorized Training Provider for specifics or other customizations.

After completing this comprehensive training, you will have the necessary skills 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

Day 1

- Introduction to SystemVerilog for Verification
- Data Types
- Tasks and Functions
- **Lab 1:** Implementing Tasks and Functions
- SystemVerilog Verification Building Blocks
- **Lab 2:** Connecting the Testbench to the DUT
- Object-Oriented Modeling
- **Lab 3:** Object-Oriented Modeling

Day 2

- Randomization
- **Lab 4:** Randomization
- Coverage
- **Lab 5:** Coverage
- Assertions
- **Lab 6:** Assertions
- Direct Programming Interface
- **Demo:** Direct Programming Interface
- Inter Process Communication

Lab Descriptions

- **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.

Register Today

Visit the Xilinx Customer Training Center to view schedules and register online.