XILINX

Advanced VHDL

FPGA 4

LANG-ADVVHDL (v1.0)

Course Specification

Course Description

Increase VHDL proficiency by learning advanced techniques for writing more robust and reusable code.

The focus is on:

- Writing efficient and reusable RTL, testbenches, and packages
- Creating self-testing testbenches
- Creating realistic models
- Using the text I/O capabilities of the VHDL language
- Storing simulation data dynamically
- Creating parameterized code for design reuse

This comprehensive course is targeted toward designers who already have some experience with VHDL.

What's New for 2021.1

- All labs have been updated to the latest software versions

Level – FPGA 4

Course Details

- 2 days
- 9 lectures
- 6 labs

Price –

Course Part Number – LANG-ADVVHDL

Who Should Attend? – VHDL users with intermediate knowledge of VHDL

Prerequisites

- *Designing with VHDL* course or equivalent knowledge of modeling, simulation, and RTL coding
- At least six months of coding experience beyond an introductory 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:

- Write efficient and reusable RTL, testbenches, and packages
- Create self-testing testbenches
- Create realistic models
- Use the text I/O capabilities of the VHDL language
- Store simulation data dynamically
- Create parameterized code for design reuse

Course Outline

Day 1

- VHDL Overview
- Simulation Concepts
- Advanced Data Types
- Subprograms and Design Attributes
- **Lab 1:** Flexible Functions
- Access Type Techniques and Blocks
- **Lab 2:** Linked Lists with Access Types
- Utilizing File IO
- **Lab 3:** TextIO Techniques

Day 2

- Advanced Techniques in VHDL
- **Lab 4:** Creating Real-World Simulations
- Supporting Multiple Platforms
- **Lab 5:** Supporting Multiple Platforms
- Non-Integer Numbers
- **Lab 6:** Implementing Fixed and Floating Point Numbers
- Appendix: Guarded Signals

Lab Descriptions

- **Lab 1:** Flexible Functions – Construct and use predefined attributes to build functions and procedures that automatically adjust to the size of the passed arguments as well as creating a reusable module with unconstrained ports.
- **Lab 2:** Linked Lists with Access Types – Create linked lists to capture arbitrarily large data sets. Also included in this lab is a reusable helper package for managing singly linked lists.
- **Lab 3:** TextIO Techniques – Load memory for synthesis via a text file using the TextIO extensions for std_logic and std_logic_vector as provided by the std_logic_TextIO package.
- **Lab 4:** Creating Real-World Simulations – Create spread-spectrum clocks with jitter and other real-world factors. Model board and behavioral component delay.
- **Lab 5:** Supporting Multiple Platforms – Effectively use configuration statements, conditional generates, and scripts to build variations on VHDL themes.
- **Lab 6:** Implementing Fixed and Floating Point Numbers – Construct a simple fixed point math example and compare to the IEEE_PROPOSED fixed and floating point models.

Register Today

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