

Course Description

Increase your VHDL proficiency by learning advanced techniques that will help you write more robust and reusable code. This comprehensive course is targeted toward designers who already have some experience with VHDL.

The course highlights modeling, testbenches, RTL/synthesizable design, and techniques aimed at creating parameterizable and reusable designs. The majority of class time is spent in challenging hands-on labs as compared to lecture modules.

Level – FPGA 4

Course Duration – 2 days

Price – \$1400 or 14 Xilinx Training Credits

Course Part Number – LANG-ADV-VHDL-ILT

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 or System Edition 2015.1

Hardware

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

* This course does not focus on any particular architecture. Check with North Pole Engineering, Inc., 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

- Review of Current Knowledge
- 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

- Cool Stuff with 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
- Course Summary

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

NPE, Inc. delivers public and private courses in locations throughout the central US region; including Iowa, Illinois, Kansas, Minnesota, Missouri, North Dakota, South Dakota and Wisconsin.

Visit www.npe-inc.com/training, for full course schedule and training information.



You must have your tuition payment information available when you enroll. We accept credit cards (Visa, MasterCard, or American Express) as well as purchase orders and Xilinx training credits.