Designing with VHDL

FPGA 1

LANG11000-ILT (v1.0)

Course Specification

Course Description

This comprehensive course is a thorough introduction to the VHDL language. The emphasis is on writing solid synthesizable code and enough simulation code to write a viable testbench. Structural, Register Transfer Level (RTL), and behavioral coding styles are covered. This class addresses targeting Xilinx devices specifically and FPGA devices in general. The information gained can be applied to any digital design by using a top-down synthesis design approach. This course combines insightful lectures with practical lab exercises to reinforce key concepts. You will also learn best coding practices that will increase your overall VHDL proficiency and prepare you for the *Advanced VHDL* course.

In this three-day course, you will gain valuable hands-on experience. Incoming students with little or no VHDL knowledge will finish this course empowered with the ability to write efficient hardware designs and perform high-level HDL simulations.

Level – FPGA 1

Course Duration – 3 days

Price – $1800 or 18 Xilinx Training Credits

Course Part Number – LANG11000-ILT

Who Should Attend? – Engineers who want to use VHDL effectively for modeling, design, and synthesis of digital designs

Prerequisites

- Basic digital design knowledge

Software Tools

- Vivado™ System Edition 2012.2

Hardware

- Architecture: N/A*
- Demo board: Kintex™-7 FPGA KC705 board*

* This course does not focus on any particular architecture. Check with your local Authorized Training Provider for the specifics of the in-class lab board or other customizations.

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

- Implement the VHDL portion of coding for synthesis
- Identify the differences between behavioral and structural coding styles
- Distinguish coding for synthesis versus coding for simulation
- Use scalar and composite data types to represent information
- Use concurrent and sequential control structure to regulate information flow
- Implement common VHDL constructs (Finite State Machines [FSMs], RAM/ROM data structures)
- Simulate a basic VHDL design
- Write a VHDL testbench and identify simulation-only constructs
- Identify and implement coding best practices
- Optimize VHDL code to target specific silicon resources within the Xilinx FPGA
- Create and manage designs within the ISE software environment

Course Outline

Day 1

- The "Shape" of VHDL
- **Lab 1:** Using the Tools
- Documentation in VHDL
- Data Types
- Concurrent Operations
- **Lab 2:** Using Concurrent Statements
- Processes and Variables
- **Lab 3:** Designing a Simple Process

Day 2

- Introduction to Testbenches
- ISim Simulation Tool Basics
- **Lab 4:** Simulating a Simple Design
- Creating Memory
- **Lab 5:** Building a Dual-Port Memory
- Finite State Machines
- **Lab 6:** Building a Moore Finite State Machine
- Targeting Xilinx FPGAs
- **Lab 7:** Xilinx Tool Flow

Day 3

- Loops and Conditional Elaboration
- **Lab 8:** Using Loops
- Attributes
- Functions and Procedures
- Packages and Libraries
- **Lab 9:** Building Your Own Package
- Interacting with the Simulation
- Writing a Good Testbench
- **Lab 10:** Building a Meaningful Testbench

Lab Descriptions

The labs for this course provide a practical foundation for creating synthesizable RTL code. All aspects of the design flow are covered in the labs. You will write, synthesize, simulate, and implement all the labs. The focus of the labs is to write code that will optimally infer reliable and high-performance circuits.

Register Today

To register for this course or to see a list of currently scheduled classes, please visit our secure **Online Store**.

To request a public or private class, inquire about course offerings, or any other specific Xilinx training needs, please contact Faster Technology through one of the following:

Web: **Request a Class**
Email: **registrar@fastertechnology.com**
Phone: 281-391-5482

Visit **www.FasterTechnology.com/training-courses** to see our full line of Xilinx education courses in the areas of FPGA Design, Embedded Systems Development, Connectivity, DSP Design, Languages, and CPLD Design.