XILINX
ALL PROGRAMMABLE™

VHDL and Advanced VHDL

FPGA 1

LANG11020-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 – $2700 or 27 Xilinx Training Credits

Course Part Number – LANG11020-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® Design or System Edition 2014.1

Hardware

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

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

Course Outline

You will be provided printed materials and labs for the full 5 day course, but the instructor will present only 3 days' worth of materials and labs from the following, and other custom labs/demos.

Day 1

- The "Shape" of VHDL
- Demo: Multiplexer
- **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

Day 4

- Review of Current Knowledge
- Simulation Concepts
- Advanced Data Types
- Subprograms and Design Attributes
- **Lab 11:** Flexible Functions
- Access Type Techniques and Blocks
- **Lab 12:** Linked Lists with Access Types
- Utilizing File IO
- **Lab 13:** TextIO Techniques

Day 5

- Cool Stuff with VHDL
- **Lab 14:** Creating Real-World Simulations
- Supporting Multiple Platforms
- **Lab 15:** Supporting Multiple Platforms
- Non-Integer Numbers
- **Lab 16:** Implementing Fixed and Floating Point Numbers
- Course Summary

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.

Advanced Lab Descriptions

- **Lab 11:** 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 12:** 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 13:** 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 14:** Creating Real-World Simulations – Create spread-spectrum clocks with jitter and other real-world factors. Model board and behavioral component delay.
- **Lab 15:** Supporting Multiple Platforms – Effectively use configuration statements, conditional generates, and scripts to build variations on VHDL themes.
- **Lab 16:** 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.

NORTH POLE npe ENGINEERING

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.