Designing with VHDL

FPGA 1

LANG-VHDL-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 – $2,400 or 24 Xilinx Training Credits

Course Part Number – LANG-VHDL-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 2018.1

Hardware

- Architecture: N/A*
- Demo board: Kintex® UltraScale™ FPGA KCU105 or Kintex-7 FPGA KC705 board*

* This course does not focus on any particular architecture.

** Check with Morgan Advanced Programmable Systems, 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

Day 1

- Introduction to VHDL {Lecture}
- VHDL Design Units {Lecture, Lab}
- VHDL Objects, Keywords, Identifiers {Lecture}
- Scalar Data Types {Lecture}
- Composite Data Types {Lecture}
- VHDL Operators {Lecture}
- Concurrency in VHDL {Lecture}
- Concurrent Assignments {Lecture, Lab}
- Processes and Variables {Lecture, Demo, Lab}

Day 2

- Conditional Statements in VHDL: if/else, case {Lecture, Lab}
- Sequential Looping Statements {Lecture, Lab}
- Delays in VHDL: Wait Statements {Lecture}
- Introduction to the VHDL Testbench {Lecture, Lab}
- VHDL Assert Statements {Lecture}
- VHDL Attributes {Lecture}
- VHDL Subprograms {Lecture}
- VHDL Functions {Lecture, Lab}
- VHDL Procedures {Lecture}

Day 3

- VHDL Libraries and Packages {Lecture, Lab}
- Interacting with the Simulation {Lecture}
- Finite State Machine Overview {Lecture}
- Mealy Finite State Machine {Lecture}
- Moore Finite State Machine {Lecture, Lab}
- FSM Coding Guidelines {Lecture}
- Vivado Simulator and Race Conditions in VHDL {Lecture}
- Writing a Good Testbench {Lecture, Lab}
- Targeting Xilinx FPGAs {Lecture, Lab}

Topic Descriptions

Day 1

- Introduction to VHDL – Discusses the history of the VHDL language and provides an overview of the different features of VHDL.
- VHDL Design Units – Provides an overview of typical VHDL code.
- VHDL Objects, Keywords, Identifiers – Discusses the data objects that are available in the VHDL language as well as keywords and identifiers.
- Scalar Data Types – Covers both intrinsic and commonly used data types.
- Composite Data Types – Covers composite data types (arrays and records).
- VHDL Operators – Reviews all VHDL operator types.
- Concurrency in VHDL – Describes concurrent statements and how signals help in achieving concurrency.
- Concurrent Assignments – Covers both conditional and unconditional assignments.
- Processes and Variables – Introduces sequential programming techniques for a concurrent language. Variables are also discussed.

Day 2

- Conditional Statements in VHDL: if/else, case – Describes conditional statements such as if/else and case statements.
- Sequential Looping Statements – Introduces the concept of looping in both the simulation and synthesis environments.
- Delays in VHDL: Wait Statements – Covers the wait statement and how it controls the execution of the process statement.
- Introduction to the VHDL Testbench – Introduces the concept of the VHDL testbench.

- VHDL Assert Statements – Describes the concept of VHDL assertions.
- VHDL Attributes – Describes attributes, both predefined and user defined.
- VHDL Subprograms – Covers the use of subprograms in verification and RTL code to model functional blocks.
- VHDL Functions – Describes functions, which are integral to reusable and maintainable code.
- VHDL Procedures – Describes procedures, common constructs that are also important for reusing and maintaining code.

Day 3

- VHDL Libraries and Packages – Demonstrates how libraries and packages are declared and used.
- Interacting with the Simulation – Describes how to interact with a simulation via text I/O.
- Finite State Machine Overview – Provides an overview of finite state machines, one of the more commonly used circuits.
- Mealy Finite State Machine – Describes the Mealy FSM and how to code for it.
- Moore Finite State Machine – Describes the Moore FSM and how to code for it.
- FSM Coding Guidelines – Discusses FSM implementation in an FPGA using VHDL.
- Vivado Simulator and Race Conditions in VHDL – Introduces the Vivado simulator simulation environment. Race conditions are also discussed.
- Writing a Good Testbench – Explores how time-agnostic, self-checking testbenches can be written and applied.
- Targeting Xilinx FPGAs – Focuses on Xilinx-specific implementation and chip-level optimization.

Register Today

Morgan Advanced Programmable Systems, Inc. (MAPS, Inc.) delivers public and private courses in locations throughout the central US region; including Iowa, Illinois, Kansas, Minnesota, Missouri, Nebraska, North Dakota, South Dakota and Wisconsin.

Visit **morgan-aps.com/training**, for full course schedule and training information.

Morgan
Advanced Programmable Systems

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.

Student Cancellation Policy

- Students cancellations received more than 7 days before the first day of class are entitled to a 100% refund. Refunds will be processed within 14 days.
- Student cancellations received less than 7 days before the first day of class are entitled to a 100% credit toward a future class.
- Student cancellations must be sent here.

MAPS Inc. Course Cancellation Policy

- We regret from time to time classes will need to be rescheduled or cancelled.
- In the event of cancellation, live on-line training may be offered as a substitute.
- MAPS may cancel a class up to 7 days before the scheduled start date of the class; all students will be entitled to a 100% refund.
- Under no circumstances is MAPS responsible or liable for travel, lodging or other incidental costs. Please be aware of this cancellation policy when making your arrangements.
- For additional information or to schedule a private class contact us here.