Designing with VHDL

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.

Level: FPGA 1

Training Duration: 4 days

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 Suite 2022.1

Hardware:

- Architecture: N/A*
- Demo board: Zynq® UltraScale+™ MPSoC ZCU104 board

Skills Gained: **After completing this 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:

1. The “Shape” of VHDL
2. Demo: Multiplexer
3. **Lab 1:** Using the Tools
4. Documentation in VHDL
5. Data Types
6. Concurrent Operations
7. **Lab 2:** Using Concurrent Statements
8. Processes and Variables
9. **Lab 3:** Designing a Simple Process
10. Introduction to Testbenches
11. ISim Simulation Tool Basics
12. **Lab 4:** Simulating a Simple Design
13. Creating Memory
14. **Lab 5:** Building a Dual-Port Memory
15. Finite State Machines
16. **Lab 6:** Building a Moore Finite State Machine
17. Targeting Xilinx FPGAs
18. **Lab 7:** Xilinx Tool Flow
19. Loops and Conditional Elaboration.
20. **Lab 8:** Using Loops
21. Attributes
22. Functions and Procedures
23. Packages and Libraries
24. **Lab 9:** Building Your Own Package
25. Interacting with the Simulation
26. Writing a Good Testbench
27. **Lab 10:** Building a Meaningful Testbench

Lab Description

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.