VHDL Coding for Synthesis

4 Days: 50% Lecture, 50% Lab **Advanced Level**

Overview

An in-depth study of VHDL RTL (FPGA and ASIC) coding styles, methodologies, design techniques, problem solving techniques, and advanced language constructs to produce better, faster, and smaller logic.

Class topics focus on mapping digital hardware structures to vendor independent VHDL code. Detailed do's and don'ts of synthesis coding styles are discussed. Lecture and laboratory materials illustrate the optimization differences achieved by different VHDL coding styles. Students will learn proven coding practices that result in smaller and faster designs. The numerous examples in this course make it suitable for a student with limited VHDL. The application focus of this course results in the student being ready for VHDL based ASIC or FPGA design.

Intended Audience

The numerous examples in this course make it suitable for a student with limited VHDL. The application focus of this course results in the student being ready for VHDL based ASIC or FPGA design.

Course Objective

Upon completion of this course, students will be able to:

- Write efficient, vendor independent VHDL code
- Use best coding styles and practices to create ASIC and FPGA logic
- Understand and avoid problematic hardware coding styles
- Use code templates to create a design from the block diagram
- Read VHDL code and draw the corresponding hardware (reverse engineer)
- Use types, overloading, and conversion functions from standard VHDL packages (std_logic_1164, numeric_std, and std_logic_arith)
- Use advanced VHDL constructs to simplify the coding process
- Identify and solve synthesis issues using VHDL coding techniques
- Force a synthesis tool to create the desired logic structure

VHDL Coding for Synthesis

Course Outline

Day 1, Module Syn1
- Synthesis Overview
- Combinational Logic
- Registers and Latches
- UART Transmitter:
  - RTL Code + Statemachine

Day 2, Module Syn2
- Numeric Types and Packages
- Arithmetic Logic
- Comparison and Multiplication
- Partitioning
- Synthesis Process

Day 3, Module AdvSyn1
- Subprograms for Synthesis
- Advanced Combinational Logic
- Advanced Sequential Logic
- Parameterizing Designs

Day 4, Module AdvSyn2
- Advanced Arithmetic
- Architecting Hardware
- TxPort Statemachine
- Fixed and Floating Point Types

Prerequisites

Students taking this course should have working knowledge of digital circuits and prior exposure to VHDL through experience or the course:

Comprehensive VHDL Introduction - 4 days

Other Recommended Courses

Students may also be interested in the following companion course:

VHDL Testbenches and Verification – 4 days

Customization

All of our courses can be customized to meet your specific needs. For some ideas, see the following class descriptions or contact us.

Intermediate VHDL Coding for Synthesis - 2 days = Syn1 and Syn2

Advanced VHDL Coding for Synthesis - 2 days = AdvSyn1 and AdvSyn2

Training Approach

This hands-on, how-to course is taught by experienced hardware designers using a computer driven projector. We prefer and encourage student and instructor interaction. Questions are welcome. Bring problematic code.

Contact

To schedule a class or for more information, contact:

Jim Lewis
Director of Training
(800) 505-8435 / (800) 505-VHDL
jim@SynthWorks.com
http://www.SynthWorks.com

Learn VHDL from a designer's perspective with SynthWorks.