

Advanced VHDL Design Techniques



Course Description

In this course, you will learn & practice efficient coding techniques for VHDL synthesis. You will gain experience writing behavioral & structural code & learn to effectively code common logic functions including registers, memory & arithmetic functions. You will use VHDL constructs to parameterize your designs to increase their flexibility and reusability. While the concepts presented will mainly be targeting Altera® devices using the Quartus® II software environment, many can be applied to synthesizing hardware using other synthesis tools as well. You will also be introduced to testbenches, VHDL constructs used to build them & common ways to write them. The hands-on exercises will use Quartus II software to process VHDL code and ModelSim®-Altera software for simulation.

Skills Developed

- Develop coding styles for efficient synthesis when:
 - Targeting device features
 - Inferring logic functions
 - Using arithmetic operators
 - Writing state machines
- Use Quartus II software RTL Viewer to verify correct synthesis results
- Incorporate Altera structural blocks in VHDL designs
- Write simple testbenches for verification
- Create parameterized designs

Prerequisites

We recommend completing the following courses:

- Introduction to VHDL
- VHDL Basics

Skills Required

- Completion of the "Introduction to VHDL" course or some prior knowledge and use of VHDL
- Background in digital logic design
- Understanding of synthesis and simulation processes

| | |
|---------------|--|
| Course Length | 1 day |
| Language | Presentation in German or English Slides and documentation in English |
| Platform | PC Windows XP / Windows 7 |
| Pricing | On request |
| Dates | On request |

Exercises

- How to use IF-THEN efficiently
- Create a 16-bit up/down counter with a modulus using variables
- Learn how to use Altera's MegaWizard to generate VHDL components and how to connect them to your own logic
- Instantiating components (structural modeling) in a VHDL design file.
- Understand the idea of "black-boxing" lower level VHDL components
- State machine encoding
- Operator balancing
- Create a VHDL Testbench
- Understand how generics are used to make code parameterized

El Camino GmbH
Landshuter Str. 1
84048 Mainburg
Germany

phone: +49-8751-8787-0
fax: +49-8751-842876
e-mail: info@elca.de
www.elcamino.de

© 2010 El Camino GmbH

Altera, Stratix, Arria, Cyclone, MAX, HardCopy, Nios, Quartus, and MegaCore are either registered trademarks or trademarks of Altera Corporation in the United States and/or other jurisdictions. All other trademarks are the property of their respective holders.

