



## Course Description

This three-day class is a general introduction to the Verilog language and its use in programmable logic design. The emphasis is on the synthesis constructs of Verilog. You will gain a basic understanding of Verilog, enabling you to create your own designs. In the hands-on laboratory sessions, you will put this knowledge to the test by writing simple but practical code. You will also learn the basic instructions needed for operating the synthesis and simulation tools Quartus II. Furthermore the implementation of test benches and the use of ModelSim for Verilog simulation will be covered.

## Skills Developed

- Overview of Verilog HDL
- Basic Structure of a Verilog HDL Model
- Components of a Verilog HDL Module
  - Ports
  - Data Types
  - Assigning Values and Numbers
  - Operators
  - Behavioral Modeling
  - Structural Modeling
- Verilog HDL Environment
- Simulation
  - Models
  - Test Benches
- Synthesis
  - Subset of the Language
- Working with ModelSim to perform both functional and gate level simulations
- Working with Quartus II to perform static timing analysis and specifying timing constraints

## Skills Required

- Background in digital logic design
- Prior, basic knowledge of a programming language or hardware description language is a plus
- No prior knowledge of Verilog, Quartus II or ModelSim software is needed.

Course Length	3 days	
Language	Presentation in German or English Slides and documentation in English	
Platform	PC Windows XP / Windows 7	
Pricing	Public In-House	1500,- EUR/attendee On Request
Dates	See schedule at <a href="http://www.elcamino.de">http://www.elcamino.de</a>	

## Exercises

- Build a 16-bit adder
- Build a 4x4 multiplier
- Build a 4-bit 2:1 multiplexer
- Build an 8-bit to 16-bit shifter
- Build a 7-segment display using CASE statement
- Build a 16-bit register with synchronous operation
- Build a 2 bit counter with asynchronous operation
- Examine the controlling state machine
- Putting it all together by declaring and instantiating the lower-level components
- How to use IF-ELSE efficiently
- Create a 16-bit up/down counter with a modulus
- State Machine Encoding
- Operator Balancing, Resource Sharing & Pipelining
- Create a testbench to simulate a multiplier design
- Create a self checking testbench to simulate a multiplier design
- Writing Parameterized Code

El Camino GmbH  
Landshuter Str. 1  
84048 Mainburg  
Germany

phone: +49-8751-8787-0  
fax: +49-8751-842876  
e-mail: [info@elca.de](mailto:info@elca.de)  
[www.elcamino.de](http://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.

