Compact Verilog

Online Live

Workshop

Applicable Technologies	Requirements	Contact
All FPGA technology	Basic knowledge of digital technology	Michael Schwarz P. +49 7664 91313-15 E. info@plc2.de

Fee (net per person)	Inclusive	Duration
OL € 1,900	Training material	3 days
WO € 2,300	Plus beverages during breaks Lunch	3 days

Workshop

The continuously rising demand for highly complex programmable logic in combination with ever-increasing clock rates create new challenges for the users. Speed, flexibility and highest-quality results are crucial factors for the economic success of a company. To achieve these goals, users must adopt modern and powerful Hardware Description Language (HDL)-based design techniques. A shortening of the development time as well as high quality designs can both be obtained with a Verilog-based approach.

This workshop teaches the Verilog syntax and its usage. The attendee learns the basic description elements as well as the application areas of this hardware programming language. The theoretical content is complemented with PC-based exercises. For synthesis and simulation, a maximum of two attendees work together on one PC. During a series of exercises, the attendees will describe multiple Verilog modules that build a complex design together. The development cycle will be completed by implementing and porting this onto the provided FPGA evaluation board, thus allowing us to learn about the configuration mechanism and to test the loaded design on real hardware as well.

Attendees of the online live course will do the practical exercises in the afternoon on their own.

Agenda

01. **Hardware modeling overview**
    - History and formalization of Verilog
    - Inference and instantiation

02. **Verilog language concepts**
    - Keywords and identifiers
    - Signals and data types

03. **Memories, modules, and ports**
    - Port rules for hierarchical modules
    - Modeling RAM and ROM

04. **Operators and expressions**
    - Using operators
    - Signed and unsigned objects

05. **Data flow modeling**
    - Continuous assignment
    - Delay specifications

06. **Tasks and functions**
    - Subprograms, tasks, functions
    - Header files

07. **Verilog procedural statements**
    - Procedural statements
    - Blocking and non-blocking statements

08. **Controlled operation statements**
    - If/else statements
    - Case statements

09. **Advanced language concepts**
    - Event-driven simulation
    - Verilog generate statements

10. **Finite state machines**
    - Encoding style
    - Single-process or multi-process FSMs

11. **Targeting Xilinx® FPGAs**
    - Timing and performance guidelines
    - Synthesis tools and options

12. **Advanced Verilog testbenches**
    - Verilog file I/O
    - Writing system tasks