PLC2 Languages

SystemVerilog
Advanced Verification for FPGA Design

Online Live

Workshop

Applicable Technologies	Requirements	Contact
None	Experience with VHDL or Verilog for design and verification	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

Modern FPGA designs have tremendously advanced in both performance and capacity. Verification of this kind of designs has become a daunting task, especially the validation of the design against the specification and test plan.

SystemVerilog provides a comprehensive set of verification tools and is a natural extension to Verilog. It also provides constructs with clearer intent like enumerated types, integrated assertions and higher language constructs, which support design hierarchy and Object-Oriented Programming (OOP). Powerful testbench features allow for more flexible and reusable testbench development, even in the context of a VHDL-based design.

This workshop will give an overview about the SystemVerilog language and will introduce into new verification methodologies »Assertion-Based Verification«, »Constrained Random Generation« and »Functional Coverage«.
The attendee will learn how to use these powerful verification tools to speed-up verification as well as to measure the verification progress and how these methodologies can be naturally applied to the verification of VHDL designs.

Due to accompanying exercises, the course offers in-depth and practice-oriented training. Attendees of the online live course will do the practical exercises in the afternoon on their own.

Agenda

01. **Motivation**

02. **Introduction to SystemVerilog**
    - Types of data
    - Fields and structures
    - Flow control
    - Hierarchy
    - Tasks, functions, dynamic processes
    - Classes in OOP

03. **SystemVerilog assertions**
    - SVA language construct
    - Sequential expressions
    - Property expressions
    - Assert and cover directive
    - Blind statement

04. **Constraint randomisation**
    - Constraints
    - Random case
    - Random sequence

05. **Functional coverage**
    - Covergroup
    - Coverpoint
    - Cross