UVM Testbench Made Easy

Online Live

Workshop

Applicable Technologies	Requirements	Contact
None	Knowledge of SystemVerilog and OOP concepts are advantageous	Michael Schwarz P. +49 7664 91313-15 E. info@plc2.de

Fee (net per person)	Inclusive	Duration
OL € 1,300	Training material	2 days
WO € 1,700	Plus beverages during breaks Lunch	2 days

Workshop

Today's FPGA designs have become complex SoC type of designs which has driven the complexity to a level that used to be specific to ASICs some years in the past.
For a complete verification of those systems, writing testbenches has become a challenging task. Different verification methodologies have addressed this on different levels. The most comprehensive approach is the Universal Verification Methodology (UVM). It has become a standard (IEEE 1800.2-2017). It provides SystemVerilog verification base components that can be used to create a testbench infrastructure with a high reuse potential.
Since the UVM library is very complex, building a testbench from scratch is a time-consuming task and requires a good knowledge of the tools the library provides. In order to help verification engineers to build a testbench infrastructure very quickly, UVM Framework has been developed. With UVM Framework a UVM testbench can be created very rapidly and with a few changes the testbench is ready for simulation within a few hours.

The workshop »UVM Made Easy for FPGA Designers« will introduce the most important UVM building blocks to provide a basic understanding of how a UVM testbench looks like, how the component instance creation process works, and how the verification components communicate with each other and the DUT.

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. Introduction

02. UVM
    - UVM verification components
    - UVM transaction interfaces
    - UVM factory
    - UVM configuration database

03. UVM Framework
    - UVMF base classes
    - UVMF base class package
    - Introducing the UVMF API
    - Practical example: creation of a UVM testbench
    - Introducing the interface API
    - Introducing the environment API
    - Introducing the testbench API
    - Conclusion

04. References

05. Contact information

Exercices
01. Run the UVM example
02. UVM transaction interfaces
03. UVM factory
04. Generating the FPU interface
05. Generating the FPU environment
06. Generating the FPU testbench
07. Completing the testbench
08. Adding another test