SynthWorks

Expert VHDL Training for Hardware Design and Verification

PLC2

Workshop Compact VHDL Testbenches and Verification with OSVVM

Today's FPGA and ASIC designs have drastically increased in size and complexity since the very beginning of digital hardware design. These elaborate circuits are described as a hierarchy of sub-systems in hardware description languages like VHDL. The sub-systems are most likely connected by standardized bus infrastructures like AXI, PLB, Avalon or WishBone. In addition, these system might add a soft CPU IP core or an embedded ARM CPU core. Such a design is way too complex to verify it with simple, assertion-based testbenches.

With Open Source VHDL Verification Methodology (OSVVM) a structured approach is given, that increases the reusability of testbench code. OSVVM is a free and open source available VHDL library that offers packages, data types, subprograms and algorithms that are needed in almost every testbench. There is no need to reinvent the wheel again and again. The latest feature of OSVVM is a predefined set of verification IPs, so a wide range of standard bus interfaces is covered.

OSVVM is offering a methodology that comprises the following topics: transaction-based modeling (TBM), self-checking, scoreboards, memory modeling, functional coverage, directed, algorithmic and constrained random as well as intelligent testbench test generation. An VHDL testbench environment based on OSVVM is as powerful as other competitive verification languages like SystemVerilog or 'e'.

This course starts with simple testbenches and progressively increases the level of abstraction. Along the way students learn about: subprogram usage, libraries, file reading and writing, modeling issues, transaction-based testbenches, bus functional models, transaction-based models, record types, resolution functions, abstractions for interface connectivity, model synchronization methods, protected types, access types (pointers), data structures (e.g. scoreboards), directed, algorithmic, constrained random, and coverage driven random test generation, self-checking (result, timing, protocol checking and error injection), functional coverage, representation of analog values and periodic waveforms, timing and execution of code, test planning, and configurations.

This 3-day course concentrates on theory, that is needed to implement efficient VHDL testbench environments. The labs are reduced to a minimum, due to the short course duration.

Applicable technologies

All (FPGA technology independent)

Requirements

Advanced knowledge in VHDL and digital circuit design (e.g. Professional VHDL)

Duration and Costs:

Duration: 3 days

Cost: € 2.100 net per person, including detailed training material, drinks in the breaks and lunch

Agenda

- From Basics to subprograms
- Transaction-based models (TBM / BFM)
- Elements of a transaction-based model
- Data structures for verification
- Creating tests
- Constrained random testing
- Functional coverage
- Execution and timing
- Advanced coverage
- Advanced randomization
- Test plans