

Compact VHDL for Synthesis

Online Live

Workshop

Applicable Technologies	Requirements	Contact
Independent of FPGA technologies	Basic knowledge of digital circuit design	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

Programmable logic devices like FPGAs have been established in daily life. They can be found in mobile phones, IoT devices, cars, or cloud data centers. Their area of operation is as broad as their size. FPGAs are used as, but are not limited to: protocol adapters, signal converters, or accelerators for video, radar, and sensor data processing.

The design of digital circuits on this scale needs a powerful hardware description language which offers different levels of abstraction, so an engineer can create a digital hardware design in a quick and effective way. VHDL fulfills these requirements.

VHDL is a strongly typed hardware description language that prohibits typical programming mistakes in the coding phase. Usually, VHDL is used on the Register Transfer Level (RTL) to design digital circuits of any complexity. Apart from the language

constructs for synthesis, VHDL offers a wide range of functionality to describe complex verification models. Thus, it is possible to verify digital designs from simple gate-up to complicated System on Chip (SoC) before going to lab tests. This workshop will teach the synthesis of relevant aspects of the VHDL hardware description language based on the IEEE Std. 1076-2008 language revision.

Basic knowledge of digital circuit design (gates, multiplexers, flip flops, memories), as well as the understanding of basic principles of scripting or programming languages are welcome.

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

- | | |
|--|--|
| <p>01. Introduction to VHDL
Language concept
Hardware modelling techniques
Design flow</p> <p>02. Hardware description with VHDL
Entity/architecture
Configuration
Package
Library/context</p> <p>03. Language constructs
Signals, variables, constants
Processes und concurrency
Control structures
Functions/procedures
Generic design descriptions</p> | <p>04. Strong typing in VHDL
Predefined types
User defined types
Predefined operators
Attributes
Fixed point package</p> <p>05. Description techniques
Finite state machines
Memory</p> <p>06. Test bench concept
Simple testbenches
Assertions</p> |
|--|--|