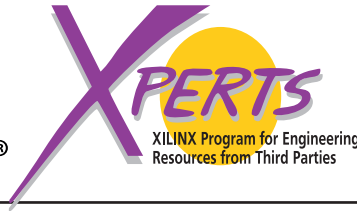




XILINX®



PLC2
PROGRAMMABLE LOGIC
COMPETENCE CENTER

Workshop Compact Verilog

Durch den stetig steigenden Bedarf nach programmierbaren Bausteinen hoher Komplexität verbunden mit hohen Taktraten werden die Anwender mit neuen, ebenfalls stetig wachsenden Herausforderungen konfrontiert. Da Schnelligkeit, Flexibilität und hohes Qualitätsniveau entscheidend für den wirtschaftlichen Erfolg sind, müssen moderne und leistungsfähige, auf Hardware-Beschreibungssprachen basierende Designmethoden eingesetzt werden, um der Forderung nach kürzesten Entwicklungszeiten und hoher Qualität gerecht zu werden.

Der PLC2-Workshop „Compact Verilog“ schult den Anwender im Einsatz von Verilog HDL. Die Teilnehmer lernen die grundlegenden Beschreibungselemente sowie die Einsatzbereiche von Verilog kennen. Die theoretischen

Inhalte werden durch Übungen am PC abgerundet. Zur Verilog Synthese und Simulation arbeiten dabei bis zu maximal zwei Teilnehmer gemeinsam an einem PC und Evaluierungsboard.

Anwendbare Technologien

alle (von Technologie unabhängig)

Voraussetzungen

Grundkenntnisse Digitaltechnik (wie z.B. in „Compact FPGA Schaltungstechnik“ vermittelt) sind hilfreich

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer
inklusive ausführlichen Schulungsunterlagen
sowie Pausengetränken und Mittagessen

Agenda

Hardware Modeling Overview

- History and Formalization of Verilog
- Levels of Abstraction
- Inference and Instantiation

Verilog Language Concepts

- Keywords and Identifiers
- Signals and Data Types
- Numerical Representation
- Arrays

Memories, Modules and Ports

- Port Rules and Restrictions
- Creating Hierarchical Modules
- Modeling RAM and ROM

Introduction to Testbenches

- The Testbench Concept
- Behavioral Coding

Operators and Expressions

- Using Operators
- Signed and Unsigned Objects

Data Flow-Level Modeling

- Continuous Assignment
- Delay Specifications

Tasks and Functions

- Subprograms, Tasks, Functions
- Header Files

Verilog Procedural Statements

- Differences Between RTL and Behavioral Coding
- Procedural Statements
- Blocking and Non-Blocking Statements

Controlled Operation Statements

- if/else Statements
- case Statements
- Loop Statements

Advanced Language Concepts

- Event-Driven Simulation
- Using wait and Local Parameters
- Verilog Generate Statements

Finite State Machines

- FSM Overview
- Encoding Style
- Single-Process or Multi-Process FSMs
- FSM Example

Targeting XILINX FPGAs

- Timing and Performance Guidelines
- Synthesis Tools and Options

Advanced Verilog Testbenches

- Verilog File I/O
- Read Functions
- Write System Tasks