XILINX®

XPERTS
XILINX Program for Engineering Resources from Third Parties

PLC2
PROGRAMMABLE LOGIC COMPETENCE CENTER

Workshop DSP Design using System Generator

The PLC2 workshop "DSP Design using System Generator" demonstrates various ways to implement DSP functions. It focuses on the XILINX System Generator and the linking of the MATLAB/SIMULINK design environment from Mathworks to the XILINX design system. The workshop is thus aimed at system engineers and developers who want to describe and verify DSP systems at an abstract level and implement them with the XILINX System Generator. The PLC2 workshop DSP Design using System Generator" concentrates on system and hardware design. How special DSP algorithms are optimally implemented in XILINX FPGAs is only touched upon; for more detailed information on this we recommend the PLC2 workshop "DSP Implementation Techniques". Using the Matlab environment accelerates the FPGA design work and simplifies the verification. No special prior knowledge is required but you should have a basic knowledge of digital signal processing and FPGAs.

The theoretical content will be rounded off with exercises on the laptop/PC.

Applicable technologies

XILINX FPGAs

Requirements

Knowledge of digital signal processing

Duration and Cost

Duration: 3 days

Cost: € 1.900, – net per person, including detailed training material, drinks in the breaks and lunch.

Agenda

Matlab Simulink Basics
- Stimulus and Response
- Sampling and Resolution
- Design Hierarchy and Subsystems

XILINX System-Generator Basics
- Interfacing and Data Types
- Constructing the Design
- System-Generator Blockset

Digital Filter Implementations
- FIR Filters
- IIR Filters
- Parallel Filters
- Sequential Filters

Verification
- Simulink Verification
- HDL Co-Simulation
- Hardware Verification

Signals in System-Generator
- Signal Conversion
- Signal Extraction

Implementing System Control
- Control Mechanisms
- Block Operations
- Implementing .m File Functions

Multi-Rate Systems
- Sample Rate Conversion
- Simulink Propagation Rules
- Timing Constraints

Designing with Shared Memories
- Memories, Shared Memories
- Register based Interfaces
- BlockRAMs and FIFOs

Achieving Higher Performance
- Timing Analysis Compilation
- Performance Improvements