PLC2

Module Overview

In-House Training

Training Modules

To give you the greatest possible flexibility, we have provided an overview of our complete portfolio into different modules. You can combine these in any way you like to create an in-house training that perfectly fits your needs. The modules are continually being expanded.

Check our website for the latest offerings and our up-to-date training course list. We are open to develop custom training courses on topics that go beyond our existing modules. Feel free to talk to us directly.

↘ Read more on our website

Training course overview | In-House training

How can we help?

Call the PLC2 Training team
+49 7664 91313 15

Send an e-mail
info@plc2.de

AXI
AXI IP Interface - Overview
AXI - Introduction
Creating custom peripherals based on AXI
Creating user peripherals based on AXI

Circuit Design Techniques
Adress based protocols
Arithmetic basics
Basic circuits (encoder, decoder)
Clock Domain Crossing
Codes and Protocols
Combinatorial Circuits
Metastability
Sequential Circuits
State Machines
Streaming based protocols
Synchronisations Circuits

DSP
Creating custom blocks in Vitis™ Model Compose
Digital filter design and implementation
Hardware Co-Simulation using AMD System Generator
Implementing DSP functions using Vitis Model Composer
Introduction to a model-based DSP design
Matlab and Simulink for AMD* System Generator
Model Composer optimized block library
Optimization methodologies using Vitis Model Composer
System Generator Design Flow
Transforming algorithmic specifications
Vivado™ ML integration

Embedded
Application Debugging with SDK / Vitis
Application Profiling with SDK / Vitis
Basics of a Micro Processor / Microcontroller
Bus Functional Modelling (BFM)
C/C++ Primer for Zynq™
Embedded C/C++ for Zynq
Hardware Co-Debugging
Interrupts
MicroBlaze
PicoBlaze

Embedded Linux
Boot loader and root filesystem
Debugging and Performance
Driver development
Embedded Open-Source Linux
Kernel configuration
Linux for MicroBlaze: PetaLinux
Petalinux
Yocto

FPGA Board Design	
Clocking Concepts and Realizations	PCB Details for Board Planning / Design Process
Configuration Board Design	Power Integrity – Advanced
Connectivity Kits – Connectivity	Power Integrity – Basics
FPGA Power Supply – Requirements and Solutions	Reflection/Crosstalk
FPGA Requirements and Limitations for Board Design	Signal Interfacing Options and Realizations (SelectIO domain)
FPGA Thermal Design	SI-Models and Tools
General Introduction to SI Problems	SI System Analysis – General
High-Speed Signal Interfacing	SI System Analysis – Memory Interfaces
Introduction to High-Speed Connectivity (Transceiver, Memory, PCIe)	SI System Analysis – Serial Transceivers
Introduction to SelectIO Connectivity	Targeted Reference Design Overview
Options and Realizations	Transmission Lines

High-Speed Memory Interfacing
FPGA Resources for Memory Interfaces
Memory Controller (all Families)
Memory Interface Board Design
Memory Interface Design / Simulation / Implementation
Memory Interface Signal Integrity Analysis
Memory Interface Test and Debugging
Memory Devices
Practical Example Designs / Labs on Real Hardware

*Xilinx / AMD

Image Processing
Accelerating applications with the KV260 Vision AI Starter Kit
Customizing the AI models
Introduction to Vitis video analytics SDK
Kria™ SoM carrier card design guide
Vitis acceleration libraries

ISE	
CORE Generator Software Integration	ISE Design Tool Flow
Creating Hardware Peripherals in XPS	Placing Dedicated Resources
Creating Software Drivers for User Peripherals	PlanAhead Software Benefits and Features Overview
Debugging with the ChipScope Pro Tool	PlanAhead Software Review
Design Development and Analysis	Project Navigator Integration with the PlanAhead Software
Design Preservation with Partitions	Routing Optimization in FPGA Devices
Floorplanning Case Studies	SDK Software Design Flow
Floorplanning Techniques	Static Timing Analysis with the PlanAhead Software
I/O Pin Planning	Tcl Scripting in the PlanAhead Software
Introduction to Pblocks	XPS Hardware Design Flow

Networking
1GE / 10GE / 40GE / 100GE
Ethernet Packet Processing
Interlaken IP
Packet Buffering
Realization of exceptionally large FPGAs
Traffic Management

PCI Express
Application Focus: DMA
Compliance and Debugging
Connecting Logic to the Core – Local Link
Endpoint Application Considerations
FPGA Root Port
Interrupts and Error Management
Introduction to the PCIe Architecture
Mechanicals, Hot Plus and Power
Packet Formatting Details
PCIe and the CORE Generator Interface
Review of the PCIe Protocol
Simulating a PCIe System Design

Serial Transceiver (general or transceiver / family specific)
Basic Transceiver Principles (PCS Domain)
Transceiver Applications (Protocols, Standards, Examples)
Transceiver Board Design (PCB, Power, Signal Interfacing)
Transceiver Design / Simulation / Implementation
Transceiver Overview
Transceiver Signal Integrity (Basics, Simulation, timation)
Transceiver Test & Debugging
Practical Example Designs / Labs on Real Hardware
Physical Link Optimization (PMA layer options and tools)

Silicon
CLB Architecture
Clocking Resources
Dedicated Hardware
DSP Resources
FPGA Overview General
FPGA Ultrascale / UltraScale+ Architecture
FPGA 7-Series
I/O Resources
Memory Controllers
Memory Recources
Slice Flip-Flops
Versal™
Zynq Ultrascale+ MPSoC
Zynq Ultrascale+ RFSoC
Zynq-7000

SystemVerilog
Additional Operators in SystemVerilog
Assertions
Coverage
Data Types
Direct Programming Interface (DPI)
Functions, Tasks and Packages
Interfaces
Introduction to SystemVerilog
Procedural Statements and Flow Control
Randomization
Structure, Unions, and Arrays

TCL / TK
TCL / TK Event-Driven Programming
TCL / TK Integration with C/C++
TCL / TK Overview

Timing Constraints	
Accessing the design database	Multicycle paths
Advanced I/O interface constraints	Multiple clocks
Advanced timing analysis	Setup checks and clocks
Basic timing constraints and reports	Static timing analysis and clocks
Basic static timing analysis	Timing Budget of digital circuits
False paths	Timing Closure
Hold checks	Timing exceptions
Input and output constraints	Timing reports
Max / min delay exceptions	

Verilog
Advanced Language Concepts
Advanced Verilog Test Benches
Controlled Operation Statements
Data-Flow Level Modelling
Finite State Machines (FSM)
Hardware Modelling Overview
Introduction to Test Benches
Modules and Ports
Verilog Language Concepts
Verilog Operators and Expressions
Verilog Procedural Statements
Verilog Tasks and Functions

Versal Adaptive SoC			
AI Engine application debug and trace	Data types: scalar and vector data types	Overview of embedded hardware development	Versal adaptive SoC software build flow: PetaLinux and Yocto
Application development and debugging	Design tool flow summary	Scalar engines, adaptable and intelligent engines	Versal AI Engine DSP library and Model Composer overview
Application partitioning on Versal adaptive SoC	Driving the IP integrator tool	Software stack for Versal adaptive SoC	Versal AI Engine memory and data movement
AI Engine API and intrinsic functions	Driving the Vitis software development tools	System design flow with Vitis	Versal AI Engine tool flow with Vitis
Advanced intrinsic functions	Introduction to debugging with AI Engine kernels	System simulation	Versal application partitioning
Advanced graph input specifications	Introduction to system integration and validation methodology	The adaptive data flow graph	Versal platform management controller
AI Engine interfaces	NoC and memory introduction and concepts	The programming model: multiple kernels	Versal programming interfaces
Boot and configuration	Operating system support	The programming model: single kernel	Vitis analyzer
Clocks, resets, and I/O connectivity	Outlook on acceleration tool flow	Versal adaptive SoC architecture introduction	Window and streaming data APIs
Data communications	Overview of AI Engine kernel optimization	Versal adaptive SoC processing system	

VHDL

Checking the Behaviour	HDL Coding Techniques	Structural Modelling
Concurrent and Sequential Statements	Introduction to VHDL	Subprograms
Configuration and IP-Cores	Loop Statements	The Build-In Timing Model
Define your own Packages	Modelling of external Components	VHDL Attributes
Designing re-usable Components	Modelling with VHDL	VHDL Build-In Packages
File I/O with VHDL	Overview Assertion Based Verification	VHDL Operators
Finite State Machines	Processes	VHDL Test Bench Concept
Generating Test Bench Stimulus	Review of Basic VHDL	VHDL Type Concept
Generics		

Vivado

Accessing the design database	I/O Pin Planning and Clock Constraints	Visualization for analysis designing with IP
Creating Hardware Peripherals in Vivado	Project-based and non-project flows	Vivado design flows
Creating Software Drivers for Users Peripherals	Reset methodology	Vivado Embedded Design Flow
Designing with IP	Scripting using project based and non-project batch flows	Vivado IDE Overview and Projects
DFX Dynamic Function eXchange	Single data rate vs. double data rate	Vivado IP Integrator
Floorplanning Techniques	Software Options for Optimizations	Vivado IP Packager
FPGA design methodology	Source-synchronous interfaces	Vivado Logic Analyzer
FPGA design methodology summary	Synchronization circuits and the clock interaction report	Vivado Simulator (XSIM)
HDL coding techniques	System-synchronous interfaces	Vivado Tool Flow
Introduction to Pblocks	TCL in the Vivado IDE	Working with IP/IP integrator
Introduction to the Vivado Design Suite	Visualization for Analysis	

Zynq UltraScale+ MPSoC

Arm TrustZone® technology
Baremetal software platform development
Deploying OpenAMP in a heterogeneous system
Driving the IP integrator tool
Driving the Vitis tool
Embedded UltraFast™ design methodology
FreeRTOS
Linux application development and debugging
Linux builds using PetaLinux and Yocto
Overview of embedded hardware development
Platform Management Unit (PMU) development
Power management using the PMU
The Arm® processing units APU and RPU
Understanding device drivers
Zynq UltraScale+™ MPSoC architecture
Zynq UltraScale+ MPSoC boot and configuration
Zynq UltraScale+ MPSoC hardware /software virtualization
Zynq UltraScale+ MPSoC software stack