

EMBD23000-14-ILT (v1.0)

Course Specification

Course Description

This two-day course introduces you to software design and development for the Xilinx Zynq™ All Programmable System on a Chip (SoC) using the Xilinx Software Development Kit (SDK). You will learn the concepts, tools, and techniques required for the software phase of the design cycle.

Topics are comprehensive, covering the design and implementation of the board support package (BSP) for resource access and management of the Xilinx Standalone library. Major topics include device driver use and custom development and user application debugging and integration. Practical implementation tips and best practices are also provided throughout to enable you to make good design decisions and keep your design cycles to a minimum. You will have enough practical information to start developing software applications for the ARM® Cortex™-A9 and MicroBlaze™ processors.

Additionally, this course covers developing software applications for a Xilinx embedded system based on a MicroBlaze processor.

Level – Embedded Software 3

Course Duration – 2 days

Price – \$1200 or 12 Xilinx Training Credits

Course Part Number – EMBD23000-14-ILT

Who Should Attend? Software design engineers interested in system design and implementation and software application development and debugging using the Xilinx Standalone library

Prerequisites

- C or C++ programming experience, including general debugging techniques
- Conceptual understanding of embedded processing systems including device drivers, interrupt routines, writing / modifying scripts, user applications, and boot loader operation

Software Tools

- Xilinx ISE® Design Suite: Embedded or System Edition 14.2

Hardware

- Architecture: Zynq-7000 All Programmable SoC and 7 series FPGAs*
- Demo board: Zynq-7000 All Programmable SoC ZC702 or Zed or Kintex™-7 FPGA KC705 board*

* This course focuses on the Zynq All Programmable SoC and 7 series FPGA architectures. Check with your local Authorized Training Provider for the specifics of the in-class lab board or other customizations.

After completing this comprehensive training, you will have the necessary skills to:

- Implement an effective software design environment for a Xilinx embedded system using the Xilinx SDK tools
- Write a basic user application (under Standalone or Linux) using the Xilinx Software Development Kit (SDK) and run it on an embedded system platform
- Use Xilinx debugger tools to troubleshoot user applications
- Apply software techniques to improve operability
- Maintain and update software projects with changing hardware

Course Outline

Day 1

- Processors, Peripherals, and Tools
- Standalone Software Platform Development
- Linux Software Application Development
- **Lab 1:** Basic System Implementation
- Software Development Using SDK
- Writing Code in the Xilinx Standalone Environment
- Writing Code in the Xilinx Linux Environment
- **Lab 2:** Application Development

- Address Management
- Interrupts
- **Lab 3:** Software Interrupts

Day 2

- Software Platform Download and Boot
- Application Debugging
- **Lab 4:** Debugging
- Application Profiling
- **Lab 5:** Standalone BSP Application Profiling
- Writing a Custom Device Driver
- **Lab 6:** Writing a Device Driver
- Advanced Services and Operating Systems
- Project Management with the Xilinx Design Tools
- **Lab 7:** File Systems

Lab Descriptions

- **Lab 1:** Basic System Implementation – Construct the hardware and software platforms used for the course labs. Begin with Processing System Configuration Wizard (Zynq All Programmable SoC) or Base System Builder (Microblaze processor) to create the hardware design. Specify a basic software platform and add an application to the system.
- **Lab 2:** Application Development – Create a simple software application project with the provided source files for a software loop-based stopwatch. Verify proper BSP settings and linker script generation. Use API documentation for the GPIO peripheral to complete the software application. Verify proper operation of the stopwatch in hardware.
- **Lab 3:** Software Interrupts – Replace a software timing loop with an interrupt-driven timer. Add the timer software and implement an interrupt handler for the timer. Download into hardware and test the application.
- **Lab 4:** Debugging – Launch the SDK debug perspective and the previous lab's stopwatch application for debugging, setting breakpoints, calculating interrupt latency, and stepping through the program's operation.
- **Lab 5:** Standalone BSP Application Profiling – Profile a program, interpret reports, then enable cache and modify code to archive optimal performance.
- **Lab 6:** Writing a Device Driver – Create a custom LCD services device driver and add it to the Standalone library BSP. Download to hardware and verify proper device driver operation.
- **Lab 7:** File Systems – Implement a standalone software platform that incorporates the XilMFS memory file system. Develop an application that performs file-related tasks on external memory.

Register Today

To register for this course or to see a list of currently scheduled classes, please visit our secure [Online Store](#).

To request a public or private class, or inquire about course offerings, please contact Faster Technology through one of the following:

Web: [Request a Class](#)

Email: registrar@fastertechnology.com

Phone: 281-391-5482

Visit www.FasterTechnology.com/training-courses to see our full line of Xilinx education courses.

