

Course Description

This two-day course introduces you to software design and development for Xilinx embedded processor systems. You will learn the basic tool use and concepts required for the software phase of the design cycle, after the hardware design is completed.

Topics are comprehensive, covering the design and implementation of the software platform for resource access and management. Major topics include device driver 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 get started developing the software platform for a Xilinx embedded system based on a PowerPC® 440 or MicroBlaze™ processor.

While this course includes many of the topics presented in the *Embedded Systems Development* and *Advanced Features and Techniques of Embedded Systems Development* courses, the focus is on software development concepts and practices rather than hardware development. Hardware design concepts and procedures are *not* covered.

Level – Embedded Software 3

Course Duration – 2 days

Price – \$1400 or 14 training credits

Course Part Number – EMBD23000-12-ILT

Who Should Attend? Software and hardware design engineers interested in system design and implementation, platform software support, and software application development and debugging. This course is not for the hardware-only embedded designer.

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 12.1

Hardware

- Architecture: Spartan®-6, Virtex®-5, and Virtex-6 FPGAs*
- Demo board: Spartan-6 FPGA SP605 or Virtex-5 FPGA ML507 board*

* This course focuses on the Spartan-6, Virtex-5, and Virtex-6 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 using the Xilinx Software Development Kit (SDK) and run it on the embedded system
- Use Xilinx debugger tools to troubleshoot user applications
- Apply software techniques to improve operability
- Reduce embedded software development time

Course Outline

Day 1

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

- **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:** SDK 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 labs. Begin with Base System Builder to create the hardware design. Specify a basic software platform and add a software application to the system.
- **Lab 2:** Application Development – Create a simple software application project from source files for a software loop-based stopwatch. Research hardware and software documentation to complete the application; then download it to hardware.
- **Lab 3:** Software Interrupts – Replace a software timing loop with an interrupt-driven timer. Add the timer software and write an interrupt handler for the timer. Configure the FPGA, download, and test the application.
- **Lab 4:** Debugging – Set up 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:** SDK Profiling – Profile a program, interpret reports, then enable cache and rewrite code to archive optimal performance.
- **Lab 6:** Writing a Device Driver – Create the skeleton driver framework, add an LCD device driver to the BSP, and verify proper device driver operation via a download to hardware test.
- **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

Vai logic, the Authorized Training Provider (ATP) for Indiana, Michigan, Ohio, Kentucky, and western Pennsylvania offers public and private training.

Please visit www.vaitechnology.com for more information, to view schedules, or to register.

Please send inquiries to info@vaitechnology.com, or contact the registrar at (317) 570-0707.

