

Course Description

This course introduces the concepts, tools, and techniques required for software design and development for the Zynq® System on a Chip (SoC) and Zynq UltraScale+™ MPSoC using the Vitis™ unified software platform.

The focus is on:

- Reviewing the basics of Vitis tool use
- Customizing board support packages (BSPs) for resource access and management of the Xilinx Standalone library
- Utilizing device drivers effectively
- Developing software applications for the Arm® Cortex™-A9 processors
- Debugging and integrating user applications
- Employing best practices to enable good design decisions

What's New for 2021.2

- All labs have been updated to the latest software versions

Level – Embedded Software 3

Course Details

- 2 days ILT
 - 21 lectures
 - 13 labs
 - 7 ILT demos / 5 OnDemand demos

Price –

Course Part Number – EMBD-SW

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 and modifying scripts, user applications, and boot loader operation

Software Tools

- Vitis unified software platform 2021.2
- Vivado® Design Suite 2021.2

Hardware

- Architectures: Zynq-7000 SoC (Cortex-A9 processor) and Zynq UltraScale+ MPSoC (Cortex-A53 and Cortex-R5 processor)
- Demo board: Zynq UltraScale+ MPSoC ZCU104 or Versal® ACAP VCK190 board

* This course focuses on the Zynq-7000 SoC and the Zynq UltraScale+ MPSoC 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 software development tools
- Write a basic user application (under Standalone or Linux) using the Vitis unified software platform and run it on an embedded system
- 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

- **Overview of Embedded Software Development**
Overview of the process for building a user application. {Lecture}
- **Embedded UltraFast Design Methodology**
Outlines the different elements that comprise the Embedded Design Methodology. {Lecture, Demo}
- **Zynq-7000 SoC Architecture Overview**
Overview of the Zynq-7000 SoC architecture. {Lecture, Lab, Demo}
- **Zynq UltraScale+ MPSoC Architecture Overview**
Overview of the Zynq UltraScale+ MPSoC architecture. {Lecture, Demo, Lab}
- **Zynq UltraScale+ MPSoC Software Environments**
Describes the software development environments for Zynq UltraScale+™ MPSoCs. {Lecture}
- **Driving the Vitis Software Development Tool**
Introduces the basic behaviors required to drive the Vitis tool to generate a debuggable C/C++ application. {Lecture, Demo, Lab}
- **System Debugger**
Describes the basics of actually running a debugger and illustrates the most commonly used debugging commands. {Lecture, Lab}
- **Standalone Software Platform Development and Coding Support**
Covers the various software components, or layers, supplied by Xilinx that aid in the creation of low-level software and includes a discussion on drivers, domains, operating systems, and libraries. Also covers the basic services (libraries) available when coding in the Standalone environment. {Lecture, Demo, Lab}
- **FAT File System for Standalone**
Introduces the FAT file system (FFS) from the Standalone/Bare-metal library. The FFS provides drivers and utilities for effectively converting a region of memory into a file system. {Lecture, Lab}
- **Using Linker Scripts**
Overview of the purpose and typical use of a linker script. {Lecture, Lab}
- **Migrating from SDK to the Vitis Platform**
Overview of migrating existing Xilinx SDK projects to Vitis software development projects. {Lecture, Demo}
- **Introduction to Interrupts**
Introduces the concept of interrupts, basic terminology, and generic implementation. {Lecture}
- **Software Interrupts: Writing**
Describes many of the considerations that a software coder must take into account when supporting interrupts. {Lecture, Lab}

Day 2

- **Operating Systems: Introduction and Concepts**
Introduces the concept of the operating system and provides a simplified view into the generic way that operating systems work. {Lecture}

- **Linux: A High-Level Introduction**
Introduces the Linux operating system, a brief history, and how to use it. {Lecture}
- **Linux Software Application Development Overview**
Highlights important parts of the underlying Linux system as it pertains to applications. {Lecture, Demo, Lab}
- **Driving the PetaLinux Tool**
Introduces the basic concepts required to build an application using the PetaLinux tool. {Lab}
- **Building a Linux Application in the Vitis IDE**
Provides an introduction to using the Xilinx Vitis IDE tool for Linux software development. {Lecture, Demo}
- **Booting Overview**
Describes the main points to how booting a processor is handled for Zynq SoC devices and MicroBlaze processors. {Lecture, Lab}
- **Software Profiling Overview**
Introduces the purpose and techniques for profiling a user application. {Lecture, Lab, Demo}
- **Understanding Device Drivers**
Explains the concept of a device driver and how it is used by embedded systems. {Lecture, Demo}
- **Custom Device Drivers**
Describes how to successfully write a custom device driver. {Lecture, Lab}