AMD
together we advance_

Operating Systems and Hypervisors in Adaptive SoCs Spec Sheet

SOC-OS-HYPER (v1.0)

Course Specification

Course Description

This course provides software developers options and techniques for selecting and implementing various types of operating systems and hypervisors on AMD Zynq™ UltraScale+™ and Versal™ devices.

The emphasis is on:

- Exploring the capabilities of the application processing unit (APU) and real-time processing unit (RPU) relative to performance improvement and OS implementation
- Reviewing the catalog of OS implementation options, including Arm® TrustZone technology, hypervisors, and various Linux® implementations
- Applying various power management techniques for Zynq UltraScale+ and Versal devices

What's New for 2025.2

- Added labs:
  - QEMU: Linux Application Development and Debugging
  - Xen Hypervisor: Configuration and Use
- Added modules:
  - Overview of the Embedded Development Framework
  - Application Development and Deployment Using the EDF
  - Migration from PetaLinux to the Embedded Development Framework
  - Yocto Build Flow for AMD SoCs
  - Linux OS Integration & Yocto Customization
- All labs have been updated according to the latest tool release

Level – Embedded Software 3

Course Details

- 3 days ILT or 4 sessions/30

Course Part Number – SOC-OS-HYPER

Who Should Attend? – Software developers interested in understanding popular OS and hypervisor options and other high-level system design issues.

Prerequisites

- General understanding of C coding
- Familiarity with issues related to complex embedded systems

Software Tools

- Vivado™ Design Suite 2025.2
- Vitis™ Unified IDE 2025.2
- Hardware emulation environment:
  - VirtualBox/CloudShare
  - QEMU
  - Ubuntu® desktop

Hardware

- Zynq UltraScale+ MPSoC ZCU104 board*
- Versal adaptive SoC VCK190 board

* This course focuses on the Zynq UltraScale+ and Versal architectures. Check with your local Authorized Training Provider for the specifics of the in-class lab environment or other customizations.

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

- Leverage the innate capabilities of the application processing unit (APU) and real-time processing unit (RPU)
- Investigate Arm TrustZone technology
- Explore the concept of hypervisors
- Implement Linux solutions, including asymmetric multiprocessing (AMP) and symmetric multiprocessing (SMP) configurations
- Able to build and customize Linux OS images using the EDF with standard Yocto Project build workflows for AMD platforms
- Deploy FreeRTOS in the RPU
- Effectively use power management strategies

Course Outline

Day 1

- **Application Processing Unit**
  Introduction to the members of the APU, specifically the Arm® Cortex®-A53 processor and how the cluster is configured and managed. {Lectures, Lab}
- **Real-Time Processing Unit**
  Focuses on the real-time processing module (RPU) in the PS, which is comprised of a pair of Arm Cortex processors and supporting elements. {Lectures, Demo, Lab}
- **Arm TrustZone Technology**
  Illustrates the use of Arm TrustZone technology. {Lectures}
- **QEMU**
  Introduction to the Quick Emulator, which is the tool used to run software for a device when hardware is not available. {Lectures, Demo, Labs}
- **HW-SW Virtualization**
  Covers the hardware and software elements of virtualization. {Lectures}

Day 2

- **Multiprocessor Software Architecture**
  Focuses on how multiple processors can communicate with each other using both software and hardware techniques. {Lecture}
- **Xen Hypervisor**
  Discusses generic hypervisors and reviews some of the details of implementing a hypervisor using Xen. {Lectures, Demo, Lab}
- **OpenAMP**
  Discusses how the OpenAMP framework can be used to construct systems containing both Linux and Standalone applications within the APU. {Lectures}
- **Linux**
  Describes how to configure Linux to manage multiple processors. {Lectures}
- **EDF**
  Introduces the AMD Embedded Development Framework (EDF), explains its hardware and software development flows using Yocto-based workflows, and outlines how common PetaLinux build and customization tasks migrate to the EDF flow. {Lectures}
- **Yocto**
  Introduces the Yocto Project and its embedded Linux build system, explains how to configure it to generate Linux images for AMD SoCs, and covers building and customizing the Linux OS using Yocto for AMD platforms. {Lectures, Lab}

Day 3

- **FreeRTOS**
  Overview of FreeRTOS with examples of how it can be used. {Lectures, Demo, Lab}
- **Software Stack**

Introduction to what a software stack is and a number of commonly used stacks. {Lectures, Demo}

- **Power Management**

  Introduction to the concepts of power requirements in embedded systems and the Zynq UltraScale+ MPSoC. {Lectures, Lab}