AMD
together we advance_

Designing with the Versal Adaptive SoC: Design Methodology

VER-DM (v1.0)

Course Specification

Course Description

Use different AMD Versal™ adaptive SoC design methodologies and techniques for developing designs targeting Versal devices. Also learn how to apply application mapping and partitioning, design closure, power, and thermal solutions to enhance the performance of a design.

The emphasis of this course is on:

- Demonstrating the embedded software development flow for Versal devices
- Demonstrating the AI Engine development flow
- Using the provided design tools and Versal adaptive SoC design methodologies to create complex systems
- Leveraging the Power Design Manager (PDM) tool for power estimation
- Identifying Versal adaptive SoC power and thermal solutions
- Enabling top-level RTL flows for Versal devices
- Applying common timing closure techniques
- Performing device configuration and debugging
- Improving Versal adaptive SoC system performance
- Performing system-level simulation

This course provides virtual access to a Versal VCK190 Evaluation Platform to complete one of the two board-based labs included in the course.

Level – VER 2

Course Details

- 3 days ILT or 4 sessions 30 20261

Course Part Number – VER-DM

Who Should Attend? – Software and hardware developers, system architects, and anyone who wants to learn about the Versal adaptive SoC design methodologies

Prerequisites

- Basic knowledge of AMD FPGAs and adaptive SoCs
- Basic knowledge of the Vivado™ and Vitis™ tools
- *Designing with the Versal Adaptive SoC: Architecture*

Software Tools

- Vivado Design Suite
- Vitis Unified IDE
- Power Design Manager

Hardware

- Architecture: Versal adaptive SoC
- Demo board: Versal VCK190 Evaluation Platform

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

- Describe the embedded software development flow for AMD Versal devices
- Describe the AI Engine development flow
- Use the provided design tools and Versal adaptive SoC design methodologies to create complex systems
- Leverage the Power Design Manager (PDM) tool for power estimation for Versal devices
- Identify Versal adaptive SoC power and thermal solutions
- Create a custom AMD Vitis platform to run acceleration applications
- Utilize the modular NoC design entry flow for Versal devices
- Identify and apply common timing closure techniques
- Describe the different configuration and debugging options available for the Versal adaptive SoC
- Perform system-level simulation

Course Outline

Day 1

- **Embedded Software Development**
  Describes the software development environments and embedded software development flows for Versal devices. Also introduces embedded software debugging. {Lecture, Lab}
- **Using the AMD Embedded Development Framework (EDF)**
  Discusses the AMD Embedded Development Framework (EDF) and its role in accelerating platform-level development. Also explains the developer roles in the EDF. {Lecture, Lab}
- **Introduction to the Software Hardware Exchange Loop (SHEL) Flow**
  Outlines the Software Hardware Exchange Loop (SHEL) Flow as a part of the EDF and reviews the key tools used in the flow such as SDTGen, Lopper, and gen-machine-conf. {Lecture}
- **Software Stack**
  Reviews the Versal device bare-metal, FreeRTOS, and Linux software stack and their components. {Lecture}
- **AI Engine Programming: Kernels and Graphs**
  Investigates AI Engine kernels and Adaptive Data Flow (ADF) graphs along with their programming flows. {Lecture, Lab}
- **System Design Planning Methodology**
  Describes system design planning, power, and thermal guidelines. Also reviews system debug, verification, and validation planning. {Lecture}
- **AI Engine System Partitioning**
  Describes the AI Engine system partitioning and planning methodology and mapping system requirements. {Lecture, Lab}

Day 2

- **Power Design Manager**
  Discusses using the new Power Design Manager tool, including import and export functions. {Lecture, Lab}
- **Power and Thermal Solutions**
  Discusses the power domains in the Versal adaptive SoC as well as power optimization and analysis techniques. Thermal design challenges are also covered. {Lecture}
- **Hardware, IP, and Platform Development Methodology**
  Describes the different Versal device design flows and covers the custom platform creation process using the Vivado IP integrator, RTL, HLS, and Vitis environment. {Lecture, Lab}
- **Enabling Top-level RTL Flows**
  Discusses two RTL-centric flows, one for accessing NoC from RTL known as modular NoC flow, and another for GTs with a new GT Wizard Subsystem flow. {Lecture, Lab}
- **Timing Closure Overview**
  Describes the timing closure and baselining of a design. Also explains QoR reports and timing violation analysis. {Lecture}
- **Timing Closure Techniques**
  Reviews the Advanced Flow for implementing Versal devices. Also covers common timing closure techniques for logic

© Copyright 2026 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, Versal, Vitis, Vivado, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective owners.

optimization, design analysis, timing closure and clock skew reduction. {Lecture}

- **Optimizing SLR Crossings in SSI Technology**
  Describes optimizing timing and designs in Versal SSIT devices through efficient SLR crossings and constraints. {Lecture}
- **Board System Design Methodology**
  Highlights PCB, power, clocking, and I/O considerations when designing a system. {Lecture}

Day 3

- **Security Management and Safety Features**
  Describes the security management and safety features of the Versal devices. {Lecture}
- **System Integration and Validation Methodology**
  Describes different simulation flows as well as timing and power closure techniques. Also explains how to improve system performance. {Lecture}
- **Configuration and Debugging**
  Describes the configuration and debug process for the Versal devices, including the Versal device debug interfaces, such as the test access port (TAP) and debug access port (DAP) controller. Also introduces the new PDI debug utility for decoding and analyzing boot configuration errors. {Lecture}
- **Segmented Configuration**
  Discusses the concept, benefits, and implementation of segmented configuration. {Lecture, Lab}
- **Advanced Configuration Editor**
  Describes the Advanced Configuration Editor for defining and managing subsystems, safety, and security requirements. Explores its GUI and demonstrates how to generate and validate device configurations. {Lecture}
- **Overview of HSDP**
  Describes the high-speed debug port (HSDP) in the Versal device. Also goes over the steps to use the SmartLynq+ module for high-speed debugging. {Lecture, Lab}
- **Fabric Debug**
  Explains the fabric debug features available in the Versal devices and reviews the different supported debug IP cores, such as the AXI Debug Hub, AXIS ILA, and AXIS VIO. {Lecture, Lab}
  Note: Virtual access to a Versal VCK190 Evaluation Platform is available for this lab.
- **System Simulation**
  Demonstrates how to perform system-level simulation in a Versal device design. {Lecture, Lab}

.

© Copyright 2026 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, Versal, Vitis, Vivado, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective owners.