# Migrating from UltraScale+ Devices to Versal Adaptive SoCs

VER MGRT (v1.0)

**Course Specification**

## Course Description

This course illustrates the different approaches for efficiently migrating existing designs to the AMD Versal™ adaptive SoC from AMD UltraScale+™ devices. The course also covers system design planning and partitioning methodologies as well as design migration considerations for different system design types.

The emphasis of this course is on:

- Identifying and comparing various functional blocks in the Versal adaptive SoC to those in previous-generation UltraScale+ devices
- Reviewing the approaches for migrating existing designs to the Versal adaptive SoC
- Describing the development platforms for all developers
- Enabling top-level RTL flows for Versal devices
- Identifying design migration considerations for PL-only designs and Zynq™ UltraScale+ MPSoC designs
- Specifying the recommended methodology for planning a system design migration based on the system design type
- Discussing AI Engine system partitioning planning
- Migrating Zynq UltraScale+ MPSoC-based system-level designs to the Versal adaptive SoC
- Detailing Versal device hardware debug features

**What's New for 2025.2**

- All labs have been updated to the latest software versions

**Level** – VER 2

**Course Details**

- 1 day instructor led training (online or in person)
- 11 lectures
- 7 labs

**Price** – $800 or 8 AMD Training Credits

**Course Part Number** – VER MGRT

**Who Should Attend?** – Software and hardware developers, system architects, and anyone who needs to migrate their designs to Versal devices

**Prerequisites**

- Familiarity with designing UltraScale+ FPGAs and adaptive SoCs
- Familiarity with the AMD Vivado™ and Vitis™ tools
- Designing with the Versal Adaptive SoC: Architecture
- Designing with the Versal Adaptive SoC: Design Methodology

**Software Tools**

- Vivado Design Suite 2025.2
- Vitis Unified IDE 2025.2

**Hardware**

- Architecture: Versal adaptive SoC

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

- Identify the different functional blocks in the AMD Versal adaptive SoC
- Utilize high-level system migration steps for efficient migration to the Versal adaptive SoC
- Describe the different tool flows for the Versal adaptive SoC
- Implement a basic Versal NoC design
- Utilize top-level RTL flows such as the modular NoC flow and GT subsystem flow for Versal devices
- Apply design migration guidelines for PL-only and PS+PL designs
- Follow the system design planning methodology
- Describe the AI Engine architecture and programming model as well as follow the AIE system partitioning methodology
- Migrate AMD Zynq UltraScale+ MPSoC system-level designs to the Versal adaptive SoC
- Describe the different debugging options available for the Versal adaptive SoC

## Course Outline

- **Architecture Overview for Existing Users**
  Introduces to students who already have familiarity with AMD SoC architectures the new and updated features found in the Versal devices. Also provides an overview of the Versal portfolio. {Lecture}
- **System Design Migration Approach**
  Describes important system design migration considerations and the high-level system steps for efficient migration to the Versal adaptive SoC. Also compares various functional blocks in the Versal adaptive SoC to those in previous-generation devices. {Lecture}
- **Design Tool Flow**
  Maps the various compute resources in the Versal architecture to the tools required and describes how to target them for final image assembly. {Lecture}
- **NoC Introduction and Concepts**
  Covers the reasons to use the network on chip, its basic elements, design entry flows, and common terminology. {Lecture, Lab}
- **Enabling Top-level RTL Flows**
  Discusses two RTL-centric flows: One for accessing the NoC from RTL, known as the modular NoC flow, and another for the gigabit transceivers with the GT Wizard Subsystem flow. {Lecture, Labs}
- **Programmable Logic Design Migration Considerations**
  Describes Versal adaptive SoC architectural enhancements as well as key programmable logic design migration considerations and best practices. Also covers the advantages of the Advanced Flow for Versal devices. {Lecture, Labs}
- **Processing System Comparison**
  Describes Versal adaptive SoC processing system architectural differences. It covers the differences in the boot sequence between Versal and Zynq UltraScale+ devices. Also introduces the segmented configuration feature for Versal devices. {Lecture}
- **System Design Planning Methodology**
  Describes system design planning, power, and thermal guidelines. Also reviews system debug, verification, and validation planning. {Lecture}
- **AI Engine Architecture Overview and Programming**
  Discusses the Versal AI Engine architecture and explains the programming model for AI Engines with kernels and graph. {Lecture}
- **AI Engine System Partitioning**
  Describes the AI Engine system partitioning and planning methodology and mapping system requirements. {Lecture, Lab}
- **System-level Design Migration**
  Demonstrates how to migrate a Zynq UltraScale+ MPSoC system-level design to the Versal adaptive SoC. Also shows how

to implement the same system design on the Versal AI Engine. {Lab}

- **Debug Overview**
  Describes the tools and techniques available to debug PL and hard blocks in Versal devices. Also covers ChipScoPy APIs, which provide a Python™ interface to program and debug Versal devices. {Lecture}

## Register Today

Morgan Advanced Programmable Systems, Inc. (Morgan A.P.S.) delivers public and private courses in locations throughout the central US region; including Iowa, Illinois, Kansas, Minnesota, Missouri, Nebraska, North Dakota, South Dakota, and Wisconsin.

Visit **morgan-aps.com/training**, for full course schedule and training information.

Morgan Advanced Programmable Systems

- You must have your tuition payment information available when you enroll. We accept credit cards (Visa, MasterCard, or American Express) as well as purchase orders and AMD training credits.

## Student Cancellation Policy

- Student cancellations received more than 7 days before the first day of class are entitled to a 100% refund. Refunds will be processed within 14 days.
- Student cancellations received less than 7 days before the first day of class are entitled to a 100% credit toward a future class.
- Student cancellations must be sent here.

## Morgan A.P.S. Course Cancellation Policy

- We regret from time-to-time classes will need to be rescheduled or cancelled.
- In the event of cancellation, live on-line training may be offered as a substitute.
- Morgan A.P.S. may cancel a class up to 7 days before the scheduled start date of the class; all students will be entitled to a 100% refund.
- Under no circumstances is Morgan A.P.S. responsible or liable for travel, lodging or other incidental costs. Please be aware of this cancellation policy when making your arrangements.
- For additional information or to schedule a private class contact us here.

## Online or in person training with real hardware

- Morgan Advanced Programmable Systems, Inc. has set up a training VPN where engineer participants can take classes online using the same computers and devCards used during in-person training.
- Even better, and upon request, you can use these computers after hours on training days to experiment with labs. This is not possible for in-person training.
- Additionally, just like in-person training, the laptops and devCards, tools, OS, and licensing are set up in advance.
- In some ways, live online-training is better than in-person…for example, you can grant the instructor permission to look at your Vivado, PetaLinux terminal, or Vitis for extended periods of time if your lab is not going exactly has planned to a missed step.
- This is often more comfortable than two engineers crowding around a laptop screen.
- Taking remote training also allows you to learn some tips and tricks for working remote. Whether your devCard is in the lab down the hall, or across the world via VPN, you can control your AMD based device quickly and efficiently.