Designing with the Versal Adaptive SoC: Design Methodology

ACAP-VDM (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

What's New for 2024.2

- Revamped the course to be a three-day course with the addition of new content and features
- Added new modules:
  - AI Engine Programming: Kernels and Graphs
  - AI Engine System Partitioning
  - Enabling Top-level RTL Flows
  - Optimizing SLR Crossings in SSI Technology
  - Segmented Configuration
- Added new labs on:
  - Embedded software development
  - AI Engine system partitioning
  - Kernels and graphs
- Introduced the PDI debug utility in the Configuration and Debugging module
- All labs have been updated to the latest software versions

Level – ACAP 2

Course Details

- 3 days live instructor led training (online or in person)
- 21 lectures
- 9 labs

Price – $2,400 or 24 AMD Training Credits

Course Part Number – ACAP-VDM

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 2024.2
- Vitis Unified IDE 2024.2
- PetaLinux Tools 2024.2

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 devicews
- 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}
- **Software Build Flow**
  Provides an overview of the different build flows, such as the do-it-yourself, Yocto Project, and PetaLinux tool flows. {Lecture, Lab}
- **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}

Designing with the Versal Adaptive SoC: Design Methodology

ACAP-VDM (v1.0)

Course Specification

- **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}
- **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 optimization, design analysis, and timing closure. {Lecture}
- **Optimizing SLR Crossings in SSI Technology**
  Describes optimizing timing and designs in Versal SSIT devices through efficient SLR crossings and constraints. {Lecture}

Day 3

- **Board System Design Methodology**
  Highlights PCB, power, clocking, and I/O considerations when designing a system. {Lecture}
- **Security Management and Safety Features**
  Describes the security management and safety features of the Versal devices. {Lecture}
- **System Integration and Validation Methodology**
  Outlines 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}
- **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}
- **System Simulation**
  Demonstrates how to perform system-level simulation in a Versal device design. {Lecture, Lab}

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 Xilinx 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 Xilinx based device quickly and efficiently.