Designing with Versal AI Engine 1: Architecture and Design Flow

ACAP 2

ACAP-AIE1 (v1.0)

Course Specification

Course Description

This course describes the Versal™ AI Engine architecture, how to program the AI Engines (single kernel programming and multiple kernel programming using data flow graphs), the data communications between the PL and AI Engines, and how to analyze the kernel program using various debugger features.

The emphasis of this course is on:

- Illustrating the AI Engine architecture
- Designing single AI Engine kernels using the Vitis™ unified software platform
- Designing multiple AI kernels using data flow graphs with the Vitis IDE
- Reviewing the data movement between AI Engines, between AI Engines via memory and DMA, and between AI Engines to programmable logic (PL)
- Analyzing and debugging kernel performance

Level – ACAP 2

Course Details

- 2 days live instructor led training (in person or online)
- 12 lectures
- 4 labs

Price – $1,600 or 16 Xilinx Training Credits

Course Part Number – ACAP-AIE1

Who Should Attend? – Software and hardware developers, system architects, and anyone who needs to accelerate their software applications using Xilinx devices

Prerequisites

- Comfort with the C/C++ programming language
- Software development flow
- Vitis software for application acceleration development flow

Subsequent Courses

- Designing with Versal AI Engine 2
- Designing with Versal AI Engine 3

Software Tools

- Vitis unified software platform 2020.2

Hardware

- Architecture: Xilinx Versal ACAPs

Check with Morgan Advanced Programmable Systems, Inc. for the specifics of the in-class lab board or other customizations. After completing this comprehensive training, you will have the necessary skills to:

- Describe the Versal ACAP architecture at a high level
- Describe the various engines in the Versal ACAP device and the motivation behind the AI Engine
- Describe the architecture of the AI Engine
- Describe the memory access structure for the AI Engine
- Describe the full application acceleration flow with the Vitis tool
- Enumerate the toolchain for Versal AI Engine programming
- Explain what intrinsic functions are
- Program a single AI Engine kernel using the Vitis IDE tool
- Program multiple AI Engine kernels using Adaptive Data Flow (ADF) graphs

Course Outline

Day 1

- **Overview of the Versal ACAP Architecture**
  Provides an overview of the Versal architecture at a high level and describes the various engines in the Versal ACAP, such as the Scalar Engines, Adaptable Engines, and Intelligent Engines. Also describes how the AI Engine in the Versal ACAP meets many dynamic market needs. {Lecture}
- **Introduction to the Versal AI Engine Architecture**
  Introduces the architecture of the AI Engine and describes the AI Engine interfaces that are available, including the memory, lock, core debug, cascaded stream, and AXI-Stream interfaces. {Lecture}
- **Versal AI Engine Memory and Data Movement**
  Describes the memory module architecture for the AI Engine and how memory can be accessed by the AI Engines in the AI Engine arrays. {Lecture}
- **Versal ACAP Tool Flow**
  Reviews the Vitis tool flow for the AI Engine and demonstrates the full application acceleration flow for the Vitis platform. {Lecture, Lab}
- **Application Partitioning on Versal ACAPs 1**
  Covers what application partitioning is and how an application can be accelerated by using various compute engines in the Versal ACAP. Also describes how different models of computation (sequential, concurrent, and functional) can be mapped to the Versal ACAP. {Lecture}
- **Scalar and Vector Data Types**
  Provides an AI Engine functional overview and identifies the supported vector data types and high-width registers for allowing single-instruction multiple-data (SIMD) instructions. {Lecture}
- **Intrinsic Functions**
  Describes what intrinsic functions are, the three types of vector management operations using intrinsic functions (load and store, element conversion, and lane insertion/extraction), multiplication functions, and application-specific functions. {Lecture}

Day 2

- **Window and Streaming Data APIs**
  Describes window and streaming APIs and reviews the various window operations for kernels. Also discusses using overlapping data and various data movement use cases. {Lecture}
- **Vitis Analyzer**
  Describes the different reports generated by the tool and how to view the reports that help to optimize and debug AI Engine kernels using the Vitis analyzer tool. {Lecture}
- **The Programming Model: Single Kernel**
  Reviews the AI Engine kernel programming flow for programming and building a single kernel. Also illustrates the steps to create, compile, simulate, and debug a single kernel program using the Vitis IDE tool. {Lecture, Lab}
- **The Programming Model: Single Kernel Using Vector Data Types**
  Illustrates Versal AI Engine kernel programming in detail, reviewing the scalar kernel code and comparing with vector kernel code that utilizes intrinsic functions and vector data types. {Lab}
- **The Programming Model: Introduction to the Adaptive Data Flow (ADF) Graph**

Provides the basics of the data flow graph model and graph input specifications for AI Engine programming. Also reviews graph input specifications, such as the number of platforms and ports. {Lecture}

- **The Programming Model: Multiple Kernels Using Graphs**

  Describes the ADF graph in detail and demonstrates the steps to create a graph and set the runtime ratio and graph control APIs from the main application program. {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 training with real hardware

During the Covid-19 period, some companies do not allow their staff to participate in live in-person training.

- Consequently, 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.