

## 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 utilize the AI Engine library for faster development.

The emphasis of this course is on:

- Illustrating the AI Engine architecture
- Designing single AI Engine kernels
- 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 Duration** – 2 days

**Price** – \$1800 or 18 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

### Software Tools

- Vitis unified software platform 2020.2

### Hardware

- Architecture: Xilinx Versal ACAPs

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
- Program multiple AI Engine kernels using static data flow (SDF) graphs

## Course Outline

### Day 1

- **Overview of 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 AI Engine 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**  
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}
- **Data Types: 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}
- **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. {Lecture, Lab}
- **The Programming Model: Introduction to the Data Flow 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 static data flow (SDF) graph and demonstrates the steps to create a graph and set the runtime ratio and graph control APIs from the main application program. {Lecture, Lab}



# Designing with Versal AI Engine 1

ACAP 2

ACAP-AIE1 (v1.0H)

## Course Specification

### Register Today

Hardent, a Xilinx Authorized Training Provider (ATP) delivers public and private courses. Visit <http://www.hardent.com/training> or contact Hardent's Training Coordinator for more information, to register for a class, or to schedule a private course.

Email: [training@hardent.com](mailto:training@hardent.com)  
Telephone: 514-284-5252

