AMD
together we advance_

Designing with Versal AI Engine: Quick Start

AIE-QSTART (v1.0)

Course Specification

Course Description

This course covers the AMD Versal™ AI Engine architecture and memory modules, programming the AI Engine (kernels and graphs), using the DSP Library, developing AI Engine designs using AMD Vitis™ Model Composer, and debugging the AI Engines.

The emphasis of this course is on:

- Illustrating the AI Engine architecture and memory modules
- Describing the Vitis and AI Engine tool flow
- Programming the AI Engine with kernels and graphs
- Describing the AI Engine APIs, including streaming data APIs, and I/O buffers
- Utilizing the Vitis DSP library for AI Engines
- Creating AI Engine designs using Vitis Model Composer
- Describing the debugging methodology for AI Engines

What's New for 2026.1

- AMD Versal AI Engine Tool Flow module: Added information on Vitis AI development flow
- All labs have been updated to the latest software versions

Level – VER 1

Course Details

- 1 day ILT or 8 hours On-Demand
- 7 lectures
- 5 labs

Course Part Number – AIE-QSTART

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

Prerequisites

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

Software Tools

- Vitis Unified IDE 2026.1
- Vitis Model Composer 2026.1

Hardware

- Architecture: Versal adaptive SoCs

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

- Describe the AMD Versal adaptive SoC and the architecture of the AI Engine
- Describe the toolchain for Versal AI Engine programming and the full application acceleration flow with the AMD Vitis Unified IDE
- Program the AI Engines for single and multiple kernels using graphs
- Describe the AI Engine APIs, including streaming data APIs, and I/O buffers that are used in programming
- Utilize the AI Engine DSP library and create a filter design in the Vitis tool
- Design a filter with the AMD Vitis Model Composer AI Engine library
- Describe the debug methodology available for AI Engines

Course Outline

Versal AI Engine Architecture

- **Introduction to the Versal AI Engine Architecture**
  Describes the Versal adaptive SoC at a high level. Also introduces the architecture of the AI Engine and its memory modules and interfaces. {Lecture}

Vitis Tool Flow

- **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, Labs}

The Programming Model

- **AI Engine Programming: Kernels and Graphs**
  Investigates AI Engine kernels and Adaptive Data Flow (ADF) graphs along with their programming flows. {Lecture, Lab}
- **Introduction to the APIs and Data Access Mechanisms**
  Describes the AI Engine APIs for loading and storing data as well as the I/O buffers and streaming data APIs that are used in programming data access mechanisms. {Lecture}

Libraries

- **AI Engine DSP Library Overview**
  Provides an overview of the available DSP library, which enables faster development and comes with ready-to-use example design that help with using the library and tools. {Lecture, Lab}

Vitis Model Composer

- **Vitis Model Composer for AI Engine Design**
  Describes the Vitis Model Composer tool and how to use the libraries available with the tool for AI Engine design development. {Lecture, Lab}

Debugging

- **Versal AI Engine Application Debug and Event Trace**
  Reviews the debugging methodology for AI Engine designs. {Lecture}

Register Today

Visit the AMD Adaptive Computing Customer Training Portal to view schedules and register online.

© Copyright 2026 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, Versal, Vitis, 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.