AMD
together we advance_

Designing with Versal AI Engine: DSP Applications Spec Sheet

AIE-DSP (v1.0)

Course Specification

Course Description

This course covers the AMD Versal™ AI Engine architecture and using the AI Engine DSP Library, system partitioning, rapid prototyping, and custom coding of AI Engine kernels. Developing AI Engine DSP designs using the AMD Vitis™ Model Composer is also demonstrated.

The emphasis of this course is on:

- Providing an overview of the AI Engine architecture
- Utilizing the Vitis DSP library and exploring matrix decomposition functions available in the Vitis Solver library for AI Engines
- Performing system partitioning and planning
- Adding custom kernel code for designs
- Creating AI Engine DSP designs using the Vitis Model Composer
- Analyzing reports using the Analysis view of the Vitis Unified IDE

What's New for 2026.1

- Added information on introduction to the Vitis AI Engine Solver library and its Cholesky and QR decomposition functions
- All labs have been updated to the latest software versions

Level – ACAP 2

Course Details

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

Course Part Number – AIE-DSP

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

Prerequisites

- Familiarity with the C/C++ programming language
- Vitis tool for acceleration development flow
- Familiarity with basic signal processing concepts
- Basic knowledge of Versal AI Engine architecture and programming

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 AI Engine architecture
- Utilize the AI Engine DSP library and create a filter design with the AMD Vitis Unified IDE
- Describe the Vitis AI Engine Solver library and its Cholesky and QR decomposition functions
- Follow the system partitioning and system mapping methodology
- Add custom kernel code to a design
- Design a DSP function with the Vitis Model Composer AI Engine library
- Analyze AI Engine designs using the Analysis view (Vitis Analyzer utility) of the Vitis Unified IDE

Course Outline

- **AMD Versal AI Engine Architecture**
  Introduces the architecture of the AI Engine and its components. {Lecture}
- **Introduction to the Vitis AI Engine DSP and Solver Libraries**
  Provides an overview of the Vitis AI Engine DSP library, which enables faster development and comes with ready-to-use example design that help with using the library and tools. It also introduces the Vitis AI Engine Solver library and its Cholesky and QR decomposition functions. {Lecture, Labs}
- **System Partitioning Methodology**
  Covers the system design planning and partitioning methodology for mapping design requirements to the AI Engine. {Lecture, Lab}
- **Rapid Prototyping and Custom Coding of AI Engine Kernels**
  Describes the AI Engine programming flow with kernels and Adaptive Data Flow (ADF) graphs. Also outlines the kernel coding methodology for writing custom kernel code and rapid prototyping. {Lecture, Lab}
- **Overview of AI Engine Kernel Optimization**
  Highlights the various AI Engine kernel optimization techniques, such as compiler directives, software pipelining, coding for performance, and core utilization. {Lecture}
- **Analyzing AI Engine Design Reports Using the Vitis Unified IDE**
  Covers the different reports generated by the Vitis Unified IDE and how to use these reports to optimize and debug AI Engine kernels. {Lecture}
- **AI Engine DSP Designs with Vitis Model Composer**
  Describes the Vitis Model Composer tool and how to use the libraries available with the tool for AI Engine DSP design development. {Lecture, Lab}

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.