Designing with Versal AI Engine 2: Graph Programming with AI Engine Kernels

ACAP 3

ACAP-AIE2 (v1.0)

Course Specification

Course Description

This course describes the system design flow and interfaces that can be used for data movements in the Versal™ AI Engine. It also demonstrates how to utilize the advanced MAC intrinsics, AI Engine library for faster development and advanced features in adaptive data flow (ADF) graph implementation, such as using streams, cascade stream, buffer location constraints, run-time parameterization and APIs to update and/read run-time parameters.

The emphasis of this course is on:

- Implementing a system-level design flow (PS + PL + AIE) and the supported simulation
- Using an interface for data movement between the PL and AI Engine
- Utilizing advanced MAC intrinsics to implement filters
- Utilizing the AI Engine library for faster development
- Applying advanced features for optimizing a system-level design

What's New for 2021.1

- ACAP Data Communications modules:
  - Deprecated support for PL kernels in the AI Engine graph
- Versal AI Engine DSP Library Overview
  - AI Engine DSP library is now part of Vitis Libraries
  - DSP library now available from GitHub
- All labs have been updated to the latest software versions

Level – ACAP 3

Course Details

- 2 days ILT or 16 hours OnDemand
- 10 lectures
- 5 labs

Course Part Number – ACAP-AIE2

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 2021.1

Hardware

- Architecture: Xilinx Versal ACAPs

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

- Describe the system-level flow, which includes PS + PL + AIE (SW-HW-SW) designs
- Describe the supported emulation for a system-level design
- Describe the data movement between the PS, PL, and AI Engines
- Describe the implementation of the AI Engine and programmable logic
- Implement a system-level design for Versal ACAPs with the Vitis tool flow
- Utilize advanced MAC intrinsic syntax and application-specific intrinsics such as DDS and FFT
- Utilize the AI Engine DSP library for faster development
- Apply location constraints on kernels and buffers in the AI Engine array
- Apply runtime parameters to modify application behavior
- Debug a system-level design

Course Outline

Day 1

Design Analysis

- **Application Partitioning on Versal ACAPs 1 (Review)**

  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}

- **Application Partitioning on Versal ACAPs 2**

  Explains how image and video processing can be targeted for the Versal ACAP by utilizing the different engines (Scalar Engine, Adaptable Engine, and Intelligent Engine). Also describes the AI engine development flow. {Lecture}

Versal AI Engine Data Movement

- **ACAP Data Communications 1**

  Describes the implementation of AI Engine and programmable logic (PL) kernels and how to implement the functions in the AI Engine that take advantage of low power. {Lecture}

- **ACAP Data Communications 2**

  Describes the programming model for the implementation of stream interfaces for the AI Engine kernels and PL kernels. Lists the stream data types that are supported by AI Engine and PL kernels. {Lecture, Lab}

Vitis Tool Flow

- **System Design Flow**

  Demonstrates the Vitis compiler flow to integrate a compiled AI Engine design graph (libadf.a) with additional kernels implemented in the PL region of the device (including HLS and RTL kernels) and link them for use on a target platform. You can call then these compiled hardware functions from a host program running in the Arm® processor in the Versal device or on an external x86 processor. {Lecture, Lab}

The Programming Model

- **Introduction to Advanced Intrinsic Functions**

  Describes how to implement filters using advanced intrinsics functions for various filters, such as non-symmetric FIR, symmetric FIR, or half-band decimators. {Lecture}

Day 2

Libraries

- **Versal AI Engine DSP Library Overview**

  Provides an overview of the available DSP library, which enables faster development and comes with ready-to-use example designs that help with using the library and tools. {Lecture, Labs}

The Programming Model

- **Advanced Graph Input Specifications 1**

  Learn advanced features such as using initialization functions, writing directly using streams from the AI Engine, cascade

stream, core location constraints, and buffer location constraints. {Lecture}

- **Advanced Graph Input Specifications 2**

  Describes how to implement runtime parameterization, which can be used as adaptive feedback and to switch functionality dynamically. {Lecture, Lab}

Debugging

- **Versal AI Engine Application Debug and Trace**

  Shows to how to debug the AI Engine application running on the Linux OS and how to debug via hardware emulation that allows simulation of the application. {Lecture}

Register Today

Visit the Xilinx Customer Training Center to view schedules and register online