XILINX

Accelerating Applications with the Vitis Unified Software Environment

AI 3

AI-ACCEL (v1.0)

Course Specification

Course Description

Learn how to develop, debug, and profile new or existing C/C++ and RTL applications in the Vitis™ unified software environment targeting both data center (DC) and embedded applications. Also learn how to run designs on the Xilinx Alveo™ accelerator card using Nimbix Cloud.

The emphasis of this course is on:

- Building a software application using the OpenCL™ API to run hardware kernels on Alveo accelerator cards
- Building a software application using the OpenCL API and the Linux-based Xilinx runtime (XRT) to schedule the hardware kernels and control data movement on an embedded processor platform
- Demonstrating the Vitis environment GUI flow and makefile flow for both DC and embedded applications
- Describing the Vitis platform execution model and XRT
- Describing kernel development using C/C++ and RTL
- Utilizing the Vitis analyzer tool to analyze reports
- Explaining the design methodology to optimize a design

Level – AI 3

Course Duration – 2 days

Price – $1,600 or 16 Xilinx Training Credits

Course Part Number – AI-ACCEL

Who Should Attend? – Anyone who needs to accelerate their software applications using FPGAs, SoCs (such as Zynq®-7000 SoCs, Zynq UltraScale+™ MPSoCs), and Versal™ ACAPs

Prerequisites

- Basic knowledge of Xilinx FPGA architecture
- Comfort with the C/C++ programming language
- Software development flow

Software Tools

- Vitis unified software environment 2019.2

Hardware

- Architecture: Xilinx Alveo accelerator cards, SoCs, and ACAPs

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

- Describe how the FPGA architecture lends itself to parallel computing
- Explain how the Vitis unified software environment helps software developers to focus on applications
- Describe the Vitis (OpenCL API) execution model
- Analyze the OpenCL API memory model
- Create kernels from C, C++, or RTL IP using the RTL Kernel Wizard
- Apply host code optimization and kernel optimization techniques
- Move data efficiently between kernel and global memory
- Profile the design using the Vitis analyzer tool

Course Outline

Day 1

Vitis Tool Flow

- **Introduction to the Vitis Unified Software Platform**

  Explains how software/hardware engineers and application developers can benefit from the Vitis unified software environment and OpenCL framework. {Lecture}
- **Vitis IDE Tool Overview**

  Describes the elements of the development flow, such as software emulation, hardware emulation, and system run as well as debugging support for the host code and kernel code. {Lecture, Lab}
- **Vitis Command Line Flow**

  Introduces the Vitis environment makefile flow where the user manages the compilation of host code and kernels. {Lecture, Lab}

Basics of Hardware Acceleration

- **Introduction to Hardware Acceleration**

  Outlines the fundamental aspects of FPGAs, SoCs, and ACAPs that are required to guide the Vitis tool to the best computational architecture for any algorithm. {Lecture}

Alveo Data Center Accelerator Cards

- **Alveo Data Center Accelerator Cards Overview**

  Describes the Alveo Data Center accelerator cards and lists the advantages of these cards and the available software solutions stack. {Lecture}
- **Alveo Accelerator Card Ecosystem Partner Solutions Overview**

  Outlines the partner solutions available in the cloud and on premises for Alveo Data Center accelerator cards. {Lecture}
- **Getting Started with Alveo Data Center Accelerator Cards**

  Describes the hardware and software installation procedures for the Alveo Data Center accelerator cards. {Lecture}
- **Introduction to the Nimbix Cloud**

  Describes the Nimbix Cloud, the availability of the Alveo Data Center accelerator cards in the Nimbix Cloud, and how to run a design on the Nimbix Cloud. {Lecture}

Vitis Execution Model and XRT

- **Vitis Execution Model and XRT**

  Describes the XRT and the OpenCL APIs used for such as setting up the platform, executing the target device and post-processing. {Lecture, Lab}
- **Synchronization**

  Describes OpenCL synchronization techniques such as events, barriers, blocking write/read, and the benefit of using out-of-order execution. {Lecture, Lab}

Day 2

NDRange (Optional)

- **Introduction to NDRanges**

  Explains the basics of NDRange (N dimensional range) and the OpenCL execution model that defines how kernels execute with the NDRange definition. {Lecture}
- **Working with NDRanges**

  Outlines the host code and kernel code changes with respect to NDRange. Also explains how NDRange works and the best way to represent the work-group size for the FPGA architecture. {Lecture}

Design Analysis

- **Profiling**

  Describes the different reports generated by the tool and how to view the reports that help to optimize data transfer and kernel optimization using the Vitis analyzer tool. {Lecture}
- **Debugging**

  Explains the support for debugging host code and kernel code as well as tips to debug the system. {Lecture}

Kernel Development

- **Introduction to C/C++ based Kernels**

  Describes the trade-offs between C/C++, OpenCL, and RTL applications and the benefits of C-based kernels. {Lecture, Lab}
- **Using the RTL Kernel Wizard to Reuse Existing IP as Accelerators**

  Describes how the Vitis unified software development provides RTL kernel developers with a framework to integrate their hardware functions into an application running on a host PC connected to an FPGA via a PCIe® interface. {Lecture, Lab}

Optimization Methodology Guide

- **Optimization Methodology**

  Describes the recommended flow for optimizing an application in the Vitis unified software development environment. {Lecture}
- **C/C++ based Kernel Optimization**

  Reviews different techniques such as loop unrolling, pipelining, and DATAFLOW. {Lecture}
- **Host Code Optimization**

  Describes the various optimization techniques such as reducing the overhead of kernel enqueing, and optimizing the data transfer between kernels and global memory. {Lecture}
- **Optimizing the Performance of the Design**

  Describes the various optimization techniques such as optimizing the host code, data transfer between kernels and global memory and the kernel performance. {Lab}

Libraries

- **Vitis Accelerated Libraries**

  Reviews available libraries such as BLAS, Fintech, and OpenCV.

  The xfOpenCV library is a set of 60+ kernels, optimized for Xilinx FPGAs and SoCs, based on the OpenCV computer vision library. {Lecture}

Register Today

Morgan Advanced Programmable Systems, Inc. (MAPS, Inc.) 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

- Students 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.

MAPS Inc. 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.
- MAPS 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 MAPS 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.