AMD
together we advance_

High-Level Synthesis with the Vitis Unified IDE

DSP-HLS (v1.0)

Course Specification

Course Description

This course provides a thorough introduction to high-level synthesis (HLS) using the AMD Vitis™ Unified IDE.

The focus of this course is on:

- Converting C/C++ designs into RTL implementations
- Learning the HLS component development flow
- Creating I/O interfaces for designs
- Applying different optimization techniques to designs
- Improving throughput, area, latency, and logic by using different HLS pragmas/directives
- Exporting IP that can be used with the Vivado™ IP catalog
- Migrating designs from the classic Vitis HLS tool to the Vitis Unified IDE

What's New for 2024.2

- Vitis HLS Libraries module: Added information on new featured libraries: Direct I/O, Fence, and DSP Intrinsics
- All labs have been updated to the latest software versions

Level – DSP 3

Course Details

- 2 days ILT/ 3 sessions

Course Part Number – DSP-HLS

Who Should Attend? – Software and hardware engineers looking to utilize high-level synthesis

Prerequisites

- C or C++ knowledge
- Basic RTL design flow knowledge

Software Tools

- Vitis Unified IDE 2024.2
- Vivado Design Suite 2024.2
- Vitis HLS tool 2024.2

Hardware

- Architecture: Zynq™ UltraScale+™ MPSoC and Versal™ AI Core series
- Demo board: Zynq UltraScale+ MPSoC ZCU104 board*

* Check with your local Authorized Training Provider for the specifics of the in-class lab board or other customizations.

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

- Enhance productivity by using the AMD Vitis Unified IDE for HLS component development
- Describe the HLS component development flow
- Use the Vitis Unified IDE to create an HLS component
- Identify common coding pitfalls as well as methods for improving RTL/hardware code
- Use directives/pragmas to improve throughput, area, latency, and logic and to select RTL interfaces
- Perform system-level integration of IP generated by the Vitis Unified IDE
- Migrate designs from the classic Vitis HLS tool to the Vitis Unified IDE

Course Outline

Day 1

- **Introduction to High-Level Synthesis**
  Provides an overview of high-level synthesis (HLS), the Vitis Unified IDE for HLS flow, and the verification advantage. {Lecture}
- **HLS Component Development Flow**
  Explores the HLS component development flow in the Vitis Unified IDE. {Lecture, Lab}
- **Abstract Parallel Programming Model for HLS**
  Describes the structuring of a design at a high level using an abstract parallel programming model. {Lecture}
- **Design Exploration with Directives**
  Explores different optimization techniques that can improve design performance. {Lecture}
- **HLS Component Development Using the Command Line**
  Describes the unified command line interface and the the `v++` and `vitis-run` commands. {Lecture, Lab}
- **Introduction to Vitis HLS Design Methodology**
  Introduces the methodology guidelines covered in this course and the HLS Design Methodology steps. {Lecture}
- **Introduction to I/O Interfaces**
  Explains interfaces such as the block-level and port-level protocols abstracted by Vitis HLS from a C design. {Lecture}
- **Block-Level Protocols**
  Explains the different types of block-level protocols abstracted by Vitis HLS. {Lecture, Lab}
- **Port-Level I/O Protocols**
  Describes the port-level interface protocols abstracted by Vitis HLS from a C design. {Lecture, Labs}
- **AXI Adapter Interface Protocols**
  Explains the different AXI interfaces (such as AXI4-Master, AXI4-Lite (Slave), and AXI4-Stream) supported by Vitis HLS. {Lecture}
- **Vitis HLS Code Analyzer**
  Provides an overview of the Vitis Code Analyzer, its features, and how to view generated reports. {Lecture, Lab}

Day 2

- **Optimizing for Performance: PIPELINE**
  Describes the PIPELINE directive for improving the throughput of a design. {Lecture, Lab}
- **Optimizing for Performance: DATAFLOW**
  Describes the DATAFLOW directive for improving the throughput of a design by pipelining the functions to execute as soon as possible. {Lecture, Lab}
- **Optimizing for Throughput**
  Describes the performance limitations caused by arrays in a design. Also explores optimization techniques to handle arrays for improving performance. {Lecture, Lab}
- **Optimizing for Latency: Default Behavior**
  Describes the default behavior of Vitis HLS on latency and throughput. {Lecture}
- **Optimizing for Latency: Reducing Latency**
  Describes how to optimize the C design to improve latency. {Lecture}
- **Optimizing for Area and Logic**
  Describes different methods for improving resource utilization and explains how some of the directives have impact on the area utilization. {Lecture, Lab}

- **Optimizing AXI System Performance**
  Describes AXI burst transfers and their types. Also outlines the optimization steps to improve system performance. {Lecture}
- **Vitis HLS Libraries**
  Describes the library support offered by Vitis HLS. {Lecture}
- **Vitis HLS Libraries: Arbitrary Precision Data Types**
  Describes Vitis HLS support for the C/C++ languages as well as arbitrary precision data types. {Lecture, Lab}
- **Using Pointers in Vitis HLS**
  Explains the use of pointers in a design and workarounds for some of the limitations. {Lecture}
- **HLS Component Design Flow – System Integration**
  Illustrates the process of developing and exporting an HLS component as Vivado IP. {Lab}
- **Migrating to the Vitis Unified IDE – HLS Component**
  Describes the need for the Vitis Unified IDE and identifies different approaches for migrating projects from the classic Vitis HLS tool to the Vitis Unified IDE. {Lecture, Lab}