High-Level Synthesis with the Vitis HLS Tool

DSP-HLS (v1.0)

Course Specification

Course Description

This course provides a thorough introduction to the Vitis™ High-Level Synthesis (HLS) tool.

The focus of this course is on:

- Converting C/C++ designs into RTL implementations
- Learning the Vitis HLS tool flow
- Creating I/O interfaces for designs by using the Vitis HLS tool
- Applying different optimization techniques
- Improving throughput, area, latency, and logic by using different HLS pragmas/directives
- Exporting IP that can be used with the Vivado® IP catalog
- Downloading for in-circuit validation

What's New for 2022.1

- Optimizing for Throughput module: Added details on automatic array partitioning and using performance constraints to simplify loop-level pragma insertion
- All labs have been updated to the latest software versions

Level – DSP 3

Course Details

- 2 days ILT
  - 21 lectures
  - 11 labs
  - 4 ILT demos

Price –

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 HLS tool 2022.1
- Vivado Design Suite 2022.1
- Vitis unified software platform 2022.1

Hardware

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

* This course focuses on the Zynq UltraScale+ MPSoC architecture. 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 using the Vitis HLS tool
- Describe the high-level synthesis flow
- Use the Vitis HLS tool for a first project
- Identify the importance of the test bench
- Use directives to improve performance and area and select RTL interfaces
- Identify common coding pitfalls as well as methods for improving code for RTL/hardware
- Perform system-level integration of IP generated by the Vitis HLS tool

Course Outline

Day 1

- **Introduction to High-Level Synthesis**
  Overview of high-level synthesis (HLS), the Vitis HLS tool flow, and the verification advantage. {Lecture}
- **Vitis HLS Tool Flow**
  Explores the basics of high-level synthesis and the Vitis HLS tool. {Lecture, Demo, Lab}
- **Design Exploration with Directives**
  Explores different optimization techniques that can improve the design performance. {Lecture}
- **Vitis HLS Tool Command Line Interface**
  Describes the Vitis HLS tool flow in command prompt mode. {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 the Vitis HLS tool from the C design. {Lecture}
- **Block-Level Protocols**
  Explains the different types of block-level protocols abstracted by the Vitis HLS tool. {Lecture, Lab}
- **Port-Level I/O Protocols**
  Describes the port-level interface protocols abstracted by the Vitis HLS tool from the C design. {Lecture, Demo, Lab}
- **AXI Adapter Interface Protocols**
  Explains the different AXI interfaces (such as AXI4-Master, AXI4-Lite (Slave), and AXI4-Stream) supported by the Vitis HLS tool. {Lecture, Demo}
- **Port-Level I/O Protocols: Memory Interfaces**
  Describes the memory interface port-level protocols (such as block RAM and FIFO) abstracted by the Vitis HLS tool from the C design. {Lecture, Lab}
- **Pipeline for Performance: PIPELINE**
  Describes the PIPELINE directive for improving the throughput of a design. {Lecture, Lab}

Day 2

- **Pipeline 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**
  Identify the performance limitations caused by arrays in your design. You will also explore optimization techniques to handle arrays for improving performance. {Lecture, Demo, Lab}
- **Optimizing for Latency: Default Behavior**
  Describes the default behavior of the Vitis HLS tool 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}

- **Migrating to the Vitis HLS Tool**
  Reviews key considerations when moving from the Vivado HLS tool to the Vitis HLS tool. {Lecture}
- **HLS Design Flow – System Integration**
  Describes the traditional RTL flow versus the Vitis HLS tool design flow. {Lecture, Lab}
- **Vitis HLS Tool C++ Libraries: Arbitrary Precision**
  Describes Vitis HLS tool support for the C/C++ languages as well as arbitrary precision data types. {Lecture, Lab}
- **Hardware Modeling**
  Describes hardware modeling with streaming data types and shift register implementation using the ap_shift_reg class. {Lecture}
- **Using Pointers in the Vitis HLS Tool**
  Explains the use of pointers in the design and workarounds for some of the limitations. {Lecture}