C-based Design: High-Level Synthesis with the Vivado HLx Tool

DSP 3

DSP-HLS (v1.0)

Course Specification

Course Description

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

The focus is on:

- Covering synthesis strategies and features
- Improving throughput, area, interface creation, latency, testbench coding, and coding tips
- Utilizing the Vivado HLS tool to optimize code for high-speed performance in an embedded environment
- Downloading for in-circuit validation

What's New for 2020.1

- Introduction to High-Level Synthesis: New content on the Vitis™ HLS tool
- Introduction to HLS UltraFast Design Methodology: New content on RTL verification, IP packaging, and design analysis
- All labs have been updated to support the Zynq® UltraScale+™ ZCU104 board

Level – DSP 3

Course Details

- 2 days ILT or 24 hours OnDemand
  - 23 lectures
  - 11 labs
  - 5 ILT demos

Price –

Course Part Number – DSP-HLS

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

Prerequisites

- C, C++, or SystemC knowledge
- High-level synthesis for software engineers OR high-level synthesis for hardware engineers

Software Tools

- Vivado HLS System Edition 2020.1
- Vitis unified software platform 2020.1

Hardware

- Architecture: Zynq UltraScale+ MPSoC
- 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 Vivado HLS tool
- Describe the high-level synthesis flow
- Use the Vivado 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 Vivado HLS tool
- Describe how to use OpenCV functions in the Vivado HLS tool

Course Outline

Day 1

- **Introduction to High-Level Synthesis**
  Overview of the High-level Synthesis (HLS), Vivado HLS tool flow, and the verification advantage. {Lecture}
- **Vivado HLS Tool Flow**
  Explore the basics of high-level synthesis and the Vivado HLS tool. {Lecture, Demo, Lab}
- **Design Exploration with Directives**
  Explore different optimization techniques that can improve the design performance. {Lecture}
- **Vivado HLS Tool Command Line Interface**
  Describes the Vivado HLS tool flow in command prompt mode. {Lecture, Lab}
- **Introduction to HLS UltraFast Design Methodology**
  Introduces the methodology guidelines covered in this course and the HLS UltraFast Design Methodology steps. {Lecture}
- **Introduction to I/O Interfaces**
  Explains interfaces such as block-level and port-level protocols abstracted by the Vivado HLS tool from the C design. {Lecture}
- **Block-Level I/O Protocols**
  Explains the different types of block-level protocols abstracted by the Vivado HLS tool. {Lecture, Lab}
- **Port-Level I/O Protocols**
  Describes the port-level interface protocols abstracted by the Vivado HLS tool from the C design. {Lecture, Demo, Lab}
- **Port-Level I/O Protocols: AXI4 Interfaces**
  Explains the different AXI interfaces (such as AXI4-Master, AXI4-Lite (Slave), and AXI4-Stream) supported by the Vivado HLS tool. {Lecture, Demo}
- **Port-Level I/O Protocols: Memory Interfaces**
  Describes the memory interface port-level protocols (such as block RAM, FIFO) abstracted by the Vivado HLS tool from the C design. {Lecture, Lab}
- **Port-Level I/O Protocols: Bus Protocol**
  Explains the bus protocol supported by the Vivado HLS tool. {Lecture}
- **Pipeline for Performance: PIPELINE**
  Describes the PIPELINE directive for improving the throughput of a design. {Lecture, Demo, 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 Structures for Performance**
  Learn the performance limitations caused by arrays in your design. You will also learn some optimization techniques to handle arrays for improving performance. {Lecture, Demo, Lab}
- **Data Pack and Data Dependencies**
  Learn how to use DATA_PACK and DEPENDENCE directives to overcome the limitations caused by structures and loops in the design. {Lecture}
- **Vivado HLS Tool Default Behavior: Latency**
  Describes the default behavior of the Vivado HLS tool on latency and throughput. {Lecture}

- **Reducing Latency**
  Describes how to optimize the C design to improve latency. {Lecture}
- **Improving Area and Resource Utilization**
  Describes different methods for improving resource utilization and explains how some of the directives have impact on the area utilization. {Lecture, Lab}
- **HLx Design Flow – System Integration**
  Describes the traditional RTL flow versus the Vivado HLx design flow. {Lecture, Lab}
- **Vivado HLS Tool C Libraries: Arbitrary Precision**
  Describes the Vivado HLS tool support for the C/C++ languages, as well as arbitrary precision data types. {Lecture, Lab}
- **Hardware Modeling**
  Explains hardware modeling with streaming data types and shift register implementation using the ap_shift_reg class. {Lecture}
- **Using Pointers in the Vivado HLS Tool**
  Explains the use of pointers in the design and workarounds for some of the limitations. {Lecture}

Register Today

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