Xilinx HLS (High-level synthesis) and SDSoC (Software-defined SoC)

DSP 3 Embedded 1 and Embedded 2

EMBD-HLS-SDSOC-ILT (v1.0)

Course Specification

Course Description

This is a combined course. The first part of this course provides a thorough introduction to the Vivado® High-Level Synthesis (HLS) tool. It covers synthesis strategies, features, improving throughput, area, interface creation, latency, testbench coding, and coding tips. Use the Vivado HLS tool to optimize code for high-speed performance in an embedded environment and download for in-circuit validation.

This second part of this course provides a thorough introduction to the Vivado® High-Level Synthesis (HLS) tool. It covers synthesis strategies, features, improving throughput, area, interface creation, latency, testbench coding, and coding tips. Use the Vivado HLS tool to optimize code for high-speed performance in an embedded environment and download for in-circuit validation.

Level – DSP 3 Embedded 1 and Embedded 2

Course Duration – 3 days

Price – $2400 or 24 Xilinx Training Credits

Course Part Number – EMBD-HLS-SDSOC-ILT

Who Should Attend? – Software and hardware engineers looking to understand the high-level synthesis technology behind SDSoC, which allows you to rapidly add acceleration to a software system.

Prerequisites

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

Recommend

- Designing FPGAs Using the Vivado Design Suite 1

Alternative training

- Embedded C/C++ SDSoC Development Environment and Methodology
- C-based Design: High-Level Synthesis with the Vivado HLx Tool

Software Tools

- Vivado System Edition 2017.1
- SDx™ development environment 2016.3
- MATLAB R2017a

Hardware

- Architecture: Zynq®-7000 All Programmable SoC and 7 series FPGAs*
- Demo board: Zynq-7000 All Programmable SoC ZC702 or Zed board and Kintex®-7 FPGA KC705 board*

* This course focuses on the Zynq-7000 All Programmable SoC and 7 series FPGA architectures. Check with North Pole Engineering, Inc. 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 testbench
- 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
- Identify candidate functions for hardware acceleration by using the TCF profiling tool
- Use the System Debugger's capabilities to control the execution flow and examine memory and variables during a debug session
- Move designated software functions to hardware and estimate the performance of the accelerator and the effect on the entire system
- Override tool defaults to improve the performance of the individual accelerators and the overall system

Course Outline

Day 1

- Introduction to High-Level Synthesis {Lecture}
- Basics of the Vivado HLS Tool {Lecture, Demo, Lab}
- Design Exploration with Directives {Lecture}
- Vivado HLS Tool Command Line Interface {Lecture, Lab}
- Introduction to HLS UltraFast Design Methodology {Lecture}
- Introduction to I/O Interfaces {Lecture}
- Block-Level Protocols {Lecture, Lab}
- Port-Level Protocols {Lecture, Demo, Lab}
- Port-Level Protocols: AXI4 Interfaces {Lecture, Demo}
- Port-Level Protocols: Memory Interfaces {Lecture, Lab}
- Port-Level Protocols: Bus Protocol {Lecture}
- Pipeline for Performance: PIPELINE {Lecture, Demo, Lab}

Day 2

- Pipeline for Performance: DATAFLOW {Lecture, Lab}
- Optimizing Structures for Performance {Lecture, Demo, Lab}
- Data Pack and Data Dependencies {Lecture}
- Vivado HLS Tool Default Behavior - Latency {Lecture}
- Reduce Latency {Lecture}
- Improving Area {Lecture, Lab}
- Introduction to HLx Design Flow {Lecture, Demo, Lab}
- HLS vs. SDSoC Development Environment Flow {Lecture, Demo}
- Vivado HLS Tool: C Code {Lecture, Lab}
- Hardware Modeling {Lecture}
- OpenCV Libraries {Lecture}
- Pointers {Lecture}

Day 3

- Zynq AP SoC Architecture Support for Accelerators [Optional]
- Software Overview [Optional]
- SDSoC Tool Overview {Lecture, Two Demos, Lab}
- SDSoC Tool Design Best Practices {Lecture, Demo}
- Application Profiling {Lecture, Demo, Lab}
- Application Debugging {Lecture, Demo, Lab}
- Understanding Estimations in the SDSoC Tool {Lecture, Demo, Lab}
- Blocking and Non-Blocking Implementations in the SDSoC Tool {Lecture, Lab}
- Implementing Multiple Accelerators in the SDSoC Tool {Lecture, Two Labs}
- SDSoC Platform Creation {Lecture, Lab}
- Hardware/Software Event Tracing {Lecture, Lab}

Topic Descriptions

Day 1

- Introduction to High-Level Synthesis – Overview of the High-level Synthesis (HLS), Vivado HLS tool flow, and the verification advantage.
- Basics of the Vivado HLS Tool – Explore the basics of high-level synthesis and the Vivado HLS tool.
- Design Exploration with Directives – Explore different optimization techniques that can improve the design performance.

- Vivado HLS Tool Command Line Interface – Describes the Vivado HLS tool flow in command prompt mode.
- Introduction to HLS UltraFast Design Methodology – Introduces the methodology guidelines covered in this course and the HLS UltraFast Design Methodology steps.
- 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.
- Block-Level Protocols – Explains the different types of block-level protocols abstracted by the Vivado HLS tool.
- Port-Level Protocols – Describes the port-level interface protocols abstracted by the Vivado HLS tool from the C design.
- Port-Level Protocols: AXI4 Interfaces – Explains the different AXI interfaces (such as AXI4-Master, AXI4-Lite (Slave) and AXI4-Stream) supported by the Vivado HLS tool.
- Port-Level Protocols: Memory Interfaces – Describes the Memory Interface port-level protocols (such as BRAM, FIFO) abstracted by the Vivado HLS tool from the C design.
- Port-Level Protocols: Bus Protocol – Explains the bus protocol supported by the Vivado HLS tool.
- Pipeline for Performance: PIPELINE – Describes the PIPELINE directive for improving the throughput of a design.

Day 2

- Pipeline for Performance: DATAFLOW – Describes the DATAFLOW directive for improving the throughput of a design by pipelining the functions to executes as soon as possible.
- 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.
- 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.
- Vivado HLS Tool Default Behavior: Latency – Describes the default behavior of the Vivado HLS tool on latency and throughput.
- Reduce Latency – Describes how to optimize the C design to improve latency.
- Improving Area – Describes different methods for improving resource utilization and explains how some of the directives have impact on the area utilization.
- Introduction to HLx Design Flow – Describes the traditional RTL flow versus the Vivado HLx design flow.
- HLS vs. SDSoC Development Environment Flow – Describes the HLS flow versus the SDSoC™ development environment flow.
- Vivado HLS Tool: C Code – Describes the Vivado HLS tool support for the C/C++ languages, as well as arbitrary precision data types.
- Hardware Modeling – Explains hardware modeling with streaming data types and shift register implementation using the ap_shift_reg class.
- OpenCV Libraries – Explains the OpenCV design flow and the Vivado HLS tool support.
- Pointers – Explains the use of pointers in the design and workarounds for some of the limitations.

Day 3

- Zynq AP SoC Architecture Support for Accelerators [Optional] – Discusses the relevant aspects of the Zynq All Programmable SoC architecture for accelerator design. The focus is on AXI ports and protocols, system latency, and memory utilization.
- Software Overview [Optional] – Provides a thorough understanding of how the integrated design environment works, including how the compiler and linker behave, basics of makefiles, DMA usage, and variable scope.
- SDSoC Tool Overview {Lecture, Two Demos, Lab} – Introduces the purpose, underlying structures, and basic functionality of the SDSoC development environment through a combination of lecture and demonstration. Student will cement their knowledge with a lab that reinforces the concepts provided in the lecture and demo.
- SDSoC Design Best Practices {Lecture, Demo} – Illustrates common mistakes and how to avoid them. Also describes approaches to refactoring software for hardware acceleration.
- Application Profiling {Lecture, Demo, Lab} – Profiling is the process that identifies how the processor is spending its time. Through profiling, the user can quickly identify which functions must be optimized or moved to hardware to satisfy the performance requirements.
- Application Debugging {Lecture, Demo, Lab} – Through the use of the System Debugger, students will learn how to follow the control flow in an executing application and see the effects of the code on memory to successfully debug software issues.
- Understanding Estimations in the SDSoC Tool {Lecture, Demo, Lab} – Once a function is moved to hardware, questions remain: Will the accelerator fit in hardware? Will it fun fast enough? Estimations can provide the answers.
- Blocking and Non-Blocking Implementations in the SDSoC Tool {Lecture, Lab} – Addresses how the processor behaves while the accelerator is producing solutions—does it wait or continue on?
- Implementing Multiple Accelerators in the SDSoC Tool {Lecture, Lab} – There are times when moving a single function to hardware is not enough—multiple functions must be moved to hardware, or one accelerator must be duplicated. Here students will learn to control how the tool produces the accelerators.
- SDSoC Platform Creation {Lecture, Lab} – Describes how to create a custom SDSoC platform starting from a hardware system built using the Vivado Design Suite, and a software run-time environment, including operating system kernel, boot loaders, file system, and libraries.
- Hardware/Software Event Tracing {Lecture, Lab} – Hardware/software event trace helps the user to understand the performance of their application given the workload, hardware/software partitioning, and system design choices. Such information helps the user to optimize and improve system implementation.

Register Today

NPE, Inc. delivers public and private courses in locations throughout the central US region; including Iowa, Illinois, Kansas, Minnesota, Missouri, North Dakota, South Dakota and Wisconsin.

Visit **www.npe-inc.com/training**, for full course schedule and training information.

NORTH POLE npe ENGINEERING

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.

NPE 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.
- NPE 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 NPE 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.