SDSoC Development Environment and Methodology, Full

Embedded 2 and 3

EMBD-SDSOC-ILT (v1.0)

Course Specification

Course Description

This three-day course is structured to help designers new to the SDSoC™ development environment to quickly understand the full "end-user" tool flow to create accelerated systems. The focus is on utilizing the tools to accelerate an existing design at the system architecture level, not on the optimization of the accelerator microarchitectures.

Several optional modules are provided to quickly provide students with the necessary background on both hardware and software.

Level – Embedded 2 and 3

Course Duration – 3 Days

Price – $2400 or 24 Xilinx Training Credits

Course Part Number – EMBD-SDSOC-ILT

Who Should Attend? – Anyone interested in quickly adding hardware acceleration to a software system.

Prerequisites

- Zynq System Architecture or Zynq Master Training for Experienced FPGA Engineers
- Experience or training on C programming language

Highly Recommend

- C-based Design: High-Level Synthesis with the Vivado HLx Tool
- Embedded Systems Software Development

Software Tools

- SDx™ development environment 2017.1

Hardware

- Architecture: Zynq-7000 All Programmable SoC*
- Demo board: Zynq-7000 All Programmable SoC ZC702 or ZedBoard*

* This course focuses on the Zynq-7000 All Programmable SoC. 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:

- 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
- Use the hardware/software event trace to understand the performance of an application given the workload, hardware/software partitioning, and system design choices
- Improve the memory accesses and data transfer rate between the PS and PL (macro-architecture optimization)
- Apply HLS directives to enhance the performance of hardware functions (micro-architecture optimization)
- Create a C-callable library for IP blocks written in a hardware description language like VHDL or Verilog
- Override tool defaults to improve the performance of individual accelerators and the overall system
- Create a custom platform using the SDSoC Platform Utility (sdspfm)
- Describe how the reVISION Stack enables users to quickly develop applications based on machine learning and computer vision with the SDx development environment

Course Outline

Day 1

- Zynq AP SoC Architecture Support for Accelerators [Optional]
- Software Overview [Optional]
- Introduction to the SDSoC Tool {Lecture}
- SDSoC Tool Flow {Lecture, Demo, Lab}
- Application Debugging {Lecture, Demo, Lab}
- Application Profiling {Lecture, Demo, Lab}
- Understanding Estimations in the SDSoC Tool {Lecture, Demo, Lab}
- QEMU Emulation {Lab}
- Hardware/Software Event Tracing {Lecture, Lab}

Day 2

- SDSoC Environment Optimization {Lecture}
- Memory Access Optimization {Lecture, Lab}
- Blocking and Non-Blocking Implementations in the SDSoC Tool {Lecture, Lab}
- Implementing Multiple Accelerators in the SDSoC Tool {Lecture, Lab}
- Basics of the Vivado HLS Tool {Lecture}
- Design Exploration with Directives (Pragmas) {Lecture}
- Pipeline for Performance: PIPELINE {Lecture, Demo, Lab}
- Pipeline for Performance: DATAFLOW {Lecture, Lab)

Day 3

- Optimizing Structures for Performance {Lecture, Demo, Lab}
- C-Callable IP Library {Lecture, Lab}
- SDSoC Platform Creation {Lecture, Lab}
- Optimizing the Design {Lecture, Lab}
- reVISION Stack {Lecture}

Topic Descriptions

- 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.
- Introduction to the SDSoC Tool {Lecture} – Introduces the purpose, underlying structures, and basic functionality of the SDSoC development environment.
- SDSoC Tool Flow {Lecture, Demo, Lab} – Explains the complete development flow of the SDSoC integrated development environment (IDE).
- 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.
- 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.

SDSoC Development Environment and Methodology, Full

Embedded 2 and 3

EMBD-SDSOC-ILT (v1.0)

Course Specification

- 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.
- QEMU Emulation {Lab} – Describes how to use the emulation feature in the SDx IDE.
- Hardware/Software Event Tracing {Lecture, Lab} – Hardware/software event tracing helps users 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.
- SDSoC Environment Optimization {Lecture} – Describes different optimization techniques in the SDSoC development environment, such as macro-architecture and micro-architecture optimizations.
- Memory Access Optimization {Lecture, Lab} – Describes how to improve the memory access and the data transfer rate between the PS and PL.
- 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.
- Basics of the Vivado HLS Tool {Lecture} – Explore the basics of high-level synthesis and the Vivado HLS tool.
- Design Exploration with Directives (Pragmas) {Lecture} – Explore different optimization techniques that can improve the design performance.
- Pipeline for Performance: PIPELINE {Lecture, Demo, Lab) – Describes the PIPELINE directive for improving the throughput of a design.
- Pipeline for Performance: DATAFLOW {Lecture, Lab) – Describes the DATAFLOW directive for improving the throughput of a design by pipelining the functions to execute as soon as possible.
- Optimizing Structures for Performance {Lecture, Demo, Lab} – Learn the performance limitations caused by arrays in your design. You will also learn some optimization techniques to handle arrays for improving performance.
- C-Callable IP Library {Lecture, Lab} – Describes how to create a C-callable library for IP blocks written in a hardware description language like VHDL or Verilog.
- 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 an operating system kernel, boot loaders, file system, and libraries.
- Optimizing the Design {Lecture, Lab} – Apply all the techniques you have learned to meet the performance goal for a given design.
- reVISION Stack {Lecture} – Describes the Xilinx reVISION Stack and how it enables application developers to quickly develop applications based on machine learning or computer vision using the SDx development environment.

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.

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.