AMD
together we advance_

Designing FPGAs Using the Vivado Design Suite 3

FPGA-VDES3 (v1.0)

Course Specification

Course Description

Learn how to effectively employ timing closure techniques.

This course includes:

- Demonstrating timing closure techniques such as baselining, pipelining, and synchronization circuits
- Showing optimum HDL coding techniques that help with design timing closure
- Illustrating the advanced capabilities of the Vivado™ logic analyzer to debug a design

This course builds further on the previous *Designing FPGAs Using the Vivado Design Suite* courses.

What's New for 2024.1

- All labs have been updated to the latest software versions

Level – FPGA 3

Course Details

- 2 days ILT /19

Course Part Number – FPGA-VDES3

Who Should Attend? – FPGA designers with intermediate knowledge of HDL and FPGA architecture and some experience with the Vivado Design Suite

Prerequisites

- Intermediate HDL knowledge (VHDL or Verilog)
- Solid digital design background
- *Designing FPGAs Using the Vivado Design Suite 1* course (recommended)
- *Designing FPGAs Using the Vivado Design Suite 2* course (recommended)

Software Tools

- Vivado Design Suite 2024.1

Hardware

- Architecture: UltraScale™ FPGAs*
- Demo board: Zynq™ UltraScale+™ ZCU104 board*

* This course focuses on the UltraScale architectures. Check with your local Authorized Training Provider for specifics or other customizations.

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

- Employ good alternative design practices to improve design reliability
- Define a properly constrained design
- Apply baseline constraints to determine if internal timing paths meet design timing objectives
- Apply appropriate I/O timing constraints and design modifications for source-synchronous and system-synchronous interfaces
- Optimize HDL code to maximize the FPGA resources that are inferred and meet performance goals
- Build a more reliable design that is less vulnerable to metastability problems and requires less design debugging later in the development cycle
- Perform quality of results (QoR) assessments at different stages to improve the QoR score
- Increase performance by utilizing FPGA design techniques
- Use Vivado Design Suite reports and utilities to full advantage, especially the Clock Interaction report
- Describe how to enable remote debug

Course Outline

Day 1

UltraFast™ Design Methodology (UFDM)

- **UltraFast Design Methodology: Implementation**
  Introduces the methodology guidelines covered in this course. {Lecture}

Simulation

- **Timing Simulation**
  Illustrates simulating a design post-implementation to verify that the design works properly on hardware. {Lecture, Lab}

Design Techniques

- **Baselining**
  Demonstrates how to use recommended baselining procedures to progressively meet timing closure. {Lecture, Demo, Lab}
- **Pipelining**
  Describes using pipelining to improve design performance. {Lecture, Lab}
- **Inference**
  Outlines how to infer AMD-dedicated hardware resources by writing appropriate HDL code. {Lecture, Lab}

Timing – Advanced

- **I/O Timing Scenarios**
  Provides an overview of various I/O timing scenarios, such as source- and system-synchronous, direct/MMCM capture, and edge/center-aligned data. {Lecture}
- **System-Synchronous I/O Timing**
  Demonstrates applying I/O delay constraints and performing static timing analysis for a system-synchronous input interface. {Lecture, Demo}
- **Source-Synchronous I/O Timing**
  Demonstrates applying I/O delay constraints and performing static timing analysis for a source-synchronous, double data rate (DDR) interface. {Lecture, Lab}
- **Timing Constraints Priority**
  Reviews how to identify the priority of timing constraints. {Lecture}

Design Analysis

- **Report Clock Interaction**
  Describes the clock interaction report, which is used to identify interactions between clock domains. {Lecture, Demo}
- **Report Datasheet**
  Describes the datasheet report, which is used to find the optimal setup and hold margin for an I/O interface. {Lecture, Demo}
- **QoR Reports Overview**
  Discusses what quality of result (QoR) is and how to analyze the QoR reports generated by the Vivado IDE. {Lecture}

Day 2

CDC

- **Sampling and Capturing Data in Multiple Clock Domains**
  Provides an overview of debugging a design with multiple clock domains that require multiple ILAs. {Lecture, Lab}
- **Clock Domain Crossing (CDC) and Synchronization Circuits**
  Highlights how to use synchronization circuits for clock domain crossings. {Lecture, Lab}

Version Control System

- **Revision Control Systems in the Vivado Design Suite**
  Investigates using version control systems with the Vivado design flows. {Lecture, Lab}

Power

- **Dynamic Power Estimation Using Vivado Report Power**

Describes how to use an SAIF (switching activity interface format) file to determine accurate power consumption for a design. {Lecture, Lab}

Configuration

- **Configuration Modes**
  Reviews the various configuration modes and selecting a suitable mode for a design. {Lecture}

Debugging

- **Netlist Insertion Debug Probing Flow**
  Covers the netlist insertion flow for debug using the Vivado logic analyzer. {Lecture, Lab}
- **JTAG to AXI Master Core**
  Describes how this debug core is used to write/read data to/from a peripheral connected to an AXI interface in a system that is running in hardware. {Lecture, Demo}
- **Debug Flow in an IP Integrator Block Design**
  Shows how to insert the debug cores into IP integrator block designs. {Lecture, Lab}
- **Remote Debugging Using the Vivado Logic Analyzer**
  Demonstrates using the Vivado logic analyzer to configure an FPGA, set up triggering, and view sampled data from a remote location. {Lecture, Lab}

Tcl Commands

- **Design Analysis Using Tcl Commands**
  Describes how to analyze a design using Tcl commands. {Lecture, Demo}