Designing FPGAs Using the Vivado Design Suite 2

FPGA-VDES2 (v1.0)

Course Specification

Course Description

Learn how to build a more effective FPGA design:

The focus is on:

- Using synchronous design techniques
- Utilizing the Vivado® IP integrator to create a sub-system
- Employing proper HDL coding techniques to improve design performance
- Debugging a design with multiple clock domains

This course builds on the concepts from the *Designing FPGAs Using the Vivado Design Suite 1* course.

What's New for 2022.1

- Added information on block design referencing
- Added new modules:
  - Block Design Containers in the Vivado IP Integrator
  - Versal ACAP: Hardware Platform Development Using the Vivado IP Integrator
- Added content on power awareness
- All labs have been updated to the latest software versions

Level – FPGA 2

Course Details

- 2 days ILT
  - 28 lectures
  - 11 labs
  - 7 demos

Price –

Course Part Number – FPGA-VDES2

Who Should Attend? – Digital designers who have a working knowledge of HDL (VHDL or Verilog) and who are new to Xilinx FPGAs

Prerequisites

- Intermediate HDL knowledge (Verilog or VHDL)
- Digital design knowledge and experience (attendees should be electrical engineers)
- Experience with the basics of the Tcl language
- *Designing FPGAs Using the Vivado Design Suite 1* (recommended)

Optional Videos

- Basic HDL Coding Techniques*

Software Tools

- Vivado Design Suite 2022.1

Hardware

- Architecture: UltraScale™ FPGAs and Versal® ACAPs**
- Demo board (optional): Zynq® UltraScale+™ MPSoC ZCU104 board**

* Go to www.xilinx.com/training and click the Online Training tab to view this video.

** This course focuses on the UltraScale and Versal architectures. 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:

- Identify synchronous design techniques
- Build resets into your system for optimum reliability and design speed
- Create a Tcl script to create a project, add sources, and implement a design
- Describe and use the clock resources in a design
- Create and package your own IP and add to the Vivado IP catalog to reuse
- Use the Vivado IP integrator to create a block design
- Describe the Versal ACAP clocking architecture and hardware platform development using Vivado IP integrator
- Apply timing exception constraints in a design as part of the Baselining procedure to fine tune the design
- Describe how power analysis and optimization is performed
- Describe the HDL instantiation flow of the Vivado logic analyzer

Course Outline

Day 1

UltraFast Design Methodology

- **UltraFast Design Methodology: Design Creation**
  Overview of the methodology guidelines covered in this course. {Lecture}

Design Techniques

- **Synchronous Design Techniques**
  Introduces synchronous design techniques used in an FPGA design. {Lecture}
- **Resets**
  Investigates the impact of using asynchronous resets in a design. {Lecture, Lab}
- **Register Duplication**
  Use register duplication to reduce high fanout nets in a design. {Lecture}

Clocking in the UltraScale Architecture

- **Clocking Resources**
  Describes various clock resources, clocking layout, and routing in a design. {Lectures, Lab}

Clocking in the Versal ACAP Architecture

- **Versal ACAP: Clocking Architecture**
  Discusses the clocking architecture, clock buffers, clock routing, clock management functions, and clock de-skew. {Lecture}

I/O in the UltraScale Architecture

- **I/O Logic Resources**
  Overview of I/O resources and the IOB property for timing closure. {Lectures}

SelectIO Resources in the Versal ACAP Architecture

- **SelectIO Resources**
  Describes the I/O bank, SelectIO™ interface, and I/O delay features. {Lecture}
- **Clocking and I/O Resources in the Versal ACAP - (Optional)**
  Describes how to use the Clocking Wizard and Advanced IO Wizard to configure clocking subsystems and I/O subsystems, respectively, in the Versal ACAP. {Lab}

IP Integrator

- **Creating and Packaging Custom IP**
  Create your own IP and package and include it in the Vivado IP catalog. {Lecture, Lab}
- **Using an IP Container**
  Use a core container file as a single file representation for an IP. {Lecture, Demo}

- **Designing with the IP Integrator**
  Use the Vivado IP integrator to create the uart_led subsystem. {Lecture, Demo, Lab, Case Study}
- **Block Design Containers in the Vivado IP Integrator**
  Describes the block design container (BDC) feature and shows how to create a BDC in the IP integrator. {Lecture}
- **Versal ACAP: Hardware Platform Development Using the Vivado IP Integrator**
  Describes the different Versal ACAP design flows and covers the platform creation process using the Vivado IP integrator. {Lecture, Lab}

Day 2

Timing – Intermediate

- **Timing Constraints Editor**
  Introduces the timing constraints editor tool to create timing constraints. {Lecture}
- **Report Clock Networks**
  Use `report clock networks` to view the primary and generated clocks in a design. {Lecture, Demo}
- **Timing Summary Report**
  Use the post-implementation timing summary report to sign-off criteria for timing closure. {Lecture, Demo}
- **Clock Group Constraints**
  Apply clock group constraints for asynchronous clock domains. {Lecture, Demo}
- **Introduction to Timing Exceptions**
  Introduces timing exception constraints and applying them to fine tune design timing. {Lecture, Demo, Lab}

Power

- **Power Analysis and Optimization Using the Vivado Design Suite**
  Use report power commands to estimate power consumption. {Lecture, Lab}

Configuration

- **Configuration Process**
  Reviews the FPGA configuration process, such as device power up, CRC checks, etc. {Lecture}

Debugging

- **HDL Instantiation Debug Probing Flow**
  Covers the HDL instantiation flow to create and instantiate a VIO core and observe its behavior using the Vivado logic analyzer. {Lecture, Lab}

Tcl

- **Scripting in Vivado Design Suite Project Mode**
  Explains how to write Tcl commands in the project-based flow for a design. {Lecture, Lab}
- **Design Analysis Using Tcl Commands**
  Analyze a design using Tcl commands. {Lecture, Demo, Lab}