AMD XILINX

Designing FPGAs Using the Vivado Design Suite 4

FPGA-VDES4 (v1.0)

Course Specification

Course Description

Learn how to use the advanced aspects of the Vivado® Design Suite and AMD-Xilinx hardware.

The focus is on:

- Applying timing constraints for source-synchronous and system-synchronous interfaces
- Utilizing floorplanning techniques
- Employing advanced implementation options
- Utilizing AMD-Xilinx security features
- Identifying advanced FPGA configurations
- Debugging a design at the device startup phase
- Using Tcl scripting in non-project batch flows

This is the final course in the *Designing FPGAs Using the Vivado Design Suite* series.

What's New for 2022.1

- Added content on timing closure techniques to reduce logic delay and net delay, improve clock skew, and improve clock uncertainty
- Added content on Abstract Shell for DFX
- All labs have been updated to the latest software versions

Level – FPGA 4

Course Details

- 2 days
  - 33 lectures
  - 12 labs
  - 4 demos

Price –

Course Part Number – FPGA-VDES4

Who Should Attend? – Engineers who seek advanced training in using AMD-Xilinx tools to improve FPGA performance and utilization while also increasing productivity

Prerequisites

- Intermediate HDL knowledge (Verilog or VHDL)
- Sound digital design background
- *Designing FPGAs Using the Vivado Design Suite 1* (recommended)
- *Designing FPGAs Using the Vivado Design Suite 2* (recommended)
- *Designing FPGAs Using the Vivado Design Suite 3* (recommended)

Software Tools

- Vivado Design Suite 2022.1

Hardware

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

* This course focuses on the UltraScale and Versal 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:

- Apply appropriate I/O timing constraints and design modifications for source-synchronous and system-synchronous interfaces
- Analyze a timing report to identify how to center the clock in the data eye
- Apply appropriate techniques to reduce logic and net delay and to improve clock skew and clock uncertainty
- Utilize floorplanning techniques to improve design performance
- Employ advanced implementation options, such as incremental compile flow, physical optimization techniques, and re-entrant mode as last mile strategies
- Utilize security features, bitstream encryption, and authentication using AES for design and IP security
- Identify advanced FPGA configurations, such as daisy chains and gangs, for configuring multiple FPGAs in a design
- Debug a design at the device startup phase to debug issues related to startup events, such as MMCM lock and design coming out of reset
- Use Tcl scripting in non-project batch flows to synthesize, implement, and generate custom timing reports

Course Outline

Day 1

UltraFast Design Methodology (UFDM)

- **Introduction to UltraFast Design Methodology Timing Closure**

  Introduces the UltraFast methodology guidelines on design closure. {Lecture}

Vivado Tool Flow

- **Hierarchical Design**

  Overview of the hierarchical design flows in the Vivado Design Suite. {Lecture}
- **Incremental Compile Flow**

  Utilize the incremental compile flow when making last-minute RTL changes. {Lecture, Lab}
- **Vivado Design Suite ECO Flow**

  Use the ECO flow to make changes to a previously implemented design and apply changes to the original design. {Lecture, Lab}

Vivado IP Catalog

- **Managing IP in Remote Locations**

  Store IP and related files remote to the current working project directory. {Lecture, Lab}

Timing – Advanced

- **I/O Timing Scenarios**

  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**

  Apply I/O delay constraints and perform static timing analysis for a system-synchronous input interface. {Lecture, Demo}
- **Source-Synchronous I/O Timing**

  Apply I/O delay constraints and perform static timing analysis for a source-synchronous, double data rate (DDR) interface. {Lecture, Lab}
- **Timing Constraints Priority**

  Identify the priority of timing constraints. {Lecture}
- **Timing Closure Using Physical Optimization Techniques**

  Use physical optimization techniques for timing closure. {Lecture, Lab}
- **Case Analysis**

  Understand how to analyze timing when using multiplexed clocks in a design. {Lecture}

- **Reducing Logic Delay**
  Describes how to optimize regular fabric paths and paths with dedicated blocks and macro primitives. {Lecture}
- **Reducing Net Delay**
  Reviews different techniques to reduce congestion and net delay. {Lecture}
- **Improving Clock Skew**
  Describes how to apply various techniques to improve clock skew. {Lecture}
- **Improving Clock Uncertainty**
  Reviews various flows for improving clock uncertainty, including using parallel BUFGCE_DIV clock buffers, changing MMCM or PLL settings, and limiting synchronous clock domain crossing (CDC) paths. {Lecture, Lab}

Floorplanning

- **Introduction to Floorplanning**
  Introduction to floorplanning and how to use Pblocks while floorplanning. {Lecture}
- **Design Analysis and Floorplanning**
  Explore the pre- and post-implementation design analysis features of the Vivado IDE. {Lecture, Lab}
- **Congestion**
  Identifies congestion and addresses congestion issues. {Lecture}

Day 2

Power

- **Power Management Techniques**
  Identify techniques used for low power design. {Lecture}
- **Versal ACAP: Power Management**
  Discusses power domains and how they can be controlled along with basic techniques used to lower overall power consumption. {Lecture}

Configuration

- **Daisy Chains and Gangs in Configuration**
  Introduces advanced configuration schemes for multiple FPGAs. {Lecture}
- **Bitstream Security**
  Understand the AMD-Xilinx bitstream security features such as readback disable, bitstream encryption, and authentication. {Lecture, Demo}

Debugging

- **Vivado Design Suite Debug Methodology**
  Understand and follow the debug core recommendations. Employ the debug methodology for debugging a design using the Vivado logic analyzer. {Lecture}
- **Trigger and Debug at Device Startup**
  Debug the events around the device startup. {Lecture, Demo}
- **Trigger Using the Trigger State Machine in the Vivado Logic Analyzer**
  Use trigger state machine code to trigger the ILA and capture data in the Vivado logic analyzer. {Lecture, Lab}
- **Versal ACAP: Debugging**
  Covers the Versal ACAP debug interfaces, such as the test access port (TAP), debug access port (DAP) controller, and high-speed debug port (HSDP). {Lecture, Lab}

Vivado Store

- **Introduction to the Vivado Store**
  Introduces the Xilinx Vivado Store. {Lecture, Demo}

Tcl Commands

- **Scripting in Vivado Design Suite Non-Project Mode**
  Write Tcl commands in the non-project batch flow for a design. {Lecture, Lab}
- **Debugging the Design Using Tcl Commands**
  Use Tcl scripting for VLA designs for adding probes and making connections to probes. {Lecture, Lab}
- **Using Procedures in Tcl Scripting**
  Employ procedures in Tcl scripting. {Lecture}
- **Using Lists in Tcl Scripting**
  Employ lists in Tcl scripting. {Lecture}
- **Using Regular Expressions in Tcl Scripting**
  Use regular expressions to find a pattern in a text file while scripting an action in the Vivado Design Suite. {Lecture, Lab}
- **Debugging and Error Handling in Tcl Scripts**
  Understand how to debug errors in a Tcl script. {Lecture}