Vivado Advanced Advanced FPGA Design

FPGA-VDES-3-4-UFDM-DC-COMBINED (v1.0)

Course Description

This course for experienced Xilinx FPGA designers allows you to maximize QoR in terms of clock rates, timing closure and power management. This class also enhance both individual and team productivity. The complete range of topics, tips and "best practices" gives you complete control of the Vivado DS tool flow. That includes both project and non-project modes. Using advanced analysis and the most up-to-date optimization strategies, you'll be able to thoroughly leverage every available Vivado DS capability. This custom class combines key elements from both the "Designing FPGAs with Vivado" -Level 3 & 4 classes, along with the "Ultra-Fast Design Methodology" and the new "FPGA Design Closure" classes from AMD Xilinx Customer Education. This combination effectively gives you all the advanced FPGA design insights in one place, including new Vivado ML Edition features and resources.

The emphasis of this course is on:

- Advanced design analysis and debug
- Source-synchronous IO timing constraints
- New Vivado ML Intelligent Design Run (IDR) features
- Timing and design closure techniques
-

Level – FPGA 3 & 4

Course Details

- 3-day classroom training / onsite
- 24 hours online (4 sessions, 6 hr. each)
- 45 lectures
- 26 labs (select labs conducted during class, attendees can choose additional relevant take-home exercises)

Course Part Number – FPGA-VDES-3-4-UFDM-DC-COMBINED

Who Should Attend? – Experienced Xilinx FPGA designers needing to maximize their QoR, individual/team productivity, and fully leverage the Vivado Design Suite.

Prerequisites

- FPGA Design Essentials
- Or Designing FPGA with Vivado DS – Level 1 or 2
- Or Ultrafast Design Methodology
- Or roughly 1 year Xilinx FPGA design experience

Software Tools

- Vivado Design Suite

Hardware

- Architecture: Xilinx 7-Series and Ultra-Scale FPGA families

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 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

Course Specification

- 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
- 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
- Apply baseline constraints to determine if internal timing paths meet design timing objectives
- 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
- Implement Intelligent Design Runs (IDR) to automate analysis and timing closure for complex designs
- Perform quality of results (QoR) assessments at different stages to improve the QoR score
- Use Vivado Design Suite reports and utilities to full advantage, especially the Clock Interaction report
- Describe how to enable remote debug

Course Outline

Session 1

Vivado Tool Flow

- **Vivado Design Suite Non-Project Mode**
  Create a design in the Vivado Design Suite non-project mode.

Tcl Commands

- **Scripting in Vivado Design Suite Non-Project Mode**
  Write Tcl commands in the non-project batch flow for a design.

Simulation

- **Timing Simulation**
  Simulate the design post-implementation to verify that a design works properly on hardware.

Design Techniques

- **Baselining**
  Use recommended baselining procedures to progressively meet timing closure.
- **Pipelining**
  Use pipelining to improve design performance. {Lecture, Lab}
- **Inference**
  Infer AMD-Xilinx dedicated hardware resources by writing appropriate HDL code.
- **Sampling and Capturing Data in Multiple Clock Domains**
  Overview of debugging a design with multiple clock domains that require multiple ILAs.
- **Clock Domain Crossing (CDC) and Synchronization Circuits**

Use synchronization circuits for clock domain crossings.

Design Runs

- **Intelligent Design Runs (IDR)**
  Introduces Intelligent Design Runs (IDR), which are special types of implementation runs that use a complex flow to attempt to close timing.

Design Analysis

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

Session 2

Version Control System

- **Revision Control Systems in the Vivado Design Suite**
  Use version control systems with Vivado design flows.

Power

- **Dynamic Power Estimation Using Vivado Report Power**
  Use an SAIF (switching activity interface format) file to determine accurate power consumption for a design.

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.
- **System-Synchronous I/O Timing**
  Apply I/O delay constraints and perform static timing analysis for a system-synchronous input interface.
- **Source-Synchronous I/O Timing**
  Apply I/O delay constraints and perform static timing analysis for a source-synchronous, double data rate (DDR) interface.
- **Timing Constraints Priority**
  Identify the priority of timing constraints.
- **Case Analysis**
  Understand how to analyze timing when using multiplexed clocks in a design.

Floorplanning

- **Introduction to Floorplanning**
  Introduction to floorplanning and how to use Pblocks while floorplanning.
- **Design Analysis and Floorplanning**
  Explore the pre- and post-implementation design analysis features of the Vivado IDE.
- **Congestion**
  Identifies congestion and addresses congestion issues.
- **Timing Closure Using Physical Optimization Techniques**
  Use physical optimization techniques for timing closure.

Power

Course Specification

- **Power Management Techniques**
  Identify techniques used for low power design.

Tcl Commands

- **Manipulating Design Properties Using Tcl**
  Query your design and make pin assignments by using various Tcl commands.

Session 3

Timing – Closure

- **Reducing Logic Delay**
  Describes how to optimize regular fabric paths and paths with dedicated blocks and macro primitives.
- **Reducing Net Delay**
  Reviews different techniques to reduce congestion and net delay.
- **Improving Clock Skew**
  Describes how to apply various techniques to improve clock skew.
- **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.
- **Introduction to UltraFast Design Methodology Timing Closure**
  Introduces the UltraFast methodology guidelines on design closure.

Vivado Tool Flow

- **Hierarchical Design**
  Overview of the hierarchical design flows in the Vivado Design Suite.
- **Incremental Compile Flow**
  Utilize the incremental compile flow when making last-minute RTL changes.
- **Vivado Design Suite ECO Flow**
  Use the ECO flow to make changes to a previously implemented design and apply changes to the original design.

Vivado IP Catalog

- **Managing IP in Remote Locations**
  Store IP and related files remote to the current working project directory. {Lecture, Lab}

Vivado Store

Introduces the Xilinx Vivado Store.

Session 4

Configuration

- **Configuration Modes**
  Understand various configuration modes and select the suitable mode for a design.
- **Daisy Chains and Gangs in Configuration**
  Introduces advanced configuration schemes for multiple FPGAs.
- **Bitstream Security**

Understand the AMD-Xilinx bitstream security features such as readback disable, bitstream encryption, and authentication.

Debugging

- **Netlist Insertion Debug Probing Flow**
  Covers the netlist insertion flow of the debug using the Vivado logic analyzer.
- **JTAG to AXI Master Core**
  Use this debug core to write/read data to/from a peripheral connected to an AXI interface in a system that is running in hardware.
- **Debug Flow in an IP Integrator Block Design**
  Insert the debug cores into IP integrator block designs.
- **Remote Debugging Using the Vivado Logic Analyzer**
  Use the Vivado logic analyzer to configure an FPGA, set up triggering, and view the sampled data from a remote location.
- **Trigger and Debug at Device Startup**
  Debug the events around the device startup.
- **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.

UltraFast Design Methodology (UFDM)

- **UltraFast Design Methodology: Implementation**
  Reinforces the methodology guidelines covered throughout this course.

AVAILABLE LABS

- **Timing Simulation**
- **Baselining**
- **Pipelining**
- **Inference**
- **Sampling and Capturing Data in Multiple Clock Domains**
- **Clock Domain Crossing (CDC) and Synchronization Circuits**
- **Debug Case Study: Report CDC**
- **Increasing Design Performance Using Report QoR**
- **Intelligent Design Runs *(Featured)***
- **Revision Control Systems in the Vivado Design Suite**
- **Dynamic Power Estimation Using Vivado Power Report**
- **Netlist Insertion Debug Probing Flow**
- **Debug Flow in an IP Integrator Block Design**
- **Remote Debugging Using the Vivado Logic Analyzer**
- **Manipulating Design Properties Using Tcl**
- **Incremental Compile Flow**
- **Vivado Design Suite ECO Flow *(Featured)***
- **Managing IP in Remote Locations**
- **Source-Synchronous I/O Timing**
- **Timing Closure Using Phys_Opt Techniques *(Featured)***
- **Improving Clock Uncertainty**
- **Design Analysis and Floorplanning**
- **Trigger Using the Trigger State Machine in the Vivado Logic Analyzer**

Course Specification

- **Scripting in Vivado DS Non-Project Mode *(Featured)***
- **Debugging a Design Using Tcl Commands**
- **Using Regular Expressions in Tcl Scripting**