-course will update experienced ISE® software users to utilize the Vivado® Design Suite. Learn the underlying database and static timing analysis (STA) mechanisms. Utilize Tcl for navigating the design, creating Xilinx design constraints (XDC), and creating timing reports. Learn to make appropriate timing constraints for SDR, DDR, source-synchronous, and system-synchronous interfaces for your FPGA design.

You will also learn to make path-specific, false path, and min/max timing constraints, as well as learn about timing constraint priority in the Vivado timing engine. Finally, you will learn about the scripting environment of the Vivado Design Suite and how to use the project-based and non-project batch flows.

You will also learn the FPGA design best practices and skills to be successful using the Vivado Design Suite. This includes the necessary skills to improve design speed and reliability, including: system reset design, synchronization circuits, optimum HDL coding techniques, and timing closure techniques using the Vivado software. This course encapsulates this information with an FPGA design methodology case study. The UltraFast™ design methodology checklist is also introduced.

**Level** – FPGA 2

**Course Duration** – 3 days

**Price** – $2100 USD

**Course Part Number** – VIVA11000-ILT

**Who Should Attend?** – Existing Xilinx ISE Design Suite FPGA designers

**Prerequisites**

- FPGA design experience
- Completion of the Essentials of FPGA Design, Designing for Performance, and Advanced FPGA Implementation courses or equivalent knowledge of Xilinx ISE software implementation tools, techniques, architecture, and FPGA design techniques. Completion of the Vivado Design Suite for ISE Project Navigator Users course is strongly recommended.
- Intermediate VHDL or Verilog knowledge

**Recommended**

- Essential Tcl Scripting for the Vivado Design Suite course
- Vivado Design Suite for ISE Software Project Navigator Users course

**Software Tools**

- Vivado Design or System Edition 2014.1

**Hardware**

- Architecture: 7 series FGPGAs*
- Demo board: None*

* This course focuses on the 7 series architecture. Check with your local Authorized Training Provider for specifics or other customizations.

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

- Access primary objects from the design database and filter lists of objects using properties
- Describe setup and hold checks and describe the components of a timing report
- Create appropriate input and output delay constraints and describe timing reports that involve input and output paths
- Explain the impact that manufacturing process variations have on timing analysis and describe how min/max timing analysis information is conveyed in a timing report
- Describe all of the options available with the report_timing and report_timing_summary commands
- Describe the timing constraints required to constrain system-synchronous and source-synchronous interfaces
- Analyze a timing report to identify how to center the clock in the data eye

**Course Specification**

- Create scripts for the project-based and non-project batch design flows
- Describe the UltraFast design methodology checklist
- Identify key areas to optimize your design to meet your design goals and performance objectives
- Define a properly constrained design
- Optimize HDL code to maximize the FPGA resources that are inferred and meet your performance goals
- Build resets into your system for optimum reliability and design speed
- Build a more reliable design that is less vulnerable to metastability problems and requires less design debugging later in the development cycle
- Use Vivado Design Suite reports and utilities to full advantage, especially the Clock Interaction report
- Identify timing closure techniques using the Vivado Design Suite
- Describe how the Xilinx design methodology techniques work effectively through case study/lab experience

**Course Outline**

**Day 1**

- UltraFast Design Methodology Summary
- Vivado IDE Review
- Accessing the Design Database
- Lab 1: Vivado IDE Database
- Demo: Clock Creation and Basic Static Timing Analysis
- Static Timing Analysis and Clocks
- Lab 2: Vivado IDE Clocks
- Inputs and Outputs
- Lab 3: I/O Clocks
- Timing Exceptions
- Lab 4: Timing Exceptions

**Day 2**

- Advanced Timing Analysis
- Demo: Timing Reports
- System-Synchronous and Source-Synchronous I/O Timing
- Demo: System-Synchronous I/O Timing
- Lab 5: Advanced I/O Timing
- Project-Based and Non-Project Batch Design Flows
- Scripting Using Project-Based and Non-Project Batch Flows
- Lab 6a: Scripting in the Project-Based Flow
- Lab 6b: Scripting in the Non-Project Batch Flow
- Appendix: Advanced Timing Analysis
- Appendix: Advanced I/O Constraints – Virtual Clocks

**Day 3**

- Demo: UltraFast Design Methodology Checklist
- UltraFast Design Methodology
- HDL Coding Techniques
- Reset Methodology
- Lab 7: Resets
- Lab 8: SRL and DSP Inference
- Synchronization Circuits and the Clock Interaction Report
- Timing Closure
- FPGA Design Methodology Case Study
- Lab 9: Timing Closure and Design Conversion
- Appendix: Synchronization Circuits and the Clock Interaction Report

© 2014 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and disclaimers are as listed at http://www.xilinx.com/legal.htm. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.
Appendix: Replication, Fanout, and Physical Optimization
Appendix: Timing Closure
Appendix: Pipelining lab

Lab Descriptions

- **Lab 1:** Vivado IDE Database – Explore the Vivado IDE database using Tcl commands. Use the Tcl Console to evaluate and enter IOB properties.
- **Lab 2:** Vivado IDE Clocks – Create complete XDC constraints for the clocking resources in a design. Implement the design and use the available clocking reports to verify results. Understand the first step in the Xilinx baselining recommendation.
- **Lab 3:** I/O Constraints – Create input and output constraints for a source-synchronous design by using the Timing Constraints utility. You will also generate useful timing reports to verify the timing results. Understand the second step in the baselining recommendation.
- **Lab 4:** Timing Exceptions – Use the Timing Constraints window to enter timing exceptions in the XDC format. You will also generate a useful timing report to verify the timing results. Understand the third step in the baselining recommendation.
- **Lab 5:** Advanced I/O Timing – Make I/O timing constraints for a source-synchronous, double data rate (DDR) interface. Perform a static timing analysis of the interfaces to determine the optimal clock and data relationship for maximum setup and hold-time margin. Finally, adjust the data path delay to realize the optimal timing solution.
- **Lab 6a:** Scripting in the Project-Based Flow – Write Tcl commands in the project-based flow for the design process (from creating a new project through implementation).
- **Lab 6b:** Scripting in the Non-Project Batch Flow – Write Tcl commands in the non-project batch flow for the design process (from creating a new project through implementation).
- **Lab 7:** Resets – Investigate the proper design and use of resets. Examine the impact of seeing a design built originally with asynchronous resets, having resets removed, and finally with synchronous resets only used where necessary.
- **Lab 8:** SRL and DSP Inference – Evaluate the implementation results of a design that uses asynchronous resets and infers more dedicated hardware resources when resets are selectively removed from the design. You will also learn how to infer the DSP hardware resources for other common functions required by most FPGA designs.
- **Lab 9:** Timing Closure and Design Conversion – Learn how a generic processor design was optimized for the 7 series device architecture with basic design changes that impacted the dedicated hardware usage, design speed, and the device utilization.

Register Today by contacting: info@vailogic.com