Vivado Design Suite Static Timing Analysis and Xilinx Design Constraints

Course Description

This course offers detailed training on the Vivado® software tool flow, Xilinx design constraints (XDC), and static timing analysis (STA). Learn to use good FPGA design practices and all FPGA resources to advantage. Learn to fully and appropriately constrain your design by using industry-standard XDC constraints. Learn how the the Vivado IDE design database is structured and learn to traverse the design. Create appropriate timing reports to perform full STA and how to appropriately synthesize your design. 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 3

Training Duration: 3 day

Who Should Attend?

FPGA designers with intermediate knowledge of HDL and FPGA architecture, and some experience with the Xilinx Vivado Design Suite

Prerequisites:

- Essentials of FPGA Design course or equivalent knowledge of FPGA architecture features; the Vivado software flow; basic FPGA design techniques; basic clock, input, and output timing constraints, and the Constraints Editor
- Intermediate HDL knowledge (VHDL or Verilog)
- Solid digital design background

Recommended Prerequisites

- Basic HDL Coding Techniques
- FPGA Power Estimation

Software Tools:

- Vivado System Edition 2013.3

Hardware:

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

Skills Gained: After completing this training, you will be able to:

- Use good alternative design practices to improve design reliability
- Increase performance by utilizing FPGA design techniques
- Describe the details of Vivado IDE database objects
- Identify Tcl commands for interacting with the database
- Apply complete Xilinx design constraints (XDC), including timing exceptions, false paths, and multi-cycle path constraints
- Utilize static timing analysis (STA) to analyze timing results
- Pinpoint design bottlenecks by using appropriate timing reports
- Apply advanced I/O timing constraints to meet performance goals
- Describe different synthesis options and how they can improve design performance
- 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

1. Review of Essentials of FPGA Design

2. Design Methodology Summary

3. FPGA Design Techniques

4. Accessing the Design Database

Lab 1: Vivado IDE Database

5. Static Timing Analysis and Clocks

Lab 2: Vivado Clocks

7. Inputs and Outputs

Lab Description

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.

Vivado Design Suite Static Timing Analysis and Xilinx Design Constraints

Course Outline

Lab 3: I/O Constraints
8. Timing Exceptions
Lab 4: Timing Exceptions
9. Synthesis Techniques
Appendix: Design Methodology
Appendix: HDL Coding Techniques
10. UltraFast Design Methodology Checklist
11. FPGA Design Methodology
12. HDL Coding Techniques
13. Reset Methodology
Lab 5: Resets
Lab 6: SRL and DSP Inference
14. Synchronization Circuits and the Clock Interaction Report
15. Timing Closure
16. FPGA Design Methodology Case Study
Lab 7: Timing Closure and Design Conversion
Appendix: Timing Constraints Review
Appendix: Synchronization Circuits and the Clock Interaction Report
Appendix: Fanout and Logic Replication
Appendix: Pipelining lab

Lab Description

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 and last step in the baselining recommendation.

Lab 5: 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 6: 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 7: 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.