Migrating to Xilinx for experienced FPGAs

Course Description

This course focuses on providing ASIC/ FPGA experienced designers with the Xilinx tool-set flow. Current designers will get familiar with the various Xilinx tools, (ISE, XST, MAP, Place and Route, Trace…) and design techniques. HDL inference of FPGA resources and coding examples are provided. The course highlights the Virtex-IV family though most concepts can also be applied to Virtex-based designs. HDL inference of FPGA resources and coding examples are provided.

Level: Intermediate to Advanced

Training Duration: 2 days

Course Objectives

- To ensure that ASIC or other vendors' FPGAs have a smooth migration to Xilinx
- Decrease time spent re-targeting existing RTL code to Xilinx FPGA
- Decrease time to market through a higher knowledge of the FPGA design f low
- Improve performance and decrease run-times by using appropriate constraints, implementation options, and implementation tools

Prerequisites:

- Knowledge ASIC/FPGA design flow
- At least six months design experience with any FPGA family
- Intermediate knowledge of Verilog/VHDL
- Knowledge in digital design tools, (Simulation, Synthesis, Place and Route)

Software Tools:

ISE
Mentor Graphics Precision RTL
Synplicity Synplify Pro

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

- Write HDL code to efficiently target Virtex-IV architectural resources
- Write constrain file
- Improve design performance and manage software runtime
- Use of third party IP cores within Xilinx ISE
- Evaluate your design goal by reading reports

Course Outline

1. Course Introduction / Agenda

2. Current Xilinx FPGA architecture

3. Designing with Virtex-4 FPGA resources

4. Xilinx tools flow
Lab 1: Xilinx Tool Flow

5. Core Generator Software
Lab 2: Core Generator Software

6. Designing Clock Resources
Lab 3: Designing Clock Resources

7. Synthesis Techniques
Lab 4: Synthesis Techniques

8. Achieving Timing Closure

9. Global Timing Constraints
Lab 5: Global Timing Constraints

10. Timing Groups and Offset Constraints

11. Path-Specific Timing Constraints
Lab 6: Achieving Timing Closure

12. Implementation Options (Advanced)
Lab 7: Implementation Options

13. Power Estimation

14. FPGA Editor Demo Lab

15. ChipScope Pro Analyzer (Optional)
Lab 8: ChipScope Pro Analyzer (Optional)

16. Course Summary

Lab Descriptions

Lab 1: Xilinx Tool Flow: Create a new project in the ISE Project Navigator and use the Architecture Wizard and PACE tool in the design process. Implement a design by using default software options. The design will be simulated.

Lab 2: CORE Generator Software System: Create a core, instantiate the core into VHDL or Verilog source code, and run behavioral simulation

Lab 3: Designing Clock Resources: Use the Clocking Wizard to configure the DCMs and global clock buffer resources

Lab 4: Synthesis Techniques: Experiment with different synthesis options and view the results. Versions of this lab are available for Synplicity Synplify Pro, Precision RTL, and Xilinx XST software

Lab 5: Global Timing Constraints: Enter global timing constraints with the Xilinx Constraints Editor. Review the Post-Map Static Timing Report to verify that the timing constraints are realistic. Use the Post-Place & Route Static Timing Report to determine the delay of the longest constrained path for each timing constraint.

Lab 6: Achieving Timing Closure: Review timing reports and enter path-specific timing constraints to meet performance goals

Lab 7: Implementation Options: Adjust process properties and I/O configuration options to improve the design performance

Lab FPGA Editor Demo: Use the FPGA Editor to view a design and add a probe to an internal net.

Lab 8: ChipScope Pro Analyzer: Add an internal logic analyzer to a design to perform real-time debugging.