

Course Description

This course will update experienced ISE® software users on how to utilize the Vivado® Design Suite.

The emphasis is on:

- Reviewing the underlying database and static timing analysis (STA) mechanisms
- Utilizing Tcl for navigating the design, creating Xilinx design constraints (XDC), and creating timing reports
- Applying appropriate timing constraints for SDR, DDR, source-synchronous, and system-synchronous interfaces
- Creating path-specific, false path, and min/max timing constraints as well as learning about timing constraint priority in the Vivado timing engine
- Utilizing a project-based scripting flow
- Employing FPGA design best practices and the UltraFast Design Methodology to improve design speed and reliability

Level – FPGA 2

Course Duration – 2 days live instructor led training (online or in person)

Price – \$1,600 or 16 Xilinx Training Credits

Course Part Number – FPGA-VAXDC4ISE

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

Prerequisites

- Completion of the [Vivado Design Suite for ISE Project Navigator Users](#) course is strongly recommended.
- Working HDL knowledge (VHDL or Verilog)
- Digital Design Experience

Optional Video

- Basic HDL Coding Techniques

Software Tools

- Vivado Design or System Edition 2018.1

Hardware

- Architecture: UltraScale™ and 7 series FPGAs*
- Demo board: None*

* This course focuses on the UltraScale and 7 series architectures. Check with [Morgan Advanced Programmable Systems, Inc.](#) for the specifics of the in-class lab board or other customizations.

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

- Create appropriate clock and input, output delay constraints and describe timing reports that involve input and output paths
- Analyze different timing reports
- Define a properly constrained design
- Describe setup and hold checks and describe the components of a timing report
- Identify key areas to optimize your design to meet your design goals and performance objectives
- Describe all of the options available with the `report_timing` and `report_timing_summary` commands
- Build a more reliable design that is less vulnerable to metastability problems and requires less design debugging later in the development cycle

- Describe the timing constraints required to constrain system-synchronous and source-synchronous interfaces
- Identify timing closure techniques using the Vivado® Design Suite
- Describe how the UltraFast design methodology techniques work effectively through case studies and lab experiences

Course Outline

Day 1

- Introduction to Clock Constraints {Lecture, Lab, Demo}
- Generated Clocks {Lecture, Demo}
- Report Clock Networks {Lecture, Demo}
- Clock Group Constraints {Lecture, Demo}
- I/O Constraints and Virtual Clocks {Lecture, Lab}
- Timing Constraints Wizard {Lecture, Lab}
- Introduction to Vivado Reports {Lecture, Demo}
- Setup and Hold Timing Analysis {Lecture}
- Timing Summary Report {Lecture, Demo}
- Report Clock Interaction {Lecture, Demo}
- Introduction to Timing Exceptions {Lecture, Lab, Demo}
- Timing Constraints Priority {Lecture}

Day 2

- Synchronization Circuits {Lecture, Lab, Case Study}
- Report Datasheet {Lecture, Demo}
- UltraFast Design Methodology: Implementation {Lecture}
- Baselineing {Lecture, Lab, Demo}
- Pipelining {Lecture, Lab}
- I/O Timing Scenarios {Lecture}
- System-Synchronous I/O Timing {Lecture, Demo}
- Source-Synchronous I/O Timing {Lecture, Lab}
- Introduction to Floorplanning {Lecture}
- Congestion {Lecture}
- Physical Optimization {Lecture, Lab}
- UltraFast Design Methodology: Design Closure {Lecture}

Topic Descriptions

Day 1

- Introduction to Clock Constraints – Apply clock constraints and perform timing analysis.
- Generated Clocks – Use the report clock networks report to determine if there are any generated clocks in a design.
- Report Clock Networks – Use report clock networks to view the primary and generated clocks in a design.
- Clock Group Constraints – Apply clock group constraints for asynchronous clock domains.
- I/O Constraints and Virtual Clocks – Apply I/O constraints and perform timing analysis.
- Timing Constraints Wizard – Use the Timing Constraints Wizard to apply missing timing constraints in a design.
- Introduction to Vivado Reports – Generate and use Vivado timing reports to analyze failed timing paths.
- Setup and Hold Timing Analysis – Understand setup and hold timing analysis.

- Timing Summary Report – Use the post-implementation timing summary report to sign-off criteria for timing closure.
- Report Clock Interaction – Use the clock interaction report to identify interactions between clock domains.
- Introduction to Timing Exceptions – Introduces timing exception constraints and applying them to fine tune design timing.
- Timing Constraints Priority – Identify the priority of timing constraints.

Day 2

- Synchronization Circuits – Use synchronization circuits for clock domain crossings.
- Report Datasheet – Use the datasheet report to find the optimal setup and hold margin for an I/O interface.
- UltraFast Design Methodology: Implementation – Introduces the methodology guidelines covered in this course.
- Baselineing – Use Xilinx-recommended baselineing procedures to progressively meet timing closure.
- Pipelining – Use pipelining to improve design performance.
- 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.
- Introduction to Floorplanning – Introduction to floorplanning and how to use Pblocks while floorplanning.
- Congestion - Identifies congestion and addresses congestion issues.
- Physical Optimization – Use physical optimization techniques for timing closure.
- UltraFast Design Methodology: Design Closure – Introduces the methodology guidelines covered in this course.

Register Today

Morgan Advanced Programmable Systems, Inc. (Morgan A.P.S.) delivers public and private courses in locations throughout the central US region; including Iowa, Illinois, Kansas, Minnesota, Missouri, Nebraska, North Dakota, South Dakota, and Wisconsin.

Visit morgan-aps.com/training, for full course schedule and training information.



You must have your tuition payment information available when you enroll. We accept credit cards (Visa, MasterCard, or American Express) as well as purchase orders and Xilinx training credits.

Student Cancellation Policy

- Student cancellations received more than 7 days before the first day of class are entitled to a 100% refund. Refunds will be processed within 14 days.
- Student cancellations received less than 7 days before the first day of class are entitled to a 100% credit toward a future class.
- Student cancellations must be sent [here](#).

Morgan A.P.S. Course Cancellation Policy

- We regret from time-to-time classes will need to be rescheduled or cancelled.
- In the event of cancellation, live on-line training may be offered as a substitute.
- Morgan A.P.S. may cancel a class up to 7 days before the scheduled start date of the class; all students will be entitled to a 100% refund.
- Under no circumstances is Morgan A.P.S. responsible or liable for travel, lodging or other incidental costs. Please be aware of this cancellation policy when making your arrangements.
- For additional information or to schedule a private class contact us [here](#).

Online training with real hardware

During the Covid-19 period, some companies do not allow their staff to participate in live in-person training.

- Consequently, Morgan Advanced Programmable Systems, Inc. has set up a training VPN where engineer participants can take classes online using the same computers and devCards used during in-person training.
- Even better, and upon request, you can use these computers after hours on training days to experiment with labs. This is not possible for in-person training.
- Additionally, just like in-person training, the laptops and devCards, tools, OS, and licensing are set up in advance.
- In some ways, live online-training is better than in-person...for example, you can grant the instructor permission to look at your Vivado, PetaLinux terminal, or Vitis for extended periods of time if your lab is not going exactly as planned to a missed step.
- This is often more comfortable than two engineers crowding around a laptop screen.
- Taking remote training also allows you to learn some tips and tricks for working remote. Whether your devCard is in the lab down the hall, or across the world via VPN, you can control your Xilinx based device quickly and efficiently.