Designing with Verilog

FPGA 1

LANG-VERILOG (v1.0)

Course Specification

Course Description

This course provides a thorough introduction to the Verilog language.

The emphasis is on:

- Writing efficient hardware designs
- Performing high-level HDL simulations
- Employing structural, register transfer level (RTL), and behavioral coding styles
- Targeting Xilinx devices specifically and FPGA devices in general
- Utilizing best coding practices

This course covers Verilog 1995 and 2001.

What's New for 2021.1

- All labs have been updated to the latest software versions

Level – FPGA 1

Course Details

- 3 days live instructor led training (online or in person)
  - 35 lectures
  - 9 labs
  - 1 demo

Price – $2,400 or 24 Xilinx Training Credits

Course Part Number – LANG-VERILOG

Who Should Attend? – Engineers who want to use Verilog effectively for modeling, design, and synthesis of digital designs

Subsequent Courses

- Designing with SystemVerilog

Software Tools

- Vivado® Design Suite 2021.1

Hardware

- Architecture: N/A*
- Demo board: Zynq® UltraScale+™ MPSoC ZCU104 board*

* This course does not focus on any particular architecture. 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:

- Write RTL Verilog code for synthesis
- Write Verilog test fixtures for simulation
- Create a finite state machine (FSM) by using Verilog
- Target and optimize Xilinx FPGAs by using Verilog
- Use enhanced Verilog file I/O capabilities
- Run a timing simulation by using Xilinx Simprim libraries
- Create and manage designs within the Vivado Design Suite environment
- Download to the evaluation demo board

Course Outline

Day 1

- **Introduction to Verilog**
  Discusses the history of the Verilog language and provides an overview of the different features of Verilog. {Lecture}
- **Verilog Keywords and Identifiers**
  Discusses the data objects that are available in the Verilog language as well as keywords and identifiers. {Lecture}
- **Verilog Data Values and Number Representation**
  Covers what data values are in Verilog, as well as how to represent numbers in Verilog. {Lecture}
- **Verilog Data Types**
  Covers the various data types in Verilog. {Lecture}
- **Verilog Buses and Arrays**
  Covers buses and arrays in Verilog. {Lecture}
- **Verilog Modules and Ports**
  Describes both the syntax and hierarchy for a Verilog module, port declarations, and the difference between reg versus wire. {Lecture, Demo, Lab}
- **Verilog Operators**
  Shows the syntax for all Verilog operators. {Lecture}
- **Continuous Assignment**
  Introduces the Verilog continuous assignment statement. {Lecture}
- **Gate-Level Modeling**
  Introduces gate-level modeling in Verilog {Lecture}
- **Procedural Assignment**
  Introduces procedural assignments in Verilog, including their usage and restrictions. {Lecture}
- **Blocking and Non-Blocking Procedural Assignment**
  Introduces blocking and non-blocking assignment statements in Verilog. {Lecture, Lab}
- **Procedural Timing Control**
  Introduces the timing control methods that are used in procedural assignments. {Lecture}

Day 2

- **Verilog Conditional Statements: if_else**
  Describes the if/else conditional statement. {Lecture, Lab}
- **Verilog Conditional Statements: case**
  Describes the case conditional statement. {Lecture}
- **Verilog Loop Statements**
  Introduces the different types of Verilog loop statements. {Lecture}
- **Introduction to the Verilog Testbench**
  Introduces the concept of the Verilog testbench {Lecture, Lab}
- **System Tasks**
  Provides a basic understanding of system tasks. {Lecture}
- **Verilog Subprograms**
  Covers the use of subprograms in verification and RTL code to model functional blocks. {Lecture}
- **Verilog Functions**
  Describes functions which are integral to reusable and maintainable code. {Lecture}
- **Verilog Tasks**
  Covers tasks in Verilog. {Lecture}
- **Verilog Compiler Directives**
  Describes Verilog compiler directives. {Lecture}
- **Verilog Parameters**
  Covers Verilog parameters and the local parameter concept. {Lecture, Lab}
- **Verilog Generate Statements**
  Introduces the Verilog generate statement. {Lecture}

© 2021 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.

Day 3

- **Timing Checks**
  Covers the timing check statements in Verilog and talks about the *specify* block. {Lecture}
- **Finite State Machines**
  Provides an overview of finite state machines, one of the more commonly used circuits. {Lecture}
- **Mealy Finite State Machine**
  Describes the Mealy FSM and how to code for it. {Lecture, Lab}
- **Moore Finite State Machine**
  Describes the Moore FSM and how to code for it. {Lecture, Lab}
- **FSM Coding Guidelines**
  Shows how to model an FSM of any complexity in Verilog and describes recommendations for performance and reliability. {Lecture}
- **Avoiding Race Conditions in Verilog**
  Describes what a race condition is and provides steps to avoid this condition. {Lecture}
- **File I/O: Introduction**
  Covers using basic and enhanced Verilog file I/O capabilities for more robust design verification. {Lecture}
- **File I/O: Read Functions**
  Covers Verilog file I/O read capabilities. {Lecture, Lab}
- **File I/O: Write Functions**
  Covers Verilog file I/O write capabilities. {Lecture}
- **Targeting Xilinx FPGAs**
  Focuses on Xilinx-specific implementation and chip-level optimization. {Lecture, Lab}
- **User-Defined Primitives**
  Describes user-defined primitives (UDPs). {Lecture}
- **Programming Language Interface**
  Introduces the programming language interface (PLI) in Verilog. {Lecture}

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.

Morgan Advanced Programmable Systems

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 has 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.

© 2021 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.