



Verilog Introduction

2 Days

70% Lecture, 30% Lab

Basic Level

Overview

The Verilog hardware description language plays a key role in design flows for ASICs and FPGAs. It is increasingly important that people involved with hardware design have a background in this language and an awareness of the features introduced by IEEE Std 1364-2001.

This course provides a basic introduction to the main features of the Verilog language. The course will familiarize the student with the language and bridge the gap between the basic concepts in digital logic (schematics, Boolean equations, truth tables, etc.) and related constructs used in Verilog-based design flows. Several examples will illustrate language features, including an introduction to modeling styles suitable for simulation and synthesis.

Benefits

Upon completion of this course, students will:

- understand basic language concepts and usage
- understand the role of Verilog in ASIC and FPGA design flows
- understand the use of Verilog in top-down design methodology
- be able to write simple models of combinational and sequential logic
- write and execute a test plan and develop a testbench for verifying a model
- be able to run a simulator to verify a model
- understand the enhancements introduced by Verilog 2001

Intended Audience

Verilog Introduction is recommended for technical personnel with little or no knowledge of Verilog who need to learn the language for understanding someone else's code. Managers and engineering support people will benefit from this course.

For people who need to learn Verilog for designing, it is recommended that they take the 4-day *Verilog for Hardware Designers* which covers more of the language and has extensive labs.

Prerequisites

Students need to be familiar with the basics of digital design of combinational and sequential logic.

Suggested follow-on course:

Advanced Verilog Coding Styles for Synthesis & Verification

Training Approach

This is an intensive, interactive course, which is approximately 70% lecture and 30% lab. Questions are highly encouraged.

Course Outline

Day 1: Introduction to Modeling and Verification with Verilog

Introduction and Overview

- Course Objectives
- Verilog-Based ASIC design flow
- Verilog-Based FPGA Design Flow
- Benefits of HDL-Based Design

Getting Started with Verilog

- Characteristics of HDLs
- Lexical conventions
- Representation of numbers
- Identifiers
- Scalar and vector signals
- Verilog primitives
- Port rules for primitives
- Verilog's logic system
- Expressions and operators
- Data types (nets and registers)
- Signal contention
- Memories and strings

Structural Models of Combinational and Sequential Logic

- Design encapsulation with modules
- Module ports
- Module instantiation
- Port connections by position and name
- Unconnected ports
- Arrays of primitives and modules
- User-defined primitives
- modes (input, output, inout)

Hierarchical Decomposition and Top-down design

- Hierarchical decomposition
- Nested modules
- Design hierarchy
- Example: four-bit-slice 16-bit adder

Propagation Delay

- Delay models in Verilog
- Gate propagation delay
- Rules for propagation delays
- Inertial delay for gates
- Wire delay

Testbenches, Simulation, and Model Verification

- Importance of simulation
- Event-driven simulation
- Basic simulator organization
- Zero-delay simulation
- Top-down design
- Testbench elements
- Stimulus generator
- Response monitor
- Unit-delay simulation
- GUI-based output
- Development of a test plan

Lab Exercises

Day 2: Behavioral Modeling with Verilog

Language Constructs for Behavioral Modeling

- Behavioral models and procedural statements
- Constructs for behavioral models
- Assuagements to variables
- Cyclic and single-pass behaviors
- Timing control
- Delay (#), event (@), and wait
- Procedural (blocked) assignment
- Nonblocking assignment
- Procedural continuous assignment
- Sequential and parallel blocks
- Conditional and case statements
- Loops (for, repeat, while, forever)
- Disabled blocks

Behavioral Models of Combinational Logic

- Implicit combinational logic
- Three-state bus model
- Level-sensitive behavior
- Cyclic behavior (always block)
- RTL models
- Algorithm-based models

Behavioral Models of Sequential Logic

- Models of level-sensitive sequential logic
- Models of edge-sensitive sequential logic
- Synchronous and asynchronous reset
- Concurrent behaviors
- Indeterminism, concurrent behaviors, and race conditions
- Nonblocking assignments and concurrent RTL models
- Nonblocking vs. blocking assignments

Additional Topics for Testbenches

- Modeling with single-pass behaviors
- Named events
- Black holes for simulators

For more information, contact:

Tom Wille • tw@tm-associates.com
TM Associates, Inc. • 503-656-4457
www.tm-associates.com

- Race conditions
- Intra-assignment delay
- Model portability and reuse
- Inline and indirect parameter redefinition

Tasks and Functions

- Examples and rules
- Overview of system tasks

Verilog 2001

- ANSI C style changes
- Initialization of variables
- Re-entrant tasks
- Recursive tasks
- Constant functions
- Variable part selects
- Signed data types, ports, literal integers, functions
- Arithmetic shift operator, Exponentiation operator
- Sensitivity list for event control
- Sensitivity list for combinational logic
- Parameters
- Instance generation

Lab Exercises