

## Course Description

This course offers an introduction to the syntax of the Verilog Hardware Description Language (HDL). The emphasis of the course is on developing the skills to write solid synthesizable code, but also teaches the basics of simulation code in order to write viable testbenches. Structural, Register Transfer Level (RTL), and behavioral coding styles are covered. This language introduction can be used as a springboard for an intensive study of Verilog, as well as to be able to understand and modify existing code.

**Level** – Beginner

**Course Duration** – 1 day

**Price** – \$800 or 8 Training Credits

**Course Part Number** – HDT-VERPRI-100-ILT

**Who Should Attend?** – Engineers who want a thorough understanding of the principles of Verilog HDL for digital design.

**Prerequisites**

- Basic digital design knowledge

**Software Tools**

- None

**Hardware**

- None

- **Testing the Design** – Discusses the basics of Verilog testbenches. The concept of time flow is also shown, along with coding techniques for clocks and common test signals.
- **Blocking and Non-Blocking Statements** – Presents coding styles for registered and combinatorial logic to ensure that there are no mismatches between RTL simulation and the synthesized design.
- **State Machine Design** – State machines are a very common and important construct in digital designs. Understanding the correct coding style for these structures allows for the best possible optimization during synthesis.
- **Memory Structures** – Explains the inclusion of RAM/ROM memory structures in Verilog designs. A discussion of both instantiating RAM/ROM primitives and inferring them from coding styles is presented. The two styles are contrasted in terms of efficiency and portability.

## Register Today

Hardent offers courses to help designers produce fast, predictable and efficient designs. For a detailed list, visit [www.hardent.com/training](http://www.hardent.com/training) or contact Hardent's Training Coordinator for additional information, to register for a class or to schedule a private course.

**Contact:** Chris Green  
**Email:** [training@hardent.com](mailto:training@hardent.com)  
**Telephone:** 514-999-3880

## Course Outline

- **Verilog Basics** – Explores the evolution and structure of the Verilog HDL language. The ideas of RTL (synthesizable) and behavioral coding are presented.
- **Vectors in Verilog** – Verilog has specific rules for interconnecting and manipulating vectored data types. These rules are presented along with strategies to avoid common pitfalls in the interconnection.
- **Operators and Expressions** – Describes the operators available in Verilog, and the rules that govern their use in expressions.
- **Hierarchy in Verilog** – Discusses how to manage larger designs in Verilog through the use hierarchical design styles. The mechanics and strategies for implementing hierarchical designs subjects are covered.
- **Procedural Statements, Always Blocks** – Presents the use of the *always* block to describe both register structures and combinatorial logic. Also covers the concept of describing digital logic structures using procedural structures such as the *if / else* and *case* structures.
- **Shifting and Concatenation** – Coding styles for shift registers are presented through the use of shifting and concatenation operators.