Overview of Verilog Courses

*For detailed course outlines and instructor write-ups visit our website: **www.tm-associates.com***

• **Verilog 2001: New Capabilities – 1 day**

Basic Level, 90% Lecture, 10% Pencil-and-Paper Exercises

This one-day Verilog-2001 seminar shows design, verification, and support engineers how to apply the latest IEEE Std 1364-2001 syntax, in the context of real-world SOC development. It's also an effective way to update the skill set of engineers who've picked up Verilog "on-the-job," with little or no actual training.

Unlike many Verilog workshops which focus on dry syntax, this seminar is rich in practical design examples, insights into Verilog semantics, and tips to avoid hidden pitfalls. The eight-hour seminar is primarily lecture. To vary the pace and reinforce the material, various pencil-and-paper exercises ask the attendee to complete or debug segments of code in the workbook. Answers will be displayed on the screen. To fully utilize the time invested, this seminar does not attempt to cover every enhancement in detail, but focuses on the 80% of Verilog-2001 features most useful to today's SOC engineers.

Because Verilog is a rapidly-evolving language, the 2001-oriented material is interspersed with brief *look-ahead* references to anticipated SystemVerilog changes. For example, the material on Verilog-2001 **@(*)** sensitivity lists is followed by a glimpse at the SystemVerilog keywords ***always_comb*** and ***always_ff @(...)***.

Attendees will take back a concise appendix of Verilog-2001 constructs for later reference, as well as Coding Guidelines for synthesis and for verification.

• **Verilog 2001 Introduction - 2 days**

Basic Level, 70% Lecture, 30% Labs

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 1364-201. 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 synthesis and verification.

• **Verilog 2001 for Hardware Designers - 4 days**

Basic Level, 50% Lecture, 50% Lab

The Verilog hardware description language plays a key role in design flows for ASICs and FPGAs. Yet many designers lack skill with this language. This course provides a comprehensive presentation of the main features of the Verilog language and the extensions introduced by Verilog 2001 (IEEE Std. 1364-2001). The course will bridge the gap between the designer's prior background in digital logic (schematics, Boolean equations, truth tables, state transition graphs, ASM charts) and related Verilog constructs used in modern design flows. Several examples (e.g., FIFO, three-state bus, synchronization across clock domains) will be

used to illustrate language features, including an introduction to correct modeling styles for synthesis and verification. A comprehensive lab on the final day will reinforce the material by using a real-world design.

- **Advanced Verilog 2001 Coding Styles - 3 days**

For Synthesis & Verification
Advanced Level, 50% Lecture, 50% Lab
Staying competitive in today's ASIC/FPGA market means designing ICs with greater functionality, higher speed, and lower cost. Effective Verilog-2001 coding techniques can make all the difference between designs that meet tough synthesis targets and verification schedules, versus those requiring re-spins. The goal of this hands-on workshop is to enhance your mastery of Verilog language features which drive the synthesis tool and maximize verification productivity. Most of the material is tool-neutral, although case studies include results for popular ASIC/FPGA tools. Each key principle or coding insight is shown in the context of a realistic design situation. You'll be exposed to a wireless-telephony chip design, both in lectures and labs. Digital video and graphics applications are also explored. This course can be customized or condensed to meet the specific needs of your design team or available schedule.

- **Verilog Verification Methodology – 3 days**

Intermediate Level, 50% Lecture, 50% Lab
Verification now takes 70% of the design cycle. To stay competitive in today's ASIC/FPGA market, you need to use the latest techniques to verify ICs. Creating a reusable test environment will allow you to create test cases and checkers quickly, and help you meet your project deadline. The goal of this hands-on class is to enhance your mastery of Verilog language, and use it to verify today's complex designs. Most of the material is tool-neutral, and verification techniques are limited in scope to the Verilog language itself-- no extra tools or languages to learn! This course can be customized or condensed to meet the specific needs of your design team or available schedule.

- **Verilog for Experienced VHDL Designers – 3 days**

Intermediate/Advanced Level, 50% Lecture, 50% Lab,
The learning curve for Verilog is not as steep as for VHDL, but it is longer. As a less restrictive language, Verilog has many quirks, idiosyncrasies, and hidden pitfalls. This workshop covers enough basic material initially to alert users to some of the well-known pitfalls and to understand the language's idiosyncrasies. The workshop then accelerates into more advanced topics like coding to enhance speed and reduce area. Several larger case studies are included to highlight principles of coding for synthesis. Previews of SystemVerilog features are included, wherever significant future enhancements are anticipated (e.g. C-like structs to replace the need for faking records in Verilog).