# ***SystemVerilog for Verification Professionals***

## 3 Days, Advanced Level

***Note:*** *This course was developed and is taught by XtremeEDA, a company which specializes in verification consulting and training.*

### Overview

In the semiconductor industry, faster time-to-market is the key to gaining a strong competitive advantage in the marketplace. Having a leading edge verification methodology is vital to achieving this. Our newest course offering *SystemVerilog for Verification Professionals* provides everything that you, the verification professional, need to create advanced constrained-random coverage-driven self-checking testbenches in the SystemVerilog language. From data types to program blocks and interfaces, to interprocess synchronization and communication, you will learn everything you need to create advanced SystemVerilog verification environments. This course details the advanced verification features of the language, focusing on constrained randomization, assertions, functional coverage, and other tools that facilitate an effective reusable verification environment. Throughout the course an example design is taken through the entire verification process, implementing a state-of-the-art, coverage-driven, constrained-random, assertion-based verification environment.

### Intended Audience

This course is designed for Hardware Verification Professionals who are interested in furthering their knowledge of verification.

### Prerequisites

A good grasp of Verilog is required.

## Course Outline:

### Day 1

**Verification Methodology Overview**

- Traditional verification approaches
- Issues with tradition verification approaches
- The SystemVerilog solution
- Introduction to Coverage Driven Verification (CDV) methodology

**SystemVerilog Syntax**

- Data types
- Arrays and queues
- Data declarations
- Operators and expressions

- New SystemVerilog statements
- Tasks and functions
- Parameterized
- Virtual

## Day 2

**Object Oriented Programming (OOP)**
- OOP methodology overview
- Classes
- Properties and methods
- Constructors
- Data hiding
- Encapsulation
- Polymorphism
- Inheritance

**Constrained-Random Generation**
- Random variables
- The randomize() function
- Defining constraints
- Random number generation
- Random stability
- Random sequences

**Driving and Receiving Data**
- Clocking blocks
- Program blocks
- Concurrency and synchronization
- Interfaces
- Modports
- Tasks/functions

## Day 3

**Intelligent Self Checking**
- Checking Methodology
- SystemVerilog Assertions (SVA)
- Immediate
- Sequences
- Properties
- Assertion coverage and control
- Data Checking

**Functional Coverage**
- Coverage methodology
- Different forms of coverage
- Defining coverage groups
- Defining coverage bins
- Transition coverage
- Cross coverage
- Coverage options
- Architecting coverage

**Miscellaneous Features**
- File I/O
- New messaging tasks
- Direct Programming Interface (DPI)

**Laboratory Exercises:**
Laboratory exercises (10 in total) will be performed by the student throughout the duration of the course.

### For more information, contact:

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