Sunburst Design

World Class SystemVerilog & UVM Training

# 13 Reasons Why UVM and OVM are Hard to Learn

**Clifford E. Cummings**
Sunburst Design, Inc.
cliffc@sunburst-design.com
www.sunburst-design.com

Many engineers believe they can learn OVM/UVM by picking up and reading a book and the OVM/UVM User Guide. They quickly discover this is exceptionally difficult to do. Why is it so hard to learn UVM from existing materials?

Through years of experience, Sunburst Design has identified the following reasons why engineers struggle with existing OVM/UVM tutorial materials:

1) The UVM User Guide was written by Cadence and teaches Cadence recommended methods, which includes the use of a large number of UVM macros.

2) The UVM tutorials on VerificationAcademy.org are shown using Siemens / Mentor recommended methods, which includes the use of fewer UVM macros and more UVM method calls.

3) The OVM Cookbook was written by Mentor employees and is based on an earlier version of OVM (the latest techniques are not shown in the book).

4) The above User Guide, tutorials and Cookbook do not acknowledge or explain the alternate methods, so users are left to draw erroneous conclusions that some of the methods shown are flawed, which is not true. Learners need to be taught the pros and cons of the alternate methods so that they understand why there are differences in the various methods presented.

5) All the people who have written UVM materials are ***really, really*** smart software engineers who assume that engineers already understand SystemVerilog syntax and semantics, object oriented programming and polymorphism semantics, and they don't know how teach these concepts to beginners.

6) Many of those who have written UVM materials are software engineers who do not have a strong grasp of good hardware design practices, and it shows in many of the diagrams and examples.

7) The UVM User Guide (chapter 2) and the OVM Cookbook (chapter 3) introduce Transaction Level Modeling (TLM) concepts, including **`put`**, **`get`** and **`transport`** communication, but do a poor job of tying the concepts into the rest of the UVM materials. Engineers often wonder why TLM was introduced in these texts. It is better to learn and use UVM with minimal initial TLM understanding, then learn TLM details on the last day of training to explain how TLM was used and why it works.

8) Most UVM materials show the driver on the right and the monitor on the left (right to left data-flow inside of the agent). This contradicts known good hardware block diagramming methods (data should flow from left to right in block diagrams) and adds an unnecessary level of confusion to the learning process for those who are familiar with good block diagramming techniques.

9) There is a huge shortage of complete simple examples. Most of the publicly available example code is in abbreviated code-snippet form, leaving the new user to guess what is missing. Finding full examples in the materials is rare. One notable example shows OVM used on a large VHDL design, which introduces yet another unknown to the learning process.

10) Of course, you must understand classes, class-extension, virtual classes, virtual methods, dynamic casting, polymorphism, randomization, constraints, covergroups, coverpoints, interfaces and virtual interfaces before you can completely understand a UVM testbench. Too many engineers try to learn UVM without a full understanding of these SystemVerilog fundamentals (this is not the fault of UVM authors).

11) Classes are applied as stimulus and sampled for verification. Existing materials do not explain why classes are used instead of structs?

12) Interfaces, virtual interfaces and their recommend usage-models are somewhat buried in the materials and are poorly explained (most authors assume you understand these concepts without much explanation - they are wrong).

13) There are a significant number of typos and mistakes sprinkled throughout the materials and examples. The mistakes force the student to try to figure out which coding styles are correct and which have typos.

Sunburst Design OVM/UVM training addresses each of these issues. Cliff Cummings periodically adds new papers to the sunburst-design.com web page to show Best Known Methods using different UVM features and capabilities.

Cliff Cummings and his instructors offer Open Enrollment UVM Verification training on a regular basis and conduct onsite training for customers upon request. Please contact Cliff for more information (email: cliffc@sunburst-design.com)