Sunburst Design

World Class SystemVerilog & UVM Training

Why Use Classes to Represent UVM Transactions?

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

Why Use Classes?

When I learned SystemVerilog class-based verification techniques, the first unanswered questions that I had were, "Why use classes to represent transactions? Why not use structs?"

I was clearly not the only engineer that had this question, and this is still a Frequently Asked Question (FAQ) in my training classes. This Cliff-note will answer that question.

Classes -vs- Structs

To understand the reason for using classes to represent transactions, the following side-by-side comparison is what I needed to see, and should prove useful to the reader. This information is also outlined in my paper, "UVM Transaction - Definitions, Methods and Usage" but since that paper is 72 pages long (who writes 72-page papers on ... data?!), the comparison is shown below:

Classes	**Structs**
✓ Can have multiple fields.	✓ Can have multiple fields.
✓ Can have randomizable fields.	✘ CANNOT have automatically randomizable fields.
✓ Can include randomization constraints.	✘ CANNOT include randomization constraints.
✓ Have a built-in randomize() method.	✘ DO NOT HAVE a built-in randomize() method.
✓ Can include important user-defined methods (such as copy, compare and print-my-contents).	✘ CANNOT include user-defined methods.
✓ Are a dynamic type that can be generated on demand.	✘ Are a static type that requires all user-defined structs to be declared at the beginning of the simulation.
✓ Class types can be extended to easily create a new version with added fields and methods.	✘ New versions of structs require a user to completely copy the original into a new struct and then add new fields.
✓ Classes can be put into a UVM factory for easy runtime substitution.	✘ Non-dynamic structs cannot be put into a UVM factory.

Classes are basically dynamic, ultra-flexible structs that can be easily randomized, easily control the randomization, and be created whenever they are needed. Classes have the multiple field encapsulation capability that exists in structs, plus so much more. That is why classes are the preferred structure to represent testbench transactions.

References

Clifford E. Cummings, "UVM Transactions - Definitions, Methods and Usage," SNUG-SV 2014 - www.sunburst-design.com/papers/CummingsSNUG2014SV_UVM_Transactions.pdf

AUTHOR & CONTACT INFORMATION

Cliff Cummings , President of Sunburst Design, Inc., is an independent EDA consultant and trainer with 32 years of ASIC, FPGA and system design experience and 23 years of SystemVerilog, synthesis and methodology training experience.

Sunburst Design, Inc. offers World Class SystemVerilog & UVM training courses. For more information, visit the www.sunburst-design.com web site.
Email address: cliffc@sunburst-design.com