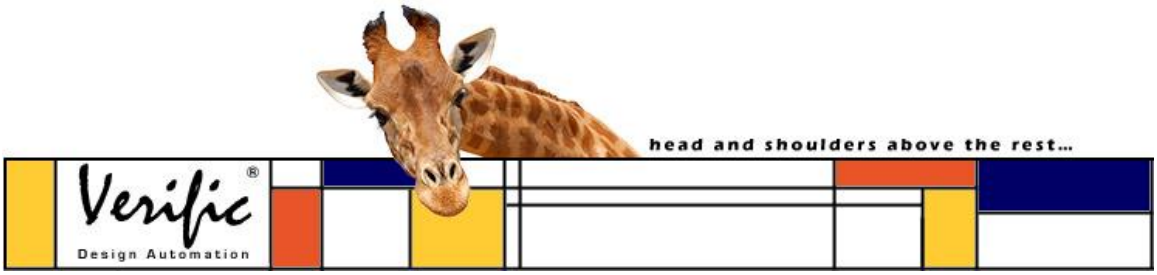

SystemVerilog Leaps From 2005 To 2009, And Beyond

In December 2004, I enthusiastically predicted that 2005 would be the year where the electronic design automation (EDA) industry would see introductions of several SystemVerilog-based design tools. Of course, I had a self serving interest there. My employer, Verific Design Automation, had put significant resources in development of a SystemVerilog parser, analyzer and elaborator, which we hoped would soon be adopted by many. Fortunately, none of the editors ever examine previous years' predictions before they invite new ones. Although 2005 turned indeed out to be the year of SystemVerilog, it was not because of EDA tool introductions.

Well, here we are, five years later. The good news is that our SystemVerilog parser has been adopted by many, but, it did not happen overnight in 2005, and we are still working on it. Our first released SystemVerilog parser was based on the Accellera 3.1 standard, with some diversion to 3.1a for assertions. 2005 became the year of SystemVerilog because of the ratification of the IEEE 1800-2005 standard. It cleaned up many of the 3.1 / 3.1a Language Reference Manual quirks and paved the way for much broader adoption in the years to come.

Let's fast forward from 2005 to 2010 and take SystemVerilog's temperature today. Indeed, most EDA companies and HDL-based EDA products support SystemVerilog. The larger EDA companies developed their own as they had already started well before Verific came out with its solution, and many others obtained support through licensing from software IP providers such as Verific.

But, as Rob Dekker, Verific's founder, pointed out in a widely read contribution aptly named "SystemVerilog and the Triangle of Truth," not all SystemVerilog front-ends are created equal. Comparing the three front-runners in SystemVerilog (Synopsys' VCS, Mentor's Questa and Verific), he found that in only 73% of the test cases these three parsers wholeheartedly agreed. This is not good news for SystemVerilog designers because it means they cannot easily try their RTL designs on a different tool or flow. Verilog-2001 users have it easier. Follow-up research showed that the very same three front-ends handled 98% of the test cases in an identical way.

It would be sweet if we could get to that same interoperability percentage for the IEEE 1800-2005, preferably on short notice, but there are some obstacles ahead. The SystemVerilog working group worked hard in the past four years on improving the language. While cleaning up and adding explanation to some of the lesser defined areas in the LRM, the language also got expanded with new constructs such as '"checkers," unwittingly introducing new avenues for misalignment between EDA tools.

First published on www.gabeoneda.com on Jan 21, 2010