

**SystemVerilog: A Panacea for Chip
Verification?**

By Shailesh Dave





SystemVerilog: A Panacea for Chip Verification?

SystemVerilog, a language which is an optimum blend for design and verification, is steadily gaining ground. An emerging language, it is contesting with HVLs which have widespread use in the verification industry. SystemVerilog's roots go back to the robust and stable Verilog. With features and requirements of a verification language now built into it, there is a lot more than meets the eye in this new 'avtaar' of Verilog.

Before the invasion of HVLs into the verification domain of the chip industry, it was the verilog/VHDL and C combination that verification engineers used to verify complex chips. After the onset of HVLs, all the facilities a verification engineer could dream of were being served on a platter. The languages picked up and verification started moving the HVL way.

However, even HVLs raised some concerns such as,

- How tested legacy verification models & testbenches developed in Verilog are re-used?
- How does one adapt to switching the domain from HDL to HVL for verification and switching back to HDL for designing or bug-fixing?

SystemVerilog is emerging as the answer to all the above questions. It enables re-use of the legacy code and keeps engineers in the same domain while working on verification or design. Also, SystemVerilog being an evolution of Verilog with a verification subset, it comes with an advantage of reduced learning curve for designers who can still use the same constructs.

For verification engineers, using HVLs, all the facilities of constrained random environment and functional coverage, process control came built into SystemVerilog so ease for them too which means lot more acceptability by the actual users of the language. Being a superset of Verilog also means that it will be natively compiled into simulator.



SystemVerilog Benefits for Verification:

SystemVerilog (SV) extends wide-spread support in order to enable enhanced and effective verification.

- It supports a variety of data structures
- SV's *Data Hiding* feature allows users to create re-usable and private models
- SV supports a variety of constructs for randomization which are useful for generating random and directed random stimuli
- The language has a rich set of functional coverage constructs, defined in terms of coverage groups
- SV has clocking blocks in the interface to easily synchronize interface signals
- It has a rich set of assertions spanning from functional to formal assertions

Compatibility with Verilog:

DPI – Direct programming interface introduced in SystemVerilog is simple implement. It seamlessly blends with Verilog. With this feature, verification engineers can extensively use C also, wherever required. SystemVerilog extends the verification aspects of Verilog and can now be rightly called a HDVL (hardware design and verification language).

Being natively compiled, it reduces lot of load on the simulators as well. All in all, it is a well-packed language with all that is required by a design and verification engineer to conveniently work for both domains with just one language.