

What is SystemVerilog?

By Nir Hamzani, Sital Technology.

What is SystemVerilog?

SystemVerilog is the most transformative technology in EDA since the birth of logic synthesis. It is an extension of the popular Verilog language, bringing a higher level of abstraction to design and verification.

SystemVerilog provides a complete verification environment, employing **Constraint Random Generation, Assertion Based Verification and Coverage Driven Verification**. These methods improve dramatically the verification process.

SystemVerilog also provides enhanced hardware-modeling features, which improve the RTL design productivity and simplify the design process.

Who should use SystemVerilog?

SystemVerilog is extremely scalable. It can be used by both HW designers and Verification engineers; SystemVerilog can be used as part of your Verilog or VHDL Test Bench (TB) environment, as part of your design or as an external module. For example, a designer can enjoy the power of the Random Generation by simply including some of the SystemVerilog Random features in the current TB with no need to change the entire verification environment.

In the same way the verification engineer may include some assertions to verify correctness of a bus or a protocol as part of the existing full chip simulation.

How to use SystemVerilog?

SystemVerilog is an integrated part of the Simulation tool. There is no need for any external tool, GUI or interface (such as PLI) in order to run it. There is no need for any new compilation commands. **Therefore, the adoption process of SystemVerilog is very fast causing designers to find it very friendly and easy to use.**

Among the benefits of this unified environment is the overall performance. Running Verification using SystemVerilog is at least x2 faster than with any other verification languages, which requires external interface to the simulation tool.

What are the advantages of SystemVerilog Vs. other Verification Languages?

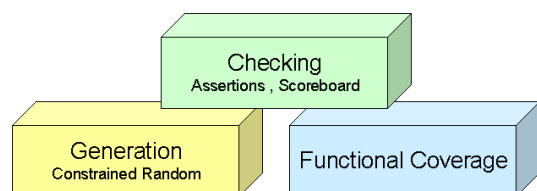
SystemVerilog has several key advantages that involve standardization, performance and adoption by engineers.

- SystemVerilog is a unified language for hardware design, specification, and verification that was approved as an IEEE Standard in 2005 to address the growing complexity of electronic system and semiconductor designs. This ensures a wide embracing and support by multiple vendors of EDA tools and verification IP's, as well as interoperability between different tools and vendors.
- Since SystemVerilog is an extension of the popular Verilog language, the adoption process of SystemVerilog by engineers is easier and straightforward. SystemVerilog enables engineers to adopt a modular approach for integrating new modules into any existing code. As a result, the risks and costs of adopting a new verification language are reduced.
- Being an integrated part of the simulation engine, eliminates the need for external verification tools and interfaces, and thus ensures optimal performance (running at least x2 faster than with any other verification languages).

SystemVerilog verification methodology

The SystemVerilog verification methodology relies on 3 building blocks, which can be used separately, or all together:

- **Stimuli** the design using automatically generated random scenarios - *constrained-random (CR)* test generation.
- **Check** the behavior of the design (*assertions*) and the output data (*scoreboard*) to verify correctness of operation.
- **Measure** the *functional coverage* metrics to provide feedback to the generation and analyze progress of verification.

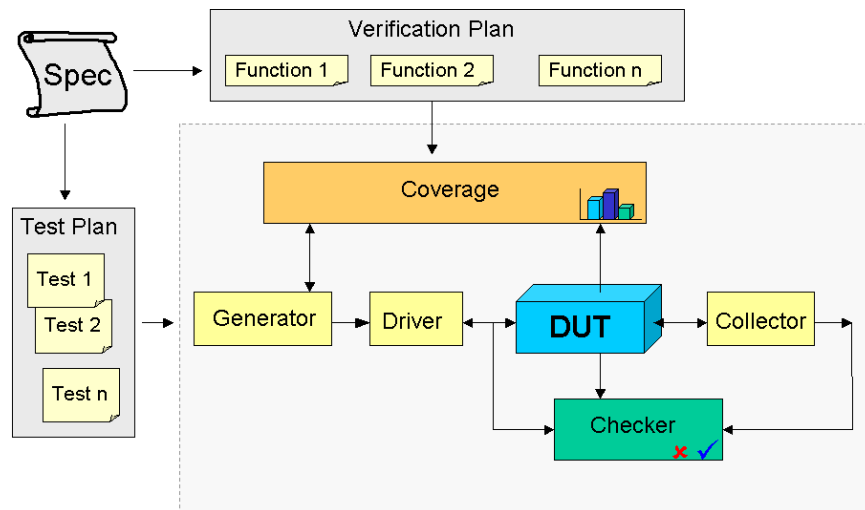


Applying the above methods that include iterative process of simulation, debug and coverage measurement is known as **Coverage Driven Verification**.

The goal of a Coverage Driven Verification methodology is to maximize verification effectiveness and at the same time, minimize redundancy and ineffective verification work.

The SystemVerilog language contains commands and constructs to support CDV including commands for constraint-random generation, assertions and functional coverage measurement.

Measuring coverage is a critical component of the CDV methodology. The SystemVerilog coverage measurement commands (like covergroup and coverpoint) provide feedback regarding the completeness of the verification process and guide the next steps of verification so that effort and resources are applied to improve overall verification.



What is the SystemVerilog's contribution to the RTL design?

SystemVerilog brings a higher level of abstraction to the Verilog designer. Constructs and commands like Interfaces, new Data types (logic, int), Enumerated types, Arrays, Hardware-specific always (always_ff, always_comb) and others allow modeling of RTL designs easily, and with less coding.

SystemVerilog also extends the modeling aspects of Verilog by adding a Direct Programming Interface which allows C, C++, SystemC and Verilog code to work together without the overhead of the Verilog PLI.

Is the SystemVerilog ready for use?

The answer to that question is a resounding **YES!**

Many experts from EDA companies and the Verilog user community that participated in the development of the SystemVerilog standard have put lots of efforts in bringing this useful language to the market. SystemVerilog is unifying the best of their learning from many different languages into a practical, easy-to-use solution that everyone can rally around.