

Advanced functional verification and debug of Serial-ATA based designs

By Shaw Yang

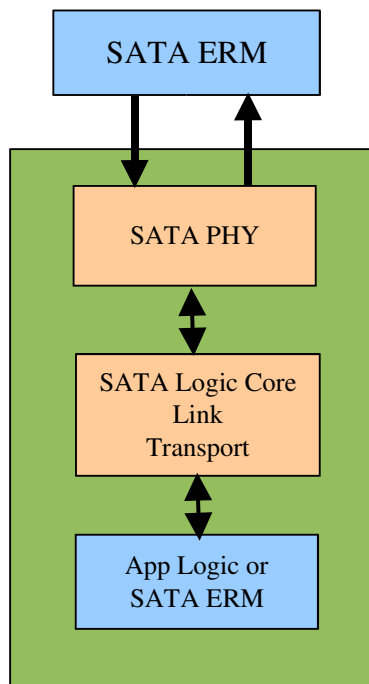
Verification Intellectual Property (VIP) streamlines the path to compliance sign-off through a reusable, layered verification methodology supported by highly configurable and feature-rich transactors, protocol assertions, advanced debug support, and comprehensive functional and compliance checklist test suites. This white paper considers the role of VIP for core- and chip-level verification of Serial-ATA based designs to ensure robust device-level and interface compliance.

The challenge

Designing with today's high-speed serial interfaces such as Serial-ATA presents a more significant functional verification challenge than its predecessors, conventional Parallel ATA. Systems and peripherals using Serial-ATA interfaces can take advantage of the 1.5Gbps and 3.0Gbps transfer rates, native command queuing, hot plug, access control services, security mode, active state power management, and advanced error reporting as specified in ATA/ATAPI-6 and SATA revision 2.5 specifications. However, these advanced features come at the expense of a more complex interface protocol, application logic, and software driver design.

This has given rise to a strong IP core market for Serial-ATA. Utilizing SATA logic core and PHY core design IP can significantly reduce the complexity of adding SATA support to chips and systems. This shifts the burden to the IP supplier for ensuring a rigorous verification of all the various core configurations before being considered trusted in any end-user design. Still, many of the advanced protocol features are substantially controlled through application logic and require custom tests to properly exercise the design functions.

The functional verification of Serial-ATA logic cores and the chips and systems utilizing them requires significant investment to develop and maintain a layered test bench environment comprised of transactors, assertions, test suites, and debug methods to isolate design bugs in different protocol layers. The test bench should be capable of core-through-chip verification including full control over the Device Under Verification's (DUT's) application logic interface as illustrated in Figure 1. Furthermore, the functional verification environment must constantly evolve to keep up-to-date with the latest specification errata and revisions. For example, since its initial release in 2003, the Serial-ATA standard has been revised several times, and major enhancements currently in the works signify that more changes will be made.



Verification framework

A designer can choose from the following two options when performing the verification:

1. Develop their own test bench that will have its own inherent risks
2. Utilize an off-the-shelf test suite that has been proven through commercial use to provide the necessary verification coverage

Designers can then execute a comprehensive compliance validation test suite that exceeds basic compliance workshop requirements and supports advanced autodebugging methods that isolate design bugs more effectively using a reference model and extensive protocol checkers.

A good verification solution should have a set of Executable Reference Models (ERMs) and test suites that simulate the behavior of any type of SATA components and links. The ERMs should be developed and presented as behavioral Verilog HDL source code models and augmented with a rich set of protocol assertions to provide the best verification. An ERM is between two to three orders of magnitude more efficient than implementation RTL in simulation.

Figure 1 SATA core-level and full device-level

DUTs should be verified against all realistic system topologies. All types of Transport Layer Frame Information Structure (FIS) and Link Layer primitives should be generated. Robust controls over ERM operation at all protocol layers and back-end completion

generation are needed. Application-level features such as DMA must be supported.

The objective of the models is to aid in the functional verification process prior to silicon or board fabrication. A test environment supporting Verilog, VHDL, SystemC, ANSI C/C++, Vera, SystemVerilog, Specman, and other programming environments can provide a native high-level API to interact with behavioral models supporting Host and Device controllers.

A monitoring model that passively monitors and reports ATA and SATA protocol violations; validates end-to-end transactions; and measures and reports transaction trace analysis of devices by link bandwidth, latency, FIS type, Link primitive, and command types utilized helps in the debugging process. A test environment that includes a full suite of compliance test scenarios that verify Host and Device designs comply fully with the ATA/ATAP-6 and SATA specifications gives the maximum test coverage.

Compliance tests can be used to assist in the verification of each SATA component type and design. Portable tests with parameterized test sequence libraries supporting a rich set of transaction sequences for each of the SATA protocol layers and major functions best exercises the DUT. As shown in Figure 2, it is desirable to have test sequences that can be combined under directed or random modes to create a complete set of tests to verify a design against the Avery developed specification compliance checklists. End users typically reuse test sequences to create custom device-level tests that meet their specific test needs.

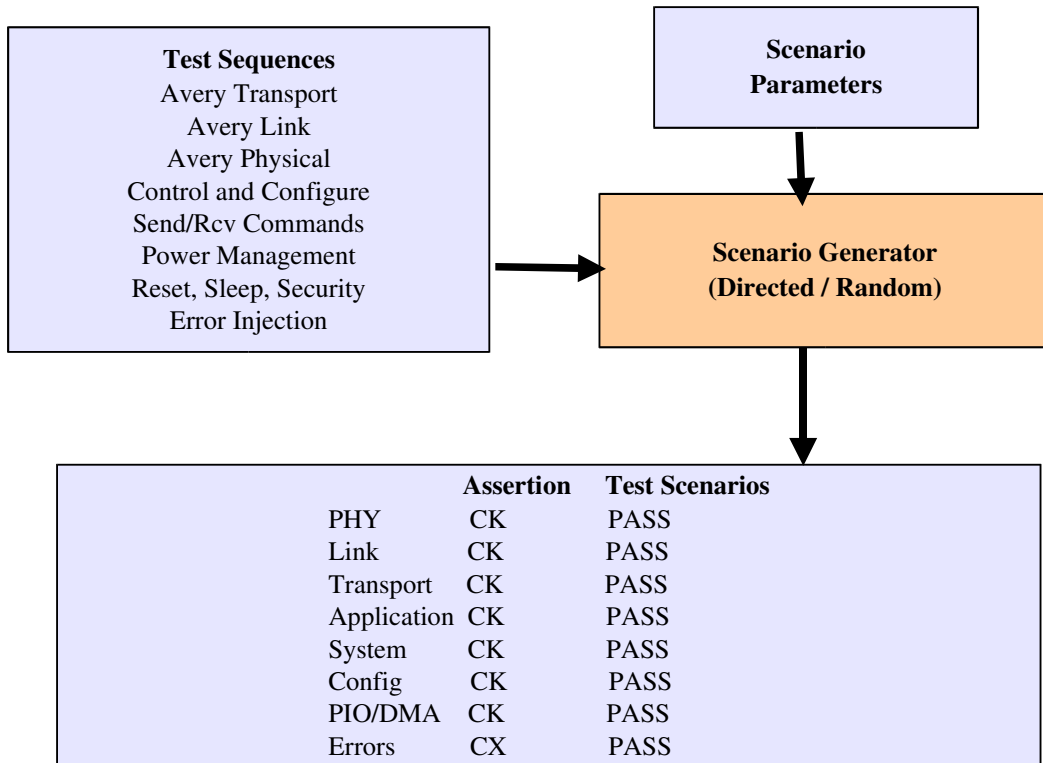


Figure 2 Scenario generation linked with compliance coverage

Bug detection and isolation methods

In response to these challenges, Avery Design Systems developed a complete verification framework for Serial-ATA that enables creating a highly accurate system model of a DUT in its system context.

An advancement in Serial-ATA verification over traditional verification methods, Avery’s SATA_Xactor coverification has an ERM and innovative autodebugging methods for bug detection and isolation. Using coverification the DUT and ERM run concurrently, applying the compliance and systems tests to both models. Transaction and sequential consistency is verified using preconfigured design match points, which track transaction flow between protocol layers of the DUT and ERM (see Figure 3). Architectural visible state and transactions are analyzed applying relaxed time, ordering, and content rules defined by the SATA protocol to ensure meaningful sequential consistency checking.

When a mismatch occurs, autodebugging is then used to perform causal analysis of the implementation model and ERM. Autodebugging utilizes enhanced behavior traversal and transaction views added to advanced behavioral debugging systems such as Novas’ Verdi for better visualization of the behavior of the DUT and shadow ERM. Coverification is also especially useful in the context of random testing where expected device operation is too complex to predict or when assertions are too complex to write. Here, match points verify the architectural state of the models on-the-fly under random input sequences.

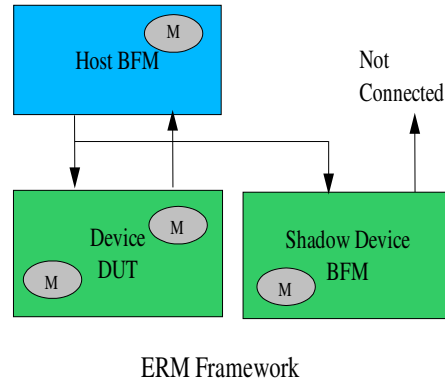


Figure 3 Verification of DUT using ERM

Some example of match points include:

- End to end checking of FIS data at the Link layer from the transmitting Host to the receiving Device on the other side (Host with Device)
- Link Layer FSM State Register between DUT Device and Shadow Device (Device with Shadow Device)
- Configuration and Status registers between DUT Device and Shadow Device

For example, Avery's design match capability isolated an incorrect link layer state transition in power management in a DUT under random test sequences.

Embracing verification

Today, many systems companies and Serial-ATA logic core IP vendors are using complete verification frameworks such as SATA_Xactor. Systems companies utilize core-level verification of logic and PHY cores in addition to complete device-level verification.