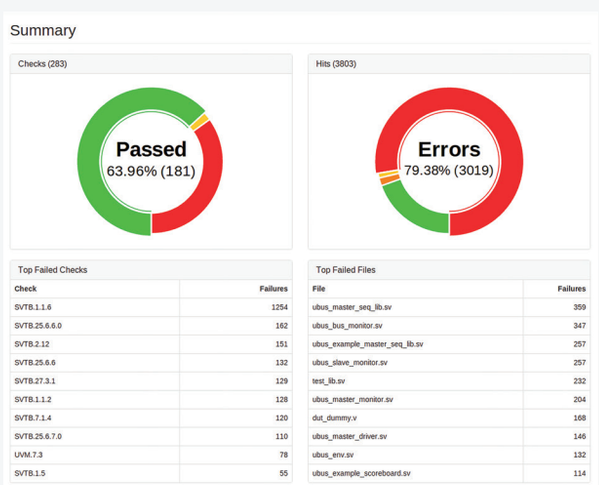
VERISSIMO

SystemVerilog Linter

Thorough audit of your designs and testbenches



BENEFITS

- Improves design and testbench code quality and reliability
- Prevents incorrect functionality and performance issues
- Automates coding guidelines checking, including UVM compliance
- Simplifies code maintenance
- Identifies dead code and copy-and-paste code
- Accelerates language and methodology learning
- Ensures best coding practices are followed

OVERVIEW

SystemVerilog provides powerful constructs and a high level of programming flexibility, at the same time introducing new challenges in code development.

For example, the ability to implement the same functionality in multiple ways may impact the simulation performance or lead to unexpected behavior.

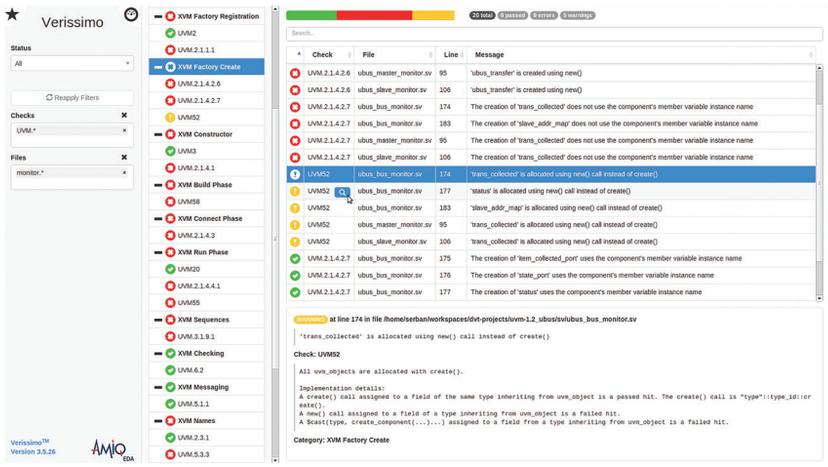
A SystemVerilog compiler checks whether the source code follows the Language Reference Manual (LRM) rules and as such, it flags only language-specific syntactic and semantic errors. However, the absence of compilation errors does not give any indication of code reliability and maintainability. Nor does it imply that best coding practices have been implemented and compliance with recommended methodologies has been met.

Verissimo SystemVerilog Linter is a coding guideline and verification methodology compliance checker that enables engineers to perform a thorough audit of their designs and testbenches. With this tool, users can check for language pitfalls, semantic and style issues, and compliance with the appropriate methodologies. Verissimo can be customized to check specific group or corporate coding guidelines to ensure consistency and best practices in code developing. For many types of check violations, Verissimo offers intelligent autocorrect suggestions to make it quick and easy for users to resolve issues.

TYPES OF CHECKS

Verissimo performs a thorough static analysis of the source code. It checks the following areas:

- Suspicious language usage such as non-standard syntax, problematic delta cycle usage, and prohibited system calls
- Semantic issues that are not caught by the SystemVerilog compiler, for example, an overridden non-virtual method, which will likely result in unexpected behavior
- Improper styling such as confusing declaration order and naming conventions
- Verification methodology violations such as inappropriate object creation, missing calls, and constructs that should be avoided
- Unused code elements such as variables that are never read or written, or functions that are never called
- Performance issues like not passing arrays by reference and thereby creating useless copies
- Copy-and-paste code duplication
- SystemVerilog and Universal Verification Methodology (UVM) compliance



Verissimo HTML Report

Customizing the Checks

Users can select from the hundreds of built-in checks in the Verissimo library and enable only those that correspond to their specific requirements. Users can customize existing rules by tuning their parameters or create new rules by using a dedicated Java application programming interface (API) that provides access to the linter's internal database.

Auto-Correcting

Verissimo can automatically correct violations found for certain classes of coding rules and users can control which rule violations are auto-corrected. This capability speeds up development and reduces code maintenance costs.

Comparing with Baseline

Users can run Verissimo and generate a baseline report, run again after adding new code or fixing errors, and ignore any violations common to the two runs. Intelligent filters enable viewing new violations, confirming violation fixes, and assessing the effects of any changes to the coding rules. This minimizes "noise" and increases productivity, especially when working with legacy code.

Integrating with DVT IDE

Verissimo runs in batch mode, as a standalone tool, or in GUI mode in Eclipse or Visual Studio Code. In GUI mode, users can perform linting and then visualize the results in the DVT Integrated Development Environment (IDE), which offers an effective way to read and understand errors and warning messages. Users can quickly jump to the problematic source line to fix any issue flagged by Verissimo. Then, all they have to do is reapply one or more rules in order to validate the fix.

Users can explicitly run Verissimo only on the current file or enable incremental linting. By enabling incremental linting, a customizable set of linting checks is automatically applied after each code modification.

Generating Reports

Verissimo features a report generator that can be used to save the results of a linting session as a text or HTML file. The HTML report includes a dashboard that shows an overview of the linting status, including information such as pass/fail percentage, top failed checks, and effort estimation for fixing failures. The report also comes with advanced functionality for searching and filtering failures. For example, author or file filtering can be used to focus the failure analysis on a specific code section.

Customizing Reports

Verissimo provides an API to generate custom reports. Users can dump the linting results in CSV or JSON formats to match a specific continuous integration (CI) engine requirement or they can prepare a small summary report that is automatically emailed to the entire team.

Merging Reports

Users can merge reports from multiple Verissimo runs, for example to combine the results from multiple configurations of an IP block, or to combine IP block results into a system-on-chip (SoC) overview. Users can also generate HTML progress reports that show how check results change over time. These are useful to gauge verification status and support continuous integration with code checking at every stage.

SUMMARY

Verissimo signals improper SystemVerilog language, semantics, and styling usage, as well as methodology violations. It helps improve design and verification code reliability, functionality, and maintainability. Verissimo can be customized to meet the demands of small teams while scaling up to larger verification groups and global enterprises. Ultimately, Verissimo is a tool that allows companies to implement the best coding practices in verification.

The seamless integration between Verissimo, as a code analysis tool, and DVT IDE, as a code development tool, further improves verification productivity and quality. It also contributes to decreasing the significant costs associated with code maintenance.

INTEGRATED SOLUTION

Verissimo SystemVerilog Linter is closely integrated with the other design and verification products available from AMIQ EDA, including **DVT IDE.**

TECHNICAL SUPPORT

The technical support team is available to promptly answer your questions, provide you with training, and work with you to determine your needs.

Your requirements and feedback are important. Whether you are looking for technical support or new features to improve your design and verification flow, AMIQ's technical support team strives to answer your requests in a timely manner.

CONTACT AMIQ

SUPPORT & EVALUATION
support@amiq.com

SALES
sales@amiq.com

WEBSITES
www.eda.amiq.com / www.amiq.com

VER-0125-A4