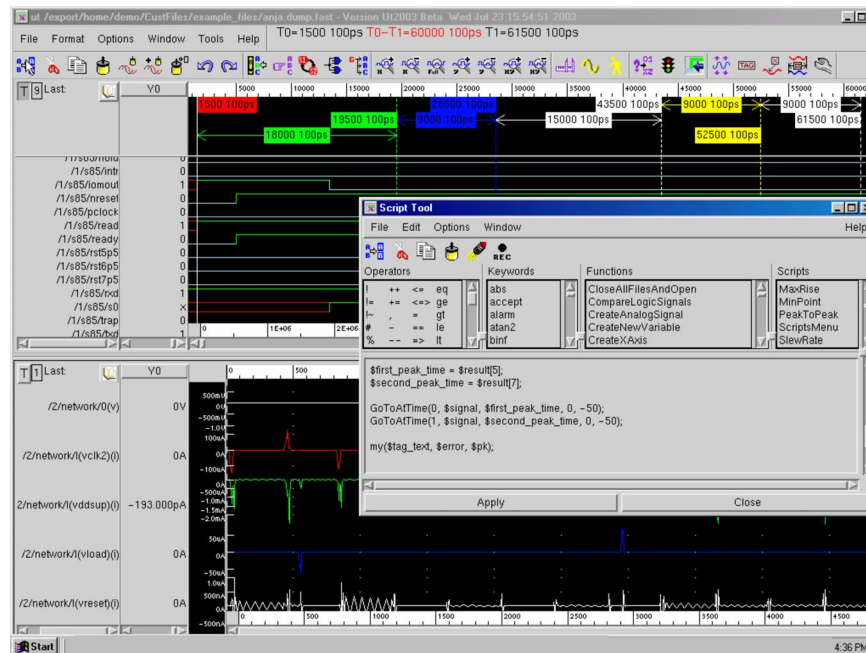


VLint: a formal predictive lint checker

VLint is an innovative predictive RTL analysis tool that simplifies and speeds up design checking using a combination of formal and structural analysis.



VLint eliminates complex design errors at all stages of your design implementation cycle from RTL to gate level, from module to full chip level.

VLint automatically checks your RTL for hundreds of rules, including the following:

Race detection

- Write-Write
- Read-Write
- Latch-Latch
- Combinational loop
- Clock gating races

Clock Domain Boundary Checks

- Clock domain boundary crossings
- Synchronization of data
- Use of specified synchronization cells

Design reuse methodology

- Coding style
- Reuse Methodology Manual (RMM)

Synthesis checks

- Simulation-synthesis mismatches
- Synthesis compatibility
- Implied latches

Testability checks

- RTL ATPG checks
- Clock gating
- Asynchronous sets/resets
- Asynchronous clocks

Net-list checks

- X-source problems
- Redundant logic
- Conflicting assignments
- Range Violation
- Non-resettable flip-flops

Download VLint from our web site: www.veritools.com

Tools for Power Users

Detect clock domain synchronization errors

VLint supports multiple clock designs. Formal clock domain boundary checking ensures that data that crosses clock domain boundaries is synchronized. VLint recognizes double register buffering, memory/register file and customized synchronization schemes. For custom schemes, you can specify allowed synchronization cells. VLint also identifies data that is generated and consumed by different edges of the same clock and checks for the appropriate use of lock-up latches.

Detect races before simulation

Race conditions are a major cause of critical design errors that can take significant amounts of time and effort to debug. VLint's powerful static analysis techniques reduce design debug time and effort by detecting potential race conditions before simulation and synthesis.

VLint identifies race conditions, such as write-write, read-write, and combinational loop races, and automatically pin-points the lines of source code that are the cause of the race conditions.

Automatically check design re-usability

VLint includes the following types of traditional built-in rules checks: Reuse Methodology Manual (RMM), coding style, design for testability (DFT), simulation, synthesis, syntax, and lint. These checks help you catch errors before you get to simulation and synthesis, and provide a way of ensuring that design teams follow a consistent design style and methodology that enables design re-use. In addition, VLint allows user-defined checks making it easy for you to add your own rules and messages to its already extensive library of built-in checks.

Enforce your design methodology

VLint's support for user re-configurable rules and rule packages allows you to customize rule packages that address your design checking requirements. Predefined packages simplify the enforcement of design methodologies such as design for re-usability, design for testability and design for verifiability. You can also add your own rules using Perl or TCL.

Find design for testability errors at RTL stage

VLint allows you to check that your RTL complies with design for testability (DFT) rules so reducing the amount of time and effort spent at the gate level finding and fixing scan DFT violations.

Eliminate design errors earlier

VLint eliminates complex design errors at all stages of your design implementation cycle from RTL to gate level, from module level to full chip. Deploy VLint early in your design cycle to drastically reduce the amount of effort you spend finding bugs using time-consuming traditional test-bench methods.

Be productive immediately

VLint's platform independent Java GUI provides an easy to use interactive display of results, allowing you to click on a message to view the line of RTL code that caused that message. VLint can also be run in batch mode from the command line.

VLint's filters allow you to focus on only the checks that are important for your design. You can exclude groups of checks or individual checks prior to running the checker or you can filter messages based on category before generating reports, so you see only what is important to you at a particular time. You can step through particular categories of errors and see the lines of code that caused them or you can step through all the errors in a particular design module. VLint's signal tracing capability allows you to trace nets through the design hierarchy to identify the source of the error.

Language Support

VLint 2.0 supports Verilog HDL.

Platforms Supported

VLint 2.0 supports Unix/Solaris, Windows NT/2000 and Linux.

Availability

VLint 2.0 is available now for Unix and Windows and Linux.



Download VLint from our web site: www.veritools.com