Blue Pearl Software
ASIC, FPGA and IP Verification

Analyze RTL

Overview

The Blue Pearl Software Visual Verification Suite provides enhanced Lint, Debug, Clock Domain Crossing (CDC) and automated SDC generation flows to accelerate RTL Verification. Analyze RTL™ provides RTL linting combined with a powerful debug environment with the industry's fastest bug find/fix rate to quickly identify critical design issues, up front, streamlining simulation and synthesis while improving overall quality of results.

Top 10 Reasons Design Teams Choose Analyze RTL

1. Ease of use – No steep learning curve to use and become proficient with the tool.
2. Focuses in on issues – Provides filtered reports, path-based schematics, and cross probing to quickly find issues and then assign waivers to fix or not.
3. Design metrics – 'Must Fix / Won't fix' waivers allows the maturity of the code to be assessed and the engineering effort to fix. This results in more accurate program management estimations as to the state of the design.
4. Easy creation of custom packages for company design rules – Automates the design review process by enabling design reviews to be consistent and focus on assigning must/won't fix waivers.
5. Design Sign off – You know required tests were actually run – good for "goods in inspection" of code as well as to understand the impact of code changes.
6. Find issues earlier in the design cycle - Enter simulation and synthesis with a better quality of code. The later issues are found the more costly they are to fix.
7. Design Scenarios – Ensure the configurations of generics does not introduce any corner cases when developing IP e.g. one generic resulting in the overflow which is not caught until much later.
8. FSM viewer – Ensure no illegal /deadlocked/unmapped states are in the FSM – simulation requires you ask the right question to find it or worse you find it after hours of simulation, which then to be done again.
9. Design Enablement – Low 'noise' text reports provide significant information the structure of the design to help optimise if necessary – they also help designers understand legacy designs and pre-existing IP blocks.
10. Built in FPGA libraries and choice of Linux and Windows streamlines setup and deployment

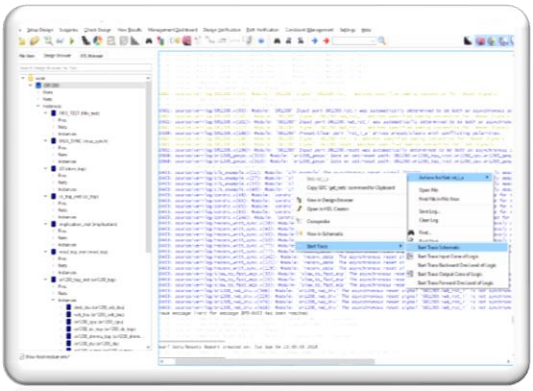
Key Benefits

Modern ASICs, SoCs and FPGAs routinely have millions of gates with memories, transceivers, third party IP and processor cores. RTL issues can be time consuming and complex to debug in the lab and through simulations. To reduce verification and debug times, designers need tools that can identify problems quickly before simulation and synthesis, and definitely before spending time in the lab.

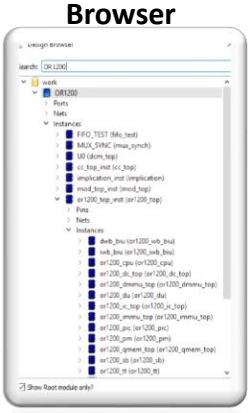
Key Features

- Fast and meaningful results with tool Setup Wizard
- IEEE Verilog/System Verilog & VHDL language specification compliance and syntax checking
- Configure checks along with standard checks | STARC, DO-254 and AMD® UltraFast™ Design Methodology
- Streamline debug; integrated RTL, schematics, and message viewer
- Easy debug message sorting, filtering and waiving to pinpoint problems
- Automated flow with Tcl-based Command Line Interface (CLI), and re-usable message waiver file

Debug Environment

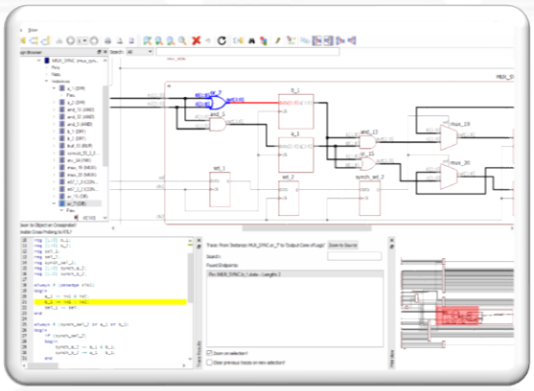


Hierarchical Design Browser

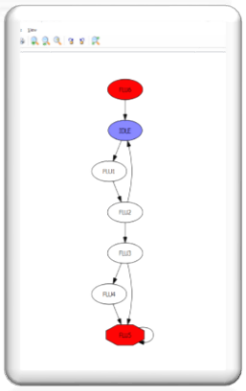


Analysis Reports

HDL Viewer



Schematic Viewer



FSM Viewer

Identify Design Issues Quickly

Analyze RTL enables users to debug design issues quickly using intelligent sorting and message filtering.

- Low Noise
- Check customization for specific design style
- Easy setup
- Waiver migration

Finite State Machine Analysis

Rather than writing exhaustive simulation test benches to validate their finite state machines (FSMs), designers can use the FSM analysis capability within Analyze RTL. With minimal effort, designers can

- Extract FSMs from their RTL
- Find dead or unreachable states
- Generate easy to read bubble diagrams to better visualize FSMs

RTL Checks for High-Speed Designs

It is important to find as early as possible RTL coding that prevents the design from getting desired speed. FPGAs, because of their more constrained fabric than ASIC, certain type of structures causes slow down. Rather than wait for synthesis or static timing analysis results, Analyze RTL users can easily identify:

- High fanout nets
- Deep nested “if-then-else” statements
- High levels of logic paths
- Reset methodology, Async/sync

Request a Demo today!