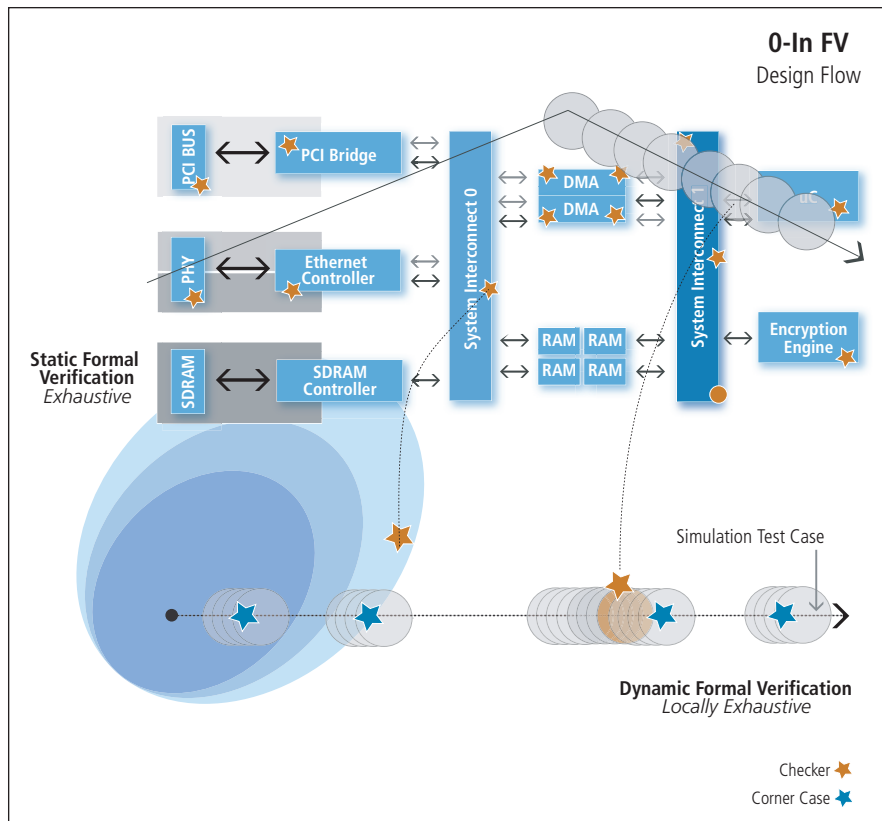


0-In Formal Verification



0-In Formal Verification combines static and dynamic formal technologies with simulation to provide the most powerful formal analysis in the industry.

The Mentor Graphics 0-In® Formal Verification solution integrates the industry's most powerful static and dynamic formal verification technologies with the industry's most comprehensive assertion-based verification (ABV) solution, enabling users to achieve functional verification closure quickly and predictably. The 0-In Formal Verification solution is easy to use and provides value all the way through the design cycle — from block-level design, where it replaces traditional block-level simulation test benches, to chip-level design, where it replaces extensive pseudo-random simulation.

Target Tough Verification Problems

The 0-In Formal Verification solution delivers the most powerful, most comprehensive, and most widely deployed property-checking engines in the industry. Users can target complex logic with exhaustive analysis to increase confidence that corner-cases have been fully explored, and they can quickly and easily find bugs missed by traditional methodologies.

Major product features:

- Incorporates the industry's most powerful, highest-capacity, formal verification technology
- Uses static formal verification to exhaustively verify block-level designs without the need for simulation
- Uses dynamic formal verification to leverage chip-level simulation to find corner-case bugs
- Supports all standard assertion formats, including Verilog, 0-In CheckerWare®, PSL, SVA, and OVL
- Supports comprehensive formal coverage metrics to guide verification
- Includes complete assertion-management system to analyze verification results, debug errors, and track progress
- Supports easy-to-use assertion-based methodology for specifying and debugging input assumptions
- Supports multiple clocks, gated clocks, memories, latches, X semantics, and non-synthesizable constructs
- Automatically reproduces assertion violations using your existing simulator

Examples of tough verification problems ideally suited for the 0-In Formal Verification solution include:

- Standard-interface protocols
- Data integrity
- Arbitration logic
- Exhaustive block-level analysis
- Elasticity buffers and flow control logic

Integrated with Easy-to-Use ABV Methodology

The 0-In Formal Verification solution is tightly integrated with 0-In's complete ABV solution — 0-In Assertion Synthesis. The designer or verification engineer can specify an assertion once and then use it without modification in simulation, formal verification, and emulation.

Use of ABV by numerous customers has resulted in faster and more thorough functional verification. The full 0-In ABV solution, including the 0-In Formal Verification solution, provides key capabilities that make the use of assertions more efficient and effective:

- Powerful set of exhaustive formal verification engines
 - Static formal analysis — providing exhaustive proofs and counterexamples
 - Dynamic formal analysis — leveraging simulation to formally analyze critical corner-cases
 - Largest capacity formal analysis engines in the industry
 - Extensive design-style support
- Full support for standard assertion languages
 - Verilog
 - Property Specification Language (PSL)
 - SystemVerilog assertions (SVA)
 - Open Verification Library (OVL)
 - 0-In CheckerWare
- Complete assertion-management system to track verification progress
 - Assertion browser
 - Assertion violation browser
 - Total coverage viewer
 - Regression analysis
- Simple use model
 - Easy-to-use constraint (input assumption) methodology using assertions and CheckerWare
 - Ready-to-use, pre-verified formal constraints for standard bus interfaces
 - Detailed coverage information spanning simulation and formal analysis
 - Ability to fine-tune formal engines for “power-users”

The 0-In Formal Verification solution allows all engineers to be productive with formal verification tools, enabling them to find bugs quicker and find bugs missed by simulation alone.

Industry's Most Advanced Formal Engines

The 0-In Formal Verification solution incorporates multiple, highly-tuned, modern, formal algorithms, including static and dynamic formal verification. The resulting formal analysis is the fastest and deepest in the industry. Static formal analysis is commonly used at the block-level to verify properties exhaustively. Dynamic formal analysis is commonly used at the chip level. Dynamic formal analysis leverages simulation by monitoring corner-case activity highlighted by assertions and 0-In CheckerWare elements embedded in the design. It then focuses exhaustive formal analysis around these corner-case states, finding design bugs that would otherwise be missed. Together, dynamic formal and static formal verification complement a directed or constrained-random simulation methodology, providing more effective verification power to engineers and reducing the time to verification closure.

The 0-In formal verification engines automatically handle the broad range of constructs that are common in today's designs, including multiple clocks, gated clocks, memories, latches, and X/Z semantics.

Best-in-Class Standards Support and Interoperability

The 0-In Formal Verification solution provides the most extensive support for standard assertion languages and libraries in the industry. The 0-In Formal Verification solution leverages 0-In Assertion Synthesis to provide full assertion interoperability. Users are free to choose their assertion format, whether it is a language such as PSL or SVA, a library of pre-verified assertions, or a combination of the two. High verification productivity is achieved by using exactly the same assertions in simulation and formal verification.

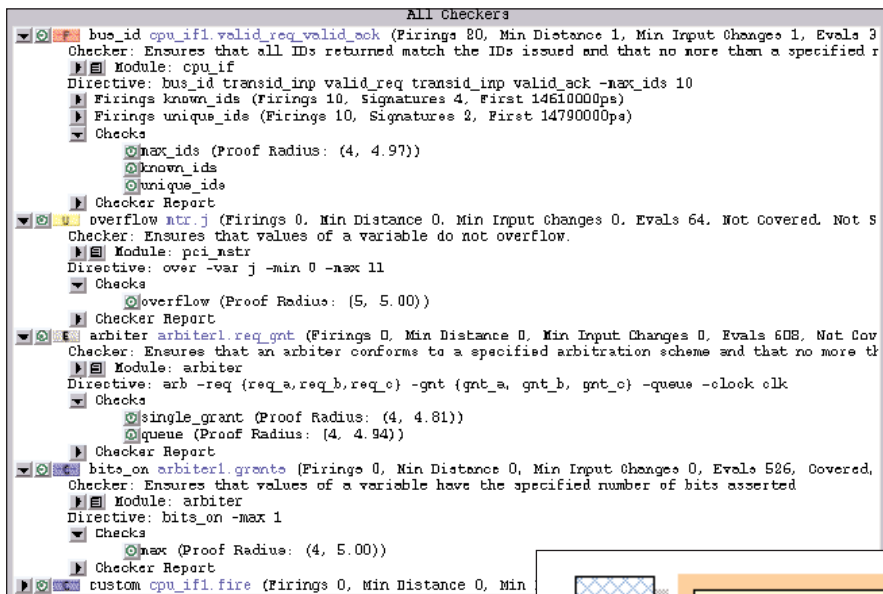
Comprehensive Library of Verification IP

By using the 0-In CheckerWare Library, you can quickly tap into the power of formal verification. The 0-In Formal Verification solution is completely compatible with the CheckerWare Library, which includes a wide range of pre-verified elements for verifying common RTL structures (e.g., handshaking, FIFOs, arbiters, and FSMs) and standard interfaces (e.g., DDR-SDRAM, AMBA, and PCI Express). Each CheckerWare Library element can be used either as an assertion to be verified or as an input assumption.

Comprehensive Formal Coverage Metrics

Even though formal verification analysis is exhaustive, users of formal verification tools need feedback to help them answer the questions: “Am I done?” and “What should I do next?” The 0-In Formal Verification solution provides the industry’s most comprehensive formal coverage metrics and methodologies to help answer these questions.

The 0-In Formal Verification solution reports a *proof radius* for each assertion, meaning that violation of the assertion is impossible within proof radius number of cycles. Typically, an infinite *proof radius* is reported for many assertions. For the remaining assertions, the *proof radius* number can be used to determine whether additional formal analysis is worthwhile.



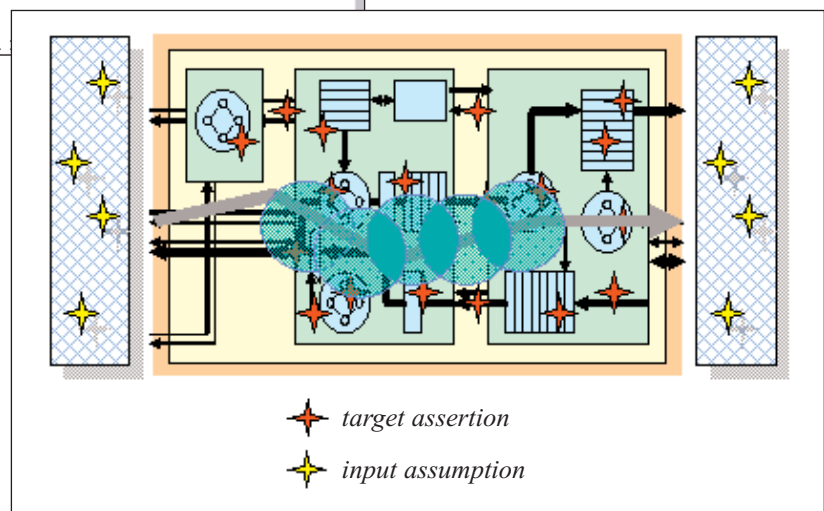
The 0-In View reporting environment displays structural coverage metrics.

Formal analysis isn’t finished until the formal tool has covered all the corner cases and preconditions of the assertions. The 0-In Formal Verification solution supports direct targeting of assertion corner cases and preconditions using formal analysis. It reports the level of coverage achieved via convenient text reports as well as the unified 0-In View reporting environment. These *structural coverage* reports can be used to find holes in the formal analysis test plan (e.g., an additional value needed in a configuration register) and to guide test plan improvements.

Easy-to-Use Methodology for Input Assumptions

All formal analysis tools require that the user specify assumptions about the behavior of inputs on the boundary of the design; otherwise, assertion violations found by formal analysis may correspond to illegal input behavior instead of real bugs. The 0-In Formal Verification solution supports the industry’s easiest-to-use methodology for specifying and managing design-boundary assumptions. Design-boundary assumptions are specified at the block or cluster level, using

assertions in any standard format or language, including the 0-In CheckerWare Library. These boundary assertions operate during normal simulation runs, and any problems with the assertions can be easily diagnosed using simulation. When blocks or clusters are being integrated, the boundary assertions directly detect inter-block communication problems, substantially accelerating the verification process. The boundary assertions are used by formal analysis ensure that any assertion violations found represent real bugs.

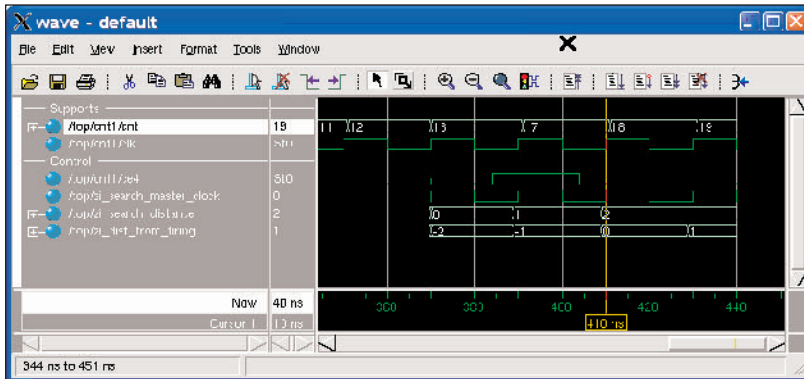


Ordinary assertions specify the behavior of design inputs.

Easy-to-Use, Simulation-Centric Debug

When the 0-In Formal Verification solution finds an assertion violation, it creates waveforms to facilitate debugging. It also automatically reproduces the violation using your existing simulator, so familiar simulation debug tools can be used to diagnose the problem. Supported simulators

include ModelSim®, VCS®, and NC-Sim. Debug options include concise textual reports, color-coded graphical debug using the unified 0-In View reporting environment, and ModelSim, Verdi™, and Debussy® waveform viewers.



Assertion violations are easily diagnosed using simulation waveforms.

For more information, visit our website at www.mentor.com/fv.

Copyright © 2005 Mentor Graphics Corporation.

CheckerWare, Mentor Graphics, ModelSim, and 0-In are registered trademarks of Mentor Graphics Corporation. All other trademarks mentioned in this document are trademarks of their respective owners.

Corporate Headquarters
Mentor Graphics Corporation
8005 S.W. Boeckman Road
Wilsonville, Oregon 97070 USA
Phone: 503-685-7000
North American Support Center
Phone: 800-547-4303
Fax: 800-684-1795

Silicon Valley
Mentor Graphics Corporation
1001 Ridder Park Drive
San Jose, California 95131 USA
Phone: 408-436-1500
Fax: 408-436-1501

Europe
Mentor Graphics
Deutschland GmbH
Arnulfstrasse 201
80634 Munich
Germany
Phone: +49.89.57096.0
Fax: +49.89.57096.400

Pacific Rim
Mentor Graphics Taiwan
Room 1603, 16F,
International Trade Building
No. 333, Section 1, Keelung Road
Taipei, Taiwan, ROC
Phone: 886-2-27576020
Fax: 886-2-27576027

Japan
Mentor Graphics Japan Co., Ltd.
Gotenyama Hills
7-35, Kita-Shinagawa 4-chome
Shinagawa-Ku, Tokyo 140
Japan
Phone: 81-3-5488-3030
Fax: 81-3-5488-3031

**Mentor
Graphics®**