



Put "phrases in quotes"

[News & Analysis](#) | [SolutionSource™](#) | [Academic Insights](#) | [Expert's Corner](#) | [Industry Access](#) | [Classifieds](#) | [About Us](#)

You are here: [Home Page](#) / [News & Analysis](#) / [OneSpin offers step-by-step formal verification suite](#)

Thursday, May, 6th 2010

[Print](#) [Email](#) [Bookmark](#)

First Look

OneSpin offers step-by-step formal verification suite

By **Richard Goering**

01/19/09

Offering a range of capabilities aimed at different levels of expertise, OneSpin Solutions this week (Jan. 19) is releasing its 360 MV product family. Five different, interoperable offerings within the family promise a structured, step-by-step approach to learning and applying formal assertion-based verification (ABV) for ASICs, FPGAs and systems-on-chip (SoCs).

OneSpin is thus offering something for almost every design and verification team – from fully automated checks aimed at novice formal verification users to “gap-free verification” that allows experienced users to achieve proof of a design’s full functionality. Additionally, OneSpin is offering a new automated “proof based” debugger for SystemVerilog Assertions (SVA).

“It is clear that most companies today are leveraging formal verification in combination with dynamic verification,” said Michael Siegel, OneSpin director of product marketing. “However, the level of expertise required to leverage formal verification has so far limited its use. This [360 MV] product family aims to provide a step-by-step approach to leveraging successively more formal verification, and provides different entry points for people, be they novices or experienced formal verification users.”

By and large, the 360 MV family reconfigures capabilities already found within OneSpin’s existing 360 MV (Module Verifier) formal property checking tool. However, the proof-based SVA debugger is new, and there is added technology that helps the tool avoid false negatives. Moreover, Siegel said, the new release provides a 2-3X improvement in performance and capacity over what OneSpin previously offered.

The 360 MV family starts with a product called Inspect, which offers fully automated checks. “People coming from the simulation side who have not been exposed to formal verification now have an easy entry point that does not require any formal knowledge at all,” he said. “Nonetheless, they can get an immediate return on their investment and use some formal technology to improve their verification.” Then, he said, users can step up to the next level, which involves writing simple assertions to verify implementation intent.

Stepping through levels

Siegel said that 360 MV covers six different “levels” of formal verification use, each one offering a different entry point in terms of capabilities and required expertise. The first level is represented by the automated checks provided by 360 MV Inspect. At the next level, “implementation intent” is verified by 360 MV Check, which uses simple assertions. A third level, “functional requirements,” allows more complex assertions, and a fourth level, “design operations,” brings verification up to the transaction level – a view used in testbenches. Both levels three and four are covered by 360 MV Verify.

Level five is the automatic gap detection provided by 360 MV Assure, and level six is the gap-free verification provided by 360 MV Certify. While both help the user find missing assertions, Assure focuses on design inputs, while Certify looks at both inputs and outputs.

Each tool is a superset of the previous one – for example, Check includes all the capabilities of Inspect, and Verify includes all the capabilities of Check (see Figure 1).

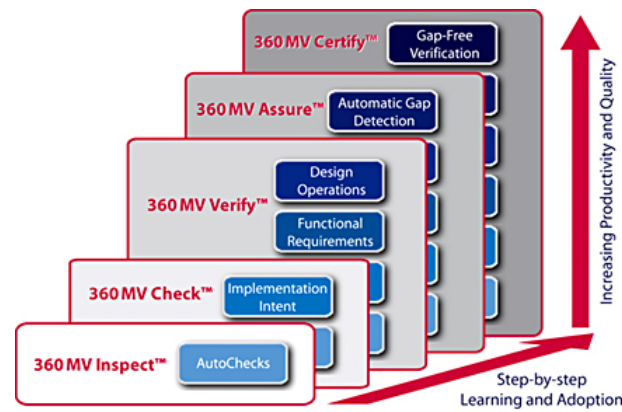


Figure 1 – The 360 MV product family includes five products that cover six levels of functional verification.

There's no need to write any assertions for 360 MV Inspect – the tool automatically extracts a set of checks from RTL code. Examples include out-of-bound array access, stuck-at faults, dead code, and synthesis/simulation mismatches. "All are proven exhaustively, far beyond the usual lint tool," Siegel said. If an error is detected, Inspect's debug environment links to the line of code that has failed, and shows a counter-example, if appropriate.

With 360 MV Check, users write simple assertions using SVA or the Open Verification Library (OVL). For example, a user could write an assertion specifying that if signal A goes high, signal B should follow within two cycles.

360 MV Verify allows more complex assertions, such as those relating to arbitration schemes, interrupt handling, or protocol compliance. To help users debug complex assertions, Verify includes OneSpin's new automated, proof-based SVA debugger. In addition to a waveform display, it includes a structural debugger that identifies the part of the assertion that failed, and a "temporal fan-in" debugger that can trace the failed part of the assertion back to the relevant signals.

OneSpin claims that Verify can scale to designs beyond 20,000 state bits. Verify also supports OneSpin's SVA Timing Diagram Assertion Library (Tidal), [introduced June 2008](#), which lets users capture functional requirements as assertions derived from timing diagrams.

According to Siegel, Verify leverages the "operational and transactional" view employed by simulation testbenches, and as such, opens the door to the "systematic formal verification" of all operations used in the design. The read and write operations of a cache controller are examples of operations.

Closing the gaps

There's a potential downside to using only Verify, however – if the user's assertions don't capture all operations, there's a risk of missing critical behavior. That's where the automatic gap detection in 360 MV Assure comes in. Gap detection identifies unverified design behavior. It then guides the user in writing additional assertions to make sure that gaps are closed.

While Assure identifies unverified input scenarios, 360 MV Certify additionally pinpoints any output signals that are not covered by assertions. For example, suppose the user has written assertions for every possible instruction of a processor, but didn't write one for the overflow bit. 360 MV Certify can identify the gap – a gap that would not be discovered by Assure. Certify's "gap free verification" guides users through verification planning, assertion development, and gap detection checks.

With 360 MV Certify, Siegel said, the SVA assertion set ultimately becomes the "golden reference model" for the design under verification. When all assertions have been written, confirmed to be gap-free, and formally proven, then the RTL functionality is proven to be functionally equivalent to the golden reference model.

OneSpin still supports its legacy ITL assertion language, but the company's focus going forward is SVA, Siegel said. OneSpin does not currently support the Property Specification Language (PSL).

All five 360 MV products are available immediately, starting at \$20,000 for Inspect and ranging up to \$180,000 for the full suite of capabilities with Certify. The 360 MV family will be demonstrated at the [EDS Fair](#) in Japan Jan. 22-23, as well as the Design and Verification Conference ([DVCon](#)) in San Jose, Calif. Feb. 24-26.

Related articles

[OneSpin timing diagram library supports 'gap-free' verification](#)

[OneSpin fills verification 'gaps' with SVA properties](#)

[Formal tool opens 'gateway' to assertion reuse](#)

[Formal property checking – what the users say](#)

[Formal property checking – what the vendors say](#)

Company Contact



OneSpin Solutions – US Office
 1183 Bordeaux Drive, Suite 16
 Sunnyvale, CA 94089 USA
<http://www.onespin-solutions.com>

Learn more

- ◆ [Download: 360 MV Product Family Datasheet](#)
- ◆ [Screen shot: Proof-Based Debugger for Complex SystemVerilog Assertions Reduces Manual Effort](#)
- ◆ [Press release: "OneSpin Mainstreams Comprehensive Formal Assertion-Based-Verification"](#)
- ◆ [Technical Article \(IDesign\): "Timing Diagrams Ease Formal Property Development"](#)
- ◆ [Interview SCDSsource: "OneSpin fills verification 'gaps' with SVA properties"](#)
- ◆ [Viewpoint: "Formal Verification: Shining a Light into the Darkness"](#)

In case you are not able to click one of the hyperlinks above please make sure to be [registered](#) and [logged-in](#).

Request for Information

Your message:

You must be [registered](#) and [logged-in](#) to submit an rfi.

[Back to Home Page](#)

All materials on this site Copyright © 2007-2009 Tech Source Media, Inc. All Rights Reserved | [Privacy Statement](#)