

The Future of SystemVerilog Verification

Cisco.com

Jon Michelson
Cisco Systems

Co-author: “The Art of Verification with SystemVerilog Assertions”

Co-author: “The Art of Verification with Vera”

Silicon Valley Verification Luncheon 11/8/05

Agenda

Cisco.com

- **Current status and verification benefits**
- **Advances require better tools**
- **Need more LRM improvements**
- **Suggested LRM improvements**

Where Are We?

Cisco.com

- **SystemVerilog greatly improves upon Verilog 2001 in 4 areas**
 - Testbench (SV-TB)**
 - Objects, constraints, coverage, etc.
 - Assertions (SVA)**
 - Temporal specifications, assertions, coverage, etc.
 - Design (SV-D)**
 - Data types, arrays, interfaces, operators, etc.
 - API (DPI)**
 - Connecting C++, Perl, and 3rd party tools
- **I'm unaware of any vendor implementing all of it yet**
 - But tools are coming (too slowly perhaps, but still...)**

Verification Benefits

Cisco.com

- **In isolation, each area of improvement is just a similar way to do it**
 - SV-D: Vera and E (structures, operators, etc.)**
 - SV-TB: Vera and E**
 - SVA: OVA and PSL**
- **But verification requires all 4 areas, and all 4 areas have finally become a cohesive language**
 - Assertions are not quarantined in comments**
 - SV-TB can react to assertions and visa-versa**
 - SV-TB can leverage data structures used by DUT and visa-versa**
 - DV and design code can be intermingled**
 - Helps blur the line between designer and DVer, eases maintenance, decreases the probability of “forgotten” verification, etc.**
 - No PLI performance bottlenecks**
 - Other performance improvements from compilation & single kernel**

Multivendor Compatibility

Cisco.com

- **Do not underestimate the benefits of 1 language – they help tremendously – but the big win is multivendor compatibility**
- **Chip/system teams are now distributed across business units, geographies, even companies**
 - Distributed teams tend to have different tool sets, but code sharing is imperative given complexity, reuse, and schedule**
- **Users need multiple vendors for solid execution and innovation**
 - New vendors (i.e. innovation) need lots of users for market justification**
- **Only possible with standardization of comprehensive constructs**
 - Other supposed standards have come and gone**
 - It's only really a standard if at least 2 major vendors have compatible implementations of an LRM**
 - (And it doesn't matter what neutral body ratifies the LRM – IEEE, Accellera, Standards R Us, etc.)**
 - SV will be first to get there for design, TB, and assertions**
 - All 3 major vendors committed & tons of smaller vendors too**

Advanced Features Require Better Tools

Cisco.com

- **Waveform viewers need more intuitive debugging features**
 - Assertions have multiple concurrent attempts**
 - Each attempt has branches containing unique local variable instances**
 - Need to see local variable flow through these branches**
 - Need to see all SVA layers since they behave differently**
 - Multiple matches at sequence layer vs. single match at property layer**
 - Need to distinguish between vacuous and non-vacuous matches**
 - Testbench is object oriented and dynamic**
 - Waveform viewers need to become more like debuggers**
- **SVA coverage should automatically bin expressions**

Advanced Features Require Better Tools (Cont.)

Cisco.com

- **Improved integration of formal analysis and functional simulation**

Sophisticated constraint solvers based on model checking to improve general and corner case coverage

Eliminate need for directed part of directed random

Current constraint solvers tout solver capacity, flexibility, and solution distributions

We need to start talking about time to coverage as well

- **Build one testbench with all legal SystemVerilog constructs that both simulation and formal can use**

Control logic is best expressed in assertions while data path is best expressed in classes

Write once, use everywhere

More natural dual-use formal / simulation environments

Will enable broad deployment of formal technology

Need to Continue Rapid LRM Improvements

Cisco.com

- **SystemVerilog is many huge steps forward from Verilog 2001, but...**
- **Accellera/IEEE/whoever needs to keep improving it rapidly to continue accelerating productivity**
- **Proprietary language extensions should not get too far ahead of the standard**

Otherwise we're back where we started with no "effective" standard

- **I encourage EDA vendors to donate their SV LRM innovations immediately**

Avoid delaying donation of proprietary extensions

Users will adopt standards (see previous definition) much faster than proprietary extensions

How to Compete with Quick LRM Submission

Cisco.com

- **There's plenty of room to differentiate tools elsewhere:**
 - Performance: speed & memory usage**
 - Quality of results**
 - Stability**
 - Integration of ancillary tools**
 - Broad support for LRM features**
 - Sophisticated implementation of ratified LRM features**
 - Become the perceived leaders with newly donated and standardized features**
- **→ Differentiate based on implementation, not based on proprietary language extensions**

Suggested SVA Improvements

Cisco.com

- **3.1a (P1800) is a great start, but...**
- **Remove distinctions between sequence and property layers**
 - It's needless complexity and confusion for users**
 - Lack of concatenation at property layer causes needless contortions**
 - Should define implications and not at the sequence layer**
- **Allow classes to instantiate sequences and properties**
 - Constraint solver can now easily use results of temporal expressions – a great way to build transactors and generators**
 - Checkers can now use assertions and easily react to them**
 - Without direct instantiation, reactivity is cumbersome**
 - Removes object oriented benefits and is not scalable**

Suggested SVA Improvements (Cont)

Cisco.com

- **Differentiate between vacuous and non-vacuous matches**
 - Perhaps add property methods `.vacuous` and `.non_vacuous` or extend `ended/matched/triggered` to apply to properties with a parameter for `vacuous`, `nonvacuous`, or `both`
 - Will clean up complemented implications, chained implications, and procedural reactivity to implications
- **Allow variables in repeat range expressions**
(With clear semantics about when they are sampled)
- **Allow infinite repeat: `a[*$]`**
(This is different than `a[*0:$]`)
- **Allow negative delays as syntactic sugar for `$past`**
- **Allow more complex data types (classes) within assertions**

Suggested Testbench Improvements

Cisco.com

- **3.1a and P1800 testbench constructs are a great start, but...**
- **Users need the rest of Vera and E in SystemVerilog**

AOP

Stream Generator

Other minor things

Summary

Cisco.com

- **SV provides a single, cohesive language to enable higher performance, multivendor, modern verification environments**
- **Multivendor compatibility is required by today's development environments**
- **Advanced features require tool innovations**
- **Advancing productivity also requires LRM improvements**
- **Multivendor standardization will accelerate tool innovations and should accelerate LRM improvements**
- **Vendors should differentiate based on implementation of the ratified spec and donate spec improvements quickly**

For More SystemVerilog Assertions...

Cisco.com

- **Topics Covered:**

Assertion-based verification with simulation and formal analysis

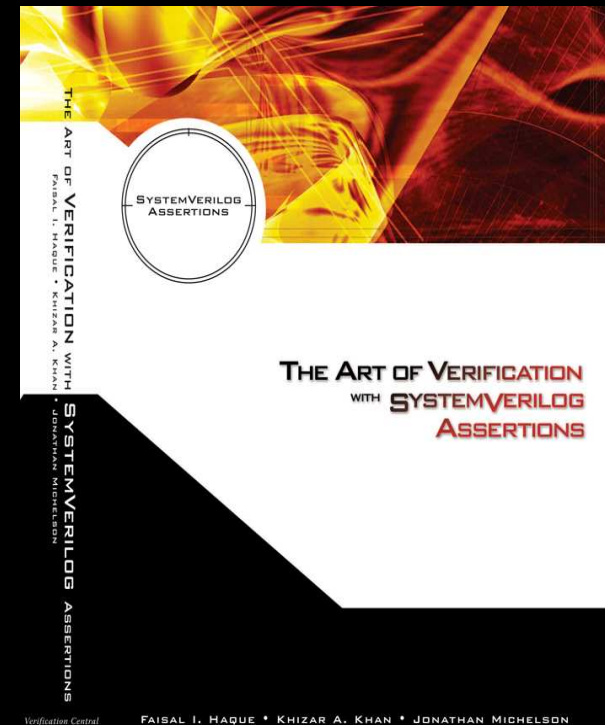
SVA capabilities with practical verification examples

Booleans

Sequences

Properties

Assertions, coverage, and assumptions



To subscribe to preorder list please visit: **Shipping 1Q 2006**

<http://www.verificationalcentral.com>