

# What Users Need From SystemVerilog Vendors

Cisco.com

**Jon Michelson**

**Cisco Systems**

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

**EDA Interoperability Forum, 4/7/2005**

# Agenda

Cisco.com

- **Why I think SVA is most important part of SV for now**
- **Users need implementations**
- **Users need more innovative implementations of current LRM**
- **Need to rapidly improve SV LRM**
- **Suggested LRM improvements**

# SVA Will Be Widely Deployed First

Cisco.com

- **SystemVerilog 3.1a (and P1800) are great improvements upon Verilog 2001 across 4 axes:**

**Assertions, testbench, design, and DPI**

- **Assertions will be widely deployed by users first**

**SVA is available ahead of SV-TB**

**Easier to implement new DV flows than new design flows**

**Assertion deployment just requires simulator and waveform viewer/debugger support**

**Design deployment requires the same tools plus synthesis, equivalency checking, etc.**

**Less perceived risk with new DV tools than new design tools**

# SVA Will Be Widely Deployed First (Cont)

Cisco.com

- **SVA is most powerful improvement in SV given where advanced users are already:**

**Users already have home grown and/or commercial tools which implement SV-like design constructs on top of Verilog**

**Users already use object oriented testbenches**

**Testbench automation tools have limited the need for custom PLI/DPI-based solutions**

**We are scarred from previous language wars, so OVA vs. PSL war hindered assertion deployments beyond template libraries such as OVL**

- **→ SVA represents the largest productivity quantum leap in SystemVerilog, so it will be embraced first**

# Need SV Implementations!

Cisco.com

- **The ratified LRM is a great first step**

**But...**

- **Verification requires simulators, waveform viewers & debuggers, and ancillary tools**

**Getting there for assertions, but...**

**Need a good viewer for debugging and training**

**Still lots of powerful constructs not implemented yet**

**Production versions of SV-TB not available yet**

- **SV-Design cannot be used until all relevant tools in the flow are compatible**

**Getting there...**

# Need More Innovative Implementations of Current LRM

Cisco.com

- **Waveform viewers need more intuitive debugging features**

**Assertions have multiple simultaneously executing attempts with branches all containing their own variable instances**

**Testbench is objected oriented and dynamic**

- **LRM is vague in certain areas and therefore leaves room for innovation**

**Allow SVA cover directive to automatically bin SVA constructs**

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

**Current constraint solvers tout solver capacity and solution distributions**

**We need to start talking about time to coverage as well**

# Need More Innovative Implementations of Current LRM (Cont.)

Cisco.com

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

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

**Write once, use everywhere**

**More natural dual formal / simulation environments**

**Will enable broad deployment of formal technology**

- **Allow classes in assertions and vice-versa**

**For example, packets are much better described in classes than in assertions**

**We don't want to write it once in classes for TB and again in assertions for formal**

**Similarly, control logic for packets is best described in assertions**

**But classes can't instantiate assertions easily**

# Need to Continue Rapid Improvement to LRM

Cisco.com

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

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

**It's not a standard unless at least 2 or more vendors have compatible implementations**

# Please Innovate via SV LRM Extensions

Cisco.com

- **EDA vendors are encouraged to donate their SystemVerilog LRM innovations immediately**

**Avoid delaying donation of proprietary extensions**

**Users will adopt standards much faster than proprietary extensions**

# Why Go Directly To SV LRM?

Cisco.com

- **Because users no longer want EDA tools based on proprietary languages**
- **Reuse, complexity, and time to market are driving the need to interoperate with many different pieces of IP**
  - Some from other groups within our company using different tools**
  - Some from other companies**
    - 3<sup>rd</sup> party IP, partners/vendors, and acquisitions**
- **Users need the productivity improvements of LRM innovations**
  - So please standardize your innovations ASAP!**

# Why Go Directly To SV LRM? (Cont.)

Cisco.com

- **There's plenty of room to differentiate your tools elsewhere:**
  - Speed & memory usage**
  - Quality of results**
  - Stability**
  - Integration of ancillary tools**
  - Broad support for LRM features (nobody has that yet!)**
  - Sophisticated implementation of ratified LRM features**
- **→ Differentiate based on your 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 in SystemVerilog**
- **Vera users are already using features not in SV-TB**

**Not likely to step backwards**

- **E users want AOP**

**More likely to stick to E than to switch to SystemVerilog 3.1a TB**

# Summary

Cisco.com

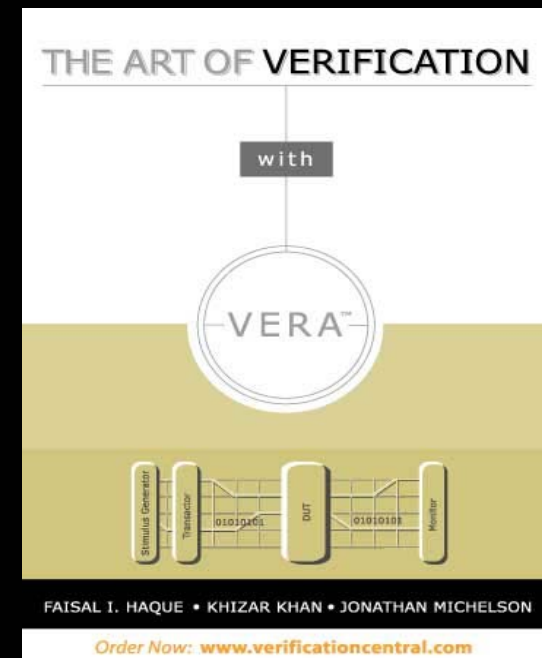
- **SVA is most important aspect of SystemVerilog in the near term**
- **Please implement what is already ratified soon**
- **Please cooperate with other EDA vendors to push the SV spec ahead quickly**
- **Differentiate based on implementation of the ratified spec**
- **Donate spec improvements quickly**

# For More on Verification...

Cisco.com

- **Focus of book**  
**Verification of complex SoCs**
- **Who can benefit**  
**Anyone involved with ASICs**
  - DV engineers*
  - ASIC engineers*
  - ASIC managers*
  - People wanting to become DV engineers*

<http://www.verificationcentral.com>



**SystemVerilog  
Assertions  
book coming!**