Of Camels and Committees:
Standardization Should Enable Innovation, Not Strangle It

Tom Fitzpatrick
Verification Evangelist, Mentor Graphics

Dave Rich
Verification Architect, Mentor Graphics

The First Commandment
For Effective Standards

“Cooperate on Standards
Compete on Products”
- Karen Bartleson
Standards Not Always Efficient

Another Perspective

“Only by providing something almost universally agreed to be **genuinely** useful can we make progress. *A standard committee is no place for single-issue fanatics.*”

- Bjarne Stroustrup
  Developer of C++

---

Standardization Can Remove Choice

Solver Innovation: Einstein Puzzle

<table>
<thead>
<tr>
<th>Nationality</th>
<th>British</th>
<th>Swedish</th>
<th>Danish</th>
<th>Norwegian</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>House Color</td>
<td>House 1</td>
<td>House 2</td>
<td>House 3</td>
<td>House 4</td>
<td>House 5</td>
</tr>
<tr>
<td>Pet</td>
<td>Dog</td>
<td>Bird</td>
<td>Cat</td>
<td>Fish</td>
<td>Figure</td>
</tr>
<tr>
<td>Drink</td>
<td>Tea</td>
<td>Coffee</td>
<td>Milk</td>
<td>Water</td>
<td>Glass</td>
</tr>
<tr>
<td>Sport</td>
<td>Baseball</td>
<td>Football</td>
<td>Soccer</td>
<td>Hockey</td>
<td>Figure</td>
</tr>
</tbody>
</table>
Multiple Sources = Bloat

- Why have two ways to do the same thing?
  - Worse: *almost* the same thing
    
    ```
    always @*
    C = A & B;
    
    always_comb
    C = A & B;
    ```

  - Verilog 1364-2001
  - Evaluated when RHS changes
  - SUPERLOG
  - Evaluated when RHS changes *and at time 0*
  - Use this one

- Once something is in a standard, it’s almost impossible to remove it

More SystemVerilog Bloat

- **program** blocks left over from Vera
  - Starts executing at time 0
  - Simulation terminates when last **program** exits
    - Not needed for UVM
  - Defined new timing semantics to avoid test/design race conditions
    - Mimics semantics of PLI application
  - Still subject to races in the testbench

- **clocking** block handles test/design races
  - Doesn’t affect timing/semantics of testbench
Use of Embedded CPUs Increasing

- Mean # of CPUs:
  - 2004: 1.06
  - 2012: 2.25 (↑ 212%)
- % Designs with >= 1 CPUs
  - 2004: 52%
  - 2012: 79% (↑ 152%)
- % Designs with >= 2 CPUs
  - 2004: 17%
  - 2012: 57% (↑ 335%)


UVM Targeting the Past?

- OVM 1.0: Jan 2008
- OVM 2.0: Sep 2008
- VIP-TSC Formed: May 2008
- UVM 1.0EA: May 2010

User Feedback is Critical

- New code should benefit users
  - Phasing still doesn’t work
  - “Top Priority”
  - Don’t repeat mistakes
- UVM1.2 introduces 13 backward-incompatible changes
  - Considerable performance degradation
  - No extended user feedback period
Troublesome New UVM Features: UVM Messaging

- 20 new macros added
- Add multiple fields to a message
- Different actions per field
- 20% performance penalty
  - Just to process existing messages
  - without using new features

Troublesome New UVM Features: UVM Transaction Recording

- Text-based reporting in the standard
  - Replaced by vendor-specific implementation for tools
- NOT Backward Compatible
  - Requires additional work by vendors to support 1.1d and 1.2
- 65% performance penalty in 1.2 vs. 1.1d
  - Recording overhead 76% in 1.1d
  - Recording overhead 126% in 1.2
Aim at the Right Target

- Verification has clearly moved beyond block-level
  - Multi-core
  - Complex interconnect
- Constrained-random sequences don’t cut it anymore
  - Impossible to write constraint equations for cache coherency
- Need higher level of abstraction

Real Reuse Requires Abstraction

- Begin with the End in Mind
- UVM ideally suited for structural reuse
  - Factory, config, build_phase(), connect_phase()
- Stimulus reuse requires abstract specification
  - Retargetable to UVM sequences and software
Real Reuse Requires Abstraction

- Begin with the End in Mind
- UVM ideally suited for structural reuse
  - Factory, config, build_phase(), connect_phase()
- Stimulus reuse requires abstract specification
  - Retargetable to UVM sequences and software

The Problem Has Changed

- System-level verification requires more than UVM
  - Software execution coordinated with external stimuli
  - Constrained-random isn’t enough anymore
- UVM 1.1d is a structural innovation platform
  - Flexibility to specify the environment
  - Allow innovative technology to drive verification
- UVM 1.2 is not necessary
  - Certainly not without extended user feedback period
The Answer is NOT:

TF, Of Camels and Committees, DVCon 2014

Thank You

Mentor Graphics®