Use, Analysis, and Debug of SystemVerilog Assertions
Agenda

- Introduction
- Source Code Tracing
- Assertion Checking
- Analyzing and Debugging
  - Waveform
  - Active Annotation
  - Property Result Table
Standards: The Life blood of Interoperability

- Standards mean (users/vendors)
  - Greater confidence
  - Less risk
- Accellera and IEEE SA allow (users/vendors) early involvement and rapid development
  - E.g. SystemVerilog VPI
    - Have been working jointly with Synopsys VCS team to enhance the VPI beyond the LRM for local variable access – plan to feedback to committees in the future
    - Need other simulators to quickly add support for VPI to accelerate adoption – customers want interoperability!

Novas is part of IEEE and Accellera SystemVerilog committees
Novas’ Open APIs

- Waveform database APIs
  - Writer for detection tools
  - Reader for analysis tools
- Design database APIs
  - Writer for adding data
  - Reader for tools
- Visualization tool control API
  - TCL functions for every command, and more
- Simulation control API
  - Debug system as simulator GUI
Debug & Analysis Requirements

Lots of interest in SVA because you can start using it without altering the rest of your flow (synthesis); lots of people looking at SV design and SVTB for next year, a few now.
Novas Technology Leadership

More Than Simple Visualization

Design, Testbench, Assertions

Open Databases

Knowledge Compilers

APIs

Full Range of Tools

Analysis Engines

Knowledge Database

Behavior Analysis

Structure Analysis

Behavior Exploration

Behavior Query

Language Extraction

Visualization

Temporal Flow Graph

Source Code Trace

Waveform / Event

Schematic / Structure

FSM / Flow Chart

Assertions

More Than Simple Visualization
Novas Platform for Assertion-Based Debug

- Wide Language Support – Available **Now** …
  - OVA
  - PSL
  - SVA

- Source Code Support
  - Unified environment for tracing assertion code or between assertion and design

- Assertion Checking
  - Unique FSDB Checker Engine to check assertions in real-time without re-running simulation
  - Simulation – PLI library support

- Results Analysis
  - Enhanced Waveform for assertions
  - Active Annotation
  - Property Results Table – sorting and filtering of assertion data
  - Specialized treatment for Local Variables
Agenda

- Introduction
- **Source Code Tracing**
- Assertion Checking
- Analyzing and Debugging
  - Waveform
  - Active Annotation
  - Property Result Table
How about a global picture of *all* assertions in design …
All assertions in design with **Filtering and Sorting**

D&D from table to source code or waveform
Property Management Window – Tree View
SVA Source Code Tracing (I)

Focus on an assertion (or property or sequence) by DC’ing in HB or D&D from some other view
Double-click on a property or sequence reference to go (change scope) to its ‘definition’
Design Variables referenced inside properties and sequences can be traced “as usual”
Agenda

- Introduction
- Source Code Tracing
- Assertion Checking
- Analyzing and Debugging
  - Waveform
  - Active Annotation
  - Property Result Table
Assertion Checking

D&D

Simulator

FSDB

$fsdbDumpSVA
$fsdbDumpPSL
... etc.
FSDB Checker Support in Verdi – Why?

- Idea: Perform **Q&A with engineer** using assertion language. Verdi can check the assertion “questions”.
  - **Flexibility**: Quickly check the assertion no need to re-compile design and re-simulate.
  - **Speed**: Check low-level “implementation” assertions instead of slowing doing system simulation
  - **Power**: Has complete FSDB time, no history (future) limitations
  - **Detail**: Compute all the dependent events and sequences for easy debug
  - **Trace data**: Need to pre-generate design trace data for analysis

- **Assertion Language Support**
  - OVA
    - OVA 1.3
  - PSL
    - PSL 1.01/1.1 “simulation subset”
  - SVA
    - SVA 3.1a
Agenda

- Introduction
- Source Code Tracing
- Assertion Checking
- Analyzing and Debugging
  - Waveform
  - Active Annotation
  - Property Result Table
Searching

Search criteria: start, fail, success/match
Associated Data Tagging

Underlying “building block” data **tagged** for easy double-click access …

… all the way down to signals and local variables
Have been working jointly with Synopsys to enhance the VPI beyond the LRM for local variable access from VCS – plan to feedback to Accellera & IEEE committees in the future.
Local Variables – Another Example

UE = Under Evaluation
SE = Start of Evaluation
NV = No Value
NF = Not Found
not in FSDB file (not in dumped scope, or (see gotchas...))

“Make sure PC_load stays high for 5 consecutive cycles”

Note: example is not real-life but illustrates the key points

Double-Click
Multiple Threads

Why? …. Cover of a sequence does not do first_match like property/assert, it tracks all matches

Note: example is not real-life but illustrates the key points

Assign to the local var after the branching results in 2 threads

Set of Set Notation – each attempt has 2 branches or threads
Value for "Current" scope will be annotated

Enable computation of genvars, Source → Parameter Annotation
Beyond Waveforms – View as a Detailed List ...

Sorting and ....

... Filtering
... Or Coverage

<table>
<thead>
<tr>
<th>Instance</th>
<th>UnitName</th>
<th>Property</th>
<th>Type</th>
<th>FSDB File</th>
<th>Success</th>
<th>Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>systemi_cpu</td>
<td>cpu</td>
<td>INCPC</td>
<td>ASSERT_CHECK</td>
<td>sv.fshdb</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>systemi_cpu</td>
<td>cpu</td>
<td>INCPC</td>
<td>ASSERT_CHECK</td>
<td>sv.fshdb</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>systemi_cpu</td>
<td>ALUE</td>
<td>OF_COVER</td>
<td>COVER</td>
<td>sv.fshdb</td>
<td>2</td>
<td>42</td>
</tr>
<tr>
<td>systemi_cpu</td>
<td>ALUE</td>
<td>OF_COVER</td>
<td>COVER</td>
<td>sv.fshdb</td>
<td>2</td>
<td>42</td>
</tr>
<tr>
<td>systemi_cpu</td>
<td>ALUE</td>
<td>OF_COVER</td>
<td>COVER</td>
<td>sv.fshdb</td>
<td>2</td>
<td>42</td>
</tr>
<tr>
<td>systemi_cpu</td>
<td>ALUE</td>
<td>OF_COVER</td>
<td>COVER</td>
<td>sv.fshdb</td>
<td>2</td>
<td>42</td>
</tr>
<tr>
<td>systemi_cpu</td>
<td>ALUE</td>
<td>OF_COVER</td>
<td>COVER</td>
<td>sv.fshdb</td>
<td>2</td>
<td>42</td>
</tr>
<tr>
<td>systemi_cpu</td>
<td>ALUE</td>
<td>OF_COVER</td>
<td>COVER</td>
<td>sv.fshdb</td>
<td>2</td>
<td>42</td>
</tr>
</tbody>
</table>
Standards are the Lifeblood for Interoperability

Novas Provides Essential Support for Productive Analysis & Debug of SVA
  - Intelligent Assertion Source Code Tracing
  - Quick Assertion Checking
  - Advanced Visualization, Analysis, & Debug

Meeting the [Debug] Challenges
  - Local Variables
    - Multiple Attempts
    - Multiple Threads

**Novas Solutions ➔ Verdi™ Automated Debug System**