

# General Update

IEEE P1850

Working Group Meeting

Tuesday March 27, 2007

# IEEE DASC-SC March 21-st Meeting

- Work underway to update the DASC Policies & Procedures
  - Provide better framework for operation and decision making
  - Most of the meeting focused on refining proposed text
- Approval vote of new P&P scheduled at DATE 2007
  - Link will be made available to new document for review
- We will refine our existing P&P based on the DASC-SC P&P
  - Our existing P&P, dated December 21 2004, can be viewed at  
:  
<http://www.verilog.org/ieee-1850/docs/PSLWG-PP-2004-12-21-approved.pdf>

# Unapproved P1850 Schedule

|              |  |
|--------------|--|
| April 2007   | Draft PAR sent to WG for approval  |
| May 2007     | New PAR sent to DASC for approval  |
| June 2007    | LRM subcommittee begin updates to LRM<br>-- Format into draft/review document<br>-- Address simple errata errors in initial draft D1           |
| October 2007 | Issues/Extension subcommittee proposals complete<br>-- Begin work on LRM text modification<br>-- Proposed word changes comes from subcommittee |

# Unapproved P1850 Schedule

|                |  |
|----------------|--|
| February 2008  | Invitation to ballot sent out<br>-- LRM under committee review and modifications |
| April 2008     | 30-day initial sponsor ballot  |
| May 2008       | Committee begins work on address ballot comments                                 |
| June 2008      | Recirculation ballot   |
| August 2008    | RevCom submission  |
| September 2008 | Final approval of new standard   |

# Surrendra's Update on IEEE 1800 SVA

# Issues sub-committee Update

# The Topics Currently Discussed

- Use models
- New architecture for vunits
- Assertion control capabilities

# Use Models

- Motivation:
  - The current definition of the verification layer allows many interpretations (use models)
  - Various tools use different use models but currently there is no common language to relate them
  - Our goal is to list all the allowed use models such that each tool can specify which use model(s) it supports
- We made a list of allowed used models. For example abstract vs. concrete use model (bound to a design)
- We now need to list the requirements from the vunit architecture implied by each use model

# New Architecture of VUNITS

- Extensions - Reusable verification IP
  - Parameterized vunits
  - Resolving name conflicts
  - rules for inheritance and overriding - should it be as in the HDL?  
should we allow full inheritance paths?
- Defined with the list of use models in mind
- The new definition is expected to resolve open issues in the previous definition

# New Architecture - Status

- We distinguish between inheritance, binding and instantiation
- We are going over examples for the use of the above
- Need to define a configuration tree for each use model that imply restrictions on inheritance, binding and instantiation

# Assertion control Capabilities

- Goal – adding assertion control capabilities to PSL for simulation
- Solution: adding procedural blocks and local variables
- Our aim is to have a simple, easy to use definition with as few restrictions as possible.
- We would like to be compatible with SVA but not in any price
- We are now working on the formal semantics

# Formal semantics of Local variables

- We started from a proposal which was very similar to SVA's formal semantics
- Difference from SVA:
  - Our goal is to avoid restrictions present in SVA such as blocking local variables from flowing out of SERE &&
  - PSL has operators that are not present in SVA. We need the semantics to take care of suffix implication with procedural blocks that are embedded in other PSL FL operators.

## Local variables - Status

- We have an initial proposal for the formal semantics
- We made a list of “sanity checks” which we use to test the proposal
- We will appreciate examples for the use of local variables in real-life
- We are looking for suggestions for the syntax of procedural blocks
- More can be found in  
<http://www.eda.org/ieee-1850/Issues/Group-P.1.html>