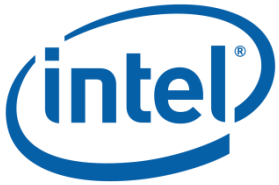

Requirements for next SystemVerilog PAR

Tom Alsop
Dmitry Korchemny
Matt Maidment
Erik Seligman

Summary

- Verification continues to prove challenging; SystemVerilog needs to improve to meet the challenges
- Polled Intel design and verification users
- Received considerable feedback on design, testbench and assertions
- Intel would like to see the next PAR address user feedback including the following items

Design

- Improved coding productivity
  - Parameterized tasks, functions and data types
  - Generate for module port definitions and instance port connections
  - Variable part-selects
  - Variable number of task and function arguments
  - Macro improvements
    - Logical expressions, looping, clarified semantics
- Improved Checking
  - Warning on out-of-bounds array writes

TestBench

- AOP – Some important features of Aspect Oriented Programming
  - **Inline Extension**: The ability in classes to 'add to' or replace properties and/or methods via use of other files or configs
  - **Method Overloading**: The ability to chose the method or constraint within a class depending on the value of a property upon instantiation.
  - **Method Modify** : The ability within inline'd extended classes to specify _*how*_ methods get modified (replaced, run before, run after)
- Soft, Default, and Solve…Before Constraint methods
- Other and Pass weight distribution methods
- Enumerate extension and name scope fixes
- Cross cover points across different cover groups
- Introspection
- Exceptions
- Java-like Multiple Inheritance
- Constraint composition

Assertions

- Relax checker restrictions imposed on checkers and support module-like code in checkers
- Reduce restrictions on where checkers and concurrent assertions may be used (functions, tasks, classes)
- Enhance assertion system functions, including 4-value bit-vector functions
- Enhance type system to allow generic integral type and explicit type compatibility check
- Allow real data types in concurrent assertions. Integration with VerilogAMS
- Introduce temporal coverage in covergroups

Summary

- Design, testbench and assertion features need continued care and improvement to meet users' verification challenges
- The next SystemVerilog PAR should be scoped to see the language improves to meet those challenges