

# Supporting Assertion-Based Verification in VHDL

An Assertions Functional Team Working Meeting

Stephen Bailey

Technical Marketing

Chair, IEEE 1076

**Model Technology**

A MENTOR GRAPHICS COMPANY

# Instructions for the WG Chair



(Not necessary to be shown)

- At Each Meeting, the Working Group Chair shall:
  - Show slides #1 and #2 of this presentation
  - Advise the WG membership that:
    - The IEEE's Patent Policy is consistent with the ANSI patent policy and is described in Clause 6 of the *IEEE SA Standards Board Bylaws*;
    - Early disclosure of patents which may be essential for the use of standards under development is encouraged;
    - Disclosures made of such patents may not be exhaustive of all patents that may be essential for the use of standards under development, and that neither the IEEE, the WG nor the WG Chairman ensure the accuracy or completeness of any disclosure or whether any disclosure is of a patent that in fact may be essential for the use of standards under development.
- Instruct the WG Secretary to record in the minutes of the relevant WG meeting:
  - that the foregoing advice was provided and the two slides were shown;
  - that an opportunity was provided for WG members to identify or disclose patents that the WG member believes may be essential for the use of that standard;
  - any responses that were given, specifically the patents and patent applications that were identified (if any) and by whom.



## Slide #1

# IEEE-SA Standards Board Bylaws on Patents in Standards

## 6. Patents

IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard. This assurance shall be provided without coercion and prior to approval of the standard (or reaffirmation when a patent becomes known after initial approval of the standard). This assurance shall be a letter that is in the form of either

- a) A general disclaimer to the effect that the patentee will not enforce any of its present or future patent(s) whose use would be required to implement the proposed IEEE standard against any person or entity using the patent(s) to comply with the standard or
- b) A statement that a license will be made available without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination

This assurance shall apply, at a minimum, from the date of the standard's approval to the date of the standard's withdrawal and is irrevocable during that period.

# Inappropriate Topics for IEEE WG Meetings

Slide #2



- Don't discuss licensing terms or conditions
- Don't discuss product pricing, territorial restrictions or market share
- Don't discuss ongoing litigation or threatened litigation
- Don't be silent if inappropriate topics are discussed... do formally object.

If you have questions,

contact the IEEE Patent Committee Administrator

at [patcom@ieee.org](mailto:patcom@ieee.org)

Approved by IEEE-SA Standards Board – December 2002

4

ModelSim at DAC 2003-Anaheim

# Agenda



## ■ Wednesday

- Welcome/Introductions
- PSL Overview & Thoughts on Integrating with VHDL – Erich Marschner
- Lunch
- Identify all aspects of VHDL impacted by ABV
- Dinner



# Agenda

## ■ Thursday

- Summary of ABV impacts to VHDL
- Identify LRM sections impacted
- Identify key corner cases (if any)
- Identify general approach/solution for each section
- Efficiency of implementation considerations
- Formal verification impacts
- Outline technical proposal



# Agenda

## ■ Friday

- Review technical proposal outline
- Assign sections of technical proposal for completion
- Discuss schedule for completing work



# Welcome/Introductions

## ■ Attendees

- Rob Anderson (arriving around 10 am)
- Stephen Bailey (Mentor/MTI)
- Dennis Brophy (Mentor/MTI)
- John Ries (Mentor/MTI)
- Erich Marschner (Cadence)
- Richard Wallace (Northrup Grumman)
- Jim Lewis (SynthWorks)



# PSL Overview & Thoughts on Integrating PSL with VHDL

Erich Marschner



# Top Level Questions

- What part of PSL to include into VHDL?
- How to define semantics?
- What other VHDL extensions are related?
- What other PSL extensions are in the works?
- API for access to assertion status?
- Where in the simulation cycle will assertions be evaluated?



# PSL Semantic Definition in VHDL

## ■ What portion of PSL to include?

- Simple combinational assertions
  - Always
  - Never
  - ->
- Sequences and properties
- Directives (assert and cover minimum; assume, restrict, etc. bonus)
- We will go with simple subset as defined in PSL LRM section 1.3.2.1

## ■ Equivalent process semantics

- Spawn new process to start each match?
- Define algorithm for translating into non-deterministic finite-state automata?
- Include PSL's formal semantics without providing an equivalent process semantic?
  - This is the preferred approach



# PSL Semantic Definition in VHDL

## ■ Additional semantics

- Keep sequential assertions simple boolean (combinatorial)
- New temporal assertions must be concurrent assertions
- Variables referenced in a sequence/property does not cause “evaluation” of a directive. Only signals (objects with events) can trigger evaluation.
- Time ticks (time advancement):
  - Clock edge if clocked
  - Unlocked: Need to enhance PSL semantics to what we want in VHDL
    - Defined only within the context of the property
    - Changes of any object referenced in the property advances time.
      - » Multiple object value changes in the same simulation cycle constitute a single advancement of time
- When are properties/assertions evaluated?
  - At boundary between end of one cycle and beginning of the next.
  - Values must be stable
  - We need to choose where we put it in chapter 12.6. Doesn't matter where as long as it is always either right before or right after a simulation cycle boundary
  - Unlocked would be on delta boundaries
  - Clocked are based on clock edge event and the cycle boundary corresponding to.



# Completeness of PSL

- Is there any functionality missing from PSL?
  - Parameterizing vunits?
    - Requirements: What's best use model for users?
      - Generic checkers (typically multiple assertions) that are easy to reuse in multiple contexts (e.g., bus protocols)
      - Need to map design-specific names to checker names
    - Use existing VHDL entity (assertions are passive, no architecture required)
      - Can you put generates, blocks, etc in an entity?
      - We may need the enhancement of eliminating need for architecture
    - or procedures to provide packaging?
    - Preserve ability to have separate PSL file for generic VIP that is HDL-independent?
  - Generate equivalent? Yes, PSL has *forall* construct
  - Default abort (do we want to add?)
- -> could be extended to be ternary to include else part
  - Boolean implication only!



# Completeness of PSL/VHDL

- **Add postponed assert: Illegal if clock is specified. (Default clock is automatically ignored.)**
  - Define a simulation constant that is the min simulation tick to allow assign after min tick (flip-flop output assignments) – Fast-Track/MP/performance issue; not assertion issue.
  - Could define “unclocked” to be TRUE so @unclocked = @TRUE which means unclocked (when there is governing default clock)



# VHDL Impacts from ABV

- Eliminate “– psl” for embedded PSL
- Reading OUT mode ports
- Signal expressions in port maps
- True/False implicit conversions (PSL defines these):
  - FALSE:
    - Bit('0')
    - Boolean(FALSE)
    - std\_ulogic('0')
  - TRUE:
    - Bit('1')
    - Boolean(TRUE)
    - std\_ulogic('1')
- Exploit assert, assume and restrict for constraint solving & stimulus generation
  - How does the property/directive get imported into the TB?



# VHDL Impacts from ABV

- **Hierarchical object referencing? (Signal Spy sufficient?)**
  - Additional delta cycle?
- **Exploit cover (and explicit/implicit assert) for defining standard functional coverage statistics reporting**
- **Coverage:**
  - Enhance VHPI to provide calls for getting coverage statistics
  - Coverage for directives only? Directives and properties? Directives, properties and sequences?
  - Cadence willing to donate their VHPI extensions
- **Support for reactive/adaptive testbench development**
  - Provide standard functions to query coverage, statistics and/or state information about directives (, properties and sequences)?



# VHDL Impacts from ABV

- **Standard functions to control directives**
  - Enable and disable
    - Enable in the middle of a sequence?
  - Disable after limit success/failure encountered
- **Reuse capabilities (declaring vunits in packages and how to bind?)**
- **Need for a formal verification subset?**
- **Declare sequences and properties in either declarative or statement parts (concurrent contexts)**
  - Named sequences and properties can be analogous to labels where the name is implicitly declared in the declarative part?
- **Are named endpoints a boolean-typed signal in VHDL?**
  - Yes (like guarded signals and SVAs)



# Language Organization

- **Declarations (sequence, endpoint, property, default clock)**
  - Any place a directive is permitted
  - Default clock is like a specification and does not create a name. (Default clock in a package is NOT visible to any design unit using that package. It only applies to the PSL within the package.)
  - Default clock in an entity can be “hidden” by a default clock in the architecture
- **Directives (assert, assume, restrict, cover, and \_guarantee versions)**
  - **Assert:** In entities, architectures, packages, package bodies (only if sequences/props can reference shared variables). Not permitted in configurations
    - NOTE: The above refers only to concurrent asserts and not sequential statement (combinatorial) asserts. Sequential statement asserts can be embedded in a subprogram body.
  - **Assume:** We won't support
  - **Cover:** entities, architectures, packages (package bodies?)
  - **Restrict:** entities, architectures



# Language Organization

## ■ Verification units (vunit, vprop, vmode)

- VHDL already has other packaging constructs so they may not be wanted/needed
- Decision: We will not include verification units in the VHDL LRM

## ■ Declaration part vs. statement part

- Allowing sequences, properties and endpoints to be declared in statement part begins blurring the lines between declarative and statement parts
- Non-symmetric to allow sequences, properties and endpoints in both parts, but restrict asserts to statement parts
- Plus other requests to allow attribute specifications and other declarations closer to use
- Should we investigate loosening or eliminating the differentiation between declarative and statement parts?
  - If so, what implications are there (labels would still probably require special rules to ensure the implicit label declaration occurs before any possible reference of the label)
- Not a critical item to be resolved for PSL/ABV
  - Nice to have, but PSL has tremendous value even if sequences, properties and endpoints are restricted to declarative part



# Action Items / To-Do

## ■ LRM Impacts

- Which portions of PSL LRM are incorporated (presumably by reference)
- Changes to concurrent assertion statements
- Changes to simulation cycle (chapter 12.6 ...) on when directives are evaluated
- Changes to declarative parts and statements grammars for various design units to allow properties, sequences, endpoints and/or directives.

■ Erich Marschner / Richard Wallace

## ■ PSL/Coverage VHPI functions

- Erich Marschner / Françoise Martinolle? And Richard Wallace

## ■ VHDL Standard packages with routines to access PSL/Coverage data

- Erich Marschner / Françoise Martinolle?



# Action Items / To-Do

## ■ Coordination

- Identify keywords to reserve for other teams' information (trailing underscore PSL keywords are problematic in VHDL)
- Review other potential language enhancements that would either improve PSL usage (and overall usage) and/or allow overall usage to exploit things being added from PSL
- Review against other teams' proposals to see if there are any potential gotchas or inconsistencies. (Biggest concern is TBV proposals.)

■ Steve Bailey