VHDL-200x Meeting Minutes (4 Mar 04)

Attendees:
Stephen Bailey
John Willis
Francoise Martinolle
John Shields
Peter Ashenden
Jim Lewis
Chuck Swart
John Ries
Nils Vander Zanden
Tim Schneider
Dennis Brophy

IEEE WG Patent and Topic Policy Slides

Need for WG Chair, Vice Chair and Secretary elections.

DASC needs WG roster (including voting and observer members) to ensure WG/DASC policies are adhered to and to ensure all members are covered under IEEE indemnification for standards work.

VHPI Status (see Peter's status report).
Review being done by various vendors. Mentor/MTI to investigate their ability to participate in the review.

Question of fast-track work interaction with VHPI.

Schedule of 200x fast-track and VHPI.

Need for a new PAR. Current PAR is 1076b which is an amendment PAR. Need a full revision PAR to incorporate fast-track revision scope along with VHPI amendment scope. General consensus to maintain individual membership and not go to organizational entity membership.

Schedule: VHPI: End of August/Early Sep LRM ready for submission as part of ballot package.

Goal: All fast-track items will fit within this schedule plus one month additional LRM editing work to integrate VHPI and fast-track into LRM.

Fast-Track proposal review.

General status at eda.org/vhdl-200x/vhdl-200x-ft.

Additional FT review comments/issues:

- FT1: Must cover implicit subprograms and not just implicit operators.
- Chuck volunteered to write up FT4 (min/max).
- Min/max will be functions and not operators for backward compatibility.
- FT5A: The to_string function should be implicitly declared for all types. Canonical form needs to be defined for composite types. to_ostring, to_hstring and to_dstring are implicitly defined for numerical types and one-dimensional arrays of character literal enum types (or whatever the proper wording is to cover bit_vectors, etc). to_string for all types could be an extension beyond FT, but FT work needs to allow upward compatibility for this. A related set of from_string operations may also be desired (also can be post-FT).
- FT5B: Issues with ambiguity when two literal operands are used with & and it will create ambiguous situations with existing code. Taken off FT.
- FT6: Closed (FT5A covers the needed capabilities).
- FT7: Hierarchical references would impact VHPI. Elaboration ordering needs to be addressed. Object aliased must be statically elaborated. Aliases that are hierarchical references are not evaluatable at elaboration time (cannot be used in statically elaborated declarations). Need LRM coverage of dynamically created drivers. If can't address the elaboration and driver issues, then fall back to signal spy type procedural solution. Today's aliases don't need constrained subtype indication. It would be desirable that hierarchical aliases should also allow an unconstrained subtype indication. Do not want to allow index or slice name within the hierarchical name string.
- Investigate if we can do a trial use amendment. In any case, FT7 may be appropriate for a trial use (along with PSL incorporation).
- FT7: Force and Release procedures: Do subsequent forces override previous or are they ignored? Need to be consistent with VHPI capabilities. Overriding/ignoring subsequent needs to be defined for VHPI as well. Cancel period default for signal force should be TIME'HIGH and the subtype is 0 ns. Force and Release should be definable via VHPI (put_value). Limit signal force to scalar types and 1 dimensional arrays of character type. String value is passed in. String name can be a selected and indexed name to handle record type elements.

  Bottom line is that these operations need to be defined in terms of VHPI capabilities. That part of the VHPI is not mature yet. VHPI team (Francoise and John Shields) will provide us with a proposal for force and release built on VHPI capabilities.
- FT8: Remove TREAD and TWRITE from the proposal.
- FT9: Add U to the size/signed bit string literal to explicitly denote an unsigned sized bit string literal. For decimal bit string literals, a size must be specified.
- FT10A: For fast-track, only address conditional assignment statement as a sequential statement. Defer ternary expressions to post-fast-track. Allow for variable assignments as well (optional for fast-track). VHPI implications for the uninstantiated (library) view, VHPI needs to cover this.

- FT10B: The selected signal assignment statement version of FT10A. Variable assignment equivalent here as well.
- FT11: If expression is globally static, behavior is same as 2002. If it is not a globally static expression, then you get the anonymous signal and concurrent signal assignment equivalence. Type conversion is part of the expression and not part of the port map (consistent with current language interpretation). VHPI implications for the anonymous signal assignment and implicit concurrent signal assignment.
- FT12: Reading output ports affects forces, especially through VHPI. Perhaps buffer ports should be deprecated. Some VHPI impact.
- FT13: Semantics of STOP, FINISH, RESET should be the same as corresponding VHPI control functions. Its not clear what RESET means during simulation. It could lead to infinite repetition if all state is reset. Suggest exit code for FINISH, perhaps also for STOP. Severity level might be appropriate. VHPI impact: reconcile parameters.
- FT14: Suggest that for fast track we not allow partially constrained subtypes. Once you hit a constraint everything below it must be constrained. This has problems with complex type for fixed-point type. Some VHPI impact on information model for types. Examine syntax for completing partially constrained types.
- How to get at attributes of unconstrained types? Suggest 'ELEMENT_TYPE. Maybe use 'POS, 'VAL, 'IMAGE to reference record fields.
- FT15: Compare with Ada discriminated types, but very difficult to fast-track.
- FT16: Default context clauses are problematic and should probably be dropped. Suggest allowing multiple context statements. VHPI impact must be analyzed.
- FT17: What is needed is some form of bundling such as interfaces. Withdrew.
- FT19: Possible problems determining drivers. Could place restrictions on impure functions or functions not locally declared. Suggest: limit for fast track, explore more general for future. Questions about names "_comb", "_latch" etc. Perhaps "latch" instead of "process" – better choice of keywords-function names. How do you check for coverage semantics of comb etc. Suggest keyword in sensitivity list. Combinatorial checks: all local variables must be written before read; all signals driven by the process must be driven under all control paths. Write this as a run-time check where the error must be reported if it occurs. And, state a design is erroneous if it depends on the fact that a control path that would result in the error will never be executed.
- FT20: Issue with overloading the use of the concatenation operators in this context (what if & is redefined?). Aggregate approach is doable with work. John Ries will own issue.