1450.4 meeting minutes – 11/02/11

Attendees: Ernie Wahl, Jim O'Reilly
Not present: Ajay Khoche, Markus Seuring, Oleg Erlich, Paul Reuter

Agenda:

- IEEE Meeting Preamble (No discussion of proprietary information).
- Continue with discussion of section 7.6, Variables and Expressions.

Summary:

- Line numbers are with regard to syntax document dated Nov 1, 2011
  - Line 2158: use comma separators for multiple attribute options
  - Line 2503: Any mathematical expression, quoted or unquoted, used to define a dimension shall be immediately evaluated to an integer, 0 or greater, at the instantiation of the array.
  - Line 2506: If only one (left-hand or right-hand side dimensions) is specified, the number of dimensions shall be required to match. Once set, array dimensions shall remain fixed.
  - Line 1855: Table 5 Large Data Structure Parameter Types
  - Line 2009: Table 6 Integral and Small Structure Variable and Parameter Types
- RECOMMENDATION: to use data type member functions, e.g., variable and parameter functions, append parentheses () to function name, e.g., a.Name becomes a.name() according to STIL.4 naming conventions (capitalization differentiates assignment from data retrieval). Parentheses differentiate data member from function member and simplifies lookup:
  - a.Name looks up object a and then object Name inside a.
  - a.name() looks up object a and executes its member function name()
  - Function parentheses clearly differentiate between an assignable data member and a function result (can't assign a value to the function result). Parentheses ensure backward compatibility should additional keywords or functions be added in phase II or other future revisions, e.g., there's no ambiguity or conflict between a.name and a.name() which refer to an object and a function respectively. Searches can differentiate, e.g., between type declaration String and function String() (converts data to string), or between units constraint setting keyword Units and function units() (extracts units in the form of a string).
  - Exclude STIL.0 data member selectors Meas, Min, Typ, Max
- **For reference only** – no consensus among the WG about which items (if any) from 1450.1-2005 to include as part of dot4. The list below was compiled by Jim O'Reilly
  - Must have: (applies to code which makes up a test program)
    - Variables - we've pretty well covered these with the dot4 variables, with the exception of the global integer variables
    - PatternBurst extensions (likely to be very useful)
    - Timing and Waveform table block extensions (add S and s CompareSubstitute event characters
    - PatternFailReport block
  - Nice to have: (applies to code which makes up a test program)
    - UserKeywords (definable on a per-block basis)
    - Signals (WFCMaps)
    - SignalGroups (WFCMaps)
    - Environment block
    - Pragma block
  - Not needed for test flow: (apply to pattern data only, which is expected to be processed separately)
    - ScanStructures (used to allow simulators to apply states to or read states from scan chains via "parallel-load" operations, rather than simulating the serial-shift-in or serial shift-out operations).
    - Pattern data extensions
    - Pattern statements
    - Procedure and macro data substitution

Actions:

- Jim to rework Verigy example from STIL paper so that the STIL translation is consistent with its original Verigy code.

Reference documents (If logged into your google account, can edit. If not, can only view.)

- Current Draft Syntax Document
- Issues List
- Namespace resolution examples document
- Communications spread sheet (scratchpad spreadsheet)
- Communications word doc (scratchpad word document)

Next meeting: 11/09/11

For reference STIL .4 information can be found at the IEEE STIL website: http://grouper.ieee.org/groups/1450/ (select the P1450.4 link from the table) or use the direct link http://grouper.ieee.org/groups/1450/dot4/index.html