1450.4 meeting minutes – 05/20/10

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

Agenda:

- IEEE Meeting Preamble (No discussion of proprietary information).
- Minutes from last week are now on the web.
- Discussion items
  - Discussion of the issues resolutions proposed by Ernie and Bruce, as summarized in Ernie's email of May 13, 2010 (Subject: Variables/Parameters Const/In?InOut?out). I'd like to close on and ratify as much of that set of proposals as possible.

    - Syntax element "Const" applied or not: what does it mean for various types, e.g., does it mean the same thing for Category or DCLevels as it does for Voltage ? Enumerate datatypes that are const by definition.

      Precept:

      Concept "constant" applies when a Test/Flow (including None, alias StdNoOp) triggered by an "On ..." event takes control until it relinquishes it.

      Test/Flow control is relinquished before or after Test/Flow execution or possibly during via a debugger breakpoint.

      (Stated another way, if a Test or Flow is NOT executing, it is permissible for tools which operate on STIL TestFlow data to modify "const" data – it's just not possible for STIL TestFlow code itself to modify such data).

      Precept:

      When an object is described as constant or mutable, it applies to that object only, i.e., it does not apply to contained objects, e.g., just because Category is constant doesn't mean that variables therein are.

      Recommendation:

      stil_data_type and BinSpec can't be used in Variables block, but only for Test/Flow Parameters. AGREED.

      Recommendation:

      Drop keyword Const from Parameter qualifier list: AGREED.

      Precept:

      Test Start/Stop control is local to test, hence PatternBurst/Pattern are intrinsically const

      Recommendation:

      PatternBurst/Pattern MUST be constant. AGREED

  - Additional notes (compiled by Ernie and Bruce) regarding which elements are constant, and which can be modified are shown below. Note that the data type BinSpec is discussed; as of this writing, the use of BinSpec as a data type and container for all levels of the bin hierarchy - BinDef (SoftBinDefs/HardBinDefs), BinGroup (Pass/Fail), BinAxis, and Bin - is not settled. We may continue to use BinSpec, or we may use a separate data type for each level in the hierarchy.

  - BinDef - Always constant
    - BinGroup - Always constant
      - BinAxis - Always constant
        - Bin - Always mutable with respect to **enable, disable**, set, clear, and related counters
          - Color - Always constant
          - Number - Always constant
          - Terse - Always constant
          - Verbose - Always constant
          - WafermapChar - Always constant
- PatternExec - Always constant
  - Timing - Mathematical expressions are constant, timing values resulting from evaluation are not constant
  - Selector - Always constant
  - (Spec) - Constant in part, see individual parts
    - Category - Constant in part, see individual parts
      - SpecVariable - Only Meas field is mutable, Min/Typ/Max fields are always constant
  - PatternBurst -
    - (PatList) - Pattern order in PatList is mutable,
      For adaptive testing: need per PatternBurst/Pattern fail counter, stand-alone flag
      - Pattern - Always constant
    - MacroDefs - Always constant. Remove from *stil_data_type*
    - Procedures - Always constant. Remove from *stil_data_type*
    - ScanStructures - Always constant. Remove from *stil_data_type*
    - Start - Always constant - Request PatLocation (address/cycle/label) and PatWindow (Start/Stop/SigGroup) variable types
    - Stop - Always constant - Request PatLocation (address/cycle/label) and PatWindow (Start/Stop/SigGroup) variable types
    - Termination - Always constant
  - DCLevels - Mathematical expressions are constant, timing values resulting from evaluation are not constant
  - DCSets - Always constant
- DCSequence - Always constant. Remove from *stil_data_type*
- Environment - Always constant - Want as TestProgram attribute (Sig-to-chan map). Remove from *stil_data_type*
- SigGroup - Always constant
- SignalGroups - Always constant - Want as TestProgram attribute. Remove from *stil_data_type*
- Signals - Always constant. Remove from *stil_data_type*

- From this discussion, the following data types are removed from *stil_data_type*:
  - MacroDefs, Procedures, ScanStructures, DCSequence, Environment, SignalGroups, Signals (AGREED)
- All other recommendations regarding mutable/constant as described above are accepted.
- Items needing additional discussion:
  - Request for PatLocation (address/cycle/label) and PatWindow (Start/Stop/SigGroup) variable types. Could be useful for adaptive testing.
  - Request for fail counter and enable flag per PatternBurst/Pattern (could be useful for adaptive testing).
  - Request for TestProgram block to contain reference to (optional?) Environment block, referenc to (optional?) SignalGroups block.

- Open issues - are there other open issues that should be considered? A review of the open issues list can guide us here.
  - Issues list: http://spreadsheets.google.com/ccc?key=0AoKiPr1I9LY9dF95dkhSTVVqOU5GbWJyWFNhY0JPX0E&hl=en
  - Namespace resolution examples document: http://docs.google.com/Doc?docid=0AYKiPr1I9LY9ZGY4dmNjNTNfMGZkOGJ2bmZy&hl=en
  - If logged into your google account, can edit. If not, can only view.
- Next Meeting 06/03/10.

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