1450.4 meeting minutes – 03/11/14

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

Agenda:

- Discuss proposal to make Device/Tester/TestHead/Partition/TestProgram a forward reference due to otherwise circular dependency between TestProgram block and Device block. One dependency (→) trail: TestProgram → Test → Timing → Device (for signals & signal groups, section 6.11.2.5)
- Lines 5253, 5264 – 5270: ref standardized mapping of PatternExec to a test-type.
- Lines 5328, 5349 – 5404, section 6.11.3.2, ref STIL.1/STIL.4 shared memory, special attention to initialization sequence.
- Annex A Name Spaces
- Annex C Event Sequence
- Annex F Block Sequence
- Section 6.9 Binning: review syntax after rescinding hard bin counters. See near line 3591.
- Seeking help with coding examples: writing coding examples is an excellent way to spot documentation short-comings. More participation equals more scrutiny. A complete production test program would be useful.

Summary:
Line numbers are from STIL.4 syntax document dated Mar. 10, 2014

- Discussion about the addition of Target TEST_TYPE_NAME to PatternExec block (as shown below)
  ```
  PatternExec (PAT_EXEC_NAME) {
       ( Target TEST_TYPE_NAME; )
       <snip, snip>
  }
  ```
  whose purpose is to specify the intended test-type (i.e., functional, search, DC, etc.) to be generated from the PatternExec
  - RESOLUTION: Defer addition of statement Target TEST_TYPE_NAME to PatternExec block to P1450.5 or ?
- Discussion about a standard mapping from PatternExec to StdFunctional:
  - ACCEPT: a PatternExec maps to a single STIL.4 StdFunctional
  - REQUIRES RESOLUTION: consider timing only then see if solution maps to levels:
    - how do we match 2+ Selectors with 2+ Spec/Category/variables ? Under consideration:
    - consider multiple cores sharing variable, e.g., period
    - no 2 selectors can refer to the same name
    - then variable names must unique among all spec.category hierarchies
- Discuss proposal to make Device/Tester/TestHead/Partition/TestProgram a forward reference due to otherwise circular dependency between TestProgram block and Device block.
  - This item generated a lot of discussion both at the meeting, and in post-meeting email discussions.
  - The seemingly innocuous proposed change to make the TestProgram statement contained in the Device block a forward reference to eliminate a circular dependency (as described above) caused some of the WG to ask for clarification about the purpose of the Device block, and its associated Chip and Package blocks.
  - From the text of the draft ballot documents:
    - A device (JO: as described in a Device block) consists of either a single chip or a package containing one or more chips including chip-to-package connections. At its core, this section describes a device and its connections to a tester
    - A Chip block defines per-chip signals and signal groups by reference to top-level named and unnamed Signal and SignalGroup blocks. The reference to unnamed Signal and SignalGroup blocks is implicit. The effect of having both named and unnamed blocks is additive. (JO: Note that named Signals blocks are a proposed new addition to P1450.4)
    - A Package block defines per package pins and planes.
    - A Device block describes a device in terms of its composition, i.e., one or more chips possibly packaged. Among other things, a Device block provides for:
      - signal to channel mapping

    - signal to package pin or plane mapping
    - per tester or test-head DCSequences
    - limited per test-head loadboard components
    - multi-site, MPW, and MCP testing
    - test program selection
  - The question was raised as to whether including the Device block (whose primary purpose is to provide enough information to ATPG - Automatic Test Program Generator – tools, which generate tests and flows from templates and descriptions such as that contained in P1450.4, to generate a test program, is within or outside the scope of P1450.4
  - In particular, the multi-project wafer (MPW) and multi-chip packages (MCP) support was questioned – and it's those capabilities that drive much of the complexity of the Device block (including references to Chip and Package blocks).
  - The question on the table is: Can we scale back this complexity, retaining perhaps only the regular structure of the ChannelMap statements, making that a named block defined at the top-level (multi-site mapping FOR THE SAME DEVICE ON ALL SITES) is supported. This question is still being discussed.

Action Items:

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

- http://spreadsheets.google.com/ccc?key=0AoKiPr1I9LY9dF95dkhSTVVqOU5GbWJyWFNhY0JPX0E&hl=en
- Namespace resolution examples document: http://docs.google.com/Doc?docid=0AYKiPr1I9LY9ZGY4dmNjNTNfMGZkOGJ2bmZy&hl=en
- Scratchpad spreadsheet: https://spreadsheets0.google.com/ccc?key=tQ93VDnAZ-Cl9RFKpPrPDzw&authkey=COzyro8K&hl=en&authkey=COzyro8K#gid=0
- Scratchpad "Word" doc: https://docs1.google.com/document/d/1zVu2M8nTJsrm0nFbBhiuM8-YRt4ErYqdy_uSa3x3_T4/edit?authkey=CLrgwrsG#

Next meeting: 03/18/14.

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