

## 1450.4 meeting minutes – 09/23/14

**Attendees:** Ernie Wahl, Jim O'Reilly, Ric Dokken, Greg Maston, Carey Garrenton, Scott Franzen, Benoit Nadeau-Dostie  
**Not present:** Paul Reuter, Markus Seuring, Oleg Erlich, Julia DiChiaro, Ajay Khoche, Mitsuo Fujii

### Agenda:

- Special presentation by Benoit Nadeau-Dostie of Mentor Graphisc to discuss Mentor's MBIST/memory repair flow methodology, and what capabilities P1450.4 needs to provide to support that methodology.

### Summary:

Line numbers if any are with regard to STIL.4 syntax document dated Sept. 22, 2014.

- Special presentation by Benoit Nadeau-Dostie of Mentor Graphisc to discuss Mentor's MBIST/memory repair flow methodology, and what capabilities P1450.4 needs to provide to support that methodology.
- An annotated version of the presentation slides will be prepared by Jim O'Reilly and Benoit, and posted to the dot4 website when reviewed and approved.
- A webcast of the presentation was made from GoToMeeting; it can be accessed at:
  - <ftp://roguevation.com>, username [wg1450@roguevation.com](mailto:wg1450@roguevation.com), password stilkraz.
  - See wmv webcast file 2014-09-23 08.09 IEEE 1450.4 9\_23\_2014.wmv
  - Note that in order to view this, you'll have to install the g2m wmv codec for Window Media Player. The installer for this codec can be found at the same site, filename g2m\_codec.exe
- The key points of the presentation, as it impacts P1450.4 are:
  - **All test setup and analysis is performed through TAP (scan in/scan out) operations.**
  - Setup and execution commands are shifted in the TAP; results shifted out the TAP. All comparisons are done via TAP readouts (i.e., comparing TAP TDO bits against expected data).
  - **There are multiple patterns and steps involved, but the pass/fail decision of each pattern (and a decision about what steps to take next) is determined ONLY by whether a particular pattern passes or fails. If any TDO shift-out bit mismatches, the test fails. Thus, stop-on-first-fail is acceptable for production testing. For debug, of course, one does need to know which bits pass or fail, but that's outside the scope of dot4. It's also important that one does NOT cycle power during pattern execution (obviously) or between pattern execution (doing so would upset the state of the part, likely causing loss of PLL lock, etc., and render the previous operations useless).**
  - Details:
    - When different memory BIST/repair controllers will have varying execution times (as in the case when memories are different sizes), the longest execution time is calculated ahead of time, and once the operations are started, the test program simply waits until the longest operation is complete before shifting out the TDR.
    - There are multiple memory BIST/repair controllers involved, and each has 2 bits (Done and Go) which dictate what to do next.
    - For N memory BIST/repair controllers, there is a 2\*N bit test data register (TDR) on the scan chain, accessed via the TAP controller. For example, with 10 individual controllers, there would be a 20-bit TDR. This 20-bit TDR would be read (scanned out), and each of the 20 bits is compared to a "1". If any bit is NOT a 1, then the test fails, and appropriate action is taken in the flow. Thus, stop-at-first fail is sufficient
    - The two bits checked are Go and Done.
    - Simple flow (see diagram 1 – Memory Repair (Basic Flow) below).
      - Pre-repair - 1=repairable, or no repair needed. (TP0, TP1.1, TP1.2, - powerup, and pre-repair LV and pre-repair HV tests as referenced in slide #3 "Memory repair (basic flow)" of the to-be-published presentation)
        - Done = 0 -> controller is bad. Fail.
        - Done = 1 and Go = 0 -> Fail.
        - Done = 1 and Go = 1 -> Pass.
      - Check if repair needed (TP1.3)
        - Check only Go (ignore Done): Go = 0 -> repair needed; Go = 1 -> no repair needed
        - Done
      - Repair: (TP2.2)
        - It's necessary to set the Vprog rail prior to repair (to allow fuse programming) – but as this happens prior to the beginning of pattern execution, it can be handled by existing STIL (1450.2) capabilities.

- Changing voltage levels DURING pattern execution appears to NOT be needed (though if it were, there are capabilities in 1450.2 to handle this situation).
- Fuse programming (for repair) – (scan) compress during programming, uncompress during verify
- Done = 0 -> controller is bad. Fail.
- Done = 1 and Go = 0 -> Fail.
- Done = 1 and Go = 1 -> Pass.
- Repair verify (TP2.3 (repair verify))
  - Go=0; repair failed; Go=1 -> repair passed
- Post-repair (TP3.2 (post-repair MBIST LV), TP3.3 (post-repair MBIST HV))
  - Done = 0 -> controller is bad. Fail.
  - Done = 1 and Go = 0 -> Fail.
  - Done = 1 and Go = 1 -> Pass.
- Each test point might iterate over multiple sets of conditions (clock frequencies, voltages). For example, note that at each test point (pattern execution), the diagrams show only a simple flow with one test execution per step. At each step, there's only one clock domain, and one set of timing in use. When multiple clock domains are involved, different sets of patterns would be run (having the same timing, but different data). For instance, given 2 clock domains, there would be a clock pin (or set of clock pins) per clock domain. Pattern A will have the clocks for clock domain A toggling, with the clocks for clock domain B static; pattern B will have the clocks for clock domain B toggling, with the clocks for clock domain A static
- Complex flow (see diagram 2 – Memory Repair (Complex Flow with binning) below).
  - Requires test program flags (flow variables) to be set after each test point.
  - Save repair status from TP1.13
  - Proceed to TP1.21 -> set flag ScB; remember flag.
  - Proceed to TP1.22
  - If tester can't set flags, need duplicate branches (based on pass/fail)
  - This is shown for single-site only; multi-site may add complexity, depending on how the tester implements multi-site operations.
- Do we need stop-on-first-fail or run-all-vectors mode in dot4 to support Capture Diag in slides 4/5?
  - This is needed, BUT it need not be handled in the context of production flows for 1450.4
- High-level: Deploy code to tester which describes flow module by module, or deploy as black-box? Which is preferable from Mentor's perspective?
  - We believe that the ideal scenario is to deploy the flow as a black-box, but it may be as an “IP-level” subflow which can be dropped into a larger flow. Either way, though, with Mentor's current methodology, P1450.4 has enough capability to support this WITHOUT the need for data capture/analysis.
- Benoit will check if there are any data-capture requirements remaining, for either memory BIST/repair, or mixed-signal BIST. Assuming there is not, we'll eliminate data-capture and associated syntax.
- Do we need stop-on-first-fail or run-all-vectors mode in dot4 to support Capture Diag in slides 4/5?
  - This is needed, BUT it need not be handled in the context of production flows for 1450.4
- **In summary, the “Capture Memory and Analysis” section (section 6.6.5) is not needed, and can be dropped from P1450.4. It's possible that it may need to be included in P1450.5, however.**

#### Action Items:

- Jim to arrange discussion with Mentor Graphics regarding data capture/fail memory analysis capability currently in P1450.4 for MBIST/memory repair flows. DONE.
- Ric Dokken to provide a complete production test program for Annex G.2. This program will include all elements (binning, test program, signal map, flows, tests, etc.) to the extent that Ric is able to.

#### 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-CI9RFKpPrPDzw&authkey=COzyro8K&hl=en&authkey=COzyro8K#gid=0>
- Scratchpad "Word" doc: [https://docs1.google.com/document/d/1zVu2M8nTJsrn0nFbBhiuM8-YRt4ErYqdy\\_uSa3x3\\_T4/edit?authkey=CLrgwrsG#](https://docs1.google.com/document/d/1zVu2M8nTJsrn0nFbBhiuM8-YRt4ErYqdy_uSa3x3_T4/edit?authkey=CLrgwrsG#)

Next meeting: 09/30/14.

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>

