P1450.4 meeting minutes - 12/16/03

Attendees: Dave Dowding, Jim O'Reilly, Ernie Wahl, Jim Mosely, Don Organ, Eric Nguyen, Jose Santiago, Tony Taylor
Not present: Doug Sprague

This meeting (a 2-hour meeting) was in addition to our regularly-scheduled meetings. We wanted to have some extra time since we'll be on break over the holidays.

Agenda:

- Update on STIL standards work-in-progress.
- Confirm/clarify diagrams and concepts of the attachment (describing the test program constructs).
- Review of minutes and email communications in the past week.

SUMMARY:

Tony Taylor gave an update on the progress of the various STIL standards (**Tony:** if there are any inaccuracies, please let me know - Jim).

P1450.4: (Test program flow)

Tony stated that the 1450.4 effort has been in progress for 5 years ongoing (though with a hiatus from 1999 through most of 2001). As such, the 5-year lifespan on the PAR has expired, and we (the working group) need to resubmit the PAR. Dave (and Tony?) will handle the resubmission.

P1450.5: (Test methods)

P1450.5 is currently not active. It will not restart until the .4 effort is complete.

P1450.1: (Semiconductor design environment)

Balloting group for P1450.1 is formed. Balloting will start shortly.

P1450.3: (Tester targetting)

The .3 WG wants to study the expression syntax resulting from the .1 WG before proceeding. We on the .4 WG will also want to look at expression syntax changes resulting from .1 for possible use in the .4 standard.

Tony also stated that he will be joining the .4 WG conference call when possible.

Dave Dowding stated that he feels we should be ready for ballot before ITC2004. Also, he'd like to have papers about .4 presented at ITC, if possible - so, authors, start thinking about that. The paper submission deadline this year is 02/14/04 (yes, Valentine's Day!)

The discussion today was focussed mainly on the test program layer, its behavior, and how it relates to some of the lower-level layers we've previously discussed.

Test program layer:

- We seem to be agreeing that, conceptually, the test program block is the top-level block or container for test program/testflow-related objects. We haven't settled on what it will be actually called, but the concept seems to be OK for everyone.
- The test program block is probably the object that would, in an actual implementation, interface to the tester OS.
- We also (I think) agreed that we should probably allow more than one test program block in a single STIL file (if we just allowed a single test program block in any single STIL file, one could, of course, have multiple test programs by having multiple STIL files . . .)
- One point still under discussion is whether the top level (the test program level) can be treated as a subflow or a test (which will, when executed, return a pass/fail result). I think we agree that it has

some of the characteristics of a subflow or test, but is fundamentally different from those types of objects.

- I believe that we're also agreeing to the utility of a variety of independent flows which can perform specific tasks (on-load, on-start, on-reset, on-alarm, etc.). The test program block would specify the name of a flow to use for a specific function (kind of like setting up a signal handler in a Unix C program . . .). Does the name **entry-point flow** work for these types of flows? Let's talk about this in the next meeting.
- I further believe that we're agreeing about the fact that having a set of these entry-point flows (whose execution is sequenced either by the tester OS in a specific implementation of STIL, or perhaps specified in some yet-to-be-defined STIL block), rather than having a single top-level flow which sequences procedurally through the various entry-point flows under control of yet-to-be-defined initialization flags (so that an on-start can't get executed until the on-load is done, for instance).