1450.4 meeting minutes – 04/01/10

Attendees: Bruce Parnas, Jim O'Reilly, Ernie Wahl, Oleg Erlich, Ajay Khoche

Not present: Markus Seuring , Greg Maston

Agenda:

- IEEE Meeting Preamble (No discussion of proprietary information).
- Minutes from last week are now on the web.
- Discussion items.
    - Next steps.
        - Let's make a list of everything we need to do to get a draft ballot document prepared. We won't discuss any of these in detail, just list the items and decide which ones are musts
            - Tester resource to signal mapping.
            - Axis block. Need for searches and shmoos.
            - For flows, use trees with nested flows (a la Verigy SmarTest flow code). We can represent that, but want to make P1450.4 syntax intuitive for representing these structures.
            - Ernie to provide a list of issues he's been keeping, and forward to the WG a list of issues still open.
            - Definition of functional test (two types – one type with single PatternExec parameter, and another with the components of PatternExec as parameters).
            - Do we need to define a required set of TestProgram global variables? If so, what should these variables be?
            - Scoping of variables, and potential conflicts of names – need to specify this in more detail.
            - Dot3 extension – legal characters for variable names, length of variable names, and reserved words for specific tester languages.
        - Discussion of these issues, along with those enumerated in Ernie's "Issues" email, dated March 31, 2010 (reproduced below). No conclusions, except that this list, along with the Google Docs Issues list (see below) are a fair and comprehensive representation of remaining open issues.

General

+ name spaces, i.e., which names can collide with others ? e.g. ids for Enumerated, BinDefs, BinMap, Variables, Test

---

Variables

+ Is it wise to use Variables keyword ? Confuses Pattern Variable space with TestProgram space.
Drop issue w having to parse dot1 syntax for dot4. How about keyword Symbols|SymTbl|SymDefs?
+ 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.

+ Address in dot3 ? Add Reserved symbol type to allow seeding symbol tables, see STIL.3, pg 40

+ Add Limits variable/parameter, e.g., "Limits x = 0ns < N <= 3ns;", "Limits y = 0ns < N <= None;", "Limits z = None;"

+ LHS "VAR_NAME[integer_expr] = expr;" should be "VAR_NAME[const_unsigned_expr] = expr;" ?

+ Consistent cell access notation for variable arrays, Spec/Category, Group/Axis/Bin:
  - an array is 1+ dimensional with each axis being either anonymous or named (associative)
  - each axis is indexed by integer or optionally by a string (associative)
  - a cell may be accessed by specifying N axis indices for an N dimensional array

+ Variables scoping precedence order: local, TestProgram block vars, Spec & unnamed global Variables block (from dot1)

+ Address Spec/Category/Variable arithmetic and array ops, e.g., "speca.cata.vara + 1ns", "speca.cata.vara + speca.cata.varb"

+ Array of type Compound: each element may have different units ?

+ Parameter Type: SpecVar|Compound VAR_NAME = 0s { Units 's'; } // SpecVar assigns Meas if not Const ?

+ NrwUnits syntax for type Compound and units restriction

+ Enumerated type: not in syntax document

+ String variable syntax: stringvar."Y" appends Y to stringvar contents
  "X"."Y" is dot0 syntax that evaluates to "X.Y", dot0, pg58
  need + concatenator ? "X"+"Y" evaluates to "XY"

+ Literals: review Boolean/Integer/Real initialization and math ops
  Is '1' of type Integer ? YES
  Is '1.0' of type Real ? YES
  Is '3/2' equal to '1' ? ****
  Is Boolean b = '10' legal ? YES
  Is Boolean b = '1.0' legal ? NO
  Do we support modulus op ? ****
  Do we support 'raise to the power' op ? NO

+ real_var_type initialization to None: allow/require units ? e.g., "Seconds s = Nones;", "Compound c = NoneA;"

+ Illegal variable names: any ? keywords ? Need explanation and/or list

+ BinSpec variable name needs resolution rules so it isn't confused with Bin names, e.g.,
  "BinSpec a = b;": if Bin b BinSpec b are both defined, what's b on the assignment RHS ?
     Would keyword Bin help, e.g., "BinSpec a = Bin b;", keyword All alleviates similar problem w Axis and Group
  When "BinSpec a = None;": what does a.Type return ? Bin ?
     Would keyword Constraint help, e.g., "BinSpec a = None { Constrain Bin; }" (constrain to Pass|Fail Bin|Axis|Group|BinDefs)

+ Clarify usage of related variable types
    SigGroup: dot4, parameter to pass Signal(s) and/or SignalGroup(s)
    SignalGroups: dot0, specifies SignalGroup name domain in PatternBurst
    Signals: dot0, what does this accomplish in dot4 ?

+ Clarify usage of variable type Environment: signal to tester channel mapping ?

+ SpecVariable (reference) initialization and assignment semantics:
  - Initialization RHS: variable under Spec/Category or another SpecVariable only
  - Assignment RHS: number w or w/o units, non-const Meas field only, units should match
  - Assignment RHS: Spec/Category or another SpecVariable copies Meas to non-const Meas field only, units should match

---

Binning

+ When a Bin is disabled, Axis.Size remains the same

+ Restrict retest max count and counter to hard bins only

+ Allow or require a NoBin mapping to cover the case where no soft bin has been set:
    Map None -> 5;

+ BinSpec: change to FailBinSpec & PassBinSpec ? or add property (see Variables section)

Test/TestFlow

+ On instantiation: should only be able to set/override parameters on Test/Flow. FlowNodes? i.e. Actions are immutable

+ Pass TestBase type Parameter to a Test/Flow ? e.g., StdLinearSearch: pass in varname, lolim, hilim, "test"

+ Move FlowNode statements inside TestExec {}: editor capability to move forward by or collapse matching braces

+ Variable TestType is also a keyword, Type is a keyword in dot1 (ref FILE_TYPE)

+ TPG3 Method/Type distinction, e.g., DCMEAS/VOH, what words do we use to distinguish ?

+ User-defined Test with FlowNodes has implicit base Flow instead of TestBase

+ Add Recurse or While Action with Boolean expression argument

+ FlowNode id scope local to Flow

TestProgram

+ Needs global-to-TestProgram SignalGroups specifier(s) to alleviate need to specify on every Timing or DCLevels block
  (want to use same blocks for package and wafer which may have different signals)

+ EntryPoint: can forward reference a Test/Flow instance but not a Test/Flow type ?
    BETTER YET, TestProgram makes a forward reference to a named EntryPoints block at the end (no forward references in EntryPoints)
+ Allow Variables reference only (no inline Variables)

==========

TestMethods

+ Functional test:
    - need to specify vector location in terms of address/cycle/label for start/stop/loop
    - need a run to last vector mechanism regardless of load size

- Spec should not be a PatternExec parameter. It is error prone. Category should be specified as "Category specname.catname"
- if default run starts at first vector, what does that mean if PatternBurst has Start and Stop specified ?
- Pattern attribute "Independent|Dependent" to allow/disallow reshuffling of Pattern sequence
  Independent: adaptive testing: persistent fail count ?
  Dependent: required precursor Pattern name(s) ?

+ Non-binning actions: assignment, skip_if, recurse_if, if-else_if-else

+ Define None instead of NoOpType (NoOp can be inherited from None)

+ Use StdFlow and StdNoOp instead of Flow and NoOpType

+ Write code for pulsewidth or clock phase measurements to be used henceforth from measurement, do in waveform table

==========

- 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 04/01/10.

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