

1450.4 meeting minutes – 05/07/09

Attendees: Bruce Parnas, Jim O'Reilly, Ernie Wahl

Not present: Ajay Khoche, Markus Seuring

Agenda:

- IEEE Meeting Preamble (No discussion of proprietary information)
- Meeting minutes from most recent meeting (04/30/09) are now posted on the web.
- Discussion of syntax document open issues list. Issues list (showing resolved and unresolved issues) is also on the web.
 - Walked through syntax issues list from the top. Resolved 6 issues (as outlined in issues #27- #32 of Google Doc open issues list - see link below).
- Continue discussion of binning. Most recent version of binning document is on the web.
 - In binning document, reviewed questions posed by the two footnotes at the bottom of page 1.
 - Teradyne has type error in addition to pass and fail. Handle errors via OnError entry point with no special bin category?
 - Resolution – defer to phase 2.
 - Explore the use of nested axes.
 - Resolution – not needed - dropped.
- Open issues - are there other open issues that should be considered? A review of the open issues list can guide us here.
 - (http://spreadsheets.google.com/ccc?key=pEI1-gPUmt2ZTw_kcCTgnKw&inv=jim_oreilly@ieee.org&t=933048453488551871&guest).
 - If logged into your google account, can edit. If not, can only view
- Other?
- Actions:
 - Jim:
 - Update syntax document with syntax changes regarding array access
 - Finalize proposals for handling spec/category/spec variable dot0-dot4 integration.
 - Jim/Ernie:
 - Update binning and syntax documents to reflect decisions made regarding binmap syntax.
 - Bruce:
 - Review and update TestBase proposals.
 - Distributed. Will be discussed in the coming weeks.
- Next Meeting 05/14/09.

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>

```

1 // =====
2 BinDefs bindefs {      // Soft bin definitions
3   Pass {
4     Axis ClockSpeed {    // Pass index 0
5       Bin "3.00GHz";     // ClockSpeed index 0
6       Bin "2.93GHz";     // ClockSpeed index 1
7       Bin "2.66GHz";     // ClockSpeed index 2
8     }
9     Axis CacheSize {     // Pass index 1
10      Bin "8Mb";          // CacheSize index 0
11      Bin "4Mb";          // CacheSize index 1
12    }
13  }
14
15  Fail {
16    Bin ContactOpens;
17    Bin ContactShorts;
18    Bin Functional;
19    Bin Timing;
20  }
21 }
22 // =====
23 // Consistent cell access notation for variable arrays, Spec/Category, Group/Axis/Bin:
24 // - an array is 1+ dimensional with each axis being either anonymous or named (associative)
25 // - each axis is indexed by integer or optionally by a string (associative)
26 // - a cell may be accessed by specifying N axis indices for an N dimensional array
27 // =====
28 // Adherence to rules adopted for Spec/Category
29 BinMap binmap1 {      // Soft to hard bin mapping
30   Map Pass.Axes[CacheSize].Bins["8Mb"] Pass.Axes[ClockSpeed].Bins["3.00GHz"] 1;
31   Map Pass.Axes[CacheSize].Bins["8Mb"] Pass.Axes[ClockSpeed].Bins["2.93GHz"] 3;
32   Map Pass.Axes[CacheSize].Bins["8Mb"] Pass.Axes[ClockSpeed].Bins["2.66GHz"] 3;
33   Map Pass.Axes[CacheSize].Bins["4Mb"] Pass.Axes[ClockSpeed].Bins["3.00GHz"] 2;
34   Map Pass.Axes[CacheSize].Bins["4Mb"] Pass.Axes[ClockSpeed].Bins["2.93GHz"] 4;
35   Map Pass.Axes[CacheSize].Bins["4Mb"] Pass.Axes[ClockSpeed].Bins["2.66GHz"] 4;
36
37   Map Fail.Axes[0].Bins[ContactOpens] 5;
38   Map Fail.Axes[0].Bins[ContactShorts] 5;
39   Map Fail.Axes[0].Bins[Functional] 5;
40   Map Fail.Axes[0].Bins[Timing] 7;
41 }
42 // -----
43 // Adherence to presumed associative array access rules
44 BinMap binmap2 {      // Soft to hard bin mapping
45   Map Pass[CacheSize]["8Mb"] Pass[ClockSpeed]["3.00GHz"] 1;
46   Map Pass[CacheSize]["8Mb"] Pass[ClockSpeed]["2.93GHz"] 3;
47   Map Pass[CacheSize]["8Mb"] Pass[ClockSpeed]["2.66GHz"] 3;
48   Map Pass[CacheSize]["4Mb"] Pass[ClockSpeed]["3.00GHz"] 2;
49   Map Pass[CacheSize]["4Mb"] Pass[ClockSpeed]["2.93GHz"] 4;
50   Map Pass[CacheSize]["4Mb"] Pass[ClockSpeed]["2.66GHz"] 4;
51
52   Map Fail[ContactOpens] 5;
53   Map Fail[ContactShorts] 5;
54   Map Fail[Functional] 5;
55   Map Fail[Timing] 7;
56 }
57 // -----
58 // Permitting abbreviated associative array access: only valid when all array indices are unique.
59 // NOTE: bracketed array access notation appears misleading in this context
60 // Will not work with ordinary 2dim array where each dim has an index of 0, for example
61 BinMap binmap3 {
62   Map ["8Mb"] ["3.00GHz"] 1; // Can only map to Pass[CacheSize]["8Mb"] Pass[ClockSpeed]["3.00GHz"]
63   Map ["8Mb"] ["2.93GHz"] 3;
64   Map ["8Mb"] ["2.66GHz"] 3;
65   Map ["4Mb"] ["3.00GHz"] 2;
66   Map ["4Mb"] ["2.93GHz"] 4;
67   Map ["4Mb"] ["2.66GHz"] 4;
68
69   Map [ContactOpens] 5; // Can only map to Fail[ContactOpens]
70   Map [ContactShorts] 5;

```

```

71 Map [Functional] 5;
72 Map [Timing] 7;
73 }
74 // -----
75 // Using hierarchical member notation (associative arrays only ?)
76 BinMap binmap4 {
77 Map Pass.CacheSize."8Mb" Pass.ClockSpeed."3.00GHz" 1;
78 Map Pass.CacheSize."8Mb" Pass.ClockSpeed."2.93GHz" 3;
79 Map Pass.CacheSize."8Mb" Pass.ClockSpeed."2.66GHz" 3;
80 Map Pass.CacheSize."4Mb" Pass.ClockSpeed."3.00GHz" 2;
81 Map Pass.CacheSize."4Mb" Pass.ClockSpeed."2.93GHz" 4;
82 Map Pass.CacheSize."4Mb" Pass.ClockSpeed."2.66GHz" 4;
83
84 Map Fail.ContactOpens 5;
85 Map Fail.ContactShorts 5;
86 Map Fail.Functional 5;
87 Map Fail.Timing 7;
88 }
89 // -----
90 // Using hierarchical member notation with abbreviated associative array access
91 BinMap binmap5 {
92 Map "8Mb" "3.00GHz" 1; // Can only map to Pass.CacheSize."8Mb" Pass.ClockSpeed."3.00GHz"
93 Map "8Mb" "2.93GHz" 3;
94 Map "8Mb" "2.66GHz" 3;
95 Map "4Mb" "3.00GHz" 2;
96 Map "4Mb" "2.93GHz" 4;
97 Map "4Mb" "2.66GHz" 4;
98
99 Map ContactOpens 5; // Can only map to Fail.ContactOpens
100 Map ContactShorts 5;
101 Map Functional 5;
102 Map Timing 7;
103 }
104 // =====

```