

1. Event List and Period Mapping

June 7, 2002, from Dan Fan

There is possible a need to have the optional event list mapping for each event list of each signal and period in WaveformTable of the STIL pattern file. These mapping will provide the possible of ATE's resource usage mapping for signal's timing sequence and period selection.

It is typical that single pattern can be used with multiple applications in a production or characterization test program for the specific device, e.g. speed binning, search of specific timing parameter of the device. It will trigger the need of using equation/specification/category to the corresponding timing values of the associated timing block for that pattern. In the other hand, a STIL pattern is typical generated with constant value of timing events and period value from an EDA tool.

The WaveformTable will provide the usage of event list (timing sequence) of each signal and the corresponding period value for a specific vector. If there are multiple WaveformTables used in one pattern and these WaveformTables may need different period values in special application (such as one period value can be stretched be longer period or shrunk to small period), then we will need a way to identify these specific period. If we will map each period in specific WaveformTable to different period memory on ATE, then we can not optimize the resource usage.

1.1 STIL Example (1450-1999)

Assume the period switching capability is needed to test the following device.

```

1: STIL 1.0
2:
3: Signals {
4:   DIR In;
5:   OE_ In;
6:   A0 InOut; A1 InOut; A2 InOut; A3 InOut;
7:   A4 InOut; A5 InOut; A6 InOut; A7 InOut;
8:   B0 InOut; B1 InOut; B2 InOut; B3 InOut;
9:   B4 InOut; B5 InOut; B6 InOut; B7 InOut;
10:} //end Signals
11:
12:SignalGroups {
13:  ABUS = 'A7 + A6 + A5 + A4 + A3 + A2 + A1 + A0';
14:  BBUS = 'B7 + B6 + B5 + B4 + B3 + B2 + B1 + B0';
15:  ALL = 'DIR + OE_ + ABUS + BBUS';
16:}
17:
18:Timing basic {
19:  WaveformTable one {
20:    Period '500ns';
21:    DIR { 01 { '0ns' D/U; }}
22:    OE_ { 01 { '0ns' U; '200ns' D/U; '300ns' U; }}
23:    ABUS { 01 { '10ns' D/U; }}
24:    BBUS { LHZX { '0ns' Z; '260ns' L/H/X; '280ns' T; }}
25:  } // W one
26:  WaveformTable two {
27:    Period '500ns';
28:    DIR { 01 { '0ns' D/U; }}

```

```

29:   OE_ { 01 { '0ns' U; '200ns' D/U; '300ns' U; }}
30:   ABUS { LHZX { '0ns' Z; '260ns' L/H/X; '280ns' T; }}
31:   BBUS { 01 { '10ns' D/U; }}
32: } // W two
33: WaveformTable three {
34:   Period '550ns';
35:   DIR { 01 { '0ns' D/U; }}
36:   OE_ { 01 { '0ns' U; '200ns' D/U; '300ns' U; }}
37:   ABUS { 01 { '10ns' D/U; }}
38:   BBUS { LHZX { '0ns' Z; '260ns' L/H/X; '280ns' T; }}
39: } // W three
40: WaveformTable four {
41:   Period '550ns';
42:   DIR { 01 { '0ns' D/U; }}
43:   OE_ { 01 { '0ns' U; '200ns' D/U; '300ns' U; }}
44:   ABUS { LHZX { '0ns' Z; '460ns' L/H/X; '480ns' T; }}
45:   BBUS { 01 { '10ns' D/U; }}
46: } // W four
47: WaveformTable five {
48:   Period '500ns';
49:   DIR { 01 { '0ns' D/U; }}
50:   OE_ { 01 { '0ns' U; '200ns' D/U; '300ns' U; }}
51:   ABUS { LHZX { '0ns' Z; '260ns' L/H/X; '280ns' T; }}
52:   BBUS { LHZX { '0ns' Z; '260ns' L/H/X; '280ns' T; }}
53: } // W five
54:} // Timing basic
55:
56:
57:Pattern basic {
58:  W five; V { ALL = 00ZZZZZZZZXXXXXXXX; }
59:  W one; V { ABUS = 00000000; BBUS = LLLLLLLL; }
60:  W three; V { ABUS = 10000000; BBUS = LHLLLLLL; }
61:  W three; V { ABUS = 00001000; BBUS = LLLLLHLL; }
62:  W two; V { DIR = 1; ABUS = LLLLLHLL; BBUS = 00001000; }
63:  W two; V { ABUS = LHLLLLLL; BBUS = 10000000; }
64:  W four; V { ABUS = LHLLLLLL; BBUS = 10000000; }
65:} // End Pattern basic

```

There is 2 different period values are used in five WaveformTable 'one' through 'five'. How should we map them to the ATE's period memory. Should we use five different period memory to allow any possible application? What if we only have four period memory? Should we just base on the different values, so map to two different period memory? How do we know the test program will stretch W three and W four together? What if they need be stretched separately?

Same issue for all the event lists of each signal - How do we map them to the ATE's resource and optimize the usage?

1.2 One Proposal to STIL Statements

The following modified Timing statement has the optional WaveformTable Container (a WaveformTable with no name) which will hold all the possible definitions of Period Label and Event List Label (with **bold face**). The regular WaveformTables (with name and can be referred in pattern) will use these pre-defined Period and Event List (with **angle bracket**).

```

66:Timing basic {
67:  WaveformTable {
68:    Period period_one: '500ns';
69:    Period period_two: '550ns';
70:    Period period_three: '550ns';
71:    DIR { sequence_one: 01 { '0ns' D/U; }
72:         sequence_two: 01 { '0ns' D/U; }}
73:    OE_ { only_one: 01 { '0ns' U; '200ns' D/U; '300ns' U; }}
74:    ABUS { sequence_one: 01 { '10ns' D/U; }
75:           sequence_two: LHZX { '0ns' Z; '260ns' L/H/X; '280ns' T; }
76:           sequence_three: 01 { '10ns' D/U; }
77:           sequence_four: LHZX { '0ns' Z; '460ns' L/H/X; '480ns' T; }}
78:    BBUS { sequence_one: LHZX { '0ns' Z; '260ns' L/H/X; '280ns' T; }
79:           sequence_two: 01 { '10ns' D/U; }
80:           sequence_three: LHZX { '0ns' Z; '260ns' L/H/X; '280ns' T; }
81:           sequence_four: 01 { '10ns' D/U; }}
82:  } // W container
83:  WaveformTable one {
84:    Period <period_one>;
85:    DIR { <sequence_one> }
86:    OE_ { <only_one> }
87:    ABUS { <sequence_one> }
88:    BBUS { <sequence_one> }
89:  } // W one
90:  WaveformTable two {
91:    Period <period_one>;
92:    DIR { <sequence_two> }
93:    OE_ { <only_one> }
94:    ABUS { <sequence_two> }
95:    BBUS { <sequence_two> }
96:  } // W two
97:  WaveformTable three {
98:    Period <period_two>;
99:    DIR { <sequence_one> }
100:   OE_ { <only_one> }
101:   ABUS { <sequence_three> }
102:   BBUS { <sequence_three> }
103: } // W three
104: WaveformTable four {
105:   Period <period_three>;
106:   DIR { <sequence_two> }
107:   OE_ { <only_one> }
108:   ABUS { <sequence_four> }
109:   BBUS { <sequence_four> }
110: } // W four
111: WaveformTable five {
112:   Period <period_one>;
113:   DIR { <sequence_one> }
114:   OE_ { <only_one> }
115:   ABUS { <sequence_two> }
116:   BBUS { <sequence_one> }
117: } // W five
118:} // Timing basic

```

With the resource reference of `<period_one>` through `<period_three>`, we do know WaveformTable one/two/five will always lock with the same period value while WaveformTable three and four will have their own period values and can be stretched or shrunk independently.

The DIR signal will have two different timing sequences to be reserved for these five WaveformTables. In other words, the DIR signal's timing can be modified differently even it has identical value in simulation.

The OE_ signal has one and only one timing sequence.

The ABUS and BBUS are both have four different sequence's labels which indicate the potential to be modified independently among different WaveformTables.

This extension also allow the de-coupling of simulation timing values from test program timing values and still provide proper mapping between them. In other words, it is possible that device's simulation timing is not intent to be used for testing purpose.