

## 1. 1450.3 Capabilities Tutorial (informative)

May 24, 2002, from Dan Fan, (Per-Pin Architecture)

There are four usage models to use the TRC statements (refer to Figure 1) - (1) Tester Rule Checking, (2) Tester Resource Reporting, (3) Tester Resource Targeting, and (4) Tester Resource Loading. The (2) and (4) are typically resident in the corresponding STIL pattern while the usage model (1) and (3) are stand-alone files.

This clause illustrates with a simplified model of a per-pin architecture tester for the usage model (1) and (3). This pseudo tester has 4 clock channels with 2 drive edges per period. The clocks have fixed format of either RTZ or RTO. The vector period is up to 200MHz (5ns). This tester also has 160 data channels with single event per vector period. The regular vector memory has 16 million vectors and 2K subroutine memory.

A simplified STIL pattern to exercise a subset of behavior for an octal bus transceiver design, modeled after a TTL LS245. Details of this design can be found in the Annex E. of IEEE std 1450-1999 standard. This simplified example is matched to the tutorial of IEEE std 1450-1999 (Clause 5 of IEEE std 1450-1999).

### 1.1 Test Rule Checking for a Tester Model

The TRC description to specify this tester which consists of 4 clock channels and 160 regular data channels for the usage model (1) and (3). Assuming the clock channels are inputs only and have 16 Meg vectors behind each channel. The data channels can be either inputs or outputs and also have 16 Meg vectors behind each channel. This tester has single period generator with both resolution and accuracy of 10ps for the usage model (1) and (3). Assuming the period value can be programmed between 5ns to 4us.

```

1: STIL 1.0 {
2:   TRC 2002
3: }
4:
5: Spec StructuralTester_vars {
6:   Category StructuralTester_category {
7:     PeriodVal {Min '5ns', Max '4us'}
8:   } //end Category
9: } //end Spec
10:
11:TRC StructuralTester_rules {
12:  PeriodCharacteristics PeriodInfo {
13:   Category StructuralTester_category;
14:   Accuracy '10ps';
15:   MaxPeriods 1;
16:   MaxPeriodGenerators 1;
17:   PeriodTimeLimit PeriodVal;
18:   Resolution '10ps';
19: } //end PeriodCharacteristics
20:
21:  WaveformCharacteristics ClockWav {
22:   Accuracy Edge '150ps';
23:   Accuracy EdgetoEdge '300ps';
24:   DriveEvents U D D/U 1;
25:   FormatSelect In {
26:     MaxShapes 2 Static; //Clock can be RTZ or RTO format

```

```

27:      MaxData Drive 2 Dynamic;
28:      MaxTimeSets 64 Dynamic { PerGroup 1024; }
29:    } //end FormatSelect
30:      MaxEvents Drive 1 ;
31:      Resolution '10ps';
32:    } //End WaveformCharacteristics ClockWav
33:
34:  WaveformCharacteristics DataWav {
35:    Accuracy Edge '500ps';
36:    Accuracy EdgetoEdge '1ns';
37:    CompareEvents Edge H/L/X 1;
38:    DriveEvents U/D/Z 1;
39:    FormatSelect InOut {
40:      MaxShapes 64 Dynamic;
41:      MaxData DriveCompare 3 Dynamic;
42:      MaxTimeSets 64 Dynamic { PerGroup 1024; }
43:    } //end Formats DataWav
44:    MaxEdgeTime '4 * PeriodVal';
45:    MaxEvents Drive 1;
46:    MaxEvents Compare 1;
47:    Resolution '40ps';
48:  } //End Data Waveform Characteristics
49:
50:  WaveformDescriptions ClockFormat Explicit {
51:    In RTZ_Format {
52:      Shape {
53:        D/U;
54:        RTZ_E2: D;
55:      } //end Shape
56:    } //end In RTZ_Format
57:    In RTO_Format {
58:      Shape {
59:        D/U;
60:        RTO_E2: U;
61:      } //end Shape
62:    } //end In RTO Format
63:  } //end WaveformDescriptions
64:
65:  Signals Clock {
66:    MaxSignals 4;
67:    DriveState U D;
68:    InOut Static;
69:    MaxVectorMemory '16*1024*1024';
70:    PeriodCharacteristics PeriodInfo Synch;
71:    WaveformCharacteristics ClockWav;
72:    WaveformDescriptions ClockFormat;
73:  } //end Signals
74:
75:  Signals Data In Out {
76:    MaxSignals 160;
77:    CompareStrobe Edge;
78:    CompareState H L X;
79:    DriveState U D Z;
80:    InOut WithinCycle;

```

```

81:   MaxVectorMemory '16*1024*1024';
82:   PeriodCharacteristics PeriodInfo Synch;
83:   WaveformCharacteristics DataWav;
84: } // end Signals
85: } // end TRC Rules

```

NOTE 5: through 9: - The **Spec** block is available in IEEE std 1450-1999 (as Clause 19). It is used to provide the corresponding period value range between 5ns to 4us.

NOTE 12: through 19: - The **PeriodCharacteristics** block contains all the property of the period generator of the corresponding ATE or ATE family. This specific ATE has single period generator with 10ps accuracy and resolution. The period value range is bounded within 5ns to 4us. It also calls out that the tester only has single period value. In other words, there is no period values switching on-the-fly capability.

NOTE 21: through 32: - The **WaveformCharacteristics ClockWav** block contains all the clock channel's characteristics of this tester. The clock channels have fixed formats - RTZ or RTO with 150ps edge accuracy. There are 64 timing sets available for each clock channel.

NOTE 34: through 48: - The **WaveformCharacteristics DataWav** block contains all the data channel's characteristics of this tester. The data channel only has single event per cycle with 500ps edge accuracy. There are 64 timing sets available for each channel.

NOTE 50: through 63: - The **WaveformDescriptions** block defines the clock channel's formats of this tester.

NOTE 65: through 73: - The **Signals Clock** block defines the corresponding property of the clock channels. In the usage model (1) Tester Rules Checking, it defines the clock channels of a particular tester's capability. In the usage model (3) Tester Resource Targeting, it defines the clock channels of a target tester (the configuration information). In this context, this tester has 4 clock channels available (**MaxSignals**). These channels can only have high (U) or low (D) state (**DriveState**) and can only support Input/Output state switch prior pattern execution (statically) - **InOut** statement. This particular tester has up to 16 Mega vectors behind each clock channel. The rest of statements refer to the period and waveform characteristics.

NOTE 75: through 84: - The **Signals Data In Out** block defines the corresponding property of the data signals. In the usage model (1) Tester Rules Checking, it defines the data channels of a particular tester's capability. In the usage model (3) Tester Resource Targeting, it defines the data channels of a target tester (the configuration information). This tester has 160 data channels available. The **CompareStrobe** statement defines this tester can only support edge strobe (i.e. no window strobe capability). These strobe events can have compare high (H), compare low (L) or don't compare (X) state (**CompareState**). the driver states can only have high (U), low (D), or don't drive (tri-state, Z) state (**DriveState**). This tester's data channel can support Input/Output state switch prior pattern execution (statically) - **InOut** statement. This tester has up to 16 Mega vectors behind each data channel.

## 1.2 Test Resource Usage in a STIL Pattern

The TRC statements that specify the pattern characteristics for this simplified TTL LS245 device will be located in the corresponding STIL pattern file. This simplified model has unidirectional bus - "A" bus signals are defined as inputs and "B" bus signals are defined as outputs. This example matches with the tutorial of IEEE std 1450-1999 Clause 5, refer to the Figure 3 of IEEE std 1450-1999 at page 9 for the STIL pattern. These TRC statements for the usage model (2) and (4) will represent as:

```

86: STIL 1.0 {
87:   TRC 2002
88: }
89:
90: TRC LS245_Resources {
91:   PeriodCharacteristics PeriodInfo {
92:     Accuracy '1ns';
93:     PeriodTimeLimit '500ns';
94:   } // end PeriodCharacteristics

```

```

95:
96: WaveformCharacteristics RTOWav {
97:   Accuracy Edge '500ps';
98:   DriveEvents U D/U 1;
99:   FormatSelect In {
100:     MaxShapes 1 Static; //OE_ needs be RTO format
101:     MaxData Drive 2 Dynamic;
102:   } //end FormatSelect
103: } //End WaveformCharacteristics RTOWav
104:
105: WaveformCharacteristics NRZWav {
106:   DriveEvents U/D 1;
107:   FormatSelect In {
108:     MaxShapes 1 Static; //DIR and ABUS only have RTZ format
109:     MaxData Drive 2 Dynamic;
110:   } //end Formats DataWav
111: } //End Data Waveform Characteristics
112:
113: WaveformCharacteristics WindowWav {
114:   CompareEvents h/l/t 1;
115:   DriveEvents Z;
116:   FormatSelect Out {
117:     MaxData Compare 3 Dynamic;
118:   } //end Formats DataWav
119: } //End Data Waveform Characteristics
120:
121: WaveformDescriptions RTOFormat Explicit {
122:   In RTO_Format {
123:     Shape {
124:       D/U;
125:       U '@+100ns';
126:     } //end Shape
127:   } //end In RTO Format
128: } //end WaveformDescriptions
129:
130: Signals Clock {
131:   DriveState U D;
132:   MaxVectorMemory 9;
133:   PeriodCharacteristics PeriodLS245 Synch;
134:   WaveformCharacteristics RTOWav;
135:   WaveformDescriptions RTOFormat;
136: } //end Signals
137:
138: Signals Data In {
139:   MaxSignals 9;
140:   DriveState U D;
141:   MaxVectorMemory 9;
142:   PeriodCharacteristics PeriodLS245 Synch;
143:   WaveformCharacteristics NRZWav;
144: } //end Signals
145:
146: Signals Data Out {
147:   MaxSignals 8;
148:   CompareStrobe Window;

```

```

149:   CompareState H L;
150:   MaxVectorMemory 9;
151:   PeriodCharacteristics PeriodLS245 Synch;
152:   WaveformCharacteristics WindowWav;
153: } // end Signals
154: } // end TRC Usage of LS245

```

NOTE 91: through 94: - The **PeriodCharacteristics** block contains all the requirements of period for the LS245's STIL pattern. The period value required by the DUT and this STIL pattern is 500 ns.

NOTE 96: through 103: - The **WaveformCharacteristics RTOWav** block contains all the signal OE\_'s characteristics of LS245. The OE\_ signal needs to have single RTO format with single timing values (200ns Down and 300ns Up).

NOTE 105: through 111: - The **WaveformCharacteristics NRZWav** block contains all the signal characteristics of DIR and ABUS of LS245. These signals only need have single event (or NRZ format) with single timing values (0ns or 10ns).

NOTE 113: through 119: - The **WaveformCharacteristics WindowWav** block contains all the signal characteristics of BBUS of LS245. The BBUS signal needs to have window strobe with single timing values (260ns - 280ns). It needs T state to turn off window strobe. It may need to have the **DriveEvents** statement, if the signal BBUS is declared as an InOut signal (as page 14 of 1450-1999). The **DriveEvents** statement can be omitted, if the signal BBUS is declared as an Out signal (as page 9 of 1450-1999).

NOTE 121: through 128: - The **WaveformDescriptions** block defines the OE\_'s formats of LS245.

NOTE 130: through 136: - The **Signals Clock** block defines the corresponding property of the signal OE\_. The **DriveState** statement describes this signal only need ForceUp (U)/ForceDown (D) states. In this example, it has 9 vectors for this signal (**MaxVectorMemory** statement). The rest of statements refer to the period and waveform characteristics. The keyword Synch indicates all signals always synchronous together.

NOTE 138: through 144: - The **Signals Data In** block defines the corresponding property of the data input signals - DIRS and ABUS. The **MaxSignals** statement indicates the STIL pattern has 9 input signals which will be treated as data inputs, e.g. the DIR and ABUS signals. It refers to a predefined WaveformCharacteristics named as "NRZWav".

NOTE 146: through 153: - The **Signals Data Out** block defines the corresponding property of the BBUS signals. The **MaxSignals** statement indicates the STIL pattern has 8 signals. The **CompareStrobe** statement defines the STIL pattern uses window strobe for the BBUS. The **CompareState** statement defines the STIL pattern only has compare high (H) and compare low (L) states. The **WaveformCharacteristics** statement refers to a predefined WaveformCharacteristics named as "WindowWav".

## 1.3 The Process of Tester Targeting

In the process of tester targeting, these two pieces of information will help the user to port a tester-independent STIL pattern to a targeted-tester loadable STIL pattern.

### 1.3.1 The Period's Statements

All the hard resource requirements must be met, such as: MaxPeriods and PeriodTimeLimit. Some of soft requirements may be compromised to fit to a tester, such as Accuracy and Resolution.

The following conditions are well covered by the pseudo tester:

- (a) **Accuracy:** The tester has 10ps accuracy. The STIL pattern only needs 1ns accuracy.
- (b) **PeriodTimeLimit** 500ns is within the boundary of 5ns to 4us range, so it is OK.
- (c) Others are not specified in STIL's PeriodCharacteristic block. These property will be treated as 'No Check' is needed.

### 1.3.2 The Signal's Statements

All the hard resource requirements must be met, such as `MaxSignals` and `MaxVectorMemory`. Some of soft requirements may be compromised to fit to a tester, such as `CompareStrobe`.

The following conditions are well covered by the pseudo tester:

- (a) **MaxSignals:** The tester has 4 clock channels and 160 data channels. The STIL pattern only needs 1 clock and 17 data signals.
- (b) **MaxVectorMemory:** The tester has 16 Mega vectors behind each channels. The STIL pattern only needs 9 vectors.
- (c) **DriveState:** The tester can handle U/D/Z while STIL pattern only uses U/D.
- (d) **InOut:** The tester can handle dynamic I/O switching within test cycle. The STIL pattern has only unidirectional bus (no I/O switching is needed, so no InOut statement is used).
- (e) **CompareState:** The tester can handle H/L/X while STIL pattern only uses H/L.

The following condition is not fully satisfied by the pseudo tester:

- (a) **CompareStrobe:** The tester only can provide edge strobe, but the STIL pattern requests to have window strobe. The user may relax the STIL pattern to use edge strobe, then a new STIL pattern will be generated with edge strobing. The user may insist to have window strobe, then he/she will search for another tester.

If it is decide to compromise the STIL pattern to fit this particular tester, then the Signals Data Out block will be modified as the following:

```

155: WaveformCharacteristics EdgeWav {
156:   CompareEvents Edge H/L 1;
157:   FormatSelect Out {
158:     MaxData Compare 2 Dynamic;
159:   } // end Formats DataWav
160:   MaxEvents Compare 1;
161: } // End Data Waveform Characteristics
162:
163: Signals Data Out {
164:   MaxSignals 8;
165:   CompareStrobe Edge;
166:   CompareState H L;
167:   MaxVectorMemory 9;
168:   PeriodCharacteristics PeriodLS245 Synch;
169:   WaveformCharacteristics EdgeWav;
170: } // end Signals

```