

```

1 // XLATION NOTE: Methods and parameters accepted at face value
2 TestProgram production {
3 // =====
4 // **** BINNING ****
5 // NOTE: presumes implicit STIL defined SoftBin NoBin
6 PassBins {
7 // Implicit anonymous Axis
8     SoftBin Good { Descriptor = "Good Device"; }
9
10    HandlerMap default {
11        HardBin 1 [Good];
12    }
13 }
14 // NOTE: reason for separating pass and fail maps:
15 // PassBins {
16 //     Axis freq {
17 //         SoftBin Indeterminate { Descriptor = "Good Device"; }
18 //         SoftBin 50Mhz { Descriptor = "Good 50Mhz Device"; }
19 //         SoftBin 100MHz { Descriptor = "Good 100MHz Device"; }
20 //     }
21 //     Axis pwr {
22 //         SoftBin 5% { Descriptor = "Good Device +- 5%"; }
23 //         SoftBin 10% { Descriptor = "Good Device +- 105"; }
24 //     }
25 //     HandlerMap default {
26 //         HardBin 1 [100MHz, 10%];
27 //         HardBin 2 [100MHz, 5%];
28 //         HardBin 3 [50Mhz, 5%|10%];
29 //         HardBin 4 [Indeterminate, *];
30 //     }
31 // }
32 FailBins {
33 // Implicit anonymous Axis
34     SoftBin Fail_Opens {}
35     SoftBin Fail_Shorts {}
36     SoftBin Fail_Icc { Descriptor = "Failed static Icc"; }
37     SoftBin Fail_InputLeakage {}
38     SoftBin Fail_dcStuckCore {}
39     SoftBin Fail_acPathCore {}
40     SoftBin Fail_Functional {}
41     SoftBin Fail_PLL {}
42     SoftBin Fail_Iddq {}
43
44     HandlerMap default {
45         // NOTE: logic presumes bit-wise behavior
46         HardBin 7 [ Fail_Opens | Fail_Shorts ];
47         HardBin 8 [!(Fail_Opens | Fail_Shorts)]; // Any FailBin other than these
48     }
49 }
50 // =====
51 // **** TEST METHOD DECLARATIONS ****
52
53 // NOTE: STIL.4 defined default. It's presence here is neither
54 // necessary nor harmful.

```

```

55 Method Test {
56     // STIL.4 defined parameters can not be replaced, only added to
57     InOut const SoftBin DefaultFailBin = NoBin; // Default
58
59     Out Bool    Failed; // Fail flag
60     Out Integer Status; // For integer or bit-wise arithmetic
61     Out NrUnits Result; // Scalar
62     // 1 to N dimensioned array not covered
63
64     // STIL.4 defined actions can be replaced
65     PreActions {}
66     PostActions {}
67     Fail OnCondition <anytest>.Fail {
68         BinAndStop; // Uses DefaultFailBin, set by derived Method (NoBin = no action)
69     }
70     Pass {} // No Condition (necessary|permitted)
71 }
72
73 Method Contact {
74     In SignalRefExpr pins;
75     In Amperes Current;
76     In Volts ShortsLimit = '200mV'; // Default
77     In Volts OpensLimit = '1V'; // Default
78     In Seconds Delay = '0s'; // Default
79     InOut const String Title = "Opens / Shorts Test"; // Default
80     // ? Diodes = VssDiode;
81
82     Private const Integer FailOpens = 1;
83     Private const Integer FailShorts = 2;
84
85     Execute Object<STIL or user defined method(s)>;
86
87     ExitPort Fail { // Method "Test" override (inherits condition)
88         if (<thistest>.Status | FailOpens) // Bit comparison
89             BinAndStop(Fail_Opens);
90         else if (<thistest>.Status | FailShorts) // Bit comparison
91             BinAndStop(Fail_Short);
92         else
93             Exception("Unexpected Contact Status: " + String(<thistest>.Status) + ".");
94     }
95 }
96
97 // Placeholders
98 Method Connect { <Assume to be STIL.5 declared> }
99 Method Current { <Assume to be STIL.5 declared> }
100 Method Delay { <Assume to be STIL.5 declared> }
101 Method Frequency { <Assume to be STIL.5 declared> }
102 Method IDDQ_PMU { <Assume to be STIL.5 declared> }
103 Method NullOperation { <Assume to be STIL.5 declared> }
104 Method Pattern { <Assume to be STIL.5 declared> }
105 Method PinMode { <Assume to be STIL.5 declared> }
106 Method Search { <Assume to be STIL.5 declared> }
107 Method Setup { <Assume to be STIL.5 declared> }
108 Method SpecContext { <Assume to be STIL.5 declared> }

```

```

109 Method Voltage      { <Assume to be STIL.5 declared> }
110 // =====
111 // **** DEFAULTS ****
112 // Defaults for the harness of FlowNode and/or any Method. Defaults
113 // are applied during object initialization.
114 Defaults {
115     // NOTE: STIL.4 defined default. It's presence here is neither
116     // necessary nor harmful.
117     FlowNode {
118         PreActions {}
119         PostActions {}
120         ExitPort Fail OnCondition test.Fail {
121             Goto end;      // End of parent Flow
122         }
123         ExitPort Pass {
124             Goto next;    // Next node in nodelist
125         }
126     }
127
128     // Add a parameter to STIL defined Method Test
129     Method Test {
130         InOut const String Title = "";
131     }
132 }
133 // =====
134 // **** TEST OBJECT DEFINITIONS/INSTANTIATIONS ****
135 // NOTE: named define-before-use instantiations only: anonymous
136 // instantiations may occur inline.
137 TestObject Flow Device_1 {
138     Title = "Stuck Fail";
139     Execute Method Setup {DCLevels = "NomLevels";}
140     Execute Method Delay {Delay = '50ms';}
141     Execute Method Connect{SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}
142     Execute Method PinMode{SignalRefExpr = 'PROGRAM';DriverMode = Low;ComparatorMode = NoChange;}
143     Execute Method Delay {Delay = '2ms';}
144     Execute Method PinMode{SignalRefExpr = 'PROGRAM';DriverMode = Off;ComparatorMode = NoChange;}
145     Execute Method Delay {Delay = '2ms';}
146     Execute Method Pattern{SignalRefExpr = 'JTAG_PINS-TDO';PatternExec = "Exec_stuck";BurstMode = True;}
147     Execute Method Connect{SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Remove;Delay = '0s';}
148 }
149 // -----
150 TestObject Flow Device_2 {
151     Title = "Good Device";
152
153     // XLATION NOTE: need more detail about scope of behavior of "ExecutionMode IgnoreAllFails"
154
155     Execute Method Setup {DCLevels = "NomLevels";}
156     Execute Method Delay {Delay = '50ms';}
157     Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}
158     Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Low;ComparatorMode = NoChange;}
159     Execute Method Delay {Delay = '2ms';}
160     Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Off;ComparatorMode = NoChange;}
161     Execute Method Delay {Delay = '2ms';}
162     Execute Method Voltage {SignalRefExpr = '"FPGA_OK"';Force Current = '10uA';Min V = '2V';Delay = '2ms';}

```

```

163     Execute Method Delay      {Delay = '2ms';}
164     Execute Method Pattern    {SignalRefExpr = 'JTAG_PINS-TDO';PatternExec = "Exec_GoodConfig";BurstMode = True;}
165     Execute Method Delay      {Delay = '2ms';}
166     Execute Method Voltage    {SignalRefExpr = '"FPGA_OK"';Force Current = '10uA';Min V = '-1V';Max V = '1V';Delay = '2ms';}
167     Execute Method Delay      {Delay = '2ms';}
168     Execute Method Connect    {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Remove;Delay = '0s';}
169 }
170 // -----
171 TestObject Flow Device_3 {
172     Title = "ac Fault";
173     Execute Method Connect    {SignalRefExpr = 'DELAYENABLE';Resource = PE;Action = Apply;Delay = '0s';}
174     Execute Method PinMode    {SignalRefExpr = 'DELAYENABLE';DriverMode = High;ComparatorMode = NoChange;}
175     Execute Method Setup      {DCLevels = "NomLevels";}
176     Execute Method Delay      {Delay = '50ms';}
177     Execute Method Connect    {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}
178     Execute Method PinMode    {SignalRefExpr = 'PROGRAM';DriverMode = Low;ComparatorMode = NoChange;}
179     Execute Method Delay      {Delay = '2ms';}
180     Execute Method PinMode    {SignalRefExpr = 'PROGRAM';DriverMode = Off;ComparatorMode = NoChange;}
181     Execute Method Delay      {Delay = '2ms';}
182     Execute Method Pattern    {SignalRefExpr = 'JTAG_PINS-TDO';PatternExec = "Exec_GoodConfig";BurstMode = True;}
183     Execute Method Connect    {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Remove;Delay = '0s';}
184 }
185 // -----
186 TestObject Flow Device_4 {
187     Title = "Chain Race";
188     Execute Method Setup      {DCLevels = "NomLevels";}
189     Execute Method Delay      {Delay = '50ms';}
190     Execute Method Connect    {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}
191     Execute Method PinMode    {SignalRefExpr = 'PROGRAM';DriverMode = Low;ComparatorMode = NoChange;}
192     Execute Method Delay      {Delay = '2ms';}
193     Execute Method PinMode    {SignalRefExpr = 'PROGRAM';DriverMode = Off;ComparatorMode = NoChange;}
194     Execute Method Delay      {Delay = '2ms';}
195     Execute Method Pattern    {SignalRefExpr = 'JTAG_PINS-TDO';PatternExec = "Exec_scrace";BurstMode = True;}
196     Execute Method Connect    {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Remove;Delay = '0s';}
197 }
198 // -----
199 TestObject Flow Device_5 {
200     Title = "Stuck Chain";
201     Execute Method Setup      {DCLevels = "NomLevels";}
202     Execute Method Delay      {Delay = '50ms';}
203     Execute Method Connect    {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}
204     Execute Method PinMode    {SignalRefExpr = 'PROGRAM';DriverMode = Low;ComparatorMode = NoChange;}
205     Execute Method Delay      {Delay = '2ms';}
206     Execute Method PinMode    {SignalRefExpr = 'PROGRAM';DriverMode = Off;ComparatorMode = NoChange;}
207     Execute Method Delay      {Delay = '2ms';}
208     Execute Method Pattern    {SignalRefExpr = 'JTAG_PINS-TDO';PatternExec = "Exec_scstuck";BurstMode = True;}
209     Execute Method Connect    {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Remove;Delay = '0s';}
210 }
211 // -----
212 TestObject Flow Device_6 {
213     Title = "Stuck & Broke";
214     Execute Method Setup      {DCLevels = "NomLevels";}
215     Execute Method Delay      {Delay = '50ms';}
216     Execute Method Connect    {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}

```

```

217 Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Low;ComparatorMode = NoChange;}
218 Execute Method Delay {Delay = '2ms';}
219 Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Off;ComparatorMode = NoChange;}
220 Execute Method Delay {Delay = '2ms';}
221 Execute Method Pattern {SignalRefExpr = 'JTAG_PINS-TDO';PatternExec = "Exec_stuck_broke";BurstMode = True;}
222 Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Remove;Delay = '0s';}
223 }
224 // -----
225 TestObject Flow DeviceConfig {
226     Title = "Device Programming";
227     // XLATION NOTE: don't know the meaning of "Action { ''; Exit=1; }"
228     // XLATION NOTE: don't know what "Type UNCONDITIONAL" is
229     Execute Object Device_1;
230     Execute Object Device_2;
231     Execute Object Device_3;
232     Execute Object Device_4;
233     Execute Object Device_5;
234     Execute Object Device_6;
235 }
236 // -----
237 TestObject Flow LeakageTests {
238     Title = "Leakage Tests";
239     DefaultFailBin = Fail_InputLeakage;
240     Execute Method Current {SignalRefExpr = "_in";Force Voltage = '0V';Min I = '-10uA';Max I = '10uA';Delay =
'800us';Measurement Sequence = MBB;Measurement Mode = Static;Max Expected Current = '100uA';}
241     Execute Method Current {SignalRefExpr = "_in";Force Voltage = '3V';Min I = '-10uA';Max I = '10uA';Delay =
'400us';Measurement Sequence = MBB;Measurement Mode = Static;Max Expected Current = '100uA';}
242 }
243 // -----
244 TestObject Flow dcStuckTest {
245     Title = "DC Stuck At Tests";
246     DefaultFailBin = Fail_dcStuckCore;
247     Execute Method Pattern {SignalRefExpr = "_so";PatternExec = "dcStuckAtExec";}
248 }
249 // -----
250 TestObject Flow acPathTests {
251     Title "Path Delay Tests";
252     DefaultFailBin = Fail_acPathCore;
253     Execute Method SpecContext {CategorySelectorExpression = Engineering+SpeedGrade;}
254     Execute Method Search {SignalRefExpr = '_po_-'SO[7]'"-'SO[10]";PatternExec = "acPathExec";ResourceSignals =
'"CLK";Search Start = '1/30MHz';Search Stop = '1/130MHz';Upper Limit = '1/InternalFreq';SearchExpression = "acPathDe-
lay";ResultVariable = #;}
255 }
256 // -----
257 TestObject Flow FunctionalTests {
258     Title = "Functional Tests";
259     DefaultFailBin = Fail_Functional;
260     Execute Method SpecContext {CategorySelectorExpression = Engineering+SpeedGrade;}
261     Execute Method Pattern {SignalRefExpr = 'fpins';PatternExec = "Exec_functional";}
262 }
263 // -----
264 TestObject Flow PllTests {
265     Title = "Phase Lock Loop Tests";
266     DefaultFailBin = Fail_PLL;

```

```

267     Execute Method SpecContext {CategorySelectorExpression = Engineering+SpeedGrade;}
268     Execute Method Frequency {SignalRefExpr = "ClkOut";Minimum Freq = '100MHz*(1-0.0002)';Maximum Freq = '100MHz*(1+
0.0002)';Measure Mode = Continuous;PatternExec = "Exec_pll";Output Mode = Average;ResultVariable = #;}
269 }
270 // -----
271 TestObject Flow IddqTest {
272     Title = "Iddq Tests";
273     DefaultFailBin = Fail_Iddq;
274     Execute Method SpecContext {CategorySelectorExpression = Engineering+SpeedGrade;}
275     Execute Method IDDQ_PMU {Test Pins = 'fpins';Meas Pin = 'IddqPin';DPS Pin = 'VCCINT';User Bits = 'Kidd';PatternExec
= "Exec_Iddq";Measurement Mode = Dynamic;Delay = '2ms';Max I = '80uA';}
276 }
277 // -----
278 TestObject Flow Main {
279     // XLATION NOTE: didn't know what to do with Diodes = VssDiode;
280     Execute Method Contact {SignalRefExpr = 'all_pins';Current = '300uA';ShortsLimit = '200mV';OpensLimit = '1V';Delay =
'300us';}
281     Execute Object DeviceConfig;
282     Execute Object LeakageTests;
283     Execute Object dcStuckTest;
284     FlowNode {
285         Execute Object acPathTests;
286         ExitPort Pass { Goto 51; } // Inherit condition, override actions
287     }
288     FlowNode 41 {
289         Execute Object FunctionalTests;
290         ExitPort Pass { Goto 15; } // Inherit condition, override actions
291     }
292     FlowNode 51 {
293         Execute Object PllTests;
294         ExitPort Pass { Goto 41; } // Inherit condition, override actions
295     }
296     FlowNode 15 {
297         Execute Object IddqTest;
298     }
299     // XLATION NOTE: xlated intent, i.e., setting SoftBin PassAll is
300     // associated with Flow Main, not IddqTest or FlowNode 15
301     Pass { BinAndStop(PassAll); }
302 }
303 // -----
304 // NOTE: there is no reference to this flow
305 TestObject Connect Start {Title = "ConnectResources";SignalRefExpr = 'all_pins';Resource = PE;Action = Apply;Delay =
'0s';}
306 // -----
307 TestObject Flow Finish {
308     Title = "RemoveResources";
309     Execute Method Connect {SignalRefExpr = 'all_pins';Resource = PE;Action = Remove;Delay = '0s';}
310     Execute Method SpecContext {CategorySelectorExpression = Engineering+SpeedGrade;}
311     Execute Method Setup {DCLevels = "Zero";}
312 }
313 // -----
314 TestObject SpecContext Load {Title = "Spec Selection";CategorySelectorExpression = Engineering+SpeedGrade;}
315 // -----
316 // NOTE: there is no reference to this flow

```

```

317 TestObject Flow Unload {
318 }
319 // -----
320 // NOTE: there is no reference to this flow
321 TestObject Flow Load_dcFault1 {
322     Execute Method Setup {DCLevels = "NomLevels";}
323     Execute Method Delay {Delay = '50ms';}
324     Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}
325     Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Low;}
326     Execute Method Delay {Delay = '2ms';}
327     Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Off;}
328     Execute Method Delay {Delay = '2ms';}
329     Execute Method Pattern {SignalRefExpr = 'JTAG_PINS-TDO';PatternExec = FPGA_dcFault;}
330     Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Remove;Delay = '0s';}
331 }
332 // -----
333 TestObject NullOperation NoOp;
334 // =====
335 // **** ENTRY POINTS ****
336
337 EntryPoints {
338     // Using inline code instead of predefined SpecContext Load
339     OnLoad Execute Method SpecContext {Title = "Spec Selection";CategorySelectorExpression = Engineer-
ing+SpeedGrade;};
340     OnPatternLoad Execute Object NoOp;
341     OnStart Execute Object Main;
342     OnReset Execute Object NoOp;
343     OnPowerDown Execute Object NoOp;
344     OnFinish Execute Object Finish;
345     OnLotStart Execute Object NoOp;
346     OnLotEnd Execute Object NoOp;
347     OnWaferStart Execute Object NoOp;
348     OnWaferEnd Execute Object NoOp;
349     OnSiteStart Execute Object NoOp;
350     OnSiteEnd Execute Object NoOp;
351     OnMultiSiteEnable Execute Object NoOp;
352     OnMultiSiteDisable Execute Object NoOp;
353 }
354 // =====
355 }

```