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       HandlerMap default {  
10          HardBin 1 [Good];
11       }
12   }
13   // NOTE: reason for separating pass and fail maps:
14   // PassBins {
15       // Axis freq {
16       //     SoftBin Indeterminate { Descriptor = "Good Device"; }  
17       //     SoftBin 50Mhz { Descriptor = "Good 50Mhz Device"; }  
18       //     SoftBin 100MHz { Descriptor = "Good 100MHz Device"; }  
19       //   }  
20       // Axis pwr {  
21       //     SoftBin 5% { Descriptor = "Good Device +- 5%"; }  
22       //     SoftBin 10% { Descriptor = "Good Device +- 10%"; }  
23       //   }
24        // HandlerMap default {  
25       //      HardBin 1 [100MHz, 10%];  
26       //      HardBin 2 [100MHz, 5%];  
27       //      HardBin 3 [50Mhz, 5%|10%];  
28       //      HardBin 4 [Indeterminate, *];  
29       //   }
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_tddq {}  
43        HandlerMap default {  
44       // NOTE: logic presumes bit-wise behavior
45       HardBin 7 [ Fail_Opens | Fail_Shorts ];  
46       HardBin 8 [!(Fail_Opens | Fail_Shorts)]; // Any FailBin other than these  
47       }
48   }
49   // ==============================================================
50   // *** TEST METHOD DECLARATIONS ***
51   // NOTE: STIL.4 defined default. It's presence here is neither
52   // necessary nor harmful.

Method Test {
  // STIL.4 defined parameters can not be replaced, only added to
  InOut const SoftBin DefaultFailBin = NoBin; // Default

  Out Bool Failed; // Fail flag
  Out Integer Status; // For integer or bit-wise arithmetic
  Out NrwUnits Result; // Scalar
  // 1 to N dimensioned array not covered

  // STIL.4 defined actions can be replaced
  PreActions ()
  PostActions ()

  Fail OnCondition <anytest>.Fail {
    BinAndStop; // Uses DefaultFailBin, set by derived Method (NoBin = no action)
  }
  Pass {} // No Condition (necessary|permitted)
}

Method Contact {
  In SignalRefExpr pins;
  In Amperes Current;
  In Volts ShortsLimit = '200mV'; // Default
  In Volts OpensLimit = '0V'; // Default
  In Seconds Delay = '0s'; // Default
  InOut const String Title = "Opens / Shorts Test"; // Default
  // ? Diodes = VssDiode;
  Private const Integer FailOpens = 1;
  Private const Integer FailShorts = 2;

  Execute Object<STIL or user defined method(s)>;

  ExitPort Fail {
    // Method "Test" override (inherits condition)
    if ( <thistest>.Status | FailOpens) // Bit comparison
      BinAndStop(Fail_Opens);
    else if ( <thistest>.Status | FailShorts) // Bit comparison
      BinAndStop(Fail_Shorts);
    else
      Exception("Unexpected Contact Status: " + String(<thistest>.Status) + ");
  }
}

// Placeholders
Method Connect { <Assume to be STIL.5 declared> }
Method Current { <Assume to be STIL.5 declared> }
Method Delay { <Assume to be STIL.5 declared> }
Method Frequency { <Assume to be STIL.5 declared> }
Method IDDQ_PMU { <Assume to be STIL.5 declared> }
Method NullOperation { <Assume to be STIL.5 declared> }
Method Pattern { <Assume to be STIL.5 declared> }
Method PinMode { <Assume to be STIL.5 declared> }
Method Search { <Assume to be STIL.5 declared> }
Method Setup { <Assume to be STIL.5 declared> }
Method SpecContext { <Assume to be STIL.5 declared> }
Method Test {
Method Voltage { <Assume to be STIL.5 declared> }

// ================================================================
// **** DEFAULTS ****

// Defaults for the harness of FlowNode and/or any Method. Defaults
// are applied during object initialization.

Defaults {  // NOTE: STIL.4 defined default. It's presence here is neither
// necessary nor harmful.
  FlowNode {
    PreActions {}
    PostActions ()
    ExitPort Fail OnCondition test.Fail {
      Goto end;  // End of parent Flow
    }
    ExitPort Pass {
      Goto next;  // Next node in nodelist
    }
  }
  ExitPort Pass {
    Goto next;  // Next node in nodelist
  }
  // Add a parameter to STIL defined Method Test
  Method Test {
    InOut const String Title = "";
  }
}

// ================================================================
// **** TEST OBJECT DEFINITIONS/INSTANTIATIONS ****

// NOTE: named define-before-use instantiations only: anonymous
// instantiations may occur inline.

TestObject Flow Device_1 {
  Title = "Stuck Fail";
  Execute Method Setup {DCLevels = "NomLevels";}
  Execute Method Delay {Delay = '50ms';}
  Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}
  Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Low;ComparatorMode = NoChange;}
  Execute Method Delay {Delay = '2ms';}
  Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Off;ComparatorMode = NoChange;}
  Execute Method Delay {Delay = '2ms';}
  Execute Method Voltage {SignalRefExpr = "FPGA_OK";Force Current = '10uA';Min V = '2V';Delay = '2ms';}
}

TestObject Flow Device_2 {
  Title = "Good Device";
  Execute Method Setup {DCLevels = "NomLevels";}
  Execute Method Delay {Delay = '50ms';}
  Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}
  Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Low;ComparatorMode = NoChange;}
  Execute Method Delay {Delay = '2ms';}
  Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Off;ComparatorMode = NoChange;}
  Execute Method Delay {Delay = '2ms';}
  Execute Method Voltage {SignalRefExpr = "FPGA_OK";Force Current = '10uA';Min V = '2V';Delay = '2ms';}
}

// XLATION NOTE: need more detail about scope of behavior of "ExecutionMode IgnoreAllFails"
Execute Method Delay {Delay = '2ms';}
Execute Method Pattern {SignalRefExpr = 'JTAG_PINS-TDO';PatternExec = "Exec_GoodConfig";BurstMode = True;}
Execute Method Delay {Delay = '2ms';}
Execute Method Voltage {SignalRefExpr = "FPGA_OK";Force Current = '10uA';Min V = '-1V';Max V = '1V';Delay = '2ms';}
Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Remove;Delay = '0s';}
}

// ----------------------------------------------------------------
TestObject Flow Device_3 {
Title = "ac Fault";
Execute Method Connect {SignalRefExpr = 'DELAYENABLE';Resource = PE;Action = Apply;Delay = '0s';}
Execute Method PinMode {SignalRefExpr = 'DELAYENABLE';DriverMode = High;ComparatorMode = NoChange;}
Execute Method Setup {DCLevels = "NomLevels";}
Execute Method Delay {Delay = '50ms';}
Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}
Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Low;ComparatorMode = NoChange;}
Execute Method Delay {Delay = '2ms';}
Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Off;ComparatorMode = NoChange;}
Execute Method Delay {Delay = '2ms';}
Execute Method Pattern {SignalRefExpr = 'JTAG_PINS-TDO';PatternExec = "Exec_scrace";BurstMode = True;}
Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Remove;Delay = '0s';}
}

// ----------------------------------------------------------------
TestObject Flow Device_4 {
Title = "Chain Race";
Execute Method Setup {DCLevels = "NomLevels";}
Execute Method Delay {Delay = '50ms';}
Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}
Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Low;ComparatorMode = NoChange;}
Execute Method Delay {Delay = '2ms';}
Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Off;ComparatorMode = NoChange;}
Execute Method Delay {Delay = '2ms';}
Execute Method Pattern {SignalRefExpr = 'JTAG_PINS-TDO';PatternExec = "Exec_scstuck";BurstMode = True;}
Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Remove;Delay = '0s';}
}

// ----------------------------------------------------------------
TestObject Flow Device_5 {
Title = "Stuck Chain";
Execute Method Setup {DCLevels = "NomLevels";}
Execute Method Delay {Delay = '50ms';}
Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}
Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Low;ComparatorMode = NoChange;}
Execute Method Delay {Delay = '2ms';}
Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Off;ComparatorMode = NoChange;}
Execute Method Delay {Delay = '2ms';}
Execute Method Pattern {SignalRefExpr = 'JTAG_PINS-TDO';PatternExec = "Exec_scstuck";BurstMode = True;}
Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Remove;Delay = '0s';}
}

// ----------------------------------------------------------------
TestObject Flow Device_6 {
Title = "Stuck & Broke";
Execute Method Setup {DCLevels = "NomLevels";}
Execute Method Delay {Delay = '50ms';}
Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}
Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Low;ComparatorMode = NoChange;}
Execute Method Delay {Delay = '2ms';}
Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Off;ComparatorMode = NoChange;}
Execute Method Delay {Delay = '2ms';}
Execute Method Pattern {SignalRefExpr = 'JTAG_PINS-TDO';PatternExec = "Exec_stuck_broke";BurstMode = True;}
Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Remove;Delay = '0s';}
// ----------------------------------------------------------------
TestObject Flow DeviceConfig {
Title = "Device Programming";
// XLATION NOTE: don't know the meaning of "Action { ''; Exit=1; }
// XLATION NOTE: don't know what "Type UNCONDITIONAL" is
Execute Object Device_1;
Execute Object Device_2;
Execute Object Device_3;
Execute Object Device_4;
Execute Object Device_5;
Execute Object Device_6;
}  // ----------------------------------------------------------------
TestObject Flow LeakageTests {
Title = "Leakage Tests";
DefaultFailBin = Fail_InputLeakage;
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';}
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';}
}  // ----------------------------------------------------------------
TestObject Flow dcStuckTest {
Title = "DC Stuck At Tests";
DefaultFailBin = Fail_dcStuckCore;
Execute Method Pattern {SignalRefExpr = '_so';PatternExec = "dcStuckAtExec";}
}  // ----------------------------------------------------------------
TestObject Flow acPathTests {
Title = "Path Delay Tests";
DefaultFailBin = Fail_acPathCore;
Execute Method SpecContext {CategorySelectorExpression = Engineering+SpeedGrade;}
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 = "acPathDelay";ResultVariable = #;}
}  // ----------------------------------------------------------------
TestObject Flow FunctionalTests {
Title = "Functional Tests";
DefaultFailBin = Fail_Functional;
Execute Method SpecContext {CategorySelectorExpression = Engineering+SpeedGrade;}
Execute Method Pattern {SignalRefExpr = 'fpins';PatternExec = "Exec_functional";}
}  // ----------------------------------------------------------------
TestObject Flow PllTests {
Title = "Phase Lock Loop Tests";
DefaultFailBin = Fail_PLL;
Execute Method SpecContext {CategorySelectorExpression = Engineering+SpeedGrade;}
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 = #;}

TestObject Flow IddqTest {
Title = "Iddq Tests";
DefaultFailBin = Fail_Iddq;
Execute Method SpecContext {CategorySelectorExpression = Engineering+SpeedGrade;}
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';}
}

TestObject Flow Main {
// XLATION NOTE: didn't know what to do with Diodes = VssDiode;
Execute Method Contact {SignalRefExpr = 'all_pins';Current = '300uA';ShortsLimit = '200mV';OpensLimit = '1V';Delay = '300us';}
Execute Object DeviceConfig;
Execute Object LeakageTests;
Execute Object dcStuckTest;
FlowNode {
ExitPort Pass { Goto 51; }  // Inherit condition, override actions
}
FlowNode 41 {
ExitObject FunctionalTests;
ExitPort Pass { Goto 15; }  // Inherit condition, override actions
}
FlowNode 51 {
ExitObject PllTests;
ExitPort Pass { Goto 41; }  // Inherit condition, override actions
}
FlowNode 15 {
ExitObject IddqTest;
}
// XLATION NOTE: xlated intent, i.e., setting SoftBin PassAll is associated with Flow Main, not IddqTest or FlowNode 15
Pass { BinAndStop(PassAll); }
// ----------------------------------------------------------------
TestObject Connect Start {Title = "ConnectResources";SignalRefExpr = 'all_pins';Resource = PE;Action = Apply;Delay = '0s';}
// ----------------------------------------------------------------
TestObject Flow Finish {
Title = "RemoveResources";
Execute Method Connect {SignalRefExpr = 'all_pins';Resource = PE;Action = Remove;Delay = '0s';}
Execute Method SpecContext {CategorySelectorExpression = Engineering+SpeedGrade;}
Execute Method Setup {DCLevels = "Zero";}
}
// ----------------------------------------------------------------
TestObject SpecContext Load {Title = "Spec Selection";CategorySelectorExpression = Engineering+SpeedGrade;}
// NOTE: there is no reference to this flow
TestObject Flow Unload {
}

// NOTE: there is no reference to this flow
TestObject Flow Load_dcFault1 {
  Execute Method Setup {DClevels = "NomLevels";}
  Execute Method Delay {Delay = '50ms';}
  Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Apply;Delay = '1ms';}
  Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Low;}
  Execute Method Delay {Delay = '2ms';}
  Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Off;}
  Execute Method Delay {Delay = '2ms';}
  Execute Method PinMode {SignalRefExpr = 'PROGRAM';DriverMode = Off;}
  Execute Method Delay {Delay = '2ms';}
  Execute Method Pattern {SignalRefExpr = 'JTAG_PINS-TDO';PatternExec = FPGA_dcFault;}
  Execute Method Connect {SignalRefExpr = 'JTAG_PINS+PROGRAM';Resource = PE;Action = Remove;Delay = '0s';}
}

TestObject NullOperation NoOp;

// **** ENTRY POINTS ****
EntryPoints {
  // Using inline code instead of predefined SpecContext Load
  OnLoad Execute Method SpecContext {Title = "Spec Selection";CategorySelectorExpression = Engineering+SpeedGrade;};
  OnPatternLoad Execute Object NoOp;
  OnStart Execute Object Main;
  OnReset Execute Object NoOp;
  OnPowerDown Execute Object NoOp;
  OnFinish Execute Object Finish;
  OnLotStart Execute Object NoOp;
  OnLotEnd Execute Object NoOp;
  OnWaferStart Execute Object NoOp;
  OnWaferEnd Execute Object NoOp;
  OnSiteStart Execute Object NoOp;
  OnSiteEnd Execute Object NoOp;
  OnMultiSiteEnable Execute Object NoOp;
  OnMultiSiteDisable Execute Object NoOp;
}

// =============================================================================