UseCase#4: Simple Complete Test Flow (with SubFlow)

Author: Conceptual Model Subgroup of Working Group

Informative statement

This use case represents a simple Test Program flow which has subflows

Priority ranking:

1. Must Absolutely required for Minimum Viable Product

1.0 Use Case #4: Simple "production" program flow

1.1 Priority - Must

1.1.1 Simplest straight through flow to be supported by P1450.4 extension

1.2 Assumptions/Prerequisites

1.2.1 This flow shows only the test program flow once all aspects of the program are loaded and tester requirements are initialized. If there are other flows to direct the loading of memories/registers and the initialization sequences, these must be discussed in an other Use Case.

1.2.2 No tester-to-DUT connect/disconnect sequences are encompassed in this flow.

1.2.3 This sequence is run from start to termination

1.2.4 This sequence is focused for single site testing

1.3 Pre-Conditions

1.3.1 This flow shows only the test program flow once all aspects of the program are loaded and tester requirements are initialized

1.4 Tasks/Scenario

1.4.1 <u>First FlowNode: A functional continuity test</u> is run, testing first for Shorts (this first TestObject is a Composite of a multiple test sequence (or subflow)

1.4.1.1 If this test passes, then the program flow passes to the Second FlowNode.

1.4.1.2 If this test fails:

1.4.1.2.1 Increment and check Consecutive Continuity Fail Counter.

1.4.1.2.1.1 If Consecutive Continuity Fail Counter does not exceed limit, then continue to run DC Continuity test.

1.4.1.2.1.2 If Consecutive Continuity Fail Counter exceeds limit, then terminate program flow execution and production run, issuing a warning.

1.4.1.2.2 Run the DC Continuity for Shorts and Opens on all pins in the continuity test list.

1.4.1.2.2.1 If this fails, bin the device as a Continuity fail and terminate the program.

1.4.1.2.2.2 If this DC Continuity Test passes continue to the Second FlowNode

1.4.2 Second FlowNode: A Basic Functional Test (wiggle) at relaxed levels and timing (Subflow entity). This may be a single Functional pattern or a PatternBurst of a list of patterns specific to verify the basic functionality and interconnectivity of the DUT/SoC.

1.4.2.1 If this test passes, then program flow passes to the Third FlowNode.

1.4.2.2 If this test fails:

1.4.2.2.1 Bin the device as a Basic Functionality fail and terminate the program.

1.4.3 Third FlowNode: At Speed Functional Test, IP Cores 1, 3, 6 and 7 (Subflow entity). This test may be composed of a test method that runs the same patterns (PatternBurst) at minimum and then maximum VDD. Fail bins have been established for Fails at each IP Core and VDD_Min or for VDD_Max possible.

1.4.3.1 If these tests pass, the program flow passes to the Fourth FlowNode.

1.4.3.2 If this test fails:

1.4.3.2.1 Determine the VDD min or max state of the test and Bin the device appropriately for the IP Core At Speed functional fail and terminate the program.

1.4.4 Fourth FlowNode: At Speed Functional Test, IP Cores 2, 4, 5, 8, 9 and 10. This test may be composed of a test method that runs the same patterns (PatternBurst) at minimum and then maximum VDD (differing from those in the Third FlowNode). Fail bins have been established for Fails at each IP Core and VDD_Min or for VDD_Max possible.

1.4.4.1 If these tests pass, then program flow passes to the Fifth FlowNode.

1.4.4.2 If this test fails:

1.4.4.2.1 Determine the VDD min or max state of the test and Bin the device appropriately for the IP Core At Speed functional fail and terminate the program.

1.4.5 Fifth FlowNode: Dynamic AC test(s) (i.e. Leakage, VOL/VOH, Output Leakage, etc.) The test is accomplished by running a functional setup pattern that stops at various points where pins are in the appropriate state to be tested. Then if the tests passes, the pattern is run progressively until all pins are tested.

Note: if this program being run at WaferSort, this test is not run and the program moves to the Sixth FlowNode.

1.4.5.1 If this run of the test program is at WaferSort, then go directly to the Sixth FlowNode.

1.4.5.2 If these tests pass, then program flow passes to the Sixth FlowNode.

1.4.5.3 If this test fails:

1.4.5.3.1 The device is Binned as a (AC_Test Name) fail and the program is terminated.

1.4.6 Sixth FlowNode: Operating Idd Test at VDD max. The device is setup to run at VDD max on a specific functional pattern and the Idd measurement(s) taken and tested to limits determined by the type of device selected at program load.

1.4.6.1 If this test passes, then program flow bins the device as Good and program flow terminates.

1.4.6.2 If this test fails:

1.4.6.2.1 The device is Binned as a Idd-VDD_Max fail and the program is terminated.

1.5 Results/Post-Conditions

1.5.1 Appropriate actions can be taken by the Tester-Handler/Prober interface software to bin/mark/map the device relative to the bin results.

1.5.2 Appropriate power down and disconnect functions are not shown in this flow but are available to the user to determine correct sequences of those events

1.5.3 Appropriate Tester-Prober/Handler interface interactions occur at appropriate points in the Device/Lot process. This may include handling of lot material and reports. These may also be selections determined at program load time.

1.6 Alternatives

1.6.1 The program can be loaded and setup to run as WaferSort, PackageTest, or QA_Test at load time.

1.7 Comments/Questions

1.7.1

1.8 P1450.4/STIL Syntax Example Using P1450.4/D16-Mar 1, 2006 Syntax Summary

1.8.1

1.9 Use Case#4 Block Diagram Representation

Figure #1: Block Diagram Flow Representation of Use Case #4