Pre-determining Comparative Tests and Utilizing Signal Levels to perform Accurate Diagnostics

Larry Kirkland
WesTest Engineering
810 W. Shepard Lane
Farmington, Utah 84025

Abstract: Standard diagnostic schemes don't do enough analysis to focus in on the actual cause of a test failure. Often measurements will be border-line on test sequences prior to an actual test failure. These border-line measurements can be used to aid in the determination of an actual fault. An actual fault and the associated test that should detect that fault can be deceiving. The specific test in question can pass but be right on the border-line. The failure might not show up in testing until a later test is performed. This later test assumes all the prior tests passed and therefore the circuitry associated with these prior tests is good. This is not necessarily the case if some of the output measurements prior to the actual failing test were right on the border-line.

What can we do? Take advantage of the order in which the faults are simulated[1]. We should structure our TPSs such that a review of preliminary tests should be evaluated before the R/R component list is presented. The review of the preliminary test can be rather straight forward. We can look at signals that are within 8-10% of the lower or upper limit. We can then use an inter-related test scheme to evaluate the test(s) that can be associated with the actual failing test. This paper will use an example TPS and show how a test scheme and an evaluation scheme can be used to determine the PCOF. The paper will show actual measurements and how these measurements can be evaluated to determine the actual cause of a failure.

Test Case

The test case shown in table 1 clearly indicates a correlation between border-line measurements and diagnostic determination.

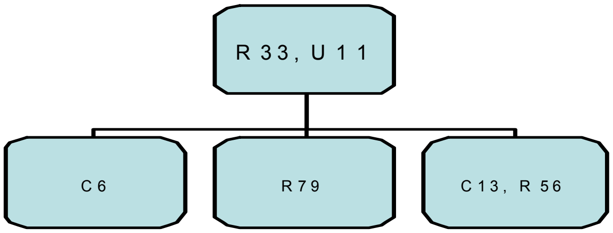
Table 1.

TEST #	LL	UL	MEASUREMENT	R/R
189004	4.04818	4.08887	4.04832	R79, U11, R33
200024	13.061	13.621	0.00228532	C6, R33, U11, C13, R56

The actual measurement for Test 189004 is 4.04832 which is a fraction of a percent higher than the lower level. Utilizing a system that identifies measurements which are close proximity to either an upper or lower limit then a correlation can be made between the actual failing test and a test which is in proximity of a limit.

Evaluation

An evaluation of the R/R callout can be made and a second list identified as the PCOF callout. The PCOF callout would indicate components that are redundant as the most likely defective part. In the case the PCOF callout would read R33, U11, R79, C13, and R56. The actual cause of this failure is in fact an out of tolerance R33. Graph 1 shows the probability based on redundant callout evaluation.



Graph 1. BORDER-LINE COMPARATIVE TESTS COMPONENT PROBABILITY GROUPING

Often test equipment measurements vary in milli-volts (see table 3) from tester to tester due to slight differences in impedance from routing hardware. These slight differences can cause precise measurements to fail on one tester and not on another tester. However, this does not necessarily cause a non-valid test because the precise measurements that will probably fail test similar components.

A force go selection from a test executive will show results of one test or multiple tests or all tests. These results can be compared from tester to tester. A useful debug menu (see table 2) from a test executive is shown below.

Table 2
Debug Menu

Debug menu commands

The Debug menu offers the following commands, which enable you to select or enable different debug tools during test program execution:

Watch Variable or 1553 data	Watch globally defined variables or 1553 data.
Stop Conditions	Stop program execution for specific conditions.
Single Step	Single step tests during program execution.
Force Test	Force a specific test to pass or fail.
Goto Test	Go to a specified test number.
Test Range	Define a range of tests to run.
Terminate Tests	Terminate test program after current test.

Tools

The Test Executive can be The Strength of Testing and Diagnosis[2] To merely run a test program and not provide the engineer and test technician with tools to debug a test program or analyze test results is of little value in a true test/diagnosis environment. Some still think ATE Test Programs are flawless and know all and can tell all. In fact most test programs are not robust enough to perform all the required diagnosis to properly fault isolate. Tools which allow the skill of the technician to analyze test data and control the flow of testing are absolutely critical. Indeed, substantial debug options are so critical to proper test/diagnosis that they should be mandated in all Test Executives.

Intricate Debug tools contribute more to the strength of a Test Executive than any thing else. The debug tools must be implemented with the developer and test program integrator in mind.. They are very useful, intuitive and easy to use.

Integration and development tools implemented must include run time commands, specific debug commands, a soft reset, system program execution and Soft Front Panels.

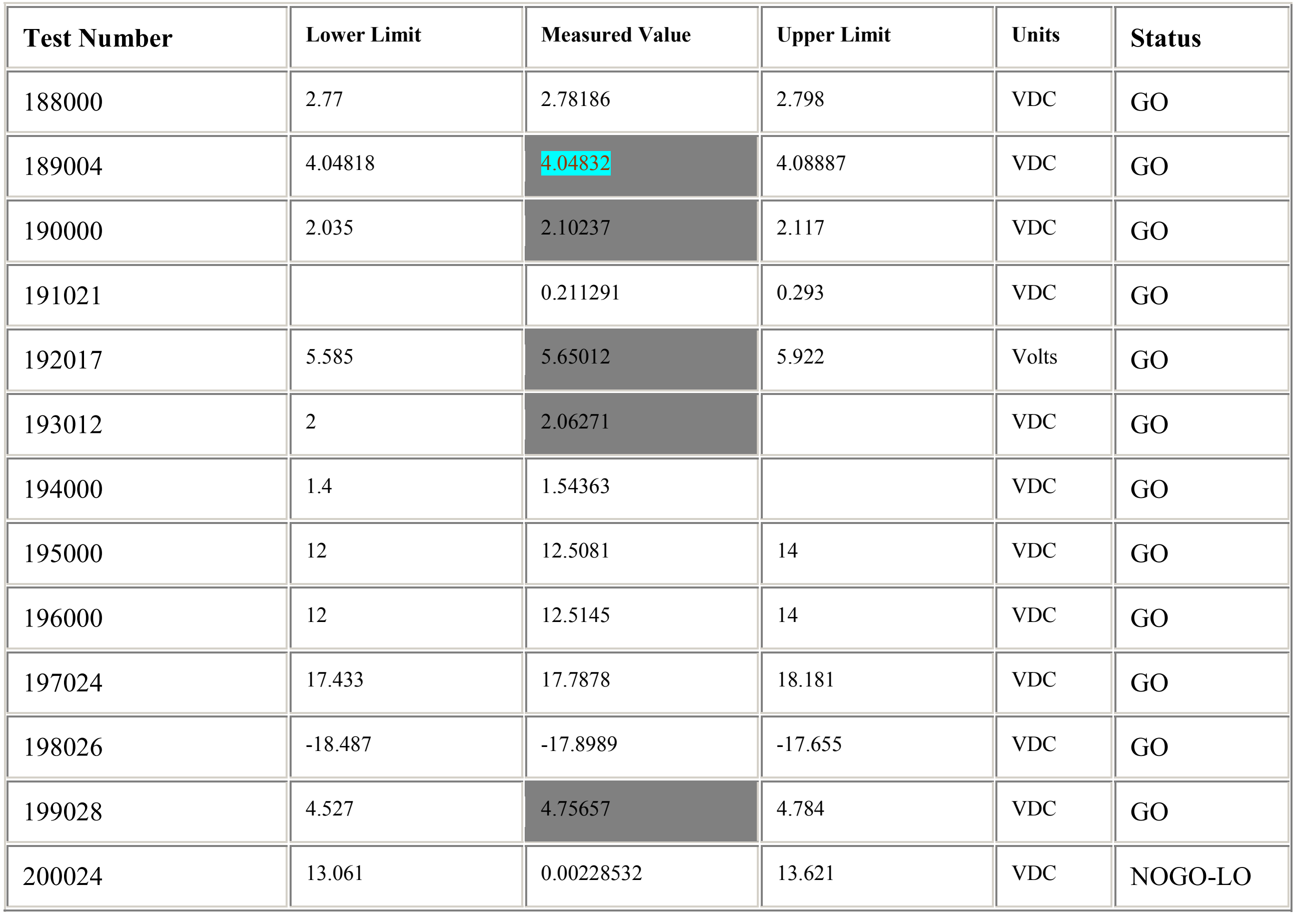
Table 3 clearly shows the measurements that are close to an upper or lower limit. This is clearly easy to perceive and is useful for TPS quality preview and test failure evaluation.

It is shown in Table 3 that various test results. The ATLAS test number is displayed in the sequence run and shows the program test flow. The Lower Limit is displayed first in the measurement sequence in the logical forward sequence. Next the actual measurement is shown; this is followed by the Upper Limit. This display sequence makes it easy to comprehend and contributes to the human analysis perception. Another key factor in the unit, often the unit factor is not included so the user or reengineer needs to refresh their memory as to what exactly the test is measuring. The final factor in the table is an indication of GO or NOGO. The NOGO factor is followed by an HI or LO to readily indicate the how the failed.

Providing a comprehensive indication of all the test results is critical for proper test sequence evaluation. The information is useful to both the technician and the engineer. Test/diagnosis information in the proper format is critical to not only PS development but to aid the technician in repair of the UUT and identify potential TPS problems.

Table 3

Test Results



Test Number	Lower Limit	Measured Value	Upper Limit	Units	Status
188000	2.77	2.78186	2.798	VDC	GO
189004	4.04818	4.04832	4.08887	VDC	GO
190000	2.035	2.10237	2.117	VDC	GO
191021		0.211291	0.293	VDC	GO
192017	5.585	5.65012	5.922	Volts	GO
193012	2	2.06271		VDC	GO
194000	1.4	1.54363		VDC	GO
195000	12	12.5081	14	VDC	GO
196000	12	12.5145	14	VDC	GO
197024	17.433	17.7878	18.181	VDC	GO
198026	-18.487	-17.8989	-17.655	VDC	GO
199028	4.527	4.75657	4.784	VDC	GO
200024	13.061	0.00228532	13.621	VDC	NOGO-LO

A soft reset has been provided by the Test Executive and is treated as if the hard reset button were pushed. This feature provides additional safety to the test system. The capability to perform test system analysis is often desired to test the integrity of the test system. Test System analysis applications include the system self test, system calibration and system BIT for all instruments. Having the capability to execute these specific system tests within the Test Executive enhances system reliability and maintainability.

The stop and go commands provide the operator the ability to stop and go at any time during test program execution. This is valuable when specific stop conditions are not known to the test program operator and can be invoked with a single touch of the escape key.

The Tools Menu shown in table 4 enables the user to select test station utilities and test program development applications. The Front panels provides an combo box to show the front panel capability. OAFI, BIT and Calibration are offered for selection. The SDE is the Software Development Environment. The SDE allows the engineer to enter the test program in a windows environment and then automatically generates the ATLAS program. The SDE is critical for TPS development in that it allows the test engineer to focus on the actual stimulus an measurements rather than coding.

Table 4

Tools Menu

Tools menu commands

The Tools menu offers the following commands, which enable you to select test station utilities and test program development applications:

Device Front Panels	Pops up a combo box listing devices which have front panel support.
Execute OAFI	Execute test stations OAFI.
Execute BIT	Execute test stations BIT.
Execute Calibration	Execute test stations calibration routines
SDE	Invoke the SDE application.
ATLAS Compiler	Invoke the ATLAS compiler.
Integration / Qual Notes	Pop up dialog box to enter integration or qualification notes.
Dump Files	Pop up dialog box to select dump file to view.

The selectable File Menu Commands as shown in Table 5 offers the user a variety of options. The Run command executes the TPS selected. The emulate command run the TPS without actually controlling the test equipment; this function is useful for test sequence debugging. The selection for the Recent Files prevents navigation repeats and helps the user prevent excessive time consumption searching for files. The Print and Print Setup are obvious and also provide the user with a readily accessible function. Another key element of the file menu is the Stop command which stops test program execution, this command is very useful for debugging purposes and for test and repair routines. The Go function allows the user to re-start the program at the point where the stop occurred. The station reset is an option which reset the station instrumentation; this is very useful when instrument when instruments have been setup improperly.

The navigation menu is critical is quick and responsive control of the TPS during development and test/repair. These facilities which were developed by actual test engineers promotes quality TPS development.

Table 5

File menu commands

File menu commands

The File menu offers the following commands:

Run	Runs a test program.
Emulate	Emulates a test program.
Recent Files	Displays most recently used test programs.
Print	Prints a document.
Print Setup	Selects a printer and printer connection.
Stop	Stops test program execution.
Go	Re-starts test program execution.
Reset...	Sends a reset to the test station.
Exit	Exits WT2000 Navigator.

Conclusion

Although it becomes somewhat obvious what needs to be done, all too often it is not done for the sake of the cost or mis-understanding. Techniques employed up-front in the TPS development process can save many dollars over the life of a weapon system in areas like changing good parts, man-hours, downtime for weapon systems, etc.

Overall, more flexibility and more money spent up front on TPS development will pay for itself during the overall weapon system life-cycle.

If we utilize our skill up-front and completely utilize knowledge gained as TPSs are developed and as units are tested, we can produce quality improvements. There is no question that the overall test performance and strength of a test system lays within the Test Executive. This is a case in which the Test Executive has the features that aid the test program developer. Providing a powerful Test Executive full of debug capabilities and integration tools is what is needed for efficiency and high level performance in the world of automated test equipment. Without a doubt, such a Test Executive has a big impact on diagnostics, reliability and maintainability. It is also important to understand that backward compatibility, built-in securities and ease of operator use is also essential for a Test Executive to be complete and maximize support for the test program developer.

References

1. Torroja, Y.; Riesgo, T.; de la Torre, E.; Uceda, J., **A comparative analysis of different fault simulation techniques forVLSI circuits testing,** Industrial Electronics, Control and Instrumentation, 1991. P1226 - 1231 vol.2

2. Kirkland, L..; Carlson, C., The Test Executive: the Strength of Testing and Diagnosis, AUTOTESTCON 2008, Track A