Stuck-Fault Tests vs. Actual Defects

Edward J. McCluskey and Chao-Wen Tseng
Center for Reliable Computing
Stanford University, Stanford, CA 94305
http://crc.stanford.edu

ABSTRACT

This paper studies some manufacturing test data collected for an experimental digital IC. Test results for a large variety of single-stuck fault based test sets are shown and compared with a number of test sets based on other fault models. The defects present in the chips studied are characterized based on the chip tester responses. The data presented shows that N-detect test sets are particularly effective for both timing and hard failures. In these test sets each single-stuck fault is detected by at least N different test patterns.

We also present data on the use of IDDq tests and VLV (very low voltage) tests for detecting defects whose presence doesn't interfere with normal operation during manufacturing test, but which cause early life failure.

INTRODUCTION

It was 1991, ITC was in Nashville, and a bunch of the boys — Bill Farwell from Hughes, Tushar Gheewalla from CrossCheck, Farzad Zarrinfar from LSI Logic, and Edward McCluskey from Stanford CRC — got together to plan an experiment.† We wanted to get actual tester data that would answer some questions about manufacturing test of digital ICs. The objective was to find out the relative effectiveness of different test techniques such as stuck fault tests, delay tests, *IDDq*, etc.

The approach of the project that resulted from these discussions was to design a chip that could be tested with as many different techniques as possible. The final design had four copies each of five different combinational circuits (called *CUTs for Circuits Under Test)*, two multiplier structures and three control logic structures. All of the CUTs are combinational since the main sponsor of the project was Hughes Aircraft Company which was committed to full internal scan path test. One of the CUTs has 12 inputs and the rest all have 24 inputs. We wanted to be able to test each CUT exhaustively (all possible input patterns) and one of them super exhaustively (all possible input transitions) in the belief that these would be complete tests that would detect all defective CUTs.††

The test chip, called the *Murphy chip*, was fabricated by LSI Logic in their LFT150K CMOS gate array technology (L_{eff} = 0.7 µ). Wafer sort was carried out at Digital Testing Services by first weeding out all dies with gross failures or failures in the non-CUT part of the chip (support circuitry). Over 300 tests were applied to each of the remaining approximately 5500 dies; 162 failed at least one of these tests. Approximately 300 of the dies were packaged, half of which passed all wafer sort tests and half of which failed at least one test. All these packaged dies were retested at Advantest and then approximately 100 were burned in for 366 hours at 130° C. The results of the test of the packaged parts did not differ significantly from the wafer sort results [2].

The only defects present in these chips are there as a natural result of the fabrication process; no defects were inserted artificially using lasers or FIBs, etc. Thus, we are able to compare, for the same input stimulus, the effects of these actual production defects on the ATE response with the effects of faults on the response predicted by simulation. In other words, we can compare defects with fault models. Let's start with the single stuck fault model.

SINGLE STUCK FAULTS

Most test patterns are generated using the *single-stuck fault (SSF)* model; so we thought it was important to check out the effectiveness of the SSF model. We collected 24 SSF test sets for each of the CUTs (a total of 120 test sets) using both academic (2 of them) and commercial (4 of them) automatic test pattern generators. Nine of the CUT test sets have 100% SSF coverage, five of them have less than 100% fault coverage, ten have more than 100% coverage. Yes, more than 100%! This will be explained later. But first a word about the single-stuck fault model.

---

† Readers unfamiliar with such gatherings are referred to the Robert W. Service poem titled "The Shooting of Dan McGrew."

†† An excellent project with similar objectives was sponsored by SEMATECH and reported on at VTS 97 [1]. The two projects are complementary since they differ in their approaches. The SEMATECH project used an existing production ASIC rather than a special design, applied far fewer test sets than the Murphy project, and collected data on many more chips than the Murphy project.

There are several different single-stuck fault models. All have something to do with the notion that some circuit lead has a signal value that is fixed and independent of any other signals in the circuit. We favor the SSF model in which:

(1) The circuit is modeled as interconnected elementary gates (AND, NAND, OR, NOR, INVERTER), and

(2) Two SSFs — a stuck-at-1 and a stuck-at-0 — can be present at each gate input and gate output.

A more complete discussion of this was presented at ITC'93 [3]. This interpretation of the SSF model leads to the following difficulty: if your netlist has more complex structures than elementary gates— for example, XOR gates or multiplexers or adders — then these have to be replaced by structures using only elementary gates for the purpose of pattern generation. This may be difficult to do; especially if you don't have access to accurate elementary gate equivalents for the complex structures. By the way, this is the obstacle to doing effective RTL test pattern generation. So what can be done? One approach is to synthesize elementary gate networks for the complex structures without knowledge of the actual implementation. This is imperfect and in our tests didn't do as well as using the accurate implementations; but it did better than the other option which is to assign stuck faults only to the inputs or outputs of the complex structures. This is sometimes called the *pin fault model.* The number of defective CUTs that escaped detection by test sets generated using the pin fault model was comparable to the escapes from test sets that have less than 100% fault coverage. These remarks are based on data presented at ITC'98 [2], and on updated data in Tables 1 and 2 which show the number of defective chips that escaped detection for each of the test sets generated using the SSF model. The details of Tables 1 and 2 will be discussed subsequently.

Table 1. Number of test escapes for SSF Test Sets with 100% or lower SSF coverage for the 116 defective chips.

				Escapes		
SSF tool	fault model	coverage	test length	characterized speed†	slow speed (1/3)	very slow speed (1/30)
1**	gate	100	427	4	7	9
2*	gate	100	313	3	5	7
3**	pin	100	857	3	6	7
	gate	100	1,000	2	5	7
4**	pin	100	456	6	8	9
	gate	100	603	2	4	5
5**	pin	100	466	6	9	9
	gate	100	571	4	6	8
6*	gate	100	547	3	7	7
	gate	99	532	4	8	8
	gate	98	490	6	9	10
	gate	95	467	10	12	13
	gate	90	427	15	17	18
	gate	80	334	20	23	23
weighted random*	gate	100	62.9K	2	7	
weighted random**	gate	100	21.9K	2	7	
exhaustive			16.7M	0	2	4

* academic ATPG tools ** commercial ATPG tools

† In previous publications the characterized speed was referred to as the rated speed. We use the term characterized here since it more accurately reflects the way in which the speed was chosen. The characterized speed for the multiplier CUTs was approximately 23 MHz and approximately 70 MHz for the control logic CUTs. The corresponding slow (very slow speeds) were approximately 7.5 (.75) MHz and 24 (2.4) MHz.

A related issue with the single stuck fault model is whether it accurately represents the effects of production defects on circuit operation. Our Murphy chip data showed that only about one third of the actual defects caused the affected CUT to respond to inputs with the same output responses as if a SSF was present in the CUT [4]. For each CUT test set, we simulated every possible SSF in that CUT† , collected the output response with that SSF present, and compared this response with the response on the tester from each defective CUT. The simulator output matched the tester output for only about one third of the defective CUTs. In our experiment, the single stuck fault model isn't very accurate. Since the SSF model was not accurate for 2/3 of the defective CUTs, one might suppose that test sets generated using this inaccurate model would miss as many as 66% of the defective CUTs. Of course, this was not what happened.

The results of applying SSF test sets with coverage between 80 and 100% are shown in Table 1. The test patterns were applied at three speeds: one, the *characterized speed,* was determined by characterizing good chips with Schmoo plots to determine that fastest speed at which the chip operated and then derating it with a 10% margin. The other two speeds are the *slow speed*, 1/3 of the characterized speed; and the *very slow speed*, 1/30 of the characterized speed.†† The reason for applying the patterns at slow and very slow speeds was to find out how many defects would be missed by using reduced speed tests. At characterized speed, the best 100% SSF test set missed only 2 (1.7%) of the defective CUTs and the worst 100% SSF test set missed 6 (5.2%) of the defective CUTs.

Only 2 (1.7%) of the defective CUTs escaped detection, but doesn't that correspond to 2 out of 116 or 17,000 dpm? Not really, since there were a total of 5500 dies; the correct number is about 360 dpm for the best 100% SSF test set and about 1000 dpm for the worst 100% SSF test set. Even these dpm values are higher than we would like. Also, when the test patterns were applied at slower speeds more defective dies escaped detection. Clearly other tests are in order.

MULTIPLE-DETECT TEST SETS

We wondered whether a test set that detected every stuck fault N times might be even more effective than a 100% SSF test set. Figure 1 shows a very simple circuit to illustrate this concept. There are three input patterns that detect the fault Z stuck-at-1: ABD = 001, 011, 101††† . If we were generating a test set to detect every SSF at least once (an 100% SSF test set) it would be sufficient to include any one of these input patterns in the test set. However, in a test set in which each fault must be detected at least **three** times all three of these patterns must be included. In such a test set each stuck fault in the circuit must be detected by three **different** test patterns. We call such a test set a *3-detect test set.*††††

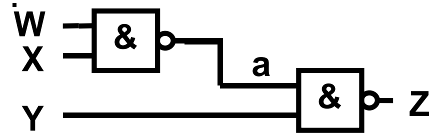

Figure 1. Circuit to illustrate N-detect test sets.

We were able to collect and generate 10 different N-detect test sets for the Murphy chip CUTs. The results of applying these test sets are shown in Table 2. This table shows that the test sets with multiple detects are more effective than the 100% SSF test sets for all three test speeds. The great success of the N-detect test sets was not predicted, and we are still trying to understand it. A specific example follows.

The circuit of Fig. 2 illustrates one possible situation in which a 3-detect test set would do better than an 100% SSF test set. This circuit shows a CMOS implementation of the Fig. 1 circuit in which there is a gate-to-source resistive short. For some values of the short resistance the response of the circuit to the three patterns for detecting Z stuck-at-1 could be as shown in Table 3. Here we are assuming that, when only one pmos pullup transistor in the WX gate conducts, the gate-source short lowers the voltage at **a** sufficiently so that the pmos **a** transistor conducts and holds the voltage at Z to a logic 1. It is also assumed that when both pmos pullup transistors in the WX gate conduct, then the **a** pmos transistor will not conduct and Z will be at logic 0 thus indicating the presence of the defect. The WXY = 001 test pattern must be present in a 3-detect test set thus guaranteeing that this defect will be detected. This test pattern could be missing in an 100% SSF test set which would thus miss the defect. This defect is interesting because its detection does not depend only on the inputs to the gate with the defect, but depends also on the inputs to the gate driving one of the defective gate's inputs. The defect acts somewhat like a neighborhood pattern sensitive fault in a RAM.

† The 6SQ CUT has 1480 collapsed SSF faults and 446 gates. The M12 CUT has 3842 collapsed SSF faults and 1146 gates. The ROB CUT has 1850 collapsed SSF faults and 898 gates. The SIM CUT has 682 collapsed SSF faults and 380 gates. The STD CUT has 650 collapsed SSF faults and 298 gates.

†† See footnote on page 2.

††† These patterns all also detect the faults C stuck-at 0 and D stuck-at-0, but this is irrelevant to the present discussion

†††† This circuit is so small that many of its SSFs are only detected by a single input pattern. Thus it is impossible to generate a complete 3-detect test set for this circuit.

Table 2. Number of test escapes for SSF test sets which detect each SSF more than once for the 116 defective chips.

	Escapes							
minimum	SSF Tool 2				SSF Tool 6			
number of detects	test length	characterized speed	slow speed (1/3)	very slow speed (1/30)	test length	characterized speed	slow speed (1/3)	very slow speed (1/30)
1	313	3	5	7	547	3	7	7
2	671	2	5	7				
3	981	0	3	5				
4	1,292	0	5	5				
5	1,605	0	3	5	2,398	0	2	5
7	2,203	1	5	5				
10	3,022	0	2	6				
12	3,578	0	2	5				
15	4,396	0	2	5	7,181	0	2	5

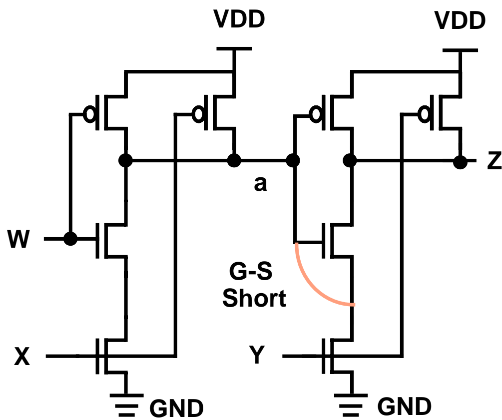

Figure 2. Transistor-level circuit for Fig.1.

W X Y	a Y	Z
0 0 1	1 1	0
0 1 1	1 1	1
1 0 1	1 1	1

Table 3. Partial table of combinations.

Multiple-detect test sets apply several different transitions to each gate in the circuit so we thought that they might be effective in detecting timing defects. The data in Table 2 suggests that this is indeed true. We thought it would be interesting to compare this data with the results of applying test sets that were generated for delay faults rather than SSF faults.

NON- SSF TEST SETS

Table 1 shows that the exhaustive test sets do identify all of the defective chips when the patterns are applied at characterized speed, but that some of these bad chips escape detection when tested at slower speeds. Also, most of the N-detect test sets, when applied at characterized speed, identify all the defective chips, but miss some at slower speeds. Clearly some chips have timing problems and it makes sense to try some type of delay testing.

Table 4 shows the results of applying 12 test sets that do not rely directly on the SSF model. Nine of these target delay faults. One uses design verification patterns. The techniques used to derive this table were explained at ITC'98 [2]. For the test sets that we were able to obtain, the N-detect sets seem to perform better for comparable test length than any of the test sets in Table 4. The remaining items, *IDDq* and VLV, will be discussed in a later section.

Table 4. Number of test escapes for non-SSF Test Sets.

			Escapes		
Tool	fault model	test length	characterized speed	slow speed (1/3)	Number of CUTs tested
	IDDQ	100	15		116
	VLV		3	4	116
1**	Transition	1,444	6	8	116
2*	Stuck open	1,610	4	9	116
3*	Gate delay fault		9	11	78§
	Gate delay fault, X->0		8	9	78§
4*	Robust path delay, X->0		5	5	60§
	Robust path delay, X->random		5	5	60§
5*	Robust path delay		3	5	60§
6**	Non-robust path delay		8	9	60§
	Non-robust path delay		2	5	60§
7	Verification		6	7	56§

* academic ATPG tools ** commercial ATPG tools

§ These patterns were not available for all of the CUTs

Two main techniques are used to apply delay tests: *at-speed*, in which successive patterns are applied at a fixed rate with no interruption, and *two-pattern*, in which there is a pause after the first pattern is applied and before applying the next pattern. These are shown in Fig. 3. The results in Table 4 were obtained using at-speed testing. To our surprise, there were more test escapes when two-pattern testing was used. This was discussed at ITC'98 [2].

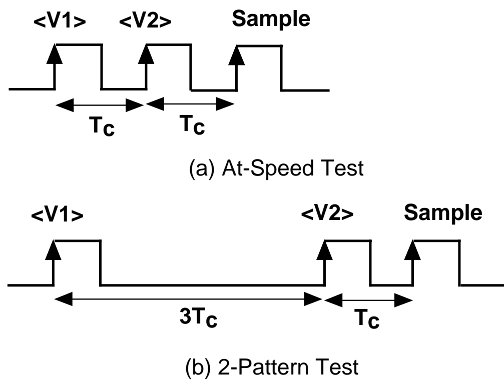

Figure 3 Delay test techniques: (a) At-speed test, (b) 2-pattern test.

DEFECT CHARACTERIZATION

In order to put some perspective on timing issues we collected data on whether the tester output response changed when the test speed was changed. We found that about 30% (39 out of the 116) of the defective dies had defects that caused timing dependent test responses. Another aspect of the defects that we were interested in was whether or not a defect caused the test response to depend on the order in which the patterns were applied. We found that about 30% (36 out of the 116) of the defects transformed the CUT from a combinational circuit into one with sequential behavior. The other defects, those that do not cause timing or sequence dependent behavior, are called *TIC (timing independent combinational)* defects. This defect characterization is shown in Fig. 4. The large number of sequence dependent failures was unexpected. We repeated many of the tests at very slow speed (1/30 characterized speed) to verify that these were not really timing defects. We now suspect that open defects are the cause of at least some of the sequence dependent failures [5][6][7].

Also of interest was the question of the existence of reliability defects. The next section discussed this issue.

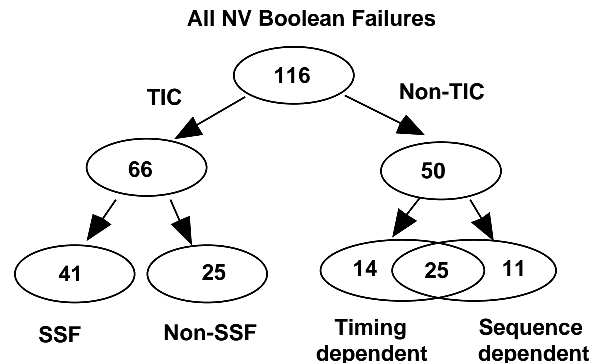

Figure 4. Characterization of defects.

RELIABILITY DEFECTS

Defects whose presence doesn't interfere with normal operation during manufacturing test, but which cause early life failure, are called *flaws* or *reliability defects.* Temperature and voltage burn-in are often used to convert such flaws into solid defects so that they can be detected and the corresponding chips, *weak chips*, discarded. There is interest in finding ways to avoid the necessity for burn in by identifying weak chips during manufacturing test. The major candidates for doing this are *IDDq* testing [8] and its many variants, *VLV* (very low voltage) testing [9], *min*V_{DD} testing [14][15], and SHOVE testing [16][17].

In *IDDq testing* the value of the power supply current is measured when all nodes in the circuit are quiescent (static). The current measurements are taken for a number of different input combinations and internal chip states. Chips whose measured current is significantly higher than the majority of chips are identified as IDDq fails and may be discarded. Issues related to IDDq testing include: (1) how to choose the maximum current value for a passing device, (2) how to deal with large background currents caused by sub-threshold currents of a large number of gates, and (3) how to use the measured IDDq values for defect location (diagnosis). In order to deal with these issues a number of enhancements to the basic IDDq test techniques have been proposed. These include current *signature* in which the values of the current measurements are sorted in increasing arithmetic order [18] and the *difference method* in which the difference between the maximum and minimum current values is used to decide whether or not to reject a chip [19].

We collected six different commercial IDDq test sets, five based on the pseudo-stuck fault model [10], and one pseudo-random. We did not find any significant differences among the results obtained from the six test sets, suggesting that the pseudo-stuck model may not be very effective. Only about 20 patterns were needed to identify the suspect chips. IDDq tests can be used to identify non-functional (failed) chips as well as weak chips. As shown in Table 4, 15 failed chips escaped our IDDq tests. A paper at VTS'98 [11], describes the details of this study.

Very low voltage testing, *VLV testing*, is a non-destructive technique for identifying weak chips. In this technique the chip is tested with a reduced supply voltage — approximately between 2 and 2.5 times the transistor threshold voltage. The actual value can be determined by characterizing the chips using a Schmoo plot; for the Murphy experiment the VLV voltage was 1.7V (The threshold voltage is 0.7V) [12]. The VLV tests must be carried out at a reduced speed to account for the effect of the lowered voltage on the chip operating speed. Some weak chips that operate correctly at normal V_{DD} fail to operate at VLV; others operate at VLV, but at a reduced speed. By correctly scaling the operating frequency, it is possible to identify weak chips which fail delay testing at VLV†. In the Murphy experiment 9 chips passed all tests at normal V_{DD} but failed VLV testing. When burned in at 130 degrees C, two of these chips failed after 6 hours. It appears that the other chips have defects that are not accelerated by temperature. This is discussed in detail in [13].

Another type of low voltage testing, *min*V_{DD} *testing*, aims to discover the lowest value of V_{DD} for which the chip operates. It is measured by varying the value of V_{DD} while applying stimulus patterns. Chips with abnormal values of min V_{DD} may be discarded. The use of *min*V_{DD} testing is discussed in [14] and [15]. We have not yet collected data for this type of testing on the Murphy chip.

Very little has been published about high voltage techniques although some form of dynamic elevated voltage stress (called *SHOVE*, dynamic voltage screen, new reliability screen, stress test, etc.) appears to be widespread [16][17]. In SHOVE testing, the value of V_{DD} is raised, patterns are applied and then V_{DD} is returned to its normal value. The chip outputs may or may not be observed while V_{DD} is elevated. Typically, either IDDq or Boolean tests are applied after normal V_{DD} is restored. Concerns include determination of the elevated value of V_{DD}, selection of patterns to apply while V_{DD} is elevated, and how long to hold the chip at the high voltage, [16]. SHOVE testing is usually carried out as part of the standard manufacturing test flow. We have not yet collected data for this type of testing on the Murphy chip.

We wondered whether it would be feasible to do Boolean testing only at VLV and not at normal V_{DD}. The Venn diagram in Fig. 5 shows the numbers of failed Murphy chips that are detected by (1) Boolean testing at normal V_{DD}, (2) Very low voltage Boolean testing, and IDDq testing at normal voltage. If only VLV testing was carried out, the diagram shows that 3 failed chips would not be detected (the IDDq failure is not counted because this chip is fully functional except for its elevated current.)

---

† In the Murphy experiment, the speeds used to test the multiplier CUTs were approximately 4 and 1.4 MHz; and for the control logic CUTs the speeds were approximately 14 and 4.6 MHz. The slower speeds were used to determine whether the fails at the scaled frequency were hard or timing failures. Of the 9 Murphy VLV only fails 2 were hard fails and the remaining 7 were timing failures.

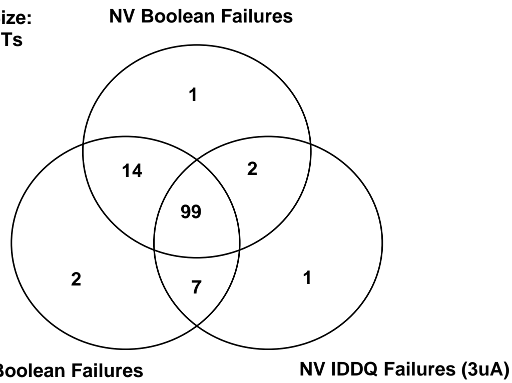

Figure 5. Venn diagram comparing *VLV*, normal V_{DD}, and *IDDq* failures.

CONCLUSIONS

The single-stuck fault model isn't a very accurate representation of actual production defects, but it is a very powerful aide in developing good manufacturing tests. Test sets that detect each single-stuck fault more than once do a very good job of detecting both hard and timing failures. Current practice is to use test inputs derived using a stuck fault model for reliability testing.

We can now look back and realize that it was obvious that the multiple-detect test sets would be effective in finding defects that do not act like single-stuck faults. Hughes studied the ability of single-stuck fault test sets to detect defects modeled by multiple-stuck faults and found that the single-stuck fault test sets were very effective in detecting the multiple-stuck faults [20]. In his study, significantly longer test sets did much better than the shorter sets. Millman concluded that single-stuck fault test sets were effective for detecting stuck-open faults and that node activity was a good indicator of whether a particular test set would or would not be effective [21]. As Billy Wilder said, "Hindsight is always twenty-twenty."

ACKNOWLEDGEMENTS

We would like to acknowledge Hughes Aircraft Co., LSI Logic, and the Office of Naval Research for supporting the Murphy experiment. We also would like to thank Digital Testing Services and Advantest for providing the testers for testing the chips. We thank Pycon Inc. for providing the burn-in equipment.

Without the contributions of many people, the Murphy experiment would not have been possible. We gratefully thank these people who have contributed to the chip design, test pattern generations, packaging and testing: Jacob Abraham, Debashis Bhattacharya, Rodolfo Betancourt, Susheel Chandra, Dave Doughty, Bob Duell, Gary Greenstein, Dong Ha, Charles Hawkins, Uwe Huebner, Michael Lynch, Rathish Jayabharathi, Rohit Kapur, Scott Keller, Raul Lopez, Alida Masctitelli, Peter Maxwell, A.K. Pramanick, Mike Purtell, Irith Pomeranz, Sassan Raissi, Lakshmi M. Reddy, Sudhakar M. Reddy, Daniel Saab, Mike Sarvian, Prab Varma, L.-T. Wang, and John Waicukauski.

Special acknowledgement is made to William D. Farwell, Janis Stokes, and Robert Stokes of Hughes and Piero Franco of Stanford for all their work in designing the chip and the wafer sort experiment.

Finally, we want to thank all the following CRC alumni and students for collecting and analyzing the experimental data: Jonathan Chang, Kan-Yuan Cheng, Yi-Chin Chu, Moon-Jung Kim, James C.-M. Li, Siyad Ma, and Sanjay Wattal.

REFERENCES

1. Nigh, P., W. Needham, K. Butler, P. Maxwell, and R. Aitken, "An Experimental Study Comparing the Relative Effectiveness of Functional, Scan, IDDq, and Delay-fault Testing," Proc. IEEE VLSI Test Symposium, pp. 459-464, 1997.
2. Chang, T.Y.J., C.W. Tseng, C.M.J. Li, M. Purtell, and E.J. McCluskey, "Analysis of Pattern-Dependent and Timing-Dependent Failures in an Experimental Test Chip," Proc. International Test Conference, pp. 184-193, 1998.
3. McCluskey, E.J., "Quality and Single-Stuck Faults," Proc. International Test Conference, pp.597, 1993.
4. Franco,P., S. Ma, T.Y.J. Chang, Y. Chu, S. Wattal, R. Stokes, W. Farwell, E.J. McCluskey, "Analysis and Detection of Timing Failures in an Experimental Test Chip," Proc. International Test Conference, pp. 671-700, 1996.
5. Wadsack, R.L., "Technology Dependent Logic Faults," Bell System Technical Journal, Vol. 57, May-June, pp. 1449-1474, 1978.
6. Chandramouli, R., "On Testing Stuck-Open Faults," Proc. International Symposium on Fault Tolerant Computing, pp. 258-265, 1983.
7. Woodhall, B.W., "Empirical Results on Undetected CMOS Stuck-Open Failures, " Proc. International Test Conference, pp. 166-170, 1987.
8. Levi, M., "CMOS is most testable," Proc. International Test Conference, pp. 217-220, 1981.
9. Hao, H., and E.J. McCluskey, “Very-Low-Voltage Testing for Weak CMOS Logic IC's,” Proc. International Test Conference, pp. 275-284, 1993.
10. Mao, W., R.K. Gulati, D.K. Goel and M. Ciletti, "QUIETEST: a quiescent current testing methodology for detecting leakage faults," Proc. International Conference on Computer-Aided Design, pp.280-283, 1990.
11. Chang, T.Y.J., C.W. Tseng, Y.C. Chu, S. Wattal, M. Purtell, and E.J. McCluskey, "Experimental Results for IDDQ and VLV Testing," Proc. VLSI Test Symposium, pp. 118-123, 1998.

12. Chang, J.T.Y. and E.J. McCluskey, "Quantitative Analysis of Very-Low-Voltage Testing," Proc. VLSI Test Symposium, pp. 332-337, 1996.
13. Li, J.C.M, E.J. McCluskey, "Testing for Tunneling Opens," Proc. International Test Conference, 2000. (To appear)
14. Righter A.W., C.F. Hawkins, J.M. Soden, P. Maxwell, "CMOS IC Reliability Indicators and Burn-in Economics," Proc. International Test Conference, pp. 275-284, 1998
15. Tseng, C.W. and E.J. McCluskey, "Cold Delay Defect Screening," Proc. VLSI Test Symposium, 2000.
16. Chang, T.Y.J. and E.J. McCluskey, "SHOrt Voltage Elevation (SHOVE) Test for Weak CMOS ICs," Proc. VLSI Test Symposium, pp. 446-451, 1997.
17. Kawahara, R., O. Nakayama and T. Kurasawa, "The Effectiveness of IDDQ and High Voltage Stress for Burn-in Elimination," Proc. IDDQ Workshop, pp. 9-13, 1996.
18. Gattiker, A.E. and W. Maly, "Current Signatures," Proc. VLSI Test Symposium, pp. 112-117, 1996.
19. Miller, A.C., "IDDQ Testing in Deep Sub Micron Circuits," Proc. International Test Conference, pp. 724-729, 1999.
20. Hughes, J.L.A., and E.J. McCluskey, "An Analysis of the Multiple Fault Detection Capabilities of Single Stuck-At Fault Test Sets," Proc. International Test Conference, pp. 52-58, 1984.
21. Millman, S.D., and E.J. McCluskey, "Detecting Bridging Faults With Stuck-at Test Sets," Proc. International Test Conference, pp. 773-783, 1988.