Multiple-Output Propagation Transition Fault Test

Chao-Wen Tseng* and Edward J. McCluskey

Center for Reliable Computing
Stanford University
http://crc.stanford.edu

Abstract

The test results of eight "challenge" Murphy chips that escaped either at least one of the 100%** single stuck-at fault test sets or the 100% transition fault test set were analyzed. The results show that: (1) An input pattern sequence is needed to detect the defects in the eight chips; (2) The detection of a transition fault depends on the outputs at which it is observed; (3) A transition fault test set is more effective if each transition fault is detected more than once. A transition fault test set, **TARO**, in which each **T**ransition fault is propagated to **A**ll the **R**eachable **O**utputs, is created and the experimental results are presented. The TARO test set detected all the eight "challenge" Murphy chips.

1. Introduction

The presence of a defect in a combination circuit can alter the circuit's logical behavior in a number of ways:

(1) The circuit's output may be incorrect for some input patterns, independent of what the previous input patterns were. Defects that affect the circuit in this fashion have been called *combinational defects*. A combinational defect can be either *timing dependent* or *timing independent*. An example of a timing-independent combinational defect, which has been called a *TIC* defect, [Franco 96] is a metal short to VDD. The logic value of the net that is shorted to VDD is permanently changed to logic "1", independent of the clock speed.

As of the day the paper is finished, we have not found any example of defects that are timing-dependent combinational. In addition, none of the defective chips in our Murphy [Franco 94] and ELF35 [Li 99] experiments behave like a timing-dependent combinational defect***.

(2) The circuit's output may be incorrect for some input patterns only if each such input pattern is preceded by another specified input pattern. Defects that affect the circuit in this fashion have been called non-combinational defects, or *sequence-dependent* defects. A sequence-dependent defect can also be either *timing independent* or *timing dependent*. An example of a timing-independent, sequence-dependent defect is an open defect at the drain or source of a transistor, *i.e.* a stuck-open defect. A two-pattern input combination is required to detect a stuck-open defect [Wadsack 78]. A stuck open defect can be detected by either at-speed tests or slower speed tests [Li 01]. Therefore, stuck open defects are timing independent, but sequence-dependent.

An example of a timing-dependent, sequence-dependent defect is a resistive open defect. A resistive open defect causes extra delay to all the paths passing through it. To detect a resistive open defect, a two-pattern input combination is required [Li 01]. The first pattern initializes the defect location and the second pattern creates a transition at the defect location. A resistive open defect might fail the at-speed test, but pass a slow speed test if the extra delay is smaller than the time slacks at the slow speed. Table 1 lists the three defect classes and an example of defects for each class.

Table 1: Defect Classifications and Examples

	Combinational	Sequence-dependent
Timing Independent	Metal shorts to VDD	Stuck Opens
Timing Dependent		Resistive Opens

* Now with Zettacom Inc., Santa Clara, CA 94054

** 100% fault coverage based on fault simulations.

*** When the data was first analyzed, it appeared that some defects were timing-dependent combinational. But a more thorough analysis showed that those defects are actually timing-dependent, sequence-dependent defects.

0-7803-7169-0/01 $10.00

With the advent of deep submicron technology and increasing clock frequency, many small defects that were previously tolerable are now starting to cause failures [Nigh 00]. Furthermore, only part of all the defects behaves like combinational defects [Nigh 00][McCluskey 00]. To detect the defects that are non-combination, appropriate pattern sequences are required

However, test sets based on the single stuck-at fault (SSF) model are still the most common in today's industry. Even though SSF test sets are generated without considering the relationship between consecutive patterns, it was reported that they were very effective for detecting both combinational and non-combinational defects [McCluskey 00]. Especially, the SSF test sets in which each SSF is detected at least three times detected all the combinational and non-combinational defects in the Murphy experiment [McCluskey 00]. In a SSF 3-detect test set, each SSF is detected at least 3 times or the maximum number of times it can be detected. Does this mean we really do not need to worry about the order of patterns in a test set?

The answer is NO. After fault grading all the SSF test sets with the transition fault model, we found that there is a correlation between the effectiveness of a SSF test set and its transition fault coverage. For two test sets with the same SSF coverage, the one with a higher transition fault coverage is more effective [Li 01]. It was also shown that a SSF N-detect test set does have a higher transition fault coverage [Chang 98]. In the first part of this paper, we will present analysis results for the eight Murphy chips that escaped either at least one 100% SSF test sets or the 100% transition test set in the Murphy experiment. The results show that these chips were detected by sequences of input patterns, not by single input pattern. Although these eight chips were detected by SSF N-detect test sets, it is the sequence of input patterns that makes the test sets effective.

Fault models are used to generate test patterns. To generate test patterns with the pattern sequence taken into account, we need to use a fault model that cannot be detected by a single pattern such as the SSF model. There are two commonly-used fault models used for generating a two-pattern test: the path delay fault model [Shedletsky 78][Lesser 80] and the transition fault model [Waicukauski 87]. The path delay fault model is used to generate a 2-pattern test for each path in a circuit. Because there are many paths in a circuit, there are many different input pattern sequence in the test set. However, its biggest drawback is that there may be too many paths in practical circuits to test each path individually. Testing a subset of all paths has been proposed to reduce the number of paths to test. One such example is to test the critical paths (longest delay) of a circuit [Reddy 87]. However, the critical path delay fault test set only detected 83 out of the 116 defective chips in the Murphy experiment. It is even less effective than the SSF test set with only 80% coverage, with which only 86 out of the 116 defective chips were detected.

On the other hand, testing all possible transition faults is possible because the number of transition faults is far less than that of path delay faults. The transition fault model assumes that the fault locates at either the inputs or outputs of a gate and the fault can be observed at any reachable outputs of a circuit. However, a two-pattern test for a transition fault usually only propagates the fault to one or few reachable outputs due to fault dropping during the test pattern generations [Waicukauski 87]. Even though this approach leads to a more compact test set, the drawback is there are fewer input pattern sequences in a transition fault test set than a path delay fault test sets. In our Murphy experiment, a transition fault test set in which each fault is propagated along the longest path missed 9 out of the 78[1] defective chips. Another transition fault test set, in which each fault is propagated to any reachable outputs, still missed 2 of the 116 defective chips.

In the second part of this paper, we will present the experimental data for the **TARO** test set. In the TARO test set, each **T**ransition fault is propagated to **A**ll the **R**eachable **O**utputs. This approach compensates for the lack of propagation path selections in the transition fault test generations, even though no path tracing is really involved. Although the path coverage of the TARO test set is still not as complete as that of the path delay fault test generations, there are more input sequences applied to the circuit than a transition fault test set. Our experimental results show that the TARO test set detected all the 116 defective chips.

This paper is organized as follows. In section 2, we study the defect behavior of eight Murphy chips that escaped either at least one of the 100% SSF test sets or the 100% transition fault test sets. The study helps in deciding the key factors that make a test set effective. Section 3 describes how we derived the TARO test set and presents the experimental results. Section 4 summarizes the paper.

[1] The test sets were generated for only three of the five designs. Therefore only 78 defective chips were tested.

2. Analysis of Eight "Challenge" Murphy Chips

The Murphy experiment was performed to evaluate the effectiveness of various test techniques. The Murphy chip was designed and manufactured in 0.7um CMOS technology with a normal supply voltage of 5V. There are 5 combinational designs on the Murphy chip. The test sets applied include verification, single stuck-at fault, transition fault, stuck open fault, gate delay fault, path delay fault, weighted random, and exhaustive test sets. For more detailed information about the chip design and the test results, the readers can refer to the following papers [Franco 94][McCluskey 00].

We identified 116 defective Murphy chips. Eight of these 116 chips were particularly challenging to detect. They each escaped detection by at least one of the three most effective 100% SSF test sets or the 100% transition fault test set. Table 2 lists the test results of the eight challenge chips. The three most effective SSF test sets are generated by Tools 2A, 3C and 4C. They were the most effective test sets in the sense that they detected the most defective chips. Tool 2A is an academic ATPG tool, while 3C and 4C are commercial ATPG tools. To simplify the notation, we use *1-detect* to represent 100% fault coverage. We also included the test results for SSF N-detect test sets from Tool 2A in Table 2. In a SSF N-detect test set, each SSF is detected at least N times or the maximum number of times it can be detected.

In Table 2, blank entries represent test detects and entries with "E" represent test escapes. As shown in Table 2, chip 1 failed all the SSF N-detect test sets generated by Tool 2A, the 100% transition fault test set, the 100% SSF test set from Tool 3C, but escaped the 100% SSF test set from Tool 4C. Chips 1 through 5 escaped one of the three 100 % SSF test sets from Tools 2A, 3C and 4C. Chip 6 escaped the 100% transition fault test, but failed all the SSF test sets. Chips 7 and 8 escaped the SSF N-detect test from Tool 2A for n less than 2. Chip 8 also failed the 100% transition fault test.

These eight chips were re-tested to record their failing patterns and the corresponding failing output bits. A *failing pattern* is the input pattern that causes the chip to produce erroneous outputs. The *failing output bits* are the output bits that have erroneous logic values when a failing pattern is applied. In the next section, we will describe the re-test experiment, including test timing, test patterns and test voltages.

2.1 Re-Test Setup

When the Murphy chip was designed, four copies of each design were placed on the chip. Figure 1 shows how the inputs were applied to the four copies of each design, also how their outputs were collected. All the four copies shared the same inputs that were applied either directly from the tester or from the on chip linear-feedback shift registers (LFSRs). The outputs from the four copies are latched. An on-chip comparator compares the outputs from the four copies to determine if any of the four copies has erroneous results. Two counters record the first failing pattern and total number of failing patterns. Only the counter values are collected on the tester during testing. The outputs from the four copies are also placed in a scan chain so they can be shifted-out for diagnosis. All the flip-flops at the output of each copy are for output data readout, not a part of the circuits that is being tested.

Figure 2 shows the timing waveforms for re-test. For each test pattern, two system clock cycles and either 24 or 48 scan cycles are applied (One design has 6 outputs and the other 4 designs have 12 outputs. Because there are 4 copies of each design, the number of scan cycles is equal to 4 times the number of outputs for each design).

Table 2: Test Escapes for the Eight "Challenge" Murphy Chips

Test Set	Chip ID							
	1	2	3	4	5	6	7	8
SSF Tool 2A, 1-detect							E	E
SSF Tool 2A, 2-detect							E	
SSF Tool 2A, 3-15 detect								
SSF Tool 3C, 1-detect		E	E		E			
SSF Tool 4C, 1-detect	E			E				
Transition						E		E

Test pattern I(n) is applied from the tester to the chip at the rising edge of the first system clock. At the rising edge of the next system clock cycle, the circuit response to input pattern I(n), O(n), is latched into the scan flip-flops at the outputs. Depending on the design, either 24 or 48 scan cycles were applied to scan out the output response O(n). After the scan out operations, test pattern I(n+1) is applied from the tester and the whole process is repeated. The system clock rate is the chip speed. Scan out operations are performed at a speed of 1MHz, which is much slower than the chip speed. The time from applying of a pattern to the chip inputs to capturing the response at the outputs is precisely controlled. The time is the chip speed. However, the rate at which two consecutive patterns were applied is slower than the chip speed. As a matter of fact, it is significantly slower due to the many scan-out cycles included and the slow speed for scan-out. This is to emulate the 2-pattern test for sequential circuits, because many scan cycles usually exists between the two consecutive patterns.

The test sets applied include 1, 2, 3, 4, 5, 7, 10, 12 and 15-detect test sets from Tool 2A, 100% SSF test sets from Tools 3C and 4C, and the 100% transition fault test set. All tests were applied at the normal supply voltage, 5.0V.

2.2 Transition Fault Diagnosis

In a previous paper [McCluskey 00], we compared the SSF fault simulation results with the tester responses for the 116 defective chips. There are forty-one chips whose test responses matched the SSF fault simulations. Because none of the 8 challenging are among the 41 chips, we tried to diagnose these chips using the transition fault model.

We first built a transition fault list. For each fault in the fault list, we then performed transition fault simulations. The transition fault simulation is performed by combining SSF fault simulations and logic simulations. A two-pattern test, (V1,V2), is required to detect a transition fault. Table 3 lists the conditions under which a transition fault at is detected. The condition for V1 was verified by logic simulations and the condition for V2 was verified by SSF fault simulations.

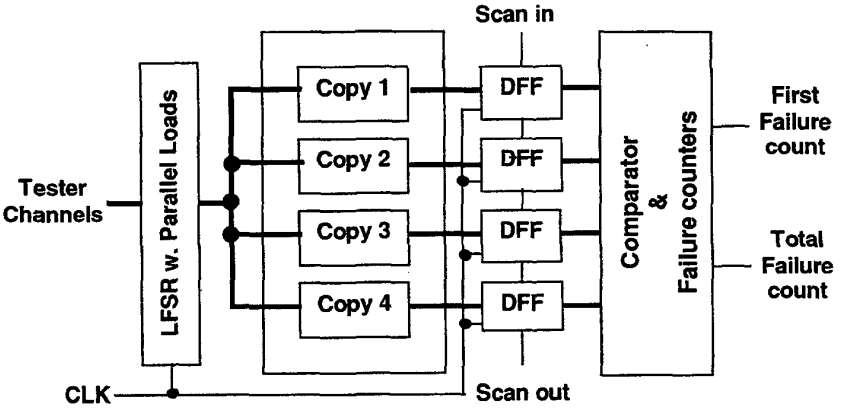

Figure 1: Murphy Chip Architecture

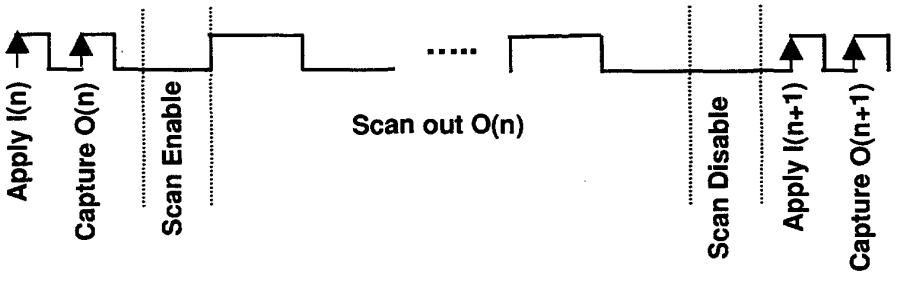

Figure 2: Re-test Timing Diagram

Table 3: Conditions for detecting a transition fault

		Transition Faults	
		Slow-to-rise	Slow-to-fall
Pattern	V1	Set x to "0"	Set x to "1"
	V2	Detect x/0	Detect x/1

After collecting all the failing patterns and the corresponding failing output bits for all the transition faults in the fault list, they were compared with the tester responses collected during re-test. According to the discussions in the previous section, the responses were collected for every input pattern during re-test. For all the SSF test sets, the comparisons between the transition fault simulations and tester responses were made for every pattern. However, the comparisons were made for every other pattern for the transition fault test set, because the transition fault test itself is a two-pattern test.

Chips 1 through 5

Table 4 lists the transition fault diagnosis results for chips 1 through 5. For chips 1 through 5, their tester responses match the transition fault simulation results. Table 4 lists the transition fault diagnosis results for the five chips. Chip 4 is diagnosed to have both slow-to-rise and slow-to-fall faults. Therefore, there are 2 transition faults for chip 4. For the other 4 chips, the diagnosis results show there is only one transition fault inside the chip.

Table 4: Diagnosis Results for Chips 1-5

Chip	Faults	Slow-to-fall	Slow-to-rise
1	1	X	
2	1	X	
3	1		X
4	2	X	X
5	1		X

For the matched transition faults in chips 1 through 5, we extracted their transition fault simulation results for all the applied test sets. Table 5 shows the results. An "**N**" in a entry indicates that the matched transition faults were not detected by the corresponding test set according to the transition fault simulations. Otherwise, an "**Y**" is placed in the entry. For example, the matched slow-to-fall fault in Chip 1 was not detected by the 100% SSF test set from Tool 4C, but was detected by all other test sets. On the other hand, the matched slow-to-fall fault in chip 2 was detected by all test sets. An "**E**" in a entry indicates that the chip escaped the test set during re-test. Otherwise, an "**F**" is placed in the entry. As indicated in Table 5, Chip 1 passed the test when the SSF test set by tool 4C was applied on the tester.

Table 5: Transition Fault Simulation Results for Chips 1 through 5

Test Set	Chip ID				
	1	2	3	4	5
SSF Tool 2A, 1-detect	Y,F	Y,F	Y,F	Y,F	Y,F
SSF Tool 2A, 2-detect	Y,F	Y,F	Y,F	Y,F	Y,F
SSF Tool 2A, 3-15 detect	Y,F	Y,F	Y,F	Y,F	Y,F
SSF Tool 3C, 1-detect	Y,F	**Y,E**	**N,E**	Y,F	**Y,E**
SSF Tool 4C, 1-detect	**N,E**	Y,F	Y,F	**N,E**	Y,F
Transition	Y,F	Y,F	Y,F	Y,F	Y,F

N: no 2-pattern sequence for detecting the faults
Y: 2-pattern sequence exists for detecting the faults
E: Test escapes, **F**: Failures.

For chips 1 and 4, the 100% SSF test set from Tool 4C does not contain a 2-pattern sequence that can detect the matched transition faults in either chip. Both chips escaped the test on the tester. For chip 3, the 100% SSF test set from Tool 3C does not contain a 2-pattern sequence that can detect the slow-to-rise fault in either chip. It also escaped the test on the tester.

Chips 2 and 5 were predicted to failed on the tester when the 100% SSF test set from Tool 3C was applied, because the matched transition faults in both chips were detected in fault simulation. However, both chips escaped. To further understand why chips 2 and 5 escaped, we checked the failing output bits and failing patterns for all the test sets. Table 6 lists all the possible failing outputs for chip 2 when different test sets were applied. The list of possible failing output bits for a test set is the union of the failing output bits for each failing pattern in the test set. For example, if the SSF 1-detect test set from Tool 2A is applied, the transition fault simulation results show that the matched slow-to-fall fault can be observed at output bits 1,2,3 and 9. If the 100% SSF test set from Tool 3C is applied, the faults will only be observed at output bits 2 and 8. The bold numbers in Table 6 indicate the failing output bits observed on the tester. When the SSF 1-detect test set from Tool 2A was applied, all the failing patterns produced failures at output bit 9. On the other hand, both output bit 5 and 9 failed when the 100% SSF test set from Tool 4C was applied. Table 7 lists the results for chip 5. For chip 5, the only exception is that output bit 9 did not fail when the 100% SSF test set

from Tool 4C was applied, while bit 9 failed when the SSF 1-detect test set from Tool 2A and the 100% transition fault test set were applied.

Table 6: Failing Output Bits for Chip 2

Test Set	Failing Output Bits	
	Simu-lation	Tester
SSF Tool 2A, 1-detect	1,2,3,9	**9**
SSF Tool 2A, 2-detect	1,2,5,8	**5**
SSF Tool 3C, 1-detect	2,8	
Tool 4C, 1-detect	1,5,9	**5,9**
Transition	1,2,5,9	**5,9**

Table 7: Failing Output Bits for Chip 5

Test Set	Failing Output Bits	
	Simu-lation	Tester
SSF Tool 2A, 1-detect	1,2,3,9	**1,9**
SSF Tool2A, 2-detect	1,2,5,8	**1**
SSF Tool 3C, 1-detect	2,8	
Tool 4C, 1-detect	1,5,9	**1**
Transition	1,2,5,9	**1,9**

Our experimental data indicate that the detection of a transition fault depends not only on the pattern sequences, but also on the outputs to which the transition fault is propagated. Even though the 100% SSF test set from Tool 3C detects the matched transition faults in both chips 2 and 5, the faults are propagated to output bits at which the faults could not be observed on the tester. One possible explanation is that the faults were propagated along a path with a time slack larger than the size of the fault. Unfortunately, we could not verify this due to the lack of timing information for the Murphy chip. What we did to verify the explanation is to extract the failing 2-pattern pairs from Tool 4C, and padded them to the test set from Tool 3C. Both chip 2 and chip 5 failed on the tester after the new test set was applied. This evidence indirectly supports our explanations. In addition, similar observations have been made in [14], but without the support of experimental data.

Chip 7

Chip 7 was detected by all the 100% SSF test sets from Tools 3C and 4C, and the 100% transition test set. However, it escaped SSF 1-detect and 2-detect test sets from Tool 2A. We were not able to match the tester responses of chip 7 to any single transition fault in fault simulation. Therefore, the defect behavior does not match either a stuck-at fault or a transition fault.

After analyzing the failing patterns in different test sets, we found that chip 7 failed a specific pattern pair (A,B) that exists in all the SSF N-detect test set for n>2. We also found pattern B in both the SSF 1-detect and 2-detect test sets, but with different preceding patterns C (1-detect) and D (2-detect). Chip 7 failed neither pattern pair (C,B) or (D,B) on the tester. For pattern pair (A,B), we reordered it to (B,A) and applied the new test sets with the reordered pattern pair to chip 7. Chip 7 passed all the test sets with the reordered pair. We also added the pattern pair (A,B) to the SSF 1-detect and 2-detect test sets and applied the new test sets, chip 7 failed the new test sets on the tester.

The results from this chip indicated the defects could only be activated and detected by an appropriate pattern sequence. The SSF N-detect test set is effective because there are more input sequences in the test sets. However, those sequences are generated randomly not deterministically. In our experiment, the reordered version of a SSF N-detect test sets has the same SSF N-detect coverage as the old one. But our results suggested their defect coverages? are different. For some types of defects, like the one(s) inside chip 7, a right sequence is more important than detecting each SSF many times.

Table 8: Failing Pattern Pair for Chip 7

SSF Test Set	V1	V2 *	Test Result (V1,V2)	Test Result (V2,V1)
1-detect	C	B	**PASS**	-
2-detect	D	B	**PASS**	-
3-15 detect	A	B	FAIL	**PASS**

*S: Strobing pattern, ** A,B,C,D: input patterns

Chips 6 and 8

Both chips 6 and 8 failed the 100% transition fault test. As discussed in section 2.1, only every other pattern was strobed in the transition fault test set. We name these strobing patterns as "even"-numbered patterns because they are the 2nd, 4th, 6th ... etc. patterns in the test set.

We post-processed the 100% transition fault test set such that the strobes can be perform for the "odd"-numbered patterns as well. We then applied the new test set to both chip 6 and chip 8. Both chips failed. The tester response indicated that chip 6 failed three odd-numbered patterns and chip 8 failed one odd-numbered pattern.

The results suggested that even 100% transition fault coverage is not good enough. The transition fault test set we used has 100% coverage, if only the "even"-

numbered patterns are strobed. Both chips passed the test in this case. Strobing the "odd"-numbered patterns means we were applying a transition fault test set with more than 100% coverage, because some transition faults were detected more times than that in the original transition fault test set. Our results suggested that a transition fault test set is more effective if each transition fault is detected more than once.

2.3 Summary

We diagnosed and analyzed the test results for the 8 challenge Murphy chips. These eight chips escaped either the 100% SSF test sets or the 100% transition fault test set. Based on the analysis results, we sub-divided the 8 chips into three groups for discussion. The results from all three groups suggest that these eight chips cannot be detected by a single pattern. For all the eight chips, the defects were detected by appropriate pattern sequences, even though only five chips completely match the transition fault behavior.

The results for chips 1 through 5 suggest that a transition fault cannot be detected if it is not propagated to the appropriate outputs. The results for chips 6 and 8 suggest that the defect coverage can be improved by detecting each transition fault more than once. The results for chip 7 show that pattern ordering can affect the defect coverage of a SSF *N-detect* test set. A reordered SSF *N-detect* test set could have a higher defect coverage, but with the same SSF *N-detect* coverage.

3. Transition Fault to all Reachable Outputs (TARO)

Based on the discussions in the previous section, we created a new transition fault test set and applied it to all the 116 defective Murphy chips, including the 8 challenge chips. In this section, we will first describe how the new transition fault test set is created. We will also present the experimental results for applying the new transition test set to the Murphy chips.

3.1 Previous Work

Pomeranz et. al. proposed a procedure to generate a transition fault N-detect test set from a SSF test set [Pomeranz 99]. The patterns in the original SSF test set were reordered such that each transition fault can be detected multiple times. For a transition fault, they tried to select the pattern pairs as different from one another as possible to improve the chance that the fault is detected via different paths. However, they did not use the information about the outputs that each fault can be propagated to. In the following next section, we will present several modifications to their algorithm to create the TARO (Transition fault propagated to All Reachable Outputs) test set. In the TARO test set each transition fault is not only detected many times, but also propagated to all the reachable outputs.

3.2 Generation of the TARO Test Set

This section describes in details how we generated the TARO test set.

Step 1: Create a list of reachable outputs for each transition fault

For each transition fault, we need to find the set of outputs to which that the fault can reach. Tracing functional sensitizable paths in combinational circuits has been well reviewed and explained in the literature [Tekumalla 01]. Instead of implementing those algorithms, we chose to perform exhaustive simulations because all the chips under study have at most 24 inputs for which the exhaustive simulations are possible.

First, we performed SSF fault simulations for each stuck-at fault in the circuit, for all the possible input patterns. Secondly, we performed logic simulations for all the possible input patterns and recorded the logic values of each node inside the chip. For a pattern pair (V1,V2), a slow-to-rise transition fault at node x is detected if V1 sets x to 0 and V2 detects x/0. If both conditions are true, the slow-to-rise fault at node x can be propagated to all the failing outputs when V2 is applied. This information was in the data log from the SSF fault simulations and can be easily extracted.

Step 2: Reorder a given SSF test.

We modified the algorithm proposed by Pomeranz [Pomeranz 99] to reorder a SSF test to create the TARO test set. We assume that every two consecutive patterns are applied as a two-pattern test for transition faults. For a combinational circuit, this assumption is fine. For a sequential circuit, this assumption requires the existence of built-in design-for-testability (DFT) circuits such as shadow registers for applying the test patterns at-speed.

A SSF test set is chosen as the initial test set. The coverage of the SSF test is not too critical because the final TARO test set will have at least 100% SSF and transition fault coverage. Starting with the transition faults that could be only sensitized to a single output, the patterns in the initial test set were re-ordered to detect these transition faults. During re-ordering, we recorded the outputs to which that a transition fault is propagated. This is different from Pomeranz's algorithm in which the number of times each transition fault is detected is tracked.

Step 3: Add new patterns

After reordering the initial test set, some transition faults are still not propagated to all the reachable outputs or not detected. For those faults, we add new patterns that can detect them and also propagate them to the designated outputs. Because we have performed exhaustive simulations in Step 1, this step only required simple data extraction from the data log. Again, the detection profile is updated after each new pattern is added. The whole process continues until all the transition faults are fully propagated to all the reachable outputs.

We wrote Perl scripts to interact with Fastscan to create the TARO test set. The resulting TARO test set is not optimized as far as the size is concerned. However, the purpose here is to demonstrate the effectiveness of the TARO test set, not on its efficiency. It is possible to use the concept developed in this work to modify the existing ATPG tool to get a better test set in terms of size.

3.3 Experimental Results

Table 9 lists the test results. The TARO test set was created by reordering the 100% SSF test set from Tool 3C. After reordering, 304 new patterns were added to make sure all the transition faults were propagated to all reachable outputs. The test length of the TARO test set is 907, which is close the size of the SSF 3-detect test set from Tool 2A. All the eight chips failed the TARO test set.

Table 10 lists the failing output bits for chip 2 and chip 5 when TARO test set was applied. The original 100% SSF test set from Tool 3C failed to propagate the transition fault to the appropriate output bits to be observed. In the TRAO test set, the transition faults were propagated to output bit 5 and 9 for chip 2, and output bit 1 and 9 for chip 5 to be observed.

4. Summary

In this paper, we first analyzed the test results for the eight "challenge" Murphy chips that escaped either at least one of the 100% SSF test set or the 100% transition fault test set. To detect these eight chips, our analysis show that the test set must contain appropriate input pattern sequences. For the five chips that were diagnosed to have transition faults, two chips escaped the test even though the transition fault was covered in the test sets. For the two chips that escaped the transition fault tests, our results show that they can be detected if each transition fault were detected more than once.

We created the TARO test set, in which each transition fault is propagated to all the reachable outputs. The experimental results showed that all the eight chips were detected by the TARO test set. For the 2 chips that escaped the 100% SSF test sets because the transition faults were not propagated to the appropriate outputs, they both were detected by the TARO test set. The TARO test set detected all the 116 defective Murphy chips during re-test.

Table 9: TARO Test Results

Test Set	Test Length	Chip ID							
		1	2	3	4	5	6	7	8
SSF Tool 2A, 1-detect	313							E	E
SSF Tool 2A, 2-detect	671							E	
SSF Tool 3C, 1-detect	603		E	E		E			
SSF Tool 4C, 1-detect	1000	E			E				
Transition	1444						E		E
TARO	**907**								

Table 10: Failing Output Bits for Applying the TARO Test Set

Chip ID	Failing Outputs
2	5,9
5	1,9

Acknowledgement

We would like to thank David Wu, J.C. Lin, James Li, Subhasish Mitra and Scott Davidson for many helpful suggestions. We would also like to thank LSI Logic for fabricating the Murphy test chip. We would like to thank Advantest Inc. for providing tester resource and Mike Purtell for technical help.

This work was supported by Semiconductor Research Corporation under Contract No. 2000-TJ-845.

Reference

[Franco 96] Franco, P., S. Ma, T.Y.J. Chang, Y. Chu, S. Wattal, R. Stokes, W. Farwell, and E.J. McCluskey, "Analysis and Detection of Timing Failures in an Experimental Test Chip," *Proc. 1996 Int. Test Conf.*, pp. 671-700, 1996.

[Franco 94] Franco, P., R.L. Stokes, W.D. Farwell, and E.J. McCluskey, "An Experimental Chip to Evaluate Test Techniques Part 1: Description of Experiment," *CRC TR 94-5,* 1994.

[Li 99] Li, J.C.M., J.T.-Y. Chang, C.W. Tseng, and E.J. McCluskey, "ELF35 Experiment - Chip and Experiment Design," *CRC TR 99-3,* 1999.

[Wadsack 78] Wadsack, R.L., "Fault Modeling and Logic Simulation of CMOS and MOS Integrated Circuits," *Bell System Tech. J.,* Vol. 57, No. 5, pp.1449-1488, May-June, 1978.

[Li 01] Li, C.-M. James and E.J. McCluskey, "Testing for Resistive Opens and Stuck Opens," to appear in *Proc. 2001 Int. Test Conf.,* 2001.

[Nigh 00] Nigh, P. and A. Gattiker, "Test Method Evaluation Experiments and Data," *Proc. 2000 Int. Test Conf.*, pp.454-463, 2000.

[McCluskey 00] McCluskey, E.J. and C.W. Tseng, "Stuck-at Tests vs. Real Defects," *Proc. 2000 Int. Test Conf.,* pp.336-343, 2000.

[Chang 98] 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. 1998 Int. Test Conf.,* pp. 184-193, 1998.

[Shedletsky 78] Shedletsky, J.J., "Delay Testing LSI Logic," *Proc. 8th Int. Symp. on Fault-Tolerant Computing,* pp. 159-164, 1978.

[Lesser 80] Lesser, J.D. and J. J. Shedletsky, "An Experimental Delay Test Generator for LSI Logic," *IEEE Trans.on Computers,* Vol. C-29, No. 3, pp. 235-248, 1980.

[Hsieh 77] Hsieh, E.P. et. al., "Delay Test Generation," *Proc. 14th Design Automation Conf.*, pp.486-491, 1977.

[Reddy 87] Reddy, S.M. et. al., "An Automatic Test Pattern Generator for the Detection of Path Delay Faults," *Proc. 1987 Int. Conf. on CAD,* pp.284-287, 1987.

[Waicukauski 87] Waicukauski, J. et. al., "Transition Fault Simulation," *IEEE Design and Test,* pp.32-38, Apr. 1987.

[Baker 99] Baker. et. al., "Defect-Based Delay Testing of Resistive Vias-Contacts: A Critical Evaluation," *Proc. 1999 Int. Test Conf.*, pp.467-476, 1999.

[Pomeranz 99] Pomeranz, I. and S.M. Reddy, "On N-detection Test Sets and Variable N-detection Test Sets for Transition Faults," *Proc. 1999 VLSI Test Symp.,* pp.173-179, 1999.

[Tekumalla 01] Tekumalla R.C., et. al., "On Diagnosing Path Delay Fault in an At-Speed Environment," *Proc. VLSI Test Symp., 2001.*