

ELF-Murphy Data on Defects and Test Sets

E. J. McCluskey, Ahmad Al-Yamani, James C.-M Li, Chao-Wen Tseng,
Erik Volkerink, Francois-Fabien Ferhani, Edward Li, and Subhasish Mitra
Stanford Center for Reliable Computing*
<http://crc.stanford.edu>

Abstract

We at CRC have designed and LSI Logic has manufactured two test chip designs; these were used to investigate the characteristics of actual production defects and the effectiveness of various test techniques in detecting their presence. This paper presents a characterization of the defects that shows that very few defective chips act as if they had a single-stuck fault present and that most of the defects cause sequence-dependent behavior.

A variety of techniques are used to reduce the size of test sets for digital chips. They typically rely on preserving the single-stuck-fault coverage of the test set. This strategy doesn't guarantee that the defect coverage is retained. This paper presents data obtained from applying a variety of test sets on two chips (Murphy and ELF35) and recording the test escapes. The reductions in test size can thus be compared with the increases in test escapes. The data shows that, even when the fault coverage is preserved, there is a penalty in test quality. Also presented is the data showing the effect of reducing the fault coverage. Techniques studied include various single-stuck-fault models including inserting faults at the inputs of complex gates such as adders, multiplexers, etc. This technique is compatible with the use of structural RTL netlists. Other techniques presented include compaction techniques and don't care bit assignment strategies.

1. Introduction

This paper presents the data that we collected on a tester and compares it with fault model derived data. Accuracy and effectiveness demonstrated by this data are commented on. The paper focuses on results from the ELF35 chip and compares some of the results with their Murphy counterparts.

The two test chips were designed to permit very thorough and varied tests to be applied and the corresponding response data to be collected. The defects that are present on the chips are only those that occurred naturally during fabrication. No artificial defects were inserted. We were interested to compare the results for these two chips from different technologies. Reliability defects are also of interest, but will be discussed in another paper.

The Murphy chip design contains 4 copies each of 5 different very simple completely combinational cores (called Circuits under Test or CUTs in previous

publications). Two cores are data path structures and the other 3 are control logic designs.

The ELF35 chip design contains multiple copies each of 6 different cores. Two of the cores are sequential data path structures (two different implementations of the 2901 arithmetic processor). The other 4 are combinational (three data path designs and one translator).

Besides nominal voltage testing, Very-Low-Voltage (VLV) testing and IDDQ testing were also applied to all the packaged chips. In VLV testing, the supply voltage is 1.4V for ELF35 and 1.7V for Murphy. These voltages are about two times the transistor threshold voltage [Chang 96]. In IDDQ testing, the threshold was set to 100 μ A for ELF35 and 300 μ A for Murphy, which are typical values that LSI Logic uses for chips of comparable sizes in these technologies. A weak suspect core is a core that passed every test at nominal voltage but failed VLV or IDDQ.

Figure 1 shows the ELF35 core classification tree. A total of 495 interesting cores are identified. Interesting cores are the union of defective cores and weak suspect cores. Among these 495 interesting cores, 324 of them are defective and 171 of them are weak suspects. Among the 324 defective cores, 101 of them are FOSTS (fail only some test sets) and the others are FATS (fail all test sets). Among the 171 weak suspect cores, 130 of them failed only IDDQ testing and 9 of them failed only VLV testing. The other 32 failed both VLV and IDDQ testing. Figure 2 shows the same classification tree for the Murphy chips.

Most of our tester data was collected by applying patterns obtained from ATPG programs. The rationale for using many sources of patterns was either to minimize any bias caused by a particular ATPG source or because some tools have capabilities lacking in other tools. We have generated (or tool vendors have donated) various test sets from many academic tools (including Rutgers University, Texas A&M, University of Illinois, University of Iowa, and Stanford CRC) and commercial tools (including Fastscan, Sunrise, Syntest, and Tetramax.)

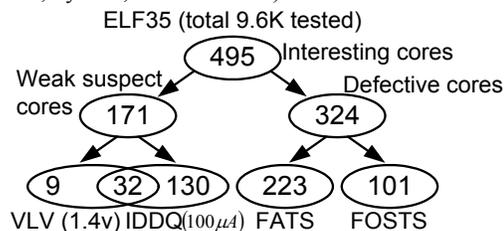


Figure 1. ELF35 classification.

* James Li is currently with National Taiwan University, Chaowen Tseng is currently with Zettacom, Edward Li is currently with Sun Microsystems and Subhasish Mitra is currently with Intel.

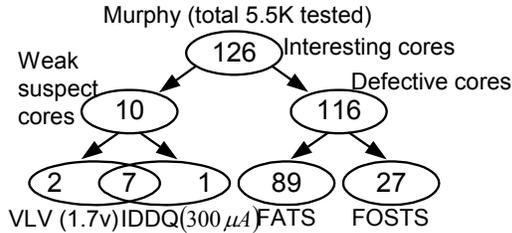


Figure 2. Murphy classification.

2. Characteristics of the defects

2.1. Sequence dependence

Since combinational circuits contain no memory elements, the response to a particular input combination should not depend on previous input combinations[§]. The insertion of a single- (or multiple-) stuck-at fault should not cause a combinational circuit to act as a sequential circuit, by exhibiting dependence of its output on previous inputs. Neither should other faults such as **non-feedback** bridging faults. We thought it would be interesting to check whether the defects on our chips transformed our combinational logic circuits into sequential circuits. To do this we applied **each** of our 100% single-stuck-at fault model test sets six times, each time using the same set of patterns but applying them in a different order. Order 1 is that obtained from the ATPG tool, order 2 is the same set of patterns with an all-0 pattern inserted between each pair of original vectors. Order 3 inserts an all-1 pattern instead of the all-0 pattern. Order 4 inserts the bit-wise complement between each pair of patterns and order 6 applies the original patterns in the reverse order.

43% of the defective Murphy chips and 42% of the defective ELF35 chips had sequence dependent test responses. Clearly the defects in these chips are not acting like single-stuck-at faults. The defects in these chips changed them from combinational circuits to sequential circuits. Naturally, we wondered what kinds of defects were causing this behavior. One possible defect that could do this is one that acts like a stuck-open fault [Li 02].* By matching the tester traces (response data) to the simulated circuit response in the presence of a particular stuck-open fault, Li identified 9 of the 45 Murphy sequence-dependent chips that act as if they contain defects causing such faults, [Li 02]. We have not carried out failure mode analysis to confirm this diagnosis.

Another possible defect that could cause sequence-dependent behavior is one that causes a feedback bridging fault. We have not yet succeeded in diagnosing all of the chips with sequence-dependent test responses and are also trying to diagnose the defective Elf35 chips. This study is continuing.

2.2. Timing-dependent defects

We define a timing dependent defect as one that escapes the test at some speeds within the specification range and is detected at some others of the same test set. The timing dependent defects were found by applying all of our 100% single-stuck test sets at 3 different speeds. Some of the defective chips with sequence-dependent behavior also have output responses that depend on the speed of the test: 105 (32%) of the defective Elf35 chips and 39 (34%) of the defective Murphy chips. Possible causes of such behavior are *resistive-opens*, connections that have significantly higher resistance than intended or transistors with lower drive than designed for.

2.3. Single stuck-at faults

A bare majority (58%) of the defects in the ELF35 chips are *combinational defects*. They cause the faulty chips to continue to act like combinational circuits. Some of these chips might be modeled as having single- stuck-at faults. To investigate this, we used the same technique of matching tester response data with simulated response; in this case the simulation was for circuits with single stuck-at faults, [Li 02]. Only 15 (5%) of the defective Elf35 chips act like circuits with single- stuck-at faults; more of the defective Murphy chips – 41 (35%) – behave like they have single-stuck-at faults.

The ratios of various defect types present in the Elf35 and Murphy chips are summarized in Figure 3 and Figure 4. This data clearly shows that the single-stuck fault model is not an accurate representation of the behavior of a chip in the presence of a manufacturing defect.[§] This suggests that the stuck-at fault model should not be relied on in diagnosing defects on faulty chips. On the other hand, the stuck-at fault model has been very effective when used to generate test patterns. The next section discusses using the stuck-at fault model for applications other than diagnosis.

3. The stuck-at fault model

This section describes using tester data for screening out defective chips rather than diagnosing defects. Most of the defective chips failed all of the test sets that we applied. We call these *FATS or Fail All Test Sets*. Out of the 324 defective ELF35 chips 101 are FOSTS and the 116 Murphy chips include 27 that fail only some test sets. Some of the test sets applied were not very thorough such as the 50% single stuck-at test set. Only the remaining FOSTS (fail only some test sets) chips are relevant to the study in this section of the effectiveness of various test techniques. Thus, the data presented here excludes the FATS chips.

One of the most important roles of the single stuck-at fault model is as a metric for evaluating the thoroughness of a test set. We will discuss this first and then mention some other applications.

[§] Often taken as the definition of a combinational circuit.

* This defect inserts a capacitive dynamic memory.

[§] Multiple stuck faults do not avoid the difficulties of the single-stuck fault model. While there is evidence that some defective chips behave as if they had multiple faults, there are still the issues of sequence dependence and complexity.

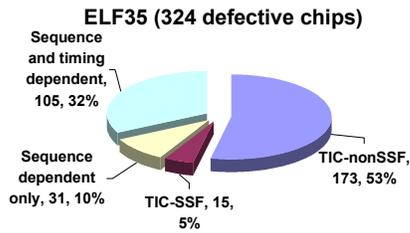


Figure 3. Defect characteristics for ELF35.(TIC = Timing-Independent Combinational).

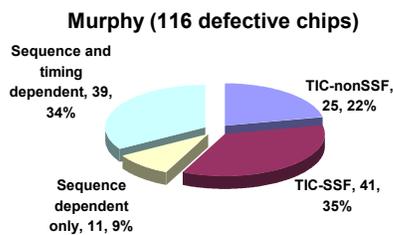


Figure 4. Defect characteristics for Murphy.

Everyone reading this paper knows what the single stuck-at fault model is; or do we each have our own definition? We would probably all agree that some node in the network is fixed at a logic value (0 or 1) independent of the values of any other nodes in the network. The areas of possible disagreement are: (a) which network representation and (b) which nodes should have fixed values.

The network could be represented using the design file made up either of gates from the cell library or using structural RTL. These library gates typically include the elementary gates: AND, OR, NAND, NOR as well as some complex gates such as XOR gates, multiplexers, full adders, etc. Another possible network representation would use only elementary gates, replacing each complex gate with a network of elementary gates having the same functionality.[§] Thus, at least two different network representations are currently used.

The other issue is the set of nodes from which to choose the node with the fixed logic value. The most careful approach is to include all primary inputs, elementary gate inputs and outputs, and primary outputs. This and other models are listed in Table 1.

Models 1, 2 and 4 are each supported by some commercial ATPG tools. There are theoretical results suggesting that Model 5 can be just as effective as Model 1 in generating test patterns [Mei 75].

[§] Some commercial ATPG tools provide an option of deriving this representation automatically. This representation may not correspond precisely to the actual silicon implementation since it isn't always possible to find the correct primitive gate equivalent of a complex gate (the library information may not be exact).

Table 1. List of Single Stuck-at Fault Models.

Model	Fault sites
1. Elementary Gate Faults	All elementary gate inputs, elementary gate outputs, primary inputs and outputs
2. Complex Gate Faults	All library gate inputs, library gate outputs, primary inputs and outputs (RTL faults)
3. Partially Complex Gate Faults	All elementary gate inputs and outputs, fan-out free library elements inputs and outputs, and primary inputs and outputs.
4. Gate-output Faults	All gate outputs (all nets), primary inputs and outputs. Gates can be complex or elementary.
5. Dominance-reduced faults	All inputs and output of fanout-free subnetworks of elementary gates, primary inputs and outputs

The way the single stuck-at fault model is used in connection with test pattern generation is by means of a program that attempts to generate input patterns causing the network output with the fault present in the network to differ from the output of the fault-free network. The *metric* or figure of merit for the set of patterns generated is the single stuck-at fault coverage, the percentage of the modeled faults that are detected by some pattern in the set. Clearly this value depends on which single stuck-at fault model is used.

But the real issue is the effectiveness of the model in producing test sets that detect the defects. We investigated this by generating test sets using each of these models, applying them to our faulty Murphy and Elf35 chips, and determining how many faulty chips were not detected by each of the test sets. A closely related issue is what percentage of the single stuck-at faults is detected by the test set, the *fault coverage*; if the fault coverage is less than 100%, does that mean that more defective chips will escape detection? Results answering this and many other test quality related questions are presented in the next section.

4. Single-stuck test data reduction

In this section, we present data relevant to the impact of test data reduction techniques on the quality of the test. We quantify the quality by the number of defective chips that escape a test set (test escapes).

4.1. Test set compaction

The number of patterns in a test set, *test set size*, is an important characteristic of the test set; it affects the amount of tester memory and test application time [Hamzaoglu 00]. Reducing the test set size is an important goal, especially if it can be done without sacrificing defective chip detection. Commercial ATPG, automatic test pattern generation, tools typically give the user a choice of (1) dynamic test compaction, (2) static test compaction, or (3) no compaction. These techniques take advantage of the fact that test patterns typically contain a large percentage of unspecified (don't care) bits [Barnhart 01]. *Dynamic compaction* is performed by running fault simulation at several stages of the test pattern generation process and dropping the faults that are detected by the generated patterns. *Static compaction* is performed by combining the patterns that don't have any conflicts in the specified bit

positions. Some ATPG tools reverse the order of the generated patterns and then perform simulation and drop the patterns that don't detect additional faults. This is believed to reduce the test data because the faults that are detected with the initial patterns are most of the time easy faults that have a high detectability (many patterns detect them). The last patterns in the test set normally detect low detectability faults. That is why reversing the pattern order eliminates the need for some of them.

Compaction preserves the fault coverage, but since there are fewer patterns it is possible that the defect detection suffers. This is sometimes discussed by calling the ability of patterns to detect defects that don't correspond to single stuck-at faults *collateral coverage* and the corresponding faults *unmodeled faults*. We now know that most of the defects are not accurately represented as single stuck-at fault; thus most of the defects correspond to unmodeled faults. In any event, it is important to determine whether compaction reduces the ability of the test set to detect defects. In order to investigate the effect of compaction on test escapes we tested the ELF chips using both compacted and uncompact test sets. The results are shown in Figure 5. The figure shows the number of escapes that occurred with uncompact test sets and the increase in the escapes due to compaction for various fault coverages. The results in the figure suggest that at all fault coverages used in the experiment, there is a significant price for compaction. Since we are not disclosing the tool used, we are reporting the results from the tool with the maximum increase in escapes at each fault coverage applied.

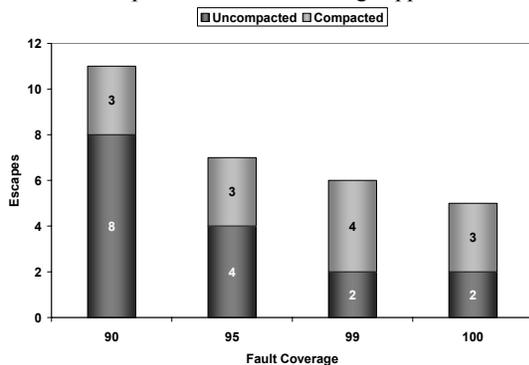


Figure 5 Increase in escapes due to compaction.

shows the test lengths for the uncompact tool 1C test sets and the reduction in test length obtained by compaction. The other tools had similar results. Although compaction preserves fault coverage, the results show that it doesn't preserve defect coverage. On the other hand, the percentage reduction in test size gained by compaction is significant as shown in the summary of this section.

4.2. Fault coverage reduction

Reducing fault coverage requirements is another way to reduce test set size. This option is appealing because it's widely observed that the last small increase in fault coverage requires a considerable number of test patterns. Eliminating these patterns results in a far smaller percentage reduction in fault coverage than in test length.

Table 2. Uncompact Test Lengths and the Compaction Reduction for Tool 1C.

	100% Fault Coverage		99% Fault Coverage		95% Fault Coverage	
	TL	Δ TL Comp	TL	Δ TL Comp	TL	Δ TL Comp
LSI	318	128	317	124	173	27
TOPS	518	202	502	174	310	122
SQR	42	20	40	18	36	13
M12	72	31	58	19	47	19
MA	103	50	66	6	32	0
PB	3176	489	2887	364	2198	174

Using three commercial ATPG tools, we generated test sets with fault coverage varying between 50% and 100% for the ELF35 cores. The number of escapes caused by decreasing the fault coverage is plotted in . Similar results for Murphy cores are plotted in .

The figures show that reducing fault coverage (even by a small fraction) consistently comes with a price in defect coverage. For all tools, the plots show that the higher the fault coverage used the higher the defect coverage achieved. Although it is not an accurate model for the actual defects (as shown in the previous section), the single-stuck fault model is a good measure of the thoroughness of the test.

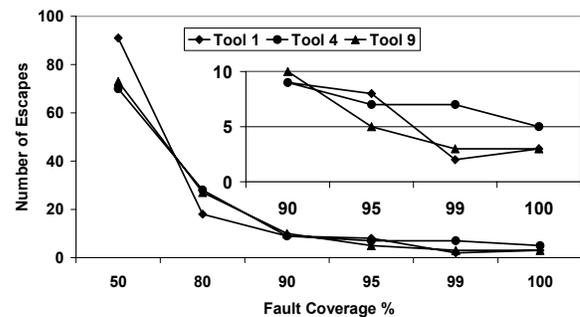


Figure 6. Test escapes vs. fault coverage for compacted test sets generated for complex gates.

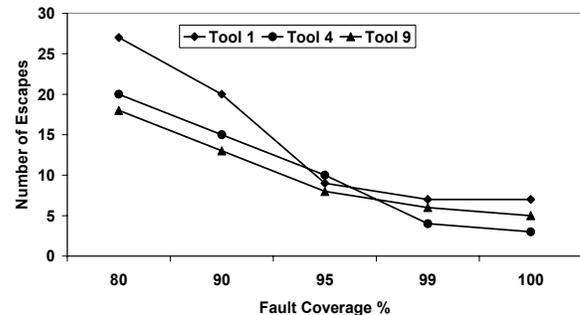


Figure 7. Test escapes vs. fault coverage for compacted test sets generated for complex gates.

4.3. Using complex gates for fault sites

Using complex gates nodes as fault sites instead of elementary gates leads to fewer fault sites in the circuit. This may reduce the number of test patterns required to test the circuit. In a way, placing faults at the RTL pins is an extreme in using complex gates as fault sites. We applied test patterns generated using both the elementary gate fault model and the complex gate fault model (Table 1). The results are shown in . The figure shows the number of escapes using elementary gates and the increase in escapes with complex gates using different fault coverage values (compacted and uncompact). The increase in escapes shown is obtained from the tool that gave the maximum increase in escapes at each fault coverage value. The results demonstrate that there can be a definite penalty in the number of test escapes due to using the complex gate model rather than the elementary gate model. The figure also shows that this applies to compacted and uncompact test sets. Based on these results, complex gate faulting results in a considerable degradation in test quality.

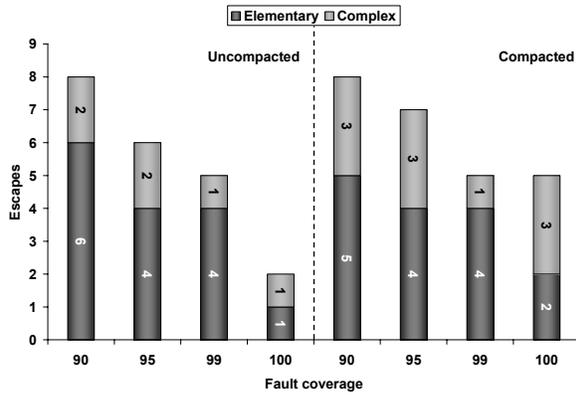


Figure 8. Escapes for complex gates vs. elementary gates as fault sites.

4.4. Using only gates outputs for fault sites

Another technique that is sometimes used to reduce the *fault list*, the number of faults for which to generate test patterns, is to consider stuck faults at only gate outputs rather than at both gate inputs and outputs. This, in effect, eliminates the possibility of a fanout branch having a stuck value while the other branches and the stem are fault-free. The test set is reduced as a consequence of using only gate outputs as fault sites. Table 3 shows the number of tester escapes that occurred for the ELF35 chip when test sets were generated using the design netlist with faults at both gate inputs and outputs compared to test sets with faults only at gate outputs. There is a consistent increase in the test escapes when the gate inputs are not faulted.

Table 3. Comparison of test escapes for faults at both complex gate inputs and outputs and faults only on complex gate outputs – Tool4

Coverage	90%	95%	99%	100%
SSF test for gates inputs and outputs as fault sites	6	4	4	4
SSF test for gates outputs only as fault sites	8	6	5	4
Difference	2	2	1	0

4.5. Assignment of unspecified bits

Compression techniques are widely used to reduce test data. In compression, some coding theory concepts are used to reduce the test pattern storage requirement. Additional decoding circuitry is needed on chip to decode the stored data into the actual test patterns. In many cases, a particular assignment of don't care values is used to maximize the compression ratio. This is the case when run-length encoding is used. Also, for some tester architectures, repeating the last care bit value through the following don't care bits reduces storage requirements.

We tried different don't care bit assignment options to find out their impact on test quality. We implemented *one-fill* (using value 1 for all don't cares), *zero-fill*, *repeat fill* (repeating the last care bit) and *random-fill*. The resulting data is shown in . The results show clear penalties for using the same value to fill the don't care bits.

Table 4. Comparison of different don't care assignment options.

	LSI2901		TOPS2901	
	Length	Escapes	Length	Escapes
One fill	5215	4	8590	5
Zero fill	5215	7	8590	9
Repeat fill	5215	0	8590	0
Random fill	5215	0	8590	0

4.6. Test data reduction summary

The previous subsections presented individual data for various test data reduction techniques. Two main criteria are what matters in test data reduction, the amount of reduction in test data and the impact on test quality. Having the chance to evaluate this impact based on real defect data with our ELF35 chips, we summarized these data for all test set reduction techniques using all tools we have.

Figure 9 shows a graph relating the increase in defective chips that escape the test and the percentage reduction in test set size with different reduction techniques for three commercial tools. The reference for the test reduction and the increase in escapes is a 100% SSF test set generated for elementary gates with each tool. In this graph, the closer the technique is to the bottom left corner the better it is. Reducing the fault coverage requirement to 90% gives the maximum reduction in test data but at the same time increases considerably the number of escapes. The reader is invited to draw all combinations of conclusions that serve his or her interest. An interesting observation is that, for tool 9, reducing the coverage requirement to 95% is better than compaction. It results in further reduction in the test set while maintaining the same quality level. For the other two tools, compaction is even better than reducing the fault coverage to 99%. The inserted graph in the figure has the same data for tools 1 and 4 only. We separated these two tools from the third one because they, unlike tool 9, had comparable results for most of the reduction options.

Another interesting observation is that at 100% coverage, using complex gate pins as fault sites instead of elementary gates did not result in any penalties in test quality for tools 1 and 4. Doing the same with tool 9 resulted in a penalty in test quality. Using only gate outputs

as fault sites was available with one of the tools only and it was worse than compacting the test set.

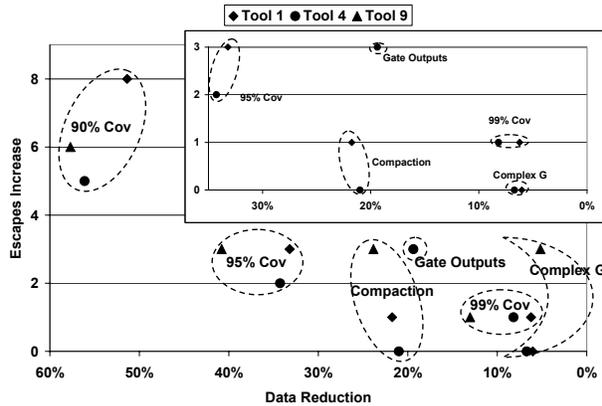


Figure 9. Increase in escapes vs. reduction in test data for the three commercial tools.

5. N-detect results

We learned from results in and related results that higher fault coverage leads to higher defect coverage and that 100% fault coverage tests missed some defects. This made us wonder whether there was some way to generate a more thorough single stuck-at fault test set. One way to do this is to have a test set in which each single stuck-at fault is detected more than once. This is called an *N-detect test set*. In a 2-detect test set, each single stuck-at fault is detected by at least two **different** test patterns [Ma 95], [McCluskey 00].

The data collected by applying test sets with varying fault coverages on the tester is shown in for ELF35 and for Murphy, where 2A is an academic ATPG tool. A *tester escape* or *escape* for short is defined as one defective chip that is not detected by any of the patterns in a particular test set. lists the number of tester escapes for test sets with single-stuck fault coverages varying from 15-detect to 50%. The columns labeled 1C, 4C and 9C correspond to three different commercial ATPG tools. The design file was used to generate the test sets.

Table 5. Number of test escapes vs. fault coverage of compacted test sets for ELF35.

Tools			1C	4C	9C
SSF	N-Detect	15	0	3	-
		10	1	2	-
		5	2	1	-
		3	2	4	-
		2	3	5	-
	Fault	1.00	3	5	3
		0.99	2	7	3
		0.95	8	7	5
		0.90	9	9	10
		0.50	18	28	27
		91	70	73	

The data in the two tables above clearly suggests that the *thoroughness* of a test set measured by the number of defective chips that escape detection by that test set is strongly correlated to the test set fault coverage.

TARO (Transition faults propagated to All Reachable Outputs) from [Tseng 01] was applied to all combinational cores of ELF35 and Murphy. All of the TARO test sets resulted in zero escapes for all cores they were applied to.

Table 6. Number of test escapes vs. fault coverage of compacted test sets for Murphy.

Tools			1C	4C	9C	2A
SSF	N-Detect	15	-	0	-	0
		10	-	-	-	0
		5	-	0	-	0
		3	-	-	-	0
		2	-	-	-	2
	Fault Coverage	1.00	7	3	5	3
		0.99	7	4	6	-
		0.95	9	10	8	-
		0.90	20	15	13	-
		0.80	27	20	18	-

Table 7 Test Escapes of Sequential and Scan Test Sets.

Test set#	Description	LSI2901 (total 89 defective chips)					TOPS2901 (total 29 defective chips)				
		Cov %	Test length	Escapes	Tester data	Tester Time	Cov %	Test length	Escapes	Tester data	Tester time
1C	Scan SSF	100	319	2	386K	174K	100	504	0	1M	485K
4C	Scan SSF	100	193	2	234K	105K	100	386	0	781K	371K
1C-.8	Scan SSF	80	64	5	78K	35K	80	114	1	230K	110K
4C-.8	Scan SSF	80	33	7	40K	18K	80	72	2	145K	69K
3C.s	Seq. SSF	77	517	4**	64K	0.5K	-	-	-	-	-
4C.s	Seq. SSF	75	710	4**	87K	0.7K	-	-	-	-	-
5A	Seq. SSF	78	888	4**	109K	0.9K	65	1,498	3*	154K	1.5K
D.0	Verif.	82	3,121	4**	384K	3K	55	429	12	44K	0.4K
T.3C	Scan transition	100	653	2	80K	356K	80	100	0	10K	96K
T.3C.s	Seq. transition	82	4,070	2*	500K	4K	-	-	-	-	-

* includes one slow escape

** includes two slow escapes

6. Sequential ATPG results

Table 7 compares the test escapes of sequential test sets and scan test sets (also known as structural tests) for the sequential cores in ELF35. Their single-stuck fault coverage and test length are shown for the reader's reference. The test lengths of scan test sets are the number of scan loads. There are 544 cycles in a scan load of LSI2901 and 961 cycles in a scan load of TOPS2901. The test lengths of sequential test sets are the number of system clocks. A dash "-" in this table means the corresponding test set is not available. The tester data column corresponds to the number of bits that need to be stored in the tester for each test set. The tester time column corresponds to the number of clock cycles needed to test the core with the given test set.

The sequential test sets were applied at three different speeds. There are some cores that failed sequential tests at characterized speed and escaped the test only at slow speed. For scan test sets, the system clocks are also applied at three different speeds and the scan load and unload operations are applied at a fixed clock rate of 1MHz.

The table shows that sequential test sets are effective when applied at speed. Figure 3 shows that more than 30% of the defects are timing defects. If the test is not applied at speed then scan testing yields a better test quality.

The verification test sets in Table 7 were provided by the designers. They are fault graded to have 82% SSF fault coverage for one core and 55% for the other. They had 4 test escapes for one core and 12 for the other. The last two rows in the table show the transition fault test sets generated by tool 3C. They both had two test escapes. The test length of the sequential transition fault test set is more than 4 times longer than the SSF sequential test set.

7. Conclusions

We classified the defects based on their behavior and found that even though 35% of the defects in Murphy behaved like SSFs, in the newer chip only 5% of the defects did. We also found that almost half of the defects for both chips exhibited sequential behavior. This suggests that ATPG techniques that ignore the order of the test patterns run the risk of missing many of these defects. For ELF35, only two test sets had no test escapes: TARO and 15-detect.

We studied a number of test set size reduction techniques: compaction, complex gates, gate outputs, fault coverage reduction, etc. Some of the techniques preserved fault coverage but none of them could be relied on to preserve test quality.

Acknowledgements

This research is supported by LSI Logic Corp, Agilent, Intel, NSF and SRC. We would like to thank Guy Dupenloup, Scott Keller and Prabhu Krishnamurthy.

We would like to thank Advantest, Mike Purtell (Advantest), Don Sireci (Advantest), Marc Loranger (Credence), Dr. Sassan Raissi (Digital Testing Services), and Steven Liaw (ARTest) for their donation of tester time.

We would like to thank Michael Grimaila, Gary Greenstein, Ilker Hamzaoglu, Michael Hsiao, Seiji

Kajihara, Rohit Kapur, Ray Mercer, Irith Pomeranz, John Waicucauski, and L.T. Wang for test sets and ATPG tools.

We would like to thank Jonathan Chang, Ray Chen, Eddie Cheng, Kan-Yuan Cheng, Yi-Chin Chu, Siyad Ma, Samy Makar, and Sanjay Wattal from CRC for their help.

References

- [Barnhart 01] K. Barnhart, B. Keller, B. Koenemann, and R. Walther, "OPMISR: The Foundation for Compressed ATPG Vectors", Proc. ITC, 2001.
- [Chang 96] Chang, J., and E.J. McCluskey, "Quantitative Analysis of Very-Low-Voltage Testing," VLSI Test Symposium, pp.332-337, 1996.
- [Chang 98] Chang, J. and et. al., "Analysis of Pattern-dependent and Timing-dependent Failures in an Experimental Test Chip," Proc. ITC, 1998.
- [Franco 95] Franco, P. and et. al., "An Experimental Chip to Evaluate Test Techniques Chip and Experiment Design," Proc. ITC, pp.653-662, 1995.
- [Hamzaoglu 00] Hamzaoglu, I., and J. Patel, "Test set compaction algorithms for combinational circuits," IEEE Trans on CAD, Vol. 19, No. 8, pp. 957-963, Aug. 2000.
- [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, Oct. 1999.
- [Li 02] Li, C.-M.J., and E.J. McCluskey, "Diagnosis of Sequence Dependent Chips," 20th IEEE VLSI Test Symposium (VTS'02), Monterey, CA, Apr. 28-May 2, 02.
- [LSI 97] LSI Logic, "G-10p Cell Based ASIC Product," Feb. 1997.
- [Ma 95] Ma, S.C., P. Franco, and E.J. McCluskey, "An Experimental Chip To evaluate Test Techniques Experimental Results," Proc. ITC, pp.663-672, 1995.
- [Mei 75] Mei, K.C.Y., Dominance Relations of Stuck-at and Bridging Faults in Logic Networks, Ph.D. Thesis, Stanford University, Stanford, CA, June 1975.
- [McCluskey 00] McCluskey, Edward J. and C.W. Tseng, "Stuck-at Fault versus Actual Defects", Proceeding of International Test Conference, pp.336-344, 2000.
- [Tseng 01] C.W. Tseng and E.J. McCluskey, "Multiple-output propagation transition fault test," Proc. ITC, 2001.

Appendix

The Murphy chip, was discussed along with some of its data in [McCluskey 00]. LSI Logic fabricated the Murphy chip in their LFT150K CMOS gate array technology ($L_{eff} = 0.7 \mu$). It has 25k gates in a 120-pin Ceramic PGA package with 96 signal pins. V_{dd} is 5 volts. This paper presents data for the 116 chips that failed at least one of the 265 test sets applied at 3 supply voltages and 4 test speeds. One objective of this paper is to compare the Murphy data with the data collected on the ELF35 chip, a more recent technology.

LSI Logic fabricated the ELF35 chip in their G10P standard cell technology ($L_{eff} = 0.35 \mu$). It has 265k gates in a 272-pin plastic BGA package with 96 signal pins. V_{dd} is 3.3 volts. Over ten thousand chips were tested. This paper presents data for the 324 chips that failed at least one of the 278 test sets applied at 2 voltages and 3 test speeds.