A Study of Undetectable Non-Feedback Shorts for the Purpose of Physical-DFT

Richard McGowen F. Joel Ferguson
Computer Engineering Department
University of California, Santa Cruz
Santa Cruz, CA. 95064

UCSC-CRL-93-25
July 11, 1993

Baskin Center for
Computer Engineering & Information Sciences
University of California, Santa Cruz
Santa Cruz, CA 95064 USA

ABSTRACT

Undetectable shorts may decrease the long term reliability of a circuit, cause intermittent failures, add noise and delay, or increase test pattern generation costs. This paper describes the undetectable non-feedback shorts that are likely to occur in standard cell implementations of the ISCAS'85 combinational test circuits. For the MCNC implementation of the circuits, all shorts between adjacent wires were extracted and the undetectable ones analyzed. We found that approximately 0.2% are undetectable and that nearly half of these can be easily predicted before the physical layout of the circuit is generated. Since only a small percentage of the shorts are undetectable, and many of the undetectables are easily identifiable, it appears that it is possible to reduce the likelihood, or completely eliminate, the occurrence of a large portion of these shorts by incorporating design for test strategies into routing software.

Keywords: DFT, Physical-DFT, Characterization, Non-Feedback Shorts, Undetectable

1 Definitions

A *short* occurs when two or more lines that are not intended to be connected are shorted together. We use the term short, rather than bridge fault, to help make it clear that we are talking about the physical manifestation of a fault rather than a behavioral model. The shorts we consider in this paper are interconnect shorts–shorts between two lines that are gate outputs and/or circuit inputs. We do not consider shorts that are internal to a logic gate. If a directed path, in the graph representing the circuit, exists between the two shorted lines in the original, unfaulted circuit, the short is said to be a *feedback short.* If no such path exists, the short is a *non-feedback short.* This paper considers only non-feedback shorts, techniques for properly evaluating feedback shorts are currently under investigation. A short is *detectable* if for some combination of the circuit inputs, at least one of the circuit outputs has a different logic value than it would in the original circuit with the same input combination. Likewise, a short is *undetectable* if no combination of the circuit inputs will generate a discrepancy on a circuit output (static voltage testing). This means that some of the shorts we catagorize as undetectable are detectable via IDDQ[Ack83] or delay testing. However, these techniques are usually more costly and less accepted than voltage based testing.

2 Motivation

Although an undetectable short does not change the logic function of a circuit, it can change other functional aspects. An undetectable short can cause reliability problems by creating unexpected delay, noise, power consumption, and heat problems. These may introduce intermittent errors or eventually cause a catastrophic failure.

An understanding of the circumstances that contribute to undetectable shorts may be used to develop Design-For-Test (DFT) strategies to reduce the number of such shorts or possibly prevent them completely. The elimination of undetectable shorts would provide three benefits:

1. It would increase the long term reliability of a circuit by reducing power consumption, heat, and average current density caused by undetectable shorts.
2. It would reduce the number of intermittent errors by eliminating electrical noise and circuit delay caused by undetectable shorts.
3. It would reduce test generation costs by decreasing the amount of time spent proving shorts undetectable.[1]

Other researchers have studied how the controllability and observability of shorted lines affects a short's detectability [KBRM91]. This paper focuses on the causes of undetectable shorts, their negative effects, and local characteristics that can be used to predict that a short is undetectable.

[1] The test pattern generation system we used, Nemesis, spent 92.7% of its total test generation time trying to find tests for shorts that it either proved undetectable or aborted on. While this percentage may be a bit high in comparison to other ATPG systems, in general a large amount of time is spent on proving shorts undetectable.

3 Method

We started our work by first generating a list of undetectable shorts. This section describes the method we used to generate the list.

We began by targetting the shorts that occur in an actual circuit layout — all non-feedback shorts that are physically realizable by small spots of metal in the metal1 or metal2 layer of cell interconnect or defects in the oxide separating these layers. Shorts in the metal layers are the most common fault type in many CMOS technologies[MTCC87].

We used Carafe[JF93] to extract the set of likely shorts from the MCNC standard cell implementation of the ISCAS[BF85] combinational test circuits. Once the likely non-feedback shorts were extracted from the physical design of the circuit, we determined their behavior. The two most commonly used methods of modeling shorts are the wired-AND and wired-OR models. However, neither of these adequately reflects the behavior of shorts in CMOS circuits[Ack88, FL91, MG91]. The logic value of shorted nodes is determined by how strongly each gate tries to force its value on the shorted node. Since the strength of a gate is generally a function of its inputs we must consider all inputs to the gates to determine what the actual logic value will be. We computed the logic function of the shorted node using the circuit simulator CaZM[2].

A representative of each class of short that is extracted from the circuit was simulated If there was a potential short between the outputs of a 2-input NAND and a 2-input NOR in the circuit, we simulated the two gates shorted together. However, if no 2-input NANDs were shorted to 2-input NORs, we skipped this class of short. For each class of short in the list, we simulated all 2^{n+m} input combinations for the shorted gates, where n and m are the number of inputs to each gate.

In order to aid the ATPG process by keeping it logic based, rather than voltage based, we used a single logic threshold value. The voltages calculated by the circuit simulator are compared to the threshold voltage of an inverter and translated into logic values which are used to generate truth tables that represent the new function of the paired gates. These truth tables were minimized by Espresso and stored for use during the ATPG process.

Assigning a logic value based on the inverter threshold is somewhat inaccurate in that it does not take into account the differing input thresholds for different gates[AM91]. We would have used a multi-threshold ATPG system if one had been readily available. As we wanted to validate the general idea of being able to find undetectable shorts for the purpose of PDFT, fault simulation with a single threshold value was sufficient and provides more meaningful results than the, typically used, wired-OR and wired-AND models.

The Nemesis[Lar92] ATPG system took the short list and generated tests based on the logic function of each short[FL91]. Nemesis provided a list of all undetected non-feedback shorts. Approximately 74% of the undetected non-feedback shorts were proven undetectable by Nemesis–it aborted on the other 26%. We chose to include both the proven undetectable and aborted shorts in our analysis since most aborted shorts are undetectable, and the few that are detectable that slip through are difficult to test for, at least with the Nemesis ATPG system, and are thus undesirable.

[2] We have since switched to spice as it is more commonly recognized and accepted.

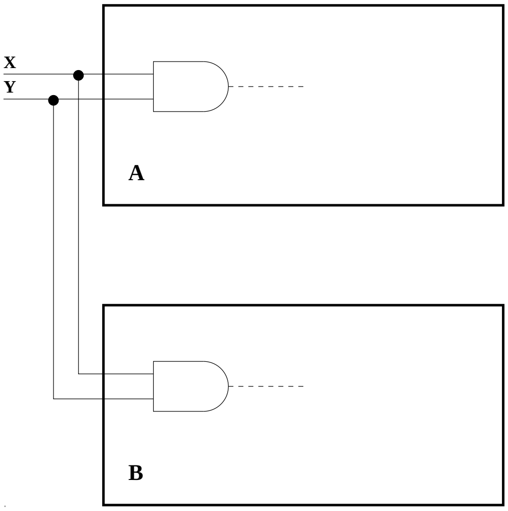

Figure 1: Example of how unintentional redundancy may be introduced.

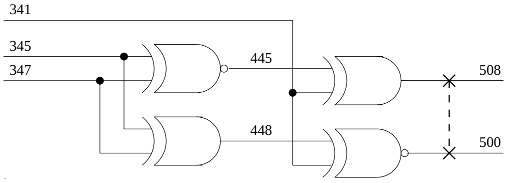

Figure 2: Example of redundant circuitry taken from the 1908.

4 Analysis

In order to detect a non-feedback short you must be able to set the shorted lines to different values in the short-free circuit. We call this *stimulation*. In addition, the resulting discrepancy then needs to be propagated to a circuit output. We call this *propagation*. Therefore, we can first divide undetectable non-feedback shorts into two broad categories: nonstimulatable non-feedback shorts and stimulatable, but nonpropagatable, non-feedback shorts. We will refer to the later as simply nonpropagatable non-feedback shorts. Clearly these two are disjoint sets whose union is the set of all undetectable non-feedback shorts.

For a short between two lines to be nonstimulatable, each node must implement the same logic function. Hence, one of the two lines is redundant. It would appear that one of the lines should be eliminated. Unfortunately it is easy to unintentionally introduce this type of redundancy.

Consider Figure 1. If a circuit designer is using a hierarchical design style. He may have two blocks, A and B, that both require signals X and Y. While designing block A the designer may determine that he needs to AND X and Y for use within block A. While designing block B he may determine that he needs to AND X and Y for use within block B. The designer may not notice that the AND could have been performed at the higher level and sent to each block separately.

It is also possible to introduce redundancies on the same level without realizing it. If the equivalency is created a few gates back from the gates whose outputs are shorted, it might simply be missed. An example taken from the 1908 benchmark circuit, Figure 2, illustrates

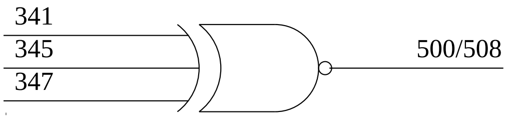

Figure 3: Single gate that performs the same function as the logic in Figure 2.

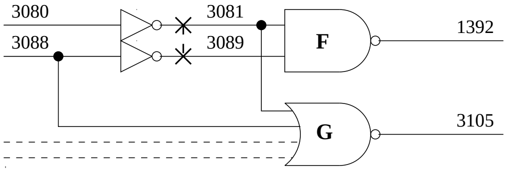

Figure 4: Example taken from 2670 that shows two types of masking that can occur.

this. In the example, lines 500 and 508 are equivalent. Lines 341, 445, and 448 do not go to any other gates so the entire circuit in Figure 2 could be replaced by the circuit in Figure 3.

Equivalence aside, it is undesirable to have lines 500 and 508 shorted together. Line 500 might have a greater load than line 508 and line 508 may be on a critical path. If these two lines are unintentionally shorted, the load will be shared and the delay on the critical path will be greater. This can be detrimental as the corresponding alteration in delay may cause the circuit to fail for some input sequences as a delay fault. Unless this circuit path is checked with a delay test, the short will not be detected.

The second category of undetectable non-feedback shorts, nonpropagatable shorts, are more subtle. All inputs that stimulate a nonpropagatable short also cause the error that is generated to be masked before it reaches a circuit output.

Fault masking can occur when the short performs the same function as a portion of its propagation path. Consider Figure 4, in which the shorted lines 3081 and 3089 are driven by CMOS inverters whose n-channel transistors are stronger than their p-channel transistors. This means that the n-channel transistor will win and a zero will be present upon both lines (wired-AND) when there is a conflict[3]. Therefore, even when the short is stimulated, 3081 and 3089 have different values, the output of gate F will not show a discrepancy. The reason for this is that a zero on either 3081 or 3089 automatically forces F's output to be one. Therefore if 3081 causes 3089 to go low when it should be high the short will be undetectable through line 1392. In some sense, the discrepancy can not be propagated out through 1392 because gate F performs an equivalent function (before the inversion) to the short. In general, an error can be masked whenever a gate on the output propagation path performs the same function as the short. Another example of this type of masking is taken from the 880 benchmark circuit and is shown in Figure 5. In this figure, the undetectable short involving lines 72 and 73 acts as a wired-AND. Because the combination of F, G, and H ANDs 72 with 73, for the input combinations that stimulate the short, the short is masked.

In addition to function masked shorts, stimulation masked shorts can occur. The values placed on lines to stimulate the short may also mask the propagation of the short. If we look at Figure 4, gate G provides an example of this type of masking for a short on lines

[3] Although wired-AND/wired-OR models do not accurately represent all shorts, some shorts do exhibit wired-logic behavior, particularly shorts between the ouput lines of identically sized inverters!

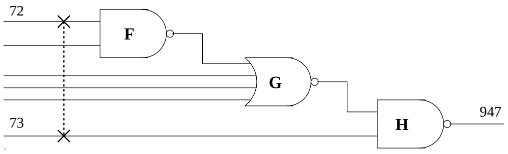

Figure 5: Additional example of masking through function equivalence.

Circuit	Proven Undetectable	ATPG Aborted	Total Not Tested	Total non-feedback shorts
17	0	0	0	5
432	2	5	7	666
499	0	0	0	1577
880	2	1	3	2594
1355	2	1	3	2570
1908	13	3	16	2842
2670	27	10	37	12195
3540	34	0	34	12709
5315	29	2	31	36632
6288	3	0	3	10885
7552	83	47	130	48993
Total	195	69	264	131688

Table 1: Distribution of non-feedback shorts by circuit.

3081 and 3089 if we assume that the short can not be propagated through gate F. For a discrepancy to be propagated through G, and out line 3105, a discrepancy must be put on line 3081. If placing a discrepancy on line 3081 requires line 3080 to be a zero, line 3088 will have to be a one (we need 3081 and 3080 to be at opposite values.) However, placing a one on line 3088 forces the output of G and, in turn, blocks the propagation of the discrepancy.

Nonpropagatable shorts are undesirable for the same reasons as nonstimulatable shorts: they may cause delay faults and increase ATPG time. In addition nonpropagatable shorts can cause further problems. Since the shorted lines can be set to different values, this type of short can cause current consumption to exceed rated values leading to reduced reliability and a violation of specifications. This allows nonpropagatable shorts to be detected by IDDQ tests if the circuit is designed appropriately.

5 Results

As stated earlier, we included both the undetectable and aborted shorts from Nemesis in our analysis. Table 1 shows how many of the non-feedback shorts in the MCNC implementations of the ISCAS circuits were proven undetectable, how many Nemesis aborted on, and the total number of non-feedback shorts. This table shows that approximately 0.2% of the total non-feedback shorts from the list created by Carafe were not detected by the test sets that Nemesis generated.

In order to examine the relationship between nonstimulatable non-feedback shorts, nonpropagatable non-feedback shorts, and non-feedback short undetectability, we used

	Nonstimulatable		Nonpropagatable	
Circuit	LD_0	Non-LD_0	LD_0	Non-LD_0
432	0	1	1	5
880	0	0	2	1
1355	2	0	0	1
1908	2	4	3	7
2670	14	2	2	19
3540	20	0	1	13
5315	17	7	3	4
6288	0	0	0	3
7552	23	18	41	48
Total	78	32	53	101

Table 2: Distribution of LD_0 non-feedback shorts.

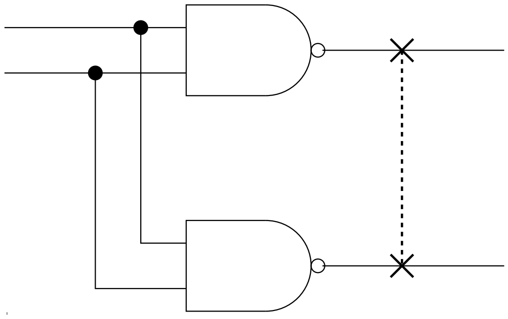

Figure 6: Example of LD nonstimulatable non-feedback short.

the Nemesis ATPG system to divide the set of undetectable non-feedback shorts into nonstimulatable non-feedback shorts and nonpropagatable non-feedback shorts. Nemesis' I_{DDQ} switch causes it to generate tests for shorts by exploiting the excess quiescent current caused by two shorted nodes being driven to different logic values[FTL90]. The undetectable shorts that are non-I_{DDQ} detectable are nonstimulatable.

These two categories of non-feedback shorts, nonstimulatable and nonpropagatable, can be further divided into two sub-categories. Those that can be found by examining local information (not having to look more than one level forward or backward from the short site) can be classified as *Locally Determinable* (LD) and the remaining as *Non-Locally Determinable* (NLD). This is important because LD non-feedback shorts can efficiently (in computational terms) be identified during the physical design of the circuit, and hence can be avoided. We originally considered two types of LD undetectable shorts which we will refer to as the LD_0 class.

For nonstimulatable non-feedback shorts, we found all the LD non-feedback shorts that can not be stimulated because the shorted lines belong to gates of the same type that share the same inputs, as in Figure 6. It turns out that 71% of the nonstimulatable non-feedback shorts can be found this way.

For nonpropagatable non-feedback shorts, we found all the LD non-feedback shorts that can not be propagated because the shorted lines are inputs to the same gate, have no fanout, and perform the same function as the gate. In other words, both shorted lines feed only

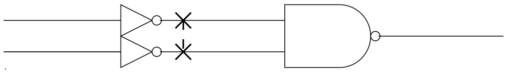

Figure 7: Example of LD nonpropagatable non-feedback short.

into a gate that performs the same function as the short, as in Figure 7[4] It turns out that 34% of the nonpropagatable non-feedback shorts can be found this way.

Table 2 shows the complete division of non-feedback shorts into nonpropagatable and nonstimulatable non-feedback shorts as well as the subdivision of each of these categories into LD_0 and Non-LD_0 categories. Approximately 50% of the 264 undetectable non-feedback shorts were LD_0.

6 Conclusion

Undetected non-feedback shorts can cause intermittent faults, reduce reliability, and increase ATPG costs. In the CMOS circuits we analyzed, approximately 0.2% of the non-feedback shorts were not detectable by the Nemesis ATPG system.

We showed that the undetectable non-feedback shorts can be split into two, fairly even, categories, nonstimulatable and nonpropagatable. For each of these categories, we have identified several characteristics that can make the non-feedback short undetectable and gave examples of each. This resulted in our identifying several classes of undetectable shorts that can be proven undetectable with very limited local information. Two classes of LD shorts, called LD_0 short, make up almost half of the undetectable shorts in the analyzed circuits.

These are encouraging results since they suggest that we may be able to eliminate many undetectable shorts in a circuit with little penalty. Since less than 0.3% of the total non-feedback shorts were undetected, changing the physical design of the circuit to prevent the possibility of these shorts may not impact the area, and hence cost, of the circuit significantly. For almost 50% of the undetectable shorts, local information is sufficient to prove the short is undetectable. For these LD undetectable shorts, computationally feasible physical-DFT rules in the routing portion of the design process could prevent the placement of the involved wires adjacent to each other and thus reduce the number of possible undetectable shorts. For cases in which it is impossible to keep wires from being adjacent, such as adjacent gate inputs, the router should still be able to keep the wires from being adjacent for long distances and thus reduce the probability of a undetectable short occurring.

Acknowledgements

The authors thank Brian Chess and Tracy Larrabee for modifying Nemesis to provide the undetected shorts, J. Alicia Grice for software that automatically creates the truth tables for Nemesis, and Krzysztof Kozḿiński at MCNC for providing us with standard-cell layouts of the ISCAS benchmarks. We also thank the Semiconductor Research Corporation for supporting this research under Contract 92-DJ-315 and The National Science Foundation for supporting this work under grant MIP-9158491.

[4] This example assumes that an inverter-inverter short acts as a wired-AND. It does in the standard cell library we use, MCNC SCMOS.

References

[Ack83] J.M. Acken. Testing for bridging faults (shorts) in CMOS circuits. *Proceedings of Design Automation Conference*, pages 717–718, 1983.

[Ack88] John M. Acken. *Deriving Accurate Fault Models.* PhD thesis, Stanford University, Department of Electrical Engineering, September 1988.

[AM91] J.M. Acken and S.D. Millman. Accurate modeling and simulation of bridging faults. *Proceedings of the Custom Integrated Circuits Conference*, pages 17.4.1–17.4.4, 1991.

[BF85] F. Brglez and H. Fujiwara. A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, 1985.

[FL91] F. Joel Ferguson and Tracy Larrabee. Test pattern generation for realistic bridge faults in CMOS ICs. In *Proceedings of International Test Conference*, pages 492–499. IEEE, 1991.

[FTL90] F. Joel Ferguson, Martin Taylor, and Tracy Larrabee. Testing for parametric faults in static CMOS circuits. In *Proceedings of International Test Conference*, pages 436–443. IEEE, 1990.

[JF93] Alvin Jee and F. Joel Ferguson. Carafe: An inductive fault analysis tool for CMOS VLSI circuits. In *Proceedings of the IEEE VLSI Test Symposium*, 1993.

[KBRM91] R. Kapur, K. Butler, D Ross, and M.R. Mercer. On bridging fault controllability and observability and their correlations to detectability. In *Proc. 2nd Annu. European. Test Conf.*, pages 333–330, 1991.

[Lar92] Tracy Larrabee. Test pattern generation using boolean satisfiability. *IEEE Transactions on Computer-Aided Design*, pages 4–15, January 1992.

[MG91] S.D. Millman and J.P. Garvey. An accurate bridging fault test pattern generator. In *Proceedings of International Test Conference*, pages 411–418. IEEE, 1991.

[MTCC87] W. Maly, M.E. Thomas, J.D. Chinn, and D.M. Campbell. Double-bridge test structure for the evaluation of type, size and density of spot defects. Technical Report CMUCAD-87-2, Carnegie Mellon University, SRC-CMU Center for Computer-Aided Design, Dept. of ECE, February 1987.