

Elimination of Undetectable Shorts During Channel Routing

Richard McGowen

F. Joel Ferguson

UCSC-CRL-93-48
November 15, 1993

Board of Studies in Computer Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064

ABSTRACT

In this paper we present a procedure for reducing the probability of undetectable shorts occurring in routing channels. This procedure is implemented by modifying a channel router to predict many of the shorts that are undetectable and place the associated signal wires in non-adjacent tracks. For the designs we routed, we found that the probability of an undetectable non-feedback short between horizontal lines in the routing channels was reduced by 32.4% with no increase in the number of routing tracks required and little increase in routing computation cost.

Keywords: Channel routing, undetectable shorts, design-for-test

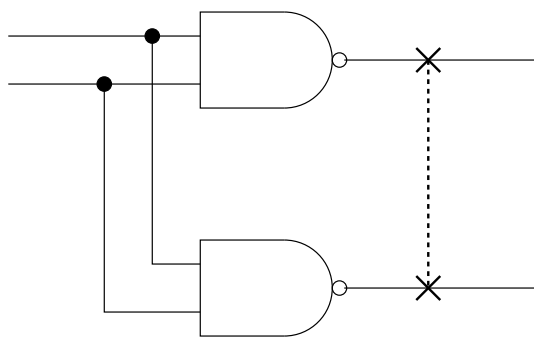


Figure 2.1: Example of a nonstimulatable non-feedback short.



Figure 2.2: Example of a nonpropagatable non-feedback short, when the short causes wired-AND behavior.

1 Introduction

For any given circuit, there are some defects that, should they occur, are likely to not be detected. Since shorts in the metal layers are the most common fault type in many CMOS technologies [MTCC87], shorts in metal interconnect lines may represent a considerable portion of undetectable shorts. It would be beneficial to change the routing of cell interconnect to reduce the likelihood of such shorts occurring. With this in mind we modified a channel router to reduce the occurrence of undetectable non-feedback shorts between horizontal lines. In this paper we explain our modifications, report the improved defect coverage, and show how further improvements in defect coverage may be obtained.

2 Motivation

A short that does not change the logic function of a circuit is unlikely to be detected by logic tests. However, such a short may change a circuit's ability to meet its specifications, make it more vulnerable to intermittent faults, or decrease its reliability. Changes in specification may include increases in delay and power dissipation. As an example of increased delay consider Figure 2.1. If the lower NAND's output is lightly loaded, to decrease the propagation delay on that path, and the upper NAND is heavily loaded, a short shown between these outputs would increase the delay of the lower path. As an example of increased power dissipation consider the wired-AND short between the CMOS gates in Figure 2.2. Normally there would be very little steady state current (I_{DDQ}) in the inverters, but the short would raise the steady state current to the milliamperere range when the inputs are at different logic values. Such changes may cause intermittent failures by changing circuit timing as described above for Figure 2.1 or by reducing noise margins as described above for Figure 2.2. A decrease in reliability could arise from the increased current densities and temperatures caused by increased power dissipation. Hence, eliminating undetectable shorts provides four potential benefits:

1. A reduction in potentially undetected I_{DDQ} or delay faults caused by undetectable shorts.
2. A reduction in the number of intermittent errors by restoring noise margins and eliminating the excess circuit delay caused by undetectable shorts.
3. An increase in the long-term reliability of a circuit by eliminating the excess power consumption, heat, and current density caused by undetectable shorts.
4. A reduction in test generation costs by decreasing the time spent attempting to generate tests for undetectable faults.

Other researchers have presented design rules that decrease the probability of occurrence of hard-to-detect faults by altering the physical layout of a circuit. Koeppel provided layout rules that decrease the likelihood of hard-to-detect stuck-open faults [Koe87]. Levitt and Abraham considered the problem of enhancing testability through the modification of cell layouts [LA90]. Teixeira *et al.* suggested dividing realistic faults into fault classes based upon their resistance to stuck-at test sets and recommended using the information to focus efforts on improving the detectability of hard to detect fault classes with large numbers of faults [TTA⁺91]. Saraiva *et al.*, in a later related paper, advocated the use of routing to improve testability by relaxing the spacing between adjacent lines [SCS⁺92].

Like Saraiva, we intend to modify the physical design of the wiring in the channel to eliminate undetectable shorts. The difference in our approach is that, instead of targeting a class of shorts that may be undetectable, we target shorts that are known to be undetectable. Since we are targeting specific shorts instead of a large class of shorts, we can reorder the signals in the track instead of increasing the spacing between signals. This allows us to eliminate the possibility of selected shorts occurring with little or no impact on chip area.

The remainder of this paper is organized as follows. First we provide an analysis of the undetectable non-feedback shorts within the channels of standard cell (CMOS) based designs, considering only shorts between adjacent signal wires in the routing channels. We then discuss the results of modifying a track-by-track channel router to predict which potential shorts are not detectable and having it use the information to avoid placing lines in adjacent tracks if a short between them will be undetectable. Results are given for runs on over 100 layouts which show that the likelihood of an undetectable fault is substantially reduced with no silicon area overhead. Finally, we show how this work can be extended.

3 Analysis of Shorts in the Routing Channels

We used the Nemesis [Lar92] ATPG system to generate tests and provide a list of all undetected non-feedback shorts. We considered only *Non-Feedback* (NFB) shorts as the system's handling of Feedback (FB) shorts is not yet accurate. For this paper, a short is *undetectable* if Nemesis can not generate a logic-level (static voltage testing) bridge fault that detects it. Note that some of the shorts we define as undetectable may be detected via excess I_{DDQ} [Ack83] or propagation delay.

We analyzed the undetectable shorts to find characteristics that would identify them with limited computational effort. We say that shorts that exhibit these characteristics are *Locally Determinable* (LD) as undetectable since one need only consider local circuit information to determine that they are undetectable. In this paper we consider only a simple class of LD shorts, the LD_0 class. LD_0 consists of two types of shorts. The first type are nonstimulatable (NS) shorts, shorts that can not have a discrepancy placed on the shorted

wires. The NS LD₀ shorts are NS because the shorted wires are outputs of the same type of gate and these gates share the same inputs. An example is shown in Figure 2.1. The second type are nonpropagatable (NP) shorts, shorts that cannot have a discrepancy propagated to a circuit output. The NP LD₀ shorts are NP because the shorted wires are inputs to the same gate, have no fanout, and perform the same function as the gate or its negation. In other words, both shorted lines feed only into a gate that performs the same function as the short. An example is shown in Figure 2.2—assuming that an inverter-inverter short acts as a wired-AND, as it does in the standard cell library we are using.

In order to collect our data, we used ten separate layouts of each of the ISCAS [BF85] combinational test circuits. The layouts were generated by TimberWolf [SSV85], a placement and global routing package, and a modified version of Yoeli’s channel router [Yoe91]. We used the MCNC Standard Cell Library that comes with Oasis. To determine the detectability of shorts within the layout, we used Carafe [JF93] to extract the set of likely shorts from the interconnect and Nemesis to generate tests for the shorts that Carafe extracted.

Carafe is a realistic fault extractor that, using an actual layout, calculates which shorts can occur if a spot defect of a given radius modifies a circuit during the manufacturing process. In our experiments, for shorts between lines in the same layer we used a defect radius that is almost large enough to bridge three wires if they are minimally spaced and for shorts between lines in different layers we used a defect radius that roughly corresponds to the actual line overlap.¹ Carafe reported which faults could occur as well as the total area within which the occurrence of a spot defect would cause two lines to short. This area is called the *critical area* of the short. Critical area information allows us to report defect coverages. Defect coverages are more closely related to quality levels than fault coverages as they measure the change in probability of a given fault occurring. If the probability of an undetectable short occurring changes by 90%, for the better or for the worse, the defect coverage will represent this change—fault coverage will not.

Table 3.1 shows the total number, across all ten runs, of non-feedback shorts that were not detected. This table shows that approximately 0.4% of the total non-feedback shorts were undetectable. Table 3.2 shows the complete division of non-feedback shorts into nonpropagatable and nonstimulatable categories. Approximately 47% of the 1470 undetectable non-feedback shorts were LD₀.

These results were encouraging since they suggested many undetectable shorts in a circuit may be eliminated with little penalty. Since less than 0.4% of the total non-feedback shorts were undetected, changing the physical design of the circuit to prevent the possibility of these shorts occurring should not impact the area, and hence cost, of the circuit significantly. Since almost 50% of the undetectable shorts are in the LD₀ class, it should be computationally feasible to add physical-Design-For-Test rules to a channel router to prevent placing almost half of the undetectable line-pairs adjacent to each other.

We modified a channel router so that it identifies LD₀ shorts during the routing process and alters the route to avoid placing lines in adjacent tracks if a short between them will be undetectable. The router and its modifications are described in Appendix A. The modifications target the LD₀ shorts between horizontal wires in the routing channels. We focused on eliminating undetectable shorts in the horizontal tracks for three reasons:

¹In exact terms, we set the defect radius equal to $((\text{line_width} + (2 * \text{line_spacing})) - 0.5 \text{ microns}) / 2$ for shorts between lines in the same layer and to 0.25 microns for shorts between lines in different layers.

Circuit	Total	Total Not Detected
17	45	0
432	1987	77
499	5378	2
880	11016	44
1355	7020	6
1908	14686	94
2670	29318	254
3540	58477	194
5315	97548	166
6288	59385	1
7552	98434	632
Total	383294	1470

Table 3.1: Distribution of non-feedback shorts by circuit. Totals are for all 10 runs.

Circuit	Nonstimulatable		Nonpropagatable	
	LD ₀	Non-LD ₀	LD ₀	Non-LD ₀
17	0	0	0	0
432	0	4	54	19
499	2	0	0	0
880	0	0	24	20
1355	0	0	0	6
1908	14	13	27	40
2670	30	15	62	147
3540	125	6	32	31
5315	41	24	34	67
6288	0	0	0	1
7552	36	66	216	314
Total	248	128	449	645

Table 3.2: Distribution of LD₀ non-feedback shorts. Totals are for all 10 runs.

1. The critical area totals for horizontal shorts is larger than the critical area totals for vertical shorts. Shorts between horizontal wires accounted for 52% of the total critical area—shorts between vertical wires accounted for 38% of the total critical area. (See Table 3.3. Metal1 lines run horizontally and Metal2 lines run vertically.)
2. Horizontal wires can run adjacent to each other for much longer distances than vertical wires. Eliminating a potentially undetectable horizontal short should on average provide more potential benefit than eliminating a potentially undetectable vertical short.
3. Horizontal wires have more opportunity to have their placement altered than do vertical wires. The placement of vertical wires is, at least where the signals enter the cells, determined by the cell's placement and the location of the cell inputs and

Horizontal	Metal1-Metal1	751310303600
Oxide	Metal1-Metal2	142343122500
Vertical	Metal2-Metal2	553437108664

Table 3.3: Total Critical Areas for all shorts before testability enhancements in square centimicrons.

outputs.

4 The Modified Router’s Performance

The modified router’s performance is measured along three dimensions: the change in critical area of undetectable shorts, the change in routing area, and the change in routing time. To find the change in the performance of the router due to the modifications described in Appendix A, we ran the modified router on the same ten placements of the ISCAS benchmark circuits, 110 physical designs in all. The shorts in the channel were extracted by Carafe and tests were generated for the non-feedback shorts by Nemesis.

Circuit	Total Metal1-Metal1	Total Metal1-Metal2	Total Metal2-Metal2
c17	0	0	0
c432	36415000	8707500	98388125
c499	3030625	405000	0
c880	46720625	5467500	54471250
c1355	3061250	810000	1455625
c1908	78605624	17010000	159177500
c2670	182283750	39690000	382065625
c3540	215434998	39487500	327166875
c5315	174897499	29565000	241298750
c6288	630625	0	630625
c7552	599043123	100035000	1229871250
Total	1340123119	241177500	2494525625
Percentage of Total	32.9%	5.9%	61.2%

Table 4.1: Critical areas of undetectable NFB shorts in original layouts in square centimicrons.

Since the critical area of each short is proportional to its probability of occurrence, the percent change in critical area relates to the percent change in probability of an undetectable non-feedback short occurring. Table 4.1 shows the critical area totals for undetectable non-feedback shorts in the original layouts. As Metal1 is used for the horizontal wires, the Metal1-Metal1 totals represent the shorts that the modifications to the router are targeting. Note that the undetectable Metal1-Metal1 shorts account for only 33% of the total critical area. This is contrary to our expectation since the Metal1-Metal1 critical area of all shorts, both detectable and undetectable, is greater than the critical areas of the Metal1-Metal2 and the Metal2-Metal2 shorts (Table 3.3). Such a large percentage of the undetectable shorts

occurring between adjacent Metal2 lines suggests that shorts between cell inputs cause a majority of the undetectable NFB shorts. Note also that the critical area of undetectable "crossover" shorts, caused by defects in the oxide between Metal1 and Metal2, is less than 6%. This is encouraging since the total is low and placing restrictions on where nets cross would be difficult and costly.

Circuit	Total Metal1-Metal1	Total Metal1-Metal2	Total Metal2-Metal2
c17	0	0	0
c432	14600625	8505000	90135000
c499	0	405000	0
c880	37086875	5467500	54471250
c1355	3061250	810000	1455625
c1908	53184374	17010000	151824375
c2670	153832500	40095000	383003125
c3540	41821875	40297500	292817500
c5315	117407499	29565000	236526250
c6288	630625	0	630625
c7552	484698123	101250000	1221144375
Total	906323746	243405000	2432008125

Table 4.2: Critical areas of undetectable NFB shorts using modified router in square centimicrons.

	Metal1-Metal1	Metal1-Metal2	Metal2-Metal2	All Layers
Original	1340123119	241177500	2494525625	4075826244
Modified	906323746	243405000	2432008125	3581736871
Decrease	32.4%	-1%	2.5%	12.1%

Table 4.3: Change in critical area of all undetectable shorts using Modified Router. (Square centimicrons.)

Table 4.2 shows the critical area totals for the undetectable shorts in the modified layouts and Table 4.3 shows the difference in absolute values of critical areas. These tables show that the critical area of undetectable horizontal shorts was reduced by roughly 32.4% using only LD₀ information without adversely affecting the critical area of undetectable vertical and cross-over shorts.

	Metal1-Metal1	Metal1-Metal2	Metal2-Metal2	All Layers
Original	479094373	78570000	1306465001	1864129374
Modified	49556250	80190000	1246579376	1376325626
Decrease	89.7%	-2.1%	4.6%	26.2%

Table 4.4: Change in critical area of LD₀ shorts using Modified Router. (Square centimicrons.)

If we compare only LD_0 shorts the improvement is more dramatic. Of the total critical area of undetectable LD_0 horizontal shorts, the modified router eliminated all but 10.3% (Table 4.4). Hence the router eliminated nearly all shorts that it targeted. The LD_0 shorts that were not eliminated had one signal wire in the last track to be placed in the channel or were not eliminated as the signal wires were placed into the same track and their “ends” could short together. The first type can not be eliminated without changing the underlying routing algorithm and the second type will be eliminated if vertical shorts are targeted.

With the unmodified router there were 13540 total channels in the 110 placements. With the modified router there were no additional tracks used. Hence, for these designs there is no overhead in silicon area associated with placing problematic signal wires into non-adjacent tracks.

With the unmodified router the total time for routing the 110 placements was 155 cpu seconds on a SPARCstation 1+. For the modified router the total time for the same placements was 164 cpu seconds.

In summary, the probability of a undetectable non-feedback short occurring in the routing of the circuit is reduced by 12.1% (32.4% if only shorts between horizontal lines are considered) with no increase in silicon area, and little impact on routing time. Note that these are complete reductions since at least one net is usually placed in a track between the targeted nets. Therefore any defect large enough to bridge the two target nets would also bridge any nets placed between them.

5 Conclusions and Future Work

The modified router was able to eliminate 89.7% of all undetectable NFB shorts that it targeted, the LD_0 shorts in the horizontal tracks. The remaining undetectable non-feedback shorts were non- LD_0 shorts in the horizontal tracks, undetectable shorts between Metal2 wires, and undetectable shorts between Metal1 and Metal2 wires. As the Metal2-Metal2 shorts are responsible for 61% of all undetectable shorts and 70.1% of the LD_0 shorts, this suggests that the greatest further improvement in eliminating undetectable NFB shorts would result from targeting the shorts between vertical wires.

	Metal1-Metal1	Metal1-Metal2	Metal2-Metal2	All Layers
Original	1340123119	241177500	2494525625	4075826244
Modified	146443750	245430000	2327174999	2719048749
Decrease	89.1%	-1.8%	6.7%	33.3%

Table 5.1: Change in critical area of all undetectable shorts using ATPG to generate a list of undetectable faults. (Square centimicrons)

To further reduce the critical area totals of undetectable shorts in the horizontal tracks would require a more sophisticated LD algorithm, perhaps one that would consider logic two levels deep. To determine the affect on the critical areas if we were able to predict more undetectable NFB shorts, we used the ATPG system to tell us which NFB shorts in the original, unmodified routes, were undetectable. We then used this list of undetectable shorts as a list of shorts to avoid. Using this list 89.1% of the critical area of undetectable shorts in the horizontal tracks was eliminated (Table 5.1). The remaining 10.9% consists of new undetectables introduced when separating the known undetectables and undetectables

that results from nets being placed in the center track or abutting horizontally within a spot defect’s radius. There was a 0.04% increase in number of channels in the 110 placements—the number of tracks required increased from 13540 to 13545. This shows that if a more sophisticated LD class were used the router could take into account more of the undetectables without seriously affecting silicon area. In addition, it also suggests that if the resources are available that you could make the routing process an iterative one by using the steps below.

1. Route the design.
2. Use a bridge fault ATPG system to find the undetectable faults.
3. Update the undetectable list based on the results of Step 2.
4. Re-route the design using the list from Step 3 to decide which lines should not be placed adjacent to each other.
5. Repeat Steps 2-4 until the amount of new undetectables found in Step 2 falls below a specified threshold.

While this will yield good results, we feel that the ideal system is one in which the router is only ran once and can find predict enough of the undetectable faults to generate better routes without requiring the expense of iterating through the above cycle—performing multiple iterations of routing, fault extraction, and test pattern generation can be expensive.

The work in this paper considered only non-feedback shorts. We feel that feedback shorts should also be targeted by the router. The method we present should be extendable to undetectable feedback shorts once a technique for efficiently predicting their behavior is developed.

We showed an improvement in the defect coverage of shorts for an ATPG system that directly targets shorts. One would expect a similar increase in coverages, although probably not as large, would occur if random patterns, single stuck-at patterns, or functional patterns were used. We have yet to verify that this occurs and, if it does occur, to what extent there is a reduction.

6 Acknowledgments

This research was supported by the Semiconductor Research Corporation under Contract 92-DJ-315 and National Science Foundation grant MIP-9158491. We gratefully acknowledge their support.

A The Channel Router and its Modifications

The channel router is a modified version of Uzi Yoeli’s channel router [Yoe91]. The key feature of the channel router is that it places nets on a track by track basis. It first fills the top track with nets, then the bottom track, and then continues alternating between filling the top-most free track and the bottom-most free track until all the nets have been placed. For each track it selects the set of nets that “best” fill the track with selection being done based on the weight of each net. For each track under consideration, each unplaced net has a weight that reflects the desirability of placing the net in the current track. The greater the weight of the net, the more desirable it is to place the net in the track. In addition, a net is assigned to only one track; the net is not spread out over several tracks. Figure A.1 gives an example of this process.

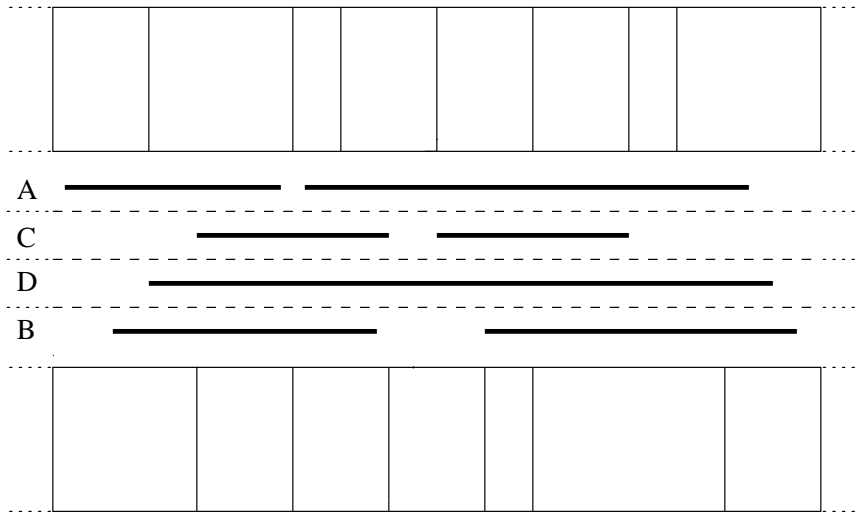


Figure A.1: Nets are placed in tracks, alternating between the top-most and bottom-most free track. For example, A would get filled, then B, then C, and, finally, D.

Since the nets are placed on a track by track basis, this means that when we go to place a net in an interior track we know which nets were placed in the track immediately preceding it on the same side of the channel. Since this information is available, we can check if placing a net in the current track will cause it to be placed adjacent to a net that, should a short between them occur, will cause an undetectable short. If this is the case, we can defer the placement of the net to a later track to avoid this situation. Because the assignment of nets is based on weights, we can recommend that a net not be placed in a track by penalizing its weight.

Being able to influence the placement of a net by changing its weight is appealing since, if we are careful, the underlying routing algorithm need not change. The only drawback is that once the center of the channel is reached, the last net cannot be moved without increasing the width of the channel.

In order to understand how we modified the channel router, it is useful to understand how the weighting scheme works. Yoeli assigns a weight to each net so that the most desirable net is the one with the highest weight. Each time we select a track to place nets in, we calculate the weight of all unassigned nets. The weight of each net is calculated as follows:

1. Set the weight of the net equal to zero.
2. For each column in which the net has a connection to the channel edge that is closest to the track under consideration, add the density of the column (number of nets that originate at or cross the column) to the weight of the net. Consider the case in which we are placing nets in a track that is closer to the top of the channel than it is to the bottom of the channel. If a net has two connections to the top of the channel, one in a column whose density is 1, and one in a column whose density is 2, this step would assign a weight of 3, to the net.
3. For each column in which assigning the net to the current track will create a Vertical Constraint Violation, VCV, (a situation in which a net connected to the top of a column must connect to a track near the bottom of the channel and a net connected to the bottom of the same column must connect to a track near the top of the

channel), subtract the quantity *column density* * VCVWEIGHT from the weight of the net. VCVWEIGHT is a parameter that tells us how much to penalize a net if it will cause a VCV; we set its value to 16. The subtraction is performed only if placing the net in the track under consideration will cause a VCV. If a VCV already exists in the column because of the placement of a previous net, we do not penalize the weight of the net.

4. For each column that the net crosses in which the number of unrouted nets originating at or crossing the column (remaining density) is equal to the number of remaining tracks, add a large number, which Yoeli calls DENSECOL, to the weight of the net. Yoeli sets DENSECOL equal to 10,000. The reason for adding DENSECOL to the weight of a net is that you need to place one of a dense column's net down each iteration or you will run out of tracks.

Yoeli chose the above weighting scheme with three goals in mind:

1. The number of Vertical Constraint Violations must be kept to a minimum.
2. The length of each vertical line should be kept to a minimum.
3. If either of the above rules is to be violated, it should be done in an area of minimum density.

By trying to satisfy the above three goals, the chances of resolving any VCVs that cannot be avoided through the use of maze routing is improved.

With this in mind, we needed to be careful about how we modified the weighting scheme. Since we are penalizing nets, this means we should subtract from a net's weight. However, we must be careful not to offset the value of DENSECOL too much or we can violate the integrity of the algorithm and cause it to be unable to generate a valid route. We chose to penalize a net 500 weighting units for each net, which could cause an undetectable short, that it would be adjacent to if placed in the current track. Since we don't expect, in the designs we are working with, for more than 2 or 3 undesirable nets to be placed adjacent to any single net, a penalty of 1000 to 1500 is dwarfed by DENSECOL and hence won't "break" the router. DENSECOL needs to be kept large enough so that at least one of the nets crossing a dense column will always be placed or the integrity of the algorithm may be violated. There is still room for adjusting our penalty weight and weighting scheme, but we found that this scheme is sufficient.

References

- [Ack83] J.M. Acken. Testing for bridging faults (shorts) in CMOS circuits. *Proceedings of Design Automation Conference*, pages 717–718, 1983.
- [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.
- [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.
- [Koe87] Siegmur Koeppe. Optimal layout to avoid CMOS stuck-open faults. In *Proceedings of Design Automation Conference*, pages 829–835. IEEE, 1987.
- [LA90] Marc E. Levitt and Jacob A. Abraham. Physical design of testable VLSI: Techniques and experiments. *IEEE Journal of Solid-State Circuits*, 25(2):474–481, April 1990.
- [Lar92] Tracy Larrabee. Test pattern generation using boolean satisfiability. *IEEE Transactions on Computer-Aided Design*, pages 4–15, January 1992.
- [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.
- [SCS⁺92] M. Saraiva, P. Casimiro, M. Santos, J.T. Sousa, F. Gonçalves, I Teixeira, and J.P. Teixeira. Physical DFT for high coverage of realistic faults. In *Proceedings of International Test Conference*, pages 642–647. IEEE, 1992.
- [SSV85] C. Sechen and A. Sangiovanni-Vincentelli. The timberwolf placement and routing package. *IEEE Journal of Solid-State Circuits*, April 1985.
- [TTA⁺91] J.P. Teixeira, I.C. Teixeira, C.F.B. Almeida, F.M. Gonçalves, and J. Gonçalves. A methodology for testability enhancement at layout level. *Journal of Electronic Testing: Theory and Applications*, 1(4):287–300, January 1991.
- [Yoe91] Uzi Yoeli. A robust channel router. *IEEE Trans. on Computer-Aided Design*, 10(2):212–219, February 1991.