Simulating Open-Via Defects

Stefan Spinner Jie Jiang Ilia Polian Piet Engelke Bernd Becker

Computer Architecture Group
Institute for Computer Science
Albert-Ludwigs-University
Georges-Köhler-Allee 51
D-79110 Freiburg i. Br., Germany
{stspinne|jiang|polian|engelke|becker}@informatik.uni-freiburg.de

Abstract

Open-via defects are a major systematic failure mechanism in nanoscale manufacturing processes. We present a flow for simulating open-via defects. Electrical parameters are extracted from the layout and technology data and represented in a way which allows efficient simulation on gate level. The simulator takes oscillation caused by open-via defects into account and quantifies its impact on defect coverage. The flow can be employed for manufacturing test as well as for defect diagnosis.

Keywords: Open-via defects, Fault simulation, Defect modeling

1 Introduction

Open-via defects are a major yield detractor in nanoscale CMOS [1, 2]. In state-of-the-art dual damascene copper interconnect technology, vias are filled simultaneously with metal deposition, which is error-prone due to high aspect ratios of vias [3]. Detecting and diagnosing open-via defects is difficult because traditional fault models do not adequately describe these defects. In particular, the logic value on an interconnect affected by an open-via defect (victim line) can become dominated by second-order effects due to parasitic cross-capacitances with other interconnects. Traditional fault models do not reflect such dependency.

A state-of-the-art modeling approach for open-via defects has been proposed in [2] in the context of fault diagnosis. For every via in question, the expected output behavior has been computed and validated against the values observed by the test equipment. By employing the method, a diagnostic resolution exceeding that of any purely gate-level approach has been achieved. For instance, failing via candidates on the same interconnect or on logically equivalent interconnects could be distinguished from each other. This allows to pinpoint the vias which frequently fail to guide the yield learning, i.e., the improvement of the manufacturing process. The approach has been applied to hard-to-diagnose failing ICs and its results have been validated by physical failure analysis.

In this work, we demonstrate a flow to extract and simulate open-via defects. First, the locations of vias in the circuit layout are identified and matched with the gate-level net-list. Then, aggressor lines and the corresponding cross-capacitances are determined. Commercial tools are used to extract this low-level physical parameters. Finally, a given test set is simulated to determine the open-via defect coverage using a dedicated gate-level simulator. The simulator is based on the model from [2].

We show that, even when restricted to the model from [2], an open-via defect may lead to oscillation similar to feedback bridging faults [4]. We incorporate handling of oscillation into the simulator. We report open-via defect coverages for a number of test sets taking oscillation effects into account. It turns out that the coverage of open-via defects by 1-detection and 3-detection stuck-at test sets is rather limited.

Models of interconnect opens and their interactions with succeeding gates which are more accurate than the model from [2] have been developed in the past [5, 6, 7]. Some of the effects are highly complex and difficult to integrate into a gate-level simulator. A modeling approach which abstracts from physical open defect behavior can be found in [8]. n-detection [9, 10] and its extensions [11, 12] can also be used to cover defects with non-trivial behavior such as open-via defects. A simulator for interconnect open defects based on

a model which takes intervals of trapped charge into account has been proposed in [13]. That simulator does not report the contribution of oscillation effects to fault coverage but considers detections by both voltage and I_{DDQ} testing.

The remainder of the paper is organized as follows. The model from [2] used in this work is described in Section 2. The extraction of the physical parameters is explained in Section 3. The open-via defect simulator is introduced in Section 4. The experimental results are reported in Section 5. Section 6 concludes the paper.

2 Open-Via Defect Model

An *open-via defect* is assumed to permanently disconnect a portion of an interconnect from its driving gate. The interconnect affected by the defect is called *victim interconnect* and the logic line in the gate-level net-list of the circuit which corresponds to the victim interconnect is called *victim line*. The neighboring interconnects which have non-negligible parasitic cross-capacitance with the victim interconnect are called *aggressor interconnects* and their corresponding lines in the gate-level net-list are called *aggressor lines*. The aggressor interconnects which influence the victim interconnect between the open-via defect and the sinks (driven gates) of the interconnect are called *affecting aggressor interconnects* and their equivalents on the gate level are called *affecting aggressor lines*.

According to the model from [2], an open-via defect separates the victim interconnect into two parts: an affected part and an unaffected part. The unaffected part includes all segments of the victim interconnect preceding the open-via defect. The affected part includes all segments succeeding the defect. If the interconnect has a fanout and the defect is on the fanout stem, all the fanout branches are in the affected part. If the defect is on a fanout branch, the fanout stem and all other fanout branches are in the unaffected part. In a complex interconnect topology, the fanout branch may be the stem of a subsequent fanout, all of which then belong to the affected part.

The affected and the unaffected parts of the interconnect are assumed to be electrically disconnected. The unaffected part assumes the regular logic value imposed by the logic gate which drives the interconnect. In particular, it is not affected by logic values on any aggressor lines. In contrast, the logic value on the affected part is given by the cross-capacitances to the affecting aggressor interconnects and logic values on the corresponding lines. For an affecting aggressor line a, let $CC(a)$ be the parasitic coupling capacity between the interconnect corresponding to line a and the victim interconnect.

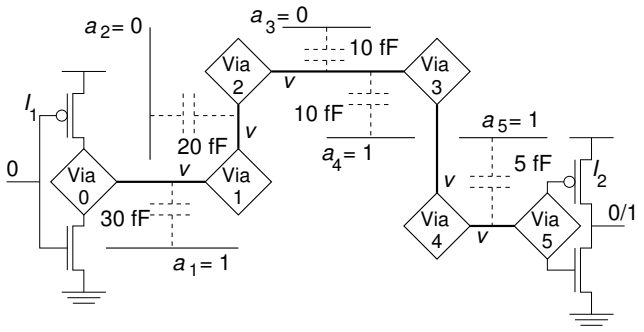

Figure 1: Example for open-via diagnosis

Given an open-via defect and a test vector, let a_1^0 through a_k^0 be all the affecting aggressor lines which assume logic value 0, and let a_1^1 through a_l^1 be all the affecting aggressor lines which assume logic value 1. Affecting aggressor lines $a_1^0, \ldots a_k^0$ impose logic 0 on the victim line with (relative) strength $C_0 := CC(a_1^0) + \cdots + CC(a_k^0)$; affecting aggressor lines $a_1^1, \ldots a_l^1$ impose logic 1 on the victim line with strength $C_1 := CC(a_1^1) + \cdots + CC(a_l^1)$.

Low-resistance via defects, which could have implications on circuit delay, are not targeted by the model. The logic value is assumed to be identical throughout the affected part. Some extensions of the base model, such as consideration of the logic thresholds of the gates driven by the affected interconnect [14], can be easily integrated into the framework.

2.1 Example

Figure 1 shows the application of the diagnostic method from [2] to a fragment of a circuit. Line v (bold in the figure) connects inverters I_1 and I_2. It has six vias (Via 0 through Via 5). Aggressor lines a_1 through a_5 and their parasitic cross-capacitances to line v are shown in the figure. Logic-0 is applied to the driving inverter I_1, so it is driving logic-1 on line v. In the defect-free circuit, logic-0 would be observed on the output of the driven inverter I_2. The logic values on the aggressor lines are logic-1 for a_1, a_4 and a_5, and logic-0 for a_2 and a_3.

If Via 0 is open (missing), the affecting aggressor lines are a_1 through a_5. a_1, a_4 and a_5 impose logic-1 on line v with the cumulative strength $C_1 = 30{+}10{+}5 = 45$ and a_2 and a_3 impose logic-0 on line v with strength $C_0 = 20 + 10 = 30$. Since C_1 exceeds C_0, line v assumes the logic value 1 and inverter I_2 produces a logic-0 on its output.

If Via 1 is open, then the affecting aggressor lines are a_2 through a_5. The value on aggressor line a_1 does not influ-

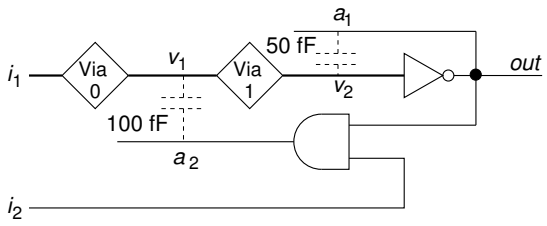

Figure 2: Example for oscillation

ence the logic value on line v seen by inverter I_2, as the part of line v which is cross-coupled with a_1 is be separated by the defect. Consequently, C_1 is calculated as $10 + 5 = 15$ while $C_0 = 30$ remains unchanged. Line v assumes logic-0 value, and I_2 produces logic-1 on its output. If Via 2 is open, the affecting aggressor lines are a_3, a_4 and a_5, $C_1 = 10 + 5 = 15$, $C_0 = 10$, line v assume value 1 and the output of I_2 is logic-0. If Via 3 or Via 4 are open, a_5 is the only affecting aggressor line, C_1 is 5, C_0 is 0, again resulting in logic-0 value at the output.

This information can be used to determine the open-via defect coverage of the test set (which consists of one vector in the example). Only one out of 6 open-via defects (Via 1) resulted in a logic values which contradicted the defect-free circuit behavior, so the fault coverage is $1/6 \approx 16.67\%$. The information can also be used for diagnosis. If the circuit does fail and logic-1 is measured by the test equipment, then the only open-via defect explaining the failure is Via 1. Such resolution is, in general, not achievable by standard fault simulation or diagnosis methods [15]. These methods work on the gate level and have no access to information such as location of vias and aggressor lines or values of parasitic cross-capacitances.

2.2 Oscillation

Consider the circuit in Figure 2. It has two inputs i_1 and i_2 and one output out. The victim line has two vias, and we refer to segments of the victim line as v_1 and v_2. Aggressor lines are a_1 and a_2.

Consider the open-via defect on Via 1. Let the logical value on v_2 be 0. This implies logic-1 on out and hence also on a_1. Since a_1 is the only affecting aggressor line, the logic value 1 is imposed on the victim line v_2. This implies logic-0 on out and a_1, imposing logic-0 on v_2. This process is repeated indefinitely; an oscillation takes place on lines v_2, out and a_1.

Consider the open-via defect on Via 0 under test vector $i_1 i_2 = 11$. If the logical value on v_1 is 1, logic-0 is implied on out, a_1 and a_2 imposing logic-0 on v_1, leading in turn to logic-1 on out and a_2, i.e., oscillation takes place on v_1, out, a_1 and v_2. Consider test vector $i_1 i_2 = 10$. Logic-1 on v_1 implies logic-0 on out, a_1 and a_2, imposing logic-0 on v_1. Logic-0 on v_1 implies logic-1 on out and a_1. However, a_2 remains at logic-0 because the side-input of the AND gate assumes the controlling value. Since the cross-capacitance between the victim line and a_2 is larger than the cross-capacitance between the victim line and a_1, logic-0 stays on the victim line. v_1 and v_2 stabilize at logic-0, out and a_1 stabilize at logic-1, and no oscillation takes place. The defect is detected at the output out because the expected value was logic-0 and the actual value produced by the defective circuit is logic-1. One open-via defect can result in oscillation for one test vector and in a detection for a different test vector.

Oscillation has been reported for interconnect open defects in [4] and validated by SPICE simulations. Oscillation is also well known for feedback bridging faults. See [16] for a study of oscillation behavior as a function of the bridge resistance.

3 Parameter Extraction

The purpose of parameter extraction is to determine the low-level data required for open-via defect simulation, i.e., exact locations of the vias on the interconnects, the locations of the aggressors and the values of parasitic cross-capacitances, and to link these data to the gate-level circuit net-list. Of particular interest are the locations of vias within a fanout structure.

The circuit instrumentation flow is shown in Figure 3. It takes a circuit in Verilog format, its layout in GDS II format and technology parameters, and generates a *via file* in XML (extended markup language) format which contains all the required physical information. The extraction is done using the layout-versus-schematic check (LvS), the design rule check (DRC) and the parameter extraction (PEX) functionality of a commercial parameter extraction tool.

Three Spice files generated by the tool are used in the flow. The main Spice file contains the gate-level net-list in Spice format. This file is employed to extract signal ports, i.e., the signal lines in the circuit, including the information on the inputs and outputs of gates connected by the interconnect in question. The cross-capacitances Spice file includes the values of parasitic cross-capacitances to other lines and the names of these lines. Using the signal port information obtained before, aggressors and the strengths of their influence on the victim line are determined and stored, for each victim line, in the *aggressor file* in XML format. The resistor tree Spice file represents an interconnect by a series

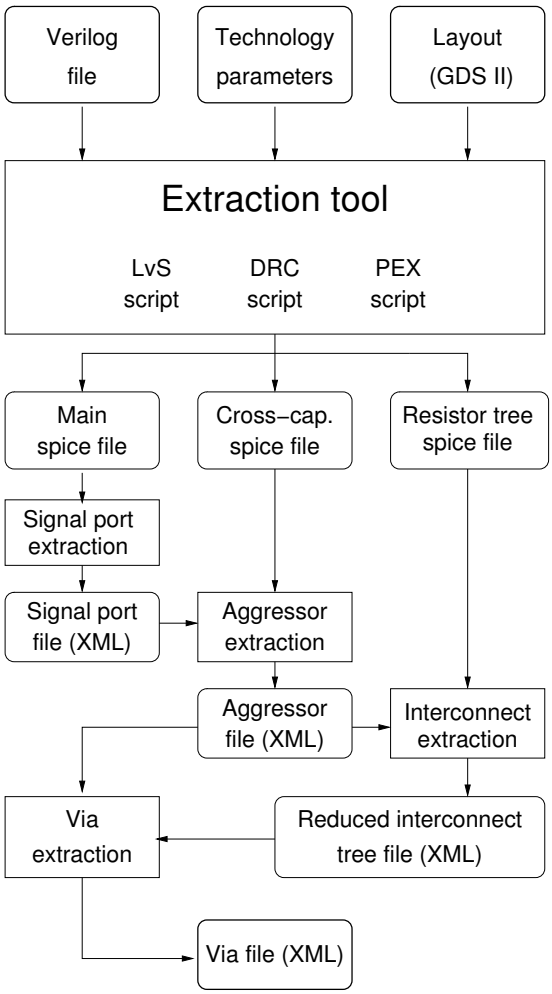

Figure 3: Parameter extraction flow

of line segments characterized by its resistance and capacitance. The coordinates of the endpoint of the segments in the layout are included in this file. The locations of vias on the interconnect can be obtained from the coordinates. The relevant information is filtered and combined with the data from the aggressor file to form the *reduced interconnect tree file* in XML format. Power supply and ground lines are excluded from consideration as victim lines, however they are taken into account as aggressor lines.

The reduced interconnect tree file contains, for each signal, the segments of the resistor tree, enriched by the information which segment represents a via, which segment is connected to an input or output port of a gate, which segment is a fanout, and which segment is subject to parasitic cross-capacitance with an aggressor line. From this file, a *via file* is generated. For each via, the affecting aggressor lines, the strengths of their influence (parasitic cross-capacitances) and the affected gates (outputs of the victim interconnect) are stored in the via file. The open-via defect simulation employs the gate-level net-list and the via file. It is sufficient to retain the via file, other files generated by the parameter extraction flow are not required for open-via defect simulation.

4 Open-Via Defect Simulation

The simulator takes the gate-level net-list of the circuit, the extracted via file, the test set and the fault list as inputs. The fault list contains candidate failing vias given by the id of the interconnect and the id of the via on the interconnect. Both are unique numbers assigned during parameter extraction. The simulator determines, for every open-via defect, all outputs on which the defect is detected and all outputs on which oscillation takes place. The information which outputs are affected by the defect is useful for diagnosis.

The simulator employs the three-valued logic $\{1,\ 0,$ OSC$\}$. For each vector from the test set, the good simulation of the defect-free circuit is performed. No oscillation can show up in a defect-free circuit. Then, an event-driven fault simulation is run for every open-via defect from the fault list. For an open-via defect, the affecting aggressor lines are looked up (this information is contained in the via file generated during parameter extraction), the values C_0 and C_1 are calculated and the logic value on the victim line is determined. If this value is not the fault-free value, it is propagated through the circuit. After the propagation is finished, the logic value on the victim line is re-calculated if the logic value of at least one affecting lines has been modified during propagation. If the logic value on the victim line has changed, oscillation is present. The affected part of the victim line is assigned value OSC and the propagation is repeated.

We distinguished between *defects detected by faulty-value* and *defects detected by oscillation*. If a logic value which has been propagated to an output is the inverted fault-free value, the defect is considered detected by faulty-value and dropped from the fault list. If value OSC has been propagated to an output, the defect is considered detected by oscillation and *not* dropped from the fault list because a different vector from the test set could detect the defect by faulty-value. Note that in a combinational circuit detection by faulty-value on one output and detection by oscillation on a different output is impossible as the only source of values which differ from the good simulation is the affected part of the victim interconnect and its value is uniform according to the model assumptions.

Consider a defect only detected by oscillation but never detected by faulty-value. It depends on the properties of the test equipment whether such a device will indeed be identified as defective. On one hand, there is at least a probability that a faulty-value will be captured by the test equipment. Moreover, oscillation may lead to anomalous behavior of the device, e.g., elevated temperature or I_{DDQ}, which might be detected by the test equipment. We consider both the pessimistic detection assumption (only defects detected by faulty-value are accounted for) and the optimistic detection assumption (both defects detected by faulty-value and oscillation are accounted for).

Let N be the number of all open-via defects in the fault list, let D be the number of all defects detected by faulty-value and let O be the number of all defects detected by oscillation but not detected by faulty-value. The *defect coverage not considering oscillation* is defined as

$$DC = \frac{D}{N}.$$

The *defect coverage considering oscillation* is defined as

$$DC_{\mathrm{OSC}} = \frac{D+O}{N}.$$

If an open-via defect causes oscillation on an output, the test equipment may detects it with a probability. If this probability is known to be p, one can calculate the *defect detection probability* as $(1-p) \cdot DC + p \cdot DC_{\mathrm{OSC}}$.

The open-via defects with no affecting aggressor lines cannot be detected based on the model assumptions. One example for such a defect is Via 5 in Figure 1. It is impossible to impose any value on the victim line; it will be floating. We call such defects *structurally untestable*. Let U be the number of such defects. The *defect efficiency not considering oscillation* is defined as

$$DE = \frac{D}{N-U}.$$

The *defect coverage considering oscillation* is defined as

$$DE_{\mathrm{OSC}} = \frac{D+O}{N-U}.$$

There may be further untestable defects, i.e., defects for which no test vector exists. For instance, a combination of logic values on the aggressor lines might be required for defect detection which contradicts the logical implications in the circuit. It would be possible to identify such untestable defects using standard techniques and to account for them when calculating the number U. We did not use untestable defect identification beyond the structural technique mentioned above in this work.

5 Experimental Results

We applied the simulator for open-via defects to ISCAS 85 circuits. We generated layouts of the circuits by an automatic place & route software using a 0.6 μm technology. The technology is not identical to the 0.6 μm technology used in [4] to demonstrate oscillation due to an interconnect open defect by SPICE simulations.

Table 1 contains the results for 1-detection stuck-at and 3-detection stuck-at test sets. The name of the circuit is followed by the number N of the open-via defects in the fault list and the number U of structurally untestable defects. For both test sets, its size S, the number D of defects detected by faulty-value, the number $D+O$ of defects detected by either faulty-value or oscillation, the four defect coverage metrics introduced above and the run times on a 1.6 GHz SunFire V240 computer with 6 GB RAM are reported. The final row of the table contains the average defect coverages.

The defect coverages calculated without considering structurally untestable defects are rather low. The defect efficiencies are better, yet still relatively far from 100%. A large share of detections is by oscillation only. Somewhat surprisingly, a 3-detection test set does not result in a significant increase of the coverage. Indeed, the coverage even slightly goes down for several circuits. (A 1-detection test set is not necessarily contained in a 3-detection test set.)

6 Conclusions

We have presented a flow for simulation for open-via defects. It is based on an electrical model from the literature extended by considering oscillation. Since the simulation is performed on the gate level, it is applicable to larger blocks. The electrical parameters required for modeling the defect behavior are extracted using commercial tools. Experimental results show that stuck-at test sets do not cover all open-via defects and that the coverage gain due to n-detection is limited. The efficiency of test sets generated using other strategies such as the Unified Fault Model [17] and generating specific test vectors for open-via defects are interesting topics for future research.

Another direction for future work is the development of a more accurate electrical model for the defect behavior and the integration of such a model into the simulator. One focus is handling of resistive vias and their effects on circuit delay. Taking trapped charge into account is a further possible extension [13, 14]. Since the trapped charge is unknown a priori, statistical approaches may be required for its modeling.

Circuit	N	U	1-detection test set								3-detection test set							
			S	D	$D+O$	DC	DC_{OSC}	DE	DE_{OSC}	Time	S	D	$D+O$	DC	DC_{OSC}	DE	DE_{OSC}	Time
c0017	36	21	5	4	13	11.11	36.11	26.67	86.67	0.03	16	4	13	11.11	36.11	26.67	86.67	0.06
c0095	177	5	12	150	169	84.75	95.48	87.21	98.26	1.28	36	164	171	92.66	96.61	95.35	99.42	3.20
c0499	995	59	63	891	928	89.55	93.27	95.19	99.15	29.95	174	882	928	88.64	93.27	94.23	99.15	81.27
c0880	1838	195	64	1488	1624	80.96	88.36	90.57	98.84	46.92	132	1476	1633	80.30	88.85	89.84	99.39	96.54
c1355	2915	488	95	2085	2394	71.53	82.13	85.91	98.64	120.31	267	2057	2399	70.57	82.30	84.75	98.85	335.44
c1908	4040	626	148	2944	3315	72.87	82.05	86.23	97.10	333.77	406	2943	3329	72.85	82.40	86.20	97.51	919.13
c2670	6285	392	109	5163	5656	82.15	89.99	87.61	95.98	589.67	251	5222	5684	83.09	90.44	88.61	96.45	1377.47
c3540	8047	681	166	6528	7062	81.12	87.76	88.62	95.87	845.48	378	6485	7079	80.59	87.97	88.04	96.10	1918.46
c6288	11846	2097	36	8533	9649	72.03	81.45	87.53	98.97	514.18	62	8509	9658	71.83	81.53	87.28	99.07	873.17
c7552	19212	1729	184	15002	16386	78.07	85.29	85.81	93.73	3539.73	444	15105	16441	78.62	85.58	86.40	94.04	8455.97
$\emptyset$						72.42	82.19	82.13	96.32					73.03	82.50	82.74	96.66	

Table 1: Open-via defect simulation results

Acknowledgment

This work has been partially funded by the DFG Graduate School Embedded Microsystems and the DFG project Be 1176/14-1. We are thankful to Dr. Martin Keim of Mentor Graphics Corp., Wilsonville, OR, for help with setting up the extraction flow.

7 References

[1] R. Blish, T. Dellin, S. Huther, M. Johnson, J. Maiz, B. Likins, N. Lycoudes, J. McPherson, Y. Peng, C. Peridier, A. Preussger, G. Propkop, and L. Tullos. *Critical Reliability Challenges for the International Technology Roadmap for Semiconductors (ITRS)*, 2003.

[2] Y. Sato, I. Yamazaki, H. Yamanaka, T. Ikeda, and M. Takakura. A persistent diagnostic technique for unstable defects. In *Int'l Test Conf.*, pages 242–249, 2002.

[3] S.P. Murarka, I.V. Verner, and R.J. Gutmann. *Copper—Fundamental Mechanisms for Microelectronis Applications*. John Wiley & Sons, 2000.

[4] H. Konuk and F. Joel Ferguson. Oscillation and sequential behavior caused by interconnect opens in digital cmos circuits. In *Int'l Test Conf.*, pages 597–606, 1997.

[5] V.H. Champac, R. Rodriguez-Montanes, J .A. Segura, J. Figueras, and J.A. Rubio. Fault modelling of gate oxide short, floating gate and bridging failures in CMOS circuits. In *European Test Conf.*, pages 143–148, 1991.

[6] M. Renovell and G. Cambon. Electrical analysis and modeling of floating-gate fault. *IEEE Trans. on CAD*, 11(11):1450–1458, 1992.

[7] S. Rafiq, A. Ivanov, S. Tabatabaei, and M. Renovell. Testing for floating gates defects in cmos circuits. In *Asian Test Symp.*, pages 228–236, 1998.

[8] S.M. Reddy, I. Pomeranz, H. Tang, S. Kajihara, and K. Kinoshita. On testing of interconnect open defects in combinational logic circuits with stems of large fanout. In *Int'l Test Conf.*, pages 83–89, 2002.

[9] S.C. Ma, P. Franco, and E.J. McCluskey. An experimental chip to evaluate test techniques experimental results. In *Int'l Test Conf.*, pages 663–672, 1995.

[10] S.M. Reddy, I. Pomeranz, and S. Kajihara. Compact test sets for high defect coverage. *IEEE Trans. on CAD*, 16:923–930, 1997.

[11] M.R. Grimaila, S. Lee, J. Dworak, K.M. Butler, B. Stewart, H. Balachandran, B. Houchins, V. Mathur, J. Park, L.-C. Wang, and M. Ray Mercer. REDO-random excitation and deterministic observation–first commercial experiment. In *VLSI Test Symp.*, pages 268–274, 1999.

[12] I. Polian, I. Pomeranz, S. Reddy, and B. Becker. On the use of maximally dominating faults in n-detection test generation. *IEE Proceedings Computers and Digital Techniques*, 151(3):235–244, 5 2004.

[13] H. Konuk. Fault simulation of interconnect opens in digital cmos circuits. In *Int'l Conf. on CAD*, pages 548–554, 1997.

[14] W. Zou, W.T. Cheng, and S.M. Reddy. Interconnect open defect diagnosis with physical information. In *Asian Test Symp.*, pages 203–208, 2006.

[15] N. Jha and S. Gupta. *Testing of Digital Systems*. Cambridge University Press, 2003.

[16] I. Polian, P. Engelke, M. Renovell, and B. Becker. Modeling feedback bridging faults with non-zero resistance. *Jour. of Electronic Testing: Theory and Applications*, 21(1):57–69, 2005.

[17] G. Chen, S. Reddy, I. Pomeranz, J. Rajski, P. Engelke, and B. Becker. An unified fault model and test generation procedure for interconnect opens and bridges. In *European Test Symp.*, pages 22–27, 2005.