Automatic Test Pattern Generation for Interconnect Open Defects*

Stefan Spinner* Ilia Polian* Piet Engelke* Bernd Becker*
Martin Keim** Wu-Tung Cheng**

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

**Mentor Graphics Corp.
8005 SW Boeckman Road
Wilsonville, OR, 97070, USA
martin_keim@mentor.com
wu-tung_cheng@mentor.com

Abstract

We present a fully automated flow to generate test patterns for interconnect open defects. Both inter-layer opens (open-via defects) and arbitrary intra-layer opens can be targeted. An aggressor-victim model used in industry is employed to describe the electrical behavior of the open defect. The flow is implemented using standard commercial tools for parameter extraction (PEX) and test generation (ATPG). A highly optimized branch-and bound algorithm to determine the values to be assigned to the aggressor lines is used to reduce both the ATPG efforts and the number of aborts. The resulting test sets are smaller and achieve a higher defect coverage than stuck-at n-detection test sets, and are robust against process variations.

Keywords: Interconnect opens, Open-via defects, ATPG

1 Introduction

Interconnect opens are a major defect mechanism in nanoscale CMOS [1, 2]. They can be divided into intra-layer opens which disconnect a portion of an interconnect on a particular metalization layer and inter-layer opens (open-via defects) which affect conducting connections between metal layers. Both types of defects can be caused by both random defects (particles) and systematic failure mechanisms. The latter include the effects of optical proximity correction (OPC) measures during lithography [3], deposition issues for vias with high aspect ratios in dual-damascene copper interconnect technology [4], and complex interactions with low-κ interlevel dielectric materials [1]. Hence, adequate coverage of interconnect opens during manufacturing test is essential.

A significant body of knowledge exists on modeling interconnect opens on electrical level [5, 6, 7, 8]. The voltage and hence the logical value on the *victim*, i.e., the floating part of the interconnect disconnected from its driving gate by the open defect, is largely determined by the voltages on the *aggressors*, i.e., its neighboring interconnects. Konuk [9] proposed a fault simulation method based on an accurate electrical model of interconnect opens. Venkataraman and Drummonds [10] performed diagnosis of interconnect opens by tracking the fanout structure. Reddy et al. [11] suggested to model interconnect opens on stems with fanouts by multiple stuck-at faults. The approaches in [10, 11] do not consider any effects induced by aggressors.

Sato et al. [2] presented a flow for diagnosing open-via defects based on an aggressor-victim model. Despite several simplifying assumptions on the electrical behavior, the flow was successfully applied to identify the root cause of a number of hard-to-diagnose defects in Hitachi's integrated circuits and validated by physical failure analysis. Zou et al. [12] used a refined electrical model which takes the logic thresholds of the driven gates and the trapped charge into account. Takahashi et al. [13] employed a simpler model. Rodríguez-Montañés et al. [14] proposed a diagnosis approach based on the relative values of voltages induced by interconnect opens and incorporating I_{DDQ} data. Further literature is available on testing open defects in board interconnects which is not a target of our work [15, 16].

In this paper, we present an automatic test pattern generation (ATPG) flow for interconnect opens. Test patterns can be produced for an arbitrary selection of fault locations which include all intra-layer and inter-layer opens. We use the aggressor-victim open defect model from [2] as it has been demonstrated to yield acceptable results in an industrial setting. For an interconnect open defect, we relate its location to a net segment extracted by a commercial parameter extraction (PEX) tool and determine the aggressor lines based on capacitance data extracted from the circuit's layout.

We perform a local analysis to find an assignment of logical values to the aggressor lines which activates the defect. This assignment is mapped to a constrained stuck-at fault test generation problem and handed to a commercial stuck-at ATPG tool. The assignment is optimized by a branch-and-bound procedure in order to reduce the number of unsuccessful ATPG invocations. This ensures applicability of the flow to large industrial designs. No manual intervention by

*Parts of this work are supported by the German Research Foundation under grant BE 1176/14-1.

the user and no data beyond the circuit's gate-level net-list, layout and parasitic coupling capacitances are required.

We report experimental results obtained targeting open-via defects (inter-layer opens) only as well as targeting all modeled inter- and intra-layer interconnect opens. The algorithm generates compact and high-quality test sets which cover close to 100% detectable defects with very few aborts. The generated patterns outperform n-detection test sets in coverage and test size. Generating patterns on top of a stuck-at test set is a particularly attractive option, yielding the highest coverage by adding only few new patterns. We also validate the stability of the fault model under moderate process variations.

The only published deterministic ATPG for an aggressor-victim model of interconnect opens known to us has been proposed in [17]. The tool in [17] considered only one defect per interconnect (while we target all defect locations on an interconnect). Its test generation strategy results in a coverage of 80% to 93%, whereas our sophisticated branch-and-bound procedure is able to cover over 99% of defects.

The remainder of the paper is organized as follows. In Section 2, the aggressor-victim interconnect open defect model is outlined and the notation is introduced. The ATPG procedure is described in Section 3. The experimental results are reported in Section 4. Section 5 concludes the paper.

2 Aggressor-Victim Model

Our procedure is based on the aggressor-victim model from [2]. The model is applied to arbitrary inter- and intra-layer opens. An interconnect has one *source*, i.e., the logic gate which drives the interconnect, and one or multiple *sinks*, i.e., logic gates driven by the interconnect. An interconnect v affected by an inter-layer or an intra-layer open defect is called *victim*.

The open defect breaks interconnect v in two parts: one connected to the source, called *stable* part and denoted v_S, and one disconnected from the source, called *floating* part and denoted v_F. We assume that only one open defect is present. If v has a non-trivial fanout topology, some of the sinks may be connected to v_S while others may be connected to v_F. The voltage on v_S is driven by the source, and all sinks connected to v_S see the logic value imposed by the source.

The voltage on v_F and thus the logical value seen by the sinks connected to the floating part is determined by the voltages on the *aggressors*, i.e., neighboring interconnects with a non-zero parasitic coupling capacitance to v_F, denoted $CC(a, v_F)$. Note that v_F and thus $CC(a, v_F)$ depends on the exact position of the open defect on the victim interconnect.

Let $C_0(v_F)$ be the sum of all values $CC(a, v_F)$ for aggressors a which assume logical value of 0 under the current input vector. Let $C_1(v_F)$ be the sum of all $CC(a, v_F)$ for aggressors assuming logical value of 1. Power supply and ground lines are allowed as aggressors; their logical values are 0 and 1, respectively, for any input vector. The value seen by all sinks connected to v_F is assumed to be logic-0 if $C_0(v_F) > C_1(v_F)$ and logic-1 otherwise. This implies that the threshold voltage of the sinks is approximated by $V_{DD}/2$.

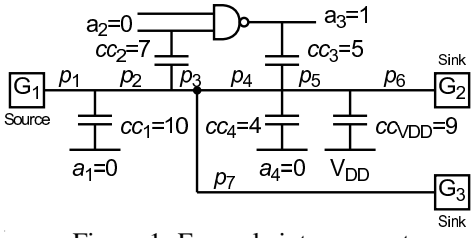

Figure 1: Example interconnect

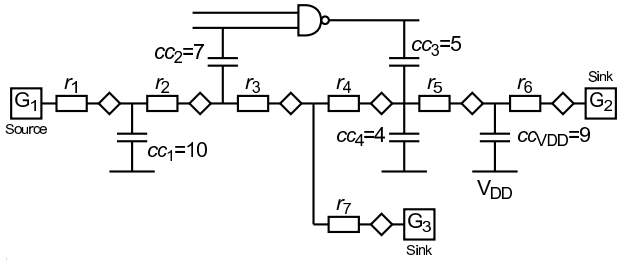

Figure 2: RC tree of interconnect from Figure 1

Parameter extraction (PEX) tools calculate, for a given interconnect, a resistor-capacitance (RC) tree. An interconnect is divided into small portions. Every portion of the interconnect is mapped to a resistor and one or multiple capacitances at its output. We call one such resistor / capacitances block an *RC element*. We target open defects located at the output of an RC element in the RC tree. Consider the portion of the interconnect corresponding to a particular RC element. Due to the aggressor-victim model assumptions, the behavior of the circuit with an open defect on any location within that portion of the interconnect is identical. Hence, a test pattern generated for the open on the output of the RC element in the RC tree detects all opens on the corresponding portion of the interconnect.

The list of defects targeted by the ATPG consists of outputs of RC elements. If all defects are detected, the coverage of the entire interconnect is achieved.

Example: Figure 1 shows a victim interconnect v divided by the PEX software into seven portions p_1 through p_7, with gate G_1 the source driving logic-0 on v, and gates G_2 and G_3 the sinks of v. Figure 2 depicts the corresponding RC tree. Resistors r_i correspond to interconnect portions p_i. Possible open defect locations are indicated by diamonds $\diamond$. The interconnect is influenced by five aggressors (four logic signals a_1 through a_4 and one power supply line V_{DD}); the logical values on the aggressors and the CC values in femtofarad are shown in Figure 1.

If the open defect occurs within portion p_1, i.e., at the output of RC element r_1, v_S consists of portions p_1 and v_F consists of portions p_2 through p_7. Since $C_1(v_F) = cc_3 + cc_{VDD} = 14$ is less than $C_0(v_F) = cc_1 + cc_2 + cc_4 = 21$, gates G_2 and G_3 see logic-0 which corresponds to the defect-free value. Hence, no fault effect could propagate to an output from any of the sinks. If the open is located within

portion p_4, v_S is composed of p_1, p_2, p_3 and p_7, and v_F consists of p_4, p_5 and p_6. The value seen by G_2 is logic-1 because $C_1(v_F) = cc_3 + cc_{VDD} = 14$ is larger than $C_0(v_F) = cc_4 = 4$. Sink G_3 is connected to v_S and thus sees the logic-0 value driven by G_1 unaffected by the defect. Fault effect propagation to an output is possible only from gate G_2.

3 ATPG Procedure

The ATPG algorithm takes a list of open defects as input. The defect list may include intra-layer and inter-layer opens, located at the end of an RC element. As explained above, such a defect represents all open defects on the portion of the interconnect corresponding to that RC element. If all RC elements from the RC tree are targeted, complete coverage of all opens is possible.

The ATPG generates a set of test patterns which detect defects for which a test pattern exists. Other defects are either proven to be untestable or aborted due to high computational complexity.

3.1 Pre-processing

The tool supports pattern generation on top of a pre-defined test set T, e.g., the stuck-at test set. In that case, T is fault-simulated first, and all detected defects are dropped.

Structurally untestable defects, i.e., defects with no aggressors influencing v_F, are considered untestable and dropped. The voltage on the floating part of the interconnect cannot be controlled for these defects within the aggressor-victim fault model, so it is impossible to detect them.

The remaining defects are targeted by deterministic ATPG outlined below.

3.2 Deterministic test generation

Given an open defect on interconnect v, the floating part v_F of v, the aggressors and the coupling capacitances to the aggressors are identified first.

There are two ways to generate a test pattern for the defect: by forcing logic-0 on v_F or by forcing logic-1 on v_F. If the first option is chosen, a pattern is determined which justifies logic-1 on v, sets a sufficient number of aggressors to logic-0 such that $C_0(v_F) > C_1(v_F)$ holds and logic-0 is imposed (forced) on v_F, and propagates the resulting 1/0 difference to an output. This corresponds to finding a test for the stuck-at-0 fault on v under additional constraints, namely the requirement that some of the aggressors are set to logic-0.

If, in contrast, the test pattern generation is done by forcing logic-1 on v_F, logic-0 is justified on v, logic-1 is justified on a subset of aggressors and the resulting 0/1 difference is propagated to an output. This corresponds to generating a test for the stuck-at-1 fault on v such that some aggressors are set to logic-1.

Identification of the subset of aggressors to be used to force the faulty value on v_F is not trivial. Assignments to aggressors may conflict with each other and with assignments necessary to detect the stuck-at fault. For instance, in Figure 1 aggressor a_3 is driven by an NAND gate with aggressor a_2 as an input. Hence, it is impossible to set both a_2 and a_3 to logic-0. If a subset of aggressors has been selected and no test could be found, the defect cannot be considered untestable as it may be detectable using a different subset of aggressors. The number of possible assignments to aggressors is 2^n, where n is the number of aggressors which influence v_F and could exceed 100.

The algorithm attempts to generate a pattern by forcing logic-0 on v_F as well as by forcing logic-1. The procedure is successful if a pattern detecting the defect is generated by one of both attempts. If both problems are proven unsolvable, the defect is considered untestable. If no detecting pattern has been generated and at least one of both calculations was aborted, the defect is considered aborted. In the following, the generation by forcing logic-0 is explained; forcing logic-1 is symmetric.

3.2.1 Forcing logic-0 value

First, the algorithm checks whether logic-0 can be forced on v_F by performing two checks. Defects which fail these checks are provably undetectable by forcing logic-0. The opposite does not hold, i.e., defects which have passed the checks are not guaranteed to be detectable.

The first check evaluates whether some of the aggressors are power supply lines and their cumulative strength, i.e., sum of their coupling capacitances to v_F, exceeds the respective number for all the other aggressors combined. In this case, $C_0(v_F) > C_1(v_F)$ cannot hold and no test pattern detecting the open defect by forcing logic-0 on v_F exists.

The second check determines whether the stuck-at-0 fault is detectable without any additional constraints. The stuck-at-0 fault is handed to a commercial stuck-at ATPG tool. If the tool reports that the fault is redundant, no detection of the open defect reduced to this fault is possible by combinational or one-cycle scan testing.

3.2.2 Aggressor selection

We use a branch-and-bound (B&B) algorithm to find an assignment of a sub-set of n aggressors which is consistent with the goal of simultaneously detecting the v-stuck-at-0 fault. The B&B algorithm generates suggestions for assignments to aggressors which are evaluated by the following three-stage process:

1. Local implication analysis: Logical values implied on other aggressors by the assignment under consideration are calculated. In the example in Figure 1, if logic-0 has been assigned to aggressor a_2, the implication routine would automatically set aggressor a_3 to logic-1. If the assignment is logically inconsistent or the condition $C_0(v_F) > C_1(v_F)$ cannot be achieved even if the unassigned aggressors are all set to logic-0, the assignment is considered unsuccessful for forcing logic-0 and the evaluation terminates.

2. Constrained stuck-at ATPG: The commercial stuck-at ATPG tool is invoked to generate a pattern for the v-stuck-at-0 fault with selected aggressors set to logic-0 as additional

constraints. If no pattern is generated, the assignment is considered unsuccessful and the evaluation terminates. It would have been possible to skip stage 1 and call the stuck-at ATPG tool directly. The local implication routine is used to reduce the number of unsuccessful calls of the ATPG tool.

3. Fault simulation: If the pattern generated in step 2 detects the targeted open defect, the test generation is considered successful. It could fail to detect the open defect because not enough aggressors are set to logic-0 and $C_0(v_F) > C_1(v_F)$ does not hold.

The B&B algorithm to select the assignments to be evaluated using the three-stage process is described next.

3.2.3 Branch-and-bound algorithm

The list of aggressors is sorted in the decreasing order of their CC values to v_F. The aggressors are set to logic-0 in this order. This is done to reduce the number of aggressors which must be selected to force logic-0 on v_F. Moreover, setting an aggressor with a low CC to logic-0 may imply a logic-1 on an aggressor with a high CC, preventing the defect from being detected.

After an aggressor has been assigned, the three-stage evaluation process described above is run. The problem is solved if a pattern which detects the defect has been generated. Then, all other yet-undetected defects are simulated and dropped if detected. If any of the first two stages of the evaluation process (the local implication check or the stuck-at ATPG) were unsuccessful, a *backtrack* is initiated. The last aggressor a_k in the sorted list which is set to logic-0 is set to logic-1, and aggressor a_{k+1} is set to logic-0. By doing so, we prune parts of the solution space which provably do not contain a valid solution. If the pattern has been generated but did not detect the defect, we continue the exploration of the solution space by setting the next unassigned aggressor to logic-0.

To ensure reasonable run times, we put a limit L on the number of different aggressor assignments considered. We set this limit to $2 \cdot n$, where n the number of aggressors, in order to account for the complexity of the solution space. If this limit has not been reached, i.e., the solution space has been exhausted, the defect is considered untestable, otherwise it is considered aborted. If the stuck-at ATPG reports an abort, we discontinue the calculation, consider the defect aborted and do not backtrack. This is done because the stuck-at ATPG instance which would be generated after the backtrack would be very similar to the one which resulted in an abort. Thus, it is likely that the stuck-at ATPG tool still would not be able to calculate a pattern after the backtrack.

3.2.4 Example

Consider test generation for an open defect on portion p_1 of the circuit from Figure 1 by forcing logic-0 on v_F. v_F consists of portions p_2 through p_7. The ordered list of aggressors is a_1, a_2, a_3 and a_4 (V_{DD} is also an aggressor but cannot be assigned). Aggressor a_1 is assigned 0 first, and test generation for v-stuck-at-0 fault with a_1 set to 0 as an additional constraint is invoked. Suppose that a pattern is generated but does not detect the open defect. Aggressor a_2 is assigned 0 and 1 is implied on a_3 next. Suppose that the pattern generated for v-stuck-0 with constraints $a_1 = a_2 = 0$ still does not detect the defect. Since a_3 is already implied, a_4 is assigned 0 next. Assume that no pattern for v-stuck-at-0 fault with $a_1 = a_2 = a_4 = 0$ could be generated.

No stuck-at test generation is invoked for assignment $a_4 = 1$ because $C_0(v_F) = cc_1 + cc_2 = 17$ and $C_1(v_F) = cc_3 + cc_4 + cc_{VDD} = 18$. Even if a pattern would be generated, no forcing of logic-0 on v_F would be possible. Hence, a backtrack to the last aggressor for which a decision was taken, i.e., a_2, is initiated. a_2 is set to 1 and a_3 is set to 0. Test generation for v-stuck-at-0 fault with $a_1 = a_3 = 0$ and $a_2 = 1$ results in a pattern which detects the open defect.

4 Experimental Results

We laid out ISCAS 85 circuits and extracted the parasitic coupling capacitances CC using the flow outlined in [18]. We applied our ATPG to the complete list of both inter-and intra-layer interconnect opens. Since open-via defects are reported to be of particular concern [1], we repeated the measurement targeting only inter-layer opens.

Table 1 summarizes the results for inter-layer opens only. The ATPG procedure was run on top of an existing stuck-at test set. The number of stuck-at patterns and the number of additional patterns generated for open defects not covered by the stuck-at test set are given in columns 2 and 3, respectively. Column 4 contains the total number of targeted defects, i.e., vias on all signal interconnects. The number of structurally untestable defects, for which no value forcing and thus no detection is possible, is reported in column 5. The number of defects for which the tool proved that no pattern detecting this defect exists is given in column 6. The number of aborts is reported in column 7. Column 8 contains the number of detected defects. The defect efficacy, defined as the number of detected defects divided by the number of detectable defects (i.e., the total number minus the numbers in columns 5 and 6), is reported in column 9. The run time (including the time for the stuck-at ATPG tool) are given in the last column.

It can be seen that the number of additional patterns is not high, totalling up to 20% of the stuck-at test set for circuit c2670. The number of aborts is low, resulting in high defect efficacies. All the aborts in this experiment were due to an abort of the stuck-at ATPG tool during the constrained stuck-at test generation. Although an abort could also occur due to the limit put on the number of aggressor assignments considered, we did not observe such aborts in the reported experiments.

Table 2 contains test generation results for inter-layer open defects obtained by the ATPG procedure which did not use an existing test set as a basis and thus targeted the complete defect list. Column 'Pat.' gives the number of generated patterns, column 'Det.' quotes the number of detected defects and column 'DE' reports the defect efficacy (which takes into account the number of undetectable defects not in-

Circuit	Patterns		Defects					Defect	Run time
	Stuck-at	Top-up	Total	Structurally untestable	No pattern	Abort	Detected	efficacy, %	[s]
c0880	64	1	1239	130	39	0	1070	100.00	356
c1355	95	4	1893	331	156	2	1404	99.86	1389
c1908	148	4	2697	392	210	0	2095	100.00	1317
c2670	109	21	4752	257	252	1	4242	99.84	2720
c3540	166	17	5459	454	361	1	4643	99.38	4148
c6288	36	6	7943	1267	296	42	6338	99.06	5479
c7552	184	24	12224	1047	747	2	10428	99.96	14411

Table 1: Experimental results for inter-layer (via) open test generation in top-up pattern mode

Circuit	Inter-layer open ATPG			Stuck-at 1-detection			Stuck-at 2-detection			Stuck-at 3-detection		
	Pat.	Det.	DE, %	Pat.	Det.	DE, %	Pat.	Det.	DE, %	Pat.	Det.	DE, %
c0880	55	1071	100.00	64	1069	99.81	97	1069	99.81	132	1070	99.91
c1355	64	1404	99.86	95	1398	99.43	179	1401	99.64	267	1403	99.79
c1908	106	2095	100.00	148	2087	99.62	272	2093	99.90	406	2094	99.95
c2670	123	4241	99.84	109	4215	99.22	173	4230	99.58	251	4233	99.65
c3540	165	4643	99.38	166	4618	98.84	277	4625	98.99	378	4633	99.17
c6288	39	6339	99.05	36	6323	98.80	46	6329	98.89	62	6332	98.94
c7552	200	10428	99.96	184	10396	99.65	299	10417	99.86	444	10416	99.85

Table 2: Results for inter-layer (via) opens and comparison with stuck-at n-detection test sets

cluded in the table). For comparison, the same values are given for stuck-at n-detection test sets with $n = 1$, 2 and 3.

The ATPG procedure achieves coverage which is almost identical to the version which generates patterns on top of the stuck-at test set (the difference to column 8 of Table 1 is at most 1 defect per circuit). The number of required patterns is smaller than in top-up mode. n-detection test sets do achieve a high coverage of open-via defects but they are still inferior to the patterns generated by the ATPG despite their larger size. This is best observed for larger circuits.

Tables 3 and 4 quote results for all open defects (intra- and inter-layer). The layout of the tables corresponds to that of Tables 1 and 2. Since an open defect corresponds to an RC element and thus to a portion of an interconnect, we calculated the length of each portion to account for the probability of defect occurrence. In addition to values computed based on absolute numbers of defects, the numbers obtained by weighting every open defect with the length of the corresponding portion are quoted in parentheses. These numbers are given in millimeters in Tables 3 and 4.

Although the number of considered opens is much larger than the number of inter-layer opens considered in Tables 1 and 2, the generated test sets do not grow significantly. The conclusions drawn for inter-layer opens are still valid. The efficacy of both test sets generated by the ATPG (with and without a stuck-at test set as a basis) is the highest. The efficacy of n-detection test sets is reasonable but less than that of the generated sets, both when considering the absolute numbers of detected defects and their cumulative length.

Taking the length of the open defects into account yields higher defect efficacy values. This implies that many short portions of interconnect correspond to hard-to-detect open defects while many longer portions of interconnect correspond to easy-to-detect open defects. This is consistent with our observation that the PEX tool extracts many RC elements corresponding to short portions in the vicinity of an interconnect's source or sinks. A smaller number of RC elements correspond to longer portions 'in the middle' of an interconnect which have many aggressors and allow more options to force the desired value on them.

To validate the stability of the generated test sets under moderate process variations, we performed a Monte-Carlo simulation of circuit c2670 assuming varying values of CC. We performed 101 simulations assuming that every parasitic coupling capacitance is an independent random variable according to a Gaussian distribution with mean given by its extracted value and σ chosen such that 3σ corresponds to 15%. Table 5 reports the number of defects detected when no process variations are present (column 'No proc. variations'), the mean number of defects detected throughout 101 simulations, the standard deviation, the minimum and the maximum value and the difference Δ of the maximum and the minimum values.

The coverage of the stuck-at test sets is almost unaffected by process variations. Even though there is some deterioration for the generated open test sets, their mean coverage is still higher than that of the stuck-at test set. We validated that in each of the 101 simulations the generated test sets outperformed the stuck-at test sets. Moreover, the spread between the maximum and the minimum value is as narrow for open test sets as for stuck-at test set, suggesting a high stability.

5 Conclusions

Interconnect open defects are not fully covered by stuck-at test sets. We introduced a flow to generate patterns for both intra- and inter-layer opens based on accurate physical modeling accepted in the industry. The required electrical data are obtained as a by-product during layout-versus-schematic check. The test generation procedure works on gate level, utilizing an optimized branch-and-bound algorithm to for-

Circuit	Patterns		Defects					Defect
	Stuck-at	Top-up	Total	Struct. untestable	No pattern	Abort	Detected	efficacy, %
c0880	64	4	8042 (49.5)	1989 (9.5)	158 (0.5)	0 (0.0)	5895 (39.5)	100.00 (100.00)
c1355	95	8	11895 (60.6)	3492 (11.6)	749 (1.9)	15 (0.0)	7639 (47.1)	99.80 (99.94)
c1908	148	8	16814 (89.8)	4473 (17.5)	987 (2.8)	0 (0.0)	11354 (69.5)	100.00 (100.00)
c2670	109	32	29953 (266.3)	5586 (26.1)	1203 (4.5)	1 (0.0)	23163 (235.7)	99.89 (99.96)
c3540	166	30	33408 (203.9)	7738 (33.7)	1576 (5.0)	2 (0.0)	24092 (165.2)	99.49 (99.83)
c6288	36	11	48149 (263.7)	13851 (44.0)	1430 (4.6)	239 (0.7)	32629 (214.4)	99.04 (99.61)
c7552	184	49	75340 (534.2)	17036 (80.1)	3442 (11.1)	15 (0.2)	54847 (442.9)	99.96 (99.96)

Table 3: Experimental results for all opens in top-up pattern mode; numbers in parentheses indicate cumulative length in mm

Circuit	Inter- and intra- layer open ATPG			Stuck-at 1-detection			Stuck-at 2-detection			Stuck-at 3-detection		
	Pat.	Det.	DE, %	Pat.	Det.	DE, %	Pat.	Det.	DE, %	Pat.	Det.	DE, %
c0880	61	5894 (39.4)	100 (100)	64	5887 (39.4)	99.9 (100)	97	5890 (39.4)	99.9 (99.9)	132	5894 (39.5)	100 (100)
c1355	70	7638 (47.1)	99.8 (99.9)	95	7615 (47.0)	99.5 (99.7)	179	7626 (47.1)	99.6 (99.9)	267	7632 (47.1)	99.7 (99.9)
c1908	123	11354 (69.5)	100 (100)	148	11309 (69.3)	99.6 (99.8)	272	11343 (69.5)	99.9 (100)	406	11339 (69.4)	99.9 (99.9)
c2670	146	23161 (235.7)	99.9 (100)	109	23033 (235.0)	99.3 (99.7)	173	23084 (235.4)	99.6 (99.8)	251	23115 (235.5)	99.7 (99.9)
c3540	183	24094 (165.2)	99.5 (99.8)	166	23975 (164.3)	99.0 (99.3)	277	24000 (164.7)	99.1 (99.5)	378	24059 (165.0)	99.3 (99.7)
c6288	46	32640 (214.4)	99.0 (99.6)	36	32561 (214.0)	98.8 (99.5)	46	32597 (214.2)	98.9 (99.6)	62	32598 (214.1)	98.9 (99.5)
c7552	249	54845 (442.8)	100 (100)	184	54668 (441.6)	99.6 (99.7)	299	54762 (442.2)	99.8 (99.8)	444	54780 (442.3)	99.8 (99.8)

Table 4: Results for all opens and comparison with n-detection test sets; numbers in parentheses show cumulative length in mm

mulate constraints for a commercial stuck-at ATPG tool. The resulting test sets outperform n-detection stuck-at test sets in both coverage and pattern count and are stable under process variations.

One possible direction for future work is incorporating more elaborate defect models into the flow. The design of algorithms which support model accuracy without sacrificing scalability is of interest. Extending the procedure to generate diagnostic patterns which can distinguish between open defects is another highly relevant and open problem as conventional, purely gate-level diagnosis methods cannot provide sufficient resolution to accurately locate defects on long interconnects.

Test set	No proc. variations	Mean	Std. dev.	min	max	Δ
s@ 1-det.	4215	4214.5	2.8	4208	4222	14
s@ 2-det.	4230	4229.6	2.6	4223	4236	13
s@ 3-det.	4233	4232.5	2.9	4225	4240	15
s@ + open	4242	4238.6	2.8	4232	4245	13
open	4241	4239.5	2.8	4233	4248	15

Table 5: Impact of process variations (3σ corresponds to 15%) on number of detected defects of circuit c2670 for different test sets

6 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] B. Koenemann. Test in the era of "What you see is NOT what you get". In *Int'l Test Conf.*, page 12, 2004. (Keynote address).

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

[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] D. Arumí, Rodríguez-Montañés, and J. Figueras. Experimental characterization of CMOS interconnect open defects. *IEEE Trans. on CAD*, 27(1):123–136, 2008.

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

[10] S. Venkataraman and S.B. Drummonds. A technique for logic fault diagnosis of interconnect open defects. In *VLSI Test Symp.*, pages 313–318, 2000.

[11] 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.

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

[13] H. Takahashi, Y. Higami, S. Kadoyama, T. Aikyo, Y. Takamatsu, K. Yamazaki, H. Yotsuyanagi, and M. Hashizume. Clues for modeling and diagnosing open faults with considering adjacent lines. In *Asian Test Symp.*, pages 39–44, 2007.

[14] R. Rodriguez-Montanes, D. Arumi, J. Figueras, S. Einchenberger, C. Hora B. Kruseman, M. Lousberg, and A.K. Majhi. Diagnosis of full open defects in interconnecting lines. In *VLSI Test Symp.*, pages 158–166, 2007.

[15] N. Jarwala and C.W. Yau. A new framework for analyzing test generation and diagnosis algorithms for wiring interconnects. In *Int'l Test Conf.*, pages 63–70, 1989.

[16] J.G. Deshayes and G.D. Robinson. Interconnect testing of boards with partial boundary scan. In *Int'l Test Conf.*, pages 572–581, 1990.

[17] R. Gómez, A. Girón, and V. Champac. Test of interconnection opens considering coupling signals. In *IEEE Defect and Fault Tolerance Symp.*, pages 247–255, 2005.

[18] S. Spinner, J. Jiang, I. Polian, P. Engelke, and B. Becker. Simulating open-via defects. In *Asian Test Symp.*, pages 265–270, 2007.