Boolean and Pseudo-Boolean test generation for feedback bridging faults

Michele Favalli, Marcello Dalpasso

Abstract—Feedback bridging faults may give rise to oscillations within integrated circuits. This work mainly investigates the propagation of oscillations, a behavior that may have a relevant impact on the fault detection. We propose both a logic-level model of the faulty circuit and two techniques aiming to the generation of high-quality test sequences.

Index Terms—test generation, bridging faults, formal proof engines

1 Introduction

Bridging and open faults are the most likely kind of defects in CMOS ICs and this holds true in nano-CMOS ICs [1], too. Fault-simulation and test-generation tools targeting such defects require accurate fault models to describe the complex behavior of faulty circuits.

Both kind of such defects may give rise to significant sequential and analog effects in combinational logic and the uprising of oscillations [2], [3] is one of such complex behaviors. In particular, a feedback bridging fault (FBF) connecting the output line of gate g to the output line of a gate belonging to the transitive fan-out of g may give rise either to oscillations (if an inverting path is sensitized between the two bridged lines [4], [5]) or to latching effects (when a non-inverting path is sensitized [2], [3]). Oscillations and sequential behavior may also arise when the input line of a gate is open and capacitively coupled with the output line of a gate located in its transitive fan-out [2]. Moreover, FBFs have been shown [6] to be a relevant fraction of the total number of bridging faults.

As known [7], [8], a FBF may be tested by disabling its feedback loop. However [3], different values of the faulty bridging resistance may result in: *i)* an undetectable fault; *ii)* a detected fault without oscillations; *iii)* a fault giving rise to oscillations. Therefore, disregarding oscillations may result in test escapes. Moreover, controllability and observability constraints may prevent from actually disabling the feedback loop.

Furthermore, even if the FBF loop is sensitized to produce oscillations that propagate to the output(s) of a module under test, the fault may remain undetected, if the oscillating output signal(s) are sampled while having their fault-free value (see [9] for FPGAs). Conversely, such an escaped fault may still provoke the sampling of wrong values during circuit operations because, with respect to the test phase, different paths are sensitized.

Finally, if the oscillations propagate through multiple reconverging paths, they might disappear and generate right/wrong steady-state values depending on the timing of the oscillating signals that reconverge on the same gate.

As regards FBF detection, previous works [2], [3], [4], [5] accurately model the feedback loop, but pay less attention to the propagation of oscillations. Specific test vectors for oscillating faults may be required, but, at our knowledge, existing automatic test-pattern generators (ATPGs) for FBFs mainly provide tests that disable the feedback loop [8]. Conversely, accurate fault-simulation techniques have been presented to analyze FBF detectability [3].

In the case of opens, the faulty behavior depends on the effect of the transitions of lines (aggressors) capacitively coupled with the floating one (victim line). If some aggressor is influenced by the value of the victim line, oscillations may arise [2]. In this regards, the test generation method proposed in [10] makes use of a satisfiability modulo theory (SMT) solver engine to accurately model the effects of charge partition between the victim and the aggressor lines. This ATPG, may generate test vectors resulting in oscillations, but it does not consider the specific problems related to the propagation of such values.

These reasons motivated the development of ATPG strategies targeting the detection of oscillating FBFs and easily extendable to other faults that shows an oscillating behavior. To this purpose, we exploit Boolean satisfiability-based (SAT) test-generation techniques [11] that have been shown to be capable of handling large circuits [12] for a wide variety of fault models including: 1) stuck-at faults [11], [12], [13]; 2) bridging faults [8], [14]; 3) open faults [10]; 4) delay faults [15], [16].

In addition, we also explore the use of a Pseudo-Boolean (PB) solver/optimizer [17] in the generation of high-quality test sets. The PB optimization problem is more complex than the Boolean satisfiability one, but so far, the feasibility of PB techniques has been shown in [18] where they are used to optimize the detection of small delay defects by sensitizing the longest paths in a circuit.

To account for the excitation and propagation of oscillations in combinational circuits we developed a new algebra that qualitatively characterizes the changes affecting the

- *M. Favalli is with the ENDIF Department of the University of Ferrara, Italy*
- *M. Dalpasso is with the DEI Department of the University of Padua, Italy*

oscillating signals when they reconverge and is exploited to deal with the uncertainties related to the sampled values. In fact, differently from stuck-at like faults and non-feedback bridging faults, the POs to which fault effects propagate do not hold a steady erroneous value. Therefore, at clock edges, the flip-flops may sample an erroneous value or not depending on the characteristics of the oscillating signal. In this regards, our test generator avoids the generation of tests that sensitize multiple reconverging paths resulting in oscillating or steady-state values at module outputs which have a very low probability to be detected.

The proposed ATPG may use two different strategies to increase the fault detection probability. At first, we present a Pseudo-Boolean [19] approach that maximizes the number of module outputs to which fault effects propagate. As an alternative, a two-step approach has been implemented: a SAT-based ATPG attempts to generate a test to propagate the faulty value to a given number of module outputs; on failure, it then tries to generate a test vector propagating the fault effects to at least one output. Results are provided on the performance and effectiveness of the proposed test generation approaches, that is also validated by means of Monte-Carlo event-driven fault simulation.

2 Detection of feedback bridging faults

This section analyzes the detection of FBFs inside a combinational module-under-test, whose primary outputs (POs) are supposed to be sampled by latches or flip-flops in a low-frequency scan-based test environment.

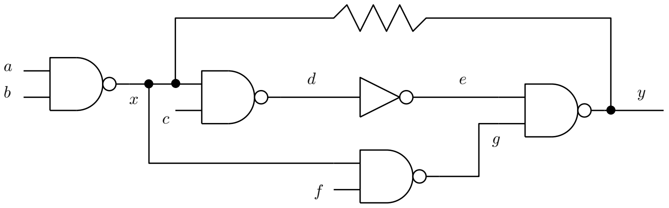

Fig. 1. A resistor representing a feedback bridging fault between lines x and y.

Fig. 1 shows a feedback bridging fault between the lines x and y, with y belonging to the transitive fan-out of x. According to [7], x is the back-line and y is the front-line; furthermore, the so-called back-/front-gate is driving the back-/front-line. In this example, the two bridged lines may be connected through two logical paths, namely $\langle x, d, e, y\rangle$ and $\langle x, g, y\rangle$. If the inverting path $\langle x, d, e, y\rangle$ is sensitized, the FBF may trigger oscillations; conversely, if the non-inverting path $\langle x, g, y\rangle$ is sensitized, a sequential behavior occurs and gives rise to a latching effect. We do not account for faults that involve gate internal nodes [20].

Let us note that the circuit can oscillate even if the inverting path is sensitized only up to the front-gate [2]. The oscillating signal propagates only to the input of the front-gate (where it is blocked by a controlling value on another input), but it makes the output conductance of the front-gate to vary, thus letting the front-gate to alternatively win or lose the conductance conflict with the back-gate. We focus on the oscillations due to FBF closing a logically-sensitized loop, even though the analysis can be easily extended.

As shown in [2], [3], [21], the oscillating behavior induced by a FBF is rather complex, as it depends on the driving strength of the conflicting transistor networks, on the delay of the sensitized path and on the bridging resistance as well. For instance, to trigger oscillations the fault-free front-gate driving strength must be larger than the back-gate one.

A full characterization of such behaviors requires circuit-level simulations to predict the waveforms of the oscillating paths: an unfeasible requirement for test generation in actual circuits, claiming for approximate models [3], [21].

Moreover, it is important to point out that signal analysis within a feedback loop is not enough to decide on the detectability of the related FBF, since the chance to sample erroneous values at POs depends on the starting instant of oscillations, on the timing of paths propagating the oscillations to POs and on the sampling instant of such signals as well.

In both fault activation and propagation, even electrical-level simulation with nominal parameter values would be of partial help, because of the significant variability of circuit parameters that affects current technologies [22], [23].

The importance of the propagation of fault effects giving rise to oscillations has been often neglected in previous works. While the frequency of the oscillations propagated to POs matches the feedback loop one, its duty cycle may be affected by different rise and fall times in gates belonging to the propagation paths and reconvergent propagation paths. The former problem does not significantly change the propagated waveforms (since CMOS gates are typically designed to have symmetric rise/fall times), while the latter one may lead to relevant waveform changes depending on the skew between reconvergent signals that may even make the oscillation disappear.

Furthermore, the skew between an oscillating signal at POs and the clock events triggering its sampling adds uncertainty about the fault detection. Such a skew, in turn, depends on both the starting instant of the oscillations and the delay of the path propagating the fault effects. For instance, when performing low-frequency scan-based testing, the oscillation frequency can be significantly larger than the test rate, in such a way that delay variations are likely to pose a relevant uncertainty on the timing of faulty signals and, therefore, on the sampled values at POs.

As for non-feedback bridging faults, several front- and back-gate input configurations may excite the fault. As an alternative to accurate models [3], conservative approaches can be used (such as the one described in [14] for non-feedback bridging faults), trying to compute a set of test vectors ensuring the detection of a non-feedback bridging fault independently of circuit parameters variations. Such approaches can be extended to FBFs.

As an example, consider again the circuit of Fig. 1: to excite the fault, the value of x in the fault-free circuit should differ from the value of y. This constraint is satisfied by the following configurations of the inputs $abeg$ of the gates driving x and y: $t_0 = 0011$, $t_1 = 0111$, $t_2 = 1011$, $t_3 = 1100$, $t_4 = 1101$ and $t_5 = 1110$. If we impose the sensitization of the path $\langle x, d, e, y\rangle$, g should be 1, thus eliminating the configurations t_3 and t_5. Moreover, in order to produce oscillations, the back-gate must lose the conductance conflict and, therefore, the configuration $ab = 00$ should not be used because it turns ON two parallel transistors in the pull-up of the NAND back-gate, thus maximizing its output conductance. Therefore the remaining configurations are: $t_1 = 0111$, $t_2 = 1011$ and $t_4 = 1101$.

If the two p-channel transistors of the NAND back-gate have the same size, we have two possible cases: a) circuit variations are (locally) small because of matching, so that such transistors have almost the same conductance; b) the variations are large enough to produce different values of conductance for such devices. In case a) we can consider configurations t_1 and t_2 as equivalent and only two tests can be considered.

In such CMOS gates as And-Or-Inverters, featuring more complex transistor networks, the number of input configurations giving rise to oscillations may be larger to account for different conductive paths within the gates.

In the remainder of this work we will focus our attention on oscillating FBFs. In fact, existing ATPG techniques for non-FBF can be used when the logic path between the back- and the front-gate is not sensitized or the values of electrical parameters do not result in oscillations.

3 Test generation for feedback bridging faults

As a first step, SAT-based test-generation techniques [12] build a Boolean model of the fault-free and faulty circuits as a single CNF (conjunctive normal form) formula; then, they use a SAT solver to generate test vectors. As regards the stuck-at fault model, the D-algebra or its extensions are used to characterize signals, but the generation of tests for FBFs requires a more complex model to account for the propagation of oscillations within the faulty circuit and their possible recombination, as well as the logic conditions for fault excitation. The following model refers to a low-frequency scan-based test environment.

3.1 A Boolean model for oscillating signals

The back-line, the front-line and the gates along the sensitized path between such gates typically oscillate with a 0.5 duty cycle. When such oscillating signals propagate within the circuit, they may recombine each other, thus resulting in duty cycle variations. To qualitatively account for this phenomenon, we use an algebra $\mathcal{A} = A \times A^*$, where:

1) $A = \{0, 1\}$ describes the fault-free value of signals;
2) $A^* = \{0^*, 1^*, o, l, h\}$ describes the faulty value of signals:
    - 0^* is a faulty 0 and 1^* is a faulty 1;
    - o is an oscillating signal with a duty cycle (δ) near 0.5;
    - h (l) is an oscillating signal whose δ remains mainly larger (smaller) than 0.5.

This algebra could be extended to handle both X (not-initialized) and Z (high-impedance) states, not considered here to keep the discussion simple.

The generation of h, l and o values in a circuit affected by a FBF is instantiated in Fig. 2: as can be seen, h and l values are generated because of the reconvergency of oscillating signals at different kind of gates. Fig. 3 shows the voltage waveforms of signals represented by h, l and o, as they are computed by circuit-level fault simulation.

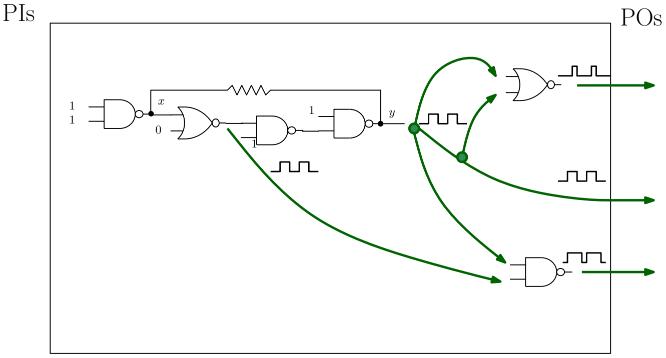

Fig. 2. An example of oscillation propagation due to a FBF.

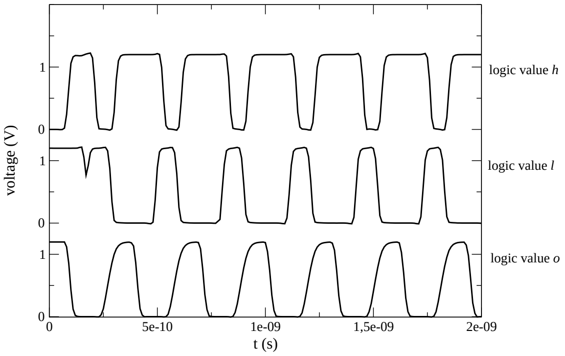

Fig. 3. Different kind of oscillating signals in a circuit affected by a FBF (h, l and o, from top to bottom).

Pairs $(1, 0^*)$ and $(0, 1^*)$ belonging to $A \times A^*$ are equivalent to the widely-used D and $\overline{D}$ values of the D-algebra, respectively. Therefore, the proposed approach can handle faults that do not result in oscillations, too.

As an example of gate operations described by A^*, Tab. 1 shows a NAND gate having a and b as input signals.

To represent the values of A and A^* in a Boolean context, we use a one-hot encoding. Any value $\alpha \in A$ is represented by a Boolean variable $is\alpha \in \{0, 1\}$, while any value $\beta \in A^*$ is represented by a variable $is\beta \in \{0, 1\}$. Using $+$ for the arithmetic sum, the one-hot constraints for signal s are $is0_s + is1_s = 1$ and $is0^*_s + is1^*_s + iso_s + ish_s + isl_s = 1$ in the PB context. As an alternative, a Boolean formula can be easily set to describe each constraint.

a \ b	0^*	1^*	o	h	l
0^*	1^*	1^*	1^*	1^*	1^*
1^*	1^*	0^*	o	l	h
o	1^*	o	h	h	h
h	1^*	l	h	o	h
l	1^*	h	h	h	h

TABLE 1
NAND gate function described by the algebra A^*; a and b are the gate inputs.

This encoding is not minimal, but it can be easily extended to handle different values (such as the unknown one, X) and provides ease of implementation in the PB context, too. Therefore, Tab. 1 and the fault-free function give the set of Boolean formula to be satisfied for a NAND gate, as follows (w is the gate output signal):

$$is0_w \leftrightarrow is1_a \wedge is1_b \tag{1}$$

$$is1_w \leftrightarrow is0_a \vee is0_b \tag{2}$$

$$is0^*_w \leftrightarrow is1^*_a \wedge is1^*_b \tag{3}$$

$$is1^*_w \leftrightarrow is0^*_a \vee is0^*_b \tag{4}$$

$$iso_w \leftrightarrow (is1^*_a \wedge iso_b) \vee (is1^*_b \wedge iso_a) \vee (ish_a \wedge ish_b) \tag{5}$$

$$\begin{aligned} ish_w \leftrightarrow\; & (is1^*_a \wedge isl_b) \vee (isl_a \wedge is1^*_b) \vee (isl_a \wedge isl_b) \vee \\ & (iso_a \wedge iso_b) \vee (isl_a \wedge ish_b) \vee (ish_a \wedge isl_b) \vee \\ & (iso_a \wedge isl_b) \vee (isl_a \wedge iso_b) \vee (iso_a \wedge ish_b) \vee \\ & (ish_a \wedge iso_b) \end{aligned} \tag{6}$$

$$isl_w \leftrightarrow (is1^*a \wedge ish_b) \vee (ish_a \wedge is1^*_b) \tag{7}$$

Each formula can be converted to a CNF by means of algebraic manipulation. For instance, formula (1) becomes $(\neg is0_w \vee is1_a) \wedge (\neg is0_w \vee is1_b) \wedge (is0_w \vee is1_a \vee is1_b)$.

The gates that cannot be involved in the propagation of fault effects are described by A only, whereas the additional constraints $is1 \leftrightarrow is1^*$ and $is0 \leftrightarrow is0^*$ are needed for the signals lying on the boundary between the region possibly affected by fault effects and the fault-free region.

Finally, let Φ be the conjunction of the CNFs of all gates, *i.e.* a CNF describing the whole circuit operations.

3.2 The activation of FBF

Given a FBF, we define an additional variable, $prop_s$, for each signal s belonging to the intersection between the transitive fan-out of its back-line x and the transitive fan-in of its front-line y. Such variables are used to handle the path sensitization from the back-line of the given FBF to s.

For a NAND gate with inputs a and b and output w, the propagation conditions are given by:

$$prop_w \leftrightarrow (is1^*_a \wedge prop_b) \vee (prop_a \wedge is1^*_b) \vee (prop_a \wedge prop_b) \tag{8}$$

Similar formulae can be written for each gate belonging to the considered region and can then be translated to CNFs and added to Φ.

Let us now consider again the circuit of Fig.1, where an oscillating behavior may be excited only if the bridged lines x and y have different fault-free values and the SAT solver succeeds justifying a true value for $prop_y$. Such conditions are translated to the following formulae:

- $prop_y \leftrightarrow 1$
- $iso_x \leftrightarrow iso_y$
- $(\neg(is1_x \leftrightarrow is1_y) \wedge prop_y) \leftrightarrow iso_x$

In this model, the conductance conflict between the front- and the back-gate is supposed to always result in oscillations if such gates are connected by a sensitized inverting path. Of course, this assumption may not hold, but it is conservative because of the relevant variations of circuit parameters in nano-CMOS ICs [24]. These variations, in fact, make the outcome of conductance conflicts as unpredictable (see [14] for a discussion in the case of non-FBFs), thus requiring to account also for oscillations.[1]

Conversely, if accurate results are required, the electrical-level details of the gates involved in the feedback loop should be accounted for. In this paper, however, we do not address this problem because the careful analysis shown in [3] could be performed and translated to a set of additional Boolean constraints.

An input vector that satisfies the excitation conditions for a FBF may sensitize more than one path between the back-line and the front-line, thus making the electrical behavior even more complex. To avoid this kind of problems, additional constraints can be used to force the sensitization of a single path from the back-line to the front-line.

3.3 The detection of oscillating FBF

The variability of circuit parameters makes the detection of an oscillating FBF a random phenomenon whose accurate characterization would be computationally unfeasible for test generation. Therefore, we propose two test-generation techniques that heuristically increase the probability of fault detection.

Consistently with our model, three different conditions are in order for the fault effects of a FBF that propagate to a PO:

1) the fault-free value is 0 (1) and the faulty value is $l\,(h)$: here defined as a *weak* detection, because the probability to detect the fault is expected to be much lower than 0.5;
2) the faulty value is o: a *potential* detection, whose probability is about 0.5;
3) the fault-free value is 0 (1) and the faulty value is $h\,(l)$: a *strong* detection, whose probability is expected to be larger than 0.5.

For a PO, j, we need one more Boolean variable for each one of the above-mentioned classes of detection probability (it can be easily shown that only one of these variables can be true):

1) (*weak*): $w_j \leftrightarrow (is0_j \wedge isl_j) \vee (is1_j \wedge ish_j)$
2) (*potential*): $p_j \leftrightarrow iso_j$
3) (*strong*): $s_j \leftrightarrow (is0_j \wedge ish_j) \vee (is1_j \wedge isl_j)$

The necessary condition for the fault detection is that at least one PO has an oscillation (let's say PO is the set of POs):

$$\bigvee_{\forall j \in PO} (s_j \vee p_j \vee w_j) = 1\ . \tag{9}$$

As verified by circuit-level simulation (and shown in the following), this constraint does not ensure an acceptable test quality. Therefore, we use Pseudo-Boolean optimization to increase the probability of fault detection (Pr_{det}), estimated by means of simplifying hypotheses. At first we suppose that the probabilities of sampling errors at different POs are independent each other, so that:

$$Pr_{det} = 1 - \prod_{\forall j \in PO} (1 - Pr_j)\ , \tag{10}$$

[1] Of course test vectors should be generated for the opposite hypothesis, but in this case any bridging fault test generator can be used.

where Pr_j is the probability that the memory element fed by the j-th PO samples an erroneous value. Such a probability can be approximated as:

$$Pr_j = \pi_s s_j + \pi_o p_j + \pi_w w_j \ ,$$

where π_s, π_p and π_w are the probabilities to sample an erroneous value at a PO conditioned to the presence of a strong, potential or weak oscillation, respectively (let us remind that only one of the variables s_j, p_j and w_j may be 1). These probabilities are both path- and pattern-dependent, but are here assumed to be the same for all POs.

By applying a first-order approximation to Pr_{det}, we get the following linear cost function to be maximized by the Pseudo-Boolean SAT solver:

$$\sum_{\forall j \in PO} (\pi_s s_j + \pi_p p_j + \pi_w w_j)$$

As shown in the following, maximizing such a function may lead to the propagation of fault effects to several POs, thus possibly resulting in several oscillating signals within the circuit. During testing, this may lead, in turn, to a large power consumption in the faulty IC, that does not matter if the faulty ICs are simply rejected when failing the test. If, on the contrary, the module under test belongs to a reconfigurable IC, power constraints may hold also for ICs containing one or more faulty modules: the maximization can be performed under an additional constraint over the maximum number of oscillating signals.

To trade-off test quality for computational efficiency, we propose a Boolean approach as an alternative to the Pseudo-Boolean one discussed so far: a two-step procedure that begins by attempting to generate a test that satisfies some threshold criterion on the approximate value of Pr_{det} and, on failure, a second attempt targets the generation of a test with no constraint (*i.e.*, oscillations should propagate to at least one PO).

In the first step, the Pseudo-Boolean constraint:

$$\sum_{\forall j \in PO} (\pi_s s_j + \pi_p p_j + \pi_w w_j) > \psi,$$

where ψ is a threshold value for Pr_{det}, is translated to a Boolean one. If the first test-generation attempt fails, we simply use Eq. 9.

These two approaches have been implemented using Minisat [25] as SAT solver and Minisat+ as PB SAT solver/optimizer [17].

4 Results

In this section we analyze the test quality achieved by the proposed methods and their computational complexity. As discussed in [3], the detection of a FBF depends on circuit-level details. Therefore, as a first reference in test-quality evaluation we use circuit-level simulation, along with a Monte Carlo approach to account for the influence of circuit parameter variations on the fault detection.

Let us consider a very simple circuit from the *mcnc* combinational benchmark set [26]: **cm82a**, with 5 PIs, 3 POs and 26 cells, mapped on the OSU FreePDK 45 nm cell library [27], using a normal distribution with 5.0% standard deviation for any relevant circuit parameter.

The circuit features $F = 111$ FBFs that may potentially result in oscillations and we generated a test set (T_{opt}) for all of them. For each fault f, T_{opt} contains two test vectors, setting the fault-free value of the back-line to 0 and 1, respectively. This because the circuit is mapped on NAND and NOR gates and, as shown in Sect. 2, two tests are sufficient.

As a reference, we generated a second test set (T_{unopt}) lacking any optimization regarding the propagation of fault effects. With the same model for the faulty circuit, we simply impose that at least one PO may oscillate independently of its faulty value (Eq. 9) without maximizing the number of potentially oscillating POs. Again, two test vectors are generated, featuring a different value for the back-line.

The bridging resistance of faults has been set to 100Ω and, for each fault f, Monte Carlo circuit-level simulations of the two generated test sets have been performed. Now, let det^f_{opt} and det^f_{unopt} be the fraction of Monte Carlo instances that lead to the detection of f when T_{opt} and T_{unopt} are applied, respectively. Estimates of the expected value of the achieved fault coverage (C_{opt} and C_{unopt}, respectively) are given by:

$$C_{opt} = \frac{1}{F} \sum_{\forall f} det^f_{opt}$$

$$C_{unopt} = \frac{1}{F} \sum_{\forall f} det^f_{unopt}$$

The values achieved by considering 1000 Monte Carlo samples are $C_{opt} = 0.764$ and $C_{unopt} = 0.490$. As expected, the optimized test sequence provides a higher fault coverage than the unoptimized one.

In addition, only one fault escaped T_{opt}, while a significant number of escapes affects T_{unopt}, because either some tests propagate the fault effects to a single PO where the fault is not detected because the oscillation is sampled while presenting the fault-free value, or faulty oscillations do not reach the sensitized PO because reconvergencies directly eliminate them or make pulses small enough to be deleted by inertial effects. These results show that the propagation of the effects of FBFs deserves a special attention even in a very simple circuit.

The computational cost and the test quality of the proposed approach are assessed here for larger combinational circuits. In particular, we consider circuits from the ITC99 [28] and ISCAS85 [29] benchmark sets. Tab. 2 shows the features of such circuits and the total number (F) of potentially oscillating FBFs targeted by test generation. We do not use layout-extracted faults [30], [31] because we are mainly interested in analyzing the computational costs of the proposed approach. For the smallest benchmarks we consider all the possible pair of lines featuring at least one inverting path from the back-gate to the front-gate, while for the largest ones we randomly chose 10,000 faults featuring such a characteristic. Of course, for a specific implementation of the benchmarks, more realistic data can be obtained by using layout extracted bridging faults [31]. Conversely, the size of the used sample is large enough to allow for the exploration of different kinds of FBFs where the back and front gate are at different levels, thus allowing to compare the proposed approaches.

Both proposed methods have been applied for the test generation; in the two-step test generation we used $\pi_s = 0.75$, $\pi_p = 0.5$, $\pi_w = 0.25$, and the threshold $\psi = 1.5$. The test generation is repeated for each fault because we have not implemented fault simulation, even though accurate approaches exist [3].

Benchmark	PIs	POs	gates	F
b01	7	7	51	162
b02	5	5	26	53
b03	35	34	178	1658
b04	77	74	743	10000
b05	35	70	315	10000
b06	11	15	59	198
b07	50	57	564	10000
b08	30	25	193	1830
b09	29	29	172	2123
b10	28	23	236	2103
b11	38	37	743	10000
b12	126	127	1550	10000
b13	63	63	230	1471
b14	277	299	10010	10000
b15	485	519	14983	10000
b17	1451	1511	42511	10000
b20	521	512	15275	10000
b21	521	512	15417	10000
b22	734	725	29872	10000
C432	36	7	152	6719
C499	41	32	320	10000
C880	60	26	311	5586
C1355	41	32	320	10000
C1908	33	25	463	10000
C2670	133	239	761	10000
C3540	50	22	996	10000
C5315	178	123	1605	10000
C6288	32	32	1643	10000
C7552	207	108	3510	10000

TABLE 2
Features and number of FBFs (capped to 10,000 randomly selected) for each considered benchmark circuit.

Tab. 3 shows the test-generation results for the PB-based optimization technique and for the Boolean procedure as well. The fraction of faults for which a test has been found will be referred to as C_{max} because it represents the maximum possible value of fault coverage under the hypothesis of an oscillating behavior. This figure catches only the logic level characteristics of the fault, but it does not consider timing effects such as: a) unfavorable signal recombinations; b) inertial effects; c) the sampling instants of POs. Therefore, the actual values of fault coverage in a specific circuit instance are typically smaller than C_{max}.

For the Pseudo-Boolean test generation, column 2 shows the achieved value of C_{max}, while columns 3 to 5 report the average number of POs (per fault) featuring strong/potential/weak detection, respectively (s/p/w in the table). Finally, column 6 shows the average CPU-time per fault elapsed on a P8600 Intel Core Duo @ 2.4GHz. Columns 7 to 10 report the same data for the Boolean test-generation procedure. Column 11 states the number of faults whose first-step test generation failed, thus requiring a second test-generation step featuring the only constraint regarding the propagation of oscillations to at least one PO.

As expected, the two-step Boolean test generation is more efficient than the Pseudo-Boolean one, that is affected by poor scaling, thus being not applicable to the largest benchmark circuits. Conversely, the test quality of the sequences generated by the Boolean approach seems rather high because fault effects can be propagated to several POs and the second test-generation step is not required. The computational cost of this latter approach is of the same order of that of simple test generation with no quality constraint.

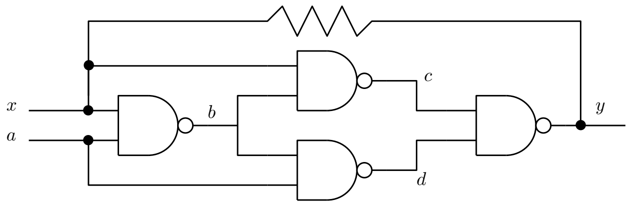

Fig. 4. Example of faulty circuit configuration where inverting and non inverting paths cannot be separately sensitized.

We analyzed a few circuits (C499 and C1355) featuring a low value of C_{max}. These circuits contain several FBFs which are undetectable because the sensitization of an inverting path from the back- to the front-gate necessarily implies the undesired sensitization of a non-inverting path. This is the case of the NAND implementation of a XOR gate in Fig. 4. In order to sensitize the inverting path $\langle x, b, c, y \rangle$, a must be set to logic 1, but also the non-inverting path $\langle x, c, y \rangle$ is sensitized. In these cases, we assume that no oscillation is produced. Note that the fault which cannot be detected under the hypothesis of oscillations may still be detected under different hypothesis.

To analyze the relationship of the test quality indicator C_{max} with the expected fault coverage estimated by accounting for circuits' timing , we implemented a logic-level event-driven fault simulator that:

1) implements a very simple fault model where a very-low-resistance feedback bridging is assumed and the oscillations are generated at the logic level on the basis of gate delays in the faulty circuit;
2) allows for Monte Carlo simulations with both the inertial and transport delay models.

Such a simulator is used also to compare the Boolean and Pseudo-Boolean techniques.

For each logic-level benchmark (in this case, we used the ISCAS85 benchmark set because of delays availability) and test set generated by the proposed methods, the detection of FBFs and the actual fault coverage (for the considered fault model) are random variables, as functions of the actual circuit timing and of the sampling instant. We estimate these quantities by considering:

1) $N = 100$ circuit samples generated accordingly to a uniform distribution of propagation delays;
2) a uniform distribution of sampling instant in a large time interval elapsing from the latest stabilization time of the POS of the faulty circuit (this is consistent with the hypothesis of low-frequency test).

In particular, for each benchmark, once a test has been generated for a FBF f, the fault is injected and simulated in each sample circuit (ξ). Then a detection probability δ_{ξ}^{f} is

bench	C_{max} (%)	Pseudo-Boolean				Boolean				
		s	p	w	CPU	s	p	w	CPU	2nd step
b01	98.14	0.07	2.28	0.04	0.003	0.02	1.72	0.75	0.003	3
b02	92.45	0.00	1.66	0.00	0.002	0.00	1.49	0.11	0.002	4
b03	71.71	0.00	7.60	0.16	0.022	0.00	3.71	1.43	0.012	469
b04	84.73	0.35	8.68	2.02	0.111	0.00	0.09	5.12	0.046	1526
b05	75.45	1.37	5.55	4.27	0.170	0.38	4.48	1.99	0.054	2499
b06	87.87	0.04	2.43	0.21	0.004	0.030	2.26	0.17	0.004	33
b07	73.95	1.12	6.74	0.66	0.141	0.17	3.81	0.57	0.038	2806
b08	79.72	0.46	2.61	0.26	0.012	0.04	2.07	0.15	0.010	375
b09	79.94	0.01	8.64	0.03	0.018	0.003	7.24	0.20	0.012	428
b10	84.78	0.41	3.20	0.02	0.020	0.07	2.48	0.07	0.017	370
b11	88.17	0.19	6.47	0.29	0.053	0.04	4.18	0.33	0.049	1192
b12	80.42	1.07	7.69	1.28	0.154	0.36	4.67	1.12	0.088	1375
b13	97.30	10.63	4.26	10.63	0.026	0.00	1.08	0.01	0.011	39
b14	90.50	0.60	34.79	1.08	40.16	0.19	14.27	3.51	1.66	950
b15	60.15					0.04	23.94	6.13	0.615	797
b17	99.80					0.12	15.60	5.95	3.203	6
b20	98.27					0.00	8.05	0.00	1.053	266
b21	98.27					0.00	8.01	0.00	1.084	256
b22	97.00					0.18	5.30	0.29	1.44	70
C432	71.10	1.40	2.54	0.30	0.035	0.44	2.00	1.26	0.021	1941
C499	63.72	0.17	0.90	0.45	0.028	0.06	0.48	0.34	0.022	3629
C880	93.16	3.30	3.14	0.38	0.098	0.57	3.11	0.17	0.032	777
C1355	33.15	0.08	0.24	0.24	0.027	0.02	0.25	0.20	0.018	6685
C1908	76.30	0.48	3.00	0.35	0.055	0.26	3.03	0.22	0.040	3177
C2670	88.10	1.37	6.62	0.40	0.096	0.09	2.80	0.20	0.064	1190
C3540	73.50	0.70	3.54	0.15	0.105	0.11	2.82	0.02	0.063	4679
C5315	94.40	2.02	25.88	0.87	0.442	0.23	6.63	0.53	0.111	103
C6288	95.29	0.23	21.92	2.87	10.54	0.086	6.67	1.04	0.17	471
C7552	82.99	0.47	2.44	0.036	2.63	0.06	1.55	0.21	0.063	7005

TABLE 3

Results for the test generation when using Pseudo-Boolean optimization (columns 3 to 6) and the two-step procedure based on Boolean satisfiability (columns 7 to 11). CPU times are shown in seconds.

measured accordingly to 2). At the end of the test generation process, for each circuit sample ξ, we compute C_ξ that is a random variable representing the fraction of logically detectable faults (whose number is denoted as D) that are actually detected:

$$C_\xi = \frac{1}{F'} \sum_{\forall detectable\ f} \delta_c^f$$

where F' represents the number of (logically) detectable faults.

Finally, let $\mathcal{C}$ and σ_{C_ξ} be the average value and standard deviation of C_ξ over all the samples considered in the Monte Carlo experiment, respectively.

This experiment has been repeated for the Pseudo-Boolean method, the Boolean method and unoptimized test generation (where we simply require that at least one PO may oscillate). The results (average values and standard deviations of $\mathcal{C}$) are shown in Tab. 4.

As expected, the values of $\mathcal{C}$ with the transport delay model are higher than those computed with the more realistic inertial delay model. The difference between the average values (for all benchmarks) of $\mathcal{C}$ obtained by the Pseudo-Boolean and Boolean approaches is 0.075 in the inertial case and 0.040 in the transport case showing that the Pseudo-Boolean test generation is less sensitive to inertial effects. The average values of $\mathcal{C}$ obtained by unoptimized test generation are much lower, with a difference of 0.196 (inertial delay) and 0.170 (transport delay) from the values obtained with the PB method.

The results show that a test satisfying the logic conditions for the detection of a FBF does not ensure fault detection. The Monte-Carlo experiment provides an accurate estimate of fault coverage, but it presents relevant computational overheads. As an alternative, we tried to use Eq. 10 to provide a simple probabilistic estimate of fault coverage that does not account neither for inertial effects, nor for POs correlations using $\pi_s = 0.75$, $\pi_p = 0.5$, $\pi_w = 0.25$. To analyze the error related to such approximation, Fig. 5a and Fig. 5b plot the probabilistic coverage results as a function of the Monte Carlo ones in the inertial and transport case. The test sequence is the Pseudo-Boolean one, the results are slightly worse than those achieved in the Boolean case.

As expected, the error is rather large in the inertial case, while in the transport case the probabilistic estimate is more accurate (5.6% instead of 11.1% in average). The development of an ATPG accounting for inertial phenomena in an efficient way remains for further research.

Finally, we used this fault simulation procedure to analyze the relevance of the used signal algebra. In particular, we have computed the average detection probability for each kind of output detection (strong/potential/weak) produced by our approach. In particular, let $\varepsilon_{s/p/w}$ be the average probability to detect a fault at a PO marked as strong, potential or weak by the test generator. These values are shown in Tab. 5 in the inertial and transport delay cases for both Pseudo-Boolean and Boolean test generation. As can be seen, it is very important to mark weak detections that give rise to relevant degradations in detection probability, while in the case of strong detections the advantages over a potential detection are small because of multiple reconvergencies.

bench	inertial						transport					
	Pseudo-Boolean		Boolean		unoptimized		Pseudo-Boolean		Boolean		unoptimized	
	$\mathcal{C}$	σ_{C_ξ}	$\mathcal{C}$	σ_{C_ξ}	$\mathcal{C}$	σ_{C_ξ}	$\mathcal{C}$	σ_{C_ξ}	$\mathcal{C}$	σ_{C_ξ}	$\mathcal{C}$	σ_{C_ξ}
C432	0.886	0.035	0.876	0.036	0.665	0.066	0.938	0.026	0.934	0.026	0.729	0.055
C499	0.564	0.053	0.433	0.051	0.416	0.049	0.593	0.050	0.493	0.048	0.468	0.049
C880	0.771	0.037	0.695	0.038	0.446	0.036	0.797	0.033	0.719	0.035	0.468	0.026
C1355	0.532	0.050	0.469	0.049	0.466	0.057	0.542	0.047	0.476	0.047	0.504	0.049
C1908	0.911	0.015	0.883	0.019	0.740	0.018	0.926	0.014	0.897	0.016	0.754	0.014
C2670	0.808	0.050	0.578	0.058	0.548	0.053	0.879	0.037	0.833	0.041	0.596	0.037
C3540	0.777	0.045	0.741	0.046	0.589	0.045	0.846	0.039	0.807	0.043	0.667	0.046
C5315	0.811	0.033	0.685	0.050	0.541	0.037	0.785	0.031	0.772	0.038	0.599	0.025
C6288	1.000	0.041	1.000	0.049	0.906	0.061	1.000	0.037	1.000	0.041	0.973	0.047
C7552	0.941	0.031	0.899	0.038	0.764	0.043	0.991	0.022	0.952	0.028	0.829	0.036
average	0.800	0.038	0.725	0.043	0.604	0.046	0.828	0.033	0.788	0.0364	0.658	0.038

TABLE 4
Average results of MonteCarlo event-driven FBF fault simulation.

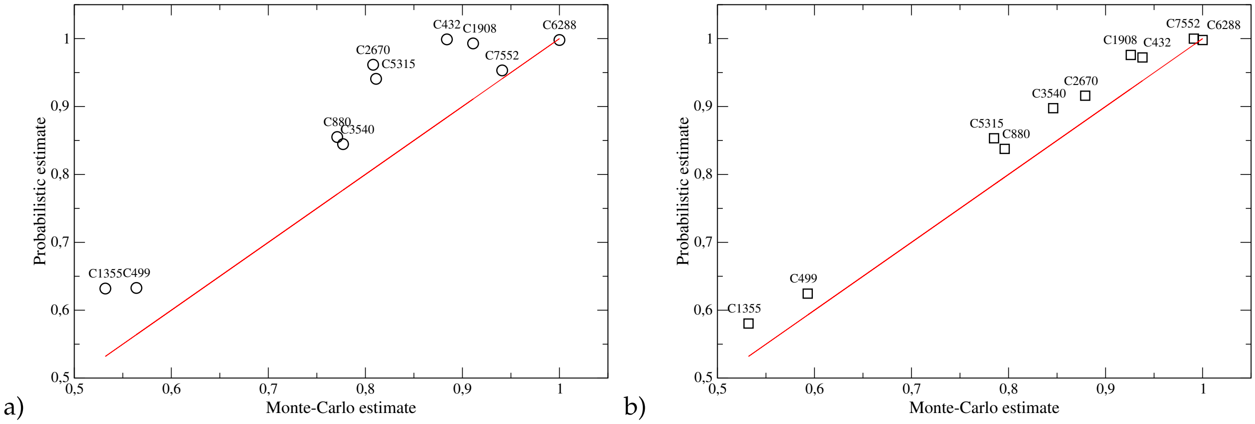

Fig. 5. Probabilistic estimate of $\mathcal{C}$ as a function of the values of $\mathcal{C}$ estimated by Monte Carlo simulations in the inertial a) and the transport b) cases for PB test sequences. The line represents the absence of error.

bench	inertial						transport					
	Pseudo-Boolean			Boolean			Pseudo-Boolean			Boolean		
	ε_s	ε_p	ε_w	ε_s	ε_p	ε_w	ε_s	ε_p	ε_w	ε_s	ε_p	ε_w
C432	0.52	0.43	0.20	0.51	0.43	0.27	0.55	0.46	0.22	0.54	0.46	0.29
C499	0.59	0.49	0.17	0.65	0.48	0.15	0.68	0.49	0.19	0.71	0.48	0.17
C880	0.52	0.47	0.20	0.57	0.47	0.21	0.55	0.49	0.20	0.59	0.49	0.23
C1355	0.64	0.49	0.21	0.65	0.53	0.20	0.68	0.40	0.21	0.68	0.53	0.20
C1908	0.55	0.49	0.27	0.48	0.47	0.30	0.58	0.45	0.30	0.52	0.49	0.31
C2670	0.50	0.46	0.17	0.33	0.46	0.18	0.55	0.50	0.18	0.40	0.50	0.21
C3540	0.53	0.45	0.05	0.46	0.47	0.08	0.56	0.49	0.09	0.50	0.50	0.12
C5315	0.70	0.50	0.055	0.69	0.46	0.02	0.62	0.50	0.12	0.62	0.50	0.10
C6288	0.54	0.49	0.07	0.51	0.49	0.03	0.51	0.48	0.04	0.53	0.49	0.04
C7552	0.57	0.45	0.20	0.57	0.45	0.12	0.62	0.51	0.23	0.62	0.49	0.14
average	0.56	0.47	0.15	0.54	0.47	0.15	0.59	0.47	0.17	0.57	0.49	0.18

TABLE 5
Average probability to sample a logic error at a PO where a strong, normal or weak detction occurs.

4.1 Improving fault coverage

In some benchmarks $\mathcal{C}$ is rather low, and this occurs also in the transport case, due to faults which are observable to only one POs and, seldom, to correlation at POs that voids the advantages of the propagation to several POs.

Here we briefly discuss two possible ways to increase fault coverage: 1) the use of multiple tests for a single fault; 2) the use of multiple sampling instants for the same test.

The first approach was developed to use stuck-at based test sequences to detect other kind of defects [32]. In our case, several solutions may exist for the Pseudo-Boolean and the Boolean problem differing in: a) the path sensitized between the back and the front line; b) the path(s) from PIs to the fault location; c) the path(s) propagating the oscillations to POs. An exhaustive test set for a FBF should sensitize each possible (single or multiple) path from a PI to a PO that includes at least one signal in the feedback loop under the constraints leading to oscillations. Of course, the number of paths may be very large and heuristic techniques generating a small fraction of these tests can be used. This approach, however, may be not able to improve the detection probabilities provided by our method because either the optimized ATPGs sensitize a large fraction of the possible paths, or all the paths related to a FBF have similar delays.

In case 2), instead, only one test vector per FBF is used,

bench	Pseudo-Boolean		Boolean		unoptimized	
	$\mathcal{C}(1)$	$\mathcal{C}(2)$	$\mathcal{C}(1)$	$\mathcal{C}(2)$	$\mathcal{C}(1)$	$\mathcal{C}(2)$
C432	0.886	0.970	0.860	0.936	0.665	0.755
C499	0.564	0.747	0.433	0.614	0.416	0.591
C880	0.771	0.908	0.695	0.850	0.446	0.650
C1355	0.532	0.714	0.469	0.657	0.466	0.655
C1908	0.911	0.982	0.883	0.972	0.740	0.867
C2670	0.808	0.901	0.578	0.675	0.548	0.718
C3540	0.777	0.853	0.741	0.837	0.589	0.741
C5315	0.811	0.961	0.685	0.888	0.541	0.805
C6288	1.000	1.000	1.000	1.000	0.906	0.997
C7552	0.941	1.000	0.899	0.978	0.764	0.881
average	0.800	0.901	0.725	0.840	0.604	0.766

TABLE 6
Expected coverage results of MonteCarlo event-driven FBFs simulation when Pseudo-Boolean, Boolean and unoptimized tests are used with one ($\mathcal{C}(1)$) or two ($\mathcal{C}(2)$) sampling instants.

but POs are sampled more than once. Here, we consider a very simple case where the test vector is first applied and then the POs are sampled twice with the same test rate. The Monte Carlo simulation results for the Pseudo-Boolean, the Boolean and the unoptimized cases are reported in Tab. 6 for the inertial delay case as compared to the coverage value achieved by using a single sampling. As can be seen, this technique increase fault coverage in a relevant way. However, by increasing the number of samplings above 2, the correlation between subsequent samplings decreases the expected benefits of this approach.

5 Conclusions

We addressed the problem of the test generation for feedback bridging faults resulting in oscillations. The work is motivated by simulation results showing that test generation procedures not accounting for the propagation and sampling of oscillating signals produce low-quality test sequences. To account for this problem we developed a logic model of the faulty circuit and two test-generation methodologies raising the probability to detect such faults. The first method exploits a Pseudo-Boolean solver/optimizer to maximize the fault detection probability but it shows scaling problems, while the second one uses Boolean satisfiability and it is able to efficiently deal with larger circuits. Improving the efficiency of PB test generation and developing a method that allows to set test parameters on the basis of actual defect probabilities and test quality requirements remain for further research.

References

[1] R. Aitken, “Defect oriented testing,” in *Advances in electronic testing* (D. Gizoupoulos, ed.), pp. 1 – 42, Springer, 2006.

[2] H. Konuk and F. Ferguson, “Oscillation and sequential behavior caused by interconnect opens in digital CMOS circuits,” in *Proc. of IEEE Int. Test Conf.*, pp. 597 – 606, 1997.

[3] I. Polian and et al., “Modeling feedback bridging faults with non-zero resistance,” *Jou. of Electronic Testing Theory and Applic.*, vol. 21, pp. 57 – 69, 2005.

[4] B. Koch and K. Muller-Glaser, “An examination of feedback bridging faults in digital CMOS circuits,” in *Proc. of IEEE Int. Symp. on Circuit And Systems*, pp. 1527–1530, 1993.

[5] P. Dahlgreen, “Switch-level bridging fault simulation in the presence of feedback,” in *Proc. of IEEE Int. Test Conf.*, pp. 363–371, 1998.

[6] M. Roca and A. Rubio, “Current testability analysis of feedback bridging faults in CMOS circuits,” *IEEE Transaction on CAD*, vol. 14, no. 10, pp. 1299–1305, 1995.

[7] M. Abramovici and P. R. Menon, “A Practical Approach to Fault Simulation and Test Generation for Bridging Faults,” *IEEE Transaction on Computers*, vol. C-34, pp. 658 – 663, Aug 1985.

[8] B. Chess and T. Larrabee, “Logic testing of bridging faults in CMOS integrated circuits,” *IEEE Transaction on Computers*, vol. vol.47, no. no.3, pp. 338–345, 1998.

[9] N. Das, P. Roy, and H. Rahaman, “On-line testing of single feedback bridging faults in cluster based FPGA by using asynchronous element,” in *On-Line Testing Symposium*, pp. 190 – 191, 2008.

[10] D. Erb, K. Scheibler, M. Sauer, and B. Becker, “Efficient SMT-based ATPG for interconnect open defects,” in *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pp. 1–6, March 2014.

[11] T. Larrabee, “Test pattern generation using Boolean satisfiability,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 11, no. 1, pp. 4–15, 1992.

[12] R. Drechsler and et al., *Test generation using Boolean proof engines*. Springer Verlag, 2009.

[13] P. Stephan, R. Brayton, and A. Sangiovanni-Vincentelli, “Combinational test generation using satisfiability,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 15, pp. 1167–1176, Sep 1996.

[14] M. Favalli and M. Dalpasso, “How many test vectors we need to detect a bridging fault?,” *J. Electronic Testing*, vol. 25, no. 1, pp. 79–95, 2009.

[15] J. Kim, J. Whittemore, J. Marques-Silva, and K. Sakallah, “On applying incremental satisfiability to delay fault testing,” in *Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings*, pp. 380–384, 2000.

[16] K. Chandrasekar and M. Hsiao, “Integration of learning techniques into incremental satisfiability for efficient path-delay fault test generation,” in *Design, Automation and Test in Europe, 2005. Proceedings*, pp. 1002–1007 Vol. 2, March 2005.

[17] N. Een and N. Sorenssön, “Translating pseudo-Boolean constraints into Sat,” in *JSAT*, 2006.

[18] S. Eggersglüß, M. Yilmaz, and K. Chakrabarty, “Robust timing-aware test generation using pseudo-boolean optimization,” in *Test Symposium (ATS), 2012 IEEE 21st Asian*, pp. 290–295, Nov 2012.

[19] A. Biere, M. Heule, H. V. Mareen, and T. Walsh, eds., *Handbook of satisfiability*. IOS press, 2009.

[20] Y. Miura and S. Seno, “Behavior analysis of internal feedback bridging faults in CMOS circuits,” *Jou. of Electronic Testing*, vol. 18, no. 2, pp. 109–120, 2002.

[21] M. Hashizume, H. Yotsuyanagi, and T. Tamesada, “Identification of feedback bridging faults with oscillation,” in *Test Symposium, 1999. (ATS ’99) Proceedings. Eighth Asian*, pp. 25–30, 1999.

[22] D. Sylvester and K. Keutzer, “Rethinking deep-submicron circuit design,” *IEEE Computer*, pp. 25 – 34, november 1999.

[23] S. Borkar and et al., “Parameter variation and impact on circuits and microarchitecture,” in *Proc. of Design Automation Conf.*, pp. 338 – 342, 2003.

[24] M. Orshanky, S. Nassif, and D. Boning, *Design for manufacturability and Statistical Design: a cosntructive approach*. Springer, 2008.

[25] N. Een and N. Sorenssön, “An estensible SAT solver,” in *SAT*, 2003.

[26] S. Yang, “Logic synthesis and optimization benchmarks,” tech. rep., Microelectronic Research Center North Carolina: http://www.cbl.ncsu.edu:16080/benchmarks/LGSynth89/, Dec. 1988.

[27] J. Stine and et al., “FreePDK: An Open-Source Variation-Aware Design Kit,” in *Microelectronic System Education*, pp. 173 – 174, 2007.

[28] F. Corno, M. S. Reorda, and G. Squillero, “RT-level ITC 99 benchmarks and first ATPG results,” *IEEE Design & Test of Computers*, pp. 44–53, July-Agost 2000.

[29] F. Brglez and H. Fujiwara, “A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran,” in *Proc. of IEEE Int. Symp. on Circuit And Systems*, pp. 663–698, 1985.

[30] S. Zachariah and S. Chakravarty, “Algorithm to extract two-node bridges,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 11, pp. 741–744, Aug 2003.

[31] J. Shen, W. Maly, and F. Ferguson, “Inductive Fault Analysis of MOS Integrated Circuits,” *IEEE Design & Test*, pp. 33 – 26, Dec. 1985.

[32] S.C. Ma, P. Franco and E.J. McCluskey, “An Experimental Test Chip to Evaluate Test Techniques Experimental Results,” in *Proc. of International Test Conference*, Oct. 1995, pp. 663 – 672.

Michele Favalli Michele Favalli received the PhD degree in Electronic and Computer Science in 1993 from the University of Bologna. From 1993 to 1998 Michele Favalli was a research associate of computer science and engineering in the University of Bologna. From 1998 Michele Favalli is associate professor of computer science and engineering in the University of Ferrara. His scientific interests are in the field of testing and reliability of digital systems and include: fault modelling, fault simulation, test generation, design for testability, on-line testing and fault tolerant design.

Marcello Dalpasso Marcello Dalpasso received the B.S. and M.S. degree in electronic engineering from the University of Bologna, Italy; in 1994 he got the Ph.D. degree in electronic engineering and computer science. Since 2003, he has been an Associate Professor at the University of Padova, Italy, where he teaches computer science. His research interests include the areas of fault modeling and simulation for digital IC design, as well as integer linear programming and operational research.