

A Simulator of Small-Delay Faults Caused by Resistive-Open Defects

Alejandro Czutro*

Nicolas Houarche**

Piet Engelke*

Ilija Polian*

Mariane Comte**

Michel Renovell**

Bernd Becker*

*Albert-Ludwigs-University
Georges-Köhler-Allee 51

79110 Freiburg im Breisgau, Germany

**LIRMM – UMII
161 Rue Ada

34392 Montpellier, France

Abstract

We present a simulator which determines the coverage of small-delay faults, i.e., delay faults with a size below one clock cycle, caused by resistive-open defects. These defects are likely to escape detection by stuck-at or transition fault patterns. For the first time, we couple the calculation of the critical size of a small-delay fault with the computation of the resistance range of the corresponding resistive-open defect for which this size is exceeded. By doing so, we are able to extend probabilistic fault coverage metrics initially developed for static resistive bridging faults to small-delay defects.

Keywords: Small-delay defects, resistive opens, probabilistic fault coverage, bridging fault simulation.

1 Introduction

Small-delay faults are not adequately covered by stuck-at and transition test sets [1]. Low-resistance interconnect open defects are a major source of small-delay faults [2, 3]. Hence, an accurate assessment of a test set’s coverage of small-delay faults should take into account the physical parameters of the open defects causing the potential defects. State-of-the-art small-delay fault simulation approaches determine for every considered fault site the sizes of the fault for which the fault is covered by a test set [4, 5] (earlier methods used less sophisticated concepts [6]). These approaches do not consider physical parameters of defects corresponding to small-delay faults. On the other hand, interconnect-open simulators concentrate on full-open defects which are detectable by stuck-at and transition fault testing [7, 8, 9].

In this paper, we present a small-delay fault simulation approach which calculates realistic coverage of such faults based on the occurrence probability of the low-resistance interconnect open defects. We concentrate on faults with size less than one clock cycle because faults with larger sizes are targeted by stuck-at and transition fault test sets. We first calculate, in a manner similar to [4, 5], the fault sizes which are covered by the given test set. Then, we compute the range of resistances of interconnect open defects which would lead to delay faults of size determined in the first step. Finally, we obtain the realistic fault coverage as the probability that the

resistance of an actual open defect indeed falls in that range of resistances.

The remainder of the paper is structured as follows. A detailed overview of the method is given in Section 2. Fault simulation in the timing domain is explained in detail in Section 3. The description is enhanced compared with [5] and provides a comprehensive coverage of all stages of the simulation process. The mapping between the size of a delay fault and the resistance of the corresponding interconnect open defect is outlined in Section 4. Experimental results are reported in Section 5. Section 6 concludes the paper.

2 Overview of the Method

The inputs of the method are a circuit (a gate-level net-list accompanied by timing and physical information such as gate delays, clock cycle, transistor and topological parameters), a set of test pairs and a list of faults. Faults are specified by a line (i.e., a gate output) on which a transition is slowed down and the direction of the transition. In case of fanouts, we currently consider faults located at the fanout stem. The exact amount δ of the slowdown (the size of the fault) is not specified in the fault list—indeed, the ranges of δ for which a given fault is detected are an (intermediate) result of the simulation. Furthermore, the probability distribution density ρ of interconnect open defect resistance obtained from manufacturing data is required. Refer to [2] for details on estimating ρ and a typical distribution measured at Philips. Based on this information, the simulator calculates a realistic coverage of delay defects with size less than one clock cycle.

Figure 1 shows the flow of the simulator in graph form. First, a line-delay fault simulation is performed for each test pair and each fault. The implemented algorithm is a slight extension of that in [5]. For every fault f_i and every test pair tp_j , the *detection interval (in time domain)* $D^t(f_i, tp_j)$ is calculated. The detection interval contains all the values of $\delta(f_i)$, the size of fault f_i , for which the circuit will fail under test pair tp_j , i.e., a transition at one or more outputs will be delayed beyond the clock cycle time.

Detection intervals are often of shape $[\delta^{\min}, \infty]$ for some value δ^{\min} . Such a delay range may be transformed into a range of resistances of an open defect $[R_{op}^{\min}, \infty]$. The

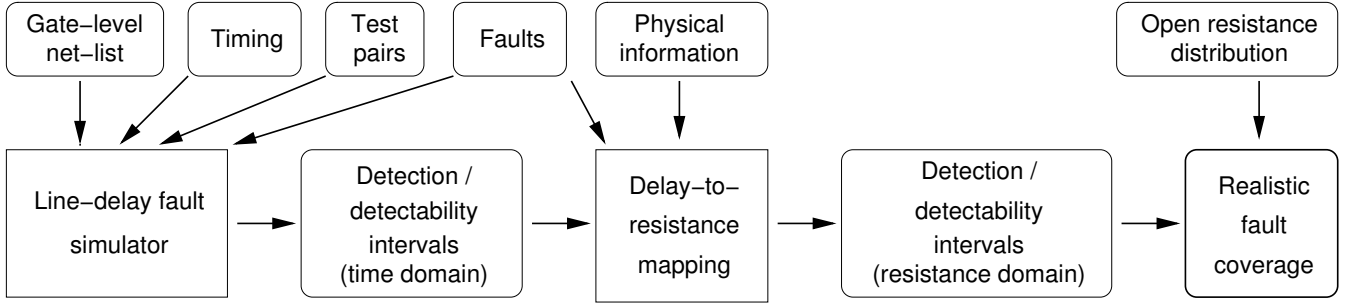


Figure 1: Flow of the method

mapping will be described later. Detection intervals can also consist of disjoint sub-intervals. If no path from the fault site to an output is sensitized, this interval becomes empty. The detection interval for the whole test pair set is obtained as the union of the individual detection intervals: $D^t(f_i) := D^t(f_i, tp_1) \cup \dots \cup D^t(f_i, tp_m)$, where m is the number of test pairs in the set.

In addition, the *global detectability interval (in time domain)* $G^t(f_i)$ is defined as the range of all sizes $\delta(f_i)$ for which the fault leads to a failure under at least one test pair (not necessarily contained in the test pair set). Thus, $D^t(f_i)$ is the range of $\delta(f_i)$ which has been covered by the test set while $G^t(f_i)$ is the range of $\delta(f_i)$ which would be detected by the test pair set of maximal quality, e.g., the exhaustive test pair set. The concept of detection interval and global detectability interval is similar to the notion of C -ADI and G -ADI known for resistive bridging faults [10, 11] and to the term DIS used in [5]. In order to focus on small-delay faults, we intersect both $D^t(f_i)$ and $G^t(f_i)$ with interval $[0, T_C]$ where T_C is the duration of the clock cycle. Without this restriction of the intervals, the major contribution to the fault coverage (defined below) would come from delay faults of size larger than T_C which are covered by stuck-at and transition fault test sets.

After the detection and detectability intervals in timing domain have been calculated for a fault f_i , the ranges of open defect resistance corresponding to these intervals are computed using electrical analysis. The resulting intervals are called *detection interval in the resistance domain* (denoted $D^r(f_i)$) and *global detectability interval in the resistance domain* (denoted $G^r(f_i)$), respectively. For this transformation, a mapping between the resistance R_{op} of an open defect on an interconnect on which fault f_i is located and the size δ of f_i is required.

Finally, the realistic fault coverage is defined, for a fault f_i , by

$$FC(f_i) = 100\% \cdot \int_{D^r(f_i)} \rho(r) dr / \int_{G^r(f_i)} \rho(r) dr. \quad (1)$$

This metric corresponds to the probability that a detectable small-delay fault (i.e., fault f_i with a size included in $G^t(f_i)$) and consequently the resistance of the corresponding open

defect included in $G^r(f_i)$) has actually been detected by the test set (i.e., the size of the fault is included in $D^t(f_i)$ and the resistance of the corresponding open defect is included in $D^r(f_i)$). For more than one fault, average is calculated.

The used definition of fault coverage differs from the metric used in [5] in two important points. First, no transformation into the resistance domain is done in [5]. In contrast, we assume a distribution of open defect resistances which can be determined from manufacturing data. We also account for the fact that an open defect of a given resistance may lead to delay fault of different size depending on the electrical parameters of the gates involved. Second, the method in [5] determines all faults for which the complete detection interval or 90% of the interval is covered by the test set and defines coverage as the percentage of faults for which this is the case. This is done to pinpoint the need for testing the same fault using multiple vectors. Our coverage definition reports the probability of detecting an open defect leading to a small-delay fault.

3 Fault Simulation in Timing Domain

The aim of the fault simulation is to determine the detection interval in timing domain $D^t(f_i)$, which will subsequently be translated into the resistance domain and used to calculate the realistic fault coverage. The simulation works as follows. First, the waveforms in the fault-free circuit are computed. Then, the circuit with an injected delay fault (of unspecified size indicated by symbol δ) is simulated using the concept of *signal descriptors*, which are a generalization of waveforms enriched by effects of a delay fault. The detection interval is yielded from a comparison of the fault-free values and the signal descriptors on the outputs of the circuit.

For a test pair (p_1, p_2) , we assume that p_1 is applied to the circuit's inputs at time $-\infty$ and all signals in the circuit are stable by time 0. p_2 is applied (i.e., all inputs change simultaneously from p_1 to p_2) at time 0 and no further transitions occur on the inputs of the circuit until time $+\infty$. For each signal s , we denote the (stable) value it assumes under p_1 as $IV(s)$ (for 'initial value') and the value to which it stabilizes under p_2 as $FV(s)$ (for 'final value'). The earli-

est arrival time $EAT(s)$ is defined as the first point in time (larger than 0) at which the value on s may change. The latest stabilization time $LST(s)$ is defined as the last point in time at which a transition (to $FV(s)$) is possible. In absence of faults, the value $IV(s)$ is present at s (at least) during the time interval $[-\infty, EAT(s)]$ and the value $FV(s)$ is present during the interval $[LST(s), \infty]$. The values assumed between $EAT(s)$ and $LST(s)$ must be determined by fault-free simulation.

3.1 Fault-free timing simulation

A *waveform* for signal s under test pair (v, w) describes which logical value (0, 1 or X) signal s assumes at which time. We formally define a waveform as a set of tuples $\{(t_1, l_1), (t_2, l_2), \dots, (t_m, l_m)\}$, where t_i 's refer to points in time and an l_i indicates which logical value s assumes starting with time point t_i . For instance, waveform $\{(-\infty, 1), (2, X), (5, 1), (7, 0)\}$ means that the signal is initially set to 1, from time 2 to time 5 the value on the signal is unknown, at time 5 the signal assumes value 1 and at time 7 it transitions to value 0 which is the stable final value.

To perform fault-free timing simulation, the following operations are defined for waveforms: 0-intersection, 1-intersection, inversion and translation. 0-intersecting a number of waveforms results in a new waveform which assumes value 0 for all time intervals for which 0 is assumed on at least one original waveform. For time intervals for which all original waveforms assume 1, the resulting waveform also assumes 1. For all other time intervals, the 0-intersection waveform assumes X.

1-intersection is defined similar to 0-intersection, however the resulting waveform assumes 1 for time intervals for which at least one original waveform assumes 1, it assumes 0 for time intervals for which all original waveforms assume 0 and X in other cases. Inversion of a waveform results in a waveform having value 0 with the original waveform had value 1 and vice versa. The translation of a waveform is defined with respect to two delay numbers rd (rising delay) and fd (falling delay). All transitions to 1 are delayed (their t_i is incremented) by rd , and all transitions to 0 are delayed by fd .

The circuit is simulated in topological order. For an input i , the waveform is given by $\{(-\infty, IV(i)), (0, FV(i))\}$. When considering a gate c with rising delay $RD(c)$ and falling delay $FD(c)$, the following operations are performed: in case of an AND or a NAND gate, the waveforms of all gate inputs are 0-intersected. In case of an OR or a NOR gate, they are 1-intersected. If c is an inverting gate, the resulting waveform is inverted. Finally, the waveform is translated with $rd = RD(c)$ and $fd = FD(c)$. To account for gate inertia effects, pulses at gate inputs which are shorter than the gate's delay are 'filtered', i.e., they are replaced by X values during the translation. This is continued until waveforms have been calculated for all the outputs of the circuit.

3.2 Timing simulation of faulty circuit

A signal descriptor $SD(s, f_i)$ of a signal s in the circuit with a present small-delay fault f_i is a set of *description intervals* of the shape $val@[l, r]_{(\delta_{\min}, \delta_{\max})}$, where val is the value (0, 1 or X) which signal s assumes between time points l and r , provided that the following additional constraint for the size δ of f_i holds: $\delta_{\min} \leq \delta \leq \delta_{\max}$. (Note that only the lower boundary δ_{\min} is used in [5]). The values l and r may be constant numbers, e.g., $l = 10$, or depend on the fault size δ , e.g. $l = 10 + \delta$. Waveforms introduced in the previous section can be seen as a special case of signal descriptors with no dependency of interval boundaries on δ and no constraints.

Signal descriptors on signals not affected by fault f_i are derived from the fault-free waveforms. For such a signal s , the signal descriptor $SD(s, f_i)$ contains only one description interval $FV(s)@[LST(s), +\infty]$. Possible transitions at s before $LST(s)$ are ignored because they cannot affect the value on a circuit's outputs at clock cycle time T_C . Similarly, if s is the output of a gate with at least one input s' which is not affected by f_i , s is not the fault site and the value $FV(s')$ is controlling, then $SD(s, f_i)$ is set to $FV(s')@[LST(s) + D(FV(s')), +\infty]$ if the gate is not inverting and $\neg FV(s')@[LST(s) + D(FV(s')), +\infty]$ if the gate is inverting. Here, $D(v)$ equals the rising delay of the gate if $v = 1$ and the falling delay if $v = 0$. If there is more than one such input s' , the minimal value of LST is taken.

The signal descriptor at the fault site is obtained from the fault-free waveform by delaying the rising or falling transitions, i.e., replacing a boundary x of an interval by $x + \delta$. If only the left bound is delayed, a constraint is generated to make sure that the left boundary never exceeds the right boundary of an interval.

A signal descriptor on the output s of a gate c with a single input s' is calculated by applying to each description interval I in $SD(s', f_i)$ the propagation rules summarized in Table 1 to obtain a description interval J . $SD(s, f_i)$ is composed of all such description intervals J . To account for the gate inertia effect, rules 1 and 2 can only be applied for intervals (pulses) which are wide enough ('Existence condition' in Table 1). Furthermore, the constraints may be *tightened* for rules 3 and 5 to prevent too small pulses from propagating.

If s is the output of a multiple-input gate c , the *equivalent input signal descriptor* is computed first. For this purpose, *cv-intervals* are calculated as intervals from the signal descriptors of gate c 's inputs for which at least one of the inputs assumes the controlling value of gate c . The *ncv-intervals* are computed as intervals $[l, r]$ such that gate c 's non-controlling value is assumed between l and r for all inputs of c . The equivalent input signal descriptor is the collection of the cv-intervals and ncv-intervals with their corresponding logical value. The constraint of a cv-interval is simply taken from the description interval to which it corresponds. The constraint of an ncv-interval is the intersection of the constraints of the corresponding description intervals.

Rule	Input description interval I	Existence condition	Output description interval J
1	$v@[x, y]_{\langle \delta_{\min}, \delta_{\max} \rangle}$	$x + D(v') < y$	$v'@[x + D(v'), y + D(-v')]_{\langle \delta_{\min}, \delta_{\max} \rangle}$
2	$v@[x + \delta, y + \delta]_{\langle \delta_{\min}, \delta_{\max} \rangle}$	$x + D(v') < y$	$v'@[x + D(v') + \delta, y + D(-v') + \delta]_{\langle \delta_{\min}, \delta_{\max} \rangle}$
3	$v@[x + \delta, y]_{\langle \delta_{\min}, \delta_{\max} \rangle}$	–	$v'@[x + D(v') + \delta, y + D(-v')]_{\langle \delta_{\min}, \min\{\delta_{\max}, y - x - D(v')\} \rangle}$
4	$v@[x + \delta, +\infty]_{\langle \delta_{\min}, \delta_{\max} \rangle}$	–	$v'@[x + D(v') + \delta, +\infty]_{\langle \delta_{\min}, \delta_{\max} \rangle}$
5	$v@[x, y + \delta]_{\langle \delta_{\min}, \delta_{\max} \rangle}$	–	$v'@[x + D(v'), y + D(-v') + \delta]_{\langle \max\{\delta_{\min}, x + D(-v') - y\}, \delta_{\max} \rangle}$
6	$v@[-\infty, y + \delta]_{\langle \delta_{\min}, \delta_{\max} \rangle}$	–	$v'@[-\infty, y + D(-v') + \delta]_{\langle \delta_{\min}, \delta_{\max} \rangle}$

Table 1: Signal description propagation rules for a gate c . v' denotes v if c is non-inverting and $\neg v$ if c is inverting.

The signal descriptor $SD(s, f_i)$ is computed by applying the rules from Table 1 to the equivalent input signal descriptor.

Figure 2 shows signal descriptors in a circuit with slow-to-rise fault on the output of gate 2 under test pair 010/111. (The constraints $\langle -\infty, +\infty \rangle$ are skipped for clarity and ‘ $\delta \geq 2$ ’ is written instead of $\langle 2, +\infty \rangle$.) The fault-affected lines are indicated by a circle and the delays of every gate c are shown as $RD(c)/FD(c)$. Figure 3 gives the waveforms on the signals of the circuit. For instance, consider the signal descriptor calculation on gate 6 (inverting, controlling value 1). There is one cv-interval $1@[3 + \delta, +\infty]_{\langle -\infty, +\infty \rangle}$ and one ncv-interval $0@[2, 3 + \delta]_{\langle -\infty, +\infty \rangle}$ (obtained as intersection of $[2, +\infty]$ and $[-\infty, 3 + \delta]$). Propagating them using rules 4 and 5 of Table 1, respectively, results in $0@[8 + \delta, +\infty]_{\langle -\infty, +\infty \rangle}$ and $1@[5, 8 + \delta]_{\langle 2, +\infty \rangle}$. Note the tightening of the constraint (indicated by $\delta \geq 2$ in Figure 2), done to prevent propagation of the pulses smaller than the rising delay of gate 6, i.e., 3.

3.3 Detection interval

The detection interval in timing domain $D^t(f_i)$ is calculated by considering, for all outputs o of the circuit, the description intervals $val@[l, r]_{\langle \delta_{\min}, \delta_{\max} \rangle}$ from the signal descriptors $SD(o, f_i)$, where $val = \neg FV(o)$. These description intervals represent time ranges in which output o assumes the logical value which is opposite to the fault-free value $FV(o)$. We denote the circuit clock cycle time by T_C . We are ultimately interested in conditions under which value $\neg FV(o)$ is assumed at time T_C . For every description interval in question, we calculate the corresponding detection interval. $D^t(f_i)$ is yielded as the union of these detection intervals.

Given a description interval $\neg FV(o)@[l, r]_{\langle \delta_{\min}, \delta_{\max} \rangle}$, its time range $[l, r]$ may have one of the following four shapes:

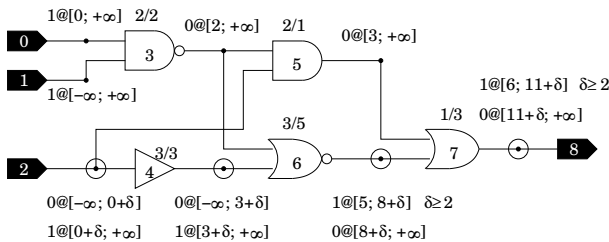


Figure 2: Example circuit with signal descriptors

$[x, y]$, $[x + \delta, y]$, $[x, y + \delta]$ and $[x + \delta, y + \delta]$, where x and y are constants. In the first two cases, the right side of the interval is not affected by the fault. Since the clock cycle time T_C is chosen such that (in the fault-free circuit) all transitions are completed before T_C , no violation of T_C is possible, and the corresponding detection interval is empty. In the third case, value $\neg FV(o)$ is present on the output for all valid values of δ for which $y + \delta$ exceeds T_C . (The left boundary x of the interval is less than T_C by a reasoning similar to the first two cases.) Hence, the values of $y + \delta$ which are both valid and impose $\neg FV(o)$ on o at T_C are situated between $\max\{T_C, y + \delta_{\min}\}$ and $y + \delta_{\max}$. The detection interval is obtained by subtracting y from these bounds. It is $[\max\{T_C, y + \delta_{\min}\} - y, \delta_{\max}]$. Note that the interval is empty if $T_C > y + \delta_{\max}$.

In the final case, the detection interval $[\delta_l, \delta_r]$ must include values of δ such that $x + \delta < T_C < y + \delta$ and δ is valid, i.e., $\delta_{\min} \leq \delta \leq \delta_{\max}$. The right boundary of the detection interval, δ_r , is set to δ_{\max} if $T_C > x + \delta_{\max}$ and to $T_C - x$ if $x + \delta_{\max} > T_C > \delta_{\min}$. δ_l is set to δ_{\min} if $T_C < y + \delta_{\min}$ and to $T_C - y$ if $y + \delta_{\min} < T_C < \delta_{\max}$. The detection interval is empty if either $T_C < x + \delta_{\min}$ or $T_C > y + \delta_{\max}$, because T_C is not included in the interval of times for which $\neg FV(o)$ is assumed on o for any legal values of δ .

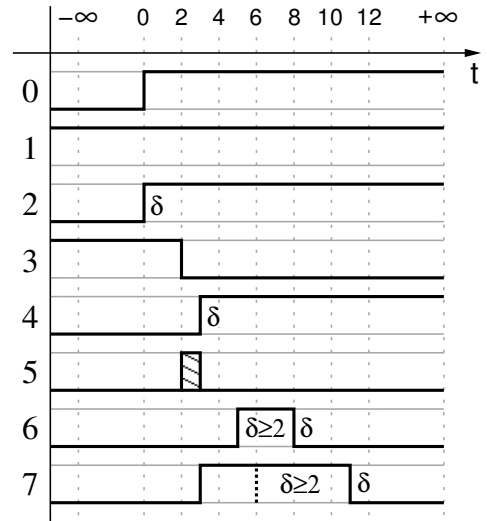


Figure 3: Waveforms in the circuit from Figure 2

3.4 Sequential circuits

The approach is applicable to full-scan (not necessarily enhanced-scan) circuits. The first pattern of the test pair consists of the primary input values and values to be scanned into the flip-flops. The second pattern consists of primary input values only. The values in the flip-flops are obtained functionally. We assume that the circuit is given sufficient time to stabilize after the scan-in of the first pattern, such that a small-delay fault (which, by definition, has a size less than the clock cycle time) has no effect on the values stored in the flip-flops before the application of the second pattern. We reiterate that we assume that gross-delay and other faults are reliably identified by transition and stuck-at test sets and we do not have to quantify the coverage of such defects by our method. The second pattern is applied under the circuit's full speed. The faulty values on the circuit's primary outputs are detected by the test equipment while such values in the flip-flops are identified after scan-out.

4 Delay-to-Resistance-Mapping

In this section, we first describe the method to calculate the nominal delay of a logic gate c in absence of a defect. Then, we give a method to compute the faulty delay of c in presence of a resistive open defect at its output.

The delay of gate c in absence of a defect is modeled by

$$D_{nom} = D_{nocharge} + \alpha \cdot C_L, \quad (2)$$

where $D_{nocharge}$ is the delay value which does not consider the load capacitance. $D_{nocharge}$ and the constant factor α are typically included in a gate library. C_L is the lumped load capacitance, i.e., the sum of input capacitances of all gates driven by c and the parasitic capacitance of the interconnect. For example, if gate c drives a NOR and a NAND gate, the lumped load capacitance is given by

$$C_L = C_{NOR} + C_{NAND} + C_{line}, \quad (3)$$

where C_{NOR} and C_{NAND} are load capacitances of the transistors within NOR and NAND gates driven by gate c and C_{line} is the capacitance of the interconnect.

The delay of a gate driving an interconnect with a resistive-open defect with resistance R_{op} is

$$D = D_{nocharge} + \alpha \cdot C_L + \beta \cdot R_{op} \cdot C_L, \quad (4)$$

where factor β depends on the electrical parameters of the driving gate c . β is constant for a given type of gate and does not depend on R_{op} and C_L . Although β is not included in the gate library, it is easily determined by a SPICE simulation. This is done as a pre-characterization step. The obtained value β can be used when simulating any circuit which contains the gate of respective type and does not need to be re-calculated.

Since the additional delay δ due to the open defect is $\delta = \beta \cdot R_{op} \cdot C_L$, the defect resistance corresponding to a fault size δ is calculated as

$$R_{op} = \delta / (\beta C_L). \quad (5)$$

Since the dependence between δ and R_{op} is monotonic for a fixed fault location, the intervals $D^r(f_i)$ and $G^r(f_i)$ in the resistance domain are obtained by applying this transformation to the boundaries of the respective intervals in the timing domain.

A similar approach was suggested in [12] but no experimental results were reported for open defects modeled by their approach.

5 Experimental Results

We implemented a prototype simulator for the flow outlined above and applied 100 and 1,000 random patterns to combinational ISCAS 85 and sequential ISCAS 89 and ITC 99 circuits. The timing and the process parameters of a 0.18 μm technology with $V_{DD} = 1.8\text{V}$ were used. We assumed clock cycle time T_C equal to the critical path delay plus a safety margin of 20%. For a small-delay fault f_i , we used interval $[D_{max}, T_C]$ as an approximation of $G^t(f_i)$, where D_{max} is the slack of the longest path through the location of f_i , i.e., the difference between T_C and the delay of that path. Based on data in [2], we assumed a constant distribution ρ of defect resistance for low-resistance open defects. We did not use fault dropping.

Table 2 summarizes the results. Columns 2 and 3 contain the clock cycle time T_C in nanoseconds and the number of small-delay faults in the circuit (we considered rising and falling faults at gate outputs). Column 4 reports the realistic fault coverage of 100 random patterns according to Eq. (1). Column 5 quotes the percentage of faults which were detected for any fault size, i.e., faults with non-empty detection interval D^t . Since this metric is an over-estimation of the realistic fault coverage, we call it 'optimistic fault coverage' or O-FC. Column 6 reports the run time in CPU seconds on a 2.6 GHz AMD Opteron server. Columns 7 through 9 give the same information for simulation of 1000 random patterns. The final row quotes the average numbers.

It can be seen that the fault coverage is generally low. One needs to remember that only a specific defect class, namely low-resistance interconnect open defects leading to small-delay faults, are targeted and that the coverage of other defect classes, e.g., stuck-at faults, may be much higher. Also, there is a certain degree of pessimism in the model itself since unknown values (X) in the signal descriptors do not contribute to detection.

One interesting point is that the difference between the realistic and the optimistic fault coverage is large and tends to grow with increasing number of vectors. This suggests that it is important to consider detailed waveform information when calculating the coverage. The run times of the prototype implementation are not comparable with that of

Table 2: Small-delay fault simulation results

Circuit	T_C [ns]	#F	100 patterns			1000 patterns		
			FC	O-FC	T [s]	FC	O-FC	T [s]
c0017	0.17	22	80.43	100.00	0.1	80.43	100.00	0.4
c0095	0.63	64	66.28	100.00	0.2	70.70	100.00	1.5
c0880	2.43	886	29.59	87.58	2.4	41.32	96.39	25.2
c1355	1.86	1174	42.35	82.45	50.4	61.02	93.95	633.7
c1908	3.06	1826	19.55	60.51	7.8	34.91	84.94	115.5
c2670	3.56	2852	22.59	69.78	43.5	32.25	81.38	1137.4
c5315	3.71	4970	32.45	91.07	51.3	42.77	99.38	1371.5
s00027	0.58	34	28.66	91.18	0.1	44.19	100.00	0.7
s00208	0.87	244	21.65	65.57	0.3	32.21	84.43	4.3
s00298	1.09	272	21.76	82.35	0.4	25.93	92.28	7.9
s00344	1.67	368	32.28	89.67	0.6	42.81	97.01	11.1
s00349	1.67	370	32.38	89.46	0.7	42.87	96.76	11.4
s00382	1.86	364	23.22	78.57	0.5	28.34	88.46	10.5
s00386	1.37	344	29.52	58.72	0.6	41.68	80.23	12.2
s00400	1.86	372	19.95	67.47	0.8	28.91	89.52	10.6
s00420	1.21	504	12.89	32.14	0.9	22.58	51.39	12.3
s00444	2.05	410	20.75	79.76	0.9	25.26	86.83	11.6
s00510	1.03	472	36.46	83.69	1.0	44.74	96.61	15.8
s00526	1.09	436	17.17	53.90	0.9	23.83	71.56	10.4
s00641	5.33	866	18.87	81.41	2.2	23.66	92.15	20.9
s00713	5.60	894	18.33	77.96	2.6	22.85	88.37	30.2
s00820	2.16	624	16.49	44.07	1.5	25.83	60.58	23.6
s00832	2.20	620	16.65	44.52	1.4	25.82	60.16	16.8
s00838	1.88	1024	5.56	13.96	1.8	9.37	23.54	18.2
s00953	1.47	880	17.75	48.41	1.6	33.40	81.93	21.1
s01196	2.43	1122	16.29	57.31	2.3	28.48	82.35	26.9
s01238	2.50	1080	16.59	58.33	2.2	28.15	81.57	26.7
s01423	7.86	1496	7.04	62.63	3.5	12.17	87.23	41.1
s01488	2.87	1334	32.45	66.04	3.4	48.93	91.60	35.9
s01494	2.96	1322	32.77	66.19	3.4	49.22	92.06	39.1
s05378	2.68	5986	23.97	64.55	20.0	32.55	80.42	204.4
s09234	4.58	11688	10.35	42.27	55.6	16.32	57.78	563.6
s13207	5.73	17302	7.48	57.95	196.6	10.08	68.80	2165.3
s15850	7.20	20766	9.56	62.22	257.5	12.55	72.13	7118.9
s35932	3.36	35656	43.41	89.72	169.4	48.57	92.01	1852.0
s38417	4.79	47686	20.49	77.33	147.1	25.78	86.87	1868.0
s38584	6.49	41434	14.98	74.01	139.7	18.24	82.96	1733.2
b01c	0.56	94	53.77	100.00	0.2	61.59	100.00	2.3
b02c	0.34	52	62.17	98.08	0.1	65.45	100.00	1.3
b03c	1.11	298	33.09	98.66	0.7	40.56	100.00	7.1
b04c	2.43	1240	12.91	53.79	13.1	31.41	88.47	250.2
b05c	3.14	1076	23.47	80.11	5.9	36.90	92.10	56.1
b06c	0.44	98	53.03	100.00	0.2	56.41	100.00	2.1
b07c	2.29	838	22.29	82.82	2.8	30.58	89.86	40.2
b08c	0.92	334	23.94	74.25	0.7	36.03	95.51	7.8
b09c	0.94	318	28.42	80.50	0.7	32.41	85.22	7.3
b10c	1.20	348	35.85	88.22	0.8	49.79	100.00	7.9
b11c	2.43	1084	23.12	82.56	2.8	32.84	95.85	31.4
b12c	2.01	2000	20.92	72.10	4.7	31.70	89.00	58.2
b13c	0.99	614	23.95	92.02	1.2	34.01	98.21	13.0
Average			26.72	73.12		35.57	86.16	

tools optimized for speed, yet the general applicability of the method to mid-size blocks can be seen. The memory consumption was linear in the size of the circuit and the number of faults and never exceeded 50 MB.

6 Conclusions

The simulator presented in this paper is the first small-delay fault simulator which can handle both combinational and sequential circuits. The correspondence between physical defects, i.e., resistive opens, and their models on the gate level, i.e., small-delay faults, is taken into account. A sophisticated

simulation routine based on signal descriptors is employed in connection with a mapping between the domain of faults, i.e., timing, and the domain of defects, i.e., resistance.

The experiments suggest that an adequate coverage of small-delay faults appears to require specific patterns. Our next steps will be an extension of the electrical model by considering the parasitic capacitances with neighboring interconnects [13] and test generation for small-delay faults. Considering coverage under optimized test application methods [14, 15, 16] is another promising direction for future research.

Acknowledgments

Parts of this work are supported by the German Research Foundation under grant BE 1176/14-1. We are grateful to S. Irajpour and Prof. S. Gupta of University of Southern California for providing the source code of the simulator presented in [5] which we could use as a reference for our own implementation.

7 References

- [1] Y. Sato, S. Hamada, T. Maeda, A. Takatori, and S. Kajihara. Evaluation of the statistical delay quality model. In *ASP Design Automation Conf.*, pages 305–310, 2005.
- [2] R. Rodríguez-Montañés, J.P. de Gyvez, and P. Volf. Resistance characterization for weak open defects. *IEEE Design & Test*, 19(5):18–26, 2002.
- [3] W. Needham, C. Prunty, and E.H. Yeoh. High volume microprocessor test escapes, an analysis of defects our tests are missing. In *Int'l Test Conf.*, pages 25–34, 1998.
- [4] A.K. Pramanick and S.M. Reddy. On the fault detection coverage of gate delay fault detecting tests. *IEEE Trans. on CAD*, 16(1):78–94, 1997.
- [5] S. Irajpour, S.K. Gupta, and M.A. Breuer. Multiple tests for each gate delay fault: Higher coverage and lower test application cost. In *Int'l Test Conf.*, 2005.
- [6] E.S. Park and M.R. Mercer. An efficient delay test generation system for combinational logic circuits. In *Design Automation Conf.*, pages 522–528, 1990.
- [7] H. Konuk. Fault simulation of interconnect opens in digital CMOS circuits. In *Int'l Conf. on CAD*, pages 548–554, 1997.
- [8] 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.
- [9] S. Spinner, J. Jiang, I. Polian, P. Engelke, and B. Becker. Simulating open-via defects. In *Asian Test Symp.*, pages 265–270, 2007.
- [10] M. Renovell, P. Huc, and Y. Bertrand. The concept of resistance interval: A new parametric model for resistive bridging fault. In *VLSI Test Symp.*, pages 184–189, 1995.
- [11] P. Engelke, I. Polian, M. Renovell, and B. Becker. Simulating resistive bridging and stuck-at faults. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 25(10):2181–2192, Oct. 2006.
- [12] Z. Li, X. Lu, W. Qiu, W. Shi, and D.M.H. Walker. A circuit level fault model for resistive opens and bridges. In *VLSI Test Symp.*, pages 379–384, 2003.
- [13] D. Arumí, R. Rodríguez-Montañés, and J. Figueras. Defective behaviours of resistive opens in interconnect lines. In *European Test Symp.*, pages 28–33, 2005.
- [14] J.C.-M. Li, C.-W. Tseng, and E.J. McCluskey. Testing for resistive opens and stuck opens. In *Int'l Test Conf.*, pages 1049–1058, 2001.
- [15] K. Baker, G. Gronthoud, M. Lousberg, I. Schanstra, and C. Hawkins. Defect-based delay testing of resistive vias-contacts a critical evaluation. In *Int'l Test Conf.*, pages 467–476, 1999.
- [16] H. Yan and A.D. Singh. Evaluating the effectiveness of detecting delay defects in the slack interval: A simulation study. In *Int'l Test Conf.*, 2004.