Probabilistic Sensitization Analysis for Variation-Aware Path Delay Fault Test Evaluation

Wagner, Marcus; Wunderlich, Hans-Joachim

Proceedings of the 22nd IEEE European Test Symposium (ETS'17) Limassol, Cyprus, 22-26 May 2017

doi: http://dx.doi.org/10.1109/ETS.2017.7968226

Abstract: With the ever increasing process variability in recent technology nodes, path delay fault testing of digital integrated circuits has become a major challenge. A randomly chosen long path often has no robust test and many of the existing non-robust tests are likely invalidated by process variations. To generate path delay fault tests that are more tolerant towards process variations, the delay test generation must evaluate different non-robust tests and only those tests that sensitize the target path with a sufficiently high probability in presence of process variations must be selected. This requires a huge number of probability computations for a large number of target paths and makes the development of very efficient approximation algorithms mandatory for any practical application. In this paper, a novel and efficient probabilistic sensitization analysis is presented which is used to extract a small subcircuit for a given test vector-pair. The probability that a target path is sensitized by the vector-pair is computed efficiently and without significant error by a Monte-Carlo simulation of the subcircuit.

Preprint

Probabilistic Sensitization Analysis for Variation-Aware Path Delay Fault Test Evaluation

Marcus Wagner, Hans-Joachim Wunderlich
Institute of Computer Architecture and Computer Engineering
University of Stuttgart, Pfaffenwaldring 47, D-70569 Stuttgart, Germany

***Abstract*—With the ever increasing process variability in recent technology nodes, path delay fault testing of digital integrated circuits has become a major challenge. A randomly chosen long path often has no robust test and many of the existing non-robust tests are likely invalidated by process variations. To generate path delay fault tests that are more tolerant towards process variations, the delay test generation must evaluate different non-robust tests and only those tests that sensitize the target path with a sufficiently high probability in presence of process variations must be selected. This requires a huge number of probability computations for a large number of target paths and makes the development of very efficient approximation algorithms mandatory for any practical application.**

In this paper, a novel and efficient probabilistic sensitization analysis is presented which is used to extract a small subcircuit for a given test vector-pair. The probability that a target path is sensitized by the vector-pair is computed efficiently and without significant error by a Monte-Carlo simulation of the subcircuit.

***Index Terms*—delay test, process variations, delay test quality**

I. Introduction

Increasing process variations and high defect densities present major challenges for the delay test of digital integrated circuits [1]. The critical task of delay testing is to check that all sensitizeable paths in the manufactured circuits meet the timing constraints. A path is said to have a path delay fault if the delay of the path exceeds a given threshold, which in most cases is the circuits functional clock cycle time. Any path in a manufactured circuit can have a path delay fault due to process variations, a defect or a combination of both [2].

Delay testing is performed by applying a test vector-pair to the circuit inputs which sensitizes one or more paths in the circuit. A path is called sensitized if a transition is propagated from the beginning of the path, along the path, to the end of the path. To consider the impact of process variations on the delay test, the ability of a test vector-pair to detect a particular delay fault must be evaluated on a population of circuit instances. All circuit instances are functionally identical and each circuit instance has unique, fixed gate and interconnect delays. A test vector-pair that sensitizes a particular target path in one circuit instance might fail to sensitize the path in another circuit instance, which can result in lots of test escapes [3].

Classical path sensitization conditions are of limited use to describe the quality of path delay fault tests [4]. For example, a large number of paths have no robust test, but are non-robustly testable [5]. For any such path, usually many different non-robust tests exist [6] of which only very few might sensitize the path with sufficiently high probability in a randomly chosen circuit instance. Furthermore, a large number of paths that are neither robustly nor non-robustly testable can be functionally sensitized in a subset of all circuit instances [7].

Ignoring the impact of process variations on the target path sensitization is known to cause a significant error during the evaluation of small delay fault tests [8], [9]. Indeed, the quality of every delay test depends on the probability P_{sens} that the test vector-pair sensitizes a particular target path in a randomly chosen circuit instance. To generate and select only those delay tests that most reliably sensitize the target paths under the impact of process variations, P_{sens} must be efficiently computed for each target path, which clearly requires very efficient computation methods for any practical application.

In [10], all gate delays and transition arrival times are treated symbolically. For each path, conditions are derived under which the path is sensitized. The probability P_{sens} can then be computed from these conditions. However, this approach is only applicable to small circuits as the size and complexity of these conditions tends to grow rapidly with the circuit size.

The probability P_{sens} is approximated in [11], [12] by using the bounded delay model. However, it is assumed that all gate and path delays are independent and have a uniform distribution, which results in a large approximation error.

To efficiently compute the probability P_{sens}, a novel probabilistic sensitization analysis is proposed in this work, which extracts for a given test vector-pair a small subcircuit that initially consists only of the target paths. The analysis results are used to extend the subcircuit by additional paths that determine the sensitization of the target paths in any circuit instance. An example of the resulting *representative subcircuit* $\mathcal{S}$ for target path d-h-k-m is shown in fig. 1. Here, the analysis has extended the subcircuit by the path a-e-j-m, which was found to determine the sensitization of the target path by the test vector-pair under the impact of process variations.

The probability P_{sens} can be computed for all target paths very efficiently and without any significant error by a Monte Carlo simulation of only the *representative subcircuit* $\mathcal{S}$. The experimental results with defect free circuits are compared to a subcircuit $\mathcal{L}$, which is obtained by a logic simulation with a 13-valued logic [13]. In presence of a small delay fault, the results are compared to the support region $\mathcal{R}$ of the small delay fault [14], which consists of the union of the input cones of all circuit outputs in the output cone of the fault location. The results show that $\mathcal{S}$ is much smaller than $\mathcal{L}$ and $\mathcal{R}$ and that $\mathcal{S}$

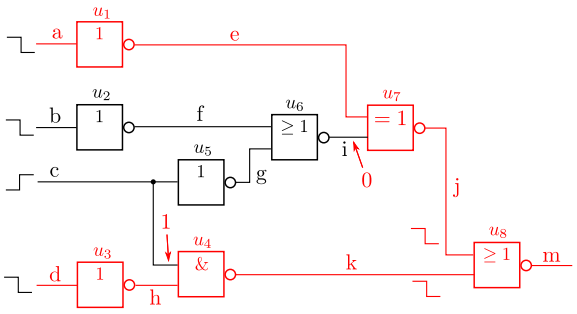


Fig. 1. Circuit and representative subcircuit (red) for a given test vector-pair

is extracted very fast, which results in a large speedup of the computation of P_{sens} by a Monte-Carlo simulation of $\mathcal{S}$.

The remainder of this paper is organized as follows. Section II details the probabilistic sensitization analysis of a single target path. The representative subcircuit for a test vector-pair is extracted in section III by applying this approach to all target paths of the test vector-pair. The experimental results for several industrial benchmark circuits are presented in section IV and conclusions are drawn in section V.

II. Sensitization Analysis of Single Target Path

Given a single target path π. At first, a subcircuit $\mathcal{S}$ that consists only of the target path is extracted, where all floating off-path inputs and all floating circuit outputs have constant logic values. Next, a circuit instance θ_1 and its subcircuit $\mathcal{S}$ are simulated with the given test vector-pair. Suppose the target path is sensitized in either $\mathcal{S}$ or θ_1, but not in both. Then the goal of the sensitization analysis is to find a small extended subcircuit $\mathcal{S}'$, such that the target path is sensitized in either both θ_1 and its subcircuit $\mathcal{S}'$ or in neither circuit. For this, the analysis traces and compares the propagation of the transitions in the circuit and its subcircuit to identify a sensitized path that determines the sensitization of π but that is missing in $\mathcal{S}$.

The analysis then proceeds with another circuit instance θ_2 and the extended subcircuit $\mathcal{S}'$. An important detail of the analysis is that the subcircuit always shares the same gate and interconnect delays with the circuit instance it is compared to.

The flowchart of the proposed analysis is presented in fig. 2. In the first step (A), the circuit instance and its subcircuit are simulated with the given test vector-pair. If the target path is sensitized in only either the circuit instance or its subcircuit, then the transition is traced backwards along the sensitized path in steps (B) and (C). It is guaranteed that the analysis will eventually trace back a transition along a sensitized path in θ_1 that is not part of the subcircuit. The subcircuit is then extended by this sensitized path in step (D). Steps (B), (C) and (D) are skipped if the target path is sensitized in either both the circuit instance and the subcircuit or in neither circuit.

This algorithm is repeated for n randomly chosen circuit instances $\theta_1, \ldots, \theta_n$ until the target path is either sensitized in both θ_i and $\mathcal{S}$ or in neither circuit, for all $1 \leq i \leq n$.

A detailed description of all major steps of the sensitization analysis is given in the following subsections.

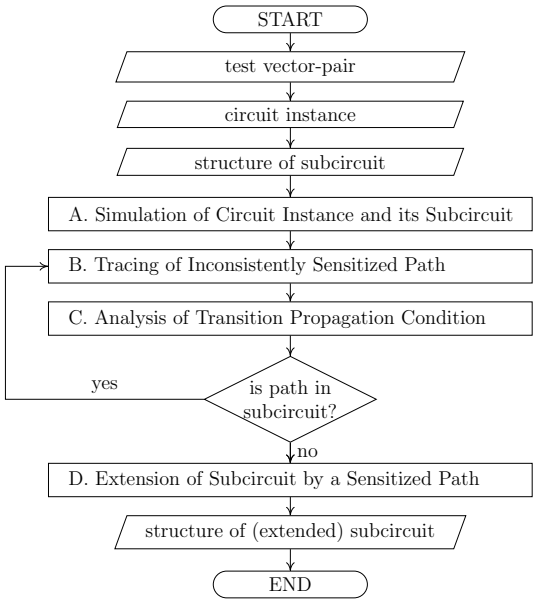


Fig. 2. Flowchart of probabilistic sensitization analysis algorithm

A. Simulation of Circuit Instance and its Subcircuit

The circuit instance and its subcircuit are simulated using a single event-driven timing simulation with the given test vector-pair. After a transition τ at an input of a gate has been propagated to the gate output, the resulting gate output transition is assigned a reference to τ. Using these references, any sensitized path can be easily identified by following these references from a transition at a circuit output until a circuit input has been reached. To efficiently identify and compare the sensitized paths in different circuit instances and the subcircuit, each target path is assigned a unique number that is used for identification. This number is computed from the structural information of the path (e.g. from the names of the interconnects along the path) using a fast hash function.

Although this step is repeatedly executed by the algorithm, each circuit instance only needs to be simulated once and the simulation of the subcircuit is only repeated after extension.

B. Tracing of Inconsistently Sensitized Path

A path π is said to be *inconsistently sensitized* if and only if π is sensitized by the test vector-pair in only either the circuit instance θ_i or the subcircuit $\mathcal{S}$. This means that π contains a gate g that propagates the transition from the on-path gate input to the gate output in only either $\mathcal{S}$ or θ_i. The goal of this step is to find this gate. In the following, a transition is said to *exist in only either* θ_i *or* $\mathcal{S}$, if the transition was propagated along an inconsistently sensitized path.

Without loss of generality, let the target path be sensitized by the test vector-pair in θ_i but not in $\mathcal{S}$. The idea is to trace the transition along the target path in the circuit instance θ_i, starting at the respective circuit output. At each gate, the reference of the gate output transition is followed to the predecessor transition at one of the gate inputs. This tracing continues as long as the following two conditions are satisfied:

(i) The gate is part of the subcircuit $\mathcal{S}$.
(ii) The gate input transition τ exists in θ_i but not in $\mathcal{S}$.

A violation of the first condition (i) implies that at least one gate of the currently traced sensitized path must be missing in the subcircuit $\mathcal{S}$ and this path is added to the subcircuit in step (D). If the second condition (ii) is not satisfied, then this analysis step is complete and the currently considered gate is gate g. The propagation of the transition τ through this gate is analysed in the following step (C).

C. Analysis of Transition Propagation Condition

Suppose the transition τ at an input of a gate causes a transition τ' at the gate output at time t', without loss of generality, only in θ_i but not in $\mathcal{S}$. By definition, the subcircuit $\mathcal{S}$ always inherits the gate delays from the circuit instance θ_i it is compared to. This implies that another transition $\tilde{\tau}$ must exist at the gate inputs in only either θ_i or $\mathcal{S}$, which is responsible for τ not being propagated to the gate output in $\mathcal{S}$. The goal of this step is to identify $\tilde{\tau}$, which can then be traced backwards along an inconsistently sensitized path in step (B).

A gate input transition is propagated to the gate output if and only if it satisfies the transition propagation condition of the gate model, which is the conjunction of the dynamic sensitization condition and the inertial delay condition.

1) Analysis of Dynamic Sensitization Condition: This condition describes that τ can only be propagated to the gate output if the result of the logic function that the gate implements changes in response to τ. Clearly, this condition is always satisfied for a buffer or an inverter. For (N)AND/(N)OR gates, the other gate inputs must have the non-controlling value at the arrival time of τ. This algorithm requires an exception for XOR/XNOR gates, where the dynamic sensitization condition is redefined to be satisfied if and only if the logic values at all other gate inputs match in θ_i and $\mathcal{S}$ at the arrival time of τ.

A violation of the dynamic sensitization condition implies at least one transition $\tilde{\tau}$ must exist at one of the off-path inputs in only either θ_i or $\mathcal{S}$ and the analysis proceeds by tracing $\tilde{\tau}$ in step (B). In case multiple transitions exist at the off-path inputs in only either θ_i or $\mathcal{S}$, the one which occurs closest in time to t is selected instead.

The example in fig. 3 shows the last NOR gate (u_8) in fig. 1, which has a falling transition at each gate input. The waveforms in fig. 3 are the result of the simulation of the circuit instance θ_i with the given test vector-pair. At the arrival time t of the on-path input transition τ, the off-path input 'j' has the controlling value so that τ is not propagated to the gate output. Suppose the subcircuit $\mathcal{S}$ consists only of the target path d-h-k-m and the off-path input 'j' has the constant value '0' so that τ satisfies the dynamic sensitization condition in the subcircuit. In this case, $\tilde{\tau}$ is responsible for that τ is propagated to the gate output only in $\mathcal{S}$ but not in θ_i and the analysis proceeds by tracing $\tilde{\tau}$ in step (B).

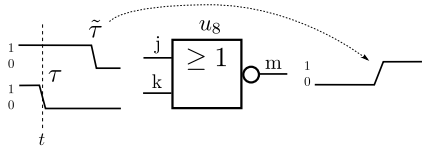


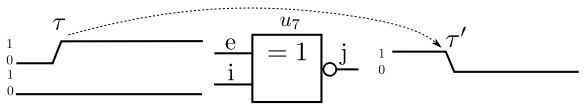
Fig. 3. Dynamic sensitization condition of NOR gate satisfied by $\tilde{\tau}$ but violated by τ

2) Analysis of Inertial Delay Condition: The inertial delay condition is used to model the limitation that a real gate cannot produce arbitrary short glitches at the gate output. This condition states that if a transition τ_1 occurs at a gate input at time t_1 and causes a transition at the gate output after a delay δ, then another transition τ_2 that arrives at a time $t_2 > t_1$ at the gate inputs will only be propagated to the gate outputs if $t_2 > t_1 + \delta$. Otherwise, the incomplete charging or discharging of the gate output capacitance initiated by τ_1 is reversed by τ_2 and the gate output remains constant.

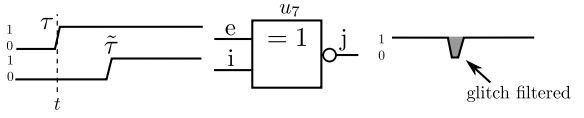
If τ violates the inertial delay condition then at least one other transition $\tilde{\tau}$ must exist at the gate inputs in only either θ_i or $\mathcal{S}$. For a logic gate with only a single input, the proposed analysis selects the last gate input transition before time t' that exists in only either θ_i or $\mathcal{S}$. The analysis then proceeds by tracing this transition in step (B).

For gates with multiple inputs, only those gate input transitions that satisfy the dynamic sensitization condition can be propagated to the gate output. Of those transitions, only those that occur before or directly after τ but no later than t' are *relevant* for the propagation of τ. Suppose L_{θ_i} and $L_{\mathcal{S}}$ are the sets of relevant gate input transitions in $\mathcal{S}$ and θ_i, respectively. Then $L_{\theta_i} \cup L_{\mathcal{S}}$ must contain a transition that exists in only either θ_i or $\mathcal{S}$, which causes τ not to be propagated to the gate output in the subcircuit $\mathcal{S}$.

For example, suppose the subcircuit $\mathcal{S}$ in fig. 1 also contains the path c-g-i-j-m and 'f' is set to logic '0'. The waveforms at the inputs and outputs of the XNOR gate u_7 in the circuit instance θ_i and the subcircuit $\mathcal{S}$ are shown in fig. 4. In the circuit instance, the glitch at the output 'i' of u_6 does not occur, so that 'i' remains constant zero. In the subcircuit $\mathcal{S}$, however, the falling transition at 'g' is propagated to the output of u_6, because 'f' is set to the non-controlling value '0'. Therefore, $\tilde{\tau}$ appears at 'i' shortly after τ occurs at 'e'. In this example, τ has not been propagated to the gate output by the time $\tilde{\tau}$ occurs at the gate input, so that the gate output will remain constant '1'.



(a) τ is propagated to gate output in circuit instance θ_i



(b) τ violates inertial delay condition in subcircuit $\mathcal{S}$

Fig. 4. Waveforms at XNOR gate inputs and output after the simulation of the test vector-pair

In this example, the sets of relevant transitions are $L_{\theta_i} = \{\tau\}$ and $L_{\mathcal{S}} = \{\tau, \tilde{\tau}\}$. The analysis then selects the first transition in $L_{\theta_i} \cup L_{\mathcal{S}}$ after τ that exists in only either θ_i or $\mathcal{S}$. If no such transition exists, then the last transition in $L_{\theta_i} \cup L_{\mathcal{S}}$ before τ that exists in only either θ_i or $\mathcal{S}$, is chosen instead. In this example, the only possible choice is $\tilde{\tau}$, which is then traced back in step (B).

D. Extension of Subcircuit by a Sensitized Path

To add a new path to the subcircuit, all gates and interconnects along this path are marked in the circuit instance and then all marked gates and interconnects are extracted. Afterwards, the floating off-path inputs of the new path are set to their respective non-controlling values. The floating off-path inputs of a XOR/XNOR gate are set the logic value that the off-path inputs had during the simulation of θ_i at the time the transition occurred at the respective on-path input. Similarly, all floating circuit outputs are set to those logic values that were observed during the simulation of θ_i at the clock cycle time. The above described constant logic assignments are stored to speed up later extensions of the subcircuit by additional paths.

III. Extraction of Representative Subcircuit

In this section, the proposed sensitization analysis for a single target path is consecutively applied to each of the target paths of a given test vector-pair to extract a so called representative subcircuit, which is defined as follows.

Let ν denote a test vector-pair that sensitizes any path in a set of target paths Π with non-negligible probability. Suppose θ is a randomly chosen circuit instance and $\mathcal{S}$ is its subcircuit, which contains at least the target paths Π and all floating inputs of $\mathcal{S}$ are set to constant logic values. The subcircuit $\mathcal{S}$ is called *representative subcircuit for* ν, if the probability P_{ips} that a randomly chosen target path is inconsistently sensitized by ν in θ and $\mathcal{S}$ is below a given threshold P_{th} with 95% confidence.

Given a subcircuit $\mathcal{S}$ that consists only of the target paths Π and a set of n circuit instances $\Theta = \{\theta_1, \ldots, \theta_n\}$. After the circuit instance $\theta_1 \in \Theta$ and its subcircuit $\mathcal{S}$ have been simulated, the algorithm in section II is applied to each of the target paths in Π that are sensitized in θ_1 or $\mathcal{S}$. This process is repeated for all circuit instances in Θ until none of the paths in Π is inconsistently sensitized in $\theta_1, \ldots, \theta_n$ and $\mathcal{S}$.

To test if $\mathcal{S}$ is a representative subcircuit, another set of n circuit instances $\Theta' = \{\theta'_1, \ldots, \theta'_n\}$ is randomly chosen. Let p denote the probability that at least one of k randomly chosen target paths is inconsistently sensitized in a randomly chosen circuit instance θ'_i and $\mathcal{S}$. If k is much smaller than $|\Pi|$, then $p \approx 1 - (1 - P_{ips})^k$. Suppose none of the k target paths is inconsistently sensitized in $\theta'_1, \ldots, \theta'_n$ and $\mathcal{S}$. Then the *rule of three* states that $p \in [0, 3/n]$ with 95% confidence [15]. Thus, $\mathcal{S}$ is a representative subcircuit if and only if

$$n \geq 3/\big(1 - (1 - P_{ips})^k\big) \tag{1}$$

is satisfied, where suitable small values for n and k must be chosen by replacing P_{ips} in ineq. (1) with the threshold P_{th}.

Otherwise, if one of the k target paths is inconsistently sensitized, then $\mathcal{S}$ is extended further using the joint set of circuit instances $\Theta \cup \Theta'$ and this process is repeated.

It is also possible to extract a subcircuit that is a representative subcircuit for multiple test vector-pairs by extending the representative subcircuit for one test vector-pair with the analysis in section II using another test vector-pair.

IV. Experimental Results

Several NXP benchmark circuits were first optimized for speed and then mapped to the NanGate 45nm Open Cell Library [16] using a commercial synthesis tool. The accurate gate model defined by the Verilog HDL standard [17] is used. For the generation of the circuit instances, every delay value of a gate is assumed to have a normal distribution with mean μ and variance $(c_v\mu)^2$, where μ is the nominal delay value from the standard delay format description of the synthesized circuit. To study innovative process technology nodes, a variation coefficient of $c_v = 0.25$ was selected [18]. The correlation coefficient between any pair of delay values is 0.5 to account for both inter-die and intra-die variations.

To detect path delay faults, a commercial static timing analysis tool was used to identify the 10000 longest paths in each benchmark circuit, where at most 5 paths were allowed to terminate at the same circuit input or output. Afterwards, a commercial ATPG tool was used to create test vector-pairs that non-robustly sensitize as many of these paths as possible.

Additional test vector-pairs for the detection of small delay faults have been generated as follows. For each circuit, a set of 20000 randomly chosen marginally detectable small delay faults was created. A small delay fault is marginally detectable if the fault size is equal to the slack of the longest sensitizable path through the fault site. Afterwards, a commercial ATPG tool was used to create test vector-pairs that non-robustly sensitize the longest sensitizable paths through each fault site.

Only sufficiently long paths, which are also sensitized by the test vector-pair, can have a significant impact on the delay test result. For path delay fault tests, it is therefore assumed that the target paths of a test vector-pair are the critical paths that are sensitized by the test vector-pair in a randomly chosen circuit instance with non-negligible probability. A logical path is called *critical path* if and only if $\mu + 3\sigma \geq T_{clk}$, where μ and σ denote the mean and standard deviation of the path delay, respectively, and T_{clk} is the clock cycle time. For small delay fault tests, the set of target paths is further restricted to only those critical paths that pass through the fault site.

For a fixed k, decreasing the threshold P_{th} increases the number of circuit instances n that are required to satisfy ineq. (1), resulting in greater size and simulation accuracy of the subcircuit. For high accuracy, a tiny threshold of $P_{th} = 0.0062$ and a small $k = 10$ was chosen, so that $n = 50$ is sufficient.

The subcircuit $\mathcal{S}$ is compared to three other subcircuits $\bar{\mathcal{S}}$, $\mathcal{L}$ and $\mathcal{R}$, where $\bar{\mathcal{S}}$ consists only of the critical paths that are sensitized in the nominal circuit instance and $\mathcal{L}$ is obtained by a simulation with a 13-valued logic [13] after which any gate with an output that is constant or that has become unconnected

is removed. In presence of a small delay fault, $\mathcal{S}$ and $\bar{\mathcal{S}}$ are compared to the support region $\mathcal{R}$ of the fault location [14].

The accuracy of the simulation of the final subcircuits $\bar{\mathcal{S}}$ and $\mathcal{S}$ was evaluated using 10^4 circuit instances of $\mathcal{L}$ and $\mathcal{R}$, respectively. After the simulation of a circuit instance θ and its subcircuits $\bar{\mathcal{S}}$ and $\mathcal{S}$, the sets of sensitized target paths in θ, $\bar{\mathcal{S}}$ and $\mathcal{S}$ are identified and compared to compute the probability that a target path is inconsistently sensitized. Furthermore, the logic values at the circuit outputs of θ, $\bar{\mathcal{S}}$ and $\mathcal{S}$ were observed at the clock cycle time. If the observed output values differ from the expected values, then a fault has been detected. A subcircuit is said to provide an *inconsistent test result*, if a fault is detected in only either the circuit instance or the subcircuit.

The experimental results for path delay fault tests with defect free circuits are presented in table Ia. Further experimental results are shown in table Ib for small delay fault tests in circuits with a marginally detectable small delay fault. Both tables show the average results over all test vector-pairs.

The name and the number of gates (#gates) of the NXP benchmark circuit is shown in the first and second column, respectively. The clock cycle time "T_{clk}" was determined such that 40%, 20% or 5% of the defect-free manufactured chips would fail the timing requirements due to process variations. The next columns "$|\mathcal{L}|$" and "$|\mathcal{R}|$" present the relative size of the subcircuits $\mathcal{L}$ and $\mathcal{R}$, respectively. The *relative size* of a subcircuit is defined as the relative number of gates in the subcircuit, compared to the number of gates in the circuit. The following six columns and the last seven columns present the results for the subcircuit $\bar{\mathcal{S}}$ and $\mathcal{S}$, respectively.

The average number of target paths of a test vector-pair is shown in column "#targ. paths". $\mathcal{S}$ contains many more target paths than $\bar{\mathcal{S}}$ because only few critical paths are sensitized by the test vector-pair in the nominal circuit instance and many other critical paths are sensitized in other circuit instances.

The relative size of the subcircuits $\bar{\mathcal{S}}$ and $\mathcal{S}$ is presented in "$|\bar{\mathcal{S}}|$" and "$|\mathcal{S}|$", respectively. The experimental results show that the representative subcircuit $\mathcal{S}$ is only slightly greater than the subcircuit $\bar{\mathcal{S}}$ and, most importantly, much smaller than the

TABLE I

AVERAGE SUBCIRCUIT SIZE, SIMULATION ACCURACY AND RUNTIME FOR CONSTRUCTION AND MONTE CARLO SIMULATION OF SUBCIRCUITS $\bar{\mathcal{S}}$ AND $\mathcal{S}$ OVER ALL TEST VECTOR-PAIRS, COMPARED TO MONTE CARLO SIMULATION OF SUBCIRCUITS $\mathcal{L}$ AND $\mathcal{R}$, RESPECTIVELY

			[13]	subcircuit $\bar{\mathcal{S}}$						**representative subcircuit $\mathcal{S}$**						
circuit name	#gates	T_{clk} [ps]	$\|\mathcal{L}\|$ [%]	#targ. paths	|cone| [%]	$\|\bar{\mathcal{S}}\|$ [%]	P_{ips} [%]	P_{itr} [%]	Speedup	#cir. inst.	#targ. paths	|cone| [%]	$\|\mathcal{S}\|$ [%]	$\boldsymbol{P_{ips}}$ **[%]**	P_{itr} [%]	**Speedup**
p35k	28115	1163.6	38.25	11.1	40.92	0.28	38.89	7.67	52.0	140.3	106.8	49.14	1.24	**0.27**	0.10	**11.7**
		1266.8		8.9	37.63	0.24	36.08	4.73	55.1	140.0	94.7	48.11	1.15	**0.29**	0.04	**12.2**
		1412.7		6.0	32.00	0.19	36.09	1.51	57.1	144.2	67.6	46.34	0.97	**0.32**	0.01	**13.0**
p45k	26954	889.3	43.76	19.2	10.23	0.81	47.81	27.39	32.7	132.1	130.2	14.03	3.50	**0.15**	0.02	**6.2**
		968.4		11.7	7.97	0.50	48.90	11.63	42.1	148.7	80.2	11.12	2.85	**0.20**	0.01	**6.7**
		1084.8		3.3	3.42	0.16	56.96	0.68	61.4	225.9	18.0	6.89	1.31	**0.30**	0.01	**7.9**
p77k	41797	5805.2	44.99	2.9	2.69	0.10	1.21	2.03	107.9	101.8	133.3	3.78	1.02	**0.02**	0.08	**11.3**
		6377.2		2.1	2.28	0.07	1.53	1.89	114.0	101.3	93.2	3.57	0.86	**0.03**	0.04	**12.2**
		7137.5		1.2	1.45	0.05	2.49	0.92	124.8	101.3	47.6	3.56	0.65	**0.04**	0.01	**15.0**
p78k	57535	1213.3	76.74	72.8	16.66	1.74	40.82	43.55	35.6	158.6	595.1	37.02	13.23	**0.22**	0.01	**3.5**
		1325.4		23.8	6.52	0.68	41.87	9.85	75.3	179.5	190.5	16.30	5.43	**0.25**	0.00	**6.2**
		1485.2		3.5	1.18	0.13	46.35	0.39	190.2	248.9	19.9	3.09	0.92	**0.28**	0.01	**9.5**
p81k	91756	1018.5	34.00	118.6	21.72	2.17	29.49	75.13	13.6	125.7	1444.4	24.13	12.17	**0.12**	0.02	**2.0**
		1109.7		60.2	18.51	1.25	33.08	61.58	21.9	146.3	721.6	22.75	10.70	**0.20**	0.01	**2.1**
		1238.6		14.5	8.65	0.31	34.85	3.80	68.2	230.3	144.0	18.01	4.97	**0.36**	0.00	**3.8**
p100k	61749	1400.1	45.02	28.7	9.10	0.71	20.33	62.44	41.7	131.5	380.2	11.42	4.52	**0.11**	0.01	**4.9**
		1533.9		14.1	6.49	0.39	22.01	30.22	59.1	155.2	179.9	8.79	2.99	**0.16**	0.01	**6.4**
		1710.6		3.2	2.42	0.11	27.34	1.15	91.6	226.0	28.5	4.48	0.98	**0.25**	0.02	**9.5**
p267k	138912	787.9	41.01	293.4	11.43	1.07	69.21	52.38	26.5	120.2	908.0	16.20	2.76	**0.14**	0.01	**6.6**
		856.2		132.9	6.40	0.53	65.12	31.38	40.6	126.7	457.2	9.78	1.54	**0.17**	0.01	**8.8**
		951.6		29.2	1.99	0.15	53.36	1.74	66.6	147.1	128.3	3.21	0.47	**0.21**	0.01	**11.7**
p330k	184425	1023.5	44.96	254.5	17.26	1.59	46.46	78.49	23.5	125.4	1933.9	21.99	7.69	**0.17**	0.01	**3.2**
		1116.9		107.4	12.43	0.81	42.93	57.99	38.0	150.7	880.4	16.82	5.67	**0.25**	0.00	**3.9**
		1246.4		18.3	3.75	0.15	52.16	1.32	85.6	269.3	103.9	8.02	1.71	**0.40**	0.01	**6.5**

(a) Test vector-pairs for the detection of path delay faults, applied to defect free circuits

			[14]	subcircuit $\bar{\mathcal{S}}$						**representative subcircuit $\mathcal{S}$**						
circuit name	#gates	T_{clk} [ps]	$\|\mathcal{R}\|$ [%]	#targ. paths	|cone| [%]	$\|\bar{\mathcal{S}}\|$ [%]	P_{ips} [%]	P_{itr} [%]	Speedup	#cir. inst.	#targ. paths	|cone| [%]	$\|\mathcal{S}\|$ [%]	$\boldsymbol{P_{ips}}$ **[%]**	P_{itr} [%]	**Speedup**
p35k	28115	1412.7	56.88	4.4	21.71	0.16	79.55	10.74	79.6	105.8	12.0	21.93	0.35	**0.13**	0.15	**36.0**
p45k	26954	1084.8	3.39	3.5	1.37	0.12	84.17	4.11	4.6	105.4	6.0	1.39	0.21	**0.10**	0.08	**2.6**
p77k	41797	7137.5	3.18	3.3	0.91	0.07	88.46	2.78	7.9	102.0	5.8	0.92	0.10	**0.10**	0.08	**4.6**
p78k	57535	1485.2	1.04	4.2	0.40	0.07	71.77	13.22	4.4	107.0	9.7	0.42	0.14	**0.09**	0.14	**2.0**
p81k	91756	1238.6	3.71	2.9	1.08	0.05	78.56	8.60	16.4	106.6	7.4	1.10	0.09	**0.12**	0.12	**6.7**
p100k	61749	1710.6	1.59	3.7	0.61	0.06	69.63	6.63	4.5	104.4	9.4	0.62	0.12	**0.09**	0.11	**2.5**
p267k	138912	951.6	1.11	4.3	0.30	0.02	89.39	9.22	3.0	104.1	7.4	0.31	0.04	**0.07**	0.06	**1.9**
p330k	184425	1246.4	1.37	5.5	0.46	0.02	82.92	9.57	3.8	107.1	10.8	0.47	0.04	**0.10**	0.10	**2.2**

(b) Test vector-pairs for the detection of small delay faults, applied to circuits with a marginally detectable small delay fault

subcircuits $\mathcal{L}$ and $\mathcal{R}$. Furthermore, the average relative size of the joint input cone of the target paths, which is shown in column "|cone|", is also much larger than $|\mathcal{S}|$.

Next, the accuracy of the Monte Carlo simulation of the subcircuits is evaluated. The probability P_{itr} that a randomly chosen target path is inconsistently sensitized in a randomly chosen circuit instance and its subcircuit is given in the column "P_{ips}" in percent. The next column "P_{itr}" shows the probability of an inconsistent test result in percent. Both probabilities are quite high for the simulation of $\bar{\mathcal{S}}$ and P_{itr} increases rapidly with the clock frequency. In contrast, the simulation of the representative subcircuit $\mathcal{S}$ is highly accurate, so that P_{itr} and P_{ips} are almost zero. For path delay fault and small delay fault tests, it was found that $P_{ips} \leq P_{th}$ holds for $96,3\%$ and $97,7\%$ of all test vector-pairs, respectively. Therefore, the proposed approach meets and exceeds the accuracy goal set by the proposed confidence driven construction of $\mathcal{S}$. Clearly, a target path is sensitized in a circuit instance if and only if the path is sensitized in its subcircuits $\mathcal{L}$, $\mathcal{R}$ and "cone".

Finally, the computational cost for the extraction and Monte-Carlo simulation of the subcircuits is evaluated. The average number of circuit instances that were used for the construction of $\mathcal{S}$ is presented in column "#cir. inst.", excluding the additional n circuit instances that are required to test if $\mathcal{S}$ is a representative subcircuit. The average speedup of the Monte-Carlo simulation of $\bar{\mathcal{S}}$ and $\mathcal{S}$ is defined as the quotient of the average runtime of the Monte-Carlo simulation of $\mathcal{L}$ or $\mathcal{R}$ and the average runtime for the extraction and Monte-Carlo simulation of the subcircuit $\bar{\mathcal{S}}$ and $\mathcal{S}$, respectively. The average speedup attained by the extraction and Monte Carlo simulation of the subcircuit $\bar{\mathcal{S}}$ and $\mathcal{S}$ is shown in column "Speedup". A large average speedup of up to $190\times$ is achieved by the extraction and Monte Carlo simulation of the subcircuit $\bar{\mathcal{S}}$. Similarly, the extraction and Monte Carlo simulation of the representative subcircuit $\mathcal{S}$ is up to $36\times$ faster than a Monte Carlo simulation of the support region $\mathcal{R}$ of the small delay fault. The speedup is particularly large for p35k because of the relatively large size of $\mathcal{R}$. By adjusting the threshold P_{th}, a suitable compromise between the speedup and the simulation accuracy can be found.

The experimental results confirm that the representative subcircuit provides very high simulation accuracy and its small size allows the efficient evaluation of the target path sensitization and the delay test with a given test vector-pair in presence of large process variations.

V. Conclusion

Under the impact of process variations, a test vector-pair that sensitizes a target path in the nominal circuit instance does not necessarily sensitize the path in a randomly chosen circuit instance, which can cause a large number of test escapes. To generate and select only those test vector-pairs that most reliably sensitize the target paths in presence of process variations, it is essential to efficiently compute the probability that a test vector-pair sensitizes the target paths in a randomly chosen circuit instance.

To efficiently compute these probabilities, a small subcircuit is extracted, which initially consists only of the target paths. By analysing the target path sensitization in both circuits under the impact of process variations, the subcircuit is gradually extended by additional paths and finally becomes a representative subcircuit. For path and small delay fault tests, the experimental results show that the target path sensitization probability is computed very efficiently and without any significant error by the extraction and Monte-Carlo simulation of the subcircuit.

VI. Acknowledgement

This work has been supported by the German Research Foundation (DFG) under grant Wu245/16-1 (PARSIVAL).

References

[1] B. Becker, S. Hellebrand, I. Polian, B. Straube, W. Vermeiren, and H.-J. Wunderlich, "Massive statistical process variations: A grand challenge for testing nanoelectronic circuits," in *Int. Conf. on Dependable Systems and Networks Workshops (DSN-W)*, Chicago, IL, USA, jun 2010, pp. 95–100.

[2] M. L. Bushnell and V. D. Agrawal, *Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits*. Kluwer Academic Publishers, 2000.

[3] U. Ingelsson, B. Al-Hashimi, S. Khursheed, S. Reddy, and P. Harrod, "Process Variation-Aware Test for Resistive Bridges," *IEEE Trans. Computer-Aided Design*, vol. 28, no. 8, pp. 1269–1274, 2009.

[4] M. Sauer, A. Czutro, I. Polian, and B. Becker, "Small-Delay-Fault ATPG with Waveform Accuracy," in *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, San Jose, CA, nov 2012, pp. 30–36.

[5] C. Lin and S. Reddy, "On Delay Fault Testing in Logic Circuits," *IEEE Trans. Computer-Aided Design*, vol. 6, no. 5, pp. 694–703, 1987.

[6] S. Eggersglüß and R. Drechsler, "As-Robust-As-Possible test generation in the presence of small delay defects using pseudo-Boolean optimization," in *Proc. Design, Automation and Test in Europe (DATE)*, Grenoble, France, mar 2011.

[7] A. Krstic and K.-T. Cheng, "Generation of high quality tests for functional sensitizable paths," in *Proc. VLSI Test Symp. (VTS)*, Princeton, NJ, USA, may 1995, pp. 374–379.

[8] M. Wagner and H.-J. Wunderlich, "Efficient Variation-Aware Statistical Dynamic Timing Analysis for Delay Test Applications," in *Proc. Design, Automation and Test in Europe (DATE)*, Grenoble, France, mar 2013.

[9] ——, "Incremental Computation of Delay Fault Detection Probability for Variation-Aware Test Generation," in *IEEE European Test Symp. (ETS)*, Paderborn, Germany, may 2014.

[10] N. Ishiura, M. Takahashi, and S. Yajima, "Time-symbolic simulation for accurate timing verification of asynchronous behavior of logic circuits," in *Proc. Design Automation Conf. (DAC)*, Las Vegas, NV, USA, jun 1989, pp. 497–502.

[11] S. Bose and V. D. Agrawal, "Delay Test Quality Evaluation Using Bounded Gate Delays," in *Proc. VLSI Test Symp. (VTS)*, Berkeley, CA, USA, may 2007, pp. 23–28.

[12] D. Jayaraman and S. Tragoudas, "A method to determine the sensitization probability of a non-robustly testable path," in *Proc. Int. Symp. on Quality Electronic Design (ISQED)*, Santa Clara, CA, USA, mar 2013, pp. 676–681.

[13] J. Hayes, "Digital Simulation with Multiple Logic Values," *IEEE Trans. Computer-Aided Design*, vol. 5, no. 2, pp. 274–283, apr 1986.

[14] I. Hamzaoglu and J. H. Patel, "New Techniques for Deterministic Test Pattern Generation," *Journal of Electronic Testing*, vol. 15, no. 1/2, pp. 63–73, 1999.

[15] J. A. Hanley and A. Lippman-Hand, "If Nothing Goes Wrong, Is Everything All Right?" *Journal of the American Medical Association*, vol. 249, pp. 1743–1745, 1983.

[16] "Nangate 45nm Open Cell Library," aug 2011.

[17] IEEE Standards Department, "Verilog Hardware Description Language," *IEEE Std 1364-2005*, apr 2006.

[18] Y. Ye, S. Gummalla, C.-C. Wang, C. Chakrabarti, and Y. Cao, "Random variability modeling and its impact on scaled CMOS circuits," *Journal of Computational Electronics*, vol. 9, no. 3, pp. 108–113, 2010.