Efficient Variation-Aware Statistical Dynamic Timing Analysis for Delay Test Applications

Wagner, Marcus; Wunderlich, Hans-Joachim

Proceedings of the Conference on Design, Automation and Test in Europe (DATE'13)
Grenoble, France, 18-22 March 2013

doi: http://dx.doi.org/10.7873/DATE.2013.069

Abstract: Increasing parameter variations, caused by variations in process, temperature, power supply, and wear-out, have emerged as one of the most important challenges in semiconductor manufacturing and test. As a consequence for gate delay testing, a single test vector pair is no longer sufficient to provide the required low test escape probabilities for a single delay fault. Recently proposed statistical test generation methods are therefore guided by a metric, which defines the probability of detecting a delay fault with a given test set. However, since run time and accuracy are dominated by the large number of required metric evaluations, more efficient approximation methods are mandatory for any practical application. In this work, a new statistical dynamic timing analysis algorithm is introduced to tackle this problem. The associated approximation error is very small and predominantly caused by the impact of delay variations on path sensitization and hazards. The experimental results show a large speedup compared to classical Monte Carlo simulations.

Preprint

Efficient Variation-Aware Statistical Dynamic Timing Analysis for Delay Test Applications

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

***Abstract*—Increasing parameter variations, caused by variations in process, temperature, power supply, and wear-out, have emerged as one of the most important challenges in semiconductor manufacturing and test. As a consequence for gate delay testing, a single test vector pair is no longer sufficient to provide the required low test escape probabilities for a single delay fault. Recently proposed statistical test generation methods are therefore guided by a metric, which defines the probability of detecting a delay fault with a given test set. However, since runtime and accuracy are dominated by the large number of required metric evaluations, more efficient approximation methods are mandatory for any practical application.**

In this work, a new statistical dynamic timing analysis algorithm is introduced to tackle this problem. The associated approximation error is very small and predominantly caused by the impact of delay variations on path sensitization and hazards. The experimental results show a large speedup compared to classical Monte Carlo simulations.

I. Introduction

In recent years, parameter variations have emerged as a new challenge for manufacturing test methods [1]–[3]. The detrimental impact of these variations on the delay test quality may lead to many test escapes due to test invalidation [4]. A delay test is called invalid, if the test fails to detect a target fault due to delay variations, hazards or other reasons. More stringent path sensitization conditions can reduce the risk of test invalidation, but these conditions may not be satisfiable for a large number of paths [5]. Hence, a single test vector pair can no longer provide sufficiently low test escape probabilities for a delay fault [6].

In order to keep test cost within an acceptable budget, statistical delay test generation methods must be aware of the diminishing returns in delay test quality by each additional test vector pair. The most promising approach is to guide the test generation procedure by a metric [7]–[10], which defines the detection probability of a delay fault with a given test set. However, this metric must be computed $O(k^2n)$ times to evaluate k applicable test vector pairs for each gate delay fault in a circuit with n gates [8]. Hence, the accuracy and the computational complexity of any practical application strongly depend upon the efficiency of the metric evaluations.

For the research and development of new statistical test generation methods, the output deviation was proposed as a low-cost surrogate metric [11]. However, the output deviation tends to saturate, and equal values are obtained for long and intermediate sensitized paths.

In [7], the detection probability is used to guide the selection of the longest paths through every gate of the circuit, which are subsequently targeted for test generation. However, in order to reduce the runtime of the algorithm, the authors approximate the path delays as independent random variables.

An alternative method is to create a superset of test vector pairs for each fault site. Then, a minimal subset of these test vector pairs with sufficiently high delay test quality is selected. A pattern-selection algorithm following this principle was proposed in [8]. However, the detection probability was computed using Monte Carlo simulations of the entire circuit, which is inefficient for practical applications. Instead, only sufficiently long paths, which are also sensitized by the given test set, can have a significant impact on the delay test result and should therefore be considered.

In a related work [12], the authors proposed to utilize statistical static timing analysis techniques to estimate the process-induced variation in pattern delays. An event-driven timing simulation was used to identify the portion of the circuit, which is sensitized by a given test set. After removing the remaining parts of the circuit, a block-based statistical static timing analysis technique was applied to estimate the pattern delay distribution. However, the block based approach results in unnecessary error accumulation and requires all gate and interconnect delays to have a normal distribution.

The approach presented here combines several efficient block- and path-based statistical static timing analysis techniques, to minimize the approximation error and the computational complexity. In contrast to [12], only sensitized paths with a significant probability of causing a timing failure are considered. The algorithm is not restricted to normally distributed gate and interconnect delays, which is particularly important for the consideration of the exponential fault size distribution of a gate delay fault. Furthermore, both structural as well as spatial correlations are taken into account.

The contribution of this work is twofold: (1) based on [12], a new advanced statistical dynamic timing analysis algorithm is introduced, which can be used in test applications; (2) the efficiency of the algorithm is demonstrated in the context of delay variation aware pattern selection for small delay defects.

The remainder of the paper is organized as follows. Section II describes the proposed approximation algorithm. The experimental results for NXP benchmark circuits are presented in section III. Conclusions are drawn in section IV.

II. Statistical Timing Analysis Algorithm

To detect a particular gate delay fault, a suitable set of test vector pairs Θ is applied to the circuit. The proposed algorithm approximates the probability, that an incorrect logic value will be captured into at least one scan flip-flop. More precisely, the algorithm computes the probability of a timing failure, which occurs if at least one primary output of the combinational network has not stabilized to its final logic value within the system clock cycle time T_{clk}. It is assumed, that the whole circuit is subject to delay variation, which may cause path delay faults by itself or in combination with a gate delay fault.

A sensitization analysis similar to [12] is performed for every test vector pair in the test set. But instead of the entire sensitized portion of the circuit, only paths with a significant probability of causing a timing failure are extracted. The delay distribution of such a long path is approximately a normal distribution, regardless of the distribution of the gate and interconnect delays (central limit theorem). The circuit delay for the considered test set is now given by the statistical maximum of all path delays. The timing failure probability is finally obtained by evaluating the cumulative distribution function of the corresponding multivariate normal distribution.

The proposed algorithm approximates the probability of a timing failure in four major steps, as depicted in Fig 1. The first step identifies all paths, along which transitions travel from the primary inputs to the primary outputs under a given test set. Only those paths, whose probabilities of causing a timing failure exceed a user defined threshold, are extracted in the second step. The set of correlated random variables, defined by all critical transition path delays, is reduced in the third step with Clark's maximum estimation method [13]. The probability of a timing failure is finally computed in the last step, using an efficient numerical integration algorithm proposed by Genz [14]. The following subsections present a detailed description of all steps of the algorithm. A method to enhance its accuracy is introduced in subsection II-E.

A. Identification of Complete Transition Paths

A complete transition path is a sensitized path, along which a transition propagates from a primary input to a primary output. The identification of complete transition paths is based on a single pass event-driven timing simulation of each vector pair in the test set. This simulation considers only nominal delay values and may identify different complete transition paths for different delays, due to dynamic path sensitization and hazards. In rare cases, this dependency may have a large impact on the circuit delay, leading to complete transition paths, which are not representative for the vector pair.

Following the simulation of a vector pair, the complete transition paths can easily be identified by tracing the events backwards from the primary outputs to the primary inputs. To avoid any ambiguities during event tracing, each event at the output of a gate has a reference to its preceding event at the gate input. A further reference to the applied delay value, selected according to the conditions of the gate timing model, is stored to improve the efficiency of the following step.

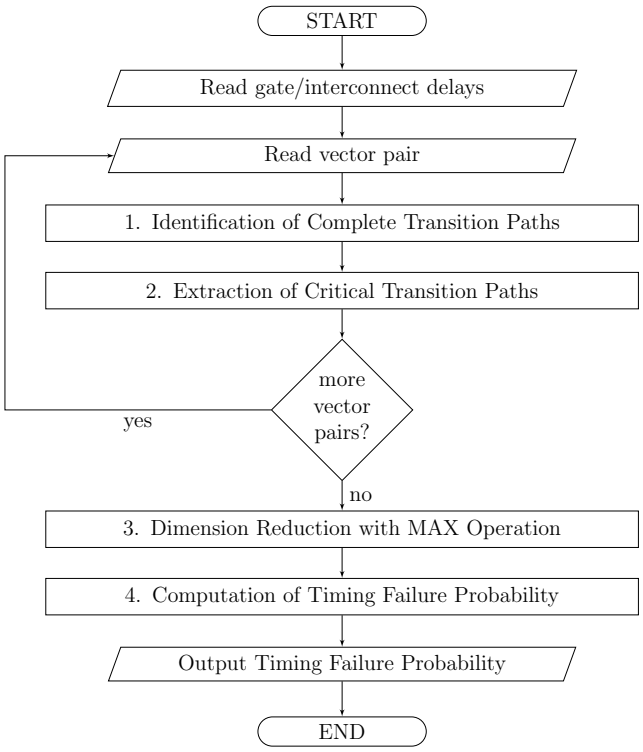

Fig. 1. Flowchart of the statistical dynamic timing analysis algorithm

B. Extraction of Critical Transition Paths

A critical transition path is defined as a complete transition path, for which the probability of the path delay exceeding the system clock cycle time T_{clk} is above a user defined threshold. If the probability is very large (e.g. greater 0.98), then a timing violation is very likely and no further analysis is required.

For a complete transition path with correlated and normally distributed gate and interconnect delays $X_1, \ldots, X_k$, the path delay Y also has a normal distribution with mean $\mu(Y)$ and variance $\sigma^2(Y)$ given by

$$\mu(Y) = \sum_{i=1}^{k} \mu(X_i) \qquad \sigma^2(Y) = \sum_{i=1}^{k} \sum_{j=1}^{k} \sigma(X_i, X_j), \quad (1)$$

where $\sigma(X_i, X_j)$ denotes the covariance of X_i and X_j. Assuming independent delays $X_1, \ldots, X_k$ of arbitrary distributions, the path delay distribution is obtained from the convolution of all probability density functions of $X_1, \ldots, X_k$.

The following steps of the algorithm require the delay of all critical transition paths to have a normal distribution. However, according to the central limit theorem, the distribution of a sum of independent random variables with arbitrary distribution converges to a normal distribution. In practice, this convergence occurs rapidly for less than 10 variables, especially if the distribution of these variables is close to a normal distribution [15]. While the central limit theorem only holds for independent random variables, for most practical models of correlation, the distribution is also guaranteed to converge to a normal distribution.

C. Dimension Reduction with Statistical Maximum Operation

The goal of the remaining two steps is to compute the probability Ψ, that the delay of at least one critical transition path exceeds the system clock cycle time T_{clk}. More formally, this probability is defined as

$$\Psi \triangleq 1 - \Pr(\max\{X_1, \ldots, X_n\} \leq T_{clk}), \tag{2}$$

where the critical transition path delays equal correlated random variables $X_1, \ldots, X_n$ of normal distribution.

The computation of the timing failure probability (2) requires knowledge about possible structural and spatial correlations between the critical transition path delays. For structural correlations, the covariance of two paths equals the sum of the variances of all gate and interconnect delays, which are shared by both paths. However, depending on the particular circuit model, the estimation of spatial correlations can be significantly more difficult and is beyond the scope of this paper. In this case, it is assumed that the covariance is obtained using one of the well-known methods.

Using Clark's approximation method [13], any two random variables X_i and X_j with $1 \leq i < j \leq n$ can be replaced by a new random variable $Y = \max\{X_i, X_j\}$, representing the statistical maximum of X_i and X_j. Following the definitions

$$\phi(x) \triangleq \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{3}$$

$$\Phi(x) \triangleq \int_{-\infty}^{x} \phi(t)dt \tag{4}$$

$$a \triangleq \sqrt{\sigma^2(X_i) + \sigma^2(X_j) - 2\sigma(X_i, X_j)} \tag{5}$$

$$\alpha \triangleq \frac{\mu(X_i) - \mu(X_j)}{a}, \tag{6}$$

the first two moments of Y are obtained by

$$\mu(Y) = \mu(X_i)\Phi(\alpha) + \mu(X_j)\Phi(-\alpha) + a\phi(\alpha) \tag{7}$$

$$\begin{aligned}\sigma^2(Y) = {} & \left(\mu^2(X_i) + \sigma^2(X_i)\right)\Phi(\alpha) \\ & + \left(\mu^2(X_j) + \sigma^2(X_j)\right)\Phi(-\alpha) \\ & + \left(\mu(X_i) + \mu(X_j)\right)a\phi(\alpha) - \mu^2(Y).\end{aligned} \tag{8}$$

Before removing X_i and X_j, it is necessary to compute all covariances $\sigma(X_k, Y)$, where X_k denotes any of the remaining random variables. The covariance of X_k and Y is given by

$$\sigma(Y, X_k) = \sigma(X_i, X_k)\Phi(\alpha) + \sigma(X_j, X_k)\Phi(-\alpha). \tag{9}$$

This approximation disregards all higher-order moments of Y, which introduces an approximation error. The accumulation of this error can be minimized by carefully choosing the order, in which the pairwise statistical maximum operations are performed [16]. In this paper, Clark's maximum estimation method is iteratively applied to the pair of random variables with maximum α. To avoid unnecessary error accumulation, this process stops once the number of random variables has dropped below a user defined threshold.

D. Computation of Timing Failure Probability

This step finally computes the timing failure probability Ψ using an efficient numerical integration method. Replacing the maximum in (2) yields

$$\Psi = 1 - \Pr(Y_1 \leq T_{clk}, \ldots, Y_m \leq T_{clk}), \tag{10}$$

where $Y_1, \ldots, Y_m$ denote the remaining random variables after the previous step. Instead of computing the entire distribution of the statistical maximum, only the cumulative distribution function must be evaluated. Using the definition of the cumulative distribution function this becomes

$$\Psi = 1 - \int_{-\infty}^{T_{clk}} \cdots \int_{-\infty}^{T_{clk}} \phi(x; \boldsymbol{\mu}, \Sigma)dx_1 ... dx_m, \tag{11}$$

where ϕ denotes the multivariate normal density function, defined by the mean vector $\boldsymbol{\mu} = [\mu(Y_1), \ldots, \mu(Y_m)]$ and the covariance matrix $\Sigma = [\sigma(Y_i, Y_j)]$ with $i, j \in \{1, \ldots, m\}$.

For this problem, the numerical integration algorithm proposed by Genz in [14] has proven to be surprisingly effective. It is applicable to very large dimensions and quickly converges to the required accuracy of about 10^{-4}. However, in some cases the dependencies between path delays may lead to a covariance matrix that is not positive-definite. To avoid severe numerical instabilities, various techniques can be applied to transform such a covariance matrix into a positive-definite matrix [17]. In this paper, the diagonal elements of the matrix are multiplied with a very small constant greater than unity.

E. Enhancement of Accuracy

Path sensitization and hazards can be very sensitive to delay variation. However, the sensitization analysis in the first step uses only nominal delay values, which is the main source of approximation error of this algorithm.

To explore the impact of delay variations on the composition of the critical transition path set, the algorithm can be executed N times using randomly chosen delays during sensitization analysis. If the approximation results $\Psi_1, \ldots, \Psi_N$ differ significantly, a classical Monte Carlo simulation of the whole circuit could be performed. However, a more efficient approach is to use the average approximation result $\bar{\Psi}$

$$\bar{\Psi} = \frac{1}{N} \sum_{i=1}^{N} \Psi_i, \tag{12}$$

where the calculation of the timing failure probability is only required for the distinct sets of critical transition paths. To focus on the most likely critical transition path sets, the standard deviation for the random number generation is reduced to 0.3σ, which was chosen based on experimental results.

However, the relative frequency and the delays of the critical transition paths are not necessarily independent as assumed by (12), which may lead to a slight underestimation of the timing failure probability. Indeed, the later a transition arrives at a gate input, the more likely the off-path inputs have already stabilized to their non-controlling values and allow the transition to be propagated.

III. Experimental Results

The experiments were performed on several NXP benchmark circuits. The circuits were first speed-optimized using a commercial synthesis tool and then mapped to the NanGate 45nm Open Cell Library [18]. To avoid an unnecessary complex experimental setup, chip layouts were not produced. As a result, interconnect delays and spatial correlations have been ignored. All experiments used the gate delay model defined by the Verilog HDL [19] standard. However, instead of real numbers, all delay values X were assumed to have a normal distribution with $\sigma(X) = c_v|\mu(X)|$ of the nominal value $\mu(X)$. The nominal gate delay values were extracted from the Standard Delay Format (SDF) description of the synthesized netlists. A variation coefficient of $c_v = 0.25$ was assumed, based on predictions for the 12nm process technology [20]. The system clock cycle time T_{clk} was chosen, such that 5% of the defect-free manufactured chips would fail due to timing failures caused by delay variations.

For each circuit, a benchmark of 20000 randomly chosen single gate delay faults was created. For every delay fault, a set of test vector pairs, suitable for small delay defects, was generated. Using only the nominal gate delay values, the k longest paths through a fault site were found with a commercial static timing analysis tool. The number of paths was limited to 1000, of which at most 100 paths were allowed to end at the same primary output. A commercial ATPG tool was later used to sensitize the resulting set of paths. Only delay faults, for which at least 20 vector pairs had been generated, were considered. The fault size was set to the slack of the complete transition path $\tilde{\pi}$ with the largest nominal delay, passing through the fault site.

For every delay fault, four different test sets $\Theta_1, \ldots, \Theta_4$ were defined as follows. The first test set Θ_1 contained only a single vector pair which sensitizes $\tilde{\pi}$ for nominal delay values. Then, Θ_{i+1} was created from Θ_i, by adding several vector pairs in decreasing order of the delay of the longest complete transition path. Thereby gradually extending the test set to five, ten and finally to twenty vector pairs.

The approximation results are compared to the fault detection probability, which is computed using 10^5 classical full circuit Monte Carlo simulations with a highly optimized event-driven timing simulator. During one iteration, a fault is said to be detected, if at least one primary output does not have the expected value at the sampling time T_{clk}. The runtime of the Monte Carlo simulations is dominated by the large number of random values required for every iteration. This is due to the detailed gate delay model defined by the Verilog HDL [19] standard, which distinguishes between different pin-to-pin, asymmetric rising/falling and conditional path delays. Hence, a two-input gate may require up to eight different delay values. The random number generation utilizes high-performance implementations of the Box-Muller transform and the Mersenne Twister pseudo-random number generator. All programs were implemented in C++ and executed on Intel Core i7-2600K processor workstations with 8 or 32GB RAM.

A. Proposed Approximation Algorithm

In this subsection, the effectiveness of the proposed approximation algorithm is demonstrated. The results for a single execution and for multiple executions are presented in Table I and II, respectively. Column (1) gives the name of the circuit, and column (2) shows the number of test vector pairs in the individual test sets Θ. All values in columns (3)-(8) represent average results over all delay faults. The average number of critical transition paths $|\Pi|$, obtained for the individual test sets, is shown in column (3). A complete transition path with normally distributed delay Y was considered critical, if $\mu(Y) + 3\sigma(Y) \geq T_{clk}$. The number of critical transition paths slowly saturated, because the additional vector pairs predominantly sensitized shorter paths and many paths may have already been sensitized by previous vector pairs.

The runtime and the accuracy of the proposed approximation algorithm were compared to 10^5 Monte Carlo simulations, which evaluated the fault detection probability as a golden reference. The absolute approximation error is presented in column (4) as the average absolute difference to the golden reference. As explained in section II-E, a single test vector pair might not sensitize a critical transition path for all possible delay realizations, due to different arrival times of the transitions at the off-path inputs. However, in this case, a different vector pair might still be able to sensitize the path. Hence, increasing the size of the test set leads to a reduction of the approximation error.

Clark's maximum estimation method was only applied in rare cases of more that 1000 critical transition paths. Hence, the impact on the average runtime and accuracy of the algorithm was very small. The multivariate normal integral (11) was computed using a FORTRAN routine named MVNDST, which was developed by Genz [14]. The largest estimated approximation error of the numerical integration algorithm was always less than 10^{-2}.

Column (5) shows the relative approximation error $\bar{\epsilon}$, defined as the average mean difference to the golden reference. A positive mean error $\bar{\epsilon}$ indicates overestimation, while a negative $\bar{\epsilon}$ shows a tendency for underestimation of the fault detection probability. A single execution of the proposed algorithm does not consider the impact of delay variations on the composition of the critical transition path set. Hence, the tendency shifts from overestimation for small test set sizes to underestimation for larger test set sizes.

As expected, the proposed extension of the algorithm improves the accuracy by computing the average result of 10 executions. For any delay fault, the number of executions was increased to 100 if the difference between the smallest and greatest timing failure probability exceeded 0.1. The fault detection probability was rarely overestimated, but contrary to a single execution, the mean error $\bar{\epsilon}$ for some circuits was increasing with the test set size $|\Theta|$. This is caused by statistical dependencies between the relative frequency of a critical transition path and its timing failure probability, as described in subsection II-E.

circuit	$\|\Theta\|$	$\|\Pi\|$	$\frac{\|\epsilon\|}{10^{-2}}$	$\frac{\bar{\epsilon}}{10^{-2}}$	t_{PA} [s]	t_{MC} [s]	S
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
p35k	1	9.01	6.43	0.27	0.0154	29.57	4347
	5	33.20	3.99	-0.82	0.0615	60.17	1592
	10	60.13	2.98	-0.66	0.1150	92.66	1176
	20	111.33	2.14	-0.50	0.2230	156.86	1046
p45k	1	4.98	4.54	1.89	0.0088	30.30	6486
	5	17.30	3.19	-0.36	0.0370	64.31	2666
	10	29.83	2.82	-0.90	0.0676	100.92	2004
	20	50.68	2.43	-1.13	0.1202	172.08	1853
p77k	1	6.81	2.81	1.42	0.0159	49.87	6320
	5	21.54	2.16	0.26	0.0591	117.93	3102
	10	35.08	2.09	0.03	0.1068	194.57	2501
	20	55.36	2.23	-0.20	0.1894	345.27	2352
p78k	1	3.93	7.68	2.58	0.0232	174.45	7754
	5	16.13	4.64	-0.30	0.1129	349.35	3158
	10	29.62	3.43	-0.64	0.2179	557.12	2592
	20	52.72	2.73	-0.90	0.4250	971.33	2313
p81k	1	2.90	4.21	0.89	0.0175	289.06	18307
	5	11.27	3.05	-0.53	0.0691	404.64	6357
	10	20.15	2.42	-0.62	0.1300	538.76	4404
	20	35.31	1.97	-0.65	0.2452	805.82	3450
p100k	1	4.63	5.51	1.29	0.0169	90.59	6409
	5	16.71	3.82	0.02	0.0712	193.44	3050
	10	29.29	3.12	-0.27	0.1340	309.98	2536
	20	49.74	2.75	-0.62	0.2502	540.13	2316
p267k	1	6.43	3.81	-0.67	0.0434	561.93	15055
	5	21.18	2.93	-1.33	0.1654	770.52	5335
	10	34.88	2.23	-0.96	0.3221	1001.76	3657
	20	57.45	1.77	-0.74	0.6819	1476.70	2803
p330k	1	11.85	5.51	1.35	0.0715	774.28	13316
	5	42.83	3.56	-0.40	0.2837	1109.19	5088
	10	69.30	2.78	-0.47	0.5871	1503.43	3689
	20	113.66	2.33	-0.50	2.0466	2299.52	2947

TABLE I
APPROXIMATION RESULTS FOR A SINGLE EXECUTION

circuit	$\|\Theta\|$	$\|\Pi\|$	$\frac{\|\epsilon\|}{10^{-2}}$	$\frac{\bar{\epsilon}}{10^{-2}}$	t_{PA} [s]	t_{MC} [s]	S
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
p35k	1	8.78	5.36	-1.84	0.1828	29.57	162
	5	32.74	3.44	-1.78	0.9910	60.17	61
	10	59.53	2.61	-1.29	1.7703	92.66	52
	20	110.58	1.90	-0.90	2.9965	156.86	52
p45k	1	4.90	3.47	0.30	0.1188	30.30	255
	5	17.06	2.78	-1.14	0.5864	64.31	110
	10	29.35	2.56	-1.45	1.0509	100.92	96
	20	49.95	2.27	-1.47	1.8789	172.08	92
p77k	1	6.70	2.09	0.56	0.1684	49.87	296
	5	21.28	1.70	-0.16	0.7546	117.93	156
	10	34.71	1.69	-0.27	1.6080	194.57	121
	20	54.83	1.89	-0.34	3.3099	345.27	104
p78k	1	3.71	6.04	-1.50	0.9253	174.45	189
	5	15.37	4.27	-2.60	5.2061	349.35	67
	10	28.29	3.33	-2.27	8.9644	557.12	62
	20	50.47	2.74	-2.05	14.4722	971.33	67
p81k	1	2.92	3.62	-0.25	0.3326	289.06	869
	5	11.33	2.75	-1.04	1.3735	404.64	295
	10	20.17	2.25	-0.98	2.2391	538.76	241
	20	35.24	1.89	-0.91	3.6286	805.82	222
p100k	1	4.48	4.59	-0.97	0.4047	90.59	224
	5	16.38	3.31	-1.25	1.9264	193.44	100
	10	28.58	2.80	-1.26	3.2609	309.98	95
	20	48.44	2.50	-1.33	5.5870	540.13	97
p267k	1	6.41	3.53	-1.64	0.7259	561.93	774
	5	21.26	2.72	-1.65	3.0388	770.52	254
	10	35.05	2.10	-1.23	5.0450	1001.76	199
	20	57.63	1.69	-0.97	9.2235	1476.70	160
p330k	1	11.76	4.43	-0.60	1.3944	774.28	555
	5	42.90	3.03	-1.22	6.1162	1109.19	181
	10	69.51	2.46	-1.06	11.3596	1503.43	132
	20	113.71	2.16	-0.93	30.9242	2299.52	74

TABLE II
APPROXIMATION RESULTS FOR MULTIPLE EXECUTIONS

Column (6) shows the average runtime for a single execution of the proposed algorithm. The runtime is dominated by the maximum estimation and numerical integration (41.8%), followed by the critical transition path extraction (34.6%) and event driven-simulation (23.6%). Clark's maximum estimation had only a minor impact on the average runtime.

The average runtime of the Monte Carlo simulation is presented in column (7) and the corresponding speed-up is given in column (8). The amount of time required for the event-driven simulation and path selection increases linearly with the test set size $|\Theta|$. However, due to the increasing number of critical transition paths, the relative contribution of the numerical integration to the overall runtime of the approach increases significantly. On the other hand, the Monte Carlo algorithm becomes more efficient, because the high cost for generating the large number of high quality random delay values is shared among a greater number of vector pairs. Hence, the speed-up decreases with the size of the test set. The runtime for multiple executions depends on the number of executions and distinct critical transition path sets.

The experimental results have shown an average speedup of between four and five orders of magnitude, compared to classical Monte Carlo simulations. The small approximation error of the algorithm is predominantly caused by the impact of delay variation on path sensitization and hazards. Multiple executions can further enhance the accuracy of the results.

B. Trade-Off Between Test Cost and Statistical Test Quality

In this subsection, the accuracy of the proposed approximation algorithm is demonstrated in the context of delay variation aware pattern selection for small delay defects. The fault detection probability is used to find a suitable compromise between the statistical test quality and the total number of vector pairs. Based on the results of the previous subsection, which were stored in a database, a suitable test set $\Theta \in \{\Theta_1, \ldots, \Theta_4\}$ is selected for each delay fault.

At first, the fault detection probability P_4 of the largest test set Θ_4 with 20 vector pairs was compared to the smaller subsets Θ_1, Θ_2 and Θ_3 with 1, 5 and 10 vector pairs, respectively. To minimize test cost, the smallest test set Θ_i with a fault detection probability P_i satisfying

$$P_i \geq P_4 - b \tag{13}$$

was selected. A threshold value of $b = 0.10$ was chosen to specify a trade-off between test quality and test cost. This process was repeated for all delay faults and circuits.

The described pattern selection approach was first performed using only the Monte Carlo simulation results. The test set size distribution over all delay faults in each circuit is presented in Fig. 2. For more than 50% of all delay faults in this example, five vector pairs provide superior fault detection probability, close to the four times larger test set Θ_4.

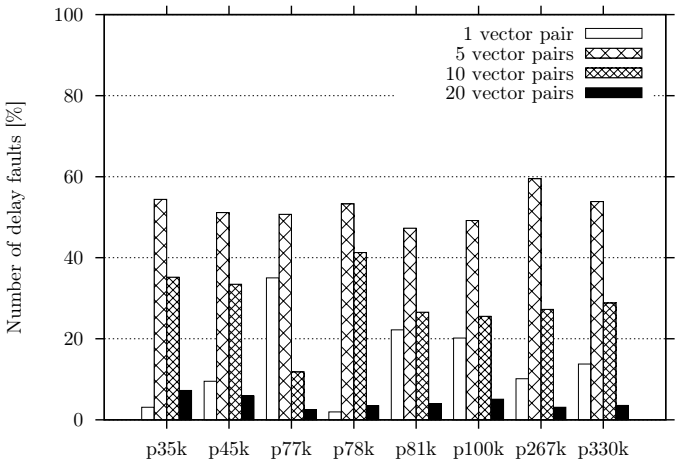

Fig. 2. Number of test vector pairs required for individual delay faults

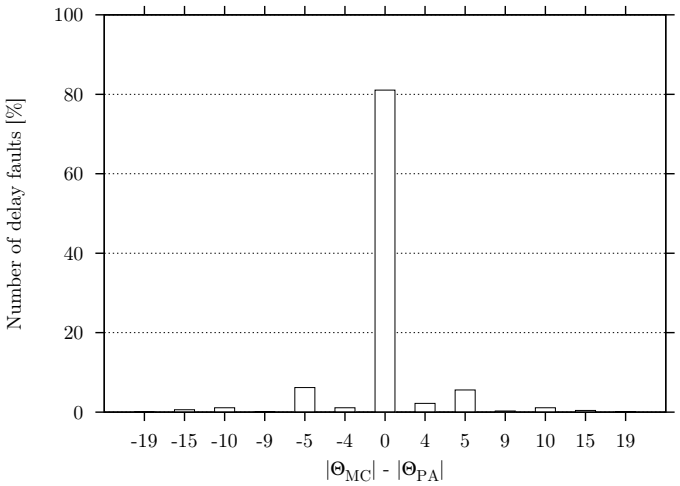

Fig. 3. Test set size differences compared to Monte Carlo based approach

The whole process was repeated using the proposed approximation algorithm, and both results were compared in Fig. 3. The abscissa shows the difference in the test set size $|\Theta_{MC}| - |\Theta_{PA}|$ over all delay faults and circuits, where Θ_{MC} and Θ_{PA} denote the test set chosen based on the results of the Monte Carlo simulation and the proposed approximation algorithm, respectively. The results match for more than 80% of all delay faults, while only for a small fraction a slightly smaller or larger test set was selected. By using the same randomly chosen delay values for all test sets, the average result of multiple executions provides even greater accuracy.

IV. Conclusion

Delay variations in recent technology nodes reduce the quality and reliability of all delay tests. Statistical test generation methods are guided by the fault detection probability to find a trade-off between statistical delay test quality and test cost. An efficient statistical dynamic timing analysis algorithm was proposed to reduce the high runtime complexity of these methods. The approach was compared with results of extensive Monte Carlo simulations and has shown a large speedup with only a small loss of accuracy.

V. Acknowledgement

This work has been supported by the German Research Foundation (DFG) under grant Wu245/5-2 (Realtest). The authors would like to thank NXP for providing the industrial circuits.

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)*, Jun. 2010, pp. 95–100.

[2] S. Kundu and A. Sreedhar, "Modeling manufacturing process variation for design and test," in *Proc. Design, Automation and Test in Europe (DATE)*, Mar. 2011, pp. 1–6.

[3] E. Yilmaz, S. Ozev, O. Sinanoglu, and P. Maxwell, "Adaptive testing: Conquering process variations," in *IEEE European Test Symp. (ETS)*, May 2012, pp. 1–6.

[4] U. Ingelsson, B. Al-Hashimi, S. Khursheed, S. Reddy, and P. Harrod, "Process variation-aware test for resistive bridges," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 8, pp. 1269–1274, Aug. 2009.

[5] C. Lin and S. Reddy, "On delay fault testing in logic circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 6, no. 5, pp. 694–703, Sep. 1987.

[6] M. Favalli and M. Dalpasso, "High quality test vectors for bridging faults in the presence of IC's parameters variations," in *Int. Symp. on Defect and Fault-Tolerance in VLSI Syst. (DFT)*, Sep. 2007, pp. 448–456.

[7] Z. Wang and D. M. Walker, "Compact delay test generation with a realistic low cost fault coverage metric," in *Proc. VLSI Test Symp. (VTS)*, May 2009, pp. 59–64.

[8] C.-T. C. Mango, L.-C. Wang, and K.-T. Cheng, "Pattern selection for testing of deep sub-micron timing defects," in *Proc. Design, Automation and Test in Europe (DATE)*, Feb. 2004, pp. 1060–1065.

[9] B. Lee, H. Li, L.-C. Wang, and M. S. Abadir, "Hazard-aware statistical timing simulation and its applications in screening frequency-dependent defects," in *Proc. Int. Test Conf. (ITC)*, Nov. 2005, pp. 91–100.

[10] U. Ingelsson and B. M. Al-Hashimi, "Investigation into voltage and process variation-aware manufacturing test," in *Proc. Int. Test Conf. (ITC)*, Sep. 2011, pp. 1–10.

[11] M. Yilmaz, K. Chakrabarty, and M. Tehranipoor, "Test-pattern grading and pattern selection for small-delay defects," in *Proc. VLSI Test Symp. (VTS)*, Apr. 2008, pp. 233–239.

[12] B. N. Lee, L.-C. Wang, and M. S. Abadir, "Reducing pattern delay variations for screening frequency dependent defects," in *Proc. VLSI Test Symp. (VTS)*, May 2005, pp. 153–160.

[13] C. E. Clark, "The greatest of a finite set of random variables," *Operations Research*, vol. 9, no. 2, pp. 145–162, Mar. 1961.

[14] A. Genz, "Numerical computation of multivariate normal probabilities," *Journal of Computational and Graphical Statistics*, vol. 1, no. 2, pp. 141–149, Jun. 1992.

[15] S. Louhichi, "Rates of convergence in the CLT for some weakly dependent random variables," *Theory of Probability and its Applications*, vol. 46, no. 2, pp. 297–315, Jul. 2002.

[16] D. Sinha, H. Zhou, and N. V. Shenoy, "Advances in computation of the maximum of a set of random variables," in *Proc. Int. Symp. on Quality Electronic Design (ISQED)*, Mar. 2006, pp. 306–311.

[17] W. Wothke, "Nonpositive definite matrices in structural modeling," in *Testing Structural Equation Models*. Sage Publications, Mar. 1993, pp. 256–293.

[18] "Nangate 45nm open cell library," Aug. 2011. [Online]. Available: http://si2.org/openeda.si2.org/projects/nangatelib/

[19] IEEE Standards Department, "Verilog hardware description language," *IEEE Std 1364-2005*, Apr. 2006.

[20] 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. 3-4, pp. 108–113, Oct. 2010.