

Longest Path Selection for Delay Test under Process Variation

Xiang Lu, Zhuo Li, Wangqi Qiu, D. M. H. Walker and Weiping Shi

Abstract—Under manufacturing process variation, a path through a net is called *longest* if there exists a process condition under which the path has the maximum delay among all paths through the net. There are often multiple longest paths for each net, due to different process conditions. In addition, a local defect, such as resistive open or a resistive bridge, increases the delay of the affected net. To detect delay faults due to local defects and process variation, it is necessary to test all longest paths through each net. Previous approaches to this problem were inefficient because of the large number of paths that are not longest.

This paper presents an efficient method to generate the set of longest paths for delay test under process variation. To capture both structural and process correlation between path delays, we use linear delay functions to express path delays under process variation. A novel technique is proposed to prune paths that are not longest, resulting in a significant reduction in the number of paths. In experiments on ISCAS circuits, our number of longest paths is 1% to 6% of the previous best approach, with 300X less running time.

Index Terms— Delay test, timing analysis, optimization, interconnect.

I. INTRODUCTION

Delay test of digital integrated circuits is to ensure that the signal from any primary input to any primary output is propagated in less time than the specification. A circuit is considered faulty if the delay of any path exceeds the specification. A delay increase due to a local defect, such as a resistive bridge or a resistive open, may cause a timing violation on the path through the defect, which is modeled as a delay fault [1][2]. Such a delay increase is localized to a gate input, output or an interconnect wire in the circuit, where the localized position is called a *local fault site* in this paper. Testing the longest path through the local fault site will capture

Manuscript received April 28, 2004. This work was supported in part by the SRC grant 000-TJ-844, NSF grants CCR-0098329, CCR-0113668, EIA-0223785, and ATP grant 512-0266-2001.

X. Lu is with Department of Electrical Engineering, Texas A&M University, College Station, Texas 77843, Email: xiang@ee.tamu.edu.

Z. Li is with Department of Electrical Engineering, Texas A&M University, College Station, Texas 77843, Email: zhuoli@ee.tamu.edu.

W. Qiu is with Department of Computer Science, Texas A&M University, College Station, Texas 77843, Email: wangqiq@cs.tamu.edu.

D. M. H. Walker is with Department of Computer Science, Texas A&M University, College Station, Texas 77843, Email: walker@cs.tamu.edu.

W. Shi is with Department of Electrical Engineering, Texas A&M University, College Station, Texas 77843, Email: wshi@ee.tamu.edu.

the delay increase due to the fault. When process variation is not considered, the problem of finding the longest and testable paths that cover all local fault sites has been extensively studied [3][4][5]. In these methods, only one path with the maximum delay is tested for each local fault site.

When process variation is considered, the path delay becomes a function of process variables. Among all paths through a fault site, there are often multiple paths whose delay can be the maximum under different process conditions [6]. For each fault site s , we call a path *longest* for s if the path has the maximum delay among all paths through s under some process conditions. On the other hand, we call a path *redundant* for s if the path can never be longest for s under any process condition. For example in Fig. 1, there are four longest paths, P_1 , P_2 , P_3 and P_4 , through a fault site in ISCAS85 circuit c432 using TSMC 180 nm technology. Two process variables, x_2 and x_3 , represent metal thickness variations of Metal 2 and Metal 3 respectively. In this example, four paths form the upper bound of the delay for all paths through the fault site within the range of process variation. Any path whose delay is below this bound is redundant for the fault site.

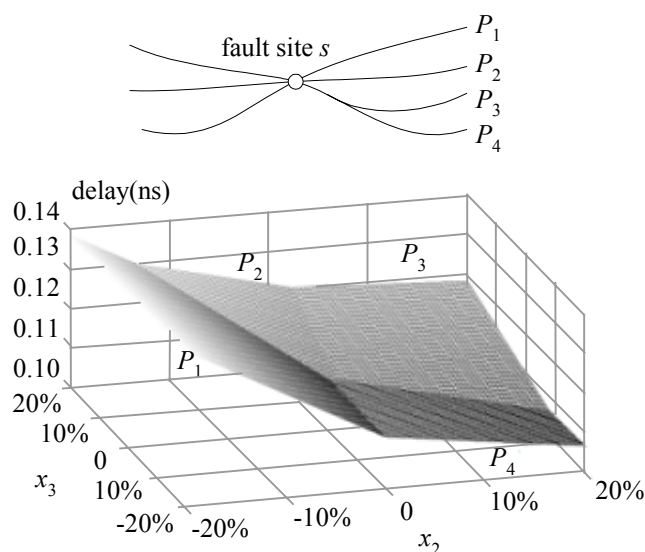


Fig. 1. Delay of four longest paths under process variation.

Traditionally, tests are only performed on the longest paths under the nominal or worst-case process condition. However, this might be insufficient. In Fig. 1, P_1 is the longest under the nominal process condition ($x_2 = 0$, $x_3 = 0$) and also the worst-case process condition ($x_2 = -20\%$ and $x_3 = 20\%$).

However, under process condition $x_2 = 20\%$ and $x_3 = -20\%$, P_1 is much shorter than P_4 . Since it is hard to know the actual process variation for a chip under test, we must test all paths that could be longest under any process condition. In this paper, we propose techniques to select all longest paths through each fault site to maximize the fault coverage.

Modern delay optimization tools tend to make many paths critical or near critical [7], resulting in too many paths for test. Pruning some of the paths based on structural correlation and process variation correlation is an effective approach to reduce the number of paths. If two paths share some nets or gates, there is a *structurally correlation* between them. Similarly, if two nets run on the same metal layer, there is a *process correlation* between them. Luong and Walker [8] proposed a pruning technique using both the structural correlation and the process correlation. As a result, they significantly reduced the number of paths. However, they only considered the longest paths for the entire circuit, instead of the longest paths through every local fault site. Furthermore, they did not consider interconnect delay. Tani *et al.* considered the longest paths through every local fault site [9]. They used a min-max comparison method, with the help of the structural correlation but not process correlation. As a result, their approach is overly pessimistic and produces too many paths. Liou *et al.* [10] used Monte Carlo simulation to select a set of critical paths that maximizes the probability of covering all critical paths under all process conditions. However, Monte Carlo simulation is very slow for large circuits and no running time is given for their method [10].

In this paper, we present a new method to select longest paths for each local fault site in the circuit. To maximize fault coverage, we want to find as many longest paths as possible. On the other hand, to minimize test costs, we want to find as few paths as possible. Given a set of testable paths, our method first models the path delay as a linear function of process variation variables, then uses two pruning algorithms to remove paths that are redundant or almost redundant. We repeat the process for each fault site in the circuit, and the remaining paths are longest paths for delay test. Experiments on the ISCAS circuits show that the new method is efficient and significantly reduces the number of paths for test, compared to the previous best method. We consider process variations of devices and interconnect in this paper, and the method can also be applied in path selection under operating variations of supply voltages and temperature [11].

This paper is organized as follows. The delay model is presented in Section II. The path pruning algorithms are described in Section III. Experimental results are shown in Section IV, and conclusions are drawn in Section V.

II. DELAY MODELING

There are many forms of process variation, see for example Stine *et al.* [12] and Nassif [13]. In this paper, we only consider die-to-die process variation, such as variations in gate length,

metal width, metal thickness, and inter-layer-dielectric (ILD) thickness. Our method can be extended to include other variations such as temperature and supply voltage, if the delay can be approximated by linear functions of the variations.

Let the *buffer-to-buffer* delay be the delay from an input pin of a cell to an input pin of a downstream cell. In this paper, we approximate each buffer-to-buffer delay as a linear function of process variation variables:

$$b(\mathbf{x}) = b_0 + b_1x_1 + b_2x_2 + \dots + b_px_p, \quad (1)$$

where $\mathbf{x}=(x_1, x_2, \dots, x_p)$ is a vector of process variation variables, each representing the deviation from the nominal value, b_0 is the nominal delay, and b_1, b_2, \dots, b_p are coefficients.

The validity of the linear model is supported by extensive simulation. After multiple parasitic extractions and circuit simulations under different process conditions, it is found that for any single process variation variable, its effect on delay is approximately linear within its variation range. In Fig. 2 we show circuit simulation results of a buffer-to-buffer delay for several typical process variation variables. The ranges of the variables are as follows: metal width $\pm 5\%$, metal thickness $\pm 20\%$, ILD thickness $\pm 40\%$, and gate length $\pm 5\%$.

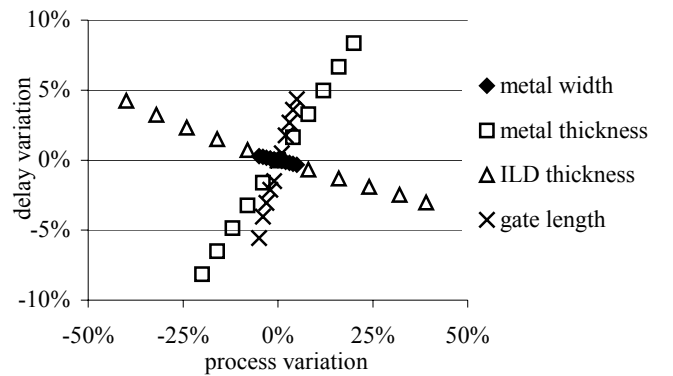


Fig. 2. Delay variations due to process variation are linear and symmetric as shown by parasitic extractions and circuit simulations.

In addition, since the process variables under consideration are determined at different stages of the manufacturing process, we can assume they are independent of each other. Furthermore, within the variation range of each variable, the delay effect is additive. For example, considering 5% of width variation on Metal 2 and Metal 3, we denote the delay variation under Metal 2 width variation and under Metal 3 width variation as Δd_{w_2} and Δd_{w_3} respectively, and denote delay variation under both variations as $\Delta d_{w_2+w_3}$. Then we use $\Delta d_{w_2} + \Delta d_{w_3}$ to approximate $\Delta d_{w_2+w_3}$. The error distribution is shown in Fig. 3 for 160 buffer-to-buffer delays in circuit c432. It is easy to see that the error is small for most buffer-to-buffer segments. The few large errors are due to non-zero cross-term coefficients in additive approximation, and usually occur with small $\Delta d_{w_2+w_3}$. Similar results are found for other process variables.

We compute coefficients in (1) using a delay sensitivity generation method [14]. The less efficient response surface method (RSM) [15] could also be used to determine

coefficients by sampling the output response. The nominal delay b_0 is computed by any commercial delay evaluation tool.

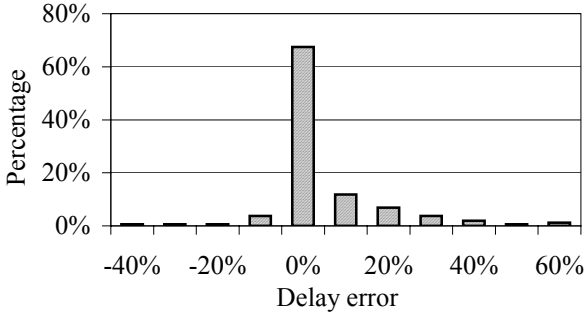


Fig. 3. The delay variation is approximately additive, which is demonstrated by the error distribution of approximating Δd_{n2+n3} with $\Delta d_{n2} + \Delta d_{n3}$ for over 160 buffer-to-buffer delays in circuit c432.

III. PATH PRUNING

Based on the linear delay model, the delay of a path can be derived as a linear function by accumulating all buffer-to-buffer delays defined in (1) along the path:

$$D(\mathbf{x}) = d_0 + d_1x_1 + d_2x_2 + \dots + d_px_p, \quad (2)$$

where d_0 is the nominal path delay, and d_1, d_2, \dots, d_p are coefficients for process variation variables.

Let $\mathbf{P} = \{P_1, P_2, \dots, P_n\}$ be a set of testable paths through a local fault site in the circuit, and let the delay of each path P_i be $D_i(\mathbf{x}) = d_{i0} + d_{i1}x_1 + \dots + d_{ip}x_p$. The range of all process variation variables is defined as $\mathbf{G} \subset \mathcal{R}^p$, where $\mathbf{G} = \{(x_1, \dots, x_p) \mid l_j \leq x_j \leq h_j, j=1, \dots, p\}$, and l_j and h_j are the lower bound and upper bound of x_j respectively. Then, any path P_q is a longest path in \mathbf{P} if and only if there exists $\mathbf{x}' \in \mathbf{G}$ such that:

$$D_q(\mathbf{x}') \geq D_1(\mathbf{x}'), D_2(\mathbf{x}'), \dots, D_n(\mathbf{x}'). \quad (3)$$

Verifying whether the set of inequalities (3) can be satisfied is known as the feasibility problem of linear programming (LP). When the dimension p is fixed, LP can be solved in $O(n)$ time [16]. However, the constant factor in the time complexity is exponential with p , resulting in high costs for large p . To reduce the running time in the case of large p , we replace LP with two heuristics. Heuristic H_1 prunes redundant paths, while Heuristic H_2 refines outputs of Heuristic H_1 to further reduce the number of paths for delay test.

Heuristic H_1

To determine if path P_q is longest, we define its *rough domain* of process variable x_k , with respect to path P_i as:

$$\begin{aligned} R_{ki} &= [l_{ki}, h_{ki}], \text{ where} \\ l_{ki} &= \min_{\mathbf{x} \in \mathbf{G}} \{x_k \mid D_q(\mathbf{x}) \geq D_i(\mathbf{x})\}, \\ h_{ki} &= \max_{\mathbf{x} \in \mathbf{G}} \{x_k \mid D_q(\mathbf{x}) \geq D_i(\mathbf{x})\}. \end{aligned}$$

Intuitively, R_{ik} specifies the possible values of x_k such that P_q is longer than P_i . The computation of l_{ki} and h_{ki} is straightforward.

The heuristic is as follows.

H_1 : Prune redundant paths

Input: path set \mathbf{P} , process range \mathbf{G}

```

1 For each path  $P_q \in \mathbf{P}$ , do
2   For each process variable  $x_k$ , do
3     Initial rough domain  $R_k = [l_k, h_k]$ .
4   For each path  $P_i, i=1, \dots, n, i \neq q$ , do
5     Compute the rough domain  $R_{ki}$ .
6     Update  $R_k = R_k \cap R_{ki}$ .
7   End
8   If  $R_k = \emptyset$ ,  $P_q$  is "redundant" and pruned.
9 End
10 End

```

Heuristic H_1 prunes path P_q if the intersection of rough domains of any process variable for P_q respect to other paths is empty. This is because if the intersection is empty, there does not exist any $\mathbf{x} \in \mathbf{G}$ such that P_q is the longest under process condition \mathbf{x} . Then according to the definition, P_q is redundant. The worst-case time complexity of the heuristic is $O(n^2p^2)$, since there are $O(n^2p)$ rough ranges, each takes $O(p)$ time to compute. Please note that although Heuristic H_1 prunes a large number of redundant paths, some redundant paths may escape when these paths are shorter than the combination of other paths.

Heuristic H_2

Among the longest paths, some paths are only slightly longer than others under every process condition. A path P_q is called *insignificant* if there is a longest path P_i such that their maximum delay difference is small:

$$\max_{\mathbf{x} \in \mathbf{G}} \{D_q(\mathbf{x}) - D_i(\mathbf{x})\} \leq \varepsilon D_i(\mathbf{0}),$$

where ε is a user-specified threshold and $D_i(\mathbf{0})$ is the nominal delay. If ε is small, say 1%, then testing P_q after testing P_i achieves little delay test coverage improvement. Therefore P_q should be pruned. The following heuristic prunes insignificant paths.

H_2 : Prune insignificant paths

Input: Path set \mathbf{P} , pre-specified threshold ε

```

1 For each path  $P_q \in \mathbf{P}$ , do
2   For any other path  $P_i \neq P_q$ , do
3     If  $\max_{\mathbf{x} \in \mathbf{G}} \{D_q(\mathbf{x}) - D_i(\mathbf{x})\} < \varepsilon D_i(\mathbf{0})$ 
4       Prune  $P_q$  as insignificant.
5   End
6 End

```

Heuristic H_2 compares the delay difference between each pair of paths under the worst-case process corners. The time complexity is $O(n^2p)$. We perform H_2 on the output of H_1 , and the remaining paths are kept for delay test.

IV. EXPERIMENT RESULTS

The experiments are performed for all ISCAS85 combinational circuits and the three largest ISCAS89 sequential circuits. Cadence Silicon Ensemble™ is used for circuit layout generation and parasitic extraction under TSMC

180 nm 1.8 V 5-metal technology. The heuristics are implemented in C on a 2.8 GHz Pentium 4 with 1 GB memory running at WindowsXP. The process variations considered are variations in transistor gate length, metal width, metal thickness and inter-layer-dielectric (ILD) thickness. There are a total of 16 variables for the 5-metal layer technology. The ranges of process variation variables are as follows: gate length $\pm 5\%$, metal width $\pm 5\%$, metal thickness $\pm 20\%$, and ILD thickness $\pm 40\%$. Under such variation ranges, path delays vary within $\pm 10\%$ in our experimental circuits.

We first compare the performance of our new method with the min-max method, which is the previous best method for the problem [9]. Considering path structural correlation, the min-max method first identifies shared gates between different paths and eliminates the delay of shared gates from path delays. Then min-max comparison is performed on remaining delays. In experiments of the min-max method, we use parameters $\alpha=1.0$ and $\beta=10\%$ to achieve a $\pm 10\%$ min-max delay range. In our new method, path structural correlation is implicitly considered in the formation of delay inequalities, and process correlation is handled by using the same set of variables in each delay function. Therefore, the new method is able to identify and prune more redundant paths than the min-max method does.

We assume the output of each cell in ISCAS85 circuits as a possible fault site. For sequential ISCAS89 circuits, we consider the combinational circuit between any pair of flip-flops and assume there can be a delay fault at the output of the driving flip-flop, as well as gate outputs. A path generator [17] [18] is used to provide critical and testable paths in batches of 50, where paths are indexed from 0 and sorted in the order of non-increasing nominal delay. For each batch we perform path pruning heuristics and collect remaining paths into the path set for test. Because of the non-increasing order of the nominal path delay, paths with greater indexes are less likely to be longest paths. Then if most paths in a batch are pruned, e.g. 45 out of 50 are pruned, which means the probability for the next batch to contain a longest path is small, the procedure stops. The stopping threshold is user-defined, and we used 90% in experiments. Although it is possible that a longest path exists in the following batch, the number of “escaped” paths is very small if we stop when 90% or more are pruned in a batch of 50 paths. In the experiments, the number of paths in the second batch is at most 1.79% of all longest paths we select, and no longest path is found in the third batch. The reason for this behavior is that path delay correlation is high enough that two paths of very different index are unlikely to both be longest and still pass heuristic H_2 . Delay test coverage will not be significantly degraded if only a small number of longest paths escape, since these paths will be only slightly longer than the tested paths. Luong and Walker [8] used a similar batch-based method to decide when to stop global longest path generation.

The comparison between our method and the min-max method is shown in Table I. In the table, column “# of critical paths” indicates the total number of paths through each fault site within 20% of the nominally longest path delay. In column

“paths for test” we list the total and the average number of longest paths for all fault sites in the circuit. The percentage of longest paths in the critical paths is shown in column “percentage”. The running time is shown in column “time (s)”, where the time for the path generator is not included. We do not list the result of the min-max method for circuit c6288 and larger circuits because the running time is more than several hours. As shown in the table, the number of longest paths selected by the new method is only 1%-6% of that selected by the min-max method. That indicates only a small percentage of paths are actually longest when structural and process correlations are used, compared to just using structural correlation in the min-max approach. The maximum average number of paths to be tested by the new method is 4.4. That means only a few paths need to be tested for each fault site. In addition, the new method is 300-3000 times faster than the min-max method. That is because the min-max method takes too much time identifying shared gates among paths. We used LP to verify the results of the new method and found that only 5% paths are pruned. That indicates that our method achieves close to the minimal test set at much lower cost.

As an example, the distribution of the path set size for all fault sites in circuit s38417, which has the largest average path set size, is shown in Fig. 4. The distribution indicates that no more than 8 paths must be tested for 90.0% of all fault sites.

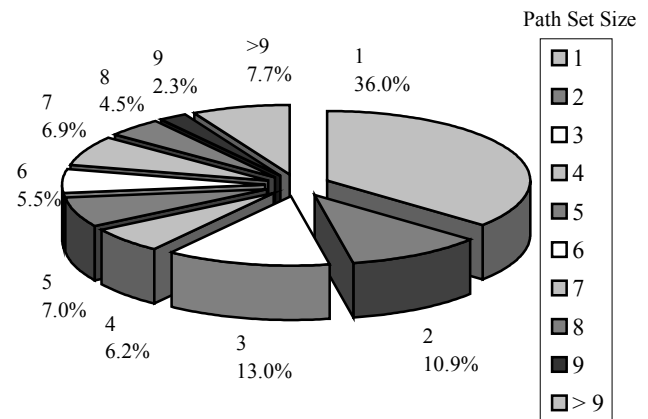


Fig. 4. Path set size distribution for fault sites in s38417.

The path selection distribution in circuit s38417 is shown in Fig. 5, where the X-axis indicates the path index, and the Y-axis indicates the percentage of local fault sites where the indexed path is selected. The path with index 0 is the longest under the nominal process condition and is selected for all fault sites. With increasing path index, the percentage selected decreases, as a path with lower nominal delay is less likely to be longest. The distribution also shows that, most of the longest paths are selected within the first batch of 50 paths, while few paths are selected in the second batch. We found that no paths were selected in the third batch. For most fault sites, the path selection procedure stops at the second path batch.

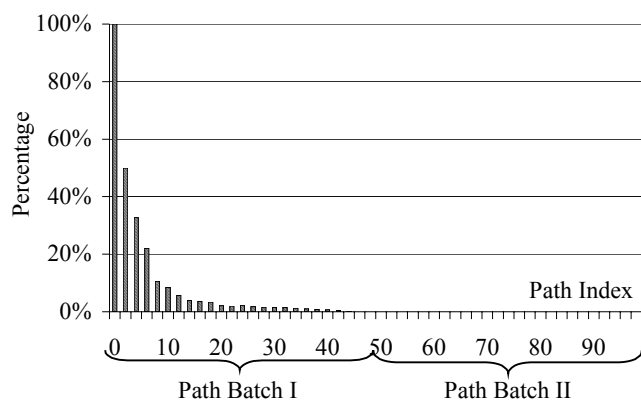


Fig. 5. Path distribution versus path indexes in s38417 using the new method.

To compare the efficiency between the two methods, in Fig. 6 and Fig. 7, we show the path selection distribution in circuit c432 using the min-max method and the new method respectively. As shown in Fig. 6, using the min-max method, the distribution goes to 0 after more than 150 paths are used for some fault sites, while for the new method the distribution goes to 0 after only 15 paths in Fig. 7.

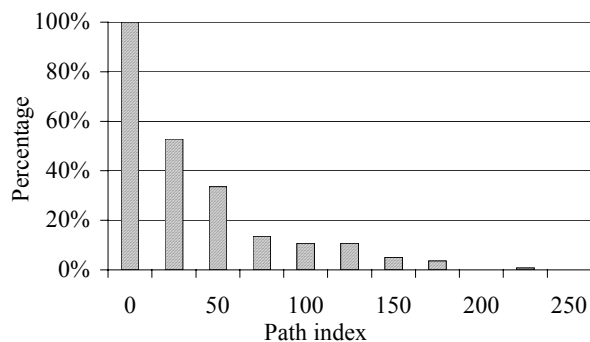


Fig. 6. Path distribution vs. path indexes in c432 using the min-max method.

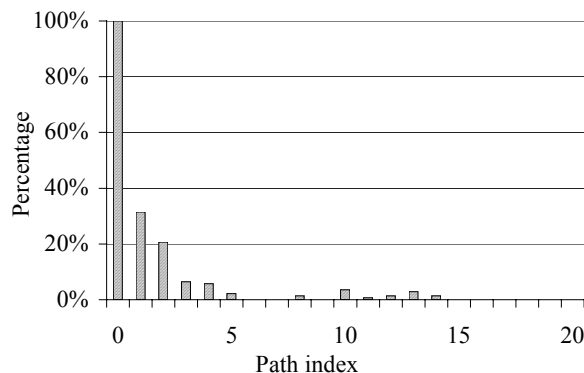


Fig. 7. Path distribution vs. path indexes in c432 using the new method.

V. CONCLUSIONS

In this paper we presented a novel and efficient method to find the set of longest paths for delay fault test under process variation. For the first time, we consider both path structural

correlation and process correlation, and consider process variation in both devices and interconnect. Two heuristics are proposed to prune redundant paths and insignificant paths. Experimental results show the heuristics are very efficient and effective. Our method can significantly reduce the number of paths and test patterns for delay test, compared with the previous best method. Experiments on ISCAS circuits show that the new method reduces the number of paths for test to 1%-6% of the results using the min-max method [9], without decreasing the fault coverage in delay test. The significant reduction indicates that considering both structural correlation and process correlation is much more effective than considering path structural correlation alone. In addition, the new method runs 300-3000 times faster than the min-max method, mainly because the min-max method examines far more paths.

The work described in this paper only considers die-to-die process variation. Systematic within-die variation, such as computed by lithography simulation tools, can be incorporated into the delay model, as it only affects the delay equation coefficients. Random within-die variation will be incorporated into the model in the future. This requires the addition of more process variables and a spatial correlation structure. The lower path correlation will result in more paths selected for testing. But these test sets will still be significantly smaller than the min-max test sets, which assume only structural correlation between path delays.

REFERENCES

- [1] R. R. Montanes, J. P. de Gyvez and P. Volf, "Resistance Characterization for Weak Open Defects," *IEEE Design & Test of Computers*, vol. 19, no. 5, Sep. 2002, pp. 18-25.
- [2] Z. Li, X. Lu, W. Qiu, W. Shi and H. Walker, "A Circuit Level Fault Model for Resistive Bridges," *ACM Trans. on Design Automation of Electronic Systems*, vol. 8, no. 4, Oct. 2003, pp. 546-559.
- [3] W. N. Li, S. M. Reddy and S. K. Sahni, "On Path Selection in Combinational Logic Circuits," *IEEE Trans. Computer-Aided Design*, vol. 8, no. 1, Jan. 1989, pp. 56-63.
- [4] Y. Shao, S. M. Reddy, I. Pomeranz and S. Kajihara, "On Selecting Testable Paths in Scan Designs," *IEEE European Test Workshop*, Corfu, Greece, May 2002, pp. 53-58.
- [5] M. Sharma and J. H. Patel, "Finding A Small Set of Longest Testable Paths That Cover Every Gate," *IEEE International Test Conf.*, Baltimore, MD, Oct. 2002, pp. 974-982.
- [6] M. Sivaraman and A. J. Strojwas, "Path Delay Fault Diagnosis and Coverage - A Metric and an Estimation Technique," *IEEE Trans. Computer-Aided Design*, vol. 20, no. 3, Mar. 2001, pp. 440-457.
- [7] T. W. Williams, B. Underwood and M. R. Mercer, "The Interdependence Between Delay Optimization of Synthesized Networks and Testing," *ACM/IEEE Design Automation Conf.*, San Francisco, CA, Jun. 1991, pp. 87-92.
- [8] G. M. Luong and D. M. H. Walker, "Test Generation for Global Delay faults," *IEEE International Test Conf.*, Washington, DC, Oct. 1996, pp. 433-442.
- [9] S. Tani, M. Teramoto, T. Fukazawa and K. Matsuhiro, "Efficient Path Selection for Delay Testing Based on Partial Path Evaluation," *IEEE VLSI Test Symp.*, Princeton, NJ, Apr. 1998, pp. 188-193.
- [10] J. J. Liou, A. Krstic, L. C. Wang and K. T. Cheng, "False-Path-Aware Statistical Timing Analysis and Efficient Path Selection for Delay Testing and Timing Validation," *ACM/IEEE Design Automation Conf.*, New Orleans, LA, Jun. 2002, pp. 566-569.

- [11] B. Seshadri, I. Pomeranz, S. M. Reddy and S. Kundu, "On Path Selection for Delay Fault Testing Considering Operating Conditions," *IEEE European Test Workshop*, Maastricht, Netherlands, May 2003, pp. 141-146.
- [12] B. Stine, D. Boning and J. Chung, "Analysis and Decomposition of Spatial Variation in Integrated Circuit Process and Devices," *IEEE Trans. Semiconductor Manufacturing*, vol. 10, no. 1, 1997, pp. 24-41.
- [13] S. R. Nassif, "Modeling and Analysis of Manufacturing Variations," *IEEE Custom Integrated Circuits Conf.*, San Diego, CA, May 2001, pp. 223-228.
- [14] X. Lu, Z. Li, W. Qiu, D. M. H. Walker and W. Shi, "PARADE: PARAMetric Delay Evaluation Under Process Variation," *IEEE International Symp. on Quality Electronic Design*, San Jose, CA, Mar. 2004, pp. 276-280.
- [15] A. R. Alvarez, B. L. Abdi, D. L. Young, H. D. Weed, J. Teplik and E. R. Herald, "Applications of Statistical Design and Response Surface Methods to Computed Aided VLSI Device Design," *ACM/IEEE Design Automation Conf.*, vol. 7, no. 2, Feb. 1988, pp. 272-287.
- [16] N. Megiddo, "Linear Programming in Linear Time When the Dimension is Fixed," *J. ACM*, vol. 31, no. 1, Jan. 1984, pp. 114-127.
- [17] W. Qiu and D. M. H. Walker, "An Efficient Algorithm for Finding the K Longest Testable Paths Through Each Gate in a Combinational Circuit," *IEEE International Test Conf.*, Charlotte, NC, Oct. 2003, pp. 592-601.
- [18] W. Qiu, J. Wang, D. M. H. Walker, D. Reddy, X. Lu, Z. Li, W. Shi and H. Balachandran, "K Longest Paths Per Gate (KLPG) Test Generation for Scan-Based Sequential Circuits," *IEEE International Test Conf.*, Charlotte, NC, Oct. 2004, pp. 223-231.

TABLE I
PERFORMANCE COMPARISON BETWEEN THE MIN-MAX PATH SELECTION METHOD AND THE NEW METHOD

circuits	# of fault sites	# of critical paths	min-max method				new method			
			paths for test		percentage (%)	time (s)	paths for test		percentage (%)	time (s)
			total	per fault site			total	per fault site		
c432	140	9263	5946	42.5	64.2	21.8	249	1.8	2.7	0.06
c499	202	5962	5384	26.7	90.3	47.1	204	1.0	3.4	0.09
c880	383	19641	9340	24.4	47.6	237.0	399	1.0	2.0	0.16
c1355	546	236771	185803	340.3	78.5	1070.5	598	1.1	0.2	0.30
c1908	845	136530	93061	110.1	68.2	1414.3	868	1.0	0.6	0.49
c2670	1246	80407	26095	20.9	32.5	1377.2	1253	1.0	1.6	0.70
c3540	1629	92617	30785	18.9	33.2	3431.2	1636	1.0	1.8	1.05
c5315	2278	129560	70394	30.9	54.3	2449.9	2312	1.0	1.8	1.19
c7552	3434	180045	87822	25.6	48.8	1872.8	3483	1.0	1.9	3.14
c6288	2384	760550	N/A	N/A	N/A	>3 hr	2384	1.0	0.3	6.35
s35932	17793	602412	N/A	N/A	N/A	N/A	18928	1.1	3.1	10.00
s38417	23815	705800	N/A	N/A	N/A	N/A	105383	4.4	14.9	40.55
s38584	20679	169821	N/A	N/A	N/A	N/A	72630	3.5	42.8	11.27