LAPS: Layout-Aware Path Selection for Post-Silicon Timing Characterization

Yu Hu††,†††, Member, Jing Ye††, Zhiping Shi††,†††, and Xiaowei Li††,†††, Nonmembers

SUMMARY Process variation has become prominent in the advanced CMOS technology, making the timing of fabricated circuits more uncertain. In this paper, we propose a Layout-Aware Path Selection (LAPS) technique to accurately estimate the circuit timing variation from a small set of paths. Three features of paths are considered during the path selection. Experiments conducted on benchmark circuits with process variation simulated with VARIUS show that, by selecting only hundreds of paths, the fitting errors of timing distribution are kept below 5.3% when both spatial correlated and spatial uncorrelated process variations exist.

key words: process variation, timing variation, sample, path selection, least square

1. Introduction

As the feature size in advanced VLSI technology continues to shrink, process variation produces more uncertainty in circuit timing behavior. Accurate timing characterization plays an important role in a variety of applications [1]–[6], such as statistical timing analysis, post-silicon tuning, post-silicon reliability analysis, and IC identification.

In general, process variation can be divided into two categories: correlated systematical variation and uncorrelated random variation. The correlated systematical variation tends to affect the closely placed gates or routed wires in a similar manner, making the gates and wires more likely to have similar process variations than those placed far apart [7], which is called the spatial correlation. On the other hand, the uncorrelated random variation means the random variation involves spatial uncorrelation. Transistor timing is heavily impacted by both the correlated systematical variation and uncorrelated random variation [8].

Figure 1-(1) shows the distribution of gate timing variations under spatially correlated process variations, generated by the VARIUS tool [9], where the deeper color represents higher timing variation. It can be seen that the timing variations of nearby gates are similar. Wires also have this kind of correlation [10], [11]. Therefore, timing characterization methods [12]–[21] usually first divide a circuit layout into grids to reduce the sample complexity, as shown in Fig. 1-(2) and Fig. 1-(3). When the grid is small enough, gates in a grid are considered to have the same timing variation. Then, by sampling the representative gates in grids or paths across grids, the timing variation of each grid can be profiled. The ways of sampling the timing information include (1) invasive measurement and (2) non-invasive testing.

For invasive measurement, on-chip monitors such as ring oscillator [12], NMOS/PMOS transistor chain [13] and slew-rate monitor [14] are inserted into grids to measure their timing. The advantage is that the timing information of grids can be directly measured. However, the more grids the layout is divided into, the higher hardware overhead it costs. Recent works [15]–[17] tried to solve the hardware overhead issue, but the accuracy of timing characterization is still more or less sacrificed.

Non-invasive testing leverages the paths in circuits to get the timing information. As the paths normally go across several grids, the path delay information must be post-processed by compressed sensing or least square techniques to obtain the timing of each grid [18]–[21]. Obviously, the accuracy of timing characterization is heavily dependent on the sampled paths. Path selection strategies have been well studied in delay fault testing works, but critical paths with long delays are the main concern. Without considering the distribution of paths on the circuit layout, these paths hardly reflect the whole timing distribution of the circuit.

In this paper, we propose a path selection method specifically for non-invasive timing characterization. This paper is an extension of our prior work in [21]. The proposed method has the following contributions:
• Sufficiency and uniformity of the sampled data for fitting timing variations are analyzed;
• A layout-aware path selection method named LAPS is proposed to sample paths for estimating the circuit timing variation.

The rest of the paper is organized as follows. Section 2 reviews the problem formulation of estimating the timing variation based on the sampled path delays. Section 3 analyzes the sufficiency and the uniformity of the sampled data, while Sect. 4 proposes the LAPS method. The experimental results are given in Sect. 5, followed by the conclusion section.

2. Problem Formulation

In this section, we will briefly review the problem formulation of estimating timing variation with sampled path delays [18], [21]. Taking the timing variation into consideration, the delay of a gate or a wire is represented by:

\[ D(K) = D^{\text{nom}}(K) \times v(K) \]  

(1)

where \( K \) represents a gate or a wire. \( D(K) \) represents the actual delay of \( K \). \( D^{\text{nom}}(K) \) represents the nominal delay of \( K \). \( v(K) \) represents the timing variation of \( K \).

If a path is a single-sensitized path, then its delay can be approximately considered as the accumulated delay of the gates and wires along the path [18]–[22]. Please notice that, different from the path selection in delay testing, we have no requirement on path length, so either long path or short path is feasible in our work.

Since the gates or wires nearby at layout have similar timing variations, their timing variations in the same layout grid are considered to be similar, especially when the grid is small. This approximation will introduce some fitting errors due to the uncorrelated random process variations, and we will show the fitting errors are still acceptable in the experiment section.

Because long wire can cross multiple grids, we divide the long wire into several segments so that every segment belongs to only one grid. The total delay of the wire is then considered as the sum of its segments. Different metal layers of a chip may have different timing variation distributions, so each layer will be accordingly divided into optimal number of grids.

For example, in Fig. 2, \( P_0 \) is a single-sensitized path when \( e = 1, f = 0, \) and \( g = 0 \). The gates \( H \) and \( I \) are both in the grid \( G_1 \), so they are approximately considered with the same timing variation \( v(H) = v(I) = v(G_1) \); \( J \) is in \( G_2 \) with \( v(J) = v(G_2) \). The wire \( b \) belongs to two grids so it is divided into \( b_1 \) and \( b_2 \); \( a \) and \( b_1 \) are both in \( G_1 \) with \( v(a) = v(b_1) = v(G_1) \); \( b_2 \), \( c \), and \( d \) are all in \( G_4 \) with \( v(b_2) = v(c) = v(d) = v(G_4) \). So the actual delay of \( P_0 \) is:

\[ D(P_0) = [D^{\text{nom}}(H) + D^{\text{nom}}(I)] \times v(G_1) \]
\[ + D^{\text{nom}}(J) \times v(G_2) \]
\[ + [D^{\text{nom}}(a) + D^{\text{nom}}(b_1)] \times v(G_3) \]

(2)

where \( D^{\text{nom}}(H) \) denotes the nominal delay of the gate \( H \).

Therefore, if the number of single-sensitized paths is \( N_{SSP} \), then the timing characterization problem can be formulated to the following equation set:

\[
D = \Pi \times V \\
V = \begin{bmatrix} v(G_1) \\ v(G_2) \\ \vdots \\ v(G_{N_{Grid}}) \end{bmatrix}
\]

\[
\Pi = \begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1N_{Grid}} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2N_{Grid}} \\ \vdots & \vdots & \cdots & \vdots \\ \pi_{N_{SSP}1} & \pi_{N_{SSP}2} & \cdots & \pi_{N_{SSP}N_{Grid}} \end{bmatrix}
\]

\[
K \in P_p \cap G_g
\]

(3)

where \( N_{Grid} \) is the total number of grids. \( K \) is a gate or wire which is located at the path \( P_p \) and also belongs to the grid \( G_g \).

In Eq. (3), as shown in [18]–[24], \( D \) can be collected using the Automatic Test Equipment (ATE) or Built-In Self Testing (BIST). Since we focus on how to select effective paths for characterizing timing variations, the testing approaches to obtain \( D \) will not be extended. Please refer to literatures [18]–[24] to find detailed explanations. As the measurement accuracy has an impact on the fitting errors, we have considered the measurement errors in the experiments. The parameter \( \Pi \) can be obtained by Electric Design Automation (EDA) tools. And \( V \) is the unknown timing variation distribution to be calculated.

Please notice that, some single-sensitized paths can propagate both a rising and a falling transition. As rising delay and falling delay of the same gate usually are different, each path can provide two equations. Thus in the rest of the paper, one such path is counted as two paths.

As Eq. (3) is formulated based on the assumption that the timing variations of gates in the same grid are similar, a solution \( V \) that satisfies every equation will not exist. Therefore, we use the method of least squares to calculate \( V \):

\[
\min ||\Pi \times V - D||_2^2
\]

(4)
3. Data Sampling

The proposed key idea is shown in Fig. 3. We will solve the above mentioned least square problem to characterize the timing variation distribution. As the quality of timing characterization is determined by the sampled data, this section will analyze the sufficiency and uniformity of the sampled data. Then in the next section, we will elaborate how to select the paths to obtain the sampled data.

The fitting error metric can evaluate the accuracy of the fitted timing distribution. The relative fitting error of a grid $g$, represented by $E(g)$, is:

$$
\bar{v}(g) = \frac{\sum v(k)}{N_{Gate}(g)} \quad k \in g \quad E(g) = \left| \frac{v_F(g) - \bar{v}(g)}{\bar{v}(g)} \right|
$$

(5)

where $N_{Gate}(g)$ represents the number of gates or wires in the grid $g$, and $k$ is a gate or wire in $g$, so $\bar{v}(g)$ represents the mean timing variations of the gates or wires in $g$, and $v_F(g)$ denotes the fitted timing variation of $g$.

As Eq. (4) contains $N_{Grid}$ unknowns, so at least $N_{Grid}$ paths should be selected for constructing $N_{Grid}$ equations, and they are expected to have no linear correlation, i.e. without redundant equations. According to the sampling theory [25], the sampled data should be sufficient and uniform. Sampling sufficiency means at least how many gates or wires in grids should be sampled, while sampling uniformity means the sampled gates or wires should be evenly distributed in the grids.

Obviously, to accurately fit timing variations, every grid should be sampled. In general, higher fitting accuracy can be achieved with more samples. Meanwhile, the decrease of fitting error will dramatically slow down after the number of samples reaches a point. Assume the number of data is $N$, and the data is in the Gaussian distribution with a standard deviation $\sigma$. To achieve an absolute fitting error $d$ with a confidence level $u$, at least $N_S$ samples should be randomly selected [25]:

$$
N_S = \frac{N \times u^2 \times \sigma^2}{N \times d^2 + u^2 \times \sigma^2}
$$

(6)

For example, assuming a grid contains $N = 100$ gates, and the $\sigma$ of their timing variations is 0.01, then to achieve the absolute fitting error $d = 0.01$ with the confidence level $u = 1.96$ (confidence coefficient 95%), at least $N_S = 4$ gates should be sampled. The theoretical analysis can help us to set the number of samples.

In regard to the sampling uniformity, to reflect the overall timing variation of the circuit, the sampled gates or wires are expected to be evenly distributed in the grids.

In summary, the selected paths are expected to meet the following constraints.

**C-1:** To meet the requirement of the Eq. (5), the selected paths must be single-sensitized paths;

**C-2:** For the sampling sufficiency, it is expected that at least $N_{Grid}$ paths should be selected to cover all the grids, and every grid is expected to be covered by different paths and through different gates or wires for reducing possibilities of constructing linearly correlated equations in the Eq. (4);

**C-3:** For the sampling uniformity, the gates or wires covered by the selected paths are expected evenly distributed in each grid.

Among them, C-1 is the strictest constraint that must be met, while C-2 and C-3 are loose constraint that may not be met completely, but our proposed LAPS method will try to achieve them as much as possible.

4. Layout Aware Path Selection

The pseudo-code of the LAPS method is given in Algorithm 1. To cover a grid, the main-function LAPS selects a gate located in the grid, and then calls the sub-function Select_One_PATH to generate a complete path across the gate. We will use the example in Fig. 4 to introduce the algorithm.

---

**Fig. 3** Overview of the proposed idea.

**Fig. 4** An example of LAPS.
Algorithm 1 LAPS

Void LAPS()
1 Define N_{Grid};
2 Define N_{SSP};
3 Set Num Selected Path = 0;
4 While (Num Selected Path < N_{SSP})
5 {
6 /Select one path
7 G_{r} := Select A Grid();
8 K_{r} := Select A Gate(G_{r});
9 Select an input K_{i} that can arrive at K_{r};
10 Set Success: Select One Path(K_{i}, K_{r});
11 If (Success Select)
12 Num Selected Path++;
13 }

Bool Select One Path(K_{i}, K_{r})
14 Set Success: Select = 0;
15 Initial a stack K_{Stack} = {};
16 Push K_{i} to K_{Stack};
17 While (K_{Stack} is not empty)
18 {
19 Pop a gate K_{j} from K_{Stack};
20 If (K_{j} is not an output)
21 Select and push K_{j}'s succeeding gates to K_{Stack};
22 Else
23 Break;
24 }
25 Success: Select = 1;
26 Break;
27 }
28 }
29 Return Success: Select;

Int Select A Grid()
30 For (g := G_{r} ∈ N_{Grid}++)
31 n_{G} := the number of selected paths that cover the grid g;
32 Min. C := the minimum value of array C;
33 For (g := G_{r} ∈ N_{Grid}++)
34 If (n_{G} = Min. C) Add g to G_{SSP};
35 g := randomly select a grid from G_{SSP};
36 Return g;

Int Select A Gate(int G_{r})
37 For (each gate g in G_{r})
38 n_{G} := the number of selected paths that cover the gate g;
39 Min. C := the minimum value of Array C;
40 For (each gate g in G_{r})
41 If (n_{G} = Min. C) Add g to G_{SSP};
42 If (G_{SSP} contains only one gate g) Return g;
43 Else
44 { 
45 For (each gate g in G_{SSP})
46 A(g) := the average layout coordinate of g and the selected gates in G_{r};
47 D(g) := the distance between A(g) and the center of G_{r};
48 }
49 Min. D := the minimum value of Array D;
50 For (each gate g in G_{SSP})
51 If (D(g) = Min. D) Add g to G_{SSP};
52 g := randomly select a grid from G_{SSP};
53 Return g;
54 }

For the sake of simplicity, only the transistor layout is illustrated in the figure. The layout is divided into 9 grids, and there is a selected path P_{1}.

Line-1 and Line-2 of Alg. 1 present two user-defined parameters N_{Grid} and N_{SSP}. From Line-6 to Line-12, one single path selection is tried each time until the total number of selected paths reaches N_{SSP}. If no more single-sensitized paths can be found, the function LAPS ends. Based on the constraint C-2, all the grids are expected to be covered by the selected paths, and the grids are expected to be covered as many times as possible through different gates. Thus, in Line-7, the LAPS starts from selecting a grid G_{r}, which has been covered by the fewest number of selected paths, and then we will try to select a path across the G_{r}. If more than one grid satisfies this requirement, then one of them is randomly selected. The sub-function Select A Grid is shown in Line-30 to Line-36. Next in Line-8, a gate K_{r} is selected. It belongs to the G_{r} and has been covered by the fewest number of selected paths. If more than one candidate gates are selected, then the constraint C-3 which considers the uniform distribution, is used to guide the further selection of K_{r} from them. Based on C-3, if a gate can make the average layout coordinate of selected gates closest to the center of its grid, the gate is selected as K_{r}. The sub-function Select A Gate is shown in Line-37 to Line-54. As K_{r} may not be an input, in Line-9, an input K_{i}, which can arrive at K_{r}, is selected by the similar strategy based on C-2 and C-3. The major difference between selecting K_{r} and selecting K_{i} is that, instead of analyzing every gate of G_{r} in Line-37 and Line-40, every gate which can arrive K_{r} is analyzed for selecting K_{i}.

The sub-function Select One Path is called in Line-10 selects a path that goes from K_{i} and through K_{r}. At the beginning, the path only contains an input gate K_{i}. Then its succeeding gates are gradually selected and connected to the path one by one until arriving at an output gate. The detailed steps are as follows. Firstly, to go through K_{r}, the succeeding gates that are not in the logic cone of K_{r} are removed. Secondly, the succeeding gates that cannot satisfy the constraint C-1 are removed. Finally, the constraints C-2 and C-3 are used to determine the order of the remaining succeeding gates to be pushed into the stack K_{Stack}. The K_{Stack} is used for backtracing when the currently selected partial path cannot satisfy the C-1. If no more gates are left in the K_{Stack}, then no single-sensitized paths are available, and this path selection fails. The failed paths will be recorded to avoid unnecessary try in the future. Otherwise, anytime the K_{s} is an output gate, the selection is success.

For example, in Fig. 4, at the beginning, both the grids G_{s} and G_{t} have not been covered by the path P_{1} yet, so we randomly select one grid from them: assuming G_{5} is selected as the G_{r}. Then, in G_{s}, since the gate C has not been covered yet, it is selected as the K_{r}. Two input gates A and B can arrive at the C. Both of them are in the same grid. In G_{4}, two other gates M and N have already been covered. The average layout coordinate of B, M, and N is closer to the center of G_{4} than that of A, M, and N, so B is selected as K_{r}. Now, K_{r} is C and K_{i} is B, so currently selected partial path is k \Rightarrow B \Rightarrow C \Rightarrow D. In this case, C has two fanout gates D and F. According to C-1, the partial path k \Rightarrow B \Rightarrow C \Rightarrow D is a single-sensitized path when b_{2} = 1, c_{2} = 1, and d_{1} = 1, while k \Rightarrow B \Rightarrow C \Rightarrow F is a single-sensitized path when b_{2} = 1, c_{2} = 1, and f_{2} = 0. Although both D and F satisfy C-1, the average layout coordinate of D and U is closer to the center of G_{8} than that of F and P to the
center of $G_2$. Hence $D$ will be selected as the gate $K$, in the next loop, and the currently selected partial path becomes $k \Rightarrow B \Rightarrow C \Rightarrow D$. The $D$ has only one succeeding gate $E$. The partial path $k \Rightarrow B \Rightarrow C \Rightarrow D \Rightarrow E$ is a single-sensitized path when $b_2 = 1$, $c_2 = 1$, $d_1 = 1$, and $e_1 = 0$. However, to make $e_1 = 0$, $i_1$ and $f_2$ must be 0 too, which conflicts $b_2 = 1$. Thus, $E$ does not satisfy the C-1. In this case, another partial path $k \Rightarrow B \Rightarrow C \Rightarrow F$ will be tried. Finally, the complete path $k \Rightarrow B \Rightarrow C \Rightarrow F \Rightarrow G \Rightarrow g$ is successfully selected.

5. Experimental Results

5.1 Experiment Flow

The experiment flow is shown in Fig. 5. First of all, commercial EDA tools are used to generate layouts and extract normalized delays of gates and wires for these benchmark circuits. In our experiments, four ISCAS’89 benchmark circuits s13207, s15850, s35932, and s38417 and two largest ISCAS’99 benchmark circuits b18 and b19 are used. We use TSMC 65nm technology library to conduct synthesis, placement and routing. For each benchmark circuit, 10$^6$ paths are randomly selected. On average, the gates contribute more than 99% of the total delay of a path, so for the benchmark circuits, the wires have a much smaller effect on path delays. Thus in this experiment, only the timing variations of gates are considered. It should be understood that in some other circuits, the wires can contribute significant amount of path delays. As explained before, the proposed method can also characterize their timing variation distributions.

Secondly, the VARIUS model [16] is used to simulate the process variations of transistor threshold voltage and channel length. In [16], for the spatially correlated process variations, a multivariate normal distribution with a spherical spatial correlation structure is applied. In the VARIUS, the phi, correlation distance relative to the grid width, is set to 0.5. The spatially uncorrelated process variations are generated with Gaussian distribution. The size of grid of VARIUS is set to let every gate in the circuit has its own variation.

Next, the proposed LAPS method is used to select the paths for timing characterization. An open-source tool [26] is adopted to check whether a partial path or a complete path is a single-sensitized path or not.

With the injected timing variation distribution, the timing variations of every gate are obtained, so the measurement of path delays ($D$ in Eq. (4)) can be simulated. Meanwhile, the matrix $H$ can be derived through the normalized delays of gates. With $D$ and $H$, the least square problem of Eq. (4) is then solved by a commercial mathematical tool to fit the timing variation distribution $V$. Finally, by comparing the injected timing variation distributions with the fitted distributions, the fitting error $E(g)$ is calculated for every grid $g$.

5.2 Selected Paths

The selected paths are expected to meet the three constraints C-1, C-2 and C-3, so $N_{Grid}$, $N_{SSP}$, and $R_{P/G} = N_{SSP}/N_{Grid}$ are set to different values according to the different scales of benchmark circuits. For ISCAS’89 benchmark circuits, all the grids can be covered. For ITC’99 benchmark circuits, when the number of grids is large, some grids are not covered, but the percentage of covered grids is still greater than 99%. The sampling sufficiency is shown in Table 1.

In Table 1, column D1 gives the average number of selected paths to cover a grid. For example, when $N_{Grid} = 100$ and $N_{SSP} = 100$, a grid is covered by 14.78 selected paths on average.

Column D2 gives the average number of selected paths to cover a gate. As explained before, if a single-sensitized path can propagate both a rising transition and a falling transition, this path is counted as two paths of $N_{SSP}$. When $N_{Grid}$ and $N_{SSP}$ are small, the selected paths indeed cover different gates in a grid. When $N_{Grid}$ and $N_{SSP}$ increase, as the total number of single-sensitized paths is limited by the circuit structure, the selected paths are more likely to cover the same gates. On average, when $N_{Grid} = 100$ and $N_{SSP} = 100$, a gate is only covered by 2.03 selected paths; when $N_{Grid} = 6400$ and $N_{SSP} = 19200$, a gate is covered by 4.68 paths.

The column D3 and D4 give the average number and the percentage of gates covered by the selected paths in a grid, respectively. For example, when $N_{Grid} = 100$ and $N_{SSP} = 100$, on average, 7.28 gates are covered in a grid. In general, Table 1 shows the selected paths achieve the expected sampling sufficiency required by C-2.

As for the third constraint C-3, the selected gates are expected uniformly distributed in each grid. Figure 6 illustrates the distributions of selected gates. For each circuit, every grid is divided into $5 \times 5 = 25$ sub-grids. The height of every column represents the accumulated number of selected gates in each sub-grid. We can see the sub-grids of a circuit contains a similar number of selected gates.

The runtime for selecting one path depends on the complexity of the circuit, so the runtime increases nearly linearly with the increasing of $N_{SSP}$. For the circuit s13207, it costs only less than 1 second to select a path, while for the circuit
Table 1  Sampling sufficiency of select paths

<table>
<thead>
<tr>
<th></th>
<th>s13207</th>
<th>s15850</th>
<th>s35932</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ngrid</td>
<td>Nssp</td>
<td>AN</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>200</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>10*10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10*10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10*10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>200</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>10*10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10*10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10*10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>200</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>10*10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10*10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10*10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b19, it costs about 15 seconds to select one path.

5.3 Fitting Errors

For simplicity, the following characters are used in the legends of subsequent figures:

\[ \Delta : \text{Delay under Process Variation} - \text{Normalized Delay} \]

\[ \frac{\text{Normalized Delay}}{\text{Normalized Delay}} \]  

MAX: Maximum  MIN: Minimum  AVG: Average  SD: Standard Deviation

First of all, the fitting errors are evaluated when only spatial correlated process variations are injected. The experimental results are shown in Fig. 7. Generally speaking, the fitting errors range from 1.72% to 4.36%. Under the same number of grids, with more selected paths, more gates in each grid are covered, so lower fitting errors are achieved. When \( N_{\text{grid}} = 100 \) and \( N_{\text{ssp}} = 100 \), the average fitting error is 3.18%; with \( N_{\text{ssp}} = 300 \), the average fitting error is 2.42%.

This data shows the proposed method can achieve high accuracy in confronting with spatial correlated process variations. In reality, spatial uncorrelated process variations also exist. Figure 8 illustrates the fitting errors, where the \( R_{\text{unc}} \) is used to represent the percentage of contributions of spatial uncorrelated process variations to the timing variations of gates. Generally speaking, when \( R_{\text{unc}} = 30\% \), the fitting errors range from 2.82% to 5.29%. Similarly, with more selected paths, lower fitting errors are achieved. As more spatial uncorrelated process variations may result in larger standard deviation of gate timing variations in each grid, same number of covered gates in a grid would result in a higher fitting error.

Besides of spatial uncorrelated process variations, in reality, measurement errors also exist. Though repeating measurement can reduce these errors [27], the delays of selected paths can still not be obtained accurately [16], [18], [27]. Hence, the fitting errors are further evaluated when measurement errors with Gaussian distribution (SD \( \approx 1.6\% \)) exist. Figure 9 illustrates the fitting errors when \( R_{\text{unc}} = 30\% \) of Fig. 8. With the increasing of the selected paths, the fitting errors decrease too. When \( R_{\text{RPG}} = 3.0 \), the average fitting error reduces to 5.47%. Roughly comparing with the previous work [18] whose average fitting error is 8.35%, the proposed method effectively improves the accuracy.
Fig. 7  Fitting errors for spatial correlated process variations.

Fig. 8  Fitting errors for spatial uncorrelated process variations ($N_{\text{Grid}}$ is 400 for s13207, s15850, s35932, and s38417; $N_{\text{Grid}}$ is 1600 for b18; $N_{\text{Grid}}$ is 3025 for b19)

Fig. 9  Fitting errors for measurement errors

6. Conclusion

This paper proposes a LAPS method for post-silicon timing characterization. We analyze the sufficiency and the uniformity of sampled data for effectively fitting timing variations. Then we select paths with consideration of single-sensitize, sampling sufficiency and sampling uniformity. Experiments on benchmark circuits show that, by selecting only hundreds of paths, we can keep the fitting errors of timing distribution below 4.4% when only spatial correlated process variations exist, below 5.3% when spatial uncorrelated process variations also exist, and below 8.2% when measurement errors are involved.
Acknowledgments

This work is supported in part by National Natural Science Foundation of China (NSFC) under grant No. 61532017, 61376043, and 61274030.

References


Yu Hu received the B.S., M.S., and Ph.D. degrees all in electrical engineering from University of Electronic Science and Technology of China (UESTC), in 1997, 1999, and 2003, respectively. She is currently a professor at the Institute of Computing Technology, Chinese Academy of Sciences. Her research interests generally include architectural-level and circuit-level reliable design and fault diagnosis techniques. She is a member of ACM, IEEE, IEICE, and CCF.

Jing Ye received the Ph.D degree in the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS) from University of Chinese Academy of Sciences (UCAS) in 2014. He received the B.S. degree in School of Electronics Engineering and Computer Science from Peking University in 2008. He is currently an assistant professor at the State Key Laboratory of Computer Architecture, ICT, CAS. His research interests generally include VLSI test, diagnosis, and hardware security.
Zhiping Shi  received the Ph.D. degree in computer software and theory from Institute of Computing Technology, Chinese Academy of Sciences, China, in 2005. Currently he is an associate professor at the Capital Normal University, Beijing, China. His research interests include formal verification and visual information analysis. He has been a research staff at Institute of Computing Technology, Chinese Academy of Sciences from 2005 to 2010.

Xiaowei Li  received the B.Eng. and M.Eng. degrees in computer science from Hefei University of Technology, China, in 1985 and 1988, respectively, and his Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), in 1991. From 1991 to 2000, he was an assistant professor and an associate professor (since 1993) in the Department of Computer Science and Technology, Peking University, China. He joined the ICT, CAS as a professor in 2000. He is now the deputy director of the State Key Lab. of Computer Architecture, ICT, CAS. Dr. Li’s research interests include VLSI testing, design for testability, design verification, dependable computing, and wireless sensor networks. Dr. Li serves as chair of Technical Committee on Fault Tolerant Computing, CCF (China Computer Federation) since 2008. He serves as vice chair of IEEE Asian Pacific Regional TTTC (Test Technology Technical Council) since 2004. He serves as the steering committee Chair of IEEE Asian Test Symposium since 2011. In addition, he serves on the Technical Program Committee of several IEEE and ACM conferences, including VTS, DATE, ASP-DAC, PRDC, etc. He also serves as member of editorial board of JCST, JOLPE, JETTA, etc.