Capturing Post-Silicon Variation by Layout-Aware Path-Delay Testing

Xiaolin Zhang[†‡] Jing Ye[†‡] Yu Hu[†*] Xiaowei Li[†]
[†]State Key Laboratory of Computer Architecture,
Institute of Computing Technology, Chinese Academy of Sciences
[‡]University of Chinese Academy of Sciences
{zhangxiaolin, yejing, huyu, lxw}@ict.ac.cn

***Abstract*—With aggressive device scaling, the impact of parameter variation is becoming more prominent, which results in the uncertainty of a chip's performance. Techniques that capture post-silicon variation by deploying on-chip monitors suffer from serious area overhead and low testing reliability, while techniques using non-invasion test are limited in small scale circuits. In this paper, a novel layout-aware post-silicon variation extraction method which is based on non-invasive path-delay test is proposed. The key technique of the proposed method is a novel layout-aware heuristic path selection algorithm which takes the spatial correlation and linear dependence between paths into consideration. Experimental results show that the proposed technique can obtain an accurate timing variation distribution with zero area overhead. Moreover, the test cost is much smaller than the existing non-invasion method.**

Keywords: variation extraction, path selection, path-delay testing, layout-aware

I. INTRODUCTION

As feature size shrinks to 65nm and below, parameter variation, including the deviation of process, voltage, and temperature (PVT), now poses even more serious challenges, which introduces inevitable and significant uncertainties in circuit timing distribution [1]. Hence, in recent years, the post-silicon variation extraction has drawn high attentions for its extensive applications, such as post-silicon tuning, fault diagnosis and post-silicon reliability prediction.

Accurate post-silicon timing characterization is not trivial because vast post-silicon measurements will introduce significant area overhead or test cost. Prior works on capturing post-silicon variation can be broadly classified into two categories. As Fig.1 shows, one is to deploy extra on-chip monitors to measure device timing [2] - [7], and the other one relies on non-invasive test [8] - [11].

The methods of deploying on-chip monitors exploit the spatial correlation characteristic of parameter variation, which implies that the parameters of neighboring transistors or wires are similar. Therefore the variation of monitors can represent the variation of the whole chip. This category can be further classified into two subcategories. One is to place ring oscillators [2]-[4], as Fig.1-(a) shows, and the other one is to place critical-path monitors (CPM) [5]-[7], as Fig.1-(b) shows. Although a ring oscillator can be placed in a spare area, hundreds even thousands of ring oscillators are needed for nowadays' large scale chips, which may result in non-trivial hardware overhead. The critical-path monitor can measure the effects of process, voltage, and temperature on timing. However, compared to the ring oscillator technique, the CPM technique generally incurs much higher area overhead.

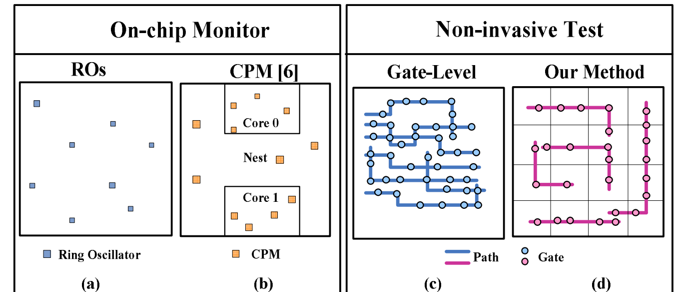


Figure 1. Comparison of the post-silicon variation extraction methods

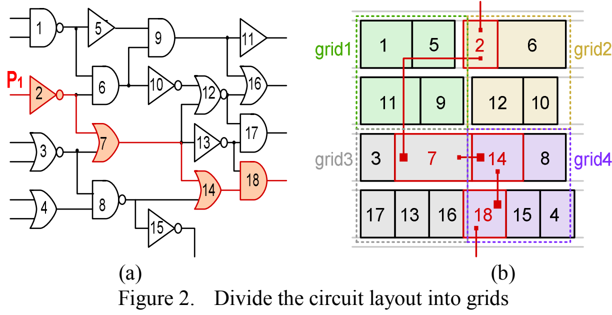
To alleviate the serious area overhead of on-chip monitors, gate-level timing characterization methods [8]-[11] have been proposed, as illustrated in Fig.1-(c). The basic idea of the gate-level post-silicon variation extraction technique is to transform the path delays to linear equations, and then calculate the timing variation of each gate. However, these methods confront a common problem that the number of the observable paths is less than the number of gates, hence the equations are indeterminate. The problem is much more serious in the large scale circuit, so the gate-level timing characterization is only available for small scale circuits. F. Koushanfar et.al [8] use *compressed sensing* [15] to address this issue, but they do not consider the uniformity of the selected gates, which reduces the accuracy of the post-silicon variation extraction. A. Gattiker, et.al. [10] [11] employ equality-constrained least squares to obtain the gate timing variation. Since the non-invasion test requires an expensive tester [16], it needs high test cost because it extracts a mass of paths indiscriminately.

In this paper, a novel method of capturing post-silicon variation by path-delay testing is developed, as Fig.1-(d) shows. The proposed method has three characteristics. 1) It does not need any on-chip monitors so there is no area overhead. 2) It uses layout-aware gate compressing so can significantly improve the accuracy of the extraction. 3) It takes the linear dependence of the selected path into consideration, so the test cost can be lowered by reducing the paths.

---

* Corresponding author: Yu HU, E-mail: huyu@ict.ac.cn
This work is supported in part by National Natural Science Foundation of China (NSFC) under grant No. 61076018 and 61274030, and in part by National Basic Research Program of China (973) under grant No. 2011CB302503.

978-3-9815370-0-0/DATE13/©2013 EDAA



(a) (b)

Figure 2. Divide the circuit layout into grids

II. MATHEMATICAL FORMULATION

Variations of transistor parameters such as channel length, transistor width, and oxide thickness could make the delay of gates and wires deviate from the normal value. Hence we can obtain the post-silicon variation by measuring the delay of wires and gates. For path P_k, its total path-delay D_k is the sum of the gate delay $d(g)$ and the wire delay $d(w)$, as Eq.(1) shows

$$D_k = \sum_i d(g_i) + \sum_j d(w_j) \quad (1)$$

where g_i and w_j represent an on-path gate and an on-path wire, respectively.

For the gate delay, $d(g_i^{norm})$ represents the normal gate delay of gate g_i, and v_i represents the timing variation ratio of gate g_i. Then the actual gate delay $d(g_i)$ can be expressed as

$$d(g_i) = (1+v_i) * d(g_i^{norm}) \quad (2)$$

For the wire delay, we make the widespread assumption [12] [13] that wire accounts for a fixed fraction of the total delay. For the path P_k, the total wire delay can be represented as $k_w * D_k$. Then Eq.(1) can be written as

$$D_k = \sum_i (1+v_i) * d(g_i^{norm}) + k_w * D_k \quad (3)$$

In this work, the circuit layout is partitioned into many grids. Owning to the spatial correlation of parameter variation, the timing variation in the same grid is assumed to be equivalent. This assumption will introduce some inherent errors, which will be analyzed in the following section. Let V_i represent the timing variation of $Grid_i$. When the circuit is partitioned into N grids, the *grid variation vector* V is

$$V = [V_1, V_2, V_3, \dots, V_{N-1}, V_N] \quad (4)$$

For a path, the normal gate delay in a grid is defined as the sum gate delay of the gates that are both on the path and in the grid. It is represented as $\pi_{i,k}^{norm}$, which can be calculated as follows

$$\Pi_{i,k}^{norm} = \sum d(g_m^{norm}) \qquad g_m \in Grid_i \cap P_k \quad (5)$$

where g_m is a gate which is on path P_k and in grid $Grid_i$. A vector G_k^{norm} represents the normal grid delay for path P_k

$$G_k^{norm} = [\pi_{1,k}^{norm}, \pi_{2,k}^{norm}, \pi_{3,k}^{norm}, \dots, \pi_{N-1,k}^{norm}, \pi_{N,k}^{norm}] \quad (6)$$

Thus, after layout partition, the delay of a path can be described by the accumulated delay of the path's segments that spread in each grid. For path P_k, Eq.(3) becomes

$$D_k = k_w * D_k + G_k^{norm} * (1+V^T) \quad (7)$$

where D_k^v represents the total gate delay deviation of the path P_k, and $D_k^{gate,norm}$ represents the sum of the normal gate delay on-path P_k. Based on Eq.(7), Eq.(8) can be represented as

$$D_k^v = D_k - w_k * D_k - D_k^{gate,norm} = G_k^{norm} * V^T \quad (8)$$

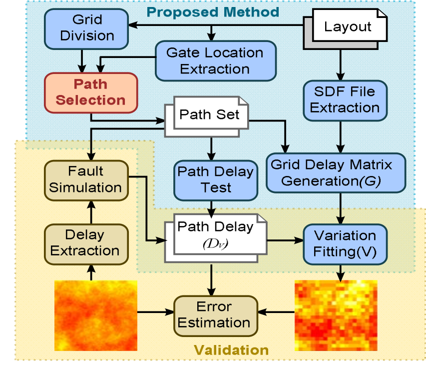


Figure 3. Overview of the proposed method

Consider a circuit netlist, as shown in Fig.2-(a), and its layout, as shown in Fig.2-(b). The layout is divided into 4 grids which are shown in different colors. Each small rectangle represents a standard logic cell, and the digital represents its index. For path P_1 which goes through g_2, g_7, g_{14} and g_{18}, we have $D_1^v = d(g_2^{norm}) * v_2 + d(g_7^{norm}) * v_7 + d(g_{14}^{norm}) * v_{14} + d(g_{18}^{norm}) * v_{18}$. Likewise, $D_1^v = \pi_{1,1}^{norm} * V_1 + \pi_{2,1}^{norm} * V_2 + \pi_{3,1}^{norm} * V_3 + \pi_{4,1}^{norm} * V_4$. Hereinto, $\pi_{1,1}^{norm} = 0$; $\pi_{2,1}^{norm} = d(g_2^{norm})$; $\pi_{3,1}^{norm} = d(g_7^{norm})$; $\pi_{4,1}^{norm} = d(g_{14}^{norm}) + d(g_{18}^{norm})$.

Because a gate has different normal delays on its rising and falling transitions, each path can provide two rows for Eq.(8). For a normal grid delay $\pi_{p,q,t}$, let p denote the index of the grid, q denote the index of the path, and t denote the type of transition (r: rising transition; f: failing transition). A matrix G given by Eq.(9) can represent the normal grid delay in N grids for M paths.

$$G = \begin{bmatrix} \pi_{1,1,r} & \pi_{2,1,r} & \dots & \pi_{N,1,r} \\ \pi_{1,2,r} & \pi_{2,2,r} & \dots & \pi_{N,2,r} \\ \dots & & & \dots \\ \pi_{1,M,r} & \pi_{2,M,r} & \dots & \pi_{N,M,r} \\ \pi_{1,1,f} & \pi_{2,1,f} & \dots & \pi_{N,1,f} \\ \pi_{1,2,f} & \pi_{2,2,f} & \dots & \pi_{N,2,f} \\ \dots & & & \dots \\ \pi_{1,M,f} & \pi_{2,M,f} & \dots & \pi_{N,M,f} \end{bmatrix} \quad (9)$$

The path-delay can be measured by a tester. According to Eq.(8), the total gate delay deviation D^v for M paths can be obtained as follows

$$D^v = [D_1^v, D_2^v, D_3^v, \dots, D_{M-1}^v, D_M^v] \quad (10)$$

Hence, according to Eq.(8), Eq.(11) can be got for M paths.

$$D^{vT} = G * V^T \quad (11)$$

Now, the *grid variation vector* V can be obtained by solving the least squares problem

$$\min \left\| G * V^T - D^{vT} \right\|_2^2 \quad (12)$$

III. THE PROPOSED METHOD

A. The overview of the proposed method

In this work, a novel layout-aware post-silicon variation extraction method by path-delay testing is proposed. An overview of the proposed method is shown in Fig.3. At first, the circuit layout is imported to extract the gate location and the standard delay format (SDF) file. In this step, the layout is divided into distributed grids evenly, and all gates are labeled by their container grids. Afterwards, testable paths are selected

according to a heuristic path search strategy which will be detailed in the following subsection. Next, the path-delay testing is carried out by a tester to measure the path-delay, from which D^v can be calculated by Eq.(8). Simultaneously, the *normal grid delay matrix* in Eq.(9) is generated. Finally, the variation is fitted by solving the least square problem as given in Eq.(12).

In order to validate the effectiveness of our method, the parameter variation is emulated by variation injection. The path-delay is obtained by path-delay simulation instead of path-delay testing by a tester. The fitting error is evaluated by comparing the injected and the extracted variation.

B. Path selection algorithm

To balance the test cost and the accuracy of variation fitting, a sufficient and small number of testable paths will be selected. Every time, a testable path is searched according to a heuristic strategy. Due to the uncertainty of the heuristic search, the searched path may be redundant. We consider a path is effective only if it can provide the unsearched gate. If a searched path is effective, it is added to the path set; otherwise, abandon it. The path selection procedure is terminated when the number of searched paths reaches the user-defined maximum path number.

Each path begins from the input gate. A path is searched by selecting a successor of the current gate. If the current gate has multiple successors, the search strategy will determines which one to be selected. The following metrics are used in the search strategy:

$Cover_{grid}^{x}$: a grid is covered by a path if the path has at least one gate which locates in the grid. The cover of a grid is the total number of gates which are not only in the grid but also on the selected paths, denoted by $Cover_{grid}^{x}$. A lower $Cover_{grid}^{x}$ implies a higher priority to select the grid. This is because a sufficient cover can improve the accuracy of variation fitting.

$Throughput_{gate}^{x}$: The throughput of a gate g_x is the amount of the selected path going through it. A lower $Throughput_{gate}^{x}$ implies a higher priority to select the gate, which boosts gate diversity. The accuracy of variation fitting could benefit from gate diversity.

$Direction_{grid}^{x}$: We define eight path arriving directions for each grid when searching a path, as shown in Fig.4. For example, the $Grid_{13}$ is in the east direction of the $Grid_{12}$, and the $Grid_{20}$ is in the south-west direction of the $Grid_{12}$. For each grid and each direction, the number of paths which have arrived in the grid from the same direction is recorded. The number of the paths arriving in a grid where a gate g_x locates in from the same direction is represented by $Direction_{grid}^{x}$. A lower $Direction_{grid}^{x}$ implies a higher priority to select the grid. That is because for timing variation fitting, an appropriate *normal grid delay matrix G* which contains enough linear independent rows is required. Since a row in the matrix corresponds to a path, two dependent rows mean one selected path is redundant. Thus, the path which may create the linear dependent rows should be avoided. In our experiments, we observe that, if the directions of paths are different, it is unlikely to create linear dependent rows in the matrix.

In the search strategy, a priority factor *f(x)* is used to determine which successor to be selected:

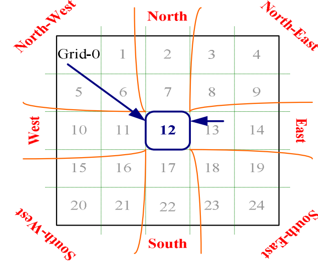


Figure 4. Solve the problem of linear dependence between the paths

$$f(x) = \frac{1}{\alpha * Cover_{grid}^{x} + \beta * Throughput_{gate}^{x} + \gamma * Direction_{grid}^{x}} \quad (13)$$

$$\alpha + \beta + \gamma = 1 \quad (14)$$

where α, β, and γ are user-defined weights. Based on a set of experiments, empirical values which balance the uniformity and the linear independence have been got, as α=0.4; β=0.3; γ=0.3. If the denominator of *f(x)* is 0, set the value of *f(x)* to a large constant, e.g. 1000. A larger *f(x)* indicates a higher priority for a successor *x* to be added to.

After adding a successor to the path-under-search, the testability of the current path or sub-path is checked by an incremental SAT solver [14]. If the current path is testable, continue to search the next gate; otherwise, remove the added gate and try to add the next-priority successor. If all the successors of the current gate are tried, but all produce untestable paths, then stop this search. Restart to search a path from the input gate. Whenever the path-under-search or the selected path set is updated, the cover, throughput and direction metrics should be updated.

IV. EXPERIMENTAL RESULTS

In order to validate the effectiveness of our method, the full-scan versions of larger ISCAS'89 and ITC'99 benchmarks with the SMIC 130nm and the SMIC 180nm libraries are used. A layout tool, Astro, produced by Synopsys Company is used. The path selection algorithm is implemented in C++ and the least square problem is solved by MATLAB. All the experiments are conducted on a 2.0GHz Linux work station with 32GB RAM.

The parameter variation is injected into circuits as [12] does, which sets δ/μ = 9%. Based on the spatial correlation, the variation in the same grid is regarded as identical, which introduces the inherent error. The inherent error is calculated as follows

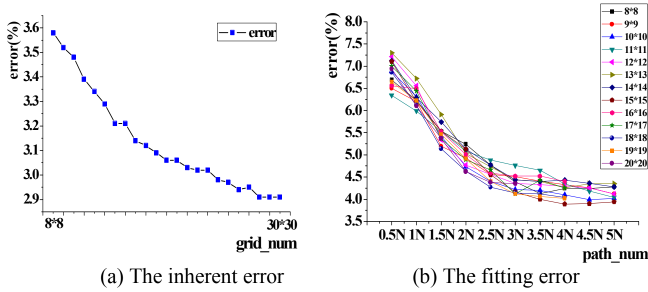
$$Error_{Inherent} = \frac{1}{N}\sum_{i} \underset{j}{mean}\left(\frac{|V_{i,avg} - v_{i,j}|}{v_{i,j}}\right) \quad (15)$$

where $V_{i,avg}$ is the average variation of $Grid_i$, and $v_{i,j}$ is the variation of the gate which locates in $Gird_i$. Similarly, the fitting error is calculated as follows

$$Error_{Fitting} = \frac{1}{N}\sum_{i} \underset{j}{mean}\left(\frac{|V_i - v_{i,j}|}{v_{i,j}}\right) \quad (16)$$

where V_i is the grid variation of $Grid_i$ which is fitted by the proposed method.

Fig.5 shows the inherent error and the fitting error of b22. In Fig.5-(a), the x-axis represents the number of grids and the y-axis represents the inherent error. It indicates the inherent error decreases with the number of grids increasing. In Fig.5-(b), the



(a) The inherent error (b) The fitting error

Figure 5. The inherent error and fitting error for b22

x-axis is the number of the selected paths, where N is the number of grids, while the y-axis shows the fitting error. Different numbers of grids are represented by different colors. The experimental results indicate the fitting error decreases with the amount of selected paths increasing. When the number of selected paths reaches to a certain amount, the fitting error keeps a relatively stable value. Considering the inherent error is about 3%, the suggested number of selected paths is 2.5N. On the other hand, as the number of division grids increases, the fitting error has a small decline as a whole, which is consistent with Fig.5-(a). Note, the strong spatial correlation gates may happen to be divided into two grids. That's why as grid_num increase, the fitting error may rebound.

Figure 6 shows the comparison of the fitting error between ICCAD'08[8] and our method. Our method not only expands the gate-level timing characterization of [8] to large scale circuits, but also obtains smaller fitting error than it. Explanations to the results are that: 1) considering linear dependent reduces the redundancy of paths; 2) considering uniformity of the selected gate ensures a good accuracy.

Table I shows the experimental results of 8 benchmarks. The column Patt_Num gives the number of the selected path. Compared with methods [10] [11], which extract all testable paths, our test cost is significant reduced. The *row rank* of the *normal grid delay matrix* is used to evaluate the redundancy of the selected paths. All benchmarks give G full rank or near full rank, which shows the selected path is essential. $Error_1$ and $Error_2$ are the fitting errors. Under column $Error_1$, we assume there is no path-delay measurement noise; while under column $Error_2$, the path-delay measurement noise is assumed to be 3%. Experimental results show the average errors of 8 benchmarks in these two scenarios are below 5% and 6% respectively. Considering the inherent error is about 3%, the accuracy of the proposed method is still very high. The last column shows the run time. The average run time is only 244 seconds.

V. CONCLUSIONS

A novel layout-aware post-silicon variation extraction method by path-delay testing has been presented. The heuristic path selection algorithm, which takes the uniformity of the selected gate and the linear independence of the selected paths into consideration, is the key technique of the proposed method. Experimental results have demonstrated that the proposed method can obtain an accurate variation distribution with zero hardware overhead and low cost test.

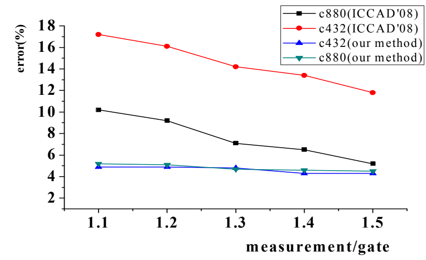


Figure 6. Comparison of the fitting error between ICCAD'08[8] and our method for c880 and c432

TABLE I THE ERROR ESTIMATION OF THE PROPOSED METHOD FOR ISCAS'89 AND ITC'99 BENCHMARKS

Circuit	Scale	Grid_Num	Path_ Num	Rank	Error1 (%)	Error2 (%)	Run_time
s9234	5619	12*12	2.5N	143	4.54	5.16	158.48
s15850	9873	14*14	2.5N	196	4.80	5.54	84.52
b20	9094	16*16	2.5N	256	5.05	5.75	197.41
b21	9520	16*16	2.5N	256	4.37	5.20	113.05
s35932	16385	20*20	2N	400	4.46	5.28	187.06
s38584	19711	20*20	2N	400	4.23	4.94	69.42
b17	23222	20*20	2N	395	5.66	6.36	850.26
b22	14557	20*20	2N	398	4.62	5.39	291.73
average					4.71	5.45	244

REFERENCES

[1] Semiconductor Industry Associate, International Technology Roadmap for Semiconductors, 2007.
[2] Milor, L., et.al. "Logic product speed evaluation and forecasting during the early phases of process technology development using ring oscillator data". In Workshop on SM, pp.20-23, 1997.
[3] Cheng Zhuo, et.al. "Active learning framework for post-silicon variation extraction and test cost reduction". In IEEE/ACM Proc. ICCAD, pp. 508 -515, 2010.
[4] X. Li, et.al. "Virtual probe: a statistically optimal framework for minimum-cost silicon characterization of nanoscale integrated circuits". In IEEE/ACM Proc. ICCAD, pp. 433-440, 2009.
[5] A. Drake, et.al. "A distributed critical-path timing monitor for a 65 nm high-performance microprocessor," In IEEE/ACM Proc. ISSCC, pp. 398-399, 2007.
[6] Qunzeng Liu, et.al. "Capturing Post-Silicon Variation Using a Representative Critical Path," In IEEE Trans.CAD, Vol.29, Iss.2, pp. 211-222, 2009.
[7] Songwei Pei, Huawei Li, Xiaowei Li, "A High-Precision On-Chip Path Delay Measurement Architecture," In IEEE Trans.VLSI, Vol.20, Iss.9, pp.1565-1577, 2012.
[8] F.Koushanfar, et.al. "Post silicon timing characterization by compressed sensing," In IEEE/ACM Proc. ICCAD, pp.158-189, 2008.
[9] Yousra Alkabani, et.al. "Trusted Integrated Circuits: A Nondestructive Hidden Characteristics Extraction Approach," In IEEE Lecture Notes in Computer Science, Vol.5284, pp.102-117, 2008.
[10] Gattiker, A. et.al. "Efficient and product-representative timing model validation," In IEEE Proc. VTS, pp.90-95, 2011.
[11] Eun Jung Jang, et.al. "Post-Silicon Timing Validation Method Using Path Delay Measurements," In IEEE Proc. ATS, pp. 232-237, 2011.
[12] Sarangi, S.R. et.al. "VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects," In IEEE Trans. SM, Vol.21, Iss.1, pp.3-13, 2008.
[13] Eric Humenay, et.al. Impact of Parameter Variation on Multi-Core Chips. In Workshop on ASGI, 2006.
[14] J. Kim, et.al. "On applying incremental satisfiability to delay fault testing," In IEEE Proc. DATE, pp. 380-384, 2000.
[15] E. Candes, et.al. "Compressive sampling," In ACM Proc. ICM, pp.50-66, 2006
[16] K. Agarwal et.al. "Characterizing process variation in nanometer CMOS", In IEEE/ACM Proc. DAC, pp.396-399, 2007