Design Dependent Process Monitoring for Wafer Manufacturing and Test Cost Reduction

Tuck-Boon Chan, Aashish Pant, Lerong Cheng, and Puneet Gupta

***Abstract*—Short-loop process monitoring structures (usually simple device I-V, C-V measurements made after M1 fabrication) are commonly put in wafer scribe-lines. These test structures are almost always design independent and measured/monitored by the foundry to keep track of process deviations. We propose a design-dependent process monitoring strategy which can accurately predict design performance based on $I_{\rm eff}$-based delay and $I_{\rm off}$-based leakage power estimates. Further, we use the predicted delay and power for early yield estimation to (1) prune bad wafers to save test and back-end manufacturing costs, and (2) prune bad dies to save test costs. Combining chip pruning with wafer pruning, we can reduce the cost per good chip by up to 13%. Such design-dependent process monitoring can help reduce process optimization effort, enable quicker yield ramp besides saving test and manufacturing costs.**

I. Introduction

Process variation has been a critical aspect of semiconductor manufacturing [19]. When new process technologies are introduced, process variation causes manufactured chips to exhibit wide performance spread [3], and wafer yield could be as low as 30% to 50% [35]. Although screening defective chips after manufacturing can reduce burn-in, testing, and packaging costs [32], the chips until this point have already incurred unnecessary manufacturing cost. Thus, it is beneficial to prune bad wafers and chips during early stages of manufacturing wherever possible using low-cost tests.

Early wafer pruning has been introduced in [24], where *cost-of-yield* (COY) is defined as a metric to guide the decision of pruning or scrapping a wafer in production. Based on a comprehensive cost analysis on wafer pruning, Wu et al. propose a genetic algorithm for making a wafer lot pruning decision [35]. These wafer pruning strategies do not address the problem of estimating chip performance and consequent parametric yield at early wafer manufacturing stages for wafer-level pruning.

Mitra et al. in [25] show an example of early chip performance estimation by using ring oscillator's (RO) delay as a measure of chip performance. This method relies on the correlation between RO and the chip's critical paths, which is inherently inaccurate as every critical path has a different sensitivity to process variation. Since inaccurate chip performance estimations may lead to wrong pruning decisions, it is necessary to have an accurate design-dependent process monitoring method. Meanwhile, the monitoring structures should be placed in the wafer scribeline to minimize the measurement cost and silicon area overhead. Though ring oscillator guided testing strategies are common [18] [12], we have not seen any work dealing with designing scribeline ring oscillators which are design specific.

To capture design-specific performance variation, authors in [21] propose a framework to estimate chip performance with post-silicon measurement. This method assumes that the process variation distribution and correlation among the variation sources are given. Alternatively, Cho et al. in [10] propose to train a neural network for chip performance prediction using data collected during manufacturing. The accuracy of the estimation is strongly related to the training data. For both methods, the required process information and training data are usually not available or inaccurate as process parameters are varying.

Design-specific monitors have been proposed in [22] [14] [31]. However, these monitors are not suitable for low-cost scribeline-based test for several reasons. Scribeline test structures are designed and tested by the foundry using a probe card; using customized test-structures and testing procedures will increase cost and manufacturing complexity. Also, the monitoring circuits may be too large to fit into the scribeline, which has limited area. Another disadvantage of using on-chip monitors (e.g., [21] [22] [14] [31]) is that probing on-chip monitors at an early manufacturing step will introduce defective particles around the monitor, which will reduce wafer yield. Using scribeline structures pose a lower risk of introducing defective particles because probing is not applied on the chip directly.

In this paper, we propose a design-dependent monitoring approach using commonly used compact scribeline test structures (e.g., those in [20]). These test structures are generic and capable of measuring the following parameters after the Metal-1 stage of manufacturing[1]:

$$\begin{aligned}
I_h &= I_{ds} \text{ at } V_{gs} = V_{dd}, && V_{ds} = V_{dd}/2\\
I_l &= I_{ds} \text{ at } V_{gs} = V_{dd}/2, && V_{ds} = V_{dd}\\
I_{\rm off} &= I_{ds} \text{ at } V_{gs} = 0, && V_{ds} = V_{dd}\\
C_{\rm gate} & \quad\ \ \text{ at } V_{gs} = V_{dd}, && V_d = V_s = 0
\end{aligned}$$

A preliminary version of this work appeared in [7].

P. Gupta is with the Department of Electrical Engineering, University of California Los Angeles, CA, 90095 (e-mail: puneet@ee.ucla.edu), T.-B. Chan is with the Department of Electrical and Computer Engineering, University of California San Diego, CA 92093 (e-mail: tuckie@ee.ucla.edu), L. Cheng is with SanDisk Corp. (e-mail:lerong.cheng@sandisk.com) and A. Pant is with Mentor Graphics Corp. (e-mail:Aashish_Pant@mentor.com)

[1] The bias points match commonly used measurements on scribeline process control monitoring test circuits in commercial foundries.

where V_{dd}, V_{gs}, V_{ds}, V_d and V_s are supply, gate-to-source, drain-to-source, drain and source voltages, respectively. I_h, I_l and I_{off} are drain-to-source current (I_{ds}) of a CMOS device (NMOS or PMOS) at the corresponding bias conditions and C_{gate} is gate capacitance of a device. Based on the measured values of I_h and I_l, we can represent circuit delay with effective drive current (I_{eff}), defined as [26]

$$I_{\text{eff}} = \frac{I_h + I_l}{2} \tag{1}$$

We estimate the design-specific delay and leakage power to changes in I_{eff} and I_{off} at the early stage of wafer manufacturing. Based on the estimated timing and leakage power of every chip, a wafer and chip pruning decision can be made for manufacturing cost reduction. The overview of our approach is depicted in Figure 1.

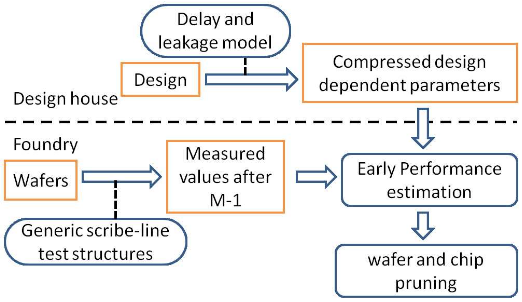

Delay and leakage model
Design
Compressed design dependent parameters
Design house
Foundry
Wafers
Measured values after M-1
Early Performance estimation
Generic scribe-line test structures
wafer and chip pruning

Fig. 1. In this wafer and chip pruning flow, the design house extracts design-dependent sensitivities of chip delay and leakage power to changes in I_{eff} and I_{off} based on our delay and leakage power models (described in Section II and III). Extracted sensitivities are compressed and transferred to the foundry. During manufacturing, the foundry will fabricate the design and generic scribeline test structures. Based on the design-dependent sensitivity data from the design house and measurements from the test structures, the foundry can estimate chip performance at an early manufacturing stage and make a pruning decision.

Our contributions are the following:

- We propose a scribeline-based design-dependent approach for chip performance and leakage power estimations.
- We analyze within die variation and measurement noise effects in the chip performance estimations.
- We show how the above information can be used to accurately identify bad wafers and help in wafer pruning and yield estimation.
- Using the estimated chip delays, we show that bad die can be readily identified and pruned from the testing lot, to save on costly tester time at wafer sort.

Rest of this paper is organized as follows. In Section II, we discuss our I_{eff} based path delay estimation model. In Section III, we describe our I_{off} based leakage power estimation model. In Section IV, we describe how our analysis can be used for early wafer and chip pruning. In Section V, we present the results using our detailed wafer level simulation setup. We conclude in section VI.

II. Delay Estimation Using I_{eff}

In this paper, we model chip delay using I_{eff}, which is defined as the average current that charges or discharges a circuit node during a logic transition[2]. The delay of a logic transition is modeled as

$$delay \propto \frac{CV}{I_{\text{eff}}} \tag{2}$$

where C is the node capacitance that is being charged (or discharged), V is the voltage swing and I_{eff} is the effective drive current. While I_{eff} cannot be physically measured, several works propose approximations using device level *I-V* characteristics [1], [15], [26]. Though more complex models (e.g. [1]) can be used as well, our experiments indicate that (1) suffices for our device models and libraries.

A. Cell Delay Model

Using (2), we can express the delay of a cell as

$$d_{cell}(c) = \sum_{t \in T} \frac{K_{cell}(c,t)CV}{I_{\text{eff}}(t)}$$

where $K_{cell}(c,t)$ is delay scaling coefficient, c denotes the cell type (e.g., INV, NAND etc), t denotes device type, T is the set of all device types and C is node capacitance that is being charged (or discharged) by the cell[3]. $K_{cell}(c,t)$ is fitted for different input slew, output load and transition combinations. This fact is implicit and we do not show it for notational convenience.

Expanding d_{cell} using Taylor series with respect to $I_{\text{eff}}(t)$ for all $t \in T$ and ignoring the cubic and higher order terms, we get

$$\begin{aligned} d_{cell}(c) &= d_{cell-nom}(c) \\ &- \sum_{t \in T} \frac{K_{cell}(c,t)CV}{I_{\text{eff}-\text{nom}}(t)} \left(\frac{\Delta I_{\text{eff}}(t)}{I_{\text{eff}-\text{nom}}(t)} - \frac{\Delta I^2_{\text{eff}}(t)}{2I^2_{\text{eff}-\text{nom}}(t)} \right) \end{aligned} \tag{3}$$

where $d_{cell-nom}$ is the delay and $I_{\text{eff}-\text{nom}}(t)$ is the $I_{\text{eff}}(t)$ of a cell at nominal process conditions. $\Delta I_{\text{eff}}(t)$ is the $I_{\text{eff}}(t)$ change due to process variations. $K_{cell}(c,t)$ are fitted for every cell using (3) by varying process conditions for different input slew and output load points. This model fitting can be done very efficiently as it can use existing process specific timing libraries which are available for various corners. In our experiments, we do not have access to a sufficient number of these libraries. Therefore, we fit the model using SPICE simulations on individual cells.

B. Path Delay Model

The delay of path j under process variations can be expressed as

$$d_{path}(j) = d_{path-nom}(j) + \Delta d_{path}(j)$$

[2] If scribeline measurements for electrical parameters such as V_{th}, channel length, electron mobility, etc., are available, our delay model can be modified to incorporate the impact of these parameters to improve delay estimation.

[3] In this work, we take four device types into account: $\{$high V_{th}, low $V_{th}\} \times \{$PMOS, NMOS$\}$. Standard cells made by the same device type have two non-zero $K_{cell}(c,t)$ coefficients.

where $d_{path-nom}(j)$ refers to nominal delay of path j. $\Delta d_{path}(j)$ is the delay change due to process variation, which is equal to the sum of delay changes of every cell in the path,

$$\Delta d_{path}(j) =$$
$$-\sum_{i\in G_j}\sum_{t\in T}\frac{K_{cell}(i,t)C(i)V}{I_{\text{eff}-\text{nom}}(t)}\left(\frac{\Delta I_{\text{eff}}(t)}{I_{\text{eff}-\text{nom}}(t)}-\frac{\Delta I^2_{\text{eff}}(t)}{2I^2_{\text{eff}-\text{nom}}(t)}\right)$$

where G_j is the set of cell instances on path j. Due to process-induced variation on slew and load, K_{cell} may differ from its value extracted during design time. To evaluate the process-induced variation on K_{cell}, we simulate standard cells with 1000 randomly sampled process conditions based on the variation model in Table II. We then extract the input slew and output capacitance of the standard cell and calculate its K_{cell} based on the proposed delay model. Results of this study show that standard deviation of K_{cell} (average of INV, NOR2 and NAND2 gates) is 6.0%. Although our model does not capture the process-induced K_{cell} variation, error induced by K_{cell} variation is included in our experiments.

The sensitivity of delay of path j to changes in $I_{\text{eff}}(t)$ can be expressed as[4]

$$K_{path}(j,t) = \sum_{i\in G_j} K_{cell}(i,t)C(i) \tag{4}$$

where $C(i)$ is the node capacitance for cell instance i. The total path delay can now be written as

$$d_{path}(j) = d_{path-nom}(j) - \sum_{t\in T}\frac{K_{path}(j,t)V}{I_{\text{eff}-\text{nom}}(t)}\left(\frac{\Delta I_{\text{eff}}(t)}{I_{\text{eff}-\text{nom}}(t)}-\frac{\Delta I^2_{\text{eff}}(t)}{2I^2_{\text{eff}-\text{nom}}(t)}\right) \tag{5}$$

C. Handling Load Capacitance Variation

In (4), the path specific delay sensitivities to I_{eff} depend on the nominal value of output load, which is seen by the cells. However, with process variations, this output load also changes. Therefore we scale the estimated delay by the ratio of actual device capacitance to nominal capacitance.

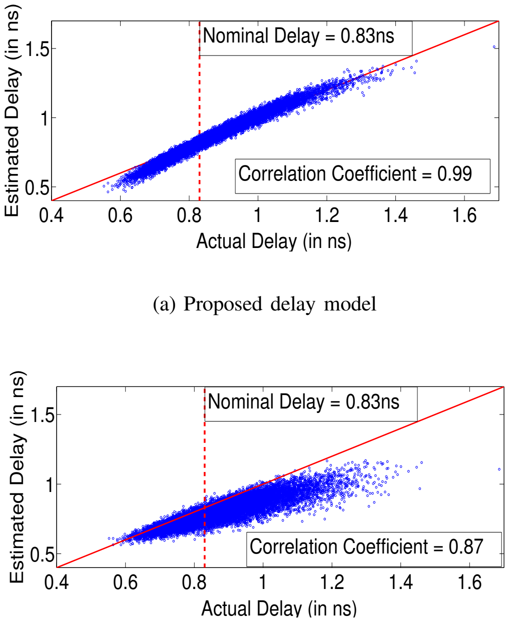
$$d'_{path}(j) = (d_{path}(j) - d_{path-interconnect}(j))\frac{C_{\text{gate}}}{C_{nom}} + d_{path-interconnect}(j) \tag{6}$$

where $d'_{path}(j)$ is the scaled delay estimation and $d_{path-interconnect}(j)$ is the interconnect delay of a path[5]. C_{gate} is the process variation affected capacitance (measured by scribe-line monitors) and C_{nom} is its nominal value.

Figure 2 shows the accuracy of the proposed design-dependent delay estimation technique using (6), compared to a design-independent approach. In this experiment, we randomly generate 1000 process condition samples for the variation model in Section V-B (without within die variation). We then characterize timing libraries for all standard cells (using SPICE) at the sampled process conditions to calculate the worst-case (actual) delay of the C432 ISCAS85 benchmark circuit using static timing analysis. Meanwhile we also extract I_{eff} and I_{off} of the PMOS and NMOS devices at the same process conditions using the SPICE simulator. After that, we apply (6) to obtain delay estimations for the proposed delay model. Since a design-independent delay estimation has no information about the circuit, we assume the design-independent approach equally weights all device types and calculate path delay as follows.

$$d_{path-indep}(j) = d_{path-nom}(j)\sum_{t\in T}\frac{I_{\text{eff}}(t)}{I_{\text{eff}}(t)+\Delta I_{\text{eff}}(t)} \tag{7}$$

where $d_{path-indep}(j)$ is the path delay estimated by a design-independent approach. The result shows that the proposed delay estimation tracks the actual delay well. The correlation coefficient is found to be 0.99, compared to 0.87 for the design independent approach. This is because the design independent methodology is oblivious of the exact nature, topology and the structure of the cells that make up the critical paths in the design, while our strategy effectively captures this dependence in the $K_{path}(j,t)$ form.



(a) Proposed delay model

(b) Design independent model

Fig. 2. Scatter plot (C432 Monte-Carlo timing simulations) that shows how the delay estimated by (a) proposed delay model, and (b) a design independent approach, compared with actual delay for an ISCAS85 C432 benchmark, obtained from static timing analysis with timing tables characterized at the sampled process conditions.

D. Effect of Within Die Variation on Delay

Intra-die variation is being captured by scribe-line test structures available next to each die. However, measurements from test structures are typically different from the ones on critical paths due to within die variation. We express the within

[4] $K_{path}(j,t)$ is instance-dependent as input slew and output load may vary with instance.

[5] In this work, interconnect delays extracted from our benchmark designs are much smaller compared to cell delays. For simplicity, we scale the entire path delay by the ratio of actual device capacitance to nominal capacitance in our experiments.

die variation as a normally distributed random variable with zero mean and standard deviation, $\mathcal{N}(0, \sigma_{wd})$. The distribution can be estimated by making multiple measurements per die[6]. Considering only the first order term in (5), the path delay vector can be rewritten in matrix form as

$$\mathbf{D} = \begin{bmatrix} d'_{path}(1) \\ \vdots \\ d'_{path}(z) \end{bmatrix} + \mathbf{W}\mathbf{I}_{wd}, \quad \mathbf{W} = \begin{bmatrix} w_{11} \dots w_{1n} \\ \vdots \ddots \vdots \\ w_{z1} \dots w_{zn} \end{bmatrix}$$

$$w_{ji} = \begin{cases} K_{cell}(i,t) & \text{if cell } i \text{ is on path } j \\ 0 & else \end{cases}$$

where z is the total number of paths, n is the total number of cell instances and $\mathbf{I}_{wd}$ represents the within die I_{eff} variation. $\mathbf{W}$ is a parameter that describes dependencies between critical paths and $\mathbf{I}_{wd}$. Every entry in $\mathbf{I}_{wd}$ is an independent Gaussian random variable, with zero mean and standard deviation σ_{wd}. Due to large numbers of critical paths and cell instances, keeping the entire covariance matrix on test machines is not practical. To reduce the size of $\mathbf{W}$, we extract and use its v largest principal components (PC). This reduces the total data size by a factor of v/z but some correlation information is lost and the variance of each path delay is less than the exact correlation value. To ensure that we do not underestimate the variance of path delays, difference between $\mathbf{W}$ and $\mathbf{W'}$ is represented as a residue term r_j for each path. This residue is assumed to be uncorrelated so that it is unlikely to underestimate the path delay. Therefore, the path delays can be expressed as

$$\mathbf{D} = \begin{bmatrix} d'_{path}(1) \\ \vdots \\ d'_{path}(z) \end{bmatrix} + \mathbf{W'}\mathbf{I}_{wd} + \sum_{j=1}^{z} r_j \tag{8}$$

where $\mathbf{W'}$ is the compressed matrix with v principal components. Though part of the correlation information is not captured, Figure 3 shows that our method is efficient in reducing pessimism in delay estimation, in contrast to assuming that all paths are completely independent. Moreover, this method is flexible as it provides a trade-off between accuracy and data size, by choosing a suitable number of principal components. The size of correlation matrix is O(v × number of paths).

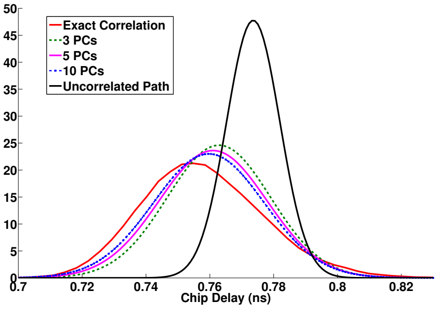


Fig. 3. Comparison between delay distributions for circuit C432.

In (8), each row of $\mathbf{D}$ represents delay of a path in the canonical form for tightness probability calculation. We use the method proposed in [34] to obtain the maximum delay of z critical paths on a chip.

$$D_{\text{chip}} = \mathcal{N}(\mu_{\text{delay}}, \sigma_{\text{delay}}) \tag{9}$$

where D_{chip} is the maximum delay of a chip, and μ_{delay} and σ_{delay} are the mean and standard deviation of maximum delay distribution of a chip.

E. Dealing with Measurement Noise

To reduce the measurement uncertainties, it is common to have multiple devices under test connected in parallel and carry out the measurement repeatedly. Thus, we assume every measurement is repeated N_e times, and the scribe-line test structure has N_d devices connected in parallel. Only the sum of device currents and capacitance of every chip are measured, i.e., the mean I_{eff}, I_{off} and device capacitance per unit width are obtained. The mean of measured I_{eff} for a chip is denoted as $\hat{I}_{\text{eff}}$, and it is expressed as

$$\hat{I}_{\text{eff}} = \frac{1}{N_e} \sum_{m=1}^{N_e} \frac{\tilde{I}_{\text{eff}}(m)}{N_d} \tag{10}$$

where $\tilde{I}_{\text{eff}}(m)$ is the sum of I_{eff} for N_d devices at the m^{th} measurement and N_e is the total number of measurements. Based on the measured $\hat{I}_{\text{eff}}$, we can represent I_{eff} as follows (see Appendix A for detailed derivations).

$$\begin{aligned} \mu_{I_{\text{eff}}} &= \hat{I}_{\text{eff}} \\ \sigma^2_{I_{\text{eff}}} &= \frac{\hat{I}_{\text{eff}} \sigma^2_{I_{wd}}}{N_d} + \frac{\sigma^2_F}{N_e} \end{aligned} \tag{11}$$

where $\sigma^2_{I_{wd}}$ and σ^2_F are the variance of within-die variation and measurement noise, respectively. Note that, the variance of I_{eff} is inversely proportional to the number of measurements and total devices in the test structure. In this paper, unless otherwise mentioned, we assume 5 measurements are taken every time ($N_e = 5$) and there are 10 devices in each test structure ($N_d = 10$). We assume $3 \times \sigma_F$ is 5% of nominal I_{eff} value. $\sigma_{I_{wd}}$ is obtained by running Monte-Carlo simulation over the variation ranges specified in Table II.

F. Interconnect Delay Variation

Since scribeline measurement is done after Metal-1 layer, the proposed model cannot fully capture interconnect-induced delay variation. However, the effect of interconnect variation is less pronounced due to the following reasons [6]:

- Interconnect variations on different metal layers are independent. Therefore, interconnect-induced delay variation averages out to a small value when a path passes through different metal layers.
- Interconnect width variation changes wire resistance and capacitance in opposite ways, reducing its net effect on RC.

Nonetheless, we include this effect in our experiments and measure the error incurred in estimation of delay because of variation in interconnect metal layers.

[6] The within-die I_{eff} variation can also be estimated from historical data.

III. Leakage Power Estimation Using I_{off}

A. Leakage Power Model

We model leakage power of a chip (P_{chip}) as a linear function of I_{off} as follows.[7]

$$P_{\text{chip}} = \sum_{t\in T}\sum_{c\in\Gamma}\sum_{l\in G_c} \alpha(c,t) I_{\text{off}}(l,c,t) \tag{12}$$

where l is the index for an instance, T is the set of device types, G_c is the set of instances for cell type c in the design, Γ is the set of all cell types, $\alpha(c,t)$ is the leakage power fitting coefficient for cell type (c) and device type (t), and $I_{\text{off}}(l,c,t)$ is device type t leakage current for an instance l with cell type c. To estimate leakage power variation, we model I_{off} as an exponential function of variation sources [29].

$$I_{\text{off}}(l,c,t) = I_{\text{off}-\text{nom}}(c,t)e^{Y(l,c,t)}$$

where $I_{\text{off}-\text{nom}}$ is the nominal I_{off} and Y represents the impact of variation sources. We model Y as a linear combination of inter-die and within-die variations, which are Gaussian random variables,

$$I_{\text{off}}(l,c,t) = I_{\text{off}-\text{nom}}(c,t)e^{Y_g(t)+Y_r(l,c,t)} \tag{13}$$

where $Y_g(t)$ denotes the total inter-die variation for device type t. $Y_r(l,c,t)$ is the within-die variation for device type t in cell type c and is specific to instance l. Combining (12) and (13), we have

$$\begin{aligned} P_{chip} &= \sum_{t\in T}\sum_{c\in\Gamma} P_{cell}(c,t) \\ P_{cell}(c,t) &= \alpha(c,t)I_{\text{off}-\text{nom}}(c,t)e^{Y_g(t)}\sum_{l\in G_c} e^{Y_r(l,c,t)} \\ \text{since} &\sum_{l\in G_c} e^{Y_r(l,c,t)} \approx |G_c|\cdot\mu_r(c,t)\ [29] \\ P_{cell}(c,t) &\approx \alpha(c,t)I_{\text{off}-\text{nom}}(c,t)e^{Y_g(t)}|G_c|\cdot\mu_r(c,t) \end{aligned} \tag{14}$$

where P_{cell} is total leakage power of cell type c for a chip, $|G_c|$ is the total number of instance of cell type c in the chip, $\mu_r(c,t)$ is the mean of $e^{Y_r(l,c,t)}$, which the foundry can extract from historical data. In our experiments, $\mu_r(c,t)$ is obtained by running Monte-Carlo simulations at randomly sampled process conditions, based on the variation model in Table II.

B. Dealing with Measurement noise

To calculate leakage power of a die, we extract $Y_g(t)$ by measuring $I_{\text{off}}(t)$ of N_d devices of type t for N_e times. I_{off} of the m^{th} measurement of device type t is modeled as follows.

$$\begin{aligned} \tilde{I}_{\text{off}}(m,t) &= \sum_{s=1}^{N_d} I_{\text{off}-\text{nom}}(t)e^{Y_g(t)+Y_{rt}(s,t)}(1+Z_m) \\ &\approx N_d I_{\text{off}-\text{nom}}(t)e^{Y_g(t)}\mu_{rt}(1+Z_m) \end{aligned} \tag{15}$$

where $\tilde{I}_{\text{off}}(m,t)$ is the sum of I_{off} for N_d devices of type t for the m^{th} measurement, and $\mu_{rt}(t)$ is the mean of $e^{Y_{rt}(s,t)}$. Z_m is the normalized measurement noise for the m^{th} measurement, which is modeled as a Gaussian random variable with zero mean and standard deviation σ_Z. Based on the measured leakage current, the mean ($\mu_{Y_g(t)}$) and variance ($\sigma^2_{Y_g(t)}$) of $Y_g(t)$ are given as follows (see Appendix B for details).

$$\begin{aligned} \mu_{Y_g(t)} &= \frac{1}{N_e}\sum_{m=1}^{N_e} ln(\frac{\tilde{I}_{\text{off}}(m,t)}{N_d I_{\text{off}-\text{nom}}(t)\mu_{rt}}) \\ \sigma^2_{Y_g(t)} &= \sigma^2_Z/N_e \end{aligned} \tag{16}$$

Equation (14) shows that P_{chip} is sum of $P_{cell}(c,t)$, each of which is a log normal distribution[8]. Thus, we can apply Wilkinson's approach in [29] to approximate P_{chip} as a lognormal random variable, and calculate its mean and variance based on the log normal distribution of P_{cell} specified by $Y_g(t)$.

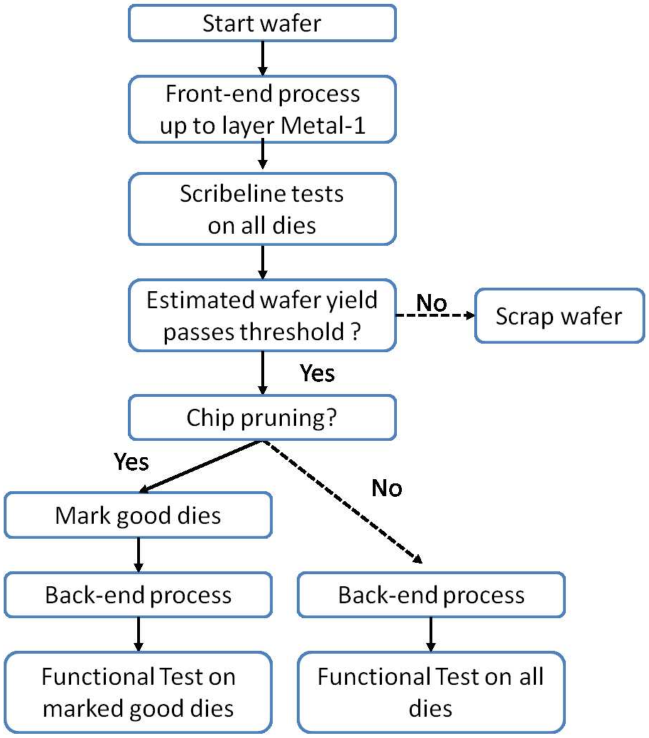

Start wafer
Front-end process up to layer Metal-1
Scribeline tests on all dies
Estimated wafer yield passes threshold ?
No
Scrap wafer
Yes
Chip pruning?
Yes
No
Mark good dies
Back-end process
Back-end process
Functional Test on marked good dies
Functional Test on all dies

Fig. 4. Proposed wafer and chip pruning flow.

IV. Wafer and Chip Pruning Strategy

In conventional manufacturing, accurate circuit performance becomes available only after dicing and packaging. Any failed chip at that stage incurs losses due to unnecessary fabrication, packaging, and testing costs. To reduce the cost per good chip, we propose a wafer and chip pruning flow illustrated in Figure 4. After processing a wafer up to layer Metal-1, scribeline measurements are carried out on every die. Based on the scribeline measurement data, we estimate chip performance and calculate the expected yield of each wafer. A

[7] In this paper, we only consider subthreshold leakage, but the model can be easily extended to consider gate leakage.

[8] $Y_g(t)$ for all device types is affected by within-die random variation and measurement noise, which are mutually independent. Therefore, the mean and variance of P_{chip} can be calculated as the sum of the mean and variance of $P_{cell}(c,t)$.

wafer will be scrapped if the expected number of good chips does not meet a pre-defined wafer pruning threshold (WPT) value. For the *wafer-level pruning only* scenario, wafers that pass the pruning threshold will go through back-end process and functional test, as in conventional manufacturing flow. For *wafer and chip pruning* scenario, good dies are marked using existing techniques (e.g., [2] [13]) so that only the good dies will be tested after back-end processes.

A. Passing Probability for a Chip

Given the measured I_{eff} and capacitance, conditional probability of a chip meeting timing constraint is given by

$$\begin{aligned}
&\Pr\ \{\text{chip delay} \le D_{\text{spec}} | (I_{\text{eff}} = \hat{I}_{\text{eff}}, C_{gate} = \hat{C}_{gate})\} \\
&\qquad = \Phi(\frac{D_{\text{spec}} - \mu_{\text{delay}}}{\sigma_{\text{delay}}}) \\
&\mu_{\text{delay}} = f(\hat{I}_{\text{eff}}, \hat{C}_{gate}) \\
&\sigma_{\text{delay}} = f(\hat{I}_{\text{eff}}, \sigma_{I_{wd}}, \sigma_F)
\end{aligned}$$

where $\hat{C}_{gate}$ is the measured capacitance, $\hat{I}_{eff}$ is the mean of measured I_{eff}, D_{spec} is the maximum allowed delay for a design, $\Phi(\cdot)$ is standard normal cumulative distribution function, μ_{delay} and σ_{delay} are mean and standard deviation of maximum delay distribution. On the other hand, the probability of a chip meeting leakage power constraint is given by

$$\begin{aligned}
&\Pr\ \{P_{\text{chip}} \le P_{\text{spec}} | I_{\text{off}} = \hat{I}_{\text{off}}\} \\
&\qquad = \Pr\ \{ln(P_{\text{chip}}) \le ln(P_{\text{spec}}) | I_{\text{off}} = \hat{I}_{\text{off}}\} \\
&\qquad = \Phi[\frac{ln(P_{\text{spec}}) - \mu_L}{\sigma_L}] \\
&\mu_L = f(\hat{I}_{\text{off}}) \\
&\sigma_L = f(\sigma_Z)
\end{aligned}$$

where μ_L is the mean of $ln(P_{chip})$ and σ_L is the variance of $ln(P_{chip})$.

Given the measured values ($\hat{I}_{\text{eff}}$, $\hat{I}_{\text{off}}$ and $\hat{C}_{gate}$) of every chip, the probability of a chip meeting timing or leakage power constraint is determined by the uncertainties in chip delay and leakage power. Note that, uncertainty in delay estimation (σ_{delay}) is due to I_{eff} within die variation and measurement noise, while uncertainty in leakage power estimation (σ_L) is only induced by measurement noise in I_{off}. Since the measurements of I_{eff} and I_{off} are taken using different measurement steps and bias conditions, the measurement noise for leakage power estimation is independent of the measurement noise for delay estimation. As a result, the uncertainties of chip delay and leakage power are modeled by two independent Gaussian random variables. Therefore, the probability of a chip meeting the timing constraint and the probability of a chip meeting the leakage power constraint are **conditionally independent** given the values of $\hat{I}_{\text{eff}}$, $\hat{I}_{\text{off}}$ and $\hat{C}_{gate}$. The passing probability of a chip is given by

$$\begin{aligned}
&\Pr\ \{P_{\text{chip}} = \text{pass} | (I_{\text{eff}} = \hat{I}_{\text{eff}}, C_{gate} = \hat{C}_{gate}, I_{\text{eff}} = \hat{I}_{\text{eff}})\} \\
&\quad = \Pr\ \{P_{\text{chip}} \le P_{\text{spec}} | I_{\text{off}} = \hat{I}_{\text{off}}\} \times \\
&\qquad \Pr\ \{\text{chip delay} \le D_{\text{spec}} | (I_{\text{eff}} = \hat{I}_{\text{eff}}, C_{gate} = \hat{C}_{gate})\}
\end{aligned} \tag{17}$$

TABLE I
MANUFACTURING AND TESTING COST SETUPS, WHERE THE COSTS ARE REPRESENTED IN PERCENTAGES.

	Setup 1	Setup 2	Setup 3	Setup 4	Setup 5	Setup 6
Scribeline test cost (%)	0	0	0	3	3	3
Front-end cost (%)	36	60	20	35	59	19
Back-end cost (%)	30	20	20	29	19	19
Test cost (%)	34	20	60	33	19	59
Total cost (%)	100	100	100	100	100	100

Meanwhile, the expected number of good chips in a wafer (EG) can be estimated as the sum of passing probability of all chips in a wafer.

$$\begin{aligned}
&EG = \\
&\sum_{\text{chips}} \Pr\ \{P_{\text{chip}} = \text{pass} | (I_{\text{eff}} = \hat{I}_{\text{eff}}, C_{gate} = \hat{C}_{gate}, I_{\text{eff}} = \hat{I}_{\text{eff}})\}
\end{aligned} \tag{18}$$

B. Cost Model

The benefit of wafer or chip pruning is strongly related to chip selling price, manufacturing-cost and testing-cost, which are affected by many factors. For instance, the chip selling price varies due to demand and supply of a product, marketing strategy, etc.; manufacturing cost depends on manufacturing equipment, raw materials, and processing costs [16]; testing cost is affected by the number of test patterns and the testing infrastructure. In this paper, we define relative costs for scribeline testing (M_s), front-end-of-line (M_f), back-end-of-line (M_b), and full-chip testing cost (M_t) in Table I, to account for different scenarios. For cost setup 1, we obtain the ratio between M_f and M_b from [35]. The cost model in [35] describes a wafer process with 20 layers, and processing each layer costs $466. In this paper, we assume the front-end cost, M_f, includes the processing cost for the first 10 layers of a wafer, and a $81.6/wafer raw wafer cost [35]; M_b includes the processing cost for the remaining 10 layers. We then estimate the testing cost, M_t, as 50% of the total manufacturing cost (M_f + M_b) [39]. Cost setup 2 and 3 are hypothetical cases to evaluate the benefit of proposed wafer pruning for different cost setups.

We assume that the scribeline testing cost is negligible in cost setups 1, 2, and 3, as scribeline measurements may be taken by a foundry as a standard procedure for process monitoring. Cost setups 4, 5 and 6 model the scenario where scribeline measurements are not taken in the standard manufacturing flow and the measurements incur additional cost. We assume scribeline measurement cost is lower than the final testing cost because the number of items to be measured is much less than the final testing ones.

We acknowledge that our cost model does not consider many practical aspects of semiconductor manufacturing. However, the cost model mainly affects wafer pruning threshold (WPT), which is determined by fixed cost (irrespective of pruning) and pruning-dependent cost. Therefore, we split total semiconductor manufacturing cost into four components that are fixed or pruning-dependent, and evaluate several scenarios by varying the relative values among the cost components. The actual pruning decision making and WPT will depend on variety of factors, including cost, volume demand, machine capacity, chip price, etc., detailed analysis of which are beyond the scope of the current work.

C. Wafer and Chip Pruning Analysis

In the proposed wafer pruning strategy, we will prune a wafer if its expected yield is lower than WPT. Clearly, the benefit of pruning is dependent on the WPT value, which can be guided by the expected profit and additional cost to continue making the wafer. In this paper, we define WPT so that we will prune a wafer only if its *expected profit* is smaller than additional cost to make the wafer. The WPT for two pruning scenarios are given as follows.

- Option 1: wafer pruning only

$$\begin{aligned}
&\text{Additional Cost} = (M_b + M_t) \\
&\text{Expected profit} = EG \times \text{Chip price} \\
&\text{Expected profit} > \text{Additional Cost} \\
&EG \times \text{Chip price} > (M_b + M_t) \\
&\implies \text{WPT} = \frac{(M_b + M_t)}{\text{Chip price}}
\end{aligned} \tag{19}$$

- Option 2: wafer and chip pruning

$$\begin{aligned}
&\text{Additional Cost} = (M_b + \text{Expected good chips} \times M_t) \\
&\text{Expected profit} = EG \times \text{Chip price} \\
&\text{Expected profit} > \text{Additional Cost} \\
&EG \times (\text{Chip price} - M_t) > M_b \\
&\implies \text{WPT} = \frac{(M_b)}{\text{Chip price} - M_t}
\end{aligned} \tag{20}$$

Note that we do not consider the cost for front-end processes in (19) and (20) because the process has been carried out and incurred processing cost regardless of the pruning decision. The chip selling price is also a factor during wafer pruning. For example, if the chip selling price is much larger than the total manufacturing cost, then the foundry is less likely to prune a wafer because its expected profit is always greater than the additional cost to make a wafer. When we combine wafer and chip pruning, the additional cost to manufacture a wafer is lower because only a subset of the chips will be tested. Thus, WPT for combined wafer and chip pruning is less than the WPT of the wafer pruning only scenario.

V. Experiments and Results

A. Experiment Setup

Figure 5 summarizes our experiment setup, which demonstrates the flow of the proposed wafer pruning method. The upper part of the figure describes procedures to obtain design-specific parameters at a design house. We use Monte-Carlo SPICE simulations with the variation model specified in Table II to generate samples for $K_{cell}(c,t)$ and $\alpha(c,t)$ characterization. Note that the Monte-Carlo SPICE simulation can be replaced by timing libraries at various process corners to speed up the characterization. In this paper, we characterize $K_{cell}(c,t)$ and $\alpha(c,t)$ with the 45nm Nangate Open Cell library [27].

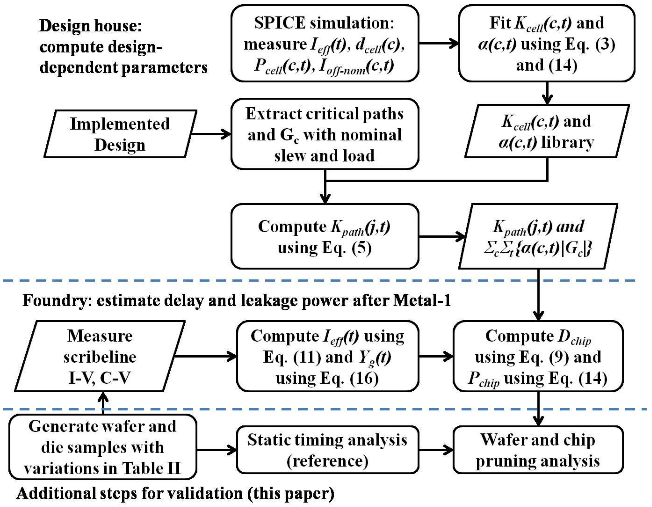

Design house: compute design-dependent parameters
SPICE simulation: measure $I_{eff}(t)$, $d_{cell}(c)$, $P_{cell}(c,t)$, $I_{off\text{-}nom}(c,t)$
Fit $K_{cell}(c,t)$ and $\alpha(c,t)$ using Eq. (3) and (14)
Implemented Design
Extract critical paths and G_c with nominal slew and load
$K_{cell}(c,t)$ and $\alpha(c,t)$ library
Compute $K_{path}(j,t)$ using Eq. (5)
$K_{path}(j,t)$ and $\Sigma_c\Sigma_t\{\alpha(c,t)|G_c|\}$
Foundry: estimate delay and leakage power after Metal-1
Measure scribeline I-V, C-V
Compute $I_{eff}(t)$ using Eq. (11) and $Y_g(t)$ using Eq. (16)
Compute D_{chip} using Eq. (9) and P_{chip} using Eq. (14)
Generate wafer and die samples with variations in Table II
Static timing analysis (reference)
Wafer and chip pruning analysis
Additional steps for validation (this paper)

Fig. 5. The proposed delay and leakage power estimation method. The upper part of the figure shows how the compressed design dependent parameters are computed, while the middle part indicates how delay and leakage power are estimated using these parameters at the foundry. The bottom part of the figure shows additional steps in this paper.

We implement a combination of ISCAS85 and OpenCores benchmark circuits with the 45nm Nangate Open Cell library. We extract the critical paths of the benchmark circuits and G_c. We consider all paths with nominal delay within 5% of the maximum path delay as critical paths[9]. Based on the nominal slew and load on critical paths, we compute $K_{path}(j,t)$, **W'**, **R**, $|G_c|$ and $\sum_c \sum_t \{\alpha(c,t)\}$ coefficients. These compressed design-dependent coefficients will be used to estimate chip delay and leakage power for the proposed pruning strategy.

Due to the lack of foundry data, we simulate wafer and die samples based on the variation model in Table II (lower part of Figure 5). For every benchmark circuit, we simulate 250 wafers, each of which has 657 chips. We obtain the actual delay and leakage power of each chip from the Primetime [40] STA and leakage power report. If both delay and leakage power of a chip meet the performance target, the chip is considered to be a good chip.

At the same time, we simulate PMOS and NMOS devices (high V_{th} and low V_{th}) using SPICE to extract I_{eff} and I_{off} (to emulate scribeline measurements). The devices have the same inter-die variation values as the chip, but there is mismatch due to within-die variation. For I_{eff} extraction, we use 5 principal components for each device type. Based on the simulated I_{eff} and I_{off} we compute the D_{chip} and P_{chip} of every chip. We perform STA and power analysis on the chip samples to obtain actual delay and leakage power for the wafer pruning benefit calculation. To evaluate the benefit of design-dependent delay and leakage power models, we implement a design-independent approach, which equally weighs high V_{th} and low V_{th} devices in the delay and leakage power estimations. We

[9] Many improved critical path selection algorithms have been proposed in literature [36] [38]. We do not implement the path selection algorithms, as it is beyond the scope of this work.

TABLE II
SUMMARY OF VARIATION PARAMETERS

Variation Source	$Wafer-$ $Wafer_{ran}\%$	$Die-$ $Die_{sys}\%$	$Die-$ $Die_{ran}\%$	$Within-$ $Die_{ran}\%$
Channel length	$\mathcal{N}(0, 2.13)$	$ax^2 + by^2 +$ $cx + dy + exy$	$\mathcal{N}(0, 1.29)$	$\mathcal{N}(0, 1.56)$
NMOS V_{th}	$\mathcal{N}(0, 6.4)$	$-$	$\mathcal{N}(0, 6.08)$	$\mathcal{N}(0, 4.7)$
PMOS V_{th}	$\mathcal{N}(0, 6.4)$	$-$	$\mathcal{N}(0, 6.08)$	$\mathcal{N}(0, 4.7)$
Interconnect width	$-$	$-$	$\mathcal{N}(0, 10)$	$-$
Interconnect thickness	$-$	$-$	$\mathcal{N}(0, 10)$	$-$

assume the design-independent delay estimation is inversely proportional to the mean of I_{eff} of all device types. Similarly, the leakage power estimation is proportional to the mean of I_{off} of all device types. Unless otherwise specified, timing constraints of the benchmark circuits are 110% of the nominal critical path delay of the respective designs, and the leakage power constraints are 5 times the nominal leakage power.

B. Variation Model

We model five independent variation sources for transistors as shown in Table II. V_{th} variations are modeled by Gaussian distributed random variables with no spatial variation [37]. Channel length is assumed to be the only variation source, which contributes to systematic delay variation across wafer and is modeled as

$$D_{sys} = ax^2 + by^2 + cx + dy + exy, \tag{21}$$

where x and y represent the coordinates of a chip's centroid [9]. The wafer diameter is $300mm$ and 657 chip centroids are distributed uniformly across the wafer. Since the model is applicable from 90nm to 45nm technologies [9] [28], we obtain the values of a, b, c, d and e by matching systematic delay variation across wafer to 65nm silicon data[10]. V_{th} variations in Table II are also extracted from the same silicon data. To model interconnect variation, we obtain σ/μ ratio of wire width from [17], and assume that wire thickness has similar ratio[11].

Interconnect variation is modeled as random Gaussian-distributed intra-die variation [5]. In our experiments, this is implemented by perturbing unit resistance and capacitance values in the LEF files of implemented benchmark circuits.

C. Wafer Pruning Results

In Table III and Table IV, we compare the *cost per good chip* resulting from the proposed wafer pruning method. The definitions of cost per good chip are defined as follows.

$$\begin{gathered}\text{cost per good chip with no pruning} = \\ \frac{(M_f + M_b + M_t) \times \text{total wafers}}{\text{total number of actual good chips}} \\[2ex] \text{cost per good chip with pruning} = \\ \frac{(M_s + M_f) \times \text{total wafers} + (M_b + M_t) \times \text{good wafers}}{\text{total number of actual good chips}}\end{gathered} \tag{22}$$

where *total wafers* is 250 and *good wafers* is the total number of wafers with a yield rate (ratio of total number of good chips to total chips on a wafer) higher than the WPT. *Total number of actual good chips* is obtained by summing up actual good chips for wafers that pass the early wafer pruning. Note that the number of good wafers varies depending on the pruning method. Therefore, the total number of actual good chips is also different across the pruning methods.

Table III and Table IV show that the cost per good chip is higher than 1.0 for no wafer pruning case. This happens because the wafer yield is smaller than 100% (due to process variation). Results in the tables show that proposed design-dependent wafer pruning method reduces cost per good chip by up to 10% compared to the no pruning case when a large portion of the total cost is spent on back-end and final testing (cost setups 1, 3, 4, and 6). When wafer cost is dominated by front-end and fixed costs (cost setups 2 and 5), wafer pruning may increase the total cost. On an average, design-dependent wafer pruning can reduce cost per good chip by 6%, compared to the design-independent wafer pruning approach.

Figure 6 shows the profit per good chip for different pruning approaches and cost setups.

$$\begin{aligned}&\text{Profit per good chip} = \\ &\text{chip selling price} - \text{cost per good chip}\end{aligned} \tag{23}$$

The results show that the proposed design-dependent method has a higher profit per good chip compared to the design-independent method. Early wafer pruning is beneficial when wafer cost is dominated by back-end processes and final test cost (cost setups 1, 3, 4 and 6). However, early wafer pruning reduces profit per good chip compared to the no pruning case, when wafer cost is dominated by front-end and fixed costs (cost setups 2 and 5).

Figure 7 shows optimal WPT that minimizes cost per good chip varies for different cost setups. Therefore, we need to set WPT according to the cost setups. When wafer cost is dominated by back-end and test costs (cost setups 3 and 6), we need to set a larger WPT such that any manufactured wafer has enough good chips to compensate for manufacturing and test cost. When wafer cost is dominated by front-end and fixed costs (cost setups 2 and 4), we need to set a lower WPT because scrapping any wafer incurs significant losses. As chip selling price reduces, the expected profit of making a wafer also reduces. As a result, a higher WPT is needed to ensure that it is beneficial to continue processing a wafer.

Results in Figure 7 also show that the WPT estimated by (19) is a good approximation to the optimal WPT that minimizes cost per good chip. When the WPT is large (> 0.5),

[10] For our model, $a = 7.7e^{-4}$, $b = 1.0e^{-3}$, $c = -1.6e^{-2}$, $d = -7.8e^{-3}$, $e = 1.6e^{-4}$

[11] Wire thickness variation is not available in ITRS reports.

TABLE III

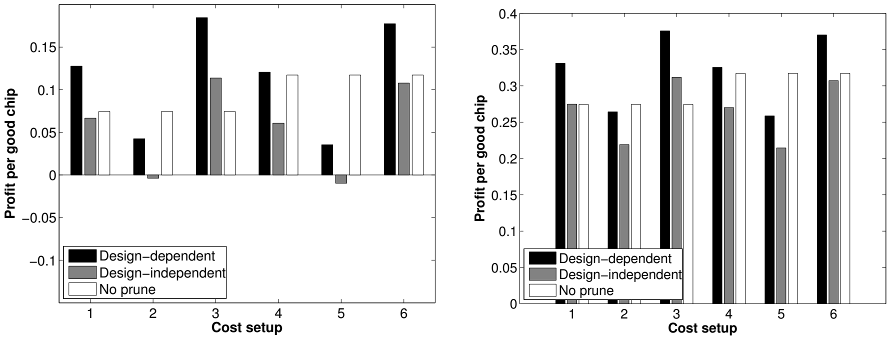
COST COMPARISON FOR DIFFERENT WAFER PRUNING STRATEGIES. COST PER GOOD CHIP IS NORMALIZED TO THE COST PER CHIP WITH 100% YIELD. WPT IS CALCULATED USING (19). *Dep.*, *Indep.* AND *Normal* REFERS TO DESIGN-DEPENDENT, DESIGN-INDEPENDENT AND NO PRUNING EXPERIMENT SETUPS, RESPECTIVELY. CHIP SELLING PRICE IS 1.5 TIMES OF THE COST PER CHIP WITH 100% YIELD.

bench-	Cost setup 1			Cost setup 2			Cost setup 3			Cost setup 4			Cost setup 5			Cost setup 6		
marks	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.
C432	1.54	1.59	1.62	1.67	1.66	1.62	1.45	1.54	1.62	1.55	1.59	1.62	1.68	1.67	1.62	1.47	1.55	1.62
C432L	1.26	1.34	1.29	1.29	1.41	1.29	1.24	1.28	1.29	1.26	1.34	1.29	1.29	1.42	1.29	1.24	1.29	1.29
s15850	1.40	1.44	1.48	1.50	1.51	1.48	1.33	1.39	1.48	1.41	1.45	1.48	1.51	1.52	1.48	1.34	1.40	1.48
s38584	1.33	1.39	1.36	1.42	1.46	1.36	1.27	1.34	1.36	1.33	1.39	1.36	1.42	1.47	1.36	1.27	1.34	1.36
mips789	1.34	1.42	1.37	1.41	1.48	1.37	1.29	1.38	1.37	1.34	1.42	1.37	1.42	1.48	1.37	1.29	1.38	1.37
Average	1.37	1.43	1.43	1.46	1.50	1.43	1.32	1.39	1.43	1.38	1.44	1.43	1.46	1.51	1.43	1.32	1.39	1.43

TABLE IV

COST COMPARISON FOR DIFFERENT WAFER PRUNING STRATEGIES. COST PER GOOD CHIP IS NORMALIZED TO THE COST PER CHIP WITH 100% YIELD. WPT IS CALCULATED USING (19). *Dep.*, *Indep.* AND *Normal* REFERS TO DESIGN-DEPENDENT, DESIGN-INDEPENDENT AND NO PRUNING EXPERIMENT SETUPS, RESPECTIVELY. CHIP SELLING PRICE IS 1.7 TIMES OF THE COST PER CHIP WITH 100% YIELD.

bench-	Cost setup 1			Cost setup 2			Cost setup 3			Cost setup 4			Cost setup 5			Cost setup 6		
marks	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.
C432	1.53	1.58	1.62	1.64	1.64	1.62	1.47	1.55	1.62	1.54	1.59	1.62	1.64	1.64	1.62	1.47	1.55	1.62
C432L	1.26	1.32	1.29	1.28	1.38	1.29	1.24	1.28	1.29	1.26	1.33	1.29	1.29	1.39	1.29	1.25	1.29	1.29
s15850	1.40	1.44	1.48	1.48	1.50	1.48	1.35	1.40	1.48	1.41	1.45	1.48	1.49	1.50	1.48	1.35	1.41	1.48
s38584	1.32	1.38	1.36	1.39	1.44	1.36	1.27	1.34	1.36	1.33	1.38	1.36	1.40	1.44	1.36	1.28	1.34	1.36
mips789	1.33	1.40	1.37	1.39	1.45	1.37	1.29	1.37	1.37	1.34	1.41	1.37	1.39	1.45	1.37	1.30	1.38	1.37
Average	1.37	1.43	1.43	1.44	1.48	1.43	1.32	1.39	1.43	1.37	1.43	1.43	1.44	1.49	1.43	1.33	1.39	1.43



(a) Normalized chip selling price = 1.5

(b) Normalized chip selling price = 1.7

Fig. 6. Average profit per good chip of all benchmarks with different cost setups. Profit per good chip and chip selling price are normalized to the cost per chip with 100% yield. WPT is obtained from (19).

TABLE V

COST PER GOOD CHIP (NORMALIZED TO THE COST PER CHIP WITH 100% YIELD) FOR DESIGN-DEPENDENT WAFER PRUNING BASED ON LIMITED SAMPLING. CHIP SELLING PRICE IS 1.7 TIMES THE COST PER CHIP WITH 100% YIELD. WPT IS OBTAINED FROM (19).

Sampling ratio (%)	5	10	30	50	80	100
cost setup 1	1.36	1.37	1.38	1.38	1.38	1.38
cost setup 2	1.54	1.54	1.54	1.54	1.54	1.54
cost setup 3	1.19	1.18	1.18	1.17	1.18	1.18
cost setup 4	1.34	1.34	1.36	1.37	1.38	1.39
cost setup 5	1.50	1.51	1.52	1.53	1.54	1.55
cost setup 6	1.16	1.16	1.17	1.18	1.19	1.20

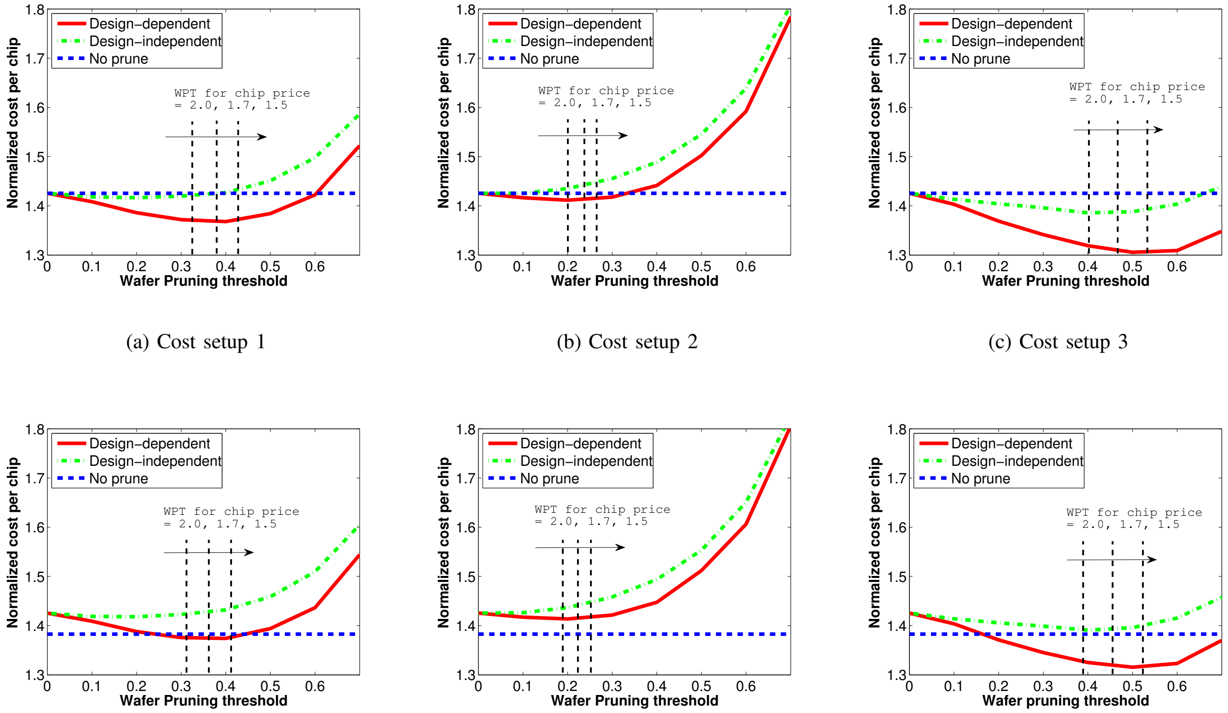
most of the wafers will be pruned even if there are many good chips on a wafer. As a result, the cost per good chip increases along with WPT.

To reduce scribeline testing cost, we study the impact of randomly sampling chips for delay and leakage power estimation (instead of measuring every chip on a wafer) on wafer pruning quality. In this experiment, we estimate the delay and leakage power based on the randomly sampled chips and the scribeline test cost is scaled proportionally with the sampling ratio. Table V shows that total cost per good chip reduces as the number of samples reduces for cost setups 3, 4, and 5. This implies that this method can minimize cost overhead incurred by scribeline testing. Note that this method can be further improved by sampling strategies like [23], [30].

To evaluate the impact of measurement noise and test-structure design, we run an experiment with different N_e and N_d. Table VI shows that the cost per good chip achieved by our strategy is insensitive to the measurement count and to the number of devices in test structures. Therefore, there is a potential of optimizing the test structures to reduce measurement time and scribeline area.

D. Chip Pruning Results

Figure 8 shows the chip pruning benefits of the proposed strategy for the C432 and MIPS789 benchmarks as described in Section IV. Y-axis shows the percentage of chips that are bad and pruned. X-axis shows the amount of yield loss that results from chip pruning. The plot is made by varying delay and leakage power guardbands, which are indicated as values



(a) Cost setup 1

(b) Cost setup 2

(c) Cost setup 3

(d) Cost setup 4

(e) Cost setup 5

(f) Cost setup 6

Fig. 7. Average cost per good chip of all benchmarks with different wafer-level pruning strategies. As chip selling price reduces, a wafer must have a higher yield rate (more good chips per wafer) to be profitable. Therefore, WPT increases along when chip selling price reduces. WPT is calculated using (19). Cost per good chip and chip selling price are normalized to the cost per chip with 100% yield.

TABLE VI
COST PER GOOD CHIP (NORMALIZED TO THE COST PER CHIP WITH 100% YIELD) OF BENCHMARK C432 FOR DIFFERENT MEASUREMENT/TEST STRUCTURE SETUP. CHIP SELLING PRICE = 1.7 TIMES THE COST PER CHIP WITH 100% YIELD. WPT IS OBTAINED FROM (19).

N_e	N_d	Cost setup 1	Cost setup 2	Cost setup 3	Cost setup 4	Cost setup 5	Cost setup 6
1	1	1.55	1.69	1.45	1.56	1.70	1.46
5	10	1.54	1.67	1.45	1.55	1.68	1.47
100	100	1.54	1.67	1.45	1.55	1.68	1.47

in brackets for some points on the plot. The square brackets indicate the prune percentage and the yield loss. There is a trade-off between the percentage of chips pruned and yield loss. Note that, a very large percentage of bad chips can be efficiently pruned at the cost of very small yield loss, which results in significant savings on the costly tester time. For example, we can prune almost 70% of bad chips with less than 1% yield loss. This corresponds to almost 15% savings on the tester time. Effective chip pruning is only possible if false positive cases (pruned chips are good chips) are less likely to happen compared to true positive cases (pruned chips are bad chips). This happens when the probability of estimation error reduces sharply as the magnitude of estimation error increases (e.g., a normal distribution). Table VII shows that experiments on other benchmark circuits show similar chip pruning results.

Figure 9 shows that chip pruning can achieve about 5% cost reduction compared to the design-independent approach. Meanwhile, the cost reduction compared to the no-pruning case varies from -1% to 10%, depending on the cost setup. The higher cost of design-independent chip pruning implies that inaccurate performance estimation in the design-independent approach can cause losses when it prunes a good working chip.

TABLE VII
PRUNE PERCENTAGE AND YIELD LOSS OF BENCHMARK CIRCUITS. THE LAST COLUMN INDICATES TOTAL BAD CHIPS (%) IN ALL WAFERS. IN THIS EXPERIMENT, WE ASSUME THERE IS NO WAFER PRUNING, I.E., ALL WAFERS PASSES THE WAFER PRUNING STAGE.

Guard-band	(delay , power) (1.06, 1.20)		(delay , power) (1.12, 1.30)		(delay , power) (1.18, 1.40)		Bad chip %
	Prune %	YL %	Prune %	YL %	Prune %	YL %	
c432	26.08	0.61	15.78	0.02	8.64	0.00	38.40
s15850	23.17	1.27	15.86	0.12	9.94	0.01	32.67
s38584	19.28	2.40	12.12	0.31	6.90	0.03	26.54
mips789	19.41	1.62	11.29	0.06	5.82	0.00	27.21
c432L	11.71	0.22	5.72	0.02	3.01	0.01	22.24

E. Wafer and Chip Pruning Results

For combined wafer and chip pruning, the cost per good chip is given as follows.

$$\begin{aligned}&\text{cost per good chip} = \\ &\quad \{\text{total wafers} \times (M_s + M_f) + \text{good wafers} \times M_b \\ &\quad + \text{good chips} \times M_t\}/\text{total number of actual good chips}\end{aligned} \tag{24}$$

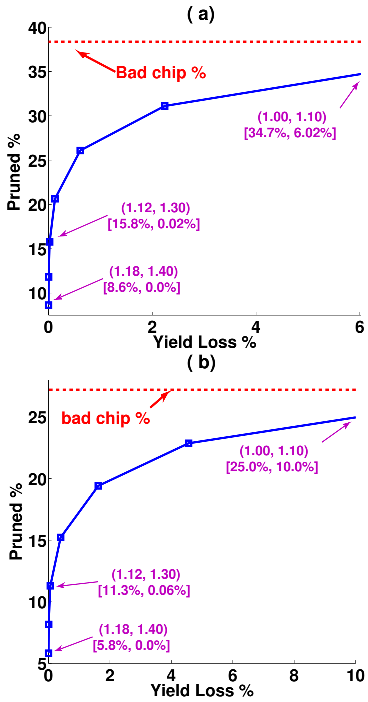


Fig. 8. Chip pruning results for benchmark design (a) c432, (b)mips789. Timing and leakage power guardbands are indicated in brackets. The square brackets indicate the prune percentage and the yield loss.

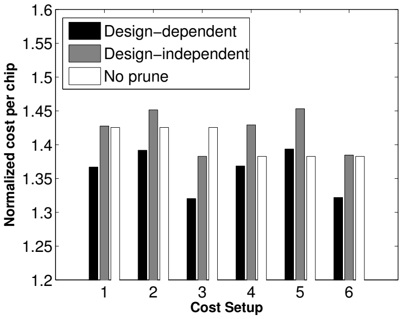


Fig. 9. Cost per good chip (normalized to the cost per chip with 100% yield) of the average of all benchmark designs using different chip-level pruning strategies. The timing and leakage power guardband used for chip pruning are 12% and 30%, respectively. Chip selling price is 1.7 times of the cost per chip with 100% yield. WPT = 0.

where *good chips* is the total number of estimated good chips on the wafers that pass WPT. Figure 10 shows that when we combine "wafer and chip pruning", the cost per good chip is lower than the no pruning scenario, except in cost setup 5, where most of the cost happens at the early manufacturing stage. In all cases, applying chip-level-only pruning can further reduce cost per good chip by 1% to 3% compared to applying wafer-level-only pruning because it has a finer pruning granularity. In cases where back-end manufacturing cost dominates the total manufacturing cost (cost setup 3 and 6), wafer-level-only pruning is very effective as it has a similar cost per good chip as the wafer and chip pruning method.

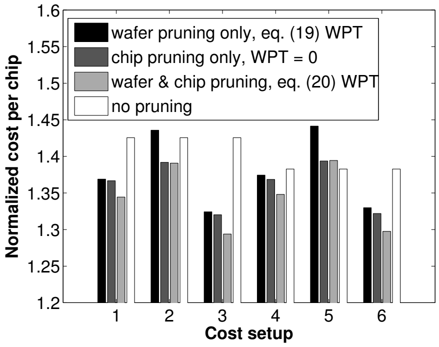


Fig. 10. Cost per good chip (normalized to the cost per chip with 100% yield) of the average of all benchmark designs using different design-dependent pruning approaches. Chip pruning's timing and leakage power guardbands are 12% and 30% of design's specifications. Chip selling price is 1.7 times of the cost per chip with 100% yield.

VI. Conclusions and Future Work

In this work, we present a novel approach for design-dependent process monitoring. Such process monitors are placed on wafer scribelines and can be tested after Metal-1 fabrication. This allows for early chip performance and wafer yield estimation dependent on the current process snapshot (as opposed to long-term statistics). We use this for cutting short the production of obviously bad wafers (i.e., where the wafer yield is too low to cover manufacturing/test costs) and avoiding testing of obviously bad chips. The wafer pruning approach based on our method can reduce cost per good chip up to 10%. Using our method, chip pruning can prune almost 70% of bad chips with less than 1% yield loss. Combining the wafer and chip pruning methods, we reduce the cost per good chip by 1% to 3% (compared to the wafer pruning only). Also, the monitoring strategy is chosen so as to minimize information exchange between the design house and the foundry as much as possible. Our future work will explore block-based delay estimation for early wafer pruning if statistical timing analysis is part of the design flow.

References

[1] K. V. Arnim, C. Pacha, K. Hofmann, T. Schulz, K. Schrfer and J. Berthold, "An Effective Switching Current Methodology to Predict the Performance of Complex Digital Circuits", *IEEE International Electron Devices Meeting*, 2007, pp. 483-486.

[2] R. J. Beffa, "Method in an Integrated Circuit (IC) Manufacturing Process for Identifying and Redirecting IC's Mis-processed During Their Manufacture",*U.S. Patent*, No. US6363329B2, Mar. 2002.

[3] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi and V. De, "Parameter Variations and Impact on Circuits and Microarchitecture", *Proc. IEEE/ACM Design Automation Conference* , 2003, pp. 338-342.

[4] M. Bhushan, A. Gattiker, M. B. Ketchen and K.K. Das, "Ring Oscillators for CMOS Process Tuning and Variability Control", *IEEE Transactions on Semiconductor Manufacturing* 1(1) (2006), pp. 10-19.

[5] Y. Cao, P. Gupta et al. ,"Design Sensitivities to variability: Extrapolation and assessments in nanometer VLSI", *Proc. IEEE International Conference on ASIC/SoC*, 2002, pp. 411-415.

[6] T.-B. Chan, R. S. Ghaida and P. Gupta, "Electrical Modeling of Lithographic Imperfections", *Proc. IEEE/ACM International Conference on VLSI Design*, 2010, pp. 423-428.

[7] T.-B. Chan, A. Pant, L. Cheng and P. Gupta, "Design Dependent Process Monitoring for Back-end Manufacturing Cost Reduction", *Proc. IEEE/ACM International Conference on Computer-Aided Design* , 2010, pp. 116-122.

[8] M. Chen and A. Orailoglu, "Test cost minimization through adaptive test development", *Proc. IEEE International Conference on Computer Design*, 2008, pp.234-239.

[9] L. Cheng, P. Gupta, C. Spanos, K. Qian and L. He ,"Physically Justifiable Die-Level modeling of Spatial Variation in View of Systematic Across Wafer Variability", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30(3) (2011), pp.388-401.

[10] C.Y. Cho, D.D. Kim, J.H. Kim D.Y. Lim and S.Y. Cho,"Early Prediction of Product Performance and Yield Via Technology Benchmark," *Proc. IEEE/ACM International Conference on Custom Integrated Circuits* , 2008, pp. 205-208.

[11] K.K. Das, S.G. Walker and M. Bhushan, "An Integrated CAD methodology for Evaluating Mosfet and Parasitic Extraction Models and Variabilitiy", *Proc. of the IEEE* 95(3) (2007), pp.670-687.

[12] J. M. Carulli Jr., D. C. Wrobbel, A. Mehta, K. E. Krause Jr., B. E. Campbell and F. A. Valente, "Frequency Distribution modeling for high-speed microprocessors using on-chip Ring-Oscillators", *spie*, vol. 3884, pp. 146-155.

[13] D. Corley and H. W. Littlebury, "Integral Semiconductor Wafer Map Recording", *U.S. Patent*, No. 5256578, Oct. 1993.

[14] A.J. Drake, R.M. Senger, H. Singh, G.D. Carpenter and N.K. James, "Dynamic Measurement of Critical-Path Timing", *Proc. IEEE Conference on Integrated Circuit Design and Technology and Tutorial*, 2008, pp. 249-252.

[15] S.-J. Han, X. Yu, N. Zamdmer, J. Deng, E.J. Nowak and K. Rim, "Improved Effective Switching Current (I^{+}_{EFF}) and Capacitance Methodology for Cmos Circuit Performance Prediction and Model-to-Hardware Correlation", *IEEE International Electron Devices Meeting*, 2008, pp. 1-4.

[16] ICknowledge, http://www.icknowledge.com/

[17] International Technology Roadmap for Semiconductors, 2007. http://public.itrs.net/ .

[18] M.B. Ketchen, M. Bhushan and D. Pearson, "High Speed Test Structures for In-Line Process Monitoring and Model Calibration," *Proc. IEEE/ACM Conference on Microelectronic Test Structures* , 2005, pp.33-38.

[19] K. J. Kuhn, M. D. Giles, D. Becher, P. Kolar, A. Kornfeld, R. Kotlyar, S. T. Ma, A. Maheshwari and S. Mudanai, "Process Technology Variation", *IEEE Transactions on Electronic Devices* 58(8) (2011), pp. 2197-2208.

[20] R. Lefferts and C. Jakubiec, "An Integrated Test Chip for the Complete Characterization and Monitoring of a 0.25um CMOS Technology that fits into five scribe line structures 150um by 5000 um", *Proc. IEEE/ACM Conference on Microelectronic Test Structures* , 2003, pp. 59-63.

[21] Q. Liu and S. S. Sapatnekar, "A Framework for Scalable Post-Silicon Statistical Delay Prediction under Spatial Variations", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 28(8) (2009), pp.1201-1212.

[22] Q. Liu and S. S. Sapatnekar, "Capturing Post Silicon Variations using a Representative Critical Path", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 29(2) (2010), pp. 211-222.

[23] X. Li, R. Rurenbar and S. Blanton, "Virtual Probe: A Statistically Optimal Framework for Minimum-Cost Silicon Characterization of Nanoscale Integrated Circuits", *Proc. IEEE/ACM International Conference on Computer-Aided Design* , 2009, pp. 433-440.

[24] D. N. Maynard, D. S. Kerr, C. Whiteside, "Cost of Yield", *Proc. IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop* , 2003, pp 165-170.

[25] S. Mitra, E. Volkerink, E.J. McCluskey and S. Eichenberger, "Delay Defect Screening using Monitoring Structures", *IEEE VLSI Test Symposium*, 2004, pp. 43-48.

[26] M.H. Na, E.J. nowak, W. Haensch and J. Cai, "The Effective Drive Current in Cmos Inverters", *IEEE International Electron Devices Meeting*, 2002, pp. 121-124.

[27] Nangate Open Cell Library,https://www.nangate.com/ .

[28] K. Qian and C. J. Spanos, "A Comprehensive Model of Process Variability for Statistical Timing Optimization," *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, vol. 6925, 2008, pp. 69251G.

[29] R. Rao, A. Devgan, D. Blaauw and D. Sylvester, "Parametric Yield Estimation Considering Leakage Variability", *Proc. IEEE/ACM Design Automation Conference* , 2004, pp. 442-447.

[30] S. Reda and S. R. Nassif, "Analyzing the Impact of Process Variations on Parametric Measurements: Novel Models and Applications", *Design, Automation and Test in Europe*, 2009, pp.375-380.

[31] F. Rigaud, J. M. Portal, H. Aziza et al. ,"Test Structure for Process and Product Evaluation", *Proc. IEEE/ACM Conference on Microelectronic Test Structures* , 2007, pp. 140-144.

[32] W. C. Riordan, R. Miller and E. R. St. Pierre, "Reliability Improvement and Burn In Optimization Through the Use of Die Level Predictive Modeling", *Proc. IEEE International Reliability Physics Symposium*, 2005, pp. 17-21.

[33] S. Natarajan, S. Patil and S. Chakravarty, "Path Delay Fault Simulation on Large Industrial Design", *IEEE VLSI Test Symposium*, 2006, pp. 1-6.

[34] C. Visweswariah, K. Ravindran, K. Kalafala, S.G. Walker, S. Narayan, D.K. Beece et al., "First-order Incremental Block-Based Statistical Timing Analysis", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 25(10) (2006), pp. 2170-2180.

[35] M.-C. Wu, C.-W. Chiou and H.-M. Hsu, "Scrapping Small Lots in a Low Yield and High-Price Scenario", *IEEE Transactions on Semiconductor Manufacturing* 17(1) (2004), pp. 55-67.

[36] L. Xie and A. Davoodi, "Bound-Based Statistically-Critical Path Extraction Under Process Variations", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30 (1) (2011), pp. 59-71.

[37] W. Zhao, Y. Chao, F. Liu, K. Agarwal, D. Acharyya, S. Nassif and K. Nowka, "Rigorous extraction of process variations for 65nm CMOS design", *Proc. IEEE Conference on Solid State Device Research*, 2007, pp. 89-92.

[38] V. Zolotov, J. Xiong, H. Fatemi and C. Visweswariah, "Statistical Path Selection for At-Speed Test", *Proc. IEEE/ACM International Conference on Computer-Aided Design* , 2008, pp. 624-631.

[39] K. A. Ramsey, "Tackling the rising cost-of-test for semiconductor devices", *Solid State Technology* 54(3) (2011).

[40] Synopsys Primetime, http://www.synopsys.com/Tools/Implementation/SignOff/PrimeTime/Pages/default.aspx.

Appendix A: $I_{\rm eff}$ Within Die Variation

We assume every measurement is repeated N_e times and the scribe-line test structure has N_d devices connected in parallel. Only the sum of device currents and capacitance of every chip are measured, i.e., the mean $I_{\rm eff}$, $I_{\rm off}$ and device capacitance per unit width are obtained. The mean of measured $I_{\rm eff}$ for a chip is denoted as $\hat{I}_{\rm eff}$, and it is given as

$$\hat{I}_{\rm eff} = \frac{1}{N_e}\sum_{m=1}^{N_e}\frac{\tilde{I}_{\rm eff}(m)}{N_d} \tag{25}$$

where $\tilde{I}_{\rm eff}(m)$ is the sum of $I_{\rm eff}$ for N_d devices at the m^{th} measurement and N_e is the total number of measurements. Considering measurement noise, $\tilde{I}_{\rm eff}(m)$ can be expressed as:

$$\tilde{I}_{\rm eff}(m) = (1+F_m)\sum_{s=1}^{N_d}[I_{\rm eff}+I_{wd}(s)] \tag{26}$$

where $I_{\rm eff}$ is the exact (unknown) value, I_{wd} is the effect of within die variation, and F_m is measurement noise. Combining (10) and (26),

$$\begin{aligned} I_{\rm eff} &= \frac{\hat{I}_{\rm eff}}{1+\sum_{m=1}^{N_e}F_m/N_e} - \frac{1}{N_d}\sum_{s=1}^{N_d}I_{wd}(s) \\ \because\ &\sum_{m=1}^{N_e}F_m/N_e \ll 1 \\ \therefore\ I_{\rm eff} &\approx \hat{I}_{\rm eff}(1+\sum_{m=1}^{N_e}F_m/N_e) - \frac{1}{N_d}\sum_{s=1}^{N_d}I_{wd}(s). \end{aligned}$$

Since I_{wd} and F are Gaussian random variables, $I_{\rm eff}$ is also a Gaussian random variable with its mean and variance given by

$$\begin{aligned} \mu_{I_{\rm eff}} &= \hat{I}_{\rm eff} \\ \sigma^2_{I_{\rm eff}} &= \frac{\hat{I}_{\rm eff}\sigma^2_{I_{wd}}}{N_d} + \frac{\sigma^2_F}{N_e} \end{aligned}$$

where $\sigma^2_{I_{wd}}$ and σ^2_F are the variance of the within-die variation and measurement noise, respectively.

Appendix B: I_{off} Within Die Variation

Equation (14) shows that we need to know Y_g to estimate total leakage power, which is derived from measurements. As mentioned earlier, we take N_e measurements of the current of N_d devices in test structures. Considering measurement noise and within die variation, the m^{th} measured I_{off} of a given device type t is modeled as

$$\begin{aligned}\tilde{I}_{\text{off}}(m,t) &= \sum_{s=1}^{N_d} I_{\text{off}-\text{nom}}(t)e^{Y_g(t)+Y_{rt}(s,t)}(1+Z_m) \qquad (27)\\ &\approx N_d I_{\text{off}-\text{nom}}\mu_{rt}e^{Y_g(t)}(1+Z_m),\end{aligned}$$

where $\tilde{I}_{\text{off}}(m,t)$ is the sum of I_{off} for N_d devices at m^{th} measurement, Z_m is the m^{th} normalized measurement noise. From (13) and (15), the estimated $Y_g(t)$ is given by

$$\begin{aligned}\hat{Y}_g(t) &= \frac{1}{N_e}\sum_{m=1}^{N_e} ln(\frac{\tilde{I}_{\text{off}}(m,t)}{N_d I_{\text{off}}(nom)\mu_r t})\\ &= Y_g(t) + \frac{1}{N_e}\sum_{m=1}^{N_e} ln(1+Z_m)\end{aligned} \qquad (28)$$

where $Y_g(t)$ denotes the exact value, $\hat{Y}_g(t)$ is the estimated value. Since the normalized measurement noise Z_m is much smaller than 1, (28) can be simplified as

$$\hat{Y}_g(t) = Y_g(t) + \frac{1}{N_e}\sum_{m=1}^{N_e} Z_m, \text{ or}$$

$$Y_g(t) = \hat{Y}_g(t) - \frac{1}{N_e}\sum_{m=1}^{N_e} Z_m$$

From the above equation, we observe that the exact inter-die variation $Y_g(t)$ is a random variable centered at $\hat{Y}_g(t)$. Since Z_m's are Gaussian random variables, $Y_g(t)$ is a Gaussian random variable given $\hat{Y}_g(t)$ is a Gaussian random variable. The mean and variance of $Y_g(t)$ are

$$\begin{aligned}\mu_{Yg}(t) &= \hat{Y}_g(t) \qquad (29)\\ \sigma^2_{Yg}(t) &= \sigma^2_Z/N_e.\end{aligned}$$

Since each $Y_g(t)$ is a Gaussian random variable, $e^{Y_g(t)}$ is a lognormal distribution. From (14), we find that P_{chip} is the sum of lognormal distribution. Thus, we can apply Wilkinson's approach [29] to approximate the sum of lognormal random variables as another lognormal random variable by matching the mean and variance.

Tuck-Boon Chan (http://vlsicad.ucsd.edu) is currently a Ph.D. student of the Electrical and Computer Engineering Department at University of California, San Diego. He received the B.S degree in Electrical Engineering from University Technology Malaysia, in 2003 and the M.S. degree in 2007 from National Taiwan University.

His research work has been focused on mitigating VLSI circuit variability and improving manufacturing yield through design/manufacturing co-optimization.

Puneet Gupta (http://nanocad.ee.ucla.edu) is currently a faculty member of the Electrical Engineering Department at UCLA. He received the B.Tech degree in Electrical Engineering from Indian Institute of Technology, Delhi in 2000 and Ph.D. in 2007 from University of California, San Diego. He co-founded Blaze DFM Inc. (acquired by Tela Inc.) in 2004 and served as its product architect till 2007.

He has authored over 80 papers, ten U.S. patents, and a book chapter. He is a recipient of NSF CAREER award, ACM/SIGDA Outstanding New Faculty Award, SRC Inventor Recognition Award, European Design Automation Association Outstanding Dissertation Award and IBM Ph.D. fellowship. Dr. Puneet Gupta has given tutorial talks at DAC, ICCAD, Intl. VLSI Design Conference and SPIE Advanced Lithography Symposium. He has served on the Technical Program Committee of DAC, ICCAD, ASPDAC, ISQED, ICCD, SLIP and VLSI Design. He served as the Program Chair of IEEE DFM&Y Workshop 2009, 2010.

Dr. Gupta's research has focused on building high-value bridges across application-architecture-implementation-fabrication interfaces for lowered cost and power, increased yield and improved predictability of integrated circuits and systems.

Aashish Pant (http://www.ee.ucla.edu/˜apant) received the B.Tech degree in Electrical Engineering from Indian Institute of Technology, Delhi in 2006 and the M.S degree in Electrical Engineering from University of California, Los Angeles in 2010. He is currently working as an R&D engineer in the Design-to-Silicon division at Mentor Graphics.

His research interests include computer aided design of VLSI circuits and systems, design for manufacturing, lithography and resolution enhancement techniques.

Lerong Cheng received the B.S. degree in electronics and communication engineering from Zhongshan University, Guangzhou, China in 2001, the M.S. degree in Electrical and Computer Engineering from Portland State University in 2003, and the Ph.D degree in Electrical Engineering From University of California, Los Angeles in 2009. He is currently a CAD Engineer in SanDisk Corperation.

His research interests include computer-aided design of VLSI circuits and systems, programmable fabrics, low-power and high-performance designs, and statistical timing analysis.