Yield Enhancement through Pre- and Post-Silicon Adaptation

by

Cheng Zhuo

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2011

Doctoral Committee:

Professor David Blaauw, Chair
Professor Marios C. Papaefthymiou
Associate Professor Dennis Michael Sylvester
Associate Professor Amy Ellen Mainville Cohn

To my family and friends for their love and support

ACKNOWLEDGEMENTS

The past four years of my graduate studies at the University of Michigan have been extremely valuable for me, not only in knowledge but also in spirit. First and foremost I would like to thank my research advisor, Prof. David Blaauw. It has been his understanding and support that helped me grow over the past four years. He is always very helpful, encouraging me to focus on my interests, advising me how to resolve the problem in a more efficient way and inspiring me to address my flaws. Without his guidance and patience, this work would not have been possible. I am very grateful to have him as my mentor and my future career will greatly benefit from such guidance.

I would also like to thank Prof. Dennis Sylvester, who was like my second research advisor throughout my graduate studies and advised me on almost all the topics in this dissertation thesis. I would also like to thank other members of my dissertation committee, Prof. Papaefthymiou and Prof. Cohn for serving on my committee and providing valuable suggestions to improve the quality of this work.

My earnest acknowledgments go to all the members in the CAD group, Kaviraj Chopra, Brian Cline, Ravikishore Gandikota, Vineeth Veetil, Vivek Joshi, Yung-Hsu Chang and Wen Chen for their invaluable help during my formative years here at the University of Michigan. I also thank Johnny Ho and Chien-Chih Yu for being so great group project partners in the VLSI graduate courses.

My years in Ann Arbor have been memorable mostly because of the friends I met here. I will take this opportunity to thank Nan Zheng for always being a very thoughtful and caring friend. I also thank Fu Sun, Cen Guo, Qiang Xu and other Zhejiang University alumni for their encouragement and support. Last but not the least, I thank my parents for always encouraging, guiding and supporting me in life.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Yield Enhancement through Pre- and Post-Silicon Adaptation

by

Cheng Zhuo

Achieving a consistently high yield is always a key design objective. However, circuits designed in aggressively scaled technologies face more stringent design constraints and increased process variability. Traditional guard-band design methodologies that assume worst-case environmental factors and minimum feature size may reach overly conservative decisions and inevitably deteriorate the yield. Hence, design for yield (DFY) in nano-meter regime has become highly imperative for chip designers.

This thesis focuses on several topics in yield enhancement and attempts to answer two basic questions (1) how to achieve a high yield and (2) how to achieve a consistently high yield. The first question is associated with several yield optimization issues. One key issue that complicates parametric yield optimization is the negative correlation among design constraints. In the first part of the thesis, we discuss the power-performance correlation and present a novel yield optimization framework by selecting body bias at design time. The framework considers both inter- and intra-die variation and then explores the possible body bias selection for gates by using a feature extraction technique. The gates with similar features are then grouped together and fed to the optimization framework to maximize the joint power-performance yield. The second and third parts of the thesis discuss the impact of oxide breakdown reliability on yield. For recent technology nodes, chip reliability has become a pressing concern in DFY. This thesis places the focus on oxide breakdown reliability, which is one of the key factors that set constraints on the operating supply voltage of the chip. Any pessimism in oxide breakdown reliability analysis may eventually degrade the yield. We

therefore propose a process and temperature variation-aware method for full chip oxide breakdown reliability analysis. Based on that, we further develop a reliability and performance management scheme by analyzing limited post-fabrication measurements. This post-silicon method helps designers tightly bound the chip reliability and hence enables the use of available margin to boost the system performance while meeting the design lifetime. Since pre- and post-silicon optimization usually targets at the same design objective, it is therefore necessary to perform certain coordination to avoid repeated optimization. In the fourth part of the thesis, we explore the interaction between gate sizing (pre-silicon) and adaptive body biasing (post-silicon) to improve the yield optimization efficiency while maintaining the tunability for a particular target.

It has been observed that even for the same design with exactly the same design optimization, the yield may happen to be inconsistent from lot to lot. A major reason behind this inconsistency is the inability to capture the process variation change during the fabrication. In the last part of the thesis, we address the second question to achieve a consistently high yield by using a dynamic variation extraction model. Unlike the traditional design-time variation model that is static and constructed by measuring hundreds of testing wafers, the proposed post-silicon model is extracted from the measurements of product wafers. The model then dynamically adapts itself to the process change by reusing information from past wafers to validate and improve the model. Such a model is more accurate (or less pessimistic) than a design-time model and also helps reduce the yield inconsistency.

CHAPTER I

Introduction

Accompanied by tremendous advancements in the semiconductor industry over the last few decades, process variation has emerged as a major design issue in nano-scale CMOS technologies [1, 2, 3, 4, 5]. The variation causes certain deviation in device characteristics from its intended value, which is non-trivial in modern design, and eventually leads to chip performance degradation or even functional failure [5, 6]. The worsening variation is due in part to the exponential growth of device integration, as predicted by the Moore's law, with billions of devices per die in 45nm and beyond. Moreover, the ever shrinking feature size and ultra-low supply voltage, as shown in Figure 1.1, also exacerbate the variation and accordingly deteriorate the ability of designers to reliably control the chip behavior [6]. As a result, it has become imperative for designers to develop techniques to well model and control the process variation in the nano-scale process technology.

1.1 Process Variation

Process variation arises from a range of factors throughout the manufacturing processes, like chemical mechanical polishing (CMP), optical proximity effects (OPC), *etc.* [5, 7]. In order to better understand and evaluate the impact of process variation, we can classify the process variation based on the spatial scale over which it manifests. In tradition, a typical classification of variation components is *inter*- and *intra-die* variation [7, 8].

- **Inter-Die Variation**: Due to long range shifts, which occur during the exposure to the mask pattern or when loading the equipment from one wafer to another, all the devices on the same die observe some common amount of fluctuation in the physical parameters [5, 7]. This *die-to-die* variation is typically referred to as *global* or *inter-die* variation. For example, the inter-die variation causes all the devices on the same die observe the same amount of increment

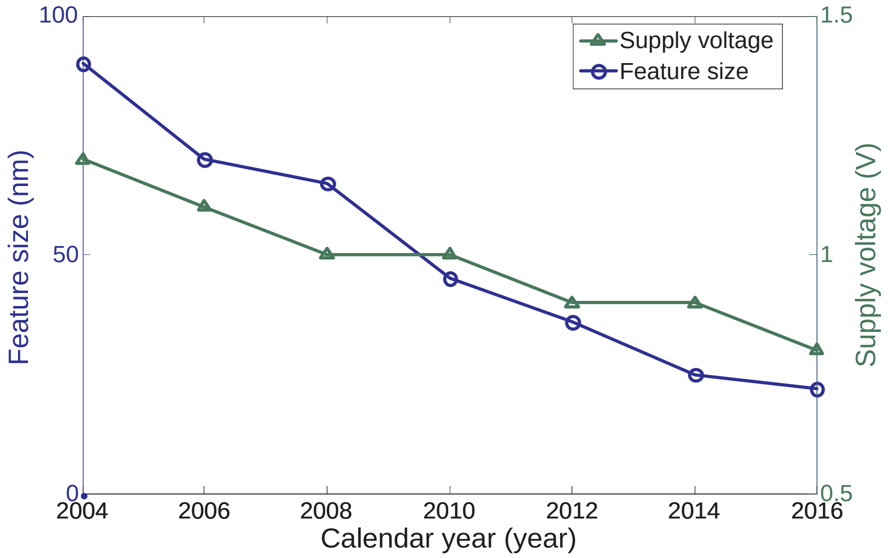


Figure 1.1: Trends for feature size and supply voltage (from ITRS 2007 [6]).

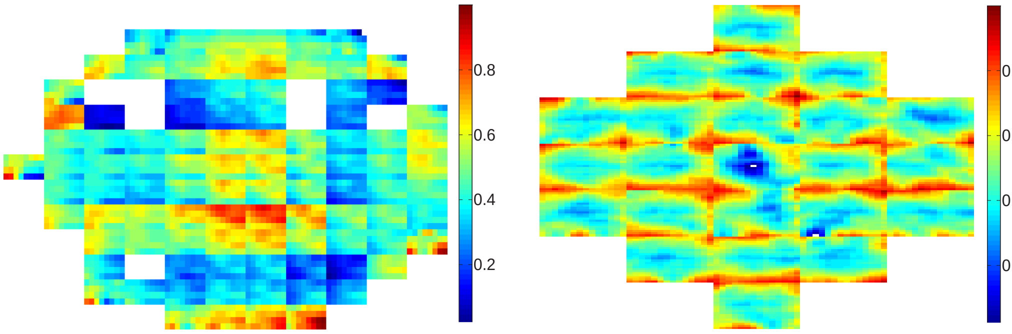
or decrement from the nominal value.

- **Intra-Die Variation**: This variation component affects each device on the same die differently. In other words, different devices may observe different changes on the physical parameters. Specifically, *intra-die* variation can be further classified into *spatially correlated* variation and *independent* variation [7]. Spatially correlated variation tends to affect all the devices that are placed close to each other in a similar manner. Thus, those closely-placed devices are more likely to have similar characteristics than those placed far apart [7, 8, 9]. For accurate analysis, it is necessary to capture the dependence of certain spatial correlation [2]. Unlike the spatially correlated variation, the independent variation is the residual variation component resulting from certain local device scaling effects such as different surface orientations or stress conditions. Typically it is considered statistically independent from all other variation components.

Given the decomposition of global inter-chip, intra-chip spatially correlated and random variation components, the chip level variation model for any device can be modeled as:

$$x = z_0 + z_g + z_{corr} + z_\epsilon \tag{1.1}$$

where z_0 is the nominal design value for the technology. z_g denotes the global-scale inter-chip variation component. Clearly, all the devices on the same chip observe the same amount of z_g in the



(a) (b)

Figure 1.2: Wafer-level contours with systematic patterns removed for (a): RO measurements (scaled) for a wafer in 65nm process; (b): ELM measurements (scaled) for a wafer in 130nm process.

parameter of interest, whereas z_g varies for different chips. The fluctuation of z_g among different chips can then be modeled by a Gaussian process $N(0, \sigma_g^2)$ [8].

z_{corr} is the intra-chip spatially correlated component that tends to affect closely-placed devices in a similar manner. A typical modeling of the vector $\mathbf{z_{corr}} = [z_{corr,1}, z_{corr,2} ... z_{corr,m}]$ for m devices is a multi-variate Gaussian process, *i.e.*, $\mathbf{z_{corr}} \sim \mathcal{N}_m(0, \Sigma_{corr})$, where the subscript of $\mathcal{N}$ denotes the dimensionality of the random vector, and Σ_{corr} is a $m \times m$ covariance matrix for m devices.

Finally, z_ϵ is the random residual variation resulting from certain local device scale effects and is modeled as a Gaussian process $N(0, \sigma_\epsilon^2)$ [8]. In general, σ_g, Σ_{corr} and σ_ϵ denote the uncertainness of the variation components at different spatial-scales, and can be either achieved from prior knowledge or robustly extracted from measurements as in [8, 10].

Some recent works investigate the origins of process and propose that a great portion of within-die spatially correlated variation is actually caused by deterministic across-wafer and across-reticle spatial patterns [11, 12]. By not recognizing systematic patterns at the reticle or wafer level, pessimism is unnecessarily increased, attributing more variation than is actually present [12, 14]. On the other hand, only extracting deterministic global trend but ignoring the non-deterministic spatially correlated variations not only obscures wafer-level trend but also leaves too much unevenly distributed across-reticle variation to the residual. Figure 1.2 demonstrates the wafer-level contours with systematic patterns removed using a similar methodology in [11, 12]. Either for RO frequency in Figure 1.1 from process 1 or ELM in Figure 1.2(b) from process 2, it can be observed the non-uniformity across the wafer and certain spatially correlated patterns within the reticle. Residual variation is supposed to be independent and evenly distributed, which is unable to explain Figure 1.2. Thus, it is necessary to include spatially correlated variations in the model.

Without loss of generality, for a wafer-level variation model, we can denote z as the measurable process parameter of interest, which can be either a physical parameter or a parametric quantity. We model z as a location dependent random variable including seven distinct parts:

$$z(x, y) = z_0 + z_{iw} + z_w(x, y) + z_r(x_0, y_0) + z_{ir} + z_{ar} + z_\epsilon \tag{1.2}$$

z_0 is the nominal design specification and turns out to be a constant for any device. z_{iw} is the inter-wafer variation that captures the long-term drifts in tools and process difference from wafer to wafer. $z_w(x, y)$ is the deterministic across-wafer spatial pattern and (x, y) is the location within a wafer. Such pattern may be caused by post-exposure bake (PEB) temperature non-uniformity, or resist thickness variation. $z_r(x_0, y_0)$ is the reticle-level spatial pattern where (x_0, y_0) is the location within a reticle. This component is primarily due to design-process interactions in the lithography steps, like lens abbreviation. Both $z_w(x, y)$ and $z_r(x_0, y_0)$ are deterministic global patterns. z_{ir} is inter-reticle variation component, which may be caused by the light source change. z_{ar} is across-reticle spatially correlated random variation. For all the devices within one reticle, z_{ar} can be understood as a zero mean multi-variate Gaussian random vector. z_{ar} may be caused by the proximity effect or coma and result in uneven within-reticle contour as in Figure 1.2. Finally, z_ϵ is the independent residual variation caused by local random effect, and typically modeled as an independent Gaussian variable[1]. The only assumption we hold in our model is the variation type of Gaussian, which has been validated by many characterization works [11, 13].

The two models in (1.1) and 1.2 express the process variation at different spatial levels and are used for different application in this dissertations thesis work. We will address more details in the following chapters.

1.2 Design for Yield

In the past few decades, various techniques have been proposed to resolve or mitigate the impact of process variation. All those techniques can be associated with the term of *design for manufacturability* (DFM), which is a broad concept and stands for certain methodology to ensure all the measures for the goal to be reliably and consistently implemented throughout the design, manufacturing, and assembly processes [15, 16].

One measure in DFM is the productivity subject to certain constraints (logical functionality, reliability and performance, *etc.*), which is also classified as *yield* [15]. In other words, yield is the

[1] For measurement results, z_ϵ may also include the measurement white noise.

fraction of functional chips that meet the design target over all the chips manufactured.

In nano-scale technologies, process variation has made the conventional approaches insufficient to achieve desired productivity [4, 17]. A slight difference in process variation modeling/extraction may lead to significant yield difference. The concept of *design for yield* (DFY) is hence widely discussed with a new set of design models/rules/methods to ensure decent yield. Since recent developments in manufacturing techniques have brought tight control of catastrophic defects, the productivity of the manufactured chips that are functional (without logical errors) can be guaranteed. Thus, the focus of this work is placed upon the parametric yield, which is the fraction of chips meeting design target (performance/power/reliability specifications) over the total number of functional manufactured chips [15, 16].

In particular, performance/power all have strong dependence on the process parameters [5]. As process variation has become more prominent, parametric yield happens to show high susceptibility to the variabilities [4]. To address this issue, numerous variation-aware optimization methods have been proposed to mitigate the impact of process variation for yield loss minimization [18, 19, 20, 21, 22, 23, 24, 25, 26]. Based on the stage the optimization is performed, these methods can be roughly classified into *pre*- and *post-silicon* optimization.

- **Pre-Silicon Optimization** is performed at *design stage* to optimize parametric yield before chips are fabricated. The optimization decisions are completely based on a priori process variation knowledge of the technology node and design structure. The decisions are therefore statistically effective for the manufactured dies. However, they are not necessarily ideal for an individual chip to meet the design specifications.

- **Post-Silicon Optimization** is carried out after the fabrication (*run-time*). With certain adaptive tuning techniques, the specific variations (both inter- and intra-die) that occur on a particular die can be mitigated to achieve the design target. Unlike pre-silicon methods, post-silicon methods can tune each chip individually to better meet design specifications at the cost of expensive post-silicon testing overhead.

Noting that pre-silicon optimization and post-silicon adaptability target the same design objective and parameter variability, some recent works [27, 28, 29] coordinate the pre- and post-silicon optimization at design time. Intuitively speaking, those joint optimization methods explore the variation space and use traditional pre-silicon techniques to statistically optimize the design, while certain margin is left for each individual die to perform post-silicon tuning after the actual process condition is verified at testing time.

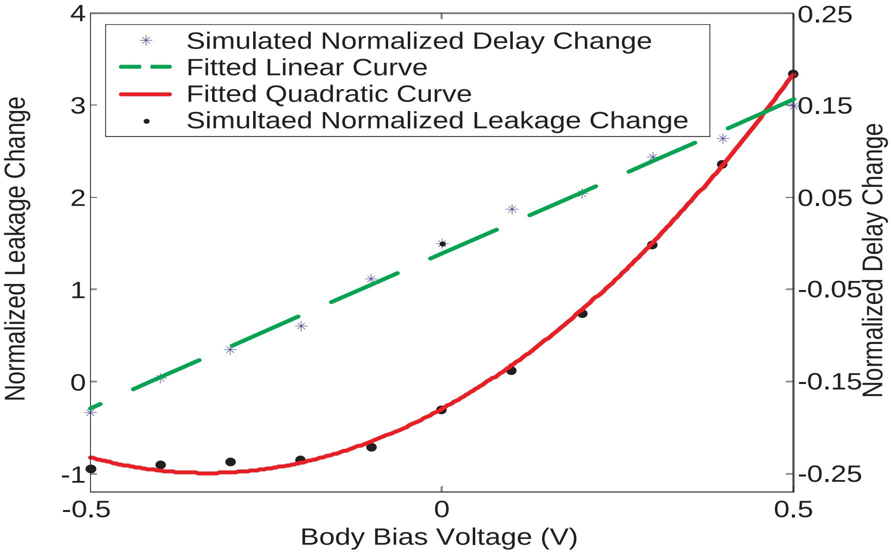


Figure 1.3: Leakage quadratic model and delay linear model

1.3 Adaptive Body Biasing

One most promising post-silicon technique for yield enhancement is adaptive body biasing (ABB) [27, 28, 30, 31, 32]. Body biasing takes advantage of the body effect phenomenon to modulate the V_{th} of a MOSFET [30, 32]. Forward body bias (FBB) can increase the speed but make the device leakier, whereas reverse body bias (RBB) reduces leakage at the cost of an increased delay. The body bias coefficient has been traditionally thought to decrease with process scaling, making ABB less effective in highly scaled process technologies. However, for recent technologies, there is indication that the body bias coefficient is again increasing [31].

Since the analytical expressions that govern the impact of body bias on delay and leakage at the gate level are fairly complex, we employ the quadratic leakage model and linear delay model of the body voltage, as shown in Figure 1.3. The models can achieve an average error of 5.9% and 1.5% in leakage and delay for 90nm process, respectively. The normalized change of the leakage and delay for gate i can therefore be computed by:

$$\delta L_i(v_{b,i}) = \Delta L_i / L_{0,i} = p_{0,i} + p_{1,i} v_{b,i} + p_{2,i} v_{b,i}^2 \tag{1.3}$$

$$\delta s_i(v_{b,i}) = \Delta D_i / D_{0,i} = d_{0,i} + d_{1,i} v_{b,i} \tag{1.4}$$

where $v_{b,i}$ is the body voltage for gate i, $L_{0,i}$ and $D_{0,i}$ are the nominal leakage and delay value,

ΔL_i and ΔD_i are the leakage and delay change, $\delta L_i(v_{b,i})$ and $\delta s_i(v_{b,i})$ are the normalized change of the leakage and delay, $p_{j,i}$'s (j=0, 1, 2) are the fitted parameters for the leakage quadratic model, $d_{j,i}$'s (j=0,1) are the fitted parameters for the delay linear model [27]. The fitted models can greatly simplify the leakage and delay formulation, while maintaining a high accuracy. Those models will be used in our yield enhancement techniques and discussed in more details in Chapter II and Chapter V.

1.4 Oxide Breakdown Reliability

In tradition, DFY techniques focus on the minimization of yield loss caused by power/performance constraints. However, in nano-meter regime, semiconductor reliability like oxide breakdown has emerged as a new challenge. Aggressive oxide thickness scaling has caused large vertical electric fields in MOSFET devices, a situation that makes oxide breakdown (OBD) a crucial issue when supply voltage is not scaled as aggressively as transistor feature size. Gate oxide degradation leads to poor device characteristics resulting in delay and voltage swing degradation, and in increased gate leakage, both of which eventually lead to functional failure and hence yield loss. It therefore becomes crucial to consider the impact of oxide breakdown in DFY methodologies to ensure the reliability of ICs over their lifetime.

Oxide, or dielectric, breakdown is a degradation mechanism that results in a low-impedance path through an insulating or dielectric barrier. Device failures related to this low-impedance path are typically manifested as abnormally high gate leakage current, changes in circuit switching delay, or failure to switch (in severe cases of degradation). Researchers have developed several models to explain the mechanisms of oxide breakdown. A widely used model is the anode hole injection model [33], according to which, injected electrons generate holes at the anode that can tunnel back into the oxide and recombine with electrons to generate electron traps. Another model, known as an electron trap density model, suggests that high gate-oxide electric fields may induce sufficient energy to trigger oxide breakdown [34]. Though researchers are still debating many details of the breakdown mechanisms, we can learn from either model that defect generation is a non-deterministic process.

To handle the breakdown statistics of device-level reliability, a model was proposed by [35] to capture the scaling of breakdown distribution with oxide area but lacked prediction power about the dependence on oxide thickness. This model was then improved and replaced by the percolation model, proposed by R. DeGraeve [36], which generates defects of tunneling charges to model the wear-out for thin dielectric films. In this model, when a critical defect density is reached inside the oxide volume, there is a high probability that a low-impedance defect path ultimately leads to uncontrolled current and oxide breakdown.

However, limited works address the chip-level oxide breakdown reliability due to the difficulty in modeling the inherent randomness of chip failure as well as the tremendous number of devices per chip. Designers have to handle OBD reliability by making worst-case assumptions for each variable. In practice, however, worst-case conditions rarely occur because some devices have thicker oxides than others, or they operate at lower temperatures. Also, because degradation is a cumulative process, periods of low stress lengthen the device's overall lifetime. Therefore, assuming worst-case conditions is conservative, and it leads to overly large reliability margins that designers could otherwise have traded for performance by raising the operating voltage.

1.5 Thesis Overview and Key Contributions

In this research, we focus on parametric yield enhancement through several pre- and post-silicon optimization techniques under different constraints for nano-meter regime VLSI circuits [37, 38, 39, 40, 41, 42]. The focus is placed upon the statistical modeling and analysis of power/preformance/reliablity conditions and then performing circuit-level yield optimization with consideration of these specifications. The outline of the thesis proposal is as follows:

- **Design Time Body Bias Selection for Parametric Yield Improvement**: Achieving high parametric yield is always a key design objective, but is complicated by the correlation between power and performance. Chapter II proposes a novel *design time* body bias selection framework for parametric yield optimization to mitigate the impact of variability while reducing testing costs. The framework considers both inter- and intra-die variations as well as power-performance correlations. The approach uses a feature extraction technique to explore the underlying similarity between the gates for effective clustering. Once the gates are clustered, a Gaussian quadrature based model enables fast yield analysis and optimization. The work also introduces an incremental method for statistical power computation to further reduce the optimization complexity.

- **Process and Temperature Variation-Aware Oxide Breakdown Reliability Analysis**: In aggressively scaled technologies, reliability concerns such as oxide breakdown have become a key issue. Unfortunately, the conventional approaches for full chip oxide breakdown reliability analysis ignore both the process and temperature/voltage variations by simply using worst-case reliability modeling assumptions that significantly degrade the accuracy. In practice, gate-oxide thickness varies from die-to-die and within-die. Thus, as the precision of process control worsens an alternative reliability analysis approach is needed. In Chapter III, we propose a new statistical chip-level oxide breakdown model that takes into consideration both process and temperature/voltage variations. The underlying statistical method uses limited variables to compactly model the problem which is typically defined on a huge sample space of several million devices.

- **Post-Fabrication Measurement-Driven Oxide Breakdown Reliability Prediction and Management**: As discussed, the conventional guard-band methodology for oxide breakdown assumes uniformly thin oxide thickness and results in overly pessimistic reliability estimation that severely degrades the system performance. In Chapter IV, we present the use of limited post-fabrication measurements of oxide thicknesses from on-chip sensors to aid in the chip-level oxide breakdown reliability prediction and quantify the trade-off between reliability margin and system performance. Given the post-fabrication measurements, chip oxide breakdown reliability can be formulated as a conditional distribution that allows us to achieve a significantly more accurate chip lifetime estimation. The estimation is then used to individually tune the supply voltage of each chip for performance maximization while maintaining or improving the reliability.

- **Variation-Aware Gate Sizing and Clustering for Post-Silicon Optimized Circuits**: To coordinate the pre- and post-silicon optimization, in Chapter V, we propose a variation-aware methodology for the simultaneous gate sizing and clustering for post-silicon tuning with adaptive body biasing. The proposed methodology uses an accurate table look-up model and fully explores the interaction between gate sizing and optimal body bias based clustering. In addition, it is suitable for industrial test cases with tens of thousands gates. Our optimization methodology includes a body bias distribution alignment strategy to mitigate the impact of critical gates. In this way, the cluster's body bias voltage is not simply determined by a few critical gates.

- **Active Learning Framework for Post-Silicon Variation Extraction** : Accurate and efficient yield analysis/optimization in the prior chapters may depend on the extracted variation model. In tradition, the model is achieved by measuring tens to hundreds of testing wafers and primarily used for design-time analysis and optimization. However, with the advances of post-silicon techniques, accurate variation model is also highly desired in various post-silicon applications. The accuracy of such post-silicon variation models is greatly improved by incorporating test measurements from each product wafer or die. However, to limit test cost, the number of measurements must be reduced as much as possible. Chapter VI proposes an active learning framework to dynamically extract post-silicon process variation models with tightened variance from measurements. The framework is composed of two stages, active training and model adaptation. Active training collects information and initializes the models to be used for the forthcoming wafers. Model adaptation stage then validates the models and optimally determines the test configuration for partial testing to reduce the test cost.

- **Conclusions and Future Work** : The last chapter concludes the thesis and discusses the possible extensions to the solutions that were proposed in the thesis.

CHAPTER II

Design Time Body Bias Selection for Parametric Yield Improvement

Semiconductor technologies are characterized by trends of ever shrinking feature dimensions and increasing integration density [2]. As a result, process variability has become more prominent in sub-nanometer regime designs, and poses a major challenge to improving circuit performance and reducing leakage [2, 3]. Given the large contribution of leakage power to total power in recent technology nodes, delay and power are now negatively correlated across process corners [3]. In such a scenario, high speed parts are also very high leakage, imposing a well-known two-sided constraint on the feasible region of delay and leakage for parametric yield optimization. This ultimately causes a significant yield loss of manufactured dies in modern integrated circuits [18].

To address this issue, numerous pre- and post-silicon statistical optimization methods have been proposed to mitigate yield loss due to process variability [18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]. However, several of these approaches neglect the correlation between performance and power dissipation by treating the impact of delay and power separately [19, 20, 21, 22]. Works in [23, 24, 25, 26] investigated nonlinear optimization by assuming that gate sizes are continuous and applied either a simplified power yield model to preserve convexity or transformed yield maximization to slack minimization in favor of robust programming. These techniques [23, 25, 26] only mitigate the variation indirectly, rather than performing true yield maximization due to the approximated formulations as well as the lack of consideration of power-performance correlation. Moreover, growing circuit size and the complexity of formulating joint yield optimization further restrict the efficacy of traditional pre-silicon techniques (gate sizing or dual-threshold voltage assignment) in guaranteeing reliable circuit operation with desired parametric yield [18, 23, 24, 25, 26].

Beyond these pre-silicon approaches, several post-silicon techniques have been proposed for design

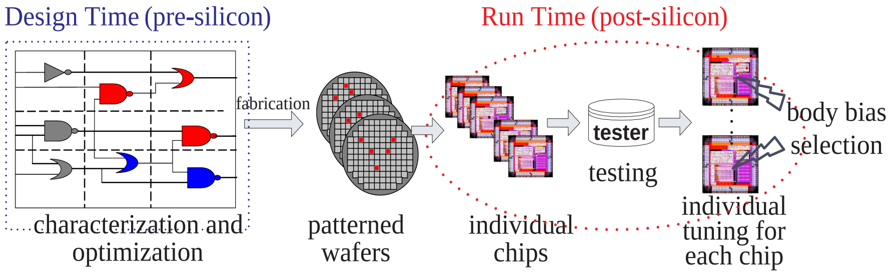


Figure 2.1: Flow of post-silicon ABB tuning employed by [27, 28]

optimization [27, 28, 29]. Among these, adaptive body biasing (ABB) is a promising post-silicon technique due to its flexibility [30]. Traditionally ABB is used to tune each chip individually after chip fabrication and testing, as shown in Figure 2.1. Thus, conventional post-silicon ABB is limited by: (1) routing/control overhead to adjust devices/gates at a very fine grained level [27, 30] and (2) increased post-fabrication testing costs to determine the optimal body voltage. To reduce overhead to a feasible level, [27] presented a heuristic clustering method, in which gates are grouped at design time into a small set of clusters and controlled by one body bias within the cluster. Reference [28] suggested the coordination of pre-silicon (gate sizing) and post-silicon (ABB) techniques, and formulated this as a robust programming problem. Similar to [23, 25, 26], both methods [27, 28] separate the correlation between power and delay and do not evaluate the true parametric yield (joint yield of power and delay). Above all, the body bias tuning in [27, 28] is carried out entirely as a post-silicon step. Clearly, such a strategy incurs large post-silicon testing costs.

In order to reduce testing overhead, this chapter presents a low-cost *pre-silicon* ABB technique for parametric yield maximization, considering process variability and the correlation between performance and power. The major difference between the proposed framework and traditional ABB is that our work does not require individual tuning of each chip during post-silicon testing to select the body bias to be applied. Instead, as shown in Figure 2.2, our framework optimizes and fixes body bias during *design time* to improve the yield of manufactured dies. Once the bias levels are chosen, simple and compact circuits can be readily designed to provide the chosen reference voltages [43, 44]. This overall approach will save significant testing time and cost.

Unlike post-silicon ABB, where bias voltages for each chip are chosen in a deterministic way (since measurement results for a particular manufactured die is known and deterministic), pre-silicon ABB must statistically incorporate the variability during optimization and tune the ensemble of all chips simultaneously. The proposed pre-silicon ABB approach eliminates the testing cost increases associated with the post-silicon approaches. Furthermore, pre-silicon ABB can be easily implemented

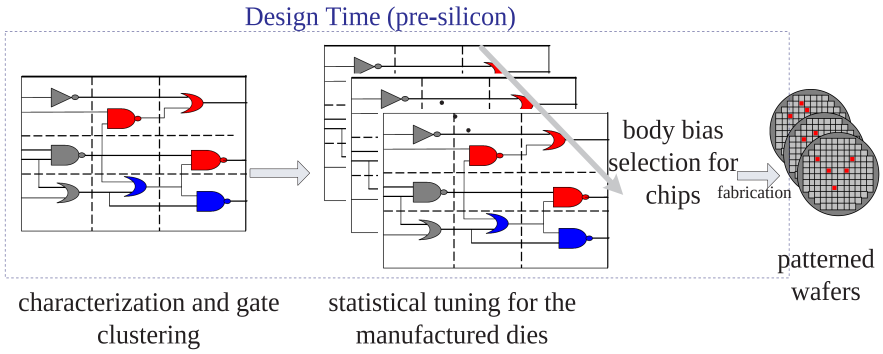

Design Time (pre-silicon)
body bias selection for chips
fabrication
patterned wafers
characterization and gate clustering
statistical tuning for the manufactured dies

Figure 2.2: Flow of the proposed pre-silicon ABB for parametric yield optimization

using on-chip reference voltages [43, 44] and hence has continuous-domain design variables. This appealing feature is in contrast to most traditional pre-silicon techniques such as gate sizing or dual V_{th}, which have discrete domain design variables. Pre-silicon ABB uses a small number of gate clusters (where each cluster is assigned to a different bias voltage), and hence enables a theoretically rigorous formulation of parametric yield as well as scalability to large circuits.

Our framework consists of two phases. We first determine the body bias profiles for each gate, which reflects the preferred body biases across an expected representative set of dies based on process variability models. Then a feature extraction technique is applied to those profiles to efficiently cluster the gates. The general idea behind gate clustering is to group gates with statistically similar behavior. A complete comparison of the profiles is highly inefficient due to the profile and circuit size. The heuristic approach in [27] uses an affine function of mean, standard deviation, and correlation coefficients to determine similarity. In addition to the large runtime and memory consumption for the greedy search and correlation matrix construction steps, the chosen weights of the affine function may not be globally applicable across all circuit topologies. Another issue with this approach is that it discards most information from the original body bias profiles and hence is not robust with respect to outliers. As a result, in our framework we propose a general and scalable clustering method based on feature extraction, without any dependence on empirical parameters. The feature extraction technique projects the original body bias profiles of the gates to a reduced set of features (feature vector) [45]. The feature vectors contain the general characteristics of the profiles and can be computed efficiently for body bias profile similarity comparison. In particular, the comparison is made by computing the distance of two feature vectors and grouping together the gates with closer distance.

After clustering the gates, the second phase formulates the body bias selection problem as a small-sized unconstrained nonlinear programming (NLP). The NLP is solved by a large-scale optimizer, Lancelot [46], with a fast yield evaluation scheme called Gaussian quadrature to compute the objective yield. An incremental method is also introduced to quickly compute the probability density function (pdf) of leakage power. Experimental results show that the proposed framework can optimize a circuit with 14592 gates within 20 minutes to achieve 52 point yield improvement. For eleven circuits of different sizes, parametric yield is improved from 39% to 80% on average. The key contributions are:

- We present a low-cost *pre-silicon* ABB framework to select body bias at design time for direct parametric yield optimization. We show that pre-silicon ABB retains the majority of the yield benefits of more complex die-specific post-silicon ABB approaches. The framework considers both process variations and correlation between performance and power.

- To effectively cluster the gates, a feature extraction based technique is employed. We apply a Haar wavelet transform to extract the features from the statistical body bias profile of each gate. Then a k-median-like algorithm is presented to optimally cluster the gates with similar features.

- In the optimization framework, the yield objective is repeatedly computed. We present a fast and accurate method using Gaussian quadrature to compute the yield in the form of a bi-variate normal integral. An incremental technique for statistical power computation is also introduced to further reduce gradient computation complexity.

2.1 Feature Extraction Based Gate Clustering

Gate clustering is a critical step in practical ABB approaches. Once the clustering is performed, the body voltage of the cluster is determined such that its most timing critical gates meet the overall circuit delay constraint, indicating that most gates in a cluster will end up requiring a larger (more forward) body bias than necessary. It is therefore vital to cluster gates with similar body bias characteristics together to minimize loss of optimality. This section discusses a new feature extraction based technique for gate clustering.

2.1.1 Design Space Exploration

We assume that each circuit constitutes its own unique design space subject to certain parameter variations. Our variation formulation incorporates both the inter-die and intra-die variations [47, 48] and employs the grid based model in [47] to estimate the correlation between the gates. To identify the difference in gates, we need to fully explore the design space to construct the statistical body bias profile for each gate. We first generate multiple "die samples" following certain variations for the given circuit in a Monte Carlo fashion. For each sample circuit we assume each gate can be tuned individually and construct the deterministic quadratic programming (QP) to find the optimal body bias of each gate for leakage minimization [27]:

Minimize

$$\sum_j \Delta L_j(v_{b,j}) \tag{2.1}$$

Subject to

$$AT_s = 0 \tag{2.2}$$

$$AT_t < Target \tag{2.3}$$

$$AT_{i,j} + D_j^{ABB} < AT_{o,j} \qquad for \;\; \forall j \tag{2.4}$$

$$D_j^{ABB} = D_{0,j} - \Delta D_j(v_{bj}) \qquad for \;\; \forall j \tag{2.5}$$

$$lb \leq v_{b,j} \leq ub \qquad for \;\; \forall j \tag{2.6}$$

where AT is the arrival time of the signal on a wire, subscripts "i" and "o" denote input and output, D_j^{ABB} denotes the delay of a biased gate, and lb and ub represent the lower and upper bounds for the body voltage. The first two constraints limit the arrival times at primary input (PI) to be zero and the arrival times at primary output (PO) to be less than the design target, where all primary inputs and primary outputs are tied to super-nodes "s" and "t", respectively. The third and fourth constraints indicate that the delay at the output of each gate should be at least equal to the arrival time at each of its inputs plus the delay of the gate D_j^{ABB}. This QP can be efficiently solved by CPLEX [49] to obtain the optimal body voltage for a particular sample (die) in the design space. Clearly, if the number of samples is sufficient, the histogram of the optimal body bias for a gate sheds insight on the statistical behavior of the gate in this design space. We can then distill information from this histogram in determining which gates should share a common body potential - this is the critical clustering step, which is described next.

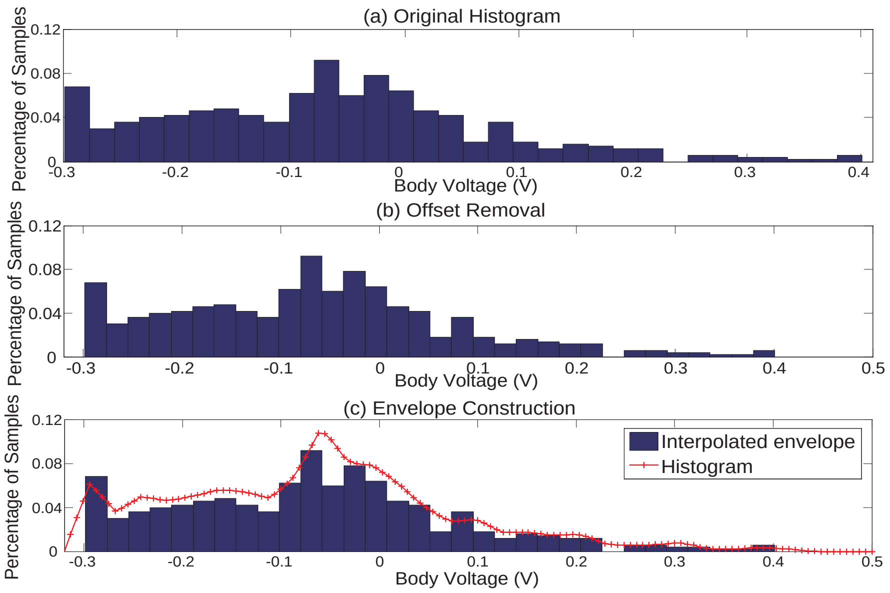


Figure 2.3: Pre-processing procedure: (a) original histogram (b) offset removal for body voltage (x-scale) (c) envelope construction

2.1.2 Feature Extraction

A straightforward approach to clustering is to group together the gates with similar body bias profiles. However, it is difficult to define ”similarity” in a quantitative way. Simply using the complete profiles to cluster the gates is impractical and inefficient, due to both their large sizes and the resulting sensitivity to noise in the distributions.

As stated earlier, [27] suggested a weighted affine function of mean, deviation and correlation to judge the similarity between the gates. However, construction of the correlation matrix between the gates leads to a memory complexity of $O(N^2)$ and limits its applicability. Beyond these runtime concerns, the greedy search is heavily dependent on the carefully chosen weights and the order of the gates to be visited. This makes the method sensitive to outliers and allows gates to be mis-grouped. Furthermore, since the affinity of the non-grouped gate to the cluster is computed by taking the average of the weights, highly deviated data may have a disproportionate impact on the average and lead to poor selections. We therefore present a faster and more robust clustering strategy in our framework.

We employ a pattern recognition technique called feature extraction to obtain the main features of the profile while filtering out noise and redundant information. The intuitive explanation of feature extraction is to extract the general characteristics of the profiles, maintaining the most common information and discarding outliers. The body bias profile for each gate is then uniquely identified

by a feature vector, $\mathbf{v_i} = [x_1, x_2, ..., x_n]^T$ with n features, which are used to measure similarity. To apply the technique across all gates and preserve important information, some pre-processing is performed to build a unified and suitable system. The pre-processing includes two stages:
(1) Offset Removal. This stage simply aligns the histograms to the same body voltage intervals, so that voltage ranges are unified for all the gates.
(2) Envelope Construction. The original histogram is based on a coarse grid and cannot be directly used. In this stage, we apply linear interpolation to map the histogram data to a finer grid and construct the basic shape of the profile envelope. The number of samples is chosen as a power of 2 for feature extraction (128 in our implementation). Figure 2.3(a)-(c) shows the pre-processing procedure for a randomly selected gate in circuit c6288.

Once the body bias profiles are available in the form of unified envelopes, we apply the feature extraction technique to determine the underlying characteristics of each gate. The proposed feature extraction is achieved by Haar wavelet transform, which has low complexity and retains the information in both the original and transformed domains. With a one-level Haar transform, the original body bias waveform with n-sample points can be transformed to two $n/2$-entry vectors (approximation coefficients $\mathrm{x_i}$ and detail coefficients $\mathrm{y_i}$),

$$\mathrm{x_i}[n] = \frac{1}{\sqrt{2}}(\mathrm{x_{i-1}}[2n] + \mathrm{x_{i-1}}[2n+1]) \tag{2.7}$$

$$\mathrm{y_i}[n] = \frac{1}{\sqrt{2}}(\mathrm{x_{i-1}}[2n] - \mathrm{x_{i-1}}[2n+1]) \tag{2.8}$$

where (2.7) is the summation operation denoting low-pass filter and (2.8) is the difference operation denoting the high-pass filter. The detail coefficients representing the local characteristics are easily disturbed by outliers and hence discarded. The approximation coefficients preserving the general characteristics are then decomposed repeatedly until a feature vector with a required number of features ($n/4$, $n/8$, etc.) is obtained. In our work, an 8-entry feature vector is extracted from the body bias profile for each gate. Figure 2.4 shows a simple example of a two-level Haar transform architecture, where $\mathrm{g}[n]$ and $\mathrm{h}[n]$ represent low-pass and high-pass filters, respectively. A 128-entry input $\mathrm{x_1}[n]$ is reduced to a 32-entry vector after a two-level Haar transform.

Since the approximation coefficients indicate the accumulated activities, the feature vectors naturally embody the mean and variance information of the profiles. Moreover, as two highly correlated gates should exhibit similar body bias profiles and hence similar feature vectors, the correlation

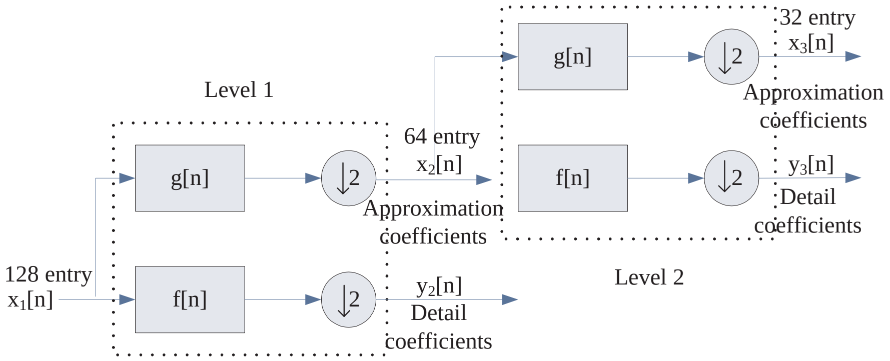


Figure 2.4: An example of a two-level Haar wavelet transform

between gates is well modeled by the proposed method. Thus, the feature vector preserves more information than the method in [27].

2.1.3 Gate Clustering

As mentioned above, a metric must be proposed to quantify the similarity of feature vectors. Thus, we propose the following definition of similarity.

Definition: The similarity of two feature vectors $\mathbf{v_1}$, $\mathbf{v_2}$ is the cosine of the angle between them:

$$S_{\mathbf{v_1},\mathbf{v_2}} = \cos(\alpha) = \frac{|\mathbf{v_1}^{\mathbf{T}}\mathbf{v_2}|}{\|\mathbf{v_1}\|\|\mathbf{v_2}\|} \tag{2.9}$$

where $\|\cdot\|$ denotes the Euclidean norm. The use of the angle between vectors provides two main advantages:

(1) it correctly measures the distance between two vectors. Since any entry in a feature vector is always non-negative, a larger Euclidean distance is equivalent to a larger angle and hence a smaller $S_{\mathbf{v_1},\mathbf{v_2}}$;

(2) the value is normalized and does not depend on any amplitude gains or empirically chosen weights.

Let us consider the simplest example of two clusters. N-cluster decomposition will be an extension of the two-cluster case and is discussed in Section 2.1.4. In this example we need to classify the gates into two clusters based on their feature vectors. The initial seed gates for each cluster may be easily assigned, namely the most forward-biased and most reverse-biased gates, which are determined by sorting the mean of the body bias profiles. These two gates should clearly be in separate clusters. The seeds become the initial *centroids* of the clusters. The *centroid* is defined as a vector that maximizes the sum of similarities of all other points within the cluster to itself. After initial seeds are selected, gates are visited in sequence and their similarities to the centroid of each cluster is computed by (2.9). Each gate is then placed in the cluster with the highest similarity after which the centroid of the corresponding cluster is updated. This procedure is described in Figure 2.5 and carried out repeatedly until all gates are classified.

```
Procedure: 2-Cluster Gate Clustering
Input: feature vectors for all the gates
Output: clustered circuit
Choose the initial seed for each cluster;
For each gate $i$ with feature vector $\mathbf{v_i}$ do
    Measure its similarity to the centroid vector $\mathbf{u}_j$ for cluster $j$
    ($j$=1, 2) by (2.9);
    Find cluster $j = arg\mathbf{max}(S_{\mathbf{v_i},\mathbf{u_j}})$, $j$=1 or 2;
    Put gate $i$ into cluster $j$;
    Update the centroid of cluster $j$;
End for
```

Figure 2.5: Algorithm for 2-cluster gate clustering

When updating the centroid, either the arithmetic mean of the vectors within one cluster or the most representative vector (median) of the cluster can be selected as the centroid. However, noting that the arithmetic mean is not robust to outliers or noise, we propose a low-cost k-median-like algorithm in this paper to compute the centroid. Such a strategy circumvents the potential problem in [27] that the arithmetic average will be skewed by highly deviated data.

Now assume we have m gates in the cluster. A theoretically rigorous formulation for the centroid is:

$$\mathbf{u} = arg\mathbf{max} \sum_{\mathbf{v_i} \in cluster} S_{\mathbf{u},\mathbf{v_i}}, \quad \mathbf{u} \in \{\mathbf{v_1}, \mathbf{v_2}, \dots \mathbf{v_m}\} \tag{2.10}$$

This is a nonlinear discrete optimization problem that is difficult to solve. We therefore employ a two-phase relaxation scheme to tackle this problem. The first phase relaxes the problem to an unconstrained continuous optimization and finds the optimal condition, which is:

$$\mathbf{max} \sum_{\mathbf{v_i} \in cluster} \frac{\mathbf{v_i}^T \mathbf{u}}{\|\mathbf{v_i}\|\|\mathbf{u}\|}, \quad \mathbf{u} \in R^{8\times 1} \tag{2.11}$$

This can be further simplified to:

$$\mathbf{max} \quad \mathbf{w}^T \mathbf{x} \tag{2.12}$$

where $\mathbf{w}$ is $\sum_i \mathbf{v_i}/\|\mathbf{v_i}\|$ and $\mathbf{x}$ is a normalized vector $\mathbf{u}/\|\mathbf{u}\|$. It is well known that the inner product of vector **a** and a normalized vector **b** is the length of projection of **a** on **b**. Thus, the maximum of (2.12) is reached when vectors $\mathbf{x}$ and $\mathbf{w}$ lie in the same direction, *i.e.*, $\mathbf{x} = \mathbf{w}/\|\mathbf{w}\|$. This is denoted as the optimal condition for centroid selection.

The second phase consists of a local search among gates in the cluster to find the closest match to the optimal centroid found above. This is achieved by computing the similarity between each normalized feature vector and the optimal centroid (using (2.9)). The vector with the largest similarity is then chosen to be the centroid of the cluster. The algorithm for centroid update is shown in Figure 2.6.

```
Procedure: Centroid Update
Input: feature vectors for all the gates in the cluster
Output: centroid of the cluster
Compute the optimal condition $\mathbf{v_{opt}}$ of the centroid using (2.12);
For each gate $i$ with feature vector $\mathbf{v_i}$ do
    Compute $S_{\mathbf{v_i},\mathbf{v_{opt}}}$ using (2.9);
End for
Set the gate with the largest $S_{\mathbf{v_i},\mathbf{v_{opt}}}$ as the centroid;
```

Figure 2.6: Algorithm for centroid update

2.1.4 Extension to N Clusters

We now extend the 2-cluster gate clustering algorithm in Figure 2.5 to an efficient successive clustering algorithm for N-clusters. The cluster is recursively bi-partitioned until the number of clusters reaches or exceeds N. The complete algorithm is presented in Figure 2.7. We use a binary-tree data structure to model the successive clustering of the gates. The root node of the tree contains all the gates in the circuit whereas a leaf node represents the resulting cluster without any children nodes.

There are two possible scenarios to create a leaf node: (1) normal termination; when the total number of leaf nodes and non-leaf nodes reaches the required N, all the nodes at the lowest level become leaf nodes; (2) fast termination; for a node with no more than 10% of the total gates [2], we consider this to be a leaf node without further decomposition. A typical example is shown in Figure 2.8(a) for $N = 3$. When the tree is decomposed to the second level, the node on the right with 13 gates is immediately considered to be a leaf without further decomposition, as it contains fewer than 10% of the total gates (fast termination). On the other hand, the node on the left with 153 gates is further decomposed to two nodes with 91 gates and 62 gates on the third level. Since the total number of non-leaf nodes (on the third level) and leaf nodes (on the second level) has reached

[2] In practice the number of the clusters will not exceed 10.

Procedure: *N-Cluster Gate Clustering*

Input: feature vectors for all the gates, number of clusters N
Output: clustered circuit

```
Set $n$=1 as the number of the cluster;
Set $i$=1 as the current level of the binary tree;
While $n < N$ do
    Perform 2-Cluster Gate Clustering for all the tree nodes at level $i$;
    $i = i + 1$;
    For each node $j$ at level $i$ do
        If node $j$ meets fast termination criterion
            Set node $j$ as a leaf node;
            $n = n + 1$;
        End if
    End for
    Set $m$ as the number of non-leaf nodes on level $i$;
    If $n + m \geq N$
        $n = n + m$;
        Break;
    End if
End while;
If $n > N$
    Perform re-combination repeatedly till $n$==$N$;
End if
```

Figure 2.7: Algorithm for N-cluster gate clustering

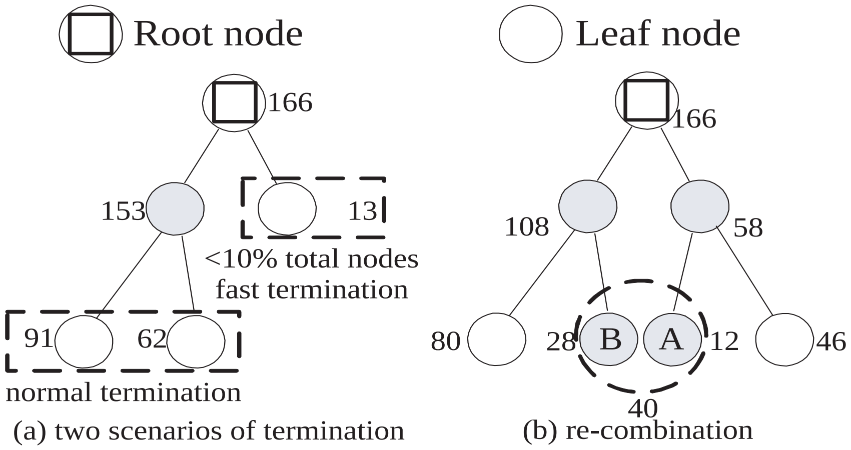


Figure 2.8: (a) Two possible scenarios to achieve leaf nodes: fast termination and normal termination; (b) re-combination of the non-leaf nodes. The number beside each node denotes the number of gates in the node

the required number N (=3), the two non-leaf nodes are then considered as leafs and terminated (normal termination).

If the number of the nodes (including leafs and non-leafs on the bottom level) exceeds "N", which commonly occurs, a re-combination stage is employed. For the example circuit in Figure 2.8(b), the decomposition concludes with 4 nodes on the bottom level while N=3. In this case the node with the fewest gates (node A in the figure) is re-combined to either node A's sibling node or the node whose parent is the sibling of node A's parent. The candidate with fewer gates will be chosen. As shown in Figure 2.8(b), the node A with 12 gates is recombined with node B (having 28 gates), as their parents are siblings and node B has the second fewest gates among the non-leafs. Since the number of clusters is limited in practice, the algorithm in general is terminated within 3-4 iterations.

2.2 Design-Time Body Bias Selection

2.2.1 Statistical Delay and Leakage Models for Body Biased Gates

This section describes the statistical gate-level models for the parametric yield optimization framework. Typically, the delay and leakage of a gate can be expressed as:

$$D = D_0 + \sum_{i=1}^{p} \alpha_p(\Delta P_p) \tag{2.13}$$

$$L = \exp\left(V_0 + \sum\nolimits_{i=1}^{p} \beta_p(\Delta P_p)\right) \tag{2.14}$$

where D_0 and $\exp(V_0)$ are the nominal values of gate delay and leakage; α_p and β_p capture the dependence of gate delay and the log of leakage on the p process parameters of interest. Following the spirit of [47, 48], each process parameter can be transformed to a linear combination of m independent Gaussian random variables (z_j) and the random residual R from principal component analysis (PCA). Both delay and log of leakage can then be canonically expressed by two Gaussian random variables:

$$\begin{aligned} D &= D_0 + \sum\nolimits_{i=1}^{m} a_i z_i + a_{m+1} R \\ \ln(L) &= V_0 + \sum\nolimits_{i=1}^{m} b_i z_i + b_{m+1} R \end{aligned} \tag{2.15}$$

where a_i's and b_i's are the corresponding coefficients obtained from PCA [47, 48]. Now assume the gate is biased at a particular body voltage v_b. With the body bias models in (1.3)-(1.4), gate delay and log of leakage can be written as:

$$D^{ABB} = \Delta k \times D, \quad \ln(L^{ABB}) = \ln(L) + \Delta V \tag{2.16}$$

where $\Delta k = 1 - d_0 - d_1 v_b$ and $\Delta V = \ln(1 + p_0 + p_1 v_b + p_2 v_b^2)$. Timing analysis is then performed by propagating the delay from gate to gate as in [47, 48] with the biased delay model, while maintaining the node delay in the same canonical form with different coefficients. Leakage power analysis is achieved by summing lognormal random variables using Wilkinson's method as in [18]. The efficiency of statistical power analysis is further improved with an incremental approach that will be introduced in Section 2.3.

Since the principal component z_i is an independent standard Gaussian random variable (RV), the correlation between D^{ABB} and $\ln(L^{ABB})$ can be easily evaluated as:

$$\mathrm{Cov}(D^{ABB}, \ln(L^{ABB})) = \sum\nolimits_{i=1}^{m+1} \Delta k a_i b_i \tag{2.17}$$

2.2.2 Yield Analysis and Optimization

Based on the biased gate models, we can perform statistical timing and power analysis and compute the correlation between delay and leakage power. Parametric yield of the circuit is defined as in [18]:

$$\mathrm{Yield} = \Pr(D < D_{con}, \ln(P_L) < \ln(P_{con} - P_D)) \tag{2.18}$$

where D_{con} and P_{con} are constraints for delay and power, respectively, P_L is the leakage power and P_D is the dynamic power of the circuit. Note that both circuit delay and log of leakage are two Gaussian random variables. The underlying problem in (2.18) is then the integral of a bi-variate normal distribution over a rectangular region. The five parameters (μ_D, σ_D, μ_L, σ_L, ρ, which are mean and standard deviation of circuit delay and log of leakage power, and their correlation coefficient, respectively) are used to define the bi-variate normal distribution.

For simplicity, (2.18) can be written as (2.19) after normalization:

$$\mathrm{Yield}= \Pr(x < a, y < b) = \frac{1}{2\pi\sqrt{(1-\rho^2)}} \int\limits_{-\infty}^{a} \int\limits_{-\infty}^{b} e^{-\frac{x^2-2\rho xy+y^2}{2(1-\rho^2)}} dxdy \tag{2.19}$$

where $x = \frac{D-\mu_D}{\sigma_D}$ and $y = \frac{\ln(P_L)-\mu_L}{\sigma_L}$ are normalized random variables, $a = \frac{D_0-u_D}{\sigma_D}$ and $b = \frac{\ln(P_0-P_d)-\mu_L}{\sigma_L}$ are the normalized constraints on delay and log of leakage power, and ρ is the correlation coefficient between the circuit delay and log of leakage. To evaluate this integral, [18] transformed the original rectangular region to a triangular region. The new region is then partitioned into several sub-domains and computed in sequence. This method can suffer from inaccuracy and a high complexity of transformation and partitioning. To avoid these problems we propose the use of the Gaussian quadrature technique [50]. Gaussian quadrature is an efficient approach to compute integrals by a weighted sum of function values at specified abscissae within the domain of integration, and can reach analytical accuracy by a suitable choice of abscissae and weights. Reference [50] suggests a Gaussian quadrature model to compute the integral $\int_0^\infty \exp(-x^2)f(x)dx$:

$$\int\limits_0^\infty \exp(-x^2)f(x)dx \approx \sum\nolimits_{i=1}^{15} w_i f(x_i) \tag{2.20}$$

where x_i's and w_i's are abscissae and weights that are fixed for the integral of the form above without any dependence on $f(x)$. As is proved in [50], (2.20) can reach an accuracy of 10^{-31}, and is therefore adopted in this paper to evaluate the yield function in (2.19).

With the substitution $u = \frac{(a-x)}{\sqrt{2(1-\rho^2)}}$, $v = \frac{(a-y)}{\sqrt{2(1-\rho^2)}}$, $a_1 = \frac{a}{\sqrt{2(1-\rho^2)}}$ and $b_1 = \frac{b}{\sqrt{2(1-\rho^2)}}$, (2.19) can be simplified to:

$$\text{Yield} = \frac{\sqrt{(1-\rho^2)}}{\pi} \int_0^\infty \int_0^\infty \exp(-u^2 - v^2) Y(u,v) du dv \tag{2.21}$$

where

$$Y(u,v) = \exp[a_1(2u - a_1) + b_1(2v - b_1) + 2\rho(u - a_1)(v - b_1)] \tag{2.22}$$

By applying the model in (2.20) to u and v separately, we obtain:

$$\text{Yield} = f(a_1, b_1, \rho) = \sum\nolimits_{i=1}^{15} \sum\nolimits_{j=1}^{15} w_i w_j Y(x_i, x_j) \tag{2.23}$$

Since x_i and w_i in (2.23) are fixed for any arbitrary function $Y(u,v)$ [50], the computation time of (2.23) is independent of the problem size.

Based on the proposed yield analysis [3], our yield optimization problem can be formulated as an unconstrained optimization problem where the objective function is (2.18) and the design variables are the body voltage of each cluster, as shown below:

$$\mathbf{max} \quad \Pr(D < D_{con}, \ln(P_L) < \ln(P_{con} - P_D)) \tag{2.24}$$

This optimization problem is then solved by the optimizer Lancelot [46]. Note that Lancelot numerically evaluates the objective function and gradient of the yield. Thus, the optimization formulation

[3] The model in (2.20) requires that when ρ <0, which is the typical case for log of leakage and delay, the constraints should be $a \leq 0$ and $b \leq 0$. The other constraint cases, $\{a \geq 0, b \geq 0\}$, $\{a \leq 0, b \geq 0\}$ and $\{a \geq 0, b \leq 0\}$, can be easily transformed to $\{a \leq 0, b \leq 0\}$ by exploiting the underlying characteristics of bi-variate normal pdf, *e.g.*, $\Pr(x \leq a, y \leq b) = \Phi(a) + \Phi(b) - 1 + \Pr(x \leq -a, y \leq -b)$, where $\Phi(x)$ is the standard normal cumulative density function [50, 51].

in this section can use high-order models or even table-look-up to compute the intrinsic gate delay and leakage so as to guarantee the accuracy in optimization.

2.2.3 Gradient Computation and Complexity

Lancelot [46] requires the computation of the gradient of yield with respect to the body voltage of each cluster. This can be estimated by increasing or decreasing the body voltage of a cluster by a small amount and then computing the yield difference due to the body voltage change. To improve the efficiency of this step, we suggest a power perturbation scheme instead of a full-circuit statistical power analysis.

Assume that the body voltage for a cluster k is changed by a small amount Δv. The change in leakage power can then be written as:

$$\begin{aligned}\Delta P_L = &\sum_{i \in clusterk} \{P_{i,0}[1 + p_{i,0} + p_{1,i}(v_{b,k} + \Delta v) + p_{2,i}(v_{b,k} + \Delta v)^2] \\ &- P_{i,0}(1 + p_{i,0} + p_{1,i}v_{b,k} + p_{2,i}v_{b,k}^2)\}\end{aligned} \tag{2.25}$$

where $P_{i,0}$ is the leakage with zero-body bias for gate i, and $p_{1,i}$ and $p_{2,i}$ are the coefficients for the leakage model in (1). This can be further simplified to:

$$\Delta P_L = \sum_{i \in clusterk} P_{i,0}(p_{1,i}\Delta v + p_{2,i}\Delta v^2) + v_{b,k} \sum_{i \in clusterk} 2P_{i,0}p_{2,i}\Delta v \tag{2.26}$$

Since the body voltage increment Δv is fixed for all the clusters, $v_{b,k}$ is the only variable in (2.26), so that the coefficients $\sum\limits_{i \in clusterk} P_{i,0}(p_{1,i}\Delta v + p_{2,i}\Delta v^2)$ and $\sum\limits_{i \in clusterk} 2P_{i,0}p_{2,i}\Delta v$ can be computed in advance and used throughout the whole optimization process. In other words, we just need to perform N summations to compute the change in the leakage pdf for N clusters in gradient computation. The complexity is reduced from $O(NN_g)$ for N full statistical power analysis to $O(N)$, where N_g is the number of gates.

Timing perturbation is performed by a full statistical static timing analysis (SSTA). Once we obtain the delay and leakage-power pdfs of the perturbed circuit (the body voltage of the k_{th} cluster is changed from $v_{b,k}$ to $v_{b,k} + \Delta v$), the yield of the perturbed circuit can be calculated by (2.23), and the change in yield is used to define the particular component of the yield gradient. Since yield analysis has a constant complexity, the overall algorithm complexity of this optimization framework is dominated by SSTA, the complexity of which is $O(N(N_g + E))$, where E is the number of edges of the timing graph. Note that the number of clusters is limited in real designs and is negligible

Table 2.1: Clustering efficiency comparison between the proposed feature extraction-based clustering method and the empirical affine weighting function based clustering method from [27]

CKT.	#gates	Leakage Comparison (μW)				Time (sec) for Clustering	
		Empir.[27]		Feature			
		u/σ	95%	u/σ	95%	Empir.	Feature
c432	166	3.8/1.3	6.2	3.5/1.2	5.6	0.8	5.9
c499	519	18.2/7.1	31.7	17.8/6.3	28.3	5.3	6.5
c880	390	4.2/1.6	7.0	3.9/1.5	6.4	4.2	6.3
c1355	558	15.7/5.2	25.0	14.8/4.5	22.1	8.6	6.7
c1908	432	8.9/3.1	14.2	7.8/2.3	12.1	7.5	6.5
c2670	964	8.5/3.3	14.7	7.5/2.8	12.7	26.0	7.4
c3540	962	14.9/6.1	26.7	14.2/5.7	24.5	25.7	7.4
c5315	1750	19.7/7.6	35.6	17.7/7.1	31.4	84.3	9.2
c6288	2502	89/35	155	82/30	134	179	11
c7552	2102	23/10	42	20/8	35	122	10
Vit1	14539	246/110	396	210/80	348	901	52
Average improvement (%)				10/17	13.4		

compared to N_g. Thus, the framework maintains a linear complexity [4]. Experimental results in Section 2.3 also validate that yield optimization takes only seconds even for a circuit with tens of thousands of gates.

2.3 Experimental Results

The proposed algorithms discussed in Section 2.1 and 2.2 were implemented in C and tested on ISCAS85 benchmark circuits and a Viterbi Decoder circuit (Vit1) that vary in size from 166 to 14539 gates. The circuits were synthesized using an industrial 1.2V 90nm triple-well dual-V_{th} technology. The two V_{th} values are 0.32V (0.33V) and 0.22V (0.24V) for NMOS (PMOS). Body voltage is varied between ±0.5V. All standard cells in the library were characterized (using SPICE) at both the high- and low-V_{th} values. Only channel length variation is considered for simplicity, however the overall approach can be extended to consider other sources of variability. We consider inter-die, spatially correlated intra-die, and random components of variation. Total $3\sigma/u$ channel length variability is set to 15% and then split evenly among the three variation components.

2.3.1 Efficacy of Feature Extraction-Based Clustering

Reference [27] proposed a clustering algorithm based on an empirical affine weighting function. Table 2.1 compares the proposed feature extraction-based clustering algorithm (Feature) with the work of [27] (Empir.) in terms of both resulting leakage and runtime. Column 2 lists the number of

[4] The number of Lancelot iterations (around 30) is limited due to the small problem size.

gates for each circuit, varying from 166 to 14539 gates. Columns 3-6 compare the mean/standard deviation and 95$^{\text{th}}$ percentile leakage of the proposed method and the method from [27], respectively. The proposed approach improves upon the prior work in all measures and achieves a 10% and 17% reduction in the mean and standard deviation of the leakage, respectively. The last two columns of Table 2.1 compare the gate clustering runtime for the two methods. The runtime for the proposed method shows linear dependence on circuit size with a small slope, whereas the runtime for the empirical function-based method [27] increases exponentially. On average, the proposed method is 5.1× faster than the method in [27]. For the largest circuit Vit1, the proposed method achieves 18× speed-up.

2.3.2 Monte Carlo Convergence

The design space exploration step described in Section 2.1.1 is executed only once in the framework but still involves solving a large number of QP problems to determine the body bias profile across process variability. To speed this step up, we employ the smart sampling approach in [52], which captures the importance of the samples to reduce the number of the samples. Figure 2.9 shows the dependence of yield optimization results on the number of Monte Carlo samples for the six largest circuits in our set of benchmarks. The quality of the yield optimization results with 100 samples is similar to the results with 1000 samples. We therefore use 100 samples in the exploration step, as design space exploration is only required to outline the general features instead of local details. Moreover, since the QP optimization for a given sample of a circuit is independent of the QP optimizations for other samples of the same circuit, this step can be easily parallelized to achieve further speedup.

2.3.3 Pre-Silicon Body Bias Selection Framework vs. Post-Silicon ABB

The proposed pre-silicon body bias selection framework chooses statistically optimal body voltages for the full ensemble of chips, while post-silicon ABB uses measurement results for a particular manufactured chip and deterministically selects the bias voltage for each cluster of that chip. It is clear that post-silicon ABB should provide higher yields at the cost of higher testing times and costs. This section quantifies the yield loss when using the proposed pre-silicon approach compared to a post-silicon ABB with the same clustering method in Section 2.1.

Given a clustering, the yield of post-silicon ABB is computed by first generating 1000 chip samples, which are then individually tuned to minimize leakage subject to a delay constraint. The number of chips that fail to simultaneously meet the leakage and delay targets is then calculated.

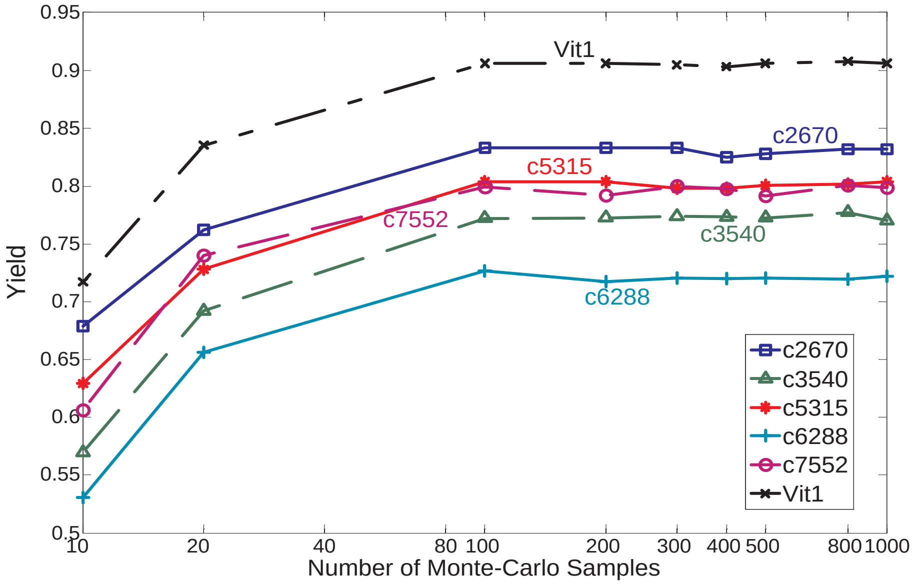


Figure 2.9: Monte Carlo convergence

The yield optimization results of our pre-silicon approach and post-silicon ABB are summarized in Table 2.2. Column 2 lists the initial pre-optimized yield of each circuit for the target constraint {Delay<μ_D+σ_D, Leakage<μ_L}. Columns 3-10 display the yield optimization results and yield point improvement using our pre-silicon framework and post-silicon ABB for 2 and 3 cluster scenarios. Although post-silicon ABB achieves slightly higher yield than the proposed pre-silicon body bias selection framework, the difference degrades for larger number of clusters and larger circuits. For circuit Vit1 with 15K gates, the optimized yield difference between pre- and post-silicon approaches is only 4.7% for 2-cluster configuration and 0.9% for 3-cluster configuration. Furthermore, while testing costs increase rapidly with the number of clusters in post-silicon ABB, the only overheads related to cluster count in our pre-silicon body bias selection framework are physical design related, and are shared by both techniques (see Section 2.3.6 below).

2.3.4 Pre-Silicon Body Bias Selection Framework vs. Traditional Pre-Silicon Approaches

We further evaluate the efficacy of our pre-silicon framework in Table 2.3 when compared to traditional pre-silicon methods (a statistical dual-V_{th} assignment approach [21] and a yield maximization approach using gate sizing [18]). Similar as Table 2.2, columns 3-10 list the yield optimization results and yield point improvement for the constraint {Delay<μ_D+σ_D, Leakage<μ_L} using our pre-silicon framework and two traditional pre-silicon statistical optimization methods (dual-V_{th} and gate sizing

Table 2.2: Comparison of yield optimization results using the proposed pre-silicon body bias selection framework and post-silicon ABB with different number of clusters (2 clusters and 3 clusters)

CKT.	Initial yield (%)	Optimized yield(%)/Yield point improvement							
		pre-silicon BB select				post-silicon ABB			
		2 cluster		3 cluster		2 cluster		3 cluster	
c432	38.4	70.5	32	79.2	41	78.2	40	88.8	50
c499	39.2	65.8	27	74.7	36	77.0	38	82.4	43
c880	38.6	68.7	30	79.1	41	78.4	40	86.4	48
c1355	39.3	68.4	29	82.7	43	81.6	42	89.2	50
c1908	39.0	68.1	29	80.6	42	78.0	39	87.0	48
c2670	38.7	73.3	35	83.1	44	83.8	45	90.7	52
c3540	38.7	65.7	27	77.7	39	75.8	37	84.1	45
c5315	39.2	69.4	30	80.7	42	78.6	39	88.2	39
c6288	38.4	63.4	25	72.1	34	71.1	33	77.8	39
c7552	38.8	69.4	31	80.1	41	76.7	38	86.9	48
Vit1	39.1	78.6	40	90.6	52	83.3	44	91.5	52
Ave. improvement		30		41		40		48	

Table 2.3: Comparison of yield optimization results using the proposed pre-silicon body bias selection framework and traditional pre-silicon approaches (dual-V_{th} [21] and gate sizing [18])

CKT.	Initial yield (%)	Optimized yield(%)/Yield point improvement							
		pre-silicon BB select				dual-V_{th}[21]		sizing[18]	
		2 cluster		3 cluster					
c2670	38.7	73.3	35	83.1	44	39.5	0.8	46.3	7.6
c3540	38.7	65.7	27	77.7	39	41.4	2.7	44.1	5.4
c5315	39.2	69.4	30	80.7	42	40.3	1.1	45.0	5.8
c6288	38.4	63.4	25	72.1	34	38.8	0.4	43.2	4.8
c7552	38.8	69.4	31	80.1	41	40.2	1.4	47.9	9.1
Vit1	39.1	78.6	40	90.6	52	50.7	11.6	52.8	13.7
Ave. improvement		30		41		3.0		7.7	

[21, 18]). The proposed approach with either 2 or 3 clusters potentially doubles the original yield of 39% (the optimized yield is 69% for 2 clusters and 80% for 3 clusters on average) while the yield improvement is limited to 3.0 point on average for the statistical dual-V_{th} package [21] and 7.7 point on average for the method using gate sizing [18]. This further validates the statement in Section I that the proposed pre-silicon body bias selection has continuous domain design variables and hence higher flexibility than the traditional pre-silicon approaches like gate sizing or dual-V_{th}.

2.3.5 Yield Analysis and Optimization

In this section we display the accuracy and optimality of the proposed pre-silicon framework in Table 2.4 when compared to Monte Carlo simulation. Columns 2-5 list the mean and standard devi-

Table 2.4: Yield analysis and optimization results for the constraint {Delay<μ_D+σ_D, Leakage<μ_L} using the proposed method with different number of clusters (2 clusters and 3 clusters), a statistical dual V_{th} method [21] and Monte Carlo approaches

CKT.	Initial Design				Initial Yield (%)		Optim. Yield(%)/point impro.				MC-sweep (%)	
	μ_D	σ_D	μ_L	σ_L	Prop.	MC	2 cluster		3 cluster		Yield	Error
c432	0.74	0.05	0.97	0.26	38.4	43.6	70.5	32	79.2	41	82.5	3.9
c499	0.68	0.04	3.80	1.02	39.2	41.9	65.8	27	74.7	36	76.2	1.9
c880	0.77	0.05	1.17	0.32	38.6	42.6	68.7	30	79.1	41	78.8	0.3
c1355	0.89	0.05	3.57	0.96	39.3	43.8	68.4	29	82.7	43	79.4	4.1
c1908	1.15	0.07	2.16	0.58	39.0	42.1	68.1	29	80.6	42	79.2	1.8
c2670	0.77	0.05	2.06	0.56	38.7	43.4	73.3	35	83.1	44	84.3	1.4
c3540	1.22	0.07	3.15	0.84	38.7	44.5	65.7	27	77.7	39	83.3	6.7
c5315	1.12	0.07	4.14	1.11	39.2	42.2	69.4	30	80.7	42	79.4	1.7
c6288	3.52	0.21	14.77	3.91	38.4	41.1	63.4	25	72.1	34	76.4	5.6
c7552	1.28	0.08	4.14	1.10	38.8	39.7	69.4	31	80.1	41	75.8	5.6
Vit1	2.41	0.14	149.44	39.54	39.1	41.8	78.6	40	90.6	52	91.6	1.0
Average yield point improvement							30		41			

Table 2.5: Runtime for critical stages of the proposed framework and total time

CKT.	#gate	Time for Critical Stages(sec.)			Total Time (sec.)	Ratio
		Explo.	Clust.	Optim.		
c432	166	0.68	5.89	2.35	36.11	0.22
c499	519	5.87	6.48	2.40	47.29	0.09
c880	390	3.29	6.32	2.48	41.05	0.11
c1355	558	6.43	6.64	2.68	46.67	0.08
c1908	432	4.13	6.46	3.28	48.82	0.11
c2670	964	18.38	7.38	3.13	64.78	0.07
c3540	962	23.50	7.36	3.37	73.53	0.08
c5315	1750	68.17	9.21	4.03	133.77	0.08
c6288	2502	185.2	10.5	5.0	257.9	0.10
c7552	2102	110.3	9.8	4.3	173.4	0.08
Vit1	14539	747.3	51.9	12.2	1176.5	0.08

ation of the delay and leakage for the initial designs. The target constraint is set to {Delay<μ_D+σ_D, Leakage<μ_L}. Given this constraint we compute the original yield and compare it with a Monte Carlo golden model with 2000 samples, shown in Columns 6-7. The absolute errors of the proposed yield analysis in Section 2.2.2 vary from 0.9% to 5.7%, which is due to the computation approximation in SSTA, *e.g.*, statistical maximization operation.

The optimized yield results and the yield point improvements are shown in Columns 8-11. The proposed approach with either 2 or 3 clusters potentially doubles the original yield of 39% (the optimized yield is 69% for 2 clusters and 80% for 3 clusters on average). The improvements are consistent among all the benchmarks studied. We also perform a Monte Carlo sweep (MC-sweep) to determine whether the optimized yield obtained by the proposed framework is globally optimal. MC-sweep performs Monte Carlo simulations on all possible combinations of body voltages for a

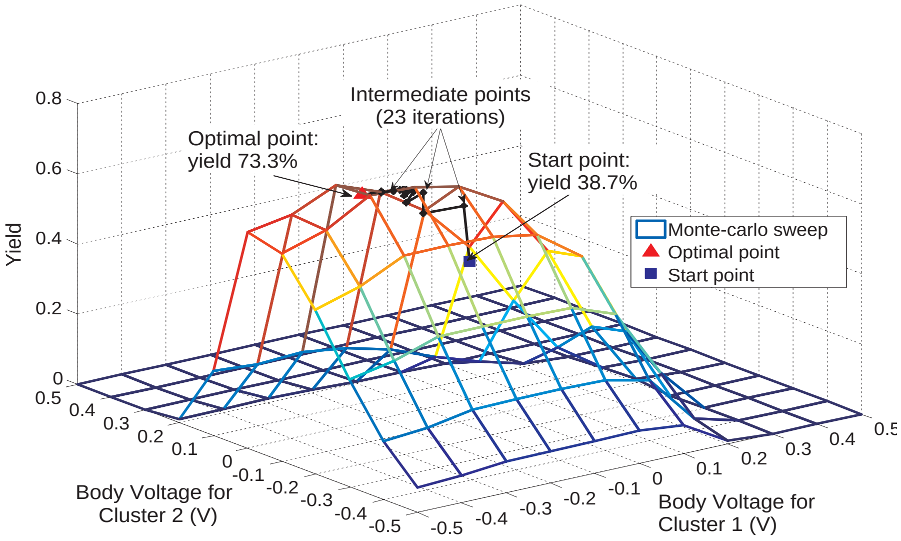


Figure 2.10: Yield optimization trace for c2670.

3-cluster-configuration. The sweep increment is set to 0.1V, *i.e.*, the sweep space is $v_{b,1} \times v_{b,2} \times v_{b,3}$, where $v_{b,i} \in [-0.5, -0.4, ..., 0.5]$ for $i = 1, 2, 3$. The last two columns of Table 2.4 show the maximum yield found by MC-sweep and the relative deviation of the proposed approach with respect to MC-sweep. The maximum deviation is 6.7%, which is due to the model approximation and the relatively coarse grid of the Monte Carlo sweep.

Table 2.5 displays the runtime for the critical stages of the proposed framework (including design space exploration (Explo.), clustering (Clust.) and yield optimization (Optim.)) as well as the total runtime. The last column lists the ratio of the total runtime to the circuit size, indicating a linear relationship. Runtime is dominated for larger circuits by the design space exploration stage, which can be parallelized across machines as mentioned above.

Figure 2.10 shows the iteration trace of Lancelot for circuit c2670 under a 2-cluster configuration. The optimizer requires 23 iterations to reach global optimality, which helps explain the small runtime for yield optimization in Table 2.5. Figure 2.11 illustrates the yield contour of c3540 for a 2-cluster configuration. We use the proposed framework to explore the interaction between yield and constraints. The contour clearly illustrates the equivalent constraint sets to achieve a similar yield and may help designers make beneficial trade-offs.

2.3.6 Implications for Physical Design

Adaptive body bias incurs physical design overheads, including generation/distribution of the body voltages and extra well spacing. Note that there are a limited number of clusters (2-3 in our

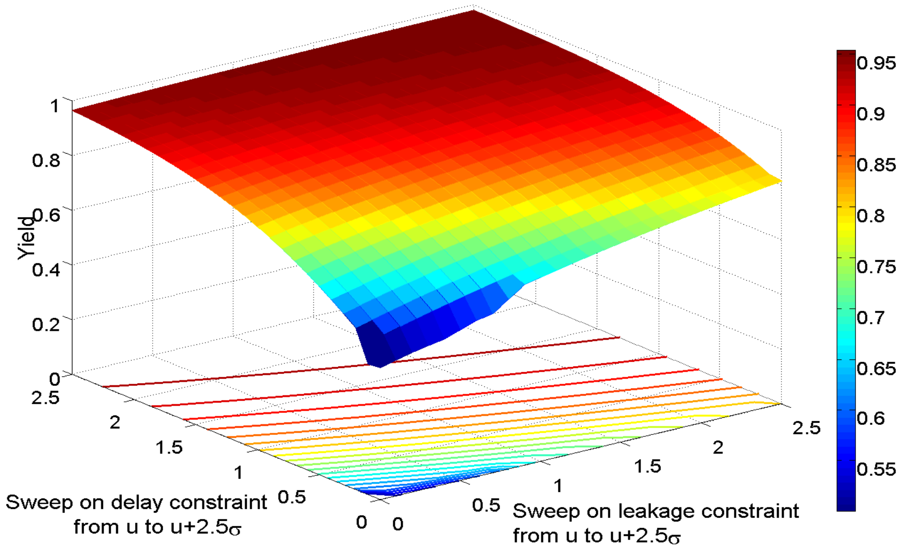


Figure 2.11: Yield contour for c3540 under 2-cluster configuration

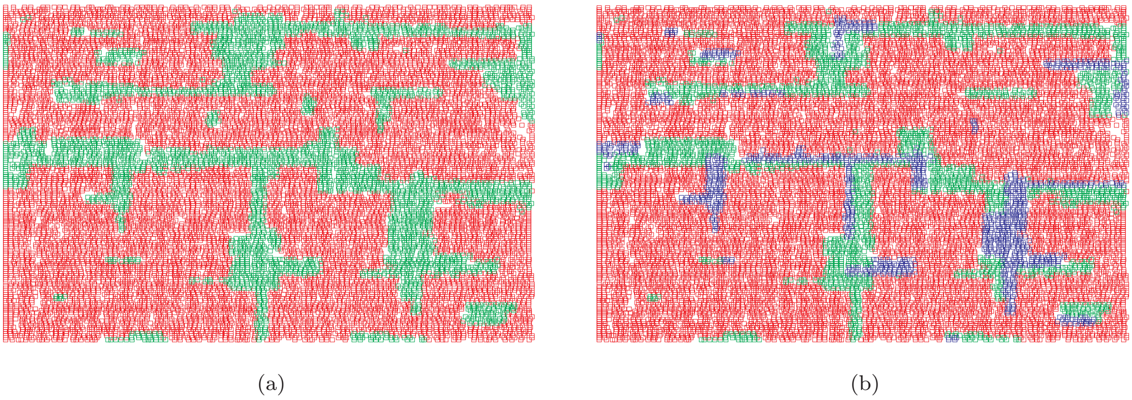


Figure 2.12: Vit1 placement with physically contiguous cluster regions by CAPO [53] (different clusters are shown with different colors): (a) 2 clusters; (b) 3 clusters

experiments) and as a result, these overheads can be reasonably bounded. The major impact of gate clustering on placement is then the extra well spacing between adjacent cells having different biases imposed by triple-well-layout rules. As stated in Section 2.1, the proposed clustering method naturally captures the spatial correlation in the feature vectors. In other words, most gates are intrinsically clustered within the physically continuous regions and this helps reduce the well spacing overheads. Moreover, we employ the incremental placer CAPO [53] to minimize the gate displacement and area overhead, following a similar flow as in [27]. CAPO works in its Engineering Change Order (ECO) mode to make limited changes to the initial placement subject to certain constraints [53]. Figure 2.12 demonstrates the resulting layout for the Vit1 circuit with both 2 clusters and

3 clusters after applying CAPO to the initial placement. It is clear that most gates in the layout are clustered in the physically continuous regions. In particular, the average gate displacement is 2.2-2.3% and half perimeter wire-length increase is 2.7-3.9% compared with the initial designs, for 2- and 3-cluster configurations.

2.4 Summary

In this chapter we presented a gate-level parametric yield optimization framework using *design time* body bias selection. The approach considers the power and performance constraints along with their correlation. A feature extraction based clustering approach is proposed that achieves speedups of 5.1× on average and up to 18× for 11 benchmark circuits compared to a recently reported clustering strategy, with leakage savings of more than 10%. In addition the framework employs a fast yield analysis calculation method and an efficient power perturbation technique for optimization and achieves 41% yield improvement on average across 11 benchmark circuits.

CHAPTER III

Process and Temperature Variation-Aware Oxide Breakdown Reliability Analysis

Over the last few decades, numerous publications have focused on understanding and modeling the mechanisms leading to defect generation and breakdown in individual devices [54, 55, 56, 57]. Some researchers have initiated an effort to understand the oxide breakdown (OBD) mechanisms of simple circuits [56]. Recently, a product level approach performing oxide breakdown analysis on full chip was proposed in [58]. In most of the existing approaches, simple test structures such as discrete devices or capacitors are used to characterize the oxide breakdown mechanism for a specific manufacturing process. These discrete device characterization results are then extrapolated to deduce a model for the full chip oxide reliability which is later calibrated using lifetime tests of sample product.

However, there are two major concerns with prior approaches:

- Prior approaches assume a uniform oxide thickness for all devices on every chip. In practice, the non-uniformity in temperature and pressure during the gate-oxidation process leads to within-die and die-to-die variations in gate oxide thickness. For a given supply voltage and operating temperature, the reliability of oxide is an exponential function of its thickness and its sensitivity to thickness variations increases for thinner oxides [59]. Therefore, in previous approaches, it was imperative to consider a uniform minimum oxide thickness across all devices on a chip and across all chips for a conservative worst-case analysis. This may lead to significantly pessimistic estimates of the overall oxide breakdown reliability of the product.

- In addition, prior works assume a worst operating temperature across the chip and throughout the lifetime. However, it has been well noted that devices operated at different temperature may deteriorate at different rates, and the mean time to failure (MTTF) of a device exponentially depends on temperature [60, 61, 62]. Thus, it is overly pessimistic to assume a device in inactive regions is as equally prone to failure as the one in hot spots.

Since oxide reliability is one of the key factors that sets constraints on the operating supply voltage and temperature of the chip, any pessimism in oxide reliability analysis limits the maximum operating voltage and thus the maximum achievable chip-performance [63, 64]. In order to find consistent supply voltage limits, it is therefore critical to quantify the product oxide breakdown strength with consideration of both process and temperature variations.

The goal of this work is to develop a new chip-level gate oxide breakdown analysis while accommodating both *process* (diet-to-die, within-die spatially correlated and independent) and *temperature variations* into the chip oxide breakdown reliability analysis. If the thickness of each device is modeled as a distinct random variable, then the full chip reliability estimation problem is defined on a huge sample space of several million devices. By noting that the reliability for a sample device with a given oxide thickness itself is a random function, the design time full chip reliability estimation problem turns out to be a multi-dimensional nested stochastic process. Furthermore, temperature variations may result in different device-level reliability models in hot spots and inactive regions, which further complicates this problem. Apparently a straightforward Monte Carlo approach is extremely expensive in both execution time and memory, as we need to perform nested Monte Carlo analysis on the sample spaces for different chips and different devices across each chip as well as the sample space of oxide breakdown of each device. The challenge here is how to reduce the tremendous number of random variables and then achieve a low space/time complexity. To address this issue, we propose a statistical approach that has the following key modeling contributions:

- First, we present a more consistent and accurate model for statistical full-chip reliability analysis. Unlike any traditional reliability analysis that simply uses the worst corner, the proposed model incorporates both the oxide thickness variation and temperature variation to ensure a reasonable result. This theoretically rigorous OBD reliability model considers variations at different spatial scales and hence involves the integration over millions of variables, which may be difficult for a direct solve.

- Second, the proposed framework discusses how to project that tremendous parameter space at device-level to the granularity of block level by characterizing the *B̲lock level O̲xide thickness D̲istribution* (BOD). Figure 3.1 illustrates the global temperature unevenness (corresponding to different functional modules) and local temperature uniformity for an alpha processors [65, 66]. Based on such observation, a "block" is defined a region on chip with uniform temperature

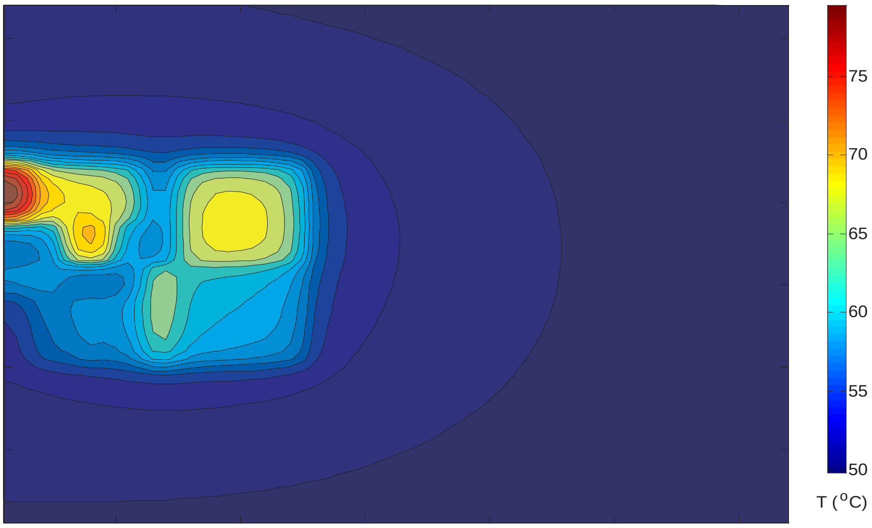


Figure 3.1: Temperature profile for an Alpha processor simulated by HotSpot [65]

spread [6]. We therefore present how to map the millions of random variables within each block to only two distinct random variables of sample mean and sample variance of BLOD. Such a projection greatly reduces the problem size to a feasible level while still capturing both the temperature variation and the process variations at different spatial scales.

- Third, we demonstrate how to characterize the sample mean/variance for each block using principal components and then compute the full-chip OBD reliability in an efficient way. By expressing the oxide thickness variation with principal components, we can achieve the closed-form representation of the sample mean/variance for each block as well as their correlation. Then, with some judicious approximations, the initial high dimensional integration for the reliability across the ensemble of chips can be simplified to the sum of double integrals, which enables fast and accurate estimation.

3.1 Oxide Thickness Variation Modeling

As is discussed in Chapter I, the oxide thickness variation can be classified based on the spatial scale over which it manifests using a chip-level variation model in (1.1) [7, 8, 47, 48]. In this work, we discuss the proposed approach using the grid based model [47, 48] by partitioning the chip into n grids and assuming perfect spatial correlation within each grid (see Figure 3.2). In other words,

[6] This "block" could be a real architecture level block or some sub-block that can ensure the assumption of uniform temperature.

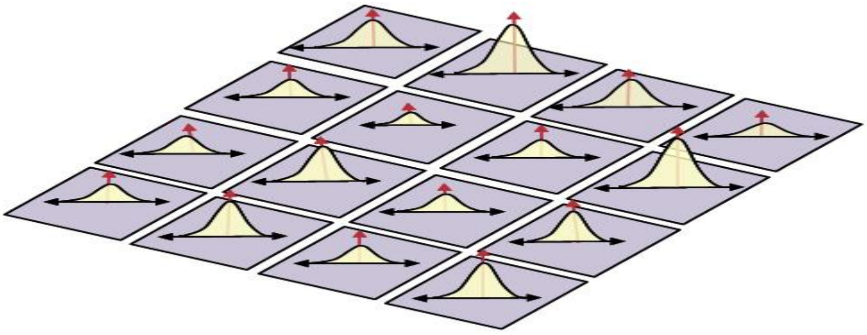


Figure 3.2: A grid based spatial correlation model

the devices within one grid have the correlation coefficient of 1 and hence bear the same spatially-correlated variation component, whereas devices in two different grids, i_{th} and j_{th} for example, have a covariance of $\rho_{i,j}\sigma^2_{corr}$, with a correlation coefficient $\rho_{i,j}$ <1 [47]. To further simplify the correlation structure, the set of n correlated random variables is mapped to another set of mutually independent random variables with zero mean and unit variance using the principal components of the original set. The original random variables are then expressed as a linear combination of the principal components. These principal components can be obtained by performing an eigenvalue decomposition of the correlation matrix. For a device in the i_{th} grid, its oxide thickness x_i can be canonically expressed as a linear combination of the principal components:

$$x_i = \lambda_{i,0} + \sum_{j=1}^{n} \lambda_{i,j} z_j + \lambda_r \epsilon, \tag{3.1}$$

where $\lambda_{i,0}$ is the mean or nominal value of oxide-thickness in i_{th} grid , z_j's represent the n independent random variables used to express the spatially correlated device parameter variations, ϵ is a distinct random variable for each device that represents the residual independent variation, and the coefficients $\lambda_{i,j}$'s represent the sensitivity of thickness variation in i_{th} principal component for every j_{th} the random variable.

3.2 Reliability Model and Problem Formulation

The gate oxide degradation depends on the oxide thickness, voltage, and temperature. There are many oxide breakdown models in the literature that attempt to explain the dependence on these factors. A widely accepted model is the anode hole injection model [67]. According to this model, injected electrons generate holes at the anode that can tunnel back into the oxide. Intrinsic breakdown occurs when a critical hole fluence is reached, creating a continuous conducting path across the oxide. A second model, known as electron trap density model, has been suggested, which claims that a critical density of electron traps generated during stress is required to trigger oxide breakdown [68]. Both models of oxide breakdown mechanisms note that the defect generation is a non-deterministic process. As a result the oxide breakdown time is inherently a statistically distributed quantity. Thus the oxide breakdown time is modeled as a random variable typically characterized by a Weibull probability distribution function, given by [55, 59]:

$$F(t) = 1 - e^{-a(\frac{t}{\alpha})^{\beta}} \tag{3.2}$$

where F is the cumulative distribution function (CDF) of time-to-breakdown t, a is the device area normalized with respect to the minimum device area, α and β are the scale and shape parameters of the Weibull distribution. The scale parameter α represents the characteristic life which is the time where 63.2% of samples fail, whereas the shape parameter β is a function of critical defect density. The critical defect density depends on device oxide thickness, the oxide field and temperature. For a given temperature and stress voltage, it has been shown that the slope parameter of the Weibull distribution varies linearly with oxide thickness [36]. Thus if x denotes the oxide thickness, we have

$$F(t) = 1 - e^{-a(\frac{t}{\alpha})^{bx}} \tag{3.3}$$

where b is a constant for given temperature and supply voltage. It has also been noted that the parameters α and b depend on temperature and can be characterized using some closed-form models or look up tables *w.r.t.* temperature for a given process [60, 61, 62, 69].

Another major factor that affects the oxide lifetime is the oxide breakdown failure criterion. A commonly used failure criterion is soft breakdown (SBD) which is characterized by a small increase in gate leakage. In practice, however, after SBD the gate leakage current monotonically increases with time eventually leading to a hard breakdown (HBD) [57]. Figure 3.3 plots the gate leakage measurement results for a stressed device in 45nm process. It can be observed from the figure how the gate leakage continuously increases after SBD until HBD is triggered. The time to breakdown is a function of the gate area, oxide quality, and the bias conditions [58]. For the purpose of this work, we limit our analysis to determining the initiation of soft breakdown and use this as our failure criteria since SBD is typically followed rapidly by HBD [58].

A chip is considered to have failed as soon as breakdown occurs for any device on the chip. We are interested in finding the reliable lifetime of the chip for which none of the devices fail. For such

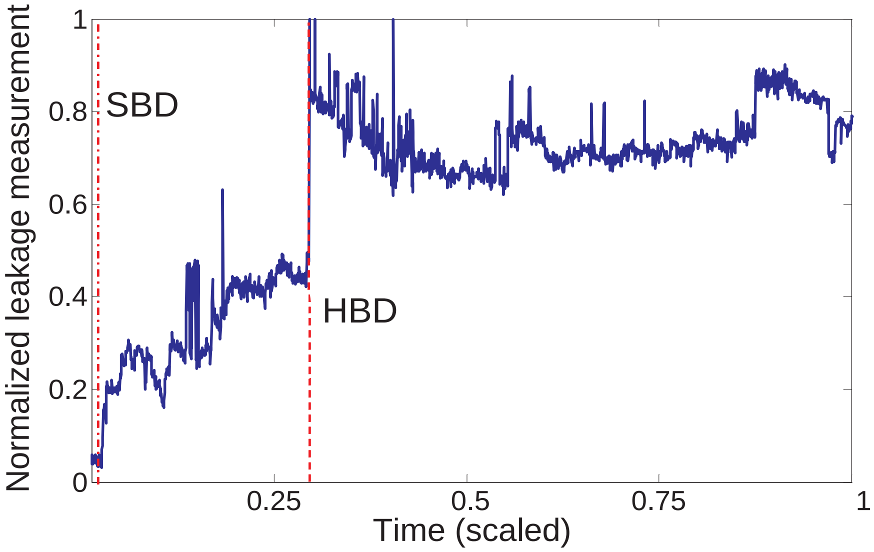


Figure 3.3: Typical oxide breakdown procedure characterized by gate leakage for a device in 45nm process (the stressed condition is 3.1V, 100 °C).

weakest link problems, it is more convenient to use an alternate representation known as reliability function $R(t)$ or survivor function, given by:

$$R(t) = 1 - F(t) = P(T > t) = \int_{t}^{\infty} f(s)ds \tag{3.4}$$

where $f(s)$ is the probability distribution function (PDF) of oxide breakdown of an individual device. The reliability function is complimentary to the cumulative distribution function (CDF) $F(t)$, taking the value 1 at $t = 0$ and tending to 0 as t tends to infinity. Simply stated, a reliability function is the probability that a device (chip) does not fail by time t. Due to manufacturing variations the thickness of gate oxide is also a non-deterministic parameter at design time. Thus the reliability function of a device can be interpreted as its conditional reliability function for a given oxide thickness. For an i^{th} device having x_i oxide thickness the conditional reliability function can be given as

$$R_i(t|x_i) = P(t > \mathbf{t}|x_i) = \int_{t}^{\infty} f(s|x_i)ds \tag{3.5}$$

Due to the spatial correlation of oxide thickness variation, the oxide thicknesses of any two devices on a chip are correlated with each other. Therefore, in general, their respective reliability functions being functions of oxide thickness are also correlated with each other. However, if the oxide thicknesses are known a priori then the defect generation mechanism in one device is independent of any other device on the chip for constant voltage and temperature. Thus for a particular chip, if the thicknesses of all devices are known then any device fails independently of all other devices. Furthermore the conditional reliability function of the chip $R_c(t|\mathbf{x})$ (with oxide thicknesses known) requires that all devices on the chip are functioning reliably. Thus, $R_c(t|\mathbf{x})$ is given by the product of reliability functions of all individual devices:

$$R_c(t|\mathbf{x}) = \prod_{i=1}^{m} R_i(t|x_i) \tag{3.6}$$

where $\mathbf{x}$ represent the vector of oxide thickness $(x_1, \ldots, x_m)$ and m is the total number of devices on the chip.

In the traditional analysis, where all oxide thicknesses are supposed to have a single, worst-case value, the product in (3.6) is taken across a large set of identical reliability functions and can be analytically solved with a low complexity. However, the key point in our analysis is that, at design time, each oxide thickness x_i is itself a random variable. In addition, these random variables are correlated across the chip. Furthermore, due to the temperature variation, many devices may have different reliability functions. If the oxide thicknesses of all devices are characterized by their joint probability density function (PDF) $f(x_1, \ldots, x_m)$ then the overall reliability function of the entire ensemble of all manufactured chips can be given by:

$$R_c(t) = \int_0^{\infty} \ldots \int_0^{\infty} \prod_{i=1}^{m} R_i(t|x_i) f(x_1, \ldots, x_m) dx_1 \ldots dx_m \tag{3.7}$$

Due to the huge dimensionality of the above integral, a straight forward numerical evaluation of the above integral is computationally impractical for full chip analysis. Using judicious approximations we develop a computationally efficient approach to address this problem in the next section.

3.3 Process Variation and Temperature-Aware Full Chip OBD Reliability Analysis

The proposed approach for efficiently estimating the overall reliability function $R_c(t)$ is discussed in a bottom up manner. We first present expressions for finding the conditional reliability function of a device. Using this expression, the conditional reliability function of a particular chip can be found given the oxide thickness of all devices on it as well as the temperature profile. We observe that although the overall reliability functions depends on the spatial and global correlation in oxide thickness variation, however, it is independent of the relative location of two devices on the chip. Hence, for a given chip, we can first partition the chip into the granularity of N functional blocks, in which devices share similar temperatures. Then, within each block, we can sum together all oxides of equal thickness and generate a frequency distribution histogram of the oxide thickness. As the oxide thickness variation of all the individual devices is modeled as a normal random variable and there are a large number of devices within a block, we show that such frequency distribution across a given block can be approximated by a normal distribution function. Henceforth, we will refer to this distribution function as the block level oxide thickness distribution (BOD). The BOD allows us to compactly represent the oxide thickness of all device within a block of a given chip using just two parameters - the mean and the variance of the underlying normal distribution function.

In section 3.3.1, we will present how a closed form expression for the block-level reliability function can be found for one BOD. Then the chip-level reliability for a given chip can also be analytically computed from N BODs. Finally, we discuss how to compute the overall reliability function across the entire ensemble of all manufactured chips. As N BODs may vary from die to die, their means and variances are in fact random variables over the sample space of all manufactured chips. Hence, the means and variances of N BODs can be represented by two random vectors, with N entries corresponding to N BODs in each vector. In other words, several million multi-variate oxide thickness distribution function for each device on the chip can be compactly modeled with just two random vectors. In section 3.3.3, we will discuss how these two random vectors can be derived from the oxide thickness process variation model given in (3.1) and thus the overall reliability function can be computed from it.

For clarity, we define the following notations in Table 3.1 that will be used throughout the remainder of the chapter.

3.3.1 OBD Reliability Analysis for One Chip

Using the definition of the reliability function and the oxide breakdown time model of an individual device in (3.3), the conditional reliability function of an i^{th} device on a chip having oxide-thickness x_i is given by

$$R_i(t|x_i) = e^{-a_i(\frac{t}{\alpha_i})^{b_i x_i}} \tag{3.8}$$

where α_i and b_i are temp device-level reliability parameters for the i_{th} device.

As explained in section 3.2, if the oxide thickens of all devices on a chip is known then the

Table 3.1: Notations used in OBD reliability analysis

Notation	Definition
N	number of functional blocks in a chip
m	number of devices of a chip
m_j	number of devices in the j_{th} block, *i.e.* $\sum_{j=1}^{N} m_j{=}m$
n	number of grids in the spatial correlation model of (3.1)
$\mathbf{x} = [x_1, \ldots, x_m]$	the oxide thicknesses for m device of a chip
$x_{i,j}$	oxide thickness for the i_{th} device in the j_{th} block of a chip, $i = 1 \ldots m$
$a_{i,j}$	area for the i_{th} device in the j_{th} block of a chip, $i = 1 \ldots m$
A_j	total area for the j_{th} block of a chip, $j = 1 \ldots N$
$\overline{x}_j{=}\sum_{i=1}^{m_j} x_{i,j}/m_j$	the sample mean for m_j devices of the j_{th} block
$v_j{=}\frac{\sum_{i=1}^{m_j}(x_{i,j}-\overline{x}_j)^2}{m_j-1}$	the sample variance for m_j device of the j_{th} block
$f_{\mathbf{x},\mathbf{y}}(x, y)$	joint PDF of x and y, where x and y can be either vector or scalar

reliability function of every device is independent of each other. Thus the reliability function of a chip is the product of the individual reliability function of all devices. Considering each device on the chip $x = (x_1, x_2, \ldots, x_m)$ and their respective area a_i, the conditional reliability of the chip is given by:

$$R_c(t|\mathbf{x}) = \prod_{i=1}^{m} R_i(t|x_i) = e^{-\sum_{i=1}^{m} a_i (\frac{t}{\alpha_i})^{b_i x_i}} \tag{3.9}$$

There may be several million devices on a chip and parameters of α_i and b_i may differ. Thus, it is impractical to evaluate the above exponent. In order to efficiently evaluate the overall reliability across all chips, we need to reduce the dimensionality of the above exponent while considering the impact of temperature variation across the chip.

It has been noted in Figure 3.1 that the on-chip temperature profile has the characteristics of global difference and local uniformity. Since both parameters b and α are heavily dependent on temperature [60, 61, 62], it is therefore unfair to assume that hot spots and inactive areas have the same reliability model and are hence equally prone to the OBD failure. In practice, temperature profile of a chip varies continuously across the chip. Transistors within a particular architecture-level block may share similar temperature due to the similar activities and supply voltage [65, 66]. On the contrary, temperature variation from block to block is much higher as functional blocks usually perform completely different operations [66]. It is therefore sufficient to construct a temperature-aware reliability analysis model at the granularity of architecture-level blocks. In other words, the analysis reasonably depends on the fact that devices within a block may bear similar temperature and hence share approximately the same parameters α_i and b_i for the reliability functions. As a result, we considers the block-level worst-case operating temperature and supply voltage in the analysis to account for the block-level temperature difference and to ensure a correct operation throughout the entire life time for any application profile. Then, (3.9) can be expressed at the functional block level:

$$R_c(t|\mathbf{x}) = \prod_{i=1}^{m} R_i(t|x_i) = e^{-\sum_{j=1}^{N}\sum_{i=1}^{m_j} a_{i,j}(\frac{t}{\alpha_j})^{b_j x_{i,j}}} \tag{3.10}$$

where N is the number of architecture-level blocks, α_j and b_j denote the parameters of the reliability functions for devices in the j_{th} block.

Equation (3.10) considers the across-chip temperature variation but cannot simplify the model. To achieve that, we represent the set of devices within a block and their individual oxide thicknesses using BOD for a particular block in a chip. For example, for the j_{th} block, the shape of its BOD can be approximated by performing histogram of the oxide thicknesses of $x_{i,j}$ for all the devices within that block. This block-specific BOD shows how many devices correspond to a particular oxide thickness within that block. For the sake of understanding, we discretize this oxide-thickness distribution for the j_{th} block into k_j discrete intervals assuming a truncated distribution. It can be seen that when we make this transformation the area of the devices with identical thickness in a block can be summed together. Let $\overline{x}_{i,j}$ denote the midpoint of the i^{th} discrete interval for the j_{th} BOD and $\overline{a}_{i,j}$ be the total area of all devices having thickness $\overline{x}_{i,j}$ in that block. By applying this transformation, the above expression for $R_c(t|\mathbf{x})$ can be rewritten as

$$R_c(t|\mathbf{x}) = e^{-\sum_{j=1}^{N}\sum_{i=1}^{k_j} \overline{a}_{i,j}(\frac{t}{\alpha_j})^{b_j \overline{x}_{i,j}}} \tag{3.11}$$

By making such a transformation the dimensionality of $R_c(t|\mathbf{x})$ can be reduced from number of devices m to the sum of the number of discrete intervals k_j, *i.e.*, $\sum_{j=1}^{N} k_j$. If we normalize the exponent with total area of each block, the above expression gives

$$R_c(t|\mathbf{x}) = \exp\left[-\sum_{j=1}^{N} A_j \sum_{i=1}^{k_j} p_{i,j}(\frac{t}{\alpha_j})^{b_j \overline{x}_{i,j}}\right] \tag{3.12}$$

where $p_{i,j} = \overline{a}_{i,j}/A_j$ represents the probability of observing an oxide thickness $x_{i,j}$ on a particular block of a sample chip. Thus the thickness of all devices on a particular sample chip can be compactly characterized by N BODs.

As discussed in section 3.1, the thickness variation of a device includes global variation (inter-die), spatially correlated intra-die variation (modeled as multi-variate Gaussian random vector for devices across the chip) and random variation (residual component, modeled as an independent Gaussian random variable, *e.g.*, $N(0, \sigma_\epsilon^2)$) [47]. Thus, for a set of devices within one particular block, they may have different oxide thicknesses due to the variability. The BOD is interpreted as the frequency distribution histogram of observing certain oxide thicknesses in this block for a sample chip. It can be characterized by the following lemma:

Lemma 1: *Following the oxide thickness variation classification above, BOD can be approximated by the curve of a Gaussian distribution.*

Proof. The proof is straightforward by analyzing the impact of different variation components. First, within one block, all the devices share the same global variation component (z_g).

Second, devices within the same block are closely placed and hence highly or even perfectly spatially correlated. This implies they may have approximately the same spatially-correlated variation component ($z_{corr,j}$).

Third, by using the variation component classification of inter-die, spatially correlated intra-die and random variations in (1.1), the oxide thickness of a device in the j_{th} block is:

$$x_{i,j} = u_0 + z_g + z_{corr,j} + z_{\epsilon,i,j} \tag{3.13}$$

For any device in the block, u_0, z_g and $z_{corr,j}$ are approximately the same. The difference of oxide thicknesses is therefore mainly caused by the random variation component $z_{\epsilon,i,j}$. In other words, oxide thickness of any device within the block can be considered as a sample from a Gaussian process $N(u_0 + z_g + z_{corr,j}, \sigma_\epsilon^2)$. Due to the independence of the random variation component $z_{\epsilon,i,j}$, oxide thicknesses of devices within one block are simply samples independently drawn from one common random process $N(u_0 + z_g + z_{corr_j}, \sigma_\epsilon^2)$.

As long as the number of devices is sufficient, BOD can be well characterized by the histogram of oxide thickness samples from the Gaussian random process $N(u_0 + z_g + z_{corr,j}, \sigma_\epsilon^2)$, which hence follows the curve of a Gaussian distribution. □

This lemma helps shed insight into the shape of a BOD and hence how $p_{i,j}$ may change *w.r.t.* x. Figure 3.4 validates the lemma with the histograms of oxide thicknesses for two blocks of a sample chip with different number of devices Monte Carlo simulation. It is clear that either for a

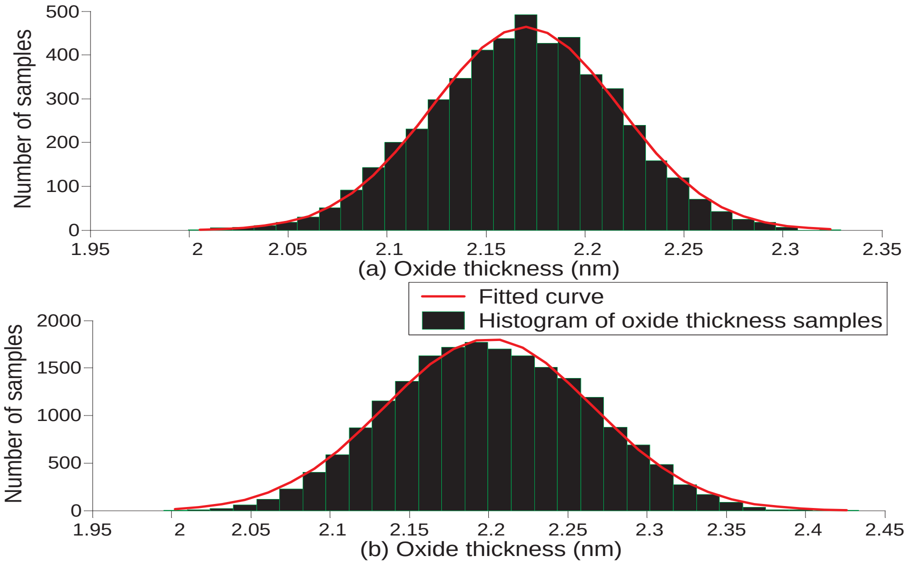


Figure 3.4: (a) Histogram of the oxide thickness for a block with 5K devices (b) Histogram of the oxide thickness for a block with 20K devices

block with 5K devices (Figure 3.4(a)) or a block with 20K devices (Figure 3.4(b)), we get distinctly Gaussian-like curves with fitting goodness (R-square) of 99.8% and 99.5%, respectively.

Thus, the summation part $\sum_{i=1}^{k_j} p_{i,j}(\frac{t}{\alpha_j})^{b_j \overline{x}_{i,j}}$ in (3.12) can be expressed by the integration over x:

$$\begin{aligned} R_c(t|\mathbf{x}) &= R_c(t|\mathbf{u},\mathbf{v}) \\ &\approx \prod_{j=1}^{N} \exp[-A_j \int\limits_{-\infty}^{\infty} \phi(\frac{x-u_j}{\sqrt{v_j}})(\frac{t}{\alpha_j})^{b_j x} dx] \end{aligned} \tag{3.14}$$

where $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ is the PDF for a standard Guassian distribution. $\mathbf{u} = (u_1, u_2, \ldots, u_N)$ and $\mathbf{v} = (v_1, v_2, \ldots, v_N)$, where u_j and v_j are the mean and variance of the j_{th} BOD.

Since (3.12) computes the conditional reliability $R_c(t|\mathbf{x})$ using only $2N$ distinct variables, the dimensionality of the problem in (3.9) is reduced from millions to $2N$. However, microprocessors usually have tens of blocks, making an integration with $2N$ variables still difficult to solve for (3.7). Note that $\exp[-A_j \int_{-\infty}^{\infty} \phi(\frac{x-u_j}{\sqrt{v_j}})(\frac{t}{\alpha_j})^{b_j x} dx]$ is approximately the product of all the device-level reliability functions in the j_{th} block, and hence very close to 1 within the lifetime of interest.

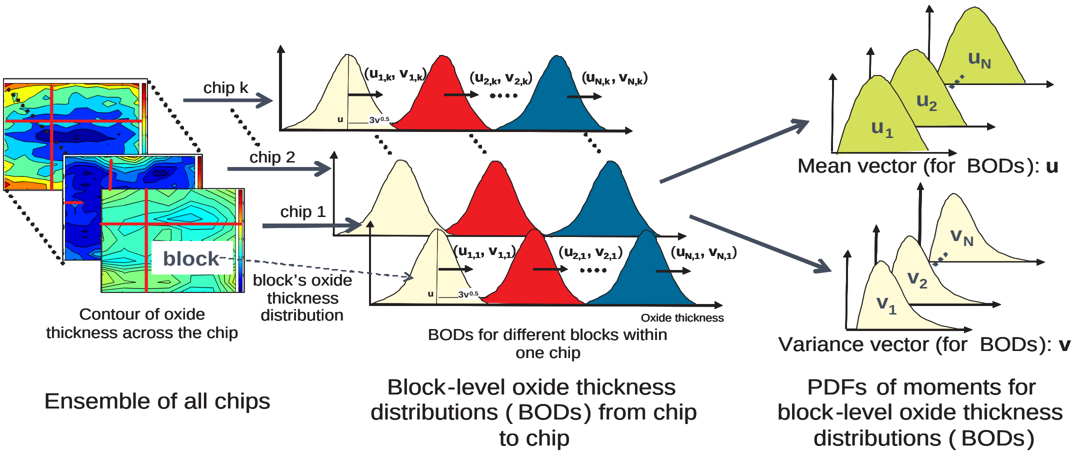

chip k
chip 2
chip 1
block
block's oxide thickness distribution
Contour of oxide thickness across the chip
Ensemble of all chips
$(u_{1,k}, v_{1,k})$
$(u_{2,k}, v_{2,k})$
$(u_{N,k}, v_{N,k})$
u
$3v^{0.5}$
$(u_{1,1}, v_{1,1})$
$(u_{2,1}, v_{2,1})$
$(u_{N,1}, v_{N,1})$
Oxide thickness
BODs for different blocks within one chip
Block-level oxide thickness distributions (BODs) from chip to chip
u_1
u_2
u_N
Mean vector (for BODs): **u**
v_1
v_2
v_N
Variance vector (for BODs): **v**
PDFs of moments for block-level oxide thickness distributions (BODs)

Figure 3.5: Compact representation of oxide thickness variation for the ensemble of chips

By applying first-order Taylor expansion, (3.12) can be further simplified to:

$$
\begin{aligned}
R_c(t|\mathbf{u},\mathbf{v}) &= \prod_{j=1}^{N} 1 - \left[1 - e^{-A_j \int_{-\infty}^{\infty} \phi(\frac{x-u_j}{\sqrt{v_j}})(\frac{t}{\alpha_j})^{b_j x} dx}\right] \\
&\approx 1 - \sum_{j=1}^{N} \left[1 - e^{-A_j \int_{-\infty}^{\infty} \phi(\frac{x-u_j}{\sqrt{v_j}})(\frac{t}{\alpha_j})^{b_j x} dx}\right]
\end{aligned} \tag{3.15}
$$

In the above equation, the integral in the exponent can be analytically evaluated by making the substitution $\frac{t}{\alpha_j} = e^{\gamma}$:

$$
\begin{aligned}
&\int_{-\infty}^{\infty} \phi(\frac{x-u_j}{\sqrt{v_j}})(\frac{t}{\alpha})^{b_j x} dx = \int_{-\infty}^{\infty} \phi(\frac{x-u_j}{\sqrt{v_j}}) e^{\gamma b_j x} dx \\
&= -\frac{1}{2} e^{\gamma b_j u_j + \gamma^2 b_j^2 v_j/2} erf(\frac{-x+u_j+\gamma b_j v_j}{\sqrt{2v_j}})|_{-\infty}^{\infty} \\
&= e^{\ln(\frac{t}{\alpha_j}) b_j u_j + (\ln(\frac{t}{\alpha_j}))^2 b_j^2 v_j/2}
\end{aligned} \tag{3.16}
$$

Equation (3.16) is denoted as $g(u_j, v_j)$ for simplicity throughout the remainder of the chapter.

Thus for N given BODs $\phi(\frac{x-u_j}{\sqrt{v_j}})$, where $j = 1...N$, the conditional reliability function of a chip can be computed by the following closed form expression:

$$
\begin{aligned}
&R_c(t|\mathbf{u},\mathbf{v}) \\
&= 1 - \sum_{j=1}^{N} \left[1 - e^{-A_j e^{\ln(\frac{t}{\alpha_j}) b_j u_j + (\ln(\frac{t}{\alpha_j}))^2 b_j^2 v_j/2}}\right]
\end{aligned} \tag{3.17}
$$

Hence, the multidimensional exponent in Equation (3.9) can now be compactly represented using a closed form analytical function of BOD parameters **u** and **v**.

3.3.2 Design-Time OBD Reliability Analysis for the Ensemble of Chips

At design time when chips are not fabricated, designers are unable to know the specific BOD distribution for any block of any chip. In other words, $\mathbf{u}$ and $\mathbf{v}$ turn out to be two random vectors at design time. The OBD reliability is then evaluated for the design (or the ensemble of chips) instead of a particular chip by integrating the conditional reliability function in (3.17) over the joint PDF of random vectors $\mathbf{u}$ and $\mathbf{v}$. In this section, we will discuss how to achieve a compact expression of the overall reliability function by enumerating the conditional reliability function derived in the previous section across the ensemble of all chips.

As shown in Figure 3.5, each sample chip from on design may result in different BODs for the same block from chip to chip, therefore, the oxide thickness variation of one block across the entire ensemble of all chips can be represented with a set of BODs for all manufactured chips. Now each such BOD is characterized by their respective mean u_j and variance v_j. Therefore, the oxide-thickness distribution of all devices across all manufactured chips with N blocks can be represented by $2N$ random variables $\mathbf{u} = [u_1, u_2 ... u_N]$ and $\mathbf{v} = [v_1, v_2 ... v_N]$. In other words, for one particular chip, its BODs simply results from the samples of two random vectors.

Now let $f_{\mathbf{uv}}(\mathbf{u}, \mathbf{v})$ denote the joint probability distribution function (PDF) of $\mathbf{u}$ and $\mathbf{v}$. For computing the overall reliability function, we need to integrate the above expression of reliability function of one chip over the joint probability distribution function $f_{\mathbf{uv}}(\mathbf{u}, \mathbf{v})$ of random vectors $\mathbf{u}$ and $\mathbf{v}$:

$$\begin{aligned} R_c(t) &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left[1 - \sum_{j=1}^{N} \left(1 - e^{-A_j g(u_j, v_j)}\right)\right] \times f_{\mathbf{u},\mathbf{v}}(\mathbf{u}, \mathbf{v}) du_1 ... du_N dv_1 ... dv_N \\ &= 1 - N + \sum_{j=1}^{N} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} e^{-A_j g(u_j, v_j)} \times f_{\mathbf{u},\mathbf{v}}(\mathbf{u}, \mathbf{v}) du_1 dv_1 ... du_N dv_N \end{aligned} \tag{3.18}$$

where $g(u_j, v_j)$ is defined in (3.16). Since $\exp[-A_j g(u_j, v_j)]$ is independent of any other u_i or v_i that $i \neq j$, we have:

$$\begin{aligned} &\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} e^{-A_j g(u_j, v_j)} f_{\mathbf{u},\mathbf{v}}(\mathbf{u}, \mathbf{v}) du_1 dv_1 ... du_N dv_N \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-A_j g(u_j, v_j)} \times \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\mathbf{u},\mathbf{v}}(\mathbf{u}, \mathbf{v}) du_1 dv_1 ... du_N dv_N \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-A_j g(u_j, v_j)} f_{\mathbf{u_j},\mathbf{v_j}}(u_j, v_j) du_j dv_j \end{aligned} \tag{3.19}$$

Thus, we can express the design time OBD reliability for the ensemble of chips in (3.19) using N double integrals:

$$R_c(t) = 1 - N + \sum_{j=1}^{N} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-A_j g(u_j, v_j)} f_{\mathbf{u_j},\mathbf{v_j}}(u_j, v_j) du_j dv_j \tag{3.20}$$

3.3.3 $f_{\mathbf{u_j},\mathbf{v_j}}(u_j, v_j)$ Characterization Using Principal Component Analysis

To compute (3.20), it is still required to know the characteristics of joint PDF $f_{\mathbf{u_j},\mathbf{v_j}}(u_j, v_j)$ of each BOD. In this section, we will discuss how to characterize those joint PDFs for blocks by using principal component analysis.

As is discussed in the previous section, each BOD approximately follows a Gaussian curve and hence can be represented by their respective mean u_j and variance v_j. For a particular block j, the mean u_j and and variance v_j of its oxide thickness distribution can be estimated by calculating the unbiased statistical BOD mean and variance of the oxide thickness values observed across the block. Likewise, the random variables u_j and v_j can be found in terms of the thickness variation model discussed in (3.1). Using the oxide thickness variation model given in (3.1), sample mean u_j can be expressed as:

$$u_j = \frac{1}{m_j} \sum_{i=1}^{m_j} x_{i,j} = u_{j,0} + \sum_{k=1}^{n} u_{j,k} z_k + u_{j,n+1} \epsilon \tag{3.21}$$

The grid-based model in (3.1) partitions the chip into several grids, as discussed in section 3.1. Assume the i_{th} device in the j_{th} block is located in a grid, *e.g.*, grid $g_{i,j}$, where $g_{i,j}$ corresponds to a grid index from 1 to n. Then, we can compute $u_{j,k}$ and $u_{j,n+1}$ as:

$$\begin{aligned} u_{j,k} &= \frac{1}{m_j} \sum_{i=1}^{m_j} \lambda_{g_{i,j},k} \quad \forall k = 0 \ldots n \\ u_{j,n+1} &= \frac{1}{m_j} \sqrt{\sum_{i=1}^{m_j} \lambda_r^2} = \frac{\lambda_r}{\sqrt{m_j}} \end{aligned}$$

The coefficient $u_{j,0}$ is the nominal value of u_j, whereas coefficient $u_{j,i}$ is the sensitivity to the i_{th} principal component. It is evident that the sensitivity of the independent random component $u_{j,n+1}$ tends to zero for a large number of devices and thus can be safely neglected for a typical industrial chip.

Similarly the expression for v_j, the sample variance of the j_{th} BOD, in terms of oxide variation model in (3.1), can be given as follows:

$$v_j = \frac{1}{m_j - 1}\sum_{i=1}^{m_j}(x_{i,j} - u_j)^2 = \frac{1}{m_j - 1}\sum_{i=1}^{m_j}(x_{i,j}^2 - {u_j}^2) \tag{3.22}$$

Again the above expression can be expressed in terms of principal components as follows:

$$v_j = v_{j,0} + \sum_{i=1}^{n}\sum_{k=1}^{n} v_{j,i,k} z_i z_k \tag{3.23}$$

where

$$v_{j,0} = \lambda_r^2 \quad \text{and} \quad v_{j,i,k} = \frac{1}{m_j - 1}\sum_{l=1}^{n}(\lambda_{l,i} - u_{j,i})(\lambda_{l,k} + u_{j,k})$$

In this manner, we can express the distributions of u_j and v_j in terms of a given process variation model. Note that random variable u_j is the sum of normal random variables so it is also a normal random variable, however, the BOD variance v_j is not a normal random variable. By exploring their characteristics, we have the following lemma for their un-correlation:

Lemma 2: *Following the oxide thickness variation model in* (3.1), u_j *and* v_j *for a BOD is uncorrelated, i.e.,* $E[u_j v_j] = E[u_j]E[v_j]$, *where* $E[\cdot]$ *denotes the expectation.*

Proof. following the principal component models discussed above, we can express $E[u_j v_j]$ as:

$$E[u_j v_j] = E[(u_{j,0} + \sum_{i=1}^{n} u_{j,i} z_i + u_{j,p+1}\epsilon) \times (v_{j,0} + \sum_{i=1}^{n}\sum_{k=1}^{n} v_{j,i,k} z_i z_k)]$$

By noting that each principal component z_i as an independent standard normal random variable, we have:

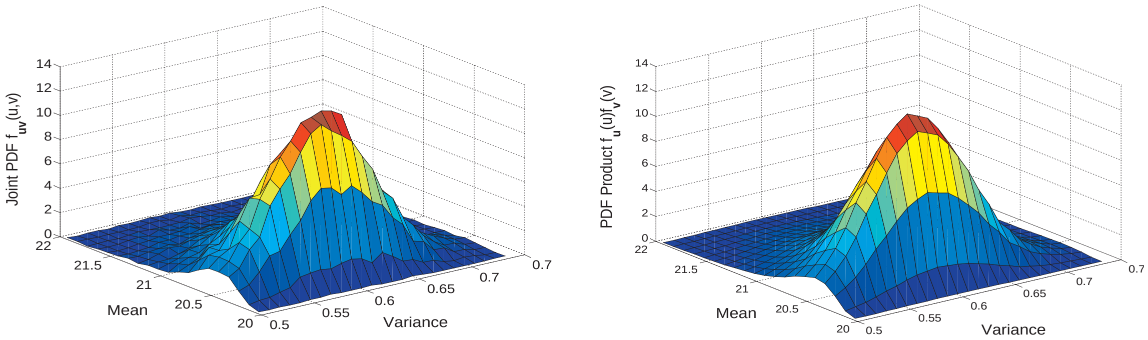
$$\begin{aligned} E[z_i] = E[z_i^2 z_j] = E[z_i z_j^2] = E[z_i^3] = 0 \\ E[z_i^2] = 1 \end{aligned} \tag{3.24}$$

for different i and j. Likewise,

$$\begin{aligned} E[\epsilon] = E[\epsilon^2 z_j] = E[z_i \epsilon^2] = E[\epsilon^3] = 0 \\ E[\epsilon^2] = 1 \end{aligned} \tag{3.25}$$

Thus, the above expression can be simplified and given by

$$E[u_j v_j] = u_{j,0} v_{j,0} + \sum_{i=1}^{n} u_{j,0} v_{j,i,i} = E[u_j]E[v_j]. \tag{3.26}$$



(a) (b)

Figure 3.6: (a) Joint PDF $f_{\mathbf{u_j v_j}}(u_j, v_j)$ (b) PDF product $f_{\mathbf{u_j}}(u_j)f_{\mathbf{v_j}}(v_j)$

□

For two normal random variables to be independent, it is sufficient to show that they are uncorrelated, but in general this is not the case for non-gaussian random variables. The sample variance v_j is not a normal random variable and has the distribution of quadratic forms in normal variables [70, 71]. However, with numerical experiments we find that the dependence between u_j and v_j is weak. As a result, it is reasonable to assume u_j and v_j as independent variables, which allows us to express the joint PDF in terms of their marginal distributions $f_{\mathbf{u_j}}(u_j)$ and $f_{\mathbf{v_j}}(v_j)$. Figure 3.6 illustrates the joint PDF of $f_{\mathbf{u_j v_j}}(u_j, v_j)$ and the product of the marginal distributions, $f_{\mathbf{u_j}}(u_j)f_{\mathbf{v_j}}(v_j)$, generated by Monte Carlo simulations. It is qualitatively evident from the figure that there does not exist significant dependence between $\mathbf{u_j}$ and $\mathbf{v_j}$. Furthermore, Figure 3.7 depicts the contour of the error between joint PDF $f_{\mathbf{u_j}}(u_j)f_{\mathbf{v_j}}(v_j)$ and PDF product $f_{\mathbf{u_j}}(u_j)f_{\mathbf{v_j}}(v_j)$ normalized with respect to the peak probability on the joint PDF. It is noted that the maximal error is around 7% in a very small region whereas most errors are almost negligible.

Thus, this independence approximation between u_j and v_j can give us a reasonably accurate estimate of oxide variation with a significantly simpler approach. In other words, the approximation enables us to enumerate the individual reliability distribution functions of each chip by simply integrating the marginal distributions $f_{\mathbf{u_j}}(u_j)$ and $f_{\mathbf{v_j}}(v_j)$:

$$R_c(t) = 1 - N + \sum_{j=1}^{N} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-A_j g(u_j, v_j)} f_{\mathbf{u_j}}(u_j) f_{\mathbf{v_j}}(v_j) du_j dv_j \tag{3.27}$$

Now the BOD sample mean u_j is a sum of normal random variables, therefore $f_{\mathbf{u_j}}(u_j)$ can be characterized by distribution of a normal random variable and analytically computed. However,

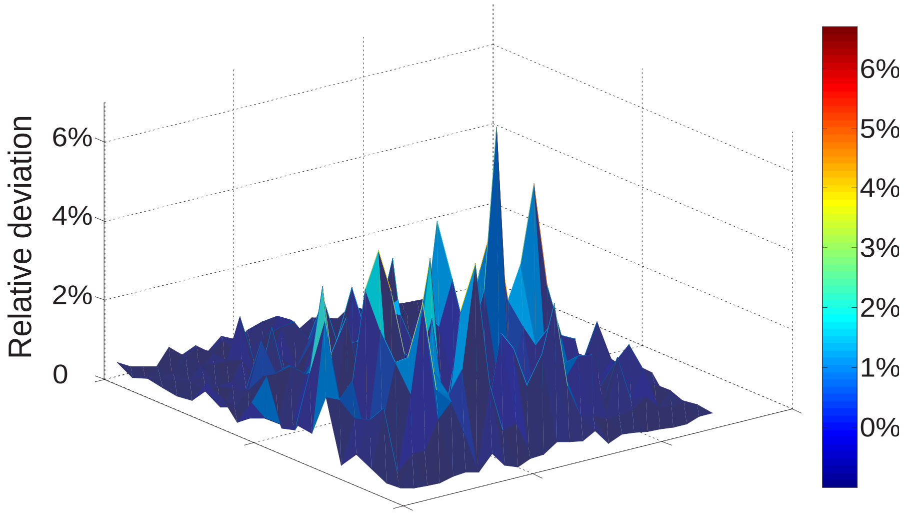


Figure 3.7: Contour of the error between joint PDF $f_{\mathbf{u_j v_j}}(u_j, v_j)$ and PDF product $f_{\mathbf{u_j}}(u_j)f_{\mathbf{v_j}}(v_j)$

BOD sample variance v_j is a quadratic expression of normal random variables. Such an expression is commonly found in several multi-variate statistics application and is referred to as quadratic normal form. In statistics literature [70], several techniques have been proposed to accurately estimate the distribution function of quadratic normal form. In this work, we implemented a computationally efficient method given in [71] to estimate the distribution of $f_{\mathbf{v_j}}(v_j)$ using a chi-square approximation:

$$v_j \sim v_{j,0} + \hat{a}\chi^2_{\hat{b}} \tag{3.28}$$

where

$$\begin{aligned} \hat{a} &= \sum_{i=1}^{n}\sum_{k=1}^{n} v_{j,i,k}^2 / \sum_{i=1}^{n} v_{j,i,i} \\ \hat{b} &= (\sum_{i=1}^{n} v_{j,i,i})^2 / \sum_{i=1}^{n}\sum_{k=1}^{n} v_{j,i,k}^2 \end{aligned} \tag{3.29}$$

In figure 3.8, we compare the CDF of the distribution of quadratic normal form of a v_j by Monte Carlo simulation and its chi-square approximation. It is apparent that the computationally efficient chi-square representation is in good agreement with the Monte Carlo simulation result.

In this manner, the marginal distributions $f_{\mathbf{u_j}}(u_j)$ and $f_{\mathbf{v_j}}(v_j)$ of u_j and v_j can be analytically found for the given process variation model of oxide thickness. Using $f_{\mathbf{u_j}}(u_j)$ and $f_{\mathbf{v_j}}(v_j)$, the overall reliability distribution function can be computed by evaluating N two-dimensional numerical integrations as in (3.27).

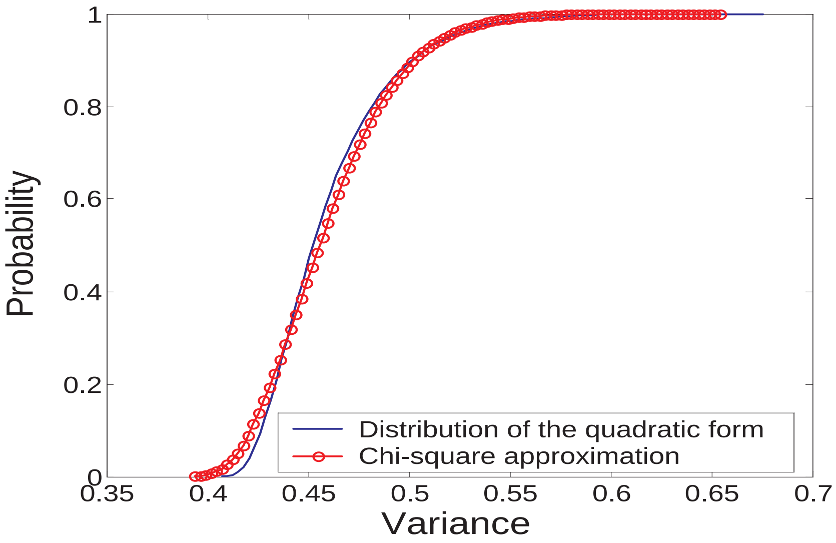


Figure 3.8: Curves for the distribution of the quadratic form and its χ^2 approximation

3.3.4 Overall Algorithm

The overall algorithm of the proposed approach is summarized in figure 3.9. Given the principal components as well as the oxide thickness variation profiles, we can characterize u_j and v_j for each BOD using (3.21) and (3.23). Then we divide the integration domain for (3.27) to $l_0 \times l_0$ sub-domains. Since the joint PDF rapidly decreases to 0 beyond a narrow domain, as illustrated in figure 4, l_0=10 is already a reasonable number for accurate integral sum evaluation, which is further confirmed by the experimental results in section 3.4. Once sample point pair in each sub-domain is obtained, we can compute analytically the reliability for one chip. Finally the overall reliability is evaluated by using the integral sum.

It is noted that PCA is a pre-processing step. Thus, we do not include it in the complexity analysis as it is performed only once and can be shared with other statistical analysis tools. The overall complexity is $O(N(n^2 + l_0^2))$, where N is the number of blocks, n^2 is the number of principal components and l_0^2 is the number of sub-domains for integration. By noting that unlike the straightforward approach, the computation complexity is independent of the total number of devices on the chip, it is therefore extremely computationally efficient compared with Monte Carlo analysis, whose complexity heavily depends on the number of devices. Moreover, since temperature and supply voltage are used as the input in our model, the correlation of the temperature/voltage profiles between the blocks are therefore naturally captured in the analysis.

```
Procedure: Full Chip OBD Reliability Analysis
Input: Number of devices, block-level configuration and related profiles, principal components for (3.1), inter- and intra-die variation profiles, spatial correlation profile.
Output: OBD reliability.
Characterize $u_j$ and $v_j$ for each BOD of a given design using (3.21) and (3.23);
Divide the integration domain for a double integration to $l_0 \times l_0$ sub-domains;
Compute the sample point pairs $(u_j, v_j)$ for each sub-domain;
For each sample point pair do
    Analytically compute $e^{A_j g(u_j,v_j)}$ using (3.16);
    Evaluate the PDF product $f_{\mathbf{u_j}}(u_j) f_{\mathbf{v_j}}(v_j)$ for the sample point;
End for
Compute the overall reliability with the integral sum;
```

Figure 3.9: Algorithm for process variation and temperature-aware full chip OBD reliability analysis

3.3.5 Fast Computation Using a Hybrid Analytical/Table-Look-Up Method

At design time, it is common for designers to repeatedly evaluate the reliability of the same design with different setup and application profiles. Different setup/application profiles may lead to different device-level reliability parameters α and b and hence require computing the integrations again. Although the formulation in (3.27) significantly reduces the computation complexity, we may achieve further speed up by combining this analytical model with a table look-up method. The pre-calculated look-up table only needs to be computed once for a particular design and can be used for various setup/application profiles, enabling a significant efficiency improvement. In other words, we now can efficiently evaluate the reliability of the same design through look-up table with an $O(1)$ time complexity.

Equation (3.27) is comprised of N double integrals. Take the j_{th} integral for example:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-A_j g(u_j,v_j)} f_{\mathbf{u_j}}(u_j) f_{\mathbf{v_j}}(v_j) du_j dv_j \tag{3.30}$$

where $g(u_j, v_j) = e^{\ln(\frac{t}{\alpha_j}) b_j u_j + (\ln(\frac{t}{\alpha_j}))^2 b_j^2 v_j/2}$ Since u_j and v_j are integration variables, they will be eliminated after the integration is computed. Thus, the result of (3.30) is determined by A_j and the parameters of $\ln(t/\alpha_j)$ and b_j in $g(u_j, v_j)$, as shown in (3.16). Once the chip is designed, A_j appears to be a constant for the j_{th} functional block. Thus, with $\ln(t/\alpha)$ and b acting as indices, we can construct a two-dimension look-up table to compute the double integral for each block [7].

Then, the system reliability at any time t under certain temperature/voltage conditions (which

[7] All the look-up tables for different functional blocks share the same indices of $\ln(t/\alpha_j)$ and b_j. The difference in look-up entries among the blocks is due to the different block area A_j.

determine the parameters of α and b), can be easily computed by using simple bilinear interpolation according to the indices of $\ln(t/\alpha)$ and b. For N functional blocks, we have N look-up tables, with $n_\alpha \times n_b$ entries in each table, where n_α(=100) and n_b(=100) are the number of indices for parameters $\ln(t/\alpha)$ and b, respectively. Experiments in section 3.4 show that the hybrid method leads to a faster speed but nearly equivalent accuracy to the analytical approach in section 3.3.4.

3.4 Experimental Results

A simple simulation methodology for estimating the critical defect density required for triggering a dielectric breakdown in an ultra thin oxide was originally developed in [36]. Using this methodology, the defect generation relationships for the technology node and the technology dependent parameters of the oxide reliability function model are then obtained from [55, 60, 61, 62], which are used in the device-level reliability model (3.3). In practice, such a model can also be characterized from real oxide breakdown distributions measured from test capacitors or discrete devices for the required process and technology.

The proposed approach was implemented in MATLAB and tested on six benchmarks (C1-C6) varying from 50K to 0.84M devices. Design C6 is a alpha processor design with 15 functional blocks and approximately 0.84M transistors. We then use HotSpot [65] to achieve the temperature profile of the design with Wattch to estimate the functional block power [72]. In the simulation we consider the inter-chip, spatially correlated intra-chip, and random components of variation. According to [6], the $3\sigma/u$ ratio for oxide thickness variation is assumed to be 4% for a nominal value of 2.2nm , and then split to 50% global variation, 25% spatially correlated variation and 25% independent variation, as estimated in [73]. As the real measurement data for thickness correlation was unavailable, the covariance matrix for thickness variations used in this work was derived from an exponential decaying function of the respective distance. The correlation distance of exponential correlation function is normalized with respect to the chip dimensions.

Given the post-layout design implementation and a process variation model of oxide thickness, the proposed methodology can compute the overall reliability distribution function. To validate the results of the proposed method, the overall reliability distribution was also computed from 1000 samples of Monte Carlo (*abbrev.* MC) simulations using the same oxide reliability model and thickness variation model. In Table 3.2, a comparison of lifetime estimation for 1-fault-per-million parts and 10-faults-per-million parts between the proposed statistical approach (statistical) as summarized in section 3.3.4, the fast hybrid analytical/table-look-up approach (hybrid) and Monte Carlo simula-

Table 3.2: Accuracy and run time comparison of the temperature-aware statistical approach in section 3.3.4 and hybrid analytical/table-look-up approach in section 3.3.5 with Monte Carlo simulation

Design	#devices	Lifetime estimation error (%) w.r.t. MC				Run time (sec.)/Speed up w.r.t. MC				
		1/million		10/million		statistical		hybrid		MC
		statistical	hybrid	statistical	hybrid					
C1	50K	0.84	0.12	1.18	1.84	1.51	177×	0.020	13498×	267
C2	80K	1.50	0.68	1.28	0.30	1.59	238×	0.022	17486×	380
C3	0.1M	2.04	0.16	1.77	2.26	1.92	245×	0.019	24122×	470
C4	0.2M	2.23	0.63	1.90	1.30	1.93	363×	0.020	35206×	702
C5	0.5M	0.20	3.42	0.12	1.65	1.86	837×	0.020	77845×	1557
C6	0.84M	0.64	1.63	0.54	0.76	1.95	1183×	0.020	115325×	2307
Average		1.24	1.11	1.13	1.35		418×		47247×	

Table 3.3: Accuracy comparison between the proposed approach in section 3.3.4 and MC simulation for different correlation distance

circuit		Lifetime estimation error w.r.t. MC (%)					
		$\rho_{dist} = 0.25$		$\rho_{dist} = 0.5$		$\rho_{dist} = 0.75$	
name	No. devices	1/million	10/million	1/million	10/million	1/million	10/million
C1	50K	2.31	2.95	0.84	1.18	1.00	1.02
C2	80K	2.26	1.98	1.50	1.28	1.28	1.37
C3	0.1M	3.35	2.72	2.04	1.77	2.17	1.50
C4	0.2M	3.77	3.51	2.23	1.90	1.96	1.73
C5	0.5M	1.62	2.06	0.20	0.12	0.76	0.92
C6	0.84M	1.70	2.18	0.64	0.54	0.86	0.80

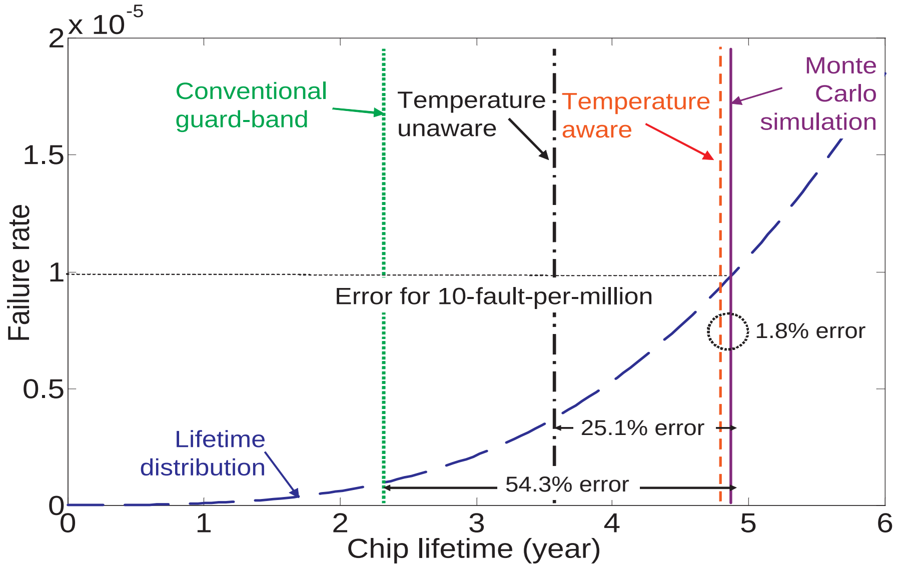


Figure 3.10: Errors of the 10-faults-per-million for Monte Carlo simulation, the proposed temperature-aware approach, temperature-unaware approach using worst-case temperature and conventional guard-band assuming minimum oxide thickness

Table 3.4: Accuracy comparison between the proposed approach in section 3.3.4 and MC Simulations for different grid resolution for design C2

Grid size	Lifetime estimation error (%) w.r.t. MC simulation					
	$\rho_{dist} = 0.25$		$\rho_{dist} = 0.5$		$\rho_{dist} = 0.75$	
	1/million	10/million	1/million	10/million	1/million	10/million
10×10	3.20	3.17	2.96	3.03	2.87	3.24
20×20	2.91	3.08	2.05	1.97	3.01	2.92
25×25	2.26	1.98	1.50	1.28	1.28	1.37

tions is shown for 6 design circuits. The criterion of n-fault-per-million parts is a commonly used term in reliability analysis [74], which is defined as the time when the first n out of a million parts fail. In the spatial correlation model, the relative correlation distance (ρ_{dist}) *w.r.t.* the chip size is set to 0.5. The size of the circuit under test in terms of number of devices is given in the second column. As can be seen from columns 3-6, the two proposed methods are in good agreement with the Monte Carlo simulation, with errors of around 1% on average. Columns 7-11 compare the run time for three methods. Unlike MC simulation, both our statistical approach and hybrid approach are able to analyze the circuit in seconds, independent of the number of devices. The statistical approach in section 3.3.4 demonstrates around 2-3 orders of magnitude speed-up for all the designs, whereas MC simulation scales super-linearly with the number of devices. The hybrid approach in section 3.3.5 is even faster. It has around 3 orders and 5 orders of magnitude speed-up compared with the statistical approach and MC simulation, respectively. Meanwhile it can maintain a similar accuracy as the statistical approach in section 3.3.4. This is an appealing feature for a real system with increasingly larger designs that may require repeated reliability calculation.

To verify the robustness of the proposed approach with respect to spatial correlation model we tested our approach for three different values of correlation distance (ρ_{dist}=0.25, 0.5, 0.75). As can be seen from Table 3.3, the proposed method can still maintain a good accuracy. We also validates the approach by choosing 4 different resolutions of grid size for design C2. The numerical results found for 3 different grid size are given in Table 3.4. As the discretization error of the grid-based model decreases for larger grid size, it can be seen that the error in estimation of reliability function also decreases in general.

We further compare the overall reliability estimation results in Figure 3.10 using Monte Carlo simulation, the proposed temperature-aware statistical approach in section 3.3.4, temperature-unaware approach by using the worst-case temperature across the chip and conventional guard-band approach that assumes minimum oxide thickness across the chip. Figure 3.10 shows the failure rate of design C3 during the selected lifetime period and reliability estimation by different methods.

The chip lifetime distribution (blue curve) is achieved by simulating the failure time of 10000 sample chips of C3 in a Monte Carlo fashion. One can see that for 10-faults-per-million criterion, the temperature-unaware approach and conventional guard-band lead to 25.1% and 54.3% errors, whereas our temperature-aware approach can achieve an accuracy of 1.8% error and is very close to the result by Monte Carlo simulation. This clearly exemplifies the necessity for a process variation and temperature-aware approach for OBD reliability analysis.

3.5 Summary

This chapter proposes a statistical methodology for process and temperature variation-aware chip-level oxide breakdown reliability analysis. It is shown that worst-case oxide reliability analysis or temperature-unaware approach may not be adequate to predict chip lifetime accurately. The complexity analysis of the proposed methodology shows that the proposed approach is independent of the number of devices and is thus scalable to large industrial size circuits. Our simulation results exemplifies the accuracy and efficiency of the proposed method.

CHAPTER IV

Post-Fabrication Measurement-Driven Oxide Breakdown Reliability Prediction and Management

Due to aggressive technology scaling, designing a reliable system has become more challenging than ever [4]. The worsening process variation increases susceptibility of the system to various wear-out mechanisms [17]. Among these reliability issues, oxide breakdown (OBD) has emerged as one of the most pressing concerns. As gate oxide thickness is scaled down to the one nanometer regime, the stronger electric field across the gate insulator results in faster formation of a conduction path through the dielectric layer, aggravating the risk of destructive breakdown [58].

Conventional worst-case guard-band methodology analyzes chip OBD reliability by assuming a minimum oxide thickness across the chip and then sets a supply voltage level to ensure the required lifetime of the chip. Clearly, such strategy is overly pessimistic and enforces an overly low supply voltage for the ensemble of chips, causing significant penalty in performance budget [17, 58]. In practice, no two transistors are exactly the same or have precisely the same characteristics. Instead, they vary significantly from wafer to wafer, reticle to reticle, die to die and across the die. Hence, some dies with thinner than average oxides are much more likely to fail than other dies. To more accurately account for the impact of thickness variation on lifetime prediction, Chapter III developed a statistical lifetime analysis by incorporating both process and temperature variations.

However, without post-fabrication measurement, designers cannot know the oxide thickness of an individual transistor on a particular die and hence cannot determine the specific lifetime expectation from one chip to another. Either the method in Chapter III or Monte-Carlo simulation only relies on the general variation knowledge of the technology node and results in a more accurate but ultimately still highly-spread lifetime distribution for any chip. This is due in part to the lack of information of the unique condition of a particular chip and unfairly implies a chip that happens

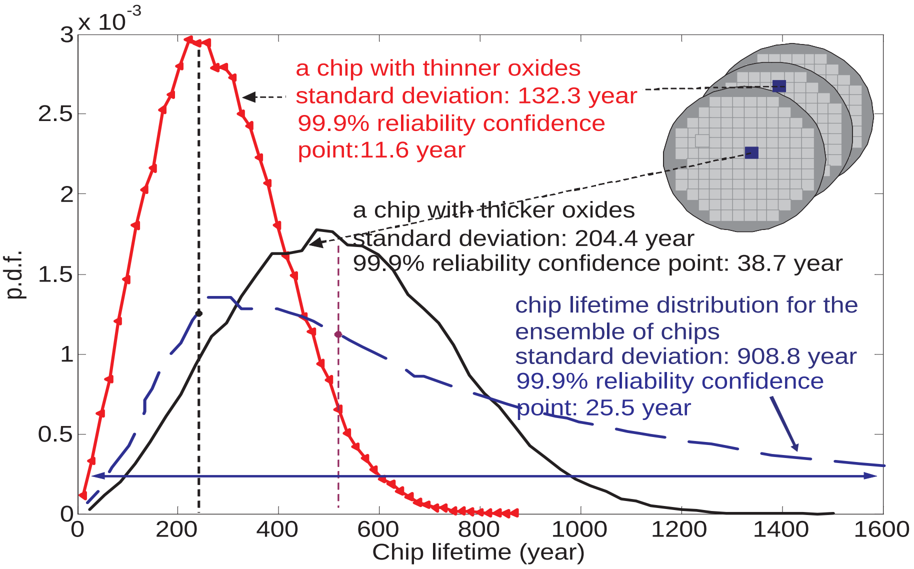


Figure 4.1: Chip lifetime distribution for the ensemble of chips (blue curve) (nominal thickness is 1.67 nm for 65nm device). Red and black curves represent lifetime distributions for two particular chips with all devices' oxide thicknesses known.

to have thicker oxides, bears the same risk to failure as the one with thinner oxides. Figure 4.1 presents the chip lifetime distribution (blue curve) by simulating the failure time of 50000 chips in a Monte-Carlo fashion. The spread in lifetime results is partly from the innate randomness of the OBD mechanism. The lifetime spread is further increased by thickness variation ($3\sigma/\mu$=4% [6]) which has an exponential effect on the tunnelling current and injected charge and eventually leads to the lognormal shape in Figure 1 with a long tail (908.8 year standard deviation / 99.9% reliability confidence point is 25.5 year) [75]. However, each chip has unique oxide thickness conditions for each transistor and hence some chips are bound to have significant lifetime margin which could be traded off for higher performance by allowing these chips to operate at a higher supply voltage.

Thus, if the oxide thickness of each individual transistor on a fabricated chip could be measured, the lifetime distribution for that chip would be significantly tightened, as shown in Figure 4.1 for two chips, one with thinner oxides (red curve, 132.3 year standard deviation / 99.9% reliability confidence point is 11.6 year) and one with thicker oxides (black curve, 204.4 year standard deviation / 99.9% reliability confidence point is 38.7 year). Then the chip with thinner oxide thickness (red curve) has a significantly higher risk to fail early and should be operated using a lower maximum supply voltage, thereby improving the overall reliability of the design. Conversely, the chip with black curve whose oxide thickness happens to be thicker is less prone to failure and could be operated at a higher supply voltage limit and therefore obtain a performance gain while still meeting reliability target. Hence, understanding the oxide thickness condition on a die can result in both a performance improvement as well as a higher reliability.

Unfortunately, obtaining oxide thickness condition for all devices on a die is impossible in today's chips with hundreds of millions to billions of transistors. However, recent advances in compact oxide thickness sensors [76, 77] allow tens to hundreds of sensors to be placed on a chip or even inside cores. Thus, a key challenge, which is the focus of this paper, is how to precisely predict and manage the reliability condition of each chip with a limited number (<1000) of post-fabrication oxide thickness measurements using on-chip sensors. This problem is non-trivial:

- First, while the number of measurements is limited, the number of transistors on a die in today's technology can be enormous, exceeding 1 billion. It is therefore crucial to fully utilize of the measurement information to predict the oxide thickness for all devices as accurately as possible.

- Second, while we can measure the oxide thickness of sensor device with reasonable accuracy, the thicknesses of all other transistors remain uncertain and must be modeled as random variables. Even with a fixed oxide thickness, the reliability for a device itself is a random function representing the probability the device can survive to a certain lifetime [58]. The measurement-driven chip reliability estimation therefore turns out to have the form of a conditional multi-dimensional nested stochastic process. Simple Monte-Carlo simulation must model both the random variation in oxide thickness as well as the innate variation of OBD itself and is therefore extremely expensive in both time and memory.

- Finally, OBD is also a strong function of the chip operating conditions, such as processor state and temperature which vary during the operation of a device. For simplicity, our discussion has not accounted for these factors up to now. However, they have a significant impact on the lifetime of a particular die and we will outline how to incorporate these effects in our proposed analysis.

In this chapter, we propose a new *post-fabrication* measurement-driven OBD reliability prediction and management methodology using a limited number of measurement points. The measurements of oxide thicknesses for a subset of devices can be conducted by on-chip sensors [76] or test-structures [77], which can be easily modified to measure the oxide thickness instead of monitoring the degradation process. Figure 4.2 displays the proposed post-fabrication flow including the OBD reliability prediction module using the introduced OBD analysis. For each fabricated chip, the measurement

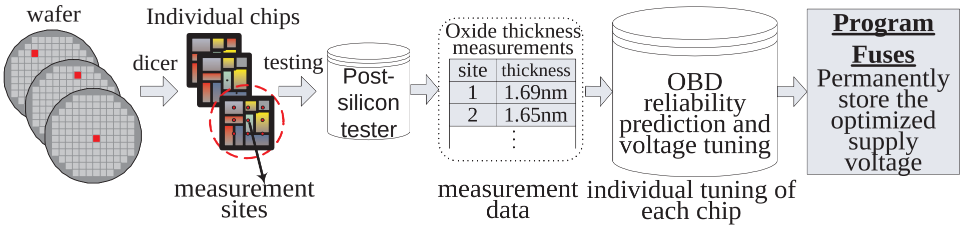

wafer
Individual chips
dicer
testing
Post-silicon tester
Oxide thickness measurements
site
thickness
1
1.69nm
2
1.65nm
⋮
OBD reliability prediction and voltage tuning
Program Fuses
Permanently store the optimized supply voltage
measurement sites
measurement data
individual tuning of each chip

Figure 4.2: Proposed post-fabrication oxide thickness measurement-driven supply voltage optimization flow.

is performed once during post-silicon testing to find the initial oxide thickness at the start of its lifetime. Then the optimal supply voltage limit is selected by the prediction module to maximize performance while maintaining or improving chip OBD reliability. Given the computed supply voltage limit, the tester permanently stores the optimized supply voltage for each chip using either fuses or embedded flash memory. This supply voltage limit is then accessed by the dynamic voltage scaling algorithms and, if available, dynamic reliability management algorithms [64] that control the chip operation during runtime.

The OBD reliability prediction and voltage tuning module in this flow consists itself of three phases. The first phase uses limited *post-fabrication* measurements to reduce the uncertainty of the oxide thickness for any unmeasured device. The proposed method accounts for both inter-chip (global), intra-chip (within-chip) spatially correlated and random residual components [8]. We compute the inter-chip component using a maximum-likelihood estimation method and the other two by leveraging the spatial correlation between devices and then constructing a conditional distribution based on the post-fabrication measurements, while still preserving the correlation between devices in a conditional covariance matrix. Based on the conditional distribution, the second phase applies principal component analysis to predict the chip reliability. The principal components are employed to derive a tightened lifetime distribution of a particular chip for a given reliability target. The chip lifetime is then bounded by certain confidence-level interval, the lower bound of which is conservatively used for lifetime evaluation. Finally, in the third phase, we present an optimization flow for efficient tuning of the chip maximum supply voltage. As a result, with proper reliability management, we can boost chip performance for many chips while maintaining or improving reliability.

The effectiveness of the proposed methodology is validated on several benchmarks varying in size from 0.08 to 50 million devices in 65nm CMOS technology. The results show that with merely 25 measurements of oxide thickness, we can achieve performance improvement of 15-19% on average and

22-27% at maximum for all the designs compared with the conventional guard-band based approach that assumes the absence of any measurements. The average runtime per chip is only approximately 0.4 second, which is sufficiently fast to allow programming of the maximum supply voltage during post-silicon testing.

4.1 Review of Oxide Breakdown Reliability Analysis

In the last chapter we discuss in details about the oxide thickness variation modeling and variation-aware oxide breakdown reliability analysis. Here, for the readers' convenience, we briefly review the oxide breakdown reliability analysis.

Conventionally, the gate oxide degradation is considered to depend on oxide thickness, transistor area, supply voltage, and temperature. Although many of the physical details are still under debate, most models note the non-deterministic process of defect generation, eventually resulting in a statistically distributed oxide breakdown time and the strong dependence of this random process on oxide thickness [55, 57]. A common failure criterion for OBD is soft breakdown (SBD) characterized by a small increase in gate leakage and eventually followed by un-recoverable hard breakdown (HBD). Due to the stochastic process nature, the oxide breakdown time for SBD is modeled as a random variable following a Weibull probability distribution [55]:

$$F(t) = 1 - e^{-a(\frac{t}{\alpha})^{\beta}} \tag{4.1}$$

where F is the cumulative distribution function (cdf) of time-to-breakdown t, a is the device area normalized with the minimum device area, α and β are the scale and shape parameters of the Weibull model. β can be further expressed as bx for a given temperature and voltage stress, where x is the gate oxide thickness of a device. The reliability function of a device can then be simply written as:

$$R(t) = P(T > t) = 1 - F(t) = e^{-a(\frac{t}{\alpha})^{bx}} \tag{4.2}$$

Due to the non-deterministic characteristic of oxide thickness at design time, the device reliability function can be interpreted as the conditional reliability function given its oxide thickness and written as $R(t|x_i)$. The overall chip-level reliability function is then given by:

$$R_c(t) = \int\limits_0^{\infty} \dots \int\limits_0^{\infty} \prod_{i=1}^{m} R_i(t|x_i) f(x_1 \dots x_m) dx_1 \dots dx_m \tag{4.3}$$

where $f(x_1, \ldots, x_m)$ is the joint probability density function (pdf) of the gate oxide thicknesses for m devices. To handle the tremendous dimensionality of (4.3), Chapter III proposed to project the parametric space to two distinct random variables, sample mean (u) and variance (v) of the chip oxide thickness distribution. Based on this, the original product $\prod_{i=1}^{m} R_i(t|x_i)$ was simplified to a conditional probability $R_c(t|u, v)$ as in Chapter III. The integral of (4.3) is then compactly expressed as [38]:

$$R_c(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_c(t|u, v) f_{\mathbf{uv}}(u, v) du\ dv \tag{4.4}$$

where

$$R_c(t|u, v) = \exp[-Ae^{\ln(\frac{t}{\alpha})bu + (\ln(\frac{t}{\alpha}))^2 b^2 v/2}] \tag{4.5}$$

and $f_{\mathbf{uv}}(u, v)$ is the joint pdf of a Gaussian random variable u and a chi-square random variable v.

However, neither the method in Chapter III nor the guard-band method in [58] allows for incorporation of oxide thickness measurements and is unable to distinguish the unique condition of a particular chip. These methods therefore result in one global lifetime estimation for the entire ensemble of chips, and unnecessarily degrade the performance for most of them.

4.2 Post-Fabrication Measurement-Driven Oxide Thickness Estimation

In this section, we will show that even with a relatively small number of oxide thickness measurements, it is possible to reduce the uncertainty of oxide thicknesses for a particular chip, and hence provide significantly more accurate lifetime estimation. However, due to the tremendous number of unmeasured devices and the constrained stochastic process nature of chip reliability, such estimation of oxide thicknesses for unmeasured devices is a difficult problem that has not been addressed to date. This section presents a statistical method to address this problem.

4.2.1 Problem Formulation

Give a *particular* chip, the inter-chip and intra-chip variation components (spatially correlated and random) play very different roles in the final transistor oxide thickness. The inter-chip component induces the same increment or decrement to the oxide thicknesses for all the devices within the chip and is a constant in (1.1). On the other hand, the intra-chip spatially correlated and random components are different from device to device. In reality we cannot distinguish the sources of the variation when the number of measurements is limited. Thus, we combine the two intra-chip variation components together in analysis and comprehensively evaluate their impact.

Given a chip dissected to N grids as in [47] with m devices in total, the vector of oxide thicknesses for all the devices can be written as:

$$\mathbf{x} = z_0 + z_g + \mathbf{z_{corr}} + \mathbf{z}_\epsilon = u_{chip} + \mathbf{z_{intra}} \tag{4.6}$$

where $\mathbf{x}$=$[x_1, x_2, \ldots, x_m]$ is the oxide thicknesses for m devices; u_{chip}=$z_0 + z_g$ denotes the *chip-level oxide thickness mean* for this particular chip and may be different from one chip to another; $\mathbf{z}_\epsilon$ is the vector containing the random variation component of each device; $\mathbf{z_{intra}}$=$\mathbf{z_{corr}}$+$\mathbf{z}_\epsilon$ is hence the combined intra-chip variation component that preserves the spatial correlation between devices. Since $\mathbf{z}_\epsilon$ can be interpreted as a multi-variate Gaussian process $\mathcal{N}_m(0, \sigma_\epsilon^2 I_m)$, where I_m is an $m \times m$ identity matrix, $\mathbf{z_{intra}}$ is then the sum of two multi-variate Gaussians and remains a multi-variate Gaussian process $\mathcal{N}_m(0, \Sigma_{intra})$, where $\Sigma_{intra} = \Sigma_{corr} + \sigma_\epsilon^2 I_m$.

The post-fabrication measurement-driven oxide thickness estimation problem is then formulated as:

Formulation: *Given the thickness variation model in* (4.6) *and the oxide thickness measurements of* n_0 *devices across a particular chip, estimate the oxide thickness of any unmeasured device, including the components of* u_{chip} *and* $\mathbf{z_{intra}}$ *as well as the corresponding variance.*

In other words, we need to characterize a conditional distribution for the oxide thickness of each unmeasured device on the condition of the post-fabrication measurements. This will reduce the variance of the conditional distribution compared with the initial model in (4.6), and results in a tightened lifetime distribution. In the following, we present the techniques to solve the above formulation.

4.2.2 Model Simplification

The grid-based spatial correlation model in [47] indicates that devices within one grid bear approximately the same inter-chip and intra-chip spatially correlated variation components. This is reasonable when we have relatively finer grids across the chip. The difference in oxide thicknesses for devices *within* one grid are then completely attributed to the random variation component, which is independent from one device to another and hence cannot be predicted. Thus, instead of performing device-level estimation and predicting device by device within one grid, we employ a grid-based prediction scheme by associating every grid with one random variable and hence achieve one estimation for each grid, including a random variation component and correlation to other grids. Clearly, such modeling greatly simplifies the complexity from the dimensionality of millions (number of devices) to $N + n_0$, where n_0 is the number of measurement sites and N denotes the number of unmeasured sites with each representing one grid.

We then re-formulate the model in (4.6) to the granularity of a grid. Both $\mathbf{x}$ and $\mathbf{z_{intra}}$ are now $(N+n_0)\times 1$ vectors. $\mathbf{z_{intra}}$ follows $\mathcal{N}_{N+n_0}(0, \Sigma_{intra,grid})$, where $\Sigma_{intra,grid}$ is an $(N+n_0)\times(N+n_0)$ covariance matrix for N unmeasured sites and n_0 measured sites.

4.2.3 Estimation of the Chip-Level Oxide Thickness Mean u_{chip}

As discussed, we need to treat the deterministic component and random component in (4.6) separately. Removal of the mean from the random data is an integral and essential step to minimize the mean square error of the estimation [74]. In this subsection, we detail the estimation of the chip-level oxide thickness mean u_{chip}.

Before measurement, the oxide thickness for the sites to be measured remain unknown and hence can be characterized by a multi-variate Gaussian model, $\mathcal{N}_{n_0}(u_{chip}, \Sigma_{mm})$. The measured thicknesses $\mathbf{s}$=$[s_1, s_2 ... s_{n_0}]$ are therefore a *sample vector* drawn from this stochastic model, with measurements acting as n_0 observations. Thus, by using the maximum likelihood estimation (MLE), the log-likelihood function is [74]:

$$\begin{aligned} \ell(\mathbf{s}|u_{chip}) = &-\ln((2\pi)^{n_0/2}|\Sigma_{mm}|^{1/2}) \\ &-\frac{1}{2}(\mathbf{s} - u_{chip} \times [\mathbf{1}]_{1\times n_0})\Sigma_{mm}^{-1}(\mathbf{s} - u_{chip} \times [\mathbf{1}]_{1\times n_0})^T \end{aligned} \tag{4.7}$$

where $[\mathbf{1}]_{1\times n_0}$ denotes a $1\times n_0$ all-one vector. The maximum in (4.7) is achieved when:

$$u_{chip} \approx \frac{[\mathbf{1}]_{1\times n_0}\Sigma_{mm}^{-1}}{[\mathbf{1}]_{1\times n_0}\Sigma_{mm}^{-1}[\mathbf{1}]_{1\times n_0}^T}\mathbf{s}^T \tag{4.8}$$

The corresponding MLE estimation variance can be approximately bounded by the Cramer-Rao bound [74]:

$$\mathrm{var}(u_{chip}) \approx [\mathbf{1}]_{1\times n_0}\Sigma_{mm}^{-1}[\mathbf{1}]_{1\times n_0}^T \tag{4.9}$$

Since the number of measurements n_0 is limited to fewer than hundreds, the matrix inverse in (4.8) can be efficiently computed within seconds.

4.2.4 Estimation of the Intra-Chip Variation Component $\mathbf{z_{intra}}$

If every site of a chip could be measured, the variance for the random vector $\mathbf{x}$ would be reduced to 0. Since the number of measurements is limited, measured oxide thicknesses can only reduce the variance of unmeasured sites, which can still provide designers with significantly more accurate information of chip oxides condition.

In order to assess the impact of measurements, we separate the oxide thickness vector $\mathbf{x}$ into two sub-vectors as $\mathbf{x} = [\mathbf{s}, \mathbf{x_u}]$, where $\mathbf{s}$ represents the sites to be measured and $\mathbf{x_u}$ represents the unmeasured sites. $\Sigma_{intra,grid}$ then can be expressed as:

$$\Sigma_{intra,grid} = [\begin{array}{cc} \Sigma_{mm} & \Sigma_{mu} \\ \Sigma_{um} & \Sigma_{uu} \end{array}] \tag{4.10}$$

where in each sub-matrix, "m" is for the sites to be measured (vector "$\mathbf{s}$") and "u" is for the unmeasured sites (vector "$\mathbf{x_u}$"). The entries in any sub-matrix can be simply obtained from the covariance matrix Σ_{intra} in (4.6) by identifying the grids the sites belong to. Note that both $\mathbf{s}$ and $\mathbf{x_u}$ are multi-variate Gaussian processes with a mean of u_{chip} and a covariance matrices of Σ_{mm} and Σ_{uu}, respectively.

Given the measurement values $\mathbf{s} = \mathbf{s_0}$ at n_0 sites, the sub-vector $\mathbf{x_u}$ for the oxide thicknesses at unmeasured sites can then be expressed in a conditional way, *i.e.*, $\mathbf{x_u}|\mathbf{s} = \mathbf{s_0}$. Such expression illustrates the impact of measurements on unmeasured sites. By exploiting the spatial correlation between $\mathbf{x_u}$ and $\mathbf{s}$, the pdf for this conditional random vector can be written as:

$$f_{\mathbf{x_u}|\mathbf{s}=\mathbf{s_0}}(\mathbf{x_u}) = \frac{f_{\mathbf{x}}(\mathbf{x_u}, \mathbf{s} = \mathbf{s_0})}{f_{\mathbf{s}}(\mathbf{s} = \mathbf{s_0})} \tag{4.11}$$

where $f_{\mathbf{x}}(\mathbf{x})$ and $f_{\mathbf{s}}(\mathbf{s})$ are pdf's for the multi-variate Gaussian random vectors $\mathbf{x}$ and $\mathbf{s}$, respectively; $f_{\mathbf{x_u}|\mathbf{s}=\mathbf{s_0}}(\mathbf{x_u})$ is the conditional pdf for $\mathbf{x_u}$ given $\mathbf{s} = \mathbf{s_0}$. Due to space limitation, we only provide an outline of the deduction.

Based on the decomposition of the covariance matrix in (4.10), we define:

$$\mathbf{u_{x_u|s}} = u_{chip} + (\mathbf{s} - u_{chip})\Sigma_{mm}^{-1}\Sigma_{mu} \tag{4.12}$$

$$\Sigma_{\mathbf{x_u}|\mathbf{s}} = \Sigma_{uu} - \Sigma_{um}\Sigma_{mm}^{-1}\Sigma_{mu} \tag{4.13}$$

and then obtain $|\Sigma_{intra,grid}| = |\Sigma_{mm}||\Sigma_{\mathbf{x_u}|\mathbf{s}}|$. Thus, when $\mathbf{s} = \mathbf{s_0}$, the conditional pdf in (4.11) can be expressed as:

$$f_{\mathbf{x_u}|\mathbf{s}=\mathbf{s_0}}(\mathbf{x_u}) = \frac{1}{(2\pi)^{N/2}|\Sigma_{\mathbf{x_u}|\mathbf{s}}|^{1/2}} \times \exp[\frac{-(\mathbf{x_u} - \mathbf{u}_{\mathbf{x_u}|\mathbf{s}=\mathbf{s_0}})\Sigma^{-1}_{\mathbf{x_u}|\mathbf{s}}(\mathbf{x_u} - \mathbf{u}_{\mathbf{x_u}|\mathbf{s}=\mathbf{s_0}})^T}{2}] \tag{4.14}$$

where $\mathbf{u}_{\mathbf{x_u}|\mathbf{s}=\mathbf{s_0}}$ and $\Sigma_{\mathbf{x_u}|\mathbf{s}}$ defined in (4.12) and (4.13) are conditional mean and covariance matrix for the conditioned random vector $\mathbf{x_u}|_{\mathbf{s}=\mathbf{s_0}}$. The details of the conditional distribution can be derived from general principles in [51], which are widely employed in various works [73, 78].

Intuitively speaking, the vector $\mathbf{u}_{\mathbf{x_u}|\mathbf{s}=\mathbf{s_0}}$ provides a natural estimation of the oxide thickness at the unmeasured sites, whereas the diagonal entries of $\Sigma_{\mathbf{x_u}|\mathbf{s}}$ evaluate the variance of the estimation. Note that every entry in the covariance matrix is positive and that both Σ_{uu} and Σ_{mm} are positive definite [8]. The conditional variance in $\Sigma_{\mathbf{x_u}|\mathbf{s}}$ is therefore reduced compared with the unconditional variance in (4.10).

Although the oxide thicknesses for closely-placed devices are non-continuous due to random variation, the spatial correlation still allows us to explore the relationship among devices and achieves improved prediction as the number of measurements increases. Figure 4.3 illustrates the trend of variance reduction of the conditional estimator $\mathbf{u}_{\mathbf{x_u}|\mathbf{s}=\mathbf{s_0}}$ for a randomly selected site with respect to the growing number of measurements. It is noted that with only 9 measurements, the variance of $u_{x_u,i|\mathbf{s}=\mathbf{s_0}}$, as computed in (4.13), is reduced by 63% compared with the initial variance when no measurement is conducted.

4.2.5 Chip-Level Oxide Thickness Mean Refinement

Unlike the unconditional random vector in (4.6) where all the variables share the same mean u_{chip}, the conditioned random vector in (4.12) may bear different mean values. This is closer to the realistic condition where the oxide thickness shows variation across the die, and hence provides the chance to refine the chip-level oxide thickness mean, using both the measured and unmeasured sites.

In theory, the chip-level oxide thickness mean is equal to the sample mean of all the sites, denoted as $\overline{x}_{N+n_0}$:

$$\overline{x}_{N+n_0} = \frac{1}{N+n_0}(\mathbf{s_0} \times [\mathbf{1}]_{n_0\times 1} + \mathbf{u}_{\mathbf{x_u}|\mathbf{s}=\mathbf{s_0}} \times [\mathbf{1}]_{N\times 1}) \tag{4.15}$$

The deviation between $\overline{x}_{N+n_0}$ and u_{chip} is primarily due to the estimation error and may degrade the analysis effectiveness. Thus, we can perform a refinement step iteratively to reduce the deviation to

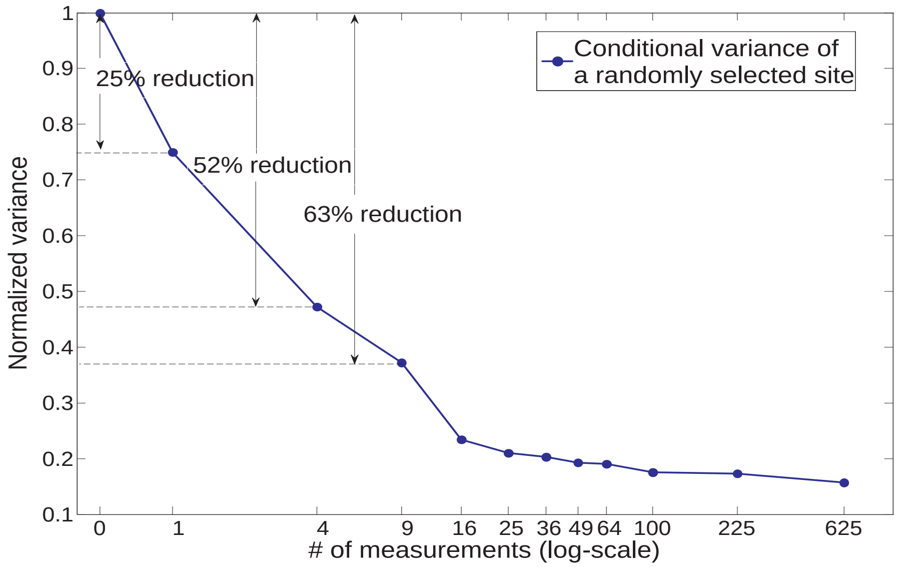


Figure 4.3: Reduction in variance of the conditional estimator $\mathbf{u_{x_u|s=s_0}}$ for a randomly selected unmeasured site with increased measurements.

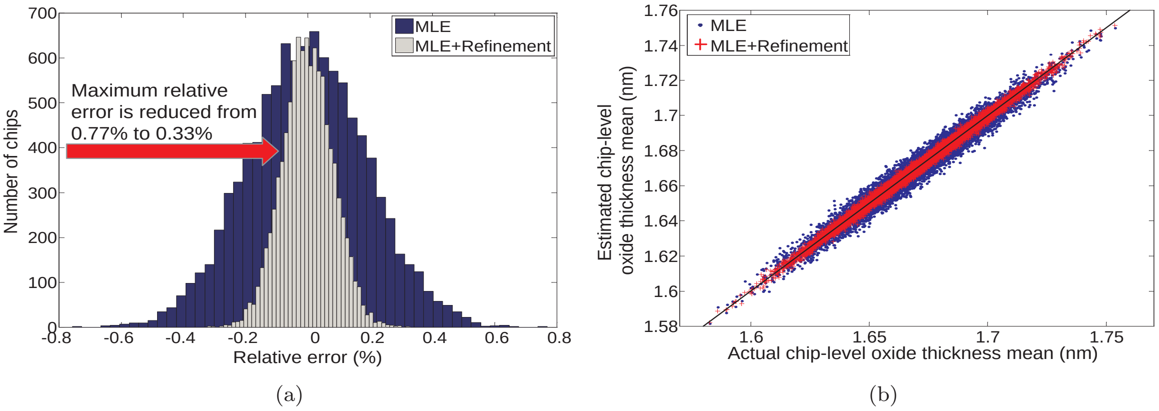
```
Procedure: Post-Fabrication Measurement-Driven Estimation
Input: measurements $\mathbf{s_0}$, process variation model in (4.6)
Output: Oxide thickness estimation for each device and the corresponding estimation variance
1:    Simplify the model as in Section 4.2.2;
2:    Compute the chip-level oxide thickness mean and corresponding variance using (4.8) and (4.9);
3:    Estimate the intra-chip variation component $\mathbf{z_{intra}}$ using (4.12)-(4.14);
4:    Perform chip-level oxide thickness mean refinement;
5:    Map the estimation and corresponding variance at the granularity of grid level to the devices in the same grid;
```

Figure 4.4: Post-fabrication measurement-driven oxide thickness estimation.

a negligible level, *i.e.*, to make $u_{chip} \approx \overline{x}_{N+n_0}$ by repeatedly replacing u_{chip} in (4.12) with $\overline{x}_{N+n_0}$ and then computing $\overline{x}_{N+n_0}$ with (4.15). In general the refinement is completed within tens of iterations to reach certain tolerance, *e.g.*, 10^{-5}. Moreover, it is worthwhile to note that either the estimation variance in (4.9) or the conditional covariance matrix in (4.13) does not rely on u_{chip} and remains unchanged for the updated chip-level oxide thickness mean.

4.2.6 Summary of Post-Fabrication Measurement-Driven Estimation

We summarize the procedure for post-fabrication measurement-driven estimation in Figure 4.4. The complexity of the procedure is very low as most computations are analytically achievable. The matrix inverse Σ_{mm}^{-1} and matrix product in (4.13) are two operations with relatively higher complexly, which depend on the spatial correlation structure of the design and only need to be computed once



(a)

(b)

Figure 4.5: Accuracy of chip-level oxide thickness mean estimation: (a) Histograms of relative errors for maximum likelihood estimation (MLE) and maximum likelihood estimation with refinement (MLE+Refinement) (b) Scatter plots for MLE and MLE+Refinement.

for a particular design with fixed measurement sites. Since the number of measurements is limited to fewer than hundreds, those operations can be numerically computed within seconds. Note that the procedure produces a single random value per grid which is the representative for all the devices in the grid. The estimation for this site is then eventually projected to all the other devices within the same grid to compute the reliability of the chip.

We apply the proposed procedure to 10000 chips in 65nm technology. Each chip has 0.5 million devices and is imposed a 50×50 (=2500) grids with 100 uniformly-distributed measurement sites, which are selected in a chessboard manner. The estimated chip-level oxide thickness mean u_{chip} is compared with the actual mean of the oxide thicknesses for all the devices in Figure 4.5. From either the histogram or the scatter plot, it can be seen that the estimation achieved by maximum likelihood estimation (MLE) in Section 4.2.3 is very accurate with a maximum relative error of 0.77% while the mean refinement algorithm (in Section 4.2.5) can further reduce the relative error to 0.33%. We then examine the estimation accuracy at the device level (achieved by step 5 in Figure 4.4) for a randomly selected chip. Figure 4.6 demonstrates the contour of the difference between the actual oxide thickness and the estimated thickness mapped from $\mathbf{u}_{\mathbf{x_u}|\mathbf{s}=\mathbf{s_0}}$ for all the devices on a chip. One can see that with 100 measurements, the accuracy of the oxide thickness estimation for each device is already very high, with average relative error of 0.59% and maximum relative error of 2.8%. Those errors are mainly due to the unpredictable random residual variation but are bounded by the covariance matrix $\Sigma_{\mathbf{x_u}|\mathbf{s}}$.

[8] Actual time to failure is a stochastic process and cannot be known until the chip fails. Thus, we introduce a quantile-based time-to-failure which can be interpreted as certain quantile of the time-to-failure distribution. In other words, it is the actual time when chip meets certain reliability target. Note that this value is a deterministic value if the oxide thicknesses of all the devices are known.

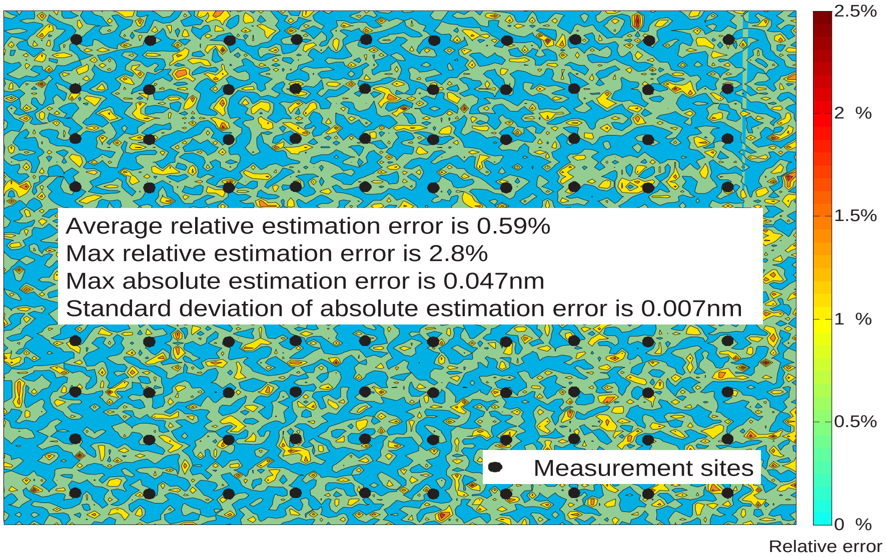


Figure 4.6: Contour of the device-level oxide thickness estimation error for a chip with 0.5M devices and 100 measurements.

4.3 Measurement-Driven OBD Reliability Prediction and Management

Using the proposed oxide thickness estimation with corresponding variance, we now perform a statistical reliability analysis to tighten the lifetime distribution. Here we focus on chip-level reliability analysis and consider the worst-case operating temperature to ensure a correct operation throughout the entire lifetime. The temperature and voltage drop variations can be easily incorporated in our flow by performing analysis at the granularity of functional blocks or sub-blocks, where devices within a block can be assumed to have the same temperature and supply voltage drop.

Given a chip with m devices and N grid cells for spatial correlation modeling, we define the following notations for the remainder of the paper as in Table 4.1.

4.3.1 Post-Fabrication Measurement-Driven Reliability Prediction

The challenge to the chip-level statistical OBD reliability analysis is the huge dimensionality of the integral in (4.3). Chapter III proposed to map millions of random variables to two random variables, sample mean and variance of the chip oxide thickness distribution. However, for a conditioned random vector $\mathbf{x_u}|\mathbf{s_0}$, the variables do not bear the same mean and cannot employ the method in Chapter III. The conditional covariance matrix (or spatial correlation) also shows completely different features from the unconditional case. We therefore present a measurement-driven OBD reliability analysis in this subsection.

Table 4.1: Notations used in OBD reliability analysis

Notation	Definition
$\mathbf{x} = [x_1, \ldots, x_m]$	the oxide thicknesses for m device of a chip
$\mathbf{x_u} \vert \mathbf{s_0}$	the conditional random vector for oxide thicknesses of unmeasured sites, given the measured oxide thicknesses of $\mathbf{s_0}$
$\overline{x}_m = \frac{\sum_{i=1}^{m} x_i}{m}$	the sample mean for m devices of a chip
$v = \frac{\sum_{i=1}^{m} (x_i - \overline{x}_m)^2}{m-1}$	the sample variance for m device of a chip
$R(t_0)$	chip reliability at time t_0, which is $\Pr(t > t_0)$
T_{target}	chip design lifetime target
R_t	chip reliability target at the end of lifetime
T_q	quantile-based time-to-failure (QTTF)[8], defined as $T_q = \arg_{T_q}\{R(T_q) = \Pr(t > T_q) = R_t\}$.
$D_0 = [d_1, ...d_N]$	d_i denotes the number of unmeasured devices in the i_{th} grid
$D = diag(D_0)$	a diagonal matrix with diagonal vector of D_0

4.3.1.1 Conditional Spatial Correlation Characterization Using Principal Component Analysis

Since $\mathbf{x_u}|\mathbf{s_0}$ is still a multi-variate Gaussian random vector, its correlation structure in (4.13) can be simplified by principal component analysis (PCA) to map the correlated variation components to another set of mutually independent random variables with zero mean and unit variance [38, 47]. For a device in the i_{th} grid, its conditional oxide thickness $x_{u,i}|\mathbf{s_0}$ can be canonically expressed as a linear combination of the principal components:

$$x_{u,i}|\mathbf{s_0} = u_{x_u,i|\mathbf{s}=\mathbf{s_0}} + \sum\nolimits_{j=1}^{N} \lambda_{i,j} z_j \tag{4.16}$$

where N is the number of principal components; z_j's represent the N independent random variables used to characterize the conditional spatially correlated variation; and the coefficients $\lambda_{i,j}$'s represent the sensitivity of thickness variation with respect to the j_{th} principal component for the random variable in the i_{th} grid. Thus, the conditional random vector of N unmeasured sites can be written compactly with principal components:

$$\mathbf{x_u}|\mathbf{s_0} = \mathbf{u}_{\mathbf{x_u}|\mathbf{s}=\mathbf{s_0}} + \mathbf{z} \times P_\lambda \tag{4.17}$$

where P_λ is an $N \times N$ matrix containing the sensitivity coefficients $\lambda_{i,j}$'s for different principal components and can be achieved by eigenvalue decomposition; $\mathbf{z} = [z_1, z_2, ...z_N]$ is a vector of principal components.

We can then estimate the *conditional* sample mean and sample variance for devices across the chip in terms of principal components. As defined earlier, $\overline{x}_m$ and v are:

$$\overline{x}_m = [(\mathbf{x_u}|\mathbf{s_0})D_0^T + \mathbf{s_0} \times [\mathbf{1}]_{n_0 \times 1}]/m \tag{4.18}$$

$$v = \frac{(\mathbf{x_u}|\mathbf{s_0} - \overline{x}_m)D(\mathbf{x_u}|\mathbf{s_0} - \overline{x}_m)^T + (\mathbf{s_0} - \overline{x}_m)(\mathbf{s_0} - \overline{x}_m)^T}{m-1} \tag{4.19}$$

Those two variables $\overline{x}_m$ and v illustrate the underlying characteristics of the conditional chip oxide thickness distribution given measurements $\mathbf{s_0}$ at n_0 sites.

By noting the equality of u_{chip} in (4.15), (4.18) is simplified to:

$$\overline{x}_m = u_{chip} + \mathbf{u_{coeff}}\mathbf{z^T} \tag{4.20}$$

where $\mathbf{u_{coeff}} = \frac{1}{m} D_0 P_\lambda^T$. Clearly $\overline{x}_m$ remains a Gaussian with mean of u_{chip} and variance as the following:

$$\text{var}(\overline{x}_m) = \text{var}(u_{chip}) + \mathbf{u_{coeff}}\mathbf{u_{coeff}}^T \tag{4.21}$$

After expanding the numerator of (4.20), we can re-write v as the sum of two random variables V_1 and V_2:

$$v = (V_0 + 2V_1 + V_2)/(m-1) \tag{4.22}$$

where

$$\begin{aligned} &V_0 = (\mathbf{u_{x_u|s=s_0}} - u_{chip})D(\mathbf{u_{x_u|s=s_0}} - u_{chip})^T + (\mathbf{s}_0 - u_{chip})(\mathbf{s}_0 - u_{chip})^T \\ &V_1 = \mathbf{v_{coeff}}\mathbf{z}^T \quad \text{and} \quad V_2 = \mathbf{z}V\mathbf{z}^T \end{aligned} \tag{4.23}$$

with $\mathbf{v_{coeff}} = \mathbf{u_{x_u|s=s_0}} D P_\lambda^T - (m \times u_{chip} - \mathbf{s_0}[\mathbf{1}]_{n_0\times 1})\mathbf{u_{coeff}}$ and $V = (P_\lambda^T + [\mathbf{1}]_{N\times 1}\mathbf{u_{coeff}})^T D (P_\lambda^T - [\mathbf{1}]_{N\times 1}\mathbf{u_{coeff}})$.

Note that V_0 is a constant and V_1 is a normal random variable. Since the matrix V is positive and symmetric, V_2 has the form of quadratic normal product and can be approximated by a chi-square distribution [71], $V_2 \sim \hat{a}\chi^2_{\hat{b}}$, with $\hat{a} = \frac{tr(V^2)}{tr(V)}$ and $\hat{b} = \frac{[tr(V)]^2}{tr(V^2)}$, where $tr[.]$ denotes the trace operation to compute the sum of diagonal entries. Since $E(V_1)E(V_2)$=$E(V_1 V_2)$, V_1 and V_2 turn out to be

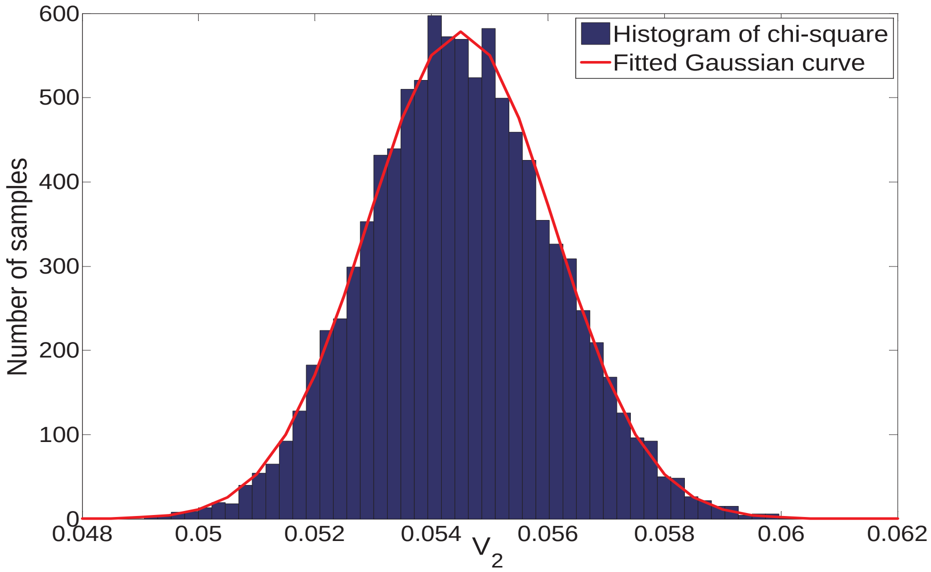


Figure 4.7: Comparison of the histogram of chi-square random variable V_2 in (4.23) with degree of freedom of 2209 (N=2500) and the fitted Gaussian curve. The fitting goodness is 0.98 (R-square).

uncorrelated. Moreover, by noting the degree of freedom for the chi-square distribution $\hat{b} = \frac{[tr(V)]^2}{tr(V^2)}$ is close to N, the chi-square distribution with a large degree of freedom can be well approximated by a Gaussian distribution [74], which is validated by the histogram of V_2 in Figure 4.7. Thus, the un-correlation between two Gaussian random variables V_1 and V_2 implies their independence. In other words, v is a Gaussian random variable, the mean and variance of which can be computed from (4.22) and (4.23):

$$E(v) = [V_0 + tr(V)]/(m-1) \tag{4.24}$$

$$\mathrm{var}(v) = \frac{2tr(V^2)}{(m-1)^2} + \frac{4}{(m-1)^2}\mathbf{v_{coeff}}\mathbf{v_{coeff}}^T \tag{4.25}$$

4.3.1.2 Post-Fabrication Measurement-Driven Lifetime Prediction

Once the underlying distribution of $\overline{x}_m$ and v are characterized, we can conduct the post-fabrication measurement-driven reliability prediction for a particular chip and analyze the quantile-based time-to-failure (QTTF) [9] for certain reliability target R_t by using (4.5):

$$R(T_q|\overline{x}_m, v) = \exp[-Ae^{\ln(\frac{T_q}{\alpha})b\overline{x}_m + (\ln(\frac{T_q}{\alpha})b)^2 v/2}] = R_t \tag{4.26}$$

[9] T_q is defined as T_q=$\arg_{T_q}\{R(T_q)$=$\Pr(t{>}T_q)$=$R_t\}$. In other words, it is the quantile of reliability distribution for certain reliability target R_t.

where A is the chip area. This equality illustrates the actual quantile-based time-to-failure when chip meets ceratin reliability target. The quantile-based time-to-failure is then compared with design lifetime T_{target} to evaluate chip reliability. To simplify the analysis, we introduce a supplementary random variable $\gamma = \ln(T_q/\alpha)b$ to derive the distribution of T_q (QTTF), and rewrite the equation above as:

$$v/2 \times \gamma^2 + \overline{x}_m \times \gamma - \ln(-\ln(R_t)/A) = 0 \tag{4.27}$$

This quadratic equation can be easily solved:

$$\gamma = \gamma(\overline{x}_m, v) = \frac{-\overline{x}_m + \sqrt{\overline{x}_m^2 + 2\ln(-\ln(R_t)/A) \times v}}{v} \tag{4.28}$$

In other words, when the reliability target R_t is given, γ is a random function depending on the underlying distributions of $\overline{x}_m$ and v.

By noting that both $\overline{x}_m$ and v have limited variance, we can further simplify (4.28) with first-order Taylor expansion:

$$\gamma \approx \gamma(E(\overline{x}_m), E(v)) + [\frac{\partial\gamma(\overline{x}_m, v)}{\partial\overline{x}_m}, \frac{\partial\gamma(\overline{x}_m, v)}{\partial v}]|_{E(\overline{x}_m),E(v)} \times [\overline{x}_m - E(\overline{x}_m), v - E(v)]^T \tag{4.29}$$

Since both $\overline{x}_m$ and v are Gaussians and almost uncorrelated, we can reasonably justify that γ follows a Gaussian process with mean and variance:

$$E(\gamma) = \frac{-E(\overline{x}_m) + \sqrt{E(\overline{x}_m)^2 + 2\ln(-\ln(R_t)/A) \times E(v)}}{E(v)} \tag{4.30}$$

$$\mathrm{var}(\gamma) = [(\frac{\partial\gamma}{\partial\overline{x}_m})^2, (\frac{\partial\gamma}{\partial v})^2]|_{E(\overline{x}_m),E(v)} \times [\mathrm{var}(\overline{x}_m), \mathrm{var}(v)]^T \tag{4.31}$$

Quantile-based time-to-failure T_q can then be characterized as a lognormal distribution, as $T_q = \alpha \exp[\gamma/b]$.

4.3.2 OBD Reliability Management and Performance Optimization

The technique in Section 4.3.1 can characterize the distribution of quantile-based time-to-failure and achieve a well-tightened lifetime distribution. In practice, the design objective may be a certain design lifetime T_{target} with a predefined reliability requirement R_t, *i.e.*, the probability of chip failure may not exceed $1-R_t$ within T_{target} years lifetime. However, due to the process variation, some chips will have thinner oxides and are quicker to fail. The tightened distribution of T_q (QTTF) enables us to quantitatively evaluate whether the chip will meet the design lifetime target or not. Those chips that are prone to failure can be tuned to a lower supply voltage limit to improve the reliability yield. On the other hand, chips with thicker oxides can operate at a higher voltage for better performance. The next question is then how much voltage we need to tune to optimize the performance, which will be discussed in the following optimization flow.

Since QTTF itself is a distribution due to the remaining uncertainty of the oxide thicknesses, we use the lower bound of the distribution with a certain confidence to ensure a robust design. Conservatively, with a 99.9% confidence level, we can derive the following one-sided confidence interval:

$$T_q \in [\alpha \exp\left[\frac{E(\gamma) - 3\sqrt{\mathrm{var}(\gamma)}}{b}\right], \infty] \tag{4.32}$$

where the moments of γ can be computed from (4.30) and (4.31). The lower bound of (4.32) is then denoted as T_{lb} and used to evaluate chip lifetime in optimization. In other words, after optimization, we may push the distribution of QTTF to the right of T_{target} and have 99.9% confidence that the chip will meet the lifetime target. Since both parameters α and b in (4.32) depend on supply voltage, we formulate the following to maximize the supply voltage while T_{lb} meets the design lifetime target:

Maximize

$$v_{chip} \tag{4.33}$$

Subject to:

$$\ln(T_{lb}) = \ln(\alpha(v_{chip})) + \frac{E(\gamma) - 3\sqrt{\mathrm{var}(\gamma)}}{b(v_{chip})} \geq \ln(T_{target}) \tag{4.34}$$

$$v_{min} \leq v_{chip} \leq v_{max} \tag{4.35}$$

where v_{chip} denotes the supply voltage; the first constraint in (4.34) implies that the 99.9% confidence lower bound of QTTF is larger than the design lifetime target; and the second constraint in (4.35) denotes the possible voltage tuning range. We find that this optimization problem is equivalent to finding the feasible domain of the inequality in (4.34), where the parameters of the device reliability

```
Procedure: Post-Fabrication Measurement-Driven OBD Reliability Prediction and Management
Input: measurements $\mathbf{s_0}$, process variation model in (4.6), reliability target and design lifetime
Output: optimized supply voltage
1: Given $\mathbf{s_0}$, estimate the conditional oxide thickness and covariance matrix with the flow in Figure 4.4;
2: Apply PCA to the conditional covariance matrix to obtain the distributions of $\overline{x}_m$ and $v$ using (4.20)-(4.25);
3: Estimate tightened chip lifetime distribution using (4.30) and (4.31);
4: Solve the optimization problem in (4.33)-(4.35) to achieve the optimized supply voltage;
```

Figure 4.8: Post-fabrication measurement-driven OBD reliability prediction and management.

function, $\alpha(v_{chip})$ and $b(v_{chip})$, indicate the underlying dependence on supply voltage. Since we only have one variable, even with a complicated physics-based model for $\alpha(v_{chip})$ and $b(v_{chip})$, we can still efficiently solve this problem in a numerical way. In our implementation we adopt the linear models in [55, 57], *i.e.*, $\ln(\alpha(v_{chip}))=a_1\times v_{chip}+a_2$ and $b(v_{chip}) = b_1 \times v_{chip} + b_2$, and hence have a quadratic inequality in (4.34), which can be analytically computed. As a result, the optimization flow above eventually reduces the failure rate to improve reliability yield, while the overall performance is also enhanced by reducing lifetime safety margins.

4.3.3 Summary of OBD Reliability Prediction and Management

The procedure for post-fabrication measurement-driven reliability prediction and management is summarized in Figure 4.8. Given n_0 measurements for a particular chip, we first estimate the oxide thicknesses and corresponding variance using a conditional multi-variate Gaussian model. The conditional spatial correlation is then explored by PCA to derive the distributions of $\overline{x}_m$ and v, which characterize the underlying conditional chip oxide thickness distribution and help achieve a tightened lifetime distribution. The lifetime estimation then allows an optimization flow to quantify trade-offs between reliability and supply voltage/performance.

4.4 Experimental Results

The proposed reliability prediction and management methodology was implemented and tested on several designs using 65nm LP devices (nominal oxide thickness is 1.67nm). The defect generation relationships for the technology node and the technology dependent parameters of the oxide reliability function model are obtained from [55, 57]. In practice, this can be obtained by a one time per technology characterization using test devices [77]. For each design, we used 10000 chips that follow the thickness variation model in (1.1) of Chapter I. The overall $3\sigma/\mu$ of oxide-thickness variation was set to 4% of the nominal value as in [6] and then split into three variation components. The relationship between oxide thickness, voltage and performance is determined by SPICE simulation on 65nm standard cells and then used in our methodology for reliability management and voltage tuning.

4.4.1 Efficacy of the Proposed OBD Reliability Prediction

Given the post-layout design implementation, a process variation model and limited measurements on device oxide thickness, the proposed method can estimate the quantile-based time-to-failure (QTTF) distribution for a certain reliability target, with which we can examine whether this chip may meet the design lifetime or not. To evaluate the accuracy of the proposed method, the conditional QTTF distribution for a chip was also computed by Monte-Carlo simulation with an accept-and-reject strategy. In other words, the simulation only accepted the sample vectors with similar entries on the measurement sites, the tolerance of which was set to 0.01nm in our implementation. This is equivalent to exploring the parameter space of the conditional random vector $\mathbf{x_u}|\mathbf{s_0}$. The results are shown in Figure 4.9 for a chip with 0.5 million devices and 25 measurements. It is clear that the histogram of 1000 sample vectors matches well with the predicted lognormal pdf using the techniques in Section 4.2 and 4.3. The difference between the mean of the histogram and lognormal pdf is 0.038 years. The 99.9% confidence lower bound of QTTF is 3.203 year demonstrating the tightness of the QTTF distribution.

We also explored how the predicted QTTF distribution changes when we increase the number of measurements. Figure 4.10 clearly shows the reduction in variance as the number of measurements grows. It is interesting to note that even one or two measurements provide sufficient information to tighten the distribution whereas 100 measurements help reduce the standard deviation of the distribution to only 0.16 year. The difference between the actual QTTF and the mean of the predicted QTTF distribution (using 100 measurements) is only 0.03 year (0.8% estimation error), while the conventional guard-band is 2.07 year with almost 50% estimation error.

Moreover, we studied the convergence of the mean and 99.9% confidence lower bound ($\mu-3\sigma$) of the predicted QTTF distribution to the exact values, as shown in the error-bar plots of Figure 4.11 for two chips, one with thicker oxides and another with thinner oxides. For each particular measurement number, we picked up 10 different configurations (placement) of measurement sites and then computed the 10 set mean/ 99.9% confidence lower bound of QTTF distribution to achieve the error bar. With an increasing number of measurements, both the estimated values and their

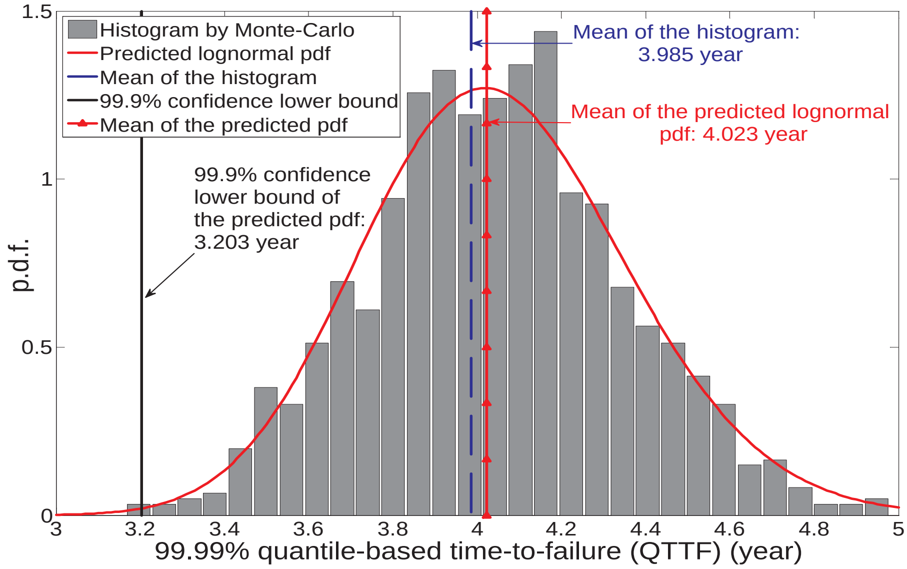


Figure 4.9: Accuracy comparison of the quantile-based time-to-failure (QTTF) histogram generated by Monte-Carlo simulation and the predicted QTTF pdf using the proposed method. The reliability target R_t is set to 99.99% (100 failures per million).

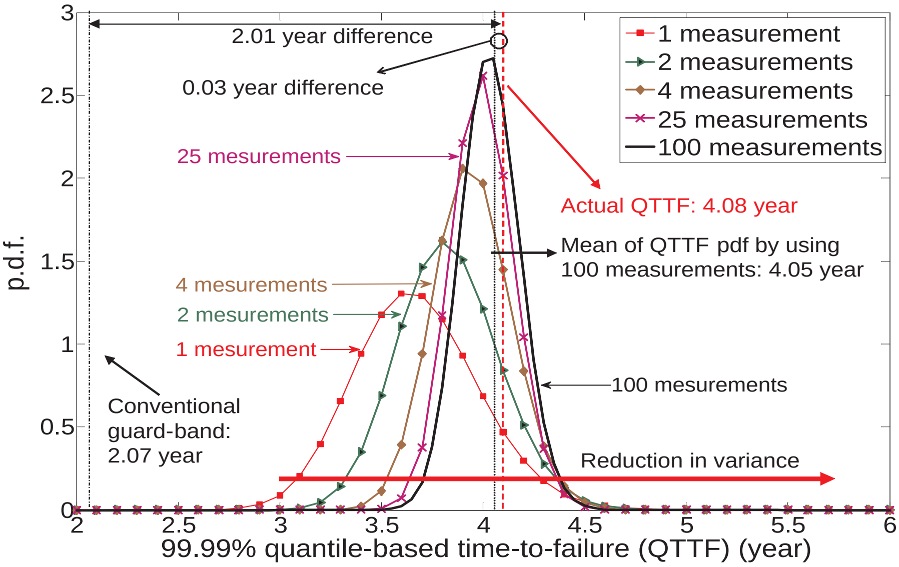
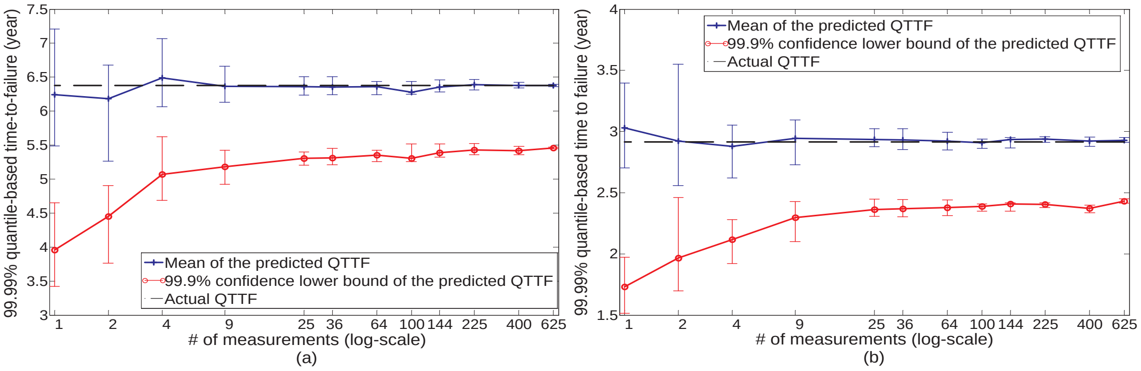


Figure 4.10: Reduction in the variance of 99.99% quantile-based time-to-failure (QTTF) distribution for a particular chip with increased measurements (1, 2, 4, 25 and 100 measurements).

variance converge quickly.

4.4.2 Reliability Management and Performance Optimization

Finally, we applied the proposed post-fabrication measurement-driven methodology to tune the supply voltage of 10000 chips of a 0.5M-device design to ensure reliability while maximizing perfor-



(a) (b)

Figure 4.11: Convergence of the mean and 99.9% confidence lower bound of the predicted quantile-based time-to-failure (QTTF) distribution with increased measurements: (a) a chip with thicker oxides (b) a chip with thinner oxides.

mance. The lifetime target was set to 4 years and the supply voltage tuning range is 0.8V-1.3V.

Figure 4.12 displays the tuning results using a conventional guard-band, the statistical analysis in Chapter III (denoted as 0 measurement in the figure) and the proposed methodology using different number of measurements. The guard-band that assumes minimum oxide thickness across the chip, achieved a single supply voltage for all the chips (0.858V) and was employed as the baseline for comparison. The other two methods used 99.9% confidence lower bound of the predicted QTTF distribution as the evaluation of chip's lifetime. Since the method in Chapter III uses a more accurate model of the oxide variation compared to the baseline approach, it assigns the ensemble of chip a slightly higher supply voltage of 0.875V. However, since it is unaware of the unique condition of each particular chip, it remains overly pessimistic and results in a merely 3% performance improvement. On the other hand, with only 25 measurements, the proposed methodology can obtain a well-tightened QTTF distributions and a more precisely optimize voltage for each chip, achieving 15% performance improvement on average and 26% improvement at maximum. Moreover, although the method in Chapter III predicts chip lifetime with 99.9% confidence lower bound, still 12 out of 10000 chips fail to meet the lifetime target after tuning, which is beyond the confidence interval. Meanwhile, since the proposed methodology provides more accurate prediction it quickly reduces the number of failures to 0 out of 10000 with increased measurements.

Figure 4.13 presents the distributions of optimized supply voltage and the resulting performance improvement using different numbers of measurements in tuning. Both the plots show a shift to the right with increased measurements, indicating the capability of the proposed method to choose a more reasonable supply voltage when the number of measurements is increased.

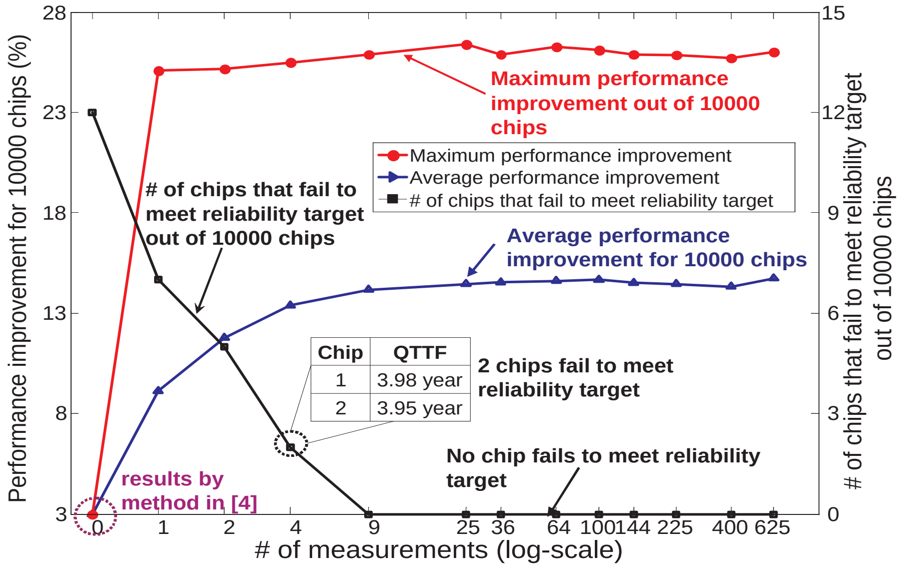
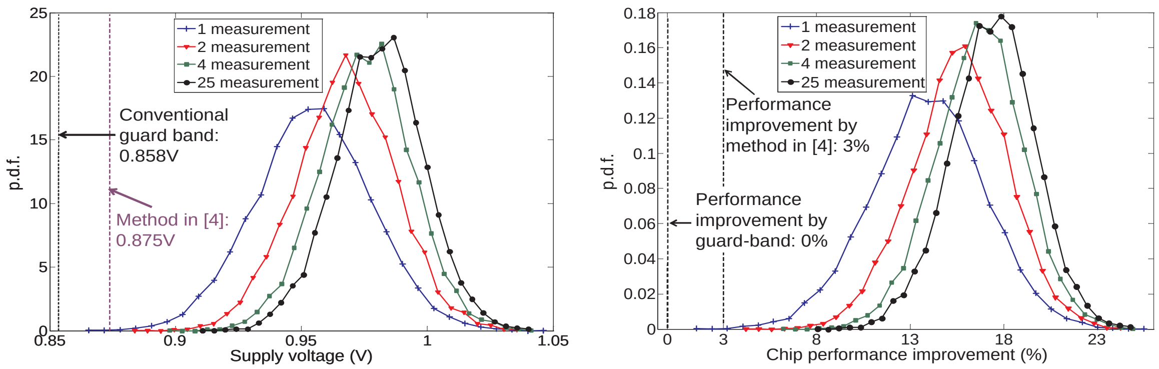


Figure 4.12: Reliability management results with increased measurements for 10000 chips of a 0.5M-device design: performance improvement and number of tuned chips that fail to meet target after optimization. Conventional guard-band is employed as baseline for comparison. The results for 0 measurement in the figure are from the method in Chapter III. Others are from the proposed method.



(a)

(b)

Figure 4.13: Distributions of (a) optimized supply voltages and (b) performance improvement with different numbers of measurements.

The scalability of the proposed methodology is examined in both its dependence on design complexity/size and run time. We first applied the approach to an alpha-processor-like design, with 15 functional blocks and 0.84M devices in total. Due to the functional block difference, the grids for spatial correlation model have non-uniform densities, i, e, each grid has different number of devices. We measured 25 devices per chip and tuned 10000 chips resulting in a performance improvement of 24.9% at maximum and 17.3% on average compared with conventional guard-band, as shown in Figure 4.14. We then applied the proposed method to tune 10000 chips of seven different designs

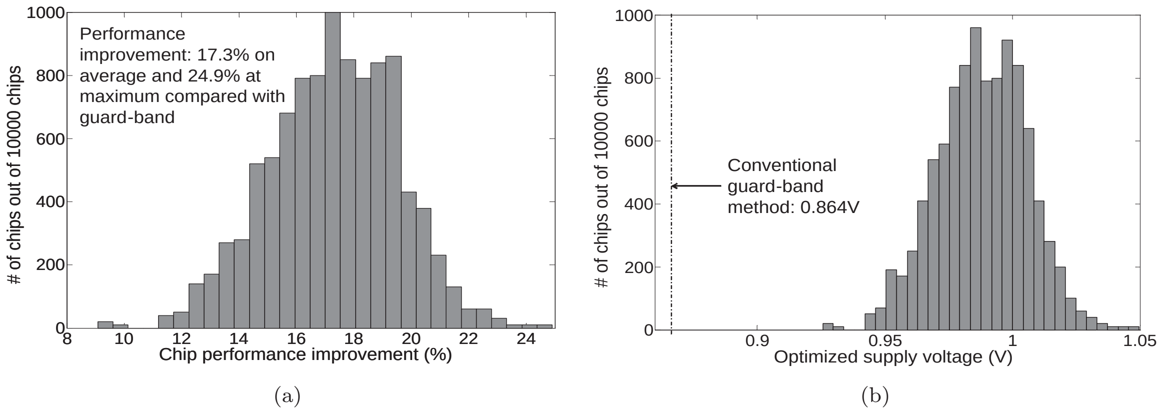


Figure 4.14: (a) Performance improvement histogram and (b) optimized supply voltage histogram of 10000 chips for an alpha-processor-like design with 0.84M devices and 25 measurements.

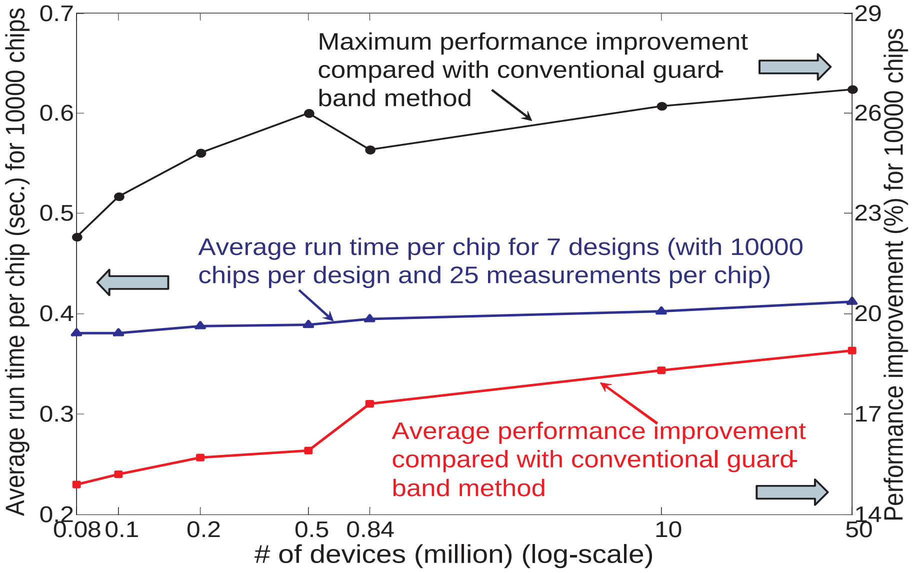


Figure 4.15: Average run time per chip and average performance improvement for seven different-sized designs (10000 chips for each design and 25 measurements per chip).

(varying in size from 80K to 50M devices) with 25 measurements for each and recorded performance improvement and average run time per chip. Figure 4.15 shows a flat curve of runtime of around 0.38 second, and a slightly growing trend of average performance improvement from 15% to 19% and maximum improvement from 22% to 27%. As stated earlier, both PCA and matrix inverse are performed once for one design with fixed measurement sites, whereas the analysis and optimization are mostly analytically achievable. Thus, the methodology runtime only relies on the number of grids for spatial correlation model instead of circuit size as validated in the figure, which is an appealing feature for modern processors with increasingly larger designs.

4.5 Summary

This chapter presents a post-fabrication measurement-driven OBD reliability prediction and management methodology. The methodology uses limited measurements to estimate the oxides condition of a chip. The estimation is then incorporated into a statistical model to predict a more accurate chip lifetime distribution, which is fed to an optimization flow to trade off reliability margin and system performance. Experimental results show that even for a design with up to 50 million devices, the methodology can achieve 19% performance improvement on average and 27% at maximum compared with conventional guard-band while average run time is only 0.4 second.

CHAPTER V

Variation-Aware Gate Sizing and Clustering for Post-Silicon Optimized Circuits

Susceptibility to system and random process variations increases along with the shrinking size of VLSI technology features in nano-regime chip designs [2, 30, 47, 79]. This process variation is of ever increasing concern in design performance estimation due to the demand for low power as well as the need for operating frequency growth [2, 47]. Because leakage and delay have strong dependence on the process parameters, the classical deterministic design method turns to be probabilistic. To address this issues, a handful of statistical optimization approaches that use pre-silicon techniques have been proposed [21, 25, 26]. These techniques are aimed at optimizing yield and therefore lead to a statistically optimal solution. The optimization decisions are therefore statistically effective for the manufactured dies; however, they are not necessarily ideal for an individual chip to meet the delay/power specifications [27, 28]. Unlike the pre-silicon approaches, post-silicon optimization techniques, such as adaptive body biasing (ABB), adaptive supply voltage (ASV) and tunable clock buffer (PST buffer), are carried out after the fabrication [27, 28, 29, 32]. With these adaptive tuning techniques, the specific variations (both die-to-die and intra-die) that occur on a particular die can be mitigated to achieve the design target [27, 28, 29].

One difficulty with the post-silicon tuning approaches is that adjusting each transistor/gate individually leads to excessive overhead in terms of routing of tuning control signals as well as the circuitry for tuning voltage generation and selection. To reduce overhead to feasible levels, several studies have investigated how to effectively cluster gates together before fabrication such that gates are positioned into a small set of clusters, each controlled by one post-silicon tuning parameter. For example, [27] presents a statistical method for clustering in which gates with statistically similar body bias voltages are placed together. However, this optimization is performed entirely as a post-processing step, and is divorced from the circuit optimizations such as gate sizing and gate restructuring and hence is sub-optimal. Other works have coordinated the pre- and post-silicon optimization at design time [28, 29], noting that pre-silicon optimization and post-silicon adaptability target the same design objective and parameter variability. The approach in [28] combines the traditional circuit optimization (gate sizing) and post silicon tuning (using ABB) to maximize design yield. The underlying formulation is a second order conic program (SOCP), which requires a somewhat simplified linear delay model. The approach in [29] describes a two-stage stochastic optimization for gate sizing in the presence of PST clock tree buffers. This approach uses a cutting-plane algorithm and is therefore limited in circuit size. A further limitation of these latter two methods is that neither addresses the issue of clustering, which is a key step necessary for brining the overhead of post-silicon tuning to feasible levels and complicates the optimization substantially.

In order to reduce the overhead while maintaining tunability to achieve a particular timing target, it is highly desirable to explore the interaction between gate sizing and clustering. In a cluster, all the gates are assigned the same body voltage and the final tuning voltage is determined by the most critical gate in the cluster. The gates on the critical path are modulated through the body biasing so that they are fast enough to meet the specified timing constraints. This may bind the other non-critical gates with an unnecessarily high body voltage. Assume that the cluster has two gates A and B whose body bias voltages have the probability distributions shown in Figure 5.1. Gate A has a higher probability of being assigned a larger body voltage, *i.e.* A is a critical gate for this cluster. The same body voltage is assigned to the non-critical gate B, resulting in an unnecessarily high leakage in gate B. The increase in body voltage may double or even triple the leakage of the cluster. Hence, it is crucial for all of the gates in a cluster to bear a high probability of having an identical criticality and therefore maximize the probability that all the gates require the same body bias voltage. This need for similarity creates dependence between the clustering and traditional circuit optimizations, such as gate sizing, and significantly complicates the optimization of the circuit. Furthermore, because this joint optimization must be considered from a statistical perspective, a typical deterministic optimization engine cannot be applied to the problem.

In this chapter, we propose a new statistical formulation and optimization technique that simultaneously considers both clustering and gate sizing. Although we focus on ABB as our post-silicon technique, our approach can be extended to other methods as well. The approach builds on our previous clustering strategy in [27] where gates are clustered according to the distribution of optimal body voltage. For the clustered gates, we coordinate the gate sizing and body voltage distribution with an alignment strategy. A linear dependence between the mean of body bias voltage distribution

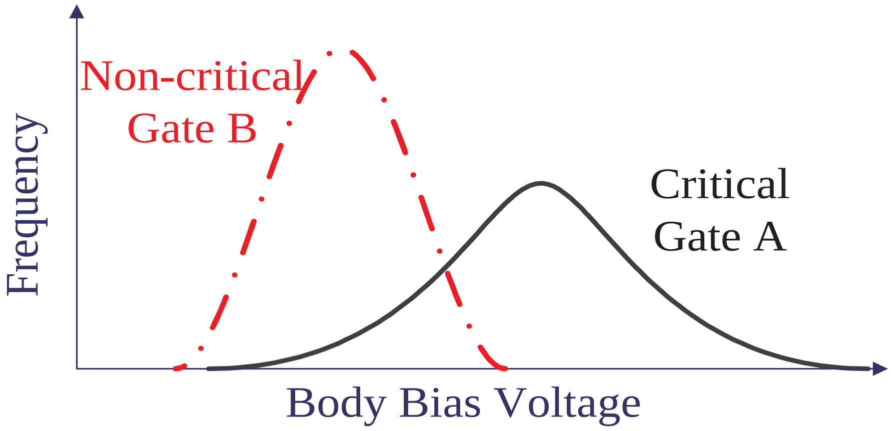


Figure 5.1: Body bias voltage distribution for critical and non-critical gates

and gate size is proven, which enables efficient gate sizing to match the body bias voltage distribution for the gates in a cluster. Rather than rely on sizing alone for better leakage saving, some gates may be further re-clustered. Finally, the circuit is re-balanced by snapping the gate sizes to the cells in the library. Experimental results show that our methodology can achieve 26.2% leakage savings for a circuit with up to 14503 gates at almost the same run time as the method in [27].

5.1 Simultaneous Gate Sizing and Clustering for Post-Silicon Tuning

5.1.1 Overview of the Methodology

This chapter proposes a two-stage optimization flow to determine the specific sizes for gates as well as the clusters to which gates are assigned. Based on the optimization flow, we can perform post-silicon ABB tuning for clusters to find the best tuning configuration for each die within the required power and delay envelope. The overall flow of the methodology is described as follows:

- **STAGE 1**: Generate multiple die samples following the expected L_{eff} distribution for a given circuit in a Monte Carlo (MC) fashion and then solve each scenario optimally with quadratic programming (QP) to obtain the optimal body bias probability distribution for each gate. After that, the gates with statistically similar body bias voltages are grouped together to obtain the initial clustering.
  - *Sub-Step 1*: Perform deterministic gate sizing on the circuit using any suitable gate sizing algorithm, such as the steepest decent algorithm used in this paper.

  - *Sub-Step 2*: Similar as Chapter II, with a fixed gate size, (2.1)-(2.6) turn to be a stochastic quadratic programming problem subject to the channel length variation, where the objective is quadratic in terms of $v_{b,i}$ and the constraint is linear with the variable AT_i. Monte-Carlo simulation is performed to obtain the die samples [27]. For each sample, the stochastic QP formulated by (2.1)-(2.6) becomes deterministic, which can be efficiently solved with CPLEX to determine the optimal body bias voltage for each gate [49].
  - *Sub-Step 3*: After the MC simulation, the frequency histograms (or the body bias profile as in Chapter II) of the optimal body voltage can be easily computed. We then perform the initial clustering to group the gates with statistically similar body bias voltage distribution [27].

- **STAGE 2**: In this stage, we use an efficient simultaneous gate sizing and clustering technique to achieve the configuration for better leakage saving. We investigate the linear dependence of the mean of body bias probability distribution and the gate size. The dependence allows us to efficiently solve the optimization problem. Based on that, gates are sized up or down and then adjusted to a new cluster if necessary. After this adjustment the optimized continuous solution is snapped to the standard library cells with a fast heuristic method.
  - *Sub-Step 1*: Size the gates in each cluster to align the body bias voltage distribution with a fast linear equation solve. Hence, gates in the cluster will have an aligned body bias probability distribution. Then, estimate the leakage for different scenarios to re-cluster the gate if necessary.
  - *Sub-Step 2*: Re-balance the circuit to remove the physical bounds violations (gate sizes outside the min - max available size in the library) and then snap the gate to the corresponding library cells while still meeting the timing constraints.

5.1.2 Body Bias Voltage Alignment

After STAGE 1 discussed in Section 5.1.1, we can obtain the body bias distribution for each gate, which is a probability distribution describing the tuning action (*i.e.* frequency of occurrence of body bias voltages) of a gate over the sample space of dies. The gates are initially clustered according to the correlation between their tuning actions [27]. For each cluster, all the gates are assigned the same body bias voltage as is determined by some critical gates as shown in Figure 5.1. In other words, the body bias voltage of the cluster is dominated by a small portion of *critical gates*, even though a large portion of *non-critical gates* actually contribute more to the cluster leakage. Note the fact that with a body bias voltage raise from 0.0V to 0.5V, the leakage almost triples, while the gate obtains about 15% speed improvement as illustrated in Figure 1.3. Thus, it is crucial to align the body bias distribution of those critical gates to the non-critical ones so that the leakage will not be determined by a small portion of gates, as shown in Figure 5.2.

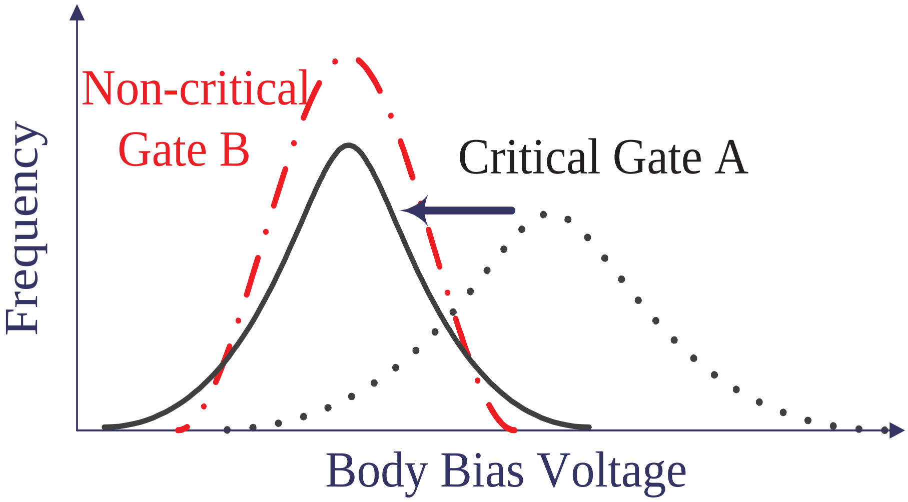


Figure 5.2: Body bias alignment for a critical gate

Sizing the critical gates is an effective way to change the body bias distribution. Generally speaking, sizing up the gate may shift the distribution to the left for a smaller body bias, whereas sizing down may increase the probability of being assigned a larger body bias compensating for the increased delay. However, aligning the distribution requires a more accurate estimation, particularly when the loaded gates are also sized up or down. Assume the optimal body bias voltage for gate i is the variable $v_{b,i}$, and the objective aligned body bias distribution is another variable $v_{i,obj}$. The motivation is to make gate i's delay before alignment almost the same as the delay after alignment. Hence, we can obtain the following equality using an Elmore delay model to guide the optimization:

$$D_i^{ABB} = q_1 \frac{\sum_k w_{i,k} l_{i,k} + C_{0,i}}{w_i/l_i}(1 - \delta s_i(v_{b,i})) \approx q_1 \frac{\sum_k w^o_{i,k} l_{i,k} + C_{0,i}}{w^o_i/l_i}(1 - \delta s_i(v_{i,obj})) \tag{5.1}$$

where w_i and w_i^o are the widths of gate i before/after sizing, $w_{i,k}$ and $w^o_{i,k}$ are the widths of the loaded gate k before/after sizing, l_i and $l_{i,k}$ are the channel lengths for gate i and its loads, respectively, constant $C_{0,i}$ is the wire capacitance or load capacitance, q_1 is a constant coefficient for the model.

Equation (3.12) can be further simplified to the following with the substitution $w^o_{i,k} l_{i,k} = n_k w_{i,k} l_{i,k}$:

$$\frac{\sum\limits_k n_k \frac{w_{i,k} l_{i,k}}{\sum\limits_k w_{i,k} l_{i,k} + C_{0,i}} + \frac{C_{0,i}}{\sum\limits_k w_{i,k} l_{i,k} + C_{0,i}}}{n_i} (1 - \delta s_i(v_{i,obj})) \approx 1 - \delta s_i(v_{b,i}) \tag{5.2}$$

where n_i is the sizing factor for gate i to be determined. In (5.2), body voltage $v_{b,i}$ is a random variable related with the channel length variation. We explore the correlation between these two random variables as shown in Figure 5.3. The curve demonstrates the correlation statistics between the channel length and corresponding body bias distribution for the ISCAS circuit C7552. The x-axis denotes the correlations whereas the y-axis denotes the percentage of gates beyond the specific correlation value. It should be noted that more than 70% gates have the correlation larger than 0.7. Only around 10% of gates have low-correlation. We can therefore reasonably assume that $v_{b,i}$ contains similar variation information of the system as channel length, so that the high order components in (5.2) can be discarded. Experimental results in Section IV show that such an approximation still maintains a desirable accuracy while simplifying the complexity of the problem. Using the nominal value of channel length, we may obtain the approximated relationship of the body bias distribution before and after alignment. With the notations $c_k = w_{i,k} l_{i,k} / (\sum\limits_k w_{i,k} l_{i,k} + C_{0,i})$, $c_0 = C_{0,i} / (\sum\limits_k w_{i,k} l_{i,k} + C_{0,i})$ and n_0=1, (5.2) can be written as:

$$\sum_k c_k n_k / n_i \times (1 - \delta s_i(v_{i,obj})) = 1 - \delta s_i(v_{b,i}) \tag{5.3}$$

As a result, the mean value of the objective body bias distribution is:

$$E(v_{i,obj}) = (1 - \frac{n_i}{\sum\limits_k c_k n_k}) \frac{1 - d_{0,i}}{d_{1,i}} + \frac{n_i}{\sum\limits_k c_k n_k} E(v_{b,i}) \tag{5.4}$$

and the deviation is:

$$D(v_{i,obj}) = (\frac{n_i}{\sum\limits_k c_k n_k})^2 D(v_{b,i}) \tag{5.5}$$

From (5.3)-(5.5), we see that both the shape and position of body bias voltage distribution can be changed by adjusting the sizing factors. When the sizing factors of the loaded gates are fixed, (5.4) demonstrates the linear dependence of the mean of body bias voltage distribution *w.r.t.* the sizing factor n_i, *i.e.* $E(v_{i,obj})$ varies linearly according to n_i.

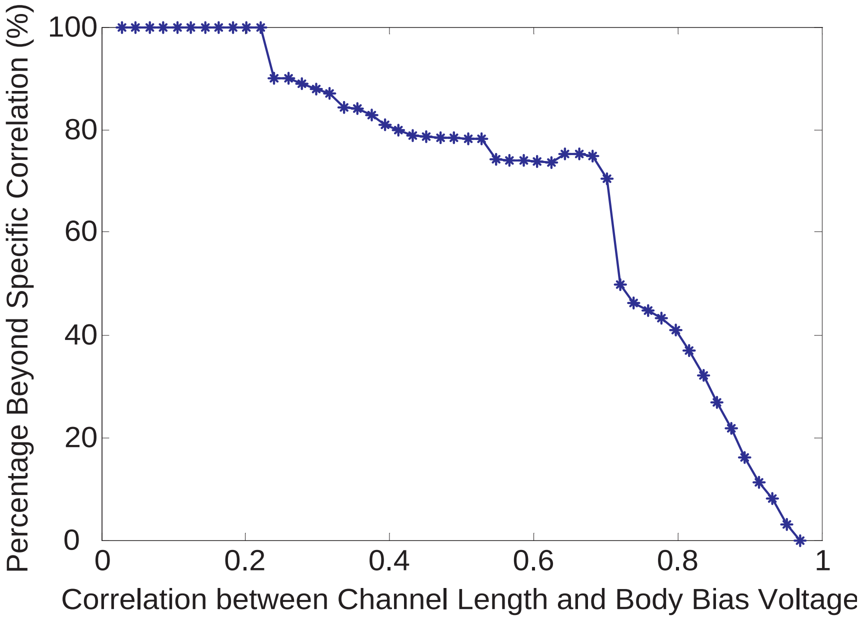


Figure 5.3: Percentage of gates beyond specific correlation for C7552

5.1.3 Simultaneous Gate Sizing and Re-clustering

Equation (5.4) shows the interaction of the gate sizing and body bias probability distribution. Assuming $V_{i,exp}$ is the expected mean of body bias voltage to achieve for any gate i, we can write a series of equations based on (5.4):

$$V_{i,exp} = (1 - \frac{n_i}{\sum\limits_k c_k n_k})\frac{1-d_{0,i}}{d_{1,i}} + \frac{n_i}{\sum\limits_k c_k n_k} E(v_{b,i}),\ i = 1...n \tag{5.6}$$

where $E(v_{b,i})$ is obtained after STAGE 1) in Section III-A. The system of (5.6) can be further represented as a linear system with n variables:

$$V_{i,exp} \sum_k c_k n_k = (\sum_k c_k n_k - n_i)\frac{1-d_{0,i}}{d_{1,i}} + E(v_{b,i}) n_i,\ i = 1...n \tag{5.7}$$

For this linear system we have the following lemma:

Lemma: The linear system defined by (5.7) has a unique solution.

Proof. For a given constant $V_{i,exp}$, each equation in (5.7) is exactly in the linear form $\sum_{i=1}^{n} t_i n_i = t_0$ where t_i's are constants. With a reversed topological sorting, we can levelize the circuit to m levels. Each level has n_j gates that $\sum_{j=1}^{m} n_j = n$. Level 1 consists of the gates connected to the primary outputs. Thus, the level 1 gates are loaded only with a constant capacitance and can be solved

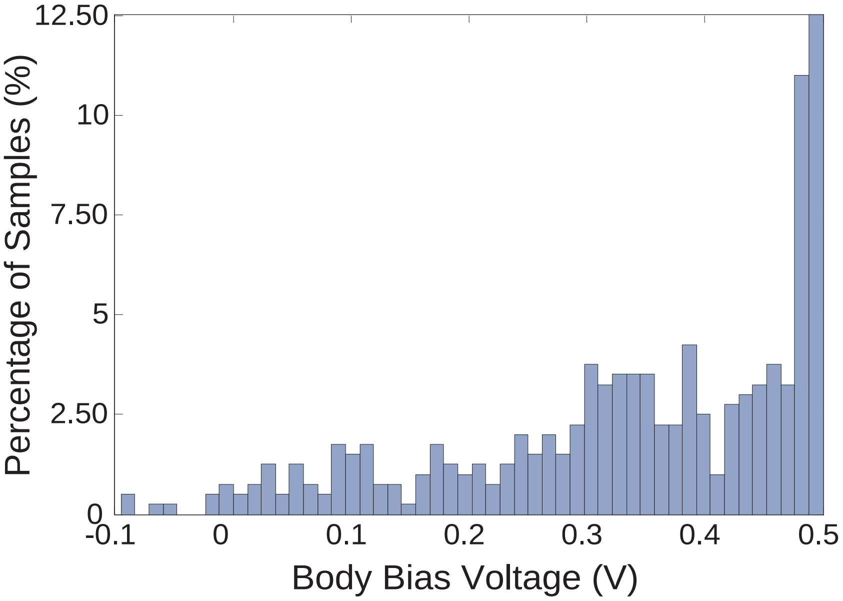


Figure 5.4: Histogram of the optimal body bias for the cluster with the largest cluster body bias voltage

directly. At level 2, any gate is only loaded with the wire capacitance and level 1 gates. After the sizing factors for the gates at level 1 are obtained, the linear equations at level 2 can also be solved. This procedure will be performed iteratively until level m. Consequently, the whole system defined by (5.7) can be solved and have a unique solution. □

It is straightforward to use the mean of *cluster body bias voltage* $E(v_{b,cluster})$ ($i.e.$ the mean of all the gates' body bias voltages in the cluster) as an evaluation of $V_{i,exp}$. However, since the normalized leakage change is quadratic, it is unnecessary to size too much for gates in the cluster with a very low body bias voltage. Thus, we should have different criteria for different clusters. The cluster with the largest cluster body bias includes the most critical gates. The critical gates in this cluster are always assigned the maximal body bias voltage as shown in Figure 5.4, which causes the non-critical gates in the cluster to consume intensive leakage. It is necessary to align $E(v_{b,i})$ for a critical gate to a small voltage $V_{i,exp} = E(v_{b,cluster}) - 2D(V_{b,i})$ so that the large tail of the distribution will not restrict the body bias voltage selection of the cluster. For the cluster with the smallest body bias voltage, the leakage change is limited. Hence we can set a larger value $V_{i,exp} = E(v_{b,cluster}) + D(V_{b,i})$ for the critical gate to avoid the unnecessary size-up. For all the other clusters, the expected value is set as $V_{i,exp} = E(v_{b,cluster})$.

Meanwhile some gates may be over-sized resulting in an intense increase in the gate leakage. In

```
Procedure: Simultaneous Gate Sizing and Re-clustering
Input: initially clustered circuit, body bias distribution $v_{b,i}$ for each gate
Output: sized gates and new clusters
Sort the cluster in the decreasing order by the mean of cluster body bias voltage;
For each cluster $n$ do;
    For each critical gate $i$ do
        Evaluate leakage for gate $i$ at three scenarios:
            Maintain the original size with $v_{b,i} = E(v_{b,i})$,
            Set $v_{b,i}$ as $E(v_{b,cluster})$ for cluster $n$ and size up the gate,
            Move the gate to the previous cluster $n$-1, set $v_{b,i}$ as $E(v_{b,cluster})$
            for cluster $n$-1 and size down the gate;
        Selects the scenario with the least leakage;
        Compute $V_{i,exp}$ for gate $i$;
    end for
end for
Solve the linear system to obtain the sized gates;
```

Figure 5.5: Algorithm for simultaneous gate sizing and re-clustering

```
Procedure: Circuit Re-balancing
Input: optimized circuit, the violated gate $i$, reduction gradient $RATIO$
Output: re-balanced circuit
Size gate $i$ to its upper bound;
Evaluate the new delay $D_{new,i}$ for gate $i$
Obtain the reduction ratio $R = D_{old,i}/D_{new,i}$;
If $R \geq RATIO$;
    Continue;
else
    Size down all the gates loaded by gate $i$ to make $R = RATIO$;
    For each gate $j$ loaded by gate $i$ do
        Call Circuit Re-balancing for gate $j$;
        Update the delay of the circuit;
    end for
end if
Update the delay of the circuit;
```

Figure 5.6: Algorithm for circuit re-balancing

this case, instead of sizing up, we can move the gate to the cluster with a larger body bias voltage. The once critical gate then becomes non-critical in the new cluster. Such re-clustering benefits both the original cluster and gate itself. Thus, we need to evaluate the leakage at different scenarios so as to determine whether the gate should be removed to a new cluster, sized up or just maintain its original size. The algorithm for the simultaneous gate sizing and re-clustering is shown in Figure 5.5.

5.1.4 Circuit Re-balancing and Snapping

The algorithm of simultaneous sizing and re-clustering is carried out without imposing physical bounds on sizes. Some gates may be sized beyond the available bounds of the cells in the library. In such a case, we need to re-distribute the over-sized area and re-balance the circuit to meet timing. Assume gate i is over-sized beyond the upper bound. The circuit re-balancing algorithm will apply a width reduction on gate i and the related gates in the cone of influence. The motivation is to make gate i itself see the greatest reduction whereas the farther gates see smaller changes. A reduction gradient $RATIO$ (=0.95 in our implementation) is introduced as a criterion. For each gate, the ratio of the original delay and updated delay after re-balancing should be restricted by $RATIO$. Consequently, the impact of reduction will gradually become negligible enabling the re-balanced circuit meeting the timing specification. Since the circuit re-balancing is merely a slight adjustment of the gate size, we can always use a nominal delay for evaluation, as body bias modulation on the same gate varies little. The algorithm for circuit re-balancing is shown in Figure 5.6. We have determined experimentally to limit the proposed strategy performed iteratively within ten levels of logic in the aforementioned cone of influence. In fact the iteration generally terminates within three or four levels.

After re-balancing, the optimization decision on each gate is snapped to the corresponding cells in the library. It can be mathematically formulated as a dynamic programming, which is computationally difficult. On the contrary, simply rounding the continuous size to the nearest discrete value may lead to timing violation. In the paper, we prefer a heuristic scheme for trade-off between run time and accuracy. For each gate, only cells with sizes close to the continuous solution will be investigated, so that the potential candidates of each gate are restricted to one or two cells. We then evaluate the possible delay combination for each gate and pick up the configuration with the delay closest to the optimized value. This procedure will be performed on the levelized circuit for all the gates leading to a snapped solution with good quality.

5.2 Experimental Results

The experiments are performed on the circuits with industrial 1.2V 90nm triple-well process. The delay target in this set of experiments is 10% faster than the original design. Body bias voltage is in the range of -0.5V to 0.5V for our analysis. The benchmarks include nine circuits from ISCAS85 and a VITERBI Decoder circuit with up to 14503 gates (*abbrev.* Vit1). We choose a recently-reported efficient post-silicon tuning method in [27] for comparison. The optimization engine for

QP is CPLEX [49]. In the experiments, the number of clusters is fixed to three. However, our methodology is a general technique without restriction on the number of clusters and can be applied to any case.

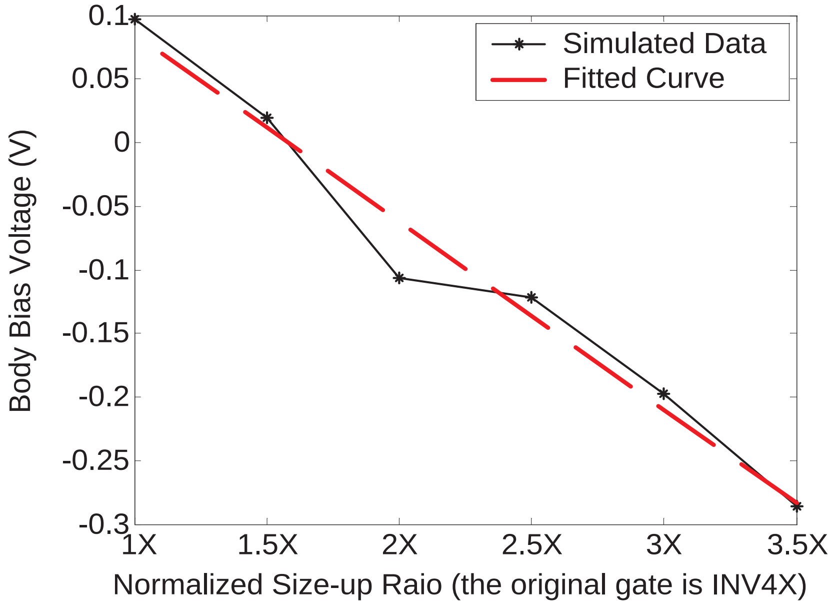


Figure 5.7: Body bias voltage v.s. normalized size-up ratio of a randomly selected gate from C432

5.2.1 Body Bias Voltage Alignment Exploration

In Section 5.1.2 we explore the general theory of the body bias voltage alignment via sizing. We use (5.4) to evaluate the mean of body bias after alignment to guide the further optimization. When the sizes of the loaded gates are fixed, (5.4) turns to be a linear equation of the size-up ratio, as shown in Figure 5.7. The y-axis is the mean of body bias voltage after sizing, whereas the x-axis is the normalized size-up ratio. Figure 5.7 clearly demonstrates the linear dependence of the mean of body bias distribution and sizing factor for a randomly selected gate. Similar results are also observed from the other circuits. Figure 5.8 shows the change of the body bias distribution after sizing. One can see that the distribution shifts to the left with a smaller mean.

5.2.2 Efficiency of the Proposed Methodology

This section shows the experimental results of the proposed methodology and related techniques discussed in Section 5.1. We use Monte-Carlo simulation to generate the samples of dies to evaluate the leakage/delay envelope of the optimized design. The assumed magnitude of channel length

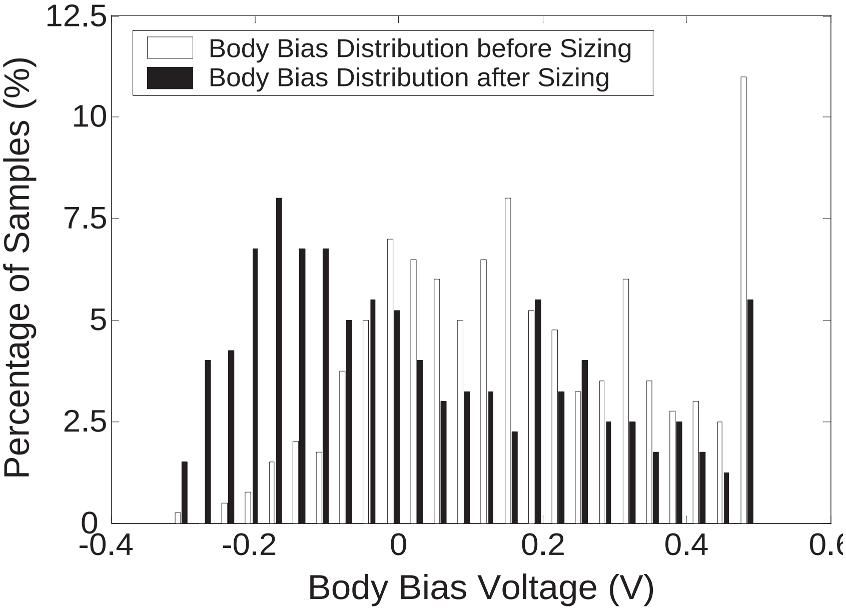


Figure 5.8: Body bias voltage distribution change before/after sizing

variability is $3\sigma/\mu$=15%. Table 5.1 and Table 5.2 show the leakage and delay on circuits in comparison with the clustering method from [27], respectively. Columns 3-7 in Table 5.1 record the mean, standard deviation and 95th percentile of leakage power. Columns 3-7 in Table 5.2 record the mean, standard deviation and 95th percentile of delay. The last columns of Table 5.1 and Table 5.2 report the improvement on the 95th leakage and delay by our methodology, respectively. It can be seen that our methodology significantly outperforms the clustering method in [27]. The 95th percentile leakage can obtain 25.5% saving in average whereas the circuits maintain almost the same or even a little faster speed.

Figure 5.9 shows the circuit leakage distributions generated by the proposed methodology and clustering method in [27] for C3540. Unlike the clustering method in [27], our methodology not only reduce the 95th percentile leakage (the tail of the leakage distribution) but also improves the mean value by 24% in average. These evidences indicate that the simultaneous sizing and clustering for post-silicon tuning is more efficient in exploring the optimization space than simply using post-processing technique as in [27].

The run time of the proposed methodology is compared in Table 5.3. It is noted that our method takes almost the same time as the method in [27]. The last column presents the run time for simultaneous sizing and re-clustering discussed in Section 5.1.3. It is very fast and almost linear *w.r.t.* the circuit size. To achieve a full-scale sampling, the Monte-Carlo analysis consumes more

Table 5.1: Leakage power comparison between the proposed methodology and the method from [27]

CKT.	#gates	Proposed Methodology			Method from [27]			Improv.(%)
		μ	σ	95%	μ	σ	95%	
C432	166	2.64	0.90	4.32	3.65	1.20	5.62	23.3%
C499	519	15.09	5.26	23.27	17.87	6.63	29.96	22.4%
C880	390	2.89	1.05	4.87	3.85	1.44	6.37	23.6%
C1908	432	7.39	2.45	11.43	8.43	3.11	14.26	20.0%
C2670	964	5.07	1.96	9.14	7.80	2.82	13.08	30.2%
C3540	962	9.50	3.30	15.00	13.64	5.15	22.70	33.9%
C5315	1750	13.49	4.67	21.46	18.00	6.96	31.00	30.8%
C6288	2502	72.84	26.13	117.80	84.96	34.29	150.08	21.5%
C7552	2102	14.89	5.75	26.09	19.89	7.61	33.62	22.6%
Vit1	14503	173.88	67.71	291.97	246.02	110.40	395.73	26.2%
Average Saving (%)								25.5%

Table 5.2: Delay comparison between the proposed methodology and the method from [27]

CKT.	#gates	Proposed Methodology			Method from [27]			Improv.(%)
		μ	σ	95%	μ	σ	95%	
C432	166	0.65	0.00	0.65	0.65	0.00	0.66	1.52%
C499	519	0.53	0.00	0.53	0.53	0.01	0.55	3.64%
C880	390	0.69	0.00	0.70	0.69	0.00	0.70	0%
C1908	432	0.99	0.01	1.00	0.99	0.01	1.00	0%
C2670	964	0.67	0.01	0.67	0.67	0.01	0.67	0%
C3540	962	1.05	0.01	1.06	1.05	0.01	1.07	0.93%
C5315	1750	0.95	0.01	0.96	0.95	0.01	0.97	1.03%
C6288	2502	2.90	0.01	2.91	2.92	0.07	3.11	6.43%
C7552	2102	1.14	0.00	1.15	1.14	0.02	1.18	2.54%
Vit1	14503	2.14	0.01	2.15	2.14	0.02	2.21	2.71%
Average Saving (%)								1.88%

Table 5.3: Run time comparison between the proposed methodology and the method from [27]

CKT.	Overall Run Time (sec.)/Normalized Ratio				Time for Sizing (sec.)
	Proposed Method		Method from [27]		
C432	43	1.02×	42	1×	0.03
C499	81	1.04×	78	1×	0.06
C880	60	1.03×	58	1×	0.05
C1908	67	1.02×	66	1×	0.05
C2670	107	1.02×	105	1×	0.08
C3540	129	1.02×	126	1×	0.08
C5315	260	1.02×	254	1×	0.16
C6288	453	1.02×	446	1×	0.28
C7552	302	1.02×	296	1×	0.21
Vit1	1633	1.02×	1608	1×	1.82

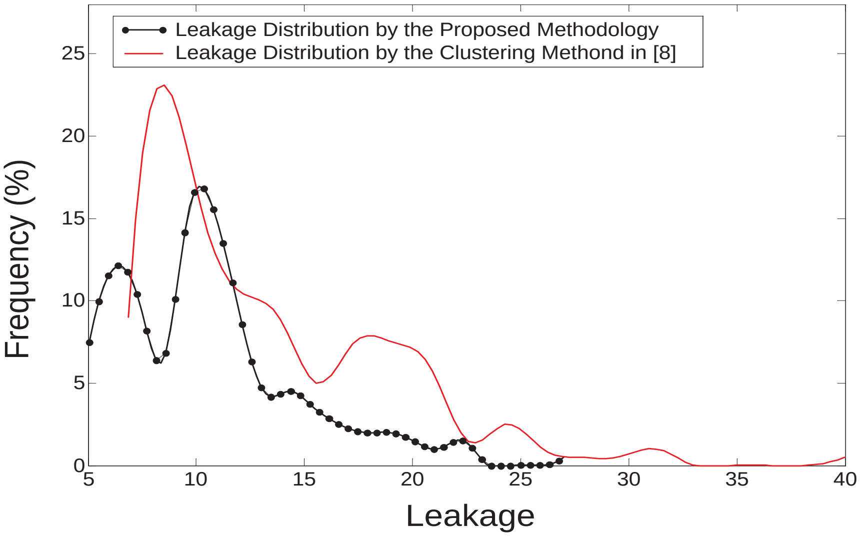


Figure 5.9: Leakage distribution comparison using the proposed methodology and the mthod from [27]

than 80% of the overall run time. In order to guarantee the convergence, the number of Monte-Carlo samples increases as the circuit size goes larger [27].

5.3 Summary

This chapter presents an efficient variation-aware methodology for simultaneous gate sizing and clustering for post-silicon tuning through ABB. It uses a practical table look-up delay model and can be applied to the circuits with tens of thousands of gates. The proposed body bias alignment technique helps to actively incorporate the post-silicon tuning procedure in the pre-silicon design cycle to fully explore the optimization space. Experimental results show 25.5% improvement of leakage in average for ten different sized benchmarks.

CHAPTER VI

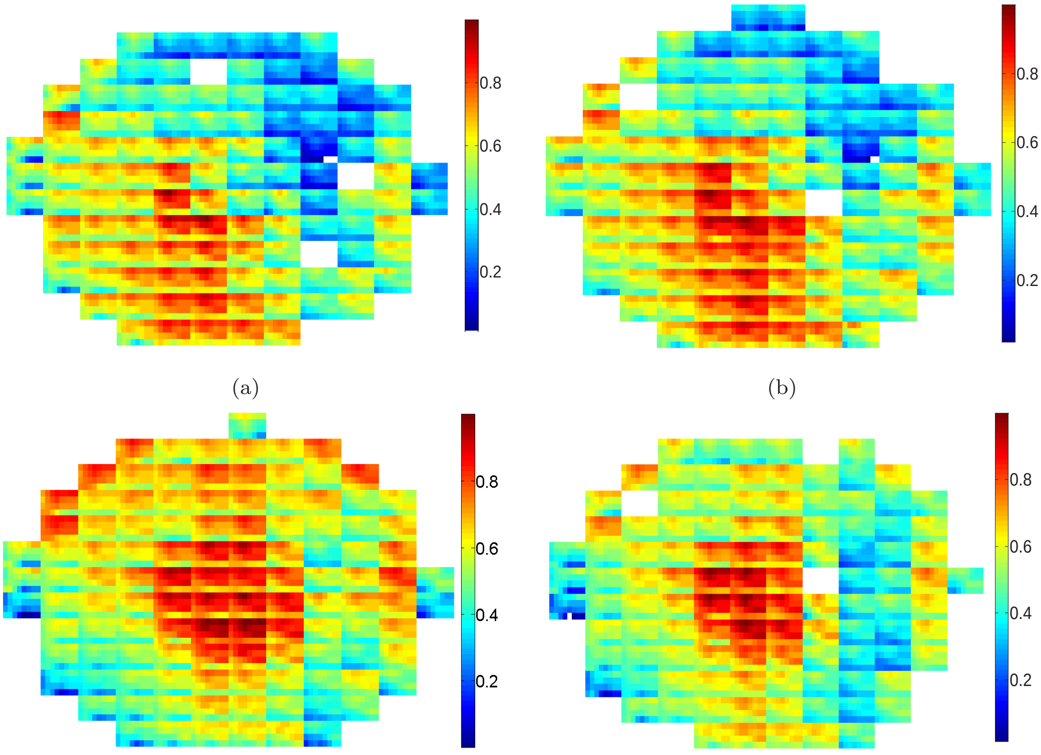
Active Learning Framework for Post-Silicon Variation Extraction and Test Cost Reduction

Susceptibility to process variation has increased with the scaling of CMOS into the nano-regime of VLSI designs [2]. The application of new resolution enhancement techniques complicates an already complex manufacturing process and makes it more difficult in maintaining process uniformity [2]. As a result, efficient and accurate process variation modeling becomes essential to ensure good yield.

Traditionally process variation modeling is targeted for design-time use and guides engineers in the optimization of their chips before silicon fabrication [26, 37]. Typically, such design-time process models rely on characterizing tens to even hundreds of test wafers [8, 11, 12, 73, 80]. The characterized model is then fed to either the statistical analysis tools to estimate design yield [47, 48] or statistical optimization engine that efficiently tunes thousands of devices to achieve a robust and high yield solution [81]. However, in recent years, due to the increasingly significant variability and the inability to measure every device on a die [82], process variation models are also critical after chips are fabricated for multiple post-silicon applications, such as:

- **Post-silicon tuning** which requires an accurate understanding of current process to appropriately adjust tuning parameters [28, 83].
- **Post-silicon timing characterization** where speed binning and critical path diagnosis require efficient process variation models to reduce test vector sets. [82, 84].
- **Post-silicon reliability analysis** where accurate models can tighten the process uncertainty to improve the chip lifetime prediction and enables specific supply voltage adjustment for the chip to obtain a better lifetime/performance trade-off [39].

Since there is limited research focused on extracting variation models for post-silicon use, most



(a)

(b)

(c)

(d)

Figure 6.1: Ring oscillator (RO) frequency measurements (scaled) of 4 wafers from 2 different lots in a 65nm process: (a) and (b) are wafers from lot 1; (c) and (d) are wafers from lot 2. The 2 lots have different global trends.

post-silicon techniques still rely on design-time variation models and do not take advantage of the availability of test structure measurements from individual wafers and dies. This leads to the following two drawbacks:

- Since a design-time process variation model must capture variations across all wafers and lots, it results in a significantly more loosely distributed or pessimistic variation model than could be obtained based on even limited measurements from one or all the wafers within a lot. Figure 6.1 shows 4 wafers from the same 65nm process but two different lots. The wafers of lot 1 in Figure 6.1(a) and (b) have similar wafer-level global trends, which are quite different from the ones of lot 2 in (c) and (d). In other words, if we can model the process variation of a wafer by using information from post-fabrication measurements on the same or another wafer, the uncertainty of the model may be significantly reduced, which helps mitigate the unnecessary pessimism in post-silicon applications.

- The extraction of a design-time model assumes that the process remains constant after the model has been generated using test wafers. However, in practice, the process recipe for test

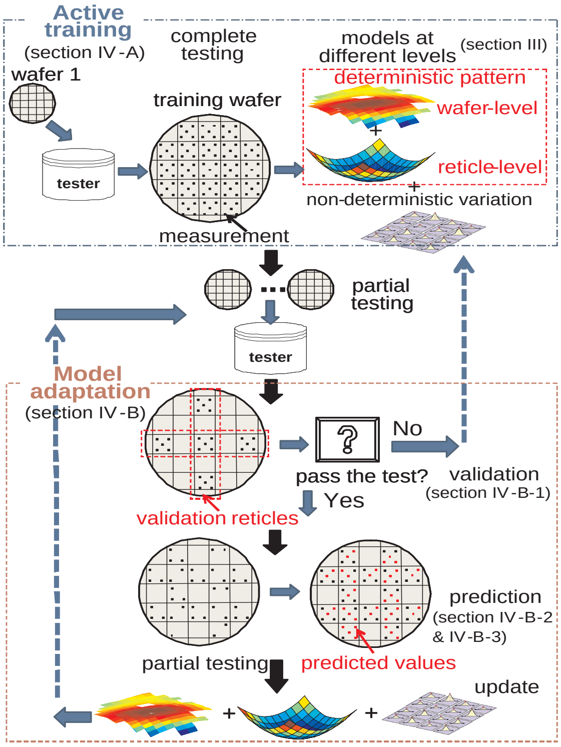


Figure 6.2: Flow chart of the proposed framework

wafers run at design time may not correlate well with the one for production chips [82]. In addition, process is continually optimized and the variation may change over time [13]. This is only captured by periodically running new test wafers, which is uncommon and expensive.

Instead, what is needed is a dynamic post-silicon variation model that automatically tracks and adapts to process changes using limited test measurements. Such a model is not only useful for post-silicon applications but could also be used for future designs in the same process.

However, constructing such a model is not trivial. It is common today to deploy hundreds or even thousands of test structures (*e.g.*, ring oscillators or resistor arrays) within a product chip or in the scribe line for process monitoring [13]. But the overhead to measure all structures for all dies across all wafers is clearly too high [13, 14]. It is also unclear how to reuse measured information from earlier wafers to facilitate the modeling on a different wafer or lot. To address these issues, we propose a new framework where we dynamically extract a variation model from measurements using wafers of product chips that are instrumented with small test structures. The extracted model accounts for both systematic and spatially-correlated patterns as well as random variations.

Prior works in [73] and [85] also attempt to reduce the number of measurements to monitor process but either require simulations on tens or hundreds of wafers to achieve converged results or limit their analysis only to the current wafer under test. In addition, they are fixed approaches that do not dynamically adapt to process change. The proposed active learning framework *allows the process model to evolve by reusing information from past wafers to validate and improve the model.* The flow chart in Figure 6.2 gives an overview of the proposed framework. In particular, initial measurements are conducted for model training with high measurement density which will then be reduced when the model fidelity increases. In this manner the number of measurements is gradually decreased over time. This allows the model to adapt and improve with the process changes while reducing the test cost to a minimum. The framework has the following key modeling contributions:

- **Hierarchical process variation modeling**. We develop a hierarchical variation model that incorporates wafer-to-wafer, across-wafer, reticle-to-reticle, across-reticle and independent variation components, accounting for systematic, spatially correlated and random variations. The variation is extracted on a reticle basis by noting the design-process interactions in lithography steps.

- **Active training**. Active training initializes the active learning models in the framework (Figure 6.2). This stage completely measures the initial wafer set to achieve deterministic spatial pattern models and quantify the uncertainty reduction ability of each test site.

- **Spatially correlated variation characterization**. We employ a sparse Bayesian learning method [87] to estimate the spatially correlated variation. Measurements from earlier wafers are used to identify the significance of bases to speed up the estimation.

- **Adaptive test configuration determination**. Test configuration is defined as a selective set of m out of n available test structures. Each measurement may reduce the model uncertainty to a different degree. This algorithm resolves how many and which measurements are conducted for a desirable accuracy.

- **Model validation and adjustment**. For an untested wafer, we apply several statistical tests on a selective set of reticles to justify if the existing model needs an incremental adjustment or a complete re-construction due to process drift.

The observation and experimental results in this chapter are based on two industrial processes with two different types of test structures and different reticle sizes. Process 1 is a 65nm technology process and has approximately 300 wafers with embedded ring oscillators (RO). The test structures within each reticle are coarse-grained but there are approximately 100 reticles within in each wafer. Process 2 is a 130nm technology process and has 5 wafers with electrical linewidth measurement (ELM). The test structures within each reticle are fine-grained with 23 reticles in each wafer. The generality and efficiency of the proposed framework is validated on both processes for the two different test structures, reticle sizes and measurement densities. Experimental results show that the proposed framework can achieve 83% and 78% variance reduction for two processes in comparison to design-time models. Meanwhile, the average estimation error is well maintained at ∼2-3% using merely ∼30% available test structures for two processes. Compared to a recently reported approach in [85], the framework can further reduce the test cost by more than 37% to achieve the same or better accuracy.

6.1 Statistical Preliminaries

In this section we briefly review several statistical techniques that will be used throughout the chapter. The details can be found in [86, 87, 88].

6.1.1 Robust Regression

The deterministic spatial pattern can be fitted from measurements to a given function. Least square fitting may be easily impacted by outliers or long-tailed error distribution. Robust regression is an alternative estimator to minimize fitting errors with the following term [86]:

$$\sum \rho(y_i - x_i^T \beta) \tag{6.1}$$

where $y_i - x_i^T \beta$ is the i_{th} estimation error and ρ is a weighting function to mitigate the impact of outliers. The details can be found in [86].

6.1.2 Statistical Tests

In the model validation step of Figure 6.2, we need to justify any model before applying it to an untested wafer. Since the non-deterministic variation for a device is typically modeled as a Gaussian random variable, we can use the following statistical tests in our framework [86]:

- *t-test* checks whether the mean of a normal distribution has a value specified in a null hypothesis. In the framework, it is used to justify if a predictor in the polynomial model is statistically significant.
- χ^2 *goodness-of-fit test* describes how well the given model fits a set of observations. This test is used to check the overall fitting goodness of an existing model, such as the fitting goodness of a wafer-level pattern model for the raw data from another wafer.

- *Kurtosis* is a measure of the peakedness of the distribution. We use this measure in our framework to justify if total variance distribution has a fundamental change and hence requires a re-fitting.

6.1.3 Sparse Bayesian Learning

This section gives a brief review of sparse Bayesian learning (SBL), the details of which can be found in [87, 88]. The basic idea of sparse Bayesian learning is to solve the following underdetermined system through a Bayesian inference [87]:

$$\mathbf{t} = \Phi\mathbf{w} + \epsilon \tag{6.2}$$

where Φ is a $m \times n$ matrix with each column called a basis and $m \leq n$, ϵ is typically considered to be zero mean Gaussian random variables with variance of σ^2. Given a $m \times 1$ vector $\mathbf{t}$ and a priori knowledge that $\mathbf{w}$ is *sparse*, SBL can find the most probable estimation of $\mathbf{w}$ and the corresponding co-variance matrix. The target vector $\mathbf{t}$ has a multi-variate Gaussian likelihood [87]:

$$p(\mathbf{t}|\mathbf{w}, \sigma^2) = (2\pi)^{-m/2}\sigma^{-m}e^{-\|t-\Phi\mathbf{w}\|^2/(2\sigma^2)} \tag{6.3}$$

and the prior over the parameters $\mathbf{w}$ is a zero mean Gaussian:

$$p(\mathbf{w}|\alpha) = (2\pi)^{-n/2}\prod_{i=1}^{n}\alpha_i^{1/2}e^{-\alpha_i w_i^2/2} \tag{6.4}$$

where $\alpha = (\alpha_1 ... \alpha_n)^T$ are n independent hyper-parameters, one per weight w_i, which represents the inverse of variance for $\mathbf{w}$ and pushes the solution to be sparse. The proof of using Gaussian priors to achieve sparsity is detailed in [87]. Given α, the posterior distribution is then a Gaussian and can be analytically written as:

$$p(\mathbf{w}|\mathbf{t}, \alpha, \sigma^2) = p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)/p(\mathbf{t}|\alpha, \sigma^2) = N(\mathbf{w}|\mu, \mathbf{\Sigma}) \tag{6.5}$$

with

$$\mu = \sigma^{-2}\Sigma\Phi^T\mathbf{t} \qquad \Sigma = (A + \sigma^{-2}\Phi^T\Phi)^{-1} \tag{6.6}$$

where A is $\text{diag}(\alpha_1, ...\alpha_n)$.

The posterior distribution can be achieved by Expectation Maximization method, which is equivalent to solving a type-II marginal likelihood maximization with respect to the hyper-parameters α [87, 88]:

$$\begin{aligned} \textbf{Max} \quad L(\alpha) &= logp(\mathbf{t}|\alpha, \sigma^2) = log\int p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)d\mathbf{w} \\ &= -1/2[Nlog2\pi + log|C| + \mathbf{t^T}C^{-1}\mathbf{t}] \end{aligned} \tag{6.7}$$

with $C = \sigma^2 + \Phi A^{-1}\Phi^T$. Once the most probable α_{MP} are found, they can be plugged into (6.6) to get μ_{MP} and the covariance matrix Σ_{MP}.

6.2 Hierarchical Modeling and Characterization of Process Variation

In this section we discuss the variation modeling at different spatial levels and the characterization of deterministic and non-deterministic parts of the model when complete or partial testing is conducted.

6.2.1 Hierarchical Modeling of Process Variation

By using the wafer-level variation model (1.2) in Chapter I, the proposed framework extracts variation on a reticle basis, which may include one or several dies. The reticle directly interacts with the lithography steps and exhibits certain regularity from reticle to reticle. Assume there are $m_1 \times n_1$ available test structures on a reticle with each representing one random variable as defined in (1.2). Then an $m_1 \times n_1$ matrix, denoted as A_i for the i_{th} reticle, can be constructed, which uniquely identifies the process variation of the reticle. According to (1.2), A_i can be written as:

$$A_i = z_0 + z_{iw} + z_{ir,i} + A_{w,i} + A_{r,i} + A_{ar,i} + R_i \tag{6.8}$$

where $A_{w,i}$ and $A_{r,i}$ are the matrices for wafer- and reticle-level systematic patterns observed in the i_{th} reticle, $A_{ar,i}$ and R_i are the matrices for the non-deterministic variations representing across-reticle spatially correlated and independent residual variations, respectively. It is noted that $A_{r,i}$ is a deterministic pattern at the reticle scale and hence the same from reticle to reticle. Thus we can rewrite $A_{r,i}$ as A_r.

6.2.2 Variation Characterization at Different Spatial Scales

6.2.2.1 Model Identification of Systematic Spatial Patterns

Deterministic spatial patterns are extracted by completely testing a set of reticles A_is. To split the mixed effect of two patterns, it is essential to first extract the pattern at lower scale, *i.e.* reticle-level pattern. In order to mitigate the impact of wafer-level pattern and other variation components in (1.2), we take the average of the matrices of A_is:

$$\overline{A} = \frac{1}{l}\sum_{i=1}^{l} A_i = z_0 + z_{iw} + \frac{1}{l}\sum_{i=1}^{l}(A_{w,i} + A_r + z_{ir,i} + A_{ar,i} + R_i) \tag{6.9}$$

where l is the number of reticles. By carefully analyzing the characteristics of each component, we have the following observations:

- Any device in the wafer observes the same z_0 and z_{iw}.
- Each entry in R_i is an independent Gaussian with zero mean and the same variance and hence can be cancelled after taking average.
- $z_{ir,i}$ is a constant for all the devices within one reticle. Thus, any entry in $\overline{A}$ observes the same constant $\frac{1}{l}\sum_{i=1}^{l} z_{ir,i}$.
- For an entry at a specific location of $A_{ar,i}$ across different reticles, it still follows a zero mean Gaussian. Thus, $\frac{1}{l}\sum_{i=1}^{l} A_{ar,i} \approx 0$.
- The wafer-level pattern typically has symmetric characteristics, *e.g.* slanted or parabolic surface [11]. By carefully choosing the reticles symmetrically placed on a wafer, the difference among entries in $\frac{1}{l}\sum_{i=1}^{l} A_{w,i}$ is limited and can be approximated to a constant.

Based on those observations, (6.9) can be simplified to:

$$\overline{A} \approx A_r + const \tag{6.10}$$

where $const \approx z_0 + z_{iw} + \frac{1}{l}\sum_{i=1}^{l}(A_{w,i} + z_{ir,i})$ is a constant. Thus, $\overline{A}$ is a compound effect of reticle-level pattern and the average of other variation components. We name it as the reticle-level common pattern.

Given $\overline{A}$ and its corresponding reticle-level locations ($\mathbf{X}_0$, $\mathbf{Y}_0$), a two-dimensional closed-form function can be fitted. The impact of outlier can be mitigated by using moving average (MA) and robust regression (section 6.1.1) [86]. Moving average helps smooth over rapidly varying features by moving a $m_0 \times n_0$ sub-block across $\overline{A}$ and replacing the original entry with the average of the sub-block. Figure 6.3 demonstrates a reticle-level pattern extraction example including: (a) the pattern $\overline{A}$ extracted from process 2 using 23 reticles in a wafer and (b) the change after moving average is conducted.

Before the regression is applied, we need to determine the model to fit. We here employ a backward elimination strategy and make the data find the model itself. For example, if the initial model is a 2_{nd} order full polynomial model, the fitted model by robust regression is then:

$$\mathbf{t}_{\overline{\mathbf{A}}} \sim [1, \mathbf{x}_0, \mathbf{y}_0, \mathbf{x_0 y_0}, \mathbf{x_0^2}, \mathbf{y_0^2}] \times \mathbf{p^T} \tag{6.11}$$

where $\mathbf{p} = [p_0, p_1 ... p_5]$ is the parameter vector to be fitted, $\mathbf{t}_{\overline{\mathbf{A}}}$, $\mathbf{x}_0$ and $\mathbf{y}_0$ are the vectorization results of matrices $\overline{A}$, $\mathbf{X}_0$ and $\mathbf{Y}_0$ respectively. A t-test is conducted on each parameter in $\mathbf{p}$ to compute the corresponding p-value. The most statistically insignificant predictor in $[1, \mathbf{x}_0, \mathbf{y}_0, \mathbf{x_0 y_0}, \mathbf{x_0^2}, \mathbf{y_0^2}]$ is then removed to simplify the model. This procedure is performed repeatedly till all the terms (predictors) in the polynomial model $f_r(\mathbf{X}_0, \mathbf{Y}_0)$ is significant. Figure 6.3(c) illustrates the acquired model using backward elimination and robust regression and (d) exhibits the relative error of fitting compared with the original pattern $\overline{A}$ in (a). It is observed approximately 2% average relative error, which indicates the necessity of modeling reticle-level patterns.

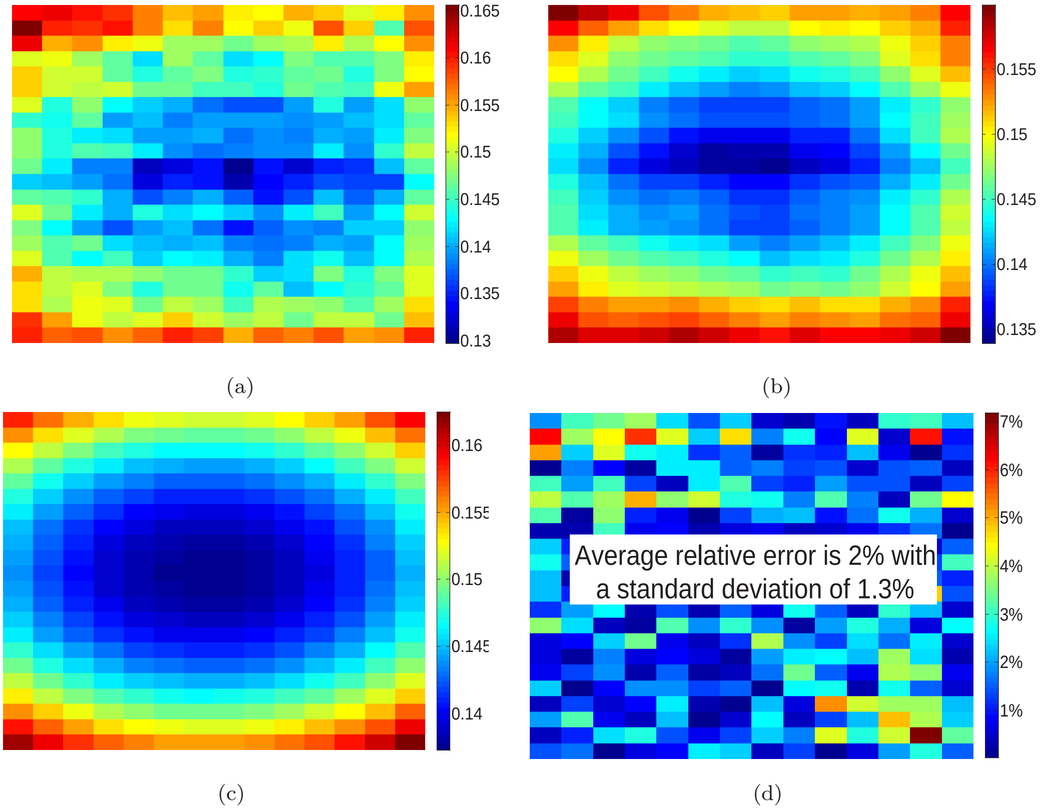
After reticle-level pattern is extracted and removed from the raw data, the wafer-level pattern $f_w(\mathbf{X}, \mathbf{Y})$ can be extracted in a similar way from the data at wafer-level, except that the coordinate matrices are now at wafer-level instead of reticle-level. Figure 6.4 demonstrates the reticle-level and wafer-level patterns for the wafers from two processes. The algorithm of deterministic pattern extraction is summarized in Figure 6.5.

6.2.2.2 Non-Deterministic Variation Estimation through SBL

After the deterministic patterns are removed from the model, the residual part is comprised of across-reticle spatially correlated and residual variations[10]:

$$A_{random} = A_i - \overline{A} - A_w(x, y) \approx A_{ar} + A_r \tag{6.12}$$

[10] In practice the model is unable to fit perfectly with the data. A_{ar} in the residual model of (6.12) is a compound effect of actual across-reticle spatially correlated variation and the residual from fitting or modeling approximation. The fitting or modeling approximation terms can be lumped into the mean of A_{ar}, and captured by SBL method afterwards. For simplicity we still use A_{ar} here to represent this compound effect.



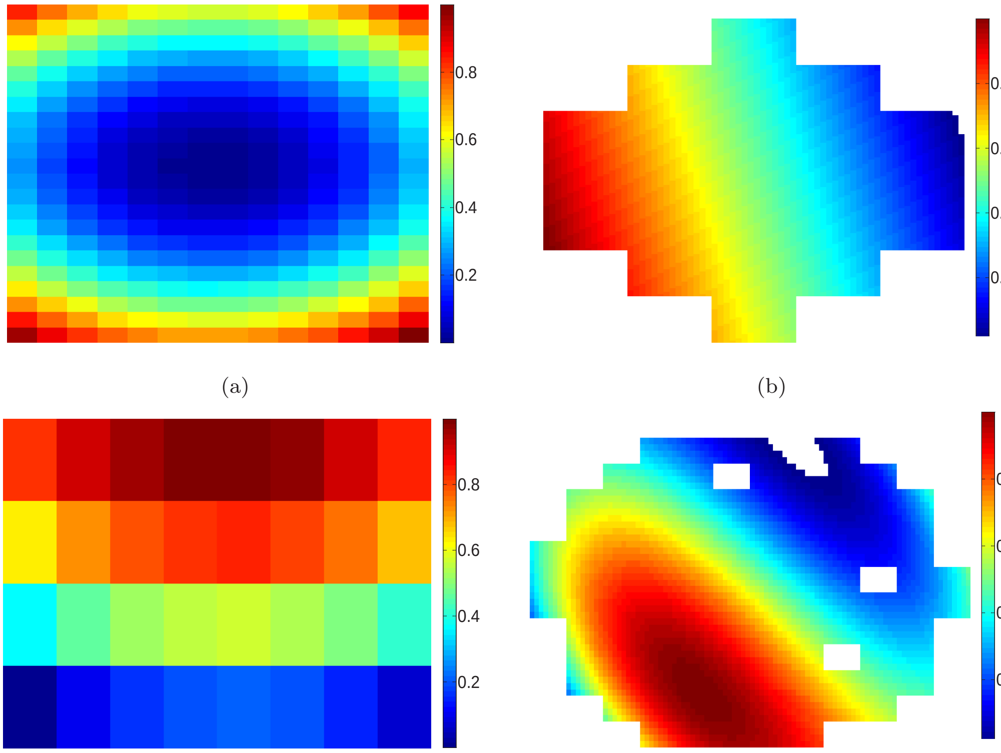
(a)

(b)

(c)

(d)

Figure 6.3: Reticle-level spatial pattern extraction for a wafer from process 2. (a): the extracted reticle-level common pattern $\overline{A}$; (b): smoothed result after MA; (c): estimated model using backward elimination and robust regression; (d): relative error of the fitted result in (c) in comparison with the original pattern in (a).



(a) (b)

(c) (d)

Figure 6.4: Deterministic spatial pattern extraction for 2 processes (scaled). (a): reticle-level pattern and (b): wafer-level pattern (slanted) for 130nm process; (c): reticle-level pattern and (d): wafer-level pattern (cubic) for 65nm process.

The non-deterministic part can be fully known only when a complete testing is conducted. Thus, a natural question is, can we characterize the reticle-level variation with certain accuracy when only partial testing is conducted? We propose to handle this problem using SBL.

Assume there are n entries within each reticle. The matrices A_{random}, A_{ar} and A_r can be vectorized to $n \times 1$ vectors, $\mathbf{t}_{\mathbf{random}}$, $\mathbf{t}_{\mathbf{ar}}$ and $\mathbf{t}_{\mathbf{r}}$, respectively. When m measurements are conducted $(m \leq n)$, we have:

$$\mathbf{t}_{\mathbf{m}} = B\mathbf{t}_{\mathbf{random}} = B\mathbf{t}_{\mathbf{ar}} + \mathbf{t}_{\mathbf{r},\mathbf{m}\times\mathbf{1}} \tag{6.13}$$

where $\mathbf{t}_{\mathbf{m}}$ is a $m \times 1$ vector, and B is a $m \times n$ selection matrix. Any row of B is a unit vector e_i, with the i_{th} entry equal to 1 and the other entries equal to 0. Since independent residual variations $\mathbf{t}_{\mathbf{r}}$ cannot be estimated but only bounded, the question turns out to be, given measurement of $\mathbf{t}_{\mathbf{m}}$ known, how to characterize $\mathbf{t}_{\mathbf{ar}}$ which are masked by $\mathbf{t}_{\mathbf{r},\mathbf{m}\times\mathbf{1}}$?

```
Procedure: Deterministic spatial pattern extraction
Input: the raw data matrix $A$ and the corresponding coordinate matrix $\mathbf{X}$, $\mathbf{Y}$,
sub-block size $[m_0, n_0]$ for MA, initial polynomial model order $n$
Output: the fitted polynomial model $f_n(\mathbf{X}, \mathbf{Y})$, the confidence interval $C = [lb, ub]$ for the fitted parameters in $f_n(\mathbf{X}, \mathbf{Y})$

Perform MA on $A$ to achieve the smoothed matrix $A_s$;
Vectorize $A_s$, $\mathbf{X}$, $\mathbf{Y}$ to $\mathbf{t_{A_s}}$, $\mathbf{x}$ and $\mathbf{y}$;
Construct a full $n_{th}$-order polynomial model $f_n(\mathbf{x}, \mathbf{y})$;
While TRUE
        Fit $\mathbf{t_{A_s}} \sim f_n(\mathbf{x}, \mathbf{y})$ to achieve the parameters in $\mathbf{p}$ and achieve its
        confidence interval $C = [lb, ub]$;
        Perform t-test on each parameter in $\mathbf{p}$;
        If all p-values are statistically significant
            return $f_n$ and corresponding coefficients;
        else;
            find the most insignificant term and remove it from the model
            $f_n$;
        endif;
end while;
```

Figure 6.5: Algorithm for deterministic spatial pattern extraction

We can associate (6.13) with SBL by applying a sparsity inducing transform Ψ^T on $\mathbf{t_{ar}}$ [89, 87]:

$$\mathbf{w} = \Psi^T \mathbf{t_{ar}} \tag{6.14}$$

where Ψ^T may be an orthogonal transform matrix for either discrete cosine transform (DCT) or discrete wavelet transform (DWT), *i.e.*, $\Phi^T = \Phi^{-1}$, and $\mathbf{w}$ is sparse or has a few entries that are more significant than the rest. (6.13) can now be written as:

$$\mathbf{t_m} = B\Psi\mathbf{w} + \mathbf{t_r} = \Phi\mathbf{w} + \mathbf{t_r} \tag{6.15}$$

where $\Phi = B\Psi$, and $\mathbf{w}$ is sparse with k significant entries (k usually is much smaller than m). $\mathbf{w}$ and $\mathbf{t_{ar}}$ have a canonical one-to-one relationship as in (6.14). Thus if $\mathbf{w}$ is accurately estimated, we can always recover $\mathbf{t_{ar}}$ from $\mathbf{w}$. If the sparsity inducing transform Ψ is applied to $\mathbf{t_{random}}$ instead of $\mathbf{t_{ar}}$, *i.e.*, $\mathbf{t_m} = B\mathbf{t_{random}} = B\Psi\mathbf{w}$, it turns out to be a compressive sensing problem as in [85]. Compressive sensing is an alternative application of SBL [87], which also requires $\mathbf{w}$ to be sparse and can be solved by linear regression in a point estimate manner [85, 89]. However, including independent variation into the transform may induce high frequency components and hence the non-sparsity in the frequency domain, thereby limiting the efficiency of compressive sensing. As a result we separate $\mathbf{t_{ar}}$ from $\mathbf{t_{random}}$ to maintain the sparsity in $\mathbf{w}$.

In our framework, the SBL method in [88] is applied to (6.15) to predict $\mathbf{w}$ through the measurements of $\mathbf{t_m}$. Similar as section 6.1.3, we have:

$$\begin{aligned}\textbf{Max} \quad L(\alpha) &= log \int p(\mathbf{t_m}|\mathbf{w}, \sigma^2) p(\mathbf{w}|\alpha) d\mathbf{w} \\ &= -1/2[Nlog2\pi + log|C| + \mathbf{t_m^T} C^{-1} \mathbf{t_m}]\end{aligned} \tag{6.16}$$

(6.16) can be solved by either using a variational Bayesian inference or more directly in a constructive manner by adding/deleting the candidate basis (column of Φ) into/from the solution model till the likelihood is converged [86, 87, 88]. Then, with the estimated $\mu_{\mathbf{MP}}$ and its covariance matrix Σ_{MP} in (6.6), the distribution of $\mathbf{t}_{ar}$ given $\mathbf{t_m}$ can be written as:

$$\mathbf{t}_{ar}|\mathbf{t}_m \sim N(\Psi \mu_{\mathbf{MP}}, \Omega), \quad \Omega = \Psi \Sigma_{MP} \Psi^T \tag{6.17}$$

However, to improve the efficiency of SBL in variation extraction, we still need to determine the best test sites to be measured (or the selection matrix B) and the order to add the bases by information collected from the test wafers, which will be detailed in the next section.

6.3 Active Learning Framework for Variation Extraction

The framework is composed of two major stages, active training to learn the models and model adaptation to adjust the models (Figure 6.2). In this section we will discuss the key modules in those two stages.

6.3.1 Active Training

In the active training stage, the framework learns the models by densely measuring each test structure in a wafer (or training wafer set). Several tasks are supposed to be conducted by learning the features of the measurements (denoted as W), including:

- *Model identification* that identifies both the wafer- and reticle-level deterministic spatial patterns.
- *Uncertainty exploration* that exploits the uncertainty reduction each measurement can contribute and scores the contribution.
- *Basis significance ranking* that gives an initial order of adding bases into the solution when solving (6.16).

```
Procedure: Model identification to extract spatial patterns
Input: the raw data $W$ of wafer 1 and the corresponding coordinate matrix $X$, $Y$
Output: the fitted polynomial model $f_w(X,Y)$ and $f_r(X_0,Y_0)$, confidence interval $C_w = [lb_w, ub_w]$ for the parameters in $f_w(X,Y)$ and $C_r = [lb_r, ub_r]$ for $f_r(X_0,Y_0)$
Compute the reticle-level coordinate matrix ($X_0$, $Y_0$);
Compute the averaged matrix $\overline{A}$ from all the reticles in $W$;
Extract the reticle-level common pattern model $f_r(X_0,Y_0)$ from $\overline{A}$ using the algorithm in Figure 6.5;
Compute the residual raw data with reticle-level common pattern removed, name it $W_g$;
Extract the wafer-level spatial pattern model $f_w(X,Y)$ from $W_g$ using the algorithm in Figure 6.5;
```

Figure 6.6: Algorithm to identify the wafer- and reticle-level pattern models

- *Linking variance to prediction accuracy* that sheds insight into the control of the estimation accuracy.

Once all those models are characterized in training stage, we start to process the forthcoming wafers with the learned models.

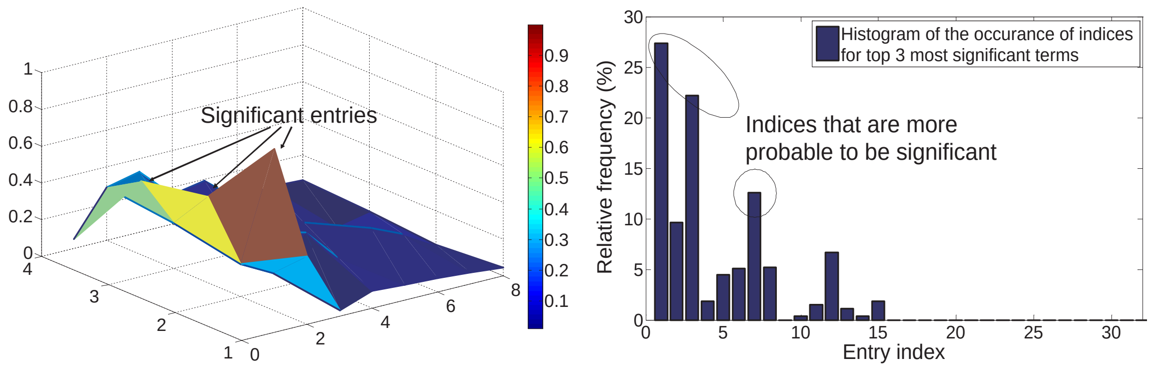
6.3.1.1 Model Identification

Most of the model identification details are discussed in section 6.2.2.1. Here we briefly summarize the flow to extract wafer- and reticle-level deterministic patterns in Figure 6.6. After the global trends are removed from W, we denote the residual raw data as W_r, which enables the exploration in the next three sections.

6.3.1.2 Uncertainty Exploration

Any measurement may provide a certain amount of information and hence reduce the model uncertainty. If all the available test structures are measured, the uncertainty of a model is exactly zero. At training stage we do not have any given variation information. Although it is always preferred to conduct the measurements that reduce the uncertainty to a maximum, it remains unclear how to quantitatively evaluate the uncertainty each measurement can reduce.

Here we propose a simple yet efficient method to evaluate the uncertainty reduction ability for any test site. Denote the raw data of a reticle from W_r as $A_{random,i}$. If all n test structures in the reticle are measured, the uncertainty of the model for this particular reticle is 0, *i.e.*, $U(A_{random,i})$=0, where $U(.)$ denotes the model uncertainty. On the other hand, if there is only one test structure (j_{th}) unmeasured, SBL in section 6.2.2.2 can estimate its value as well as a covariance matrix Ω_j as in (6.17). Clearly the variance terms in Ω_j is due to the unmeasured site, *i.e.* the uncertainty is



(a) (b)

Figure 6.7: (a): Scaled DCT transform for a reticle with deterministic patterns removed; (b): Histogram of indices for entries in $\mathbf{w}$ that fall into the top three most significant entry set. The results are collected from 100 reticles of a wafer.

attributed to the j_{th} site. Now define total variance as a measure of uncertainty,

Definition 1: Total variance is defined as the sum of variance for each test site, *i.e.*, $TV(j) = tr(\Omega_j)$, where $tr(.)$ computes the sum of diagonal entries and Ω_j is the estimation covariance matrix with j_{th} site unmeasured.

Then the uncertainty reduction by the j_{th} site can be approximated by:

$$\Delta U(j) = TV(j) \tag{6.18}$$

In this manner, we can check each site in the reticle and name the resulted vector as $\mathbf{\Delta} U_i$ for $A_{random,i}$. To mitigate some local effects, we will compute $\mathbf{\Delta} U_i$ for a representative set of l reticles (*e.g.*, the reticles in the middle column and middle row of the training wafer) and then take the average of them, $\overline{\mathbf{\Delta} U} = \sum \mathbf{\Delta} U_i/\text{l}$, as a measure of the uncertainty reduction ability. The normalized vector $\mathbf{S}_u = \overline{\mathbf{\Delta} U}/||\overline{\mathbf{\Delta} U}||$ is considered as the uncertainty score for the sites within a reticle. A site with a higher score is always preferred to be tested first.

6.3.1.3 Basis Significance Ranking

SBL can solve (6.16) by adding/deleting the candidate bases (columns of Φ) to/from the model and then updating the corresponding hyper-parameters for the selected bases [87]. For an unselected basis, its hyper-parameter is infinity, and the corresponding entry in $\mathbf{w}$ is 0. In other words, the selected bases correspond to significant entries in $\mathbf{w}$. If we know those significant entries in advance, we can simply plug the corresponding candidate basis set into the solution model, which is beneficial for both run-time and accuracy.

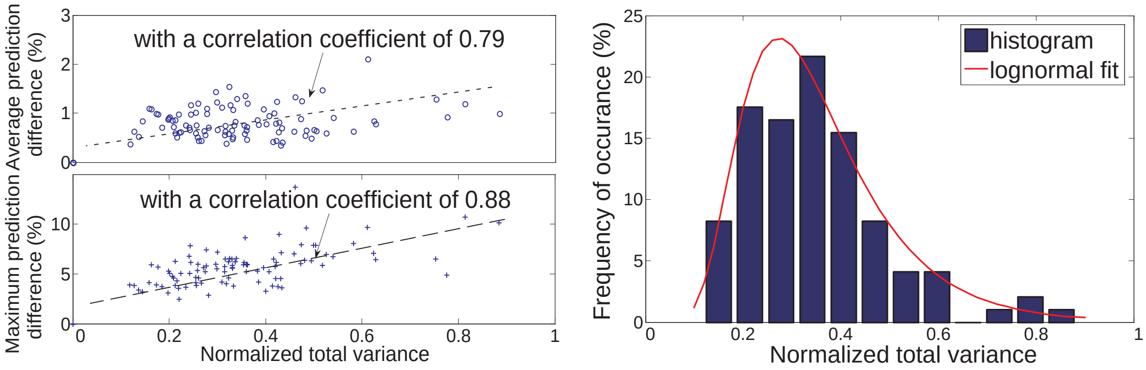
Figure 6.7(a) illustrates the DCT transform of a reticle with global patterns extracted. It can be seen most entries in the frequency domain happen to be insignificant. On the other hand, the significant entries in $\mathbf{w}$ are limited to a small subset even from reticle to reticle. Figure 6.7(b) shows the histogram of top 3 most significant entries in $\mathbf{w}$, across 100 reticles within a wafer. Though there are potentially 32 entries, the entries that are probable to be very significant are actually a small subset. We then propose a score list S_s to rank the significance of each entry. The basis with a higher significance score in S_s is first selected into the solution model. In details, we first sort the significance of orthogonal transform coefficients $\mathbf{w}$ from the most to the least significant for each reticle. The rank vector, $\mathbf{r}_i$, for a reticle is then scaled and fed into a continuous score function to achieve the score for the bases:

$$\mathbf{S_{s,i}} = \exp(-p_s \times \mathbf{r}_i/max(\mathbf{r}_i)) \tag{6.19}$$

where p_s is a customized parameter to tune the slope of the exponential function. Then the score for basis significance is achieve by taking average of $\mathbf{S_{s,i}}$ for all the reticles, *i.e.*, $\mathbf{S_s} = \sum \mathbf{S_{s,i}}/l$.

6.3.1.4 Linking Variance to Prediction Accuracy

When a reticle is under partial testing, it is essential to know how accurate the prediction may be. However, the accuracy is unable to be known till all the measurements are conducted. Thus, we need to find another measure to quantify the quality of the estimation. By noting that the error is almost always positively correlated with the model uncertainty, we propose to use total variance (TV) in **Definition 1** for a given set of k_0 measurements as a measure of the prediction accuracy. Figure 6.8(a) shows trends of the average (top) and maximum (bottom) relative prediction error with respect to TV for 100 reticles of a wafer. Either figure exhibits a strong correlation with TV, with a correlation coefficient of 0.79 and 0.88, respectively. The relationship between TV and prediction accuracy is explored in the training stage by conducting k_0 measurements on each reticle according to $\mathbf{S_u}$ and then recording the total variance for each reticle. It is noted that we are attempting to evaluate estimation accuracy in a *qualitative* instead of *quantitative* way. The data collected from the training wafer helps describe the statistical behavior of TV in this variation space given k_0 measurements. Figure 6.8(b) shows the histogram of total variance from 100 reticles and its log-normal fit. This histogram is used to decide the number of measurements to be conducted, as in section 6.3.2.2.



(a)

(b)

Figure 6.8: (a): Strong correlation of prediction accuracy and total variance (TV); (b): Histogram of total variance across 100 reticles of a wafer for 65nm process

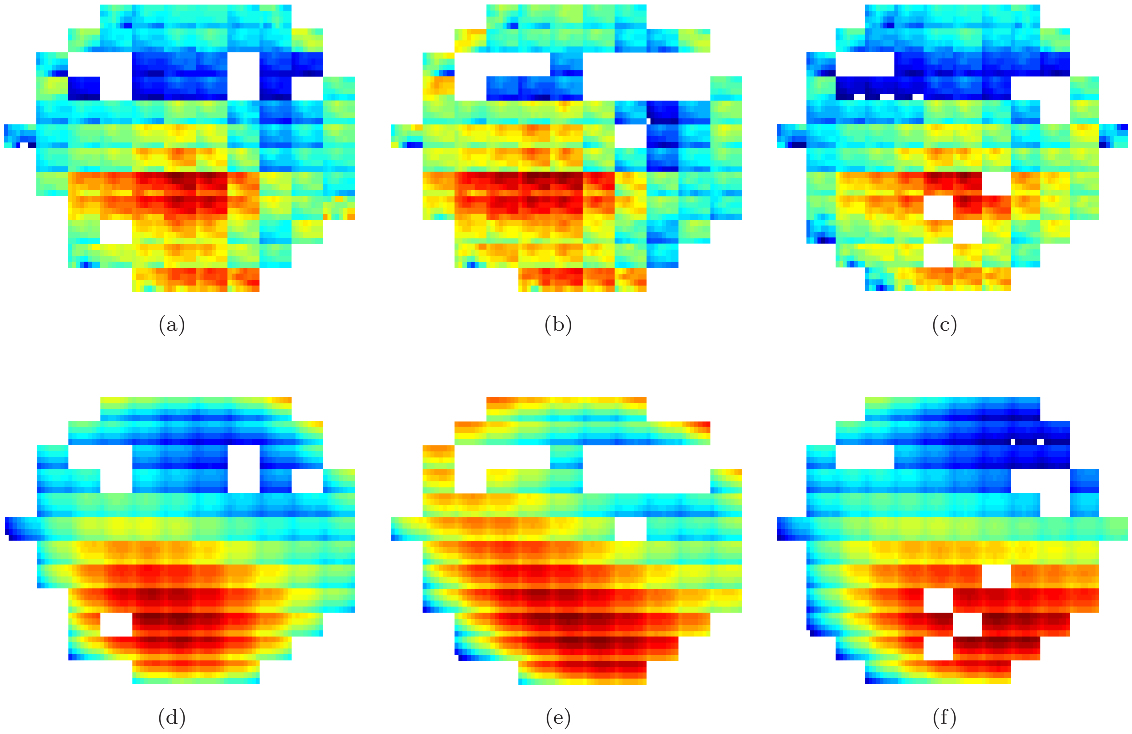
6.3.2 Model Adaptation

After models at different levels are constructed in training stage, the framework starts partial testing on the other wafers (from the same lot or different lots), which is the model adaptation stage.

6.3.2.1 Model Validation

It is essential to validate the models before applying them to an untested wafer. The model validation justifies if the deterministic pattern models $f_w(X,Y)$ and $f_r(X_0,Y_0)$ requires a complete reconstruction or just incremental adjustment. For the current wafer under test (WUT), the validation module selects a representative set of reticles. Then the χ^2 test in section 6.1.2 is used to evaluate the overall fitting goodness. If the overall fitting is good, a t-test is then applied to judge if any parameter in $f_w(X,Y)$ and $f_r(X_0,Y_0)$ needs adjustment. The validation flow is summarized as follows:

- Step 1: Compute the chi-square statistics χ^2 using the raw data from the representatives set of reticles [86].
- Step 2: If χ^2 is beyond the predefined tolerance bound, include the WUT into the training set and go back to the training stage. Otherwise, go to Step 3.
- Step 3: Extract the wafer- and reticle-level patterns from the representative reticle set using the algorithms in section 6.3.1.1 to achieve the comparison models $f_{w,c}(X,Y)$ and $f_{r,c}(X_0,Y_0)$.
- Step 4: Check if the parameters of $f_{w,c}(X,Y)$ (or $f_{r,c}(X_0,Y_0)$) are in the confidence intervals C_w (or C_r) for the parameters of $f_w(X,Y)$ (or $f_r(X_0,Y_0)$).


(a) (b) (c)

(d) (e) (f)

Figure 6.9: Contours of 3 wafers from the same lot. (a): wafer 1 (training wafer); (b): wafer 2; (c): wafer 3; and the extracted deterministic spatial patterns (including both wafer- and reticle-level patterns) for (d): wafer 1; (e): wafer 2; (f): wafer 3.

- Step 5: For those within the bounds, accept the original parameters in $f_w(X, Y)$ or $f_r(X_0, Y_0)$; for those beyond the bounds, re-fit the parameters using the representative set of reticles.

Figure 6.9 demonstrates the contours of three wafers from the same lot and how the deterministic pattern models evolve adaptively from wafer to wafer. It can be seen the deterministic patterns of wafer 2 and wafer 3 (Figure 6.9(e) and (f)) are based on the training patten (Figure 6.9(d)) but still capture the major features of the original wafer contours.

6.3.2.2 Adaptive Test Configuration Determination

The most important problem for partial test on a reticle is which and how many measurements to conduct. The uncertain score is a global evaluation across the reticles and not efficient enough for a particular reticle. To better understand the across-reticle process condition, the proposed partial test has two phases, within each, k_0 and k_1 measurements are conducted respectively.

Assume we have n_{max} available site for testing. In the first phase we conduct k_0 measurements on the reticle according to $\mathbf{S}_u$ and apply SBL to achieve the covariance matrix $\Omega_{S_{k_0}}$ [11]. k_0 is the same as the number used in section 6.3.1.4 to characterize a TV histogram (or distribution with a cumulative density function $F(.)$). Then the total variance (*e.g.*, TV_i) for the reticle is computed from $\Omega_{S_{k_0}}$. The relative location of TV_i in the distribution $F(.)$ is mapped to the number of measurements to be conducted. In other words, we have:

$$k_1 \sim F(TV_i) \times (n_{max} - k_0) \tag{6.20}$$

Thus, if TV_i is large, almost all the measurements will be conducted. Meanwhile fewer measurements are required for smaller TV_i.

The second phase determines where to conduct the k_1 measurements. The underlying motivation is to maximize the uncertainty reduction by those measurements, *i.e.*,

$$\mathbf{Max} \quad \Delta U_{S_{k_0} \cup S_{k_1}} - \Delta U_{S_{k_0}} \tag{6.21}$$

However, this formulation itself is difficult to evaluate or optimize. It is also noted that if we simply choose test sites according to S_u, most measurements may be conducted at some corner due to the spatial correlation. It is therefore desirable to conduct measurement on sites with higher S_u as well as relatively uniform distribution. The uniformity requirement is equivalent to the consideration of correlation among test sites. The closely placed sites usually have higher correlation, and hence should avoid repeated measurements. This intuition motivates a greedy search algorithm to maximize the covariance reduction instead of variance. Given the covariance matrix $\Omega_{S_{k_0}} = [\omega_1, \omega_2 ... \omega_n]$ for n test sites from phase 1 and ω_i is a column in $\Omega_{S_{k_0}}$,

- Step 1: set $S_{k_1} = \emptyset$.
- Step 2: choose the site i_0, which is:

  $$i_0 = argmax_i \quad |\omega_i|^T \times [\mathbf{1}]_{n\times 1} \tag{6.22}$$

  where $[\mathbf{1}]_{n\times 1}$ is an $n \times 1$ all-one vector.
- Step 3: $S_{k_1} = S_{k_1} \bigcup i_0$. Remove the i_{0th} column and row in $\Omega_{S_{k_0}}$.
- Step 4: Go back to step 2 till k_1 sites are found.

[11] Those measurements are typically a very small subset S_{k_0} that helps understand the process condition without much extra overhead to the framework. k_0 can be determined at training stage by sweeping several typical numbers and comparing the results.

6.3.2.3 Prediction and Model Update

After test configuration is determined, the framework moves forward to estimate the deterministic and non-deterministic variation components for any untested site. The deterministic components are calculated using those validated models, given the locations of the device in the reticle and wafer. The non-deterministic component is acquired by SBL in section 6.2.2.2. The estimated process parameter is then the sum of those components:

$$z_{esti} = f_w(x, y) + f_r(x_0, y_0) + \widetilde{z_{ar}} \tag{6.23}$$

where $\widetilde{z_{ar}}$ is the estimated spatially correlated variation component. The estimation also comes with a confidence interval that is computed from the covariance matrix Ω and the estimated σ^2 as in section 6.2.2.2 [87].

To ensure the quality of the prediction, we also employ a cross-validation stage by using another small subset S_{cv} of measurements to validate the results. Similar as S_{k_0}, S_{cv} is determined according to the uncertainty score $\mathbf{S}_u$ of those unmeasured sites. If the average error of the prediction results is unable to meet an error tolerance threshold θ_{cv}, more measurements will be conducted till cross validation requirement is satisfied. It is noted that we can always adjust θ_{cv} and n_{max} in section 6.3.2.2 to make tradeoff between estimation accuracy and test cost (number of measurements).

After the prediction results are validated, several models needs updating before the framework moves to the next wafer:

- Total variance distribution. The total variance with S_{k_0} measurements can be collected from each reticle and included into the original TV data set. Kurtosis statistics is applied here to monitor if the distribution has a fundamental change. Otherwise the next wafer will be set to a training wafer to reconstruct the TV distribution.

- Basis significance score. According to the prediction values, the significance score for each reticle can be computed and then combined with the original score.

- Uncertainty score. The covariance matrix for each reticle is achieved from SBL which also sheds insight into the uncertainty reduction ability.

6.3.3 Summary of the Active Learning Framework

Here we summarize the complete algorithm for our active learning framework in Figure 6.10.

```
Procedure: Summary of the active learning framework
Input: wafers
Output: variation components decomposition, process parameter estimation
        and corresponding confidence intevals
1: Perform complete testing on the training wafer and actively learn models
   at different scales as in section 6.3.1;
2: Validate the trained models for the next untested wafer as in section
   6.3.2.1;
3: Achieve the adjusted wafer- and reticle-level spatial pattern models
   $f_w(X, Y)$ and $f_r(X_0, Y_0)$;
4: Determine the test configuration for each untested reticle as in section
   6.3.2.2;
5: Estimate the process parameter for untested devices, validate results
   using cross-validation and update the models;
6: Move to the next wafer;
```

Figure 6.10: Summary of the active learning framework

Table 6.1: Simulation results of the proposed framework on two industrial processes (65nm and 130nm)

proc.	#wafers	ave. err.	#failure	#measure	var. reduc.	time/wafer
1	288	0.8%	0.031%	45.9%	83%	20.3 sec
2	5	1.4%	0.025%	54.2%	78%	25.7 sec

6.4 Experimental Results

In this section we demonstrate the efficiency and accuracy of the proposed framework based on the industrial measurement data from two processes in 65nm and 130nm technologies. All the experiments are conducted on a 2.0GHz Linux machine with 32GB RAM.

Table 6.1 summarizes the performance and accuracy of the framework, with a 2% average error tolerance in the cross-validation stage and with all the test structures available for measurement. Column 2 shows the total number of wafers for two processes. Column 3 is the average relative error which is computed by:

$$\frac{1}{N_t} \sum_{\text{any untested site}} \frac{|\text{estimated value} - \text{actual measurement}|}{\text{actual measurement}} \tag{6.24}$$

where N_t is the number of untested structures. Our framework can achieve 0.8% and 1.4% relative average error for two processes, respectively. In column 4, we present the relative failure number to evaluate the efficacy of the estimated confidence intervals. A failure is defined as an untested structure whose actual process (measurement) is beyond the estimated confidence interval. The relative failure is approximately 0.03% across all the lots. This is in good agreement with the expected failure of 0.1% from 3σ bounds. Column 5 exhibits the ratio of the number of measurements over the total number of available test structures. The test cost reduction is up to 50% to achieve an average

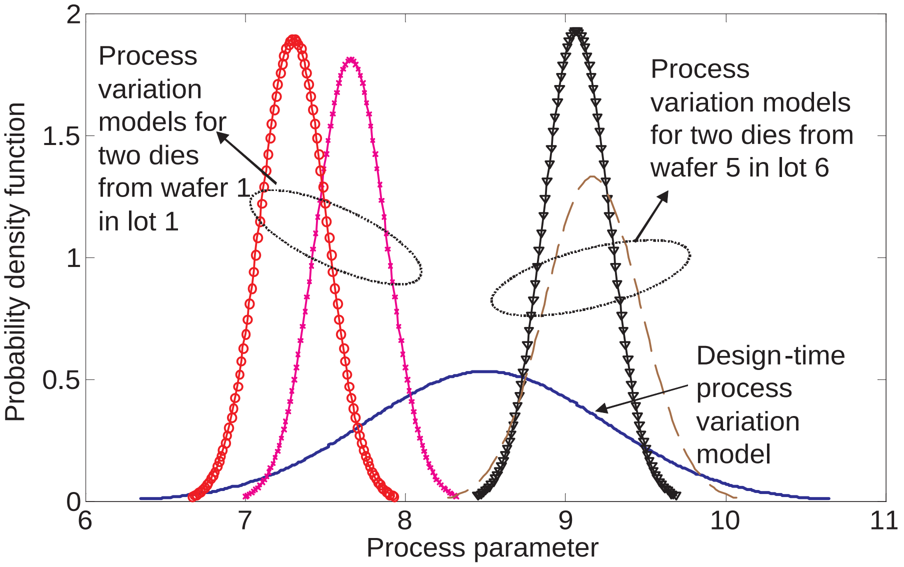


Figure 6.11: Comparison of the post-silicon variation models for 4 dies from 2 wafers in 2 lots and traditional design-time process variation model

error of ∼1%. The greater saving of test cost can be achieved by loosing the error tolerance, which will be presented shortly in this section. Column 6 is the average variance reduction of the proposed post-silicon variation model in comparison with the traditional design-time variation models (characterized from all the wafers). Even with half the test structures measured, the model variance can be reduced by approximately 80% (5× tighter) for either process, which may significantly reduce the pessimism in post-silicon applications. The average run time per wafer is listed in the last column and is expected to be smaller with increased number of wafers. Figure 6.11 further illustrates the significantly tightened post-silicon process variation models for 4 dies from 2 different wafers in 2 different lots. In the contrast, the design-time model is widely distributed because the wafer/reticle specific data is not available at design-time.

Figure 6.12 shows the evolution of the average relative error (bottom) and test measurement ratio (top) from wafer to wafer for all the lots of process 1. The black dashed line denotes the transition from one lot to another. The spikes of the measurement ratio at some of the transitions are due to the global pattern difference between two different lots (as shown in Figure 6.1). However, the framework can adapt the model to capture this difference. Across the lots, the relative average error is well maintained at approximately 1%.

Figure 6.13 presents the histograms of average relative error and maximum relative error for 100 reticles from the same wafer. Most reticles have limited average error of approximately 1% and maximum error smaller than 10%. Figure 6.14 clearly demonstrates the reduction trend of both average

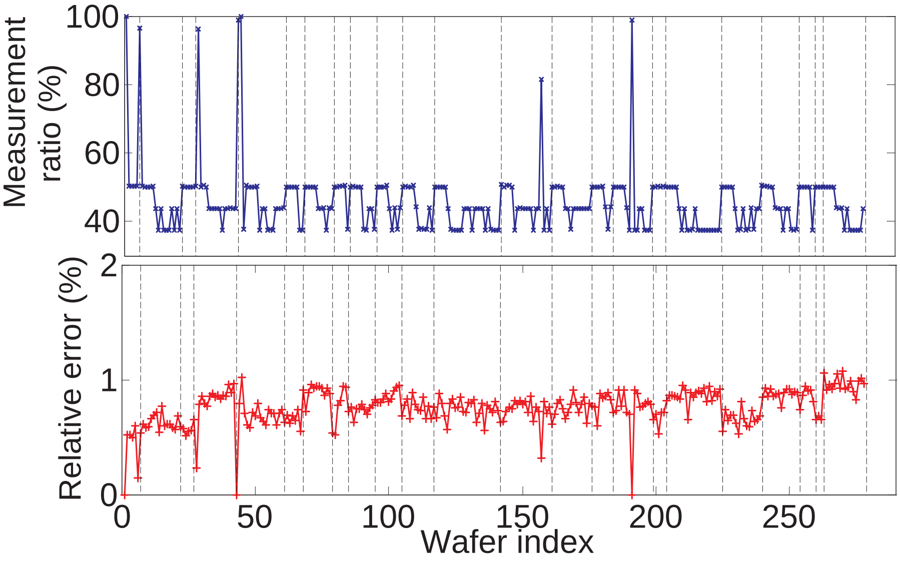


Figure 6.12: Evolvement of relative average error (bottom) and measurement ratio (top) across the wafers and lots from process 1

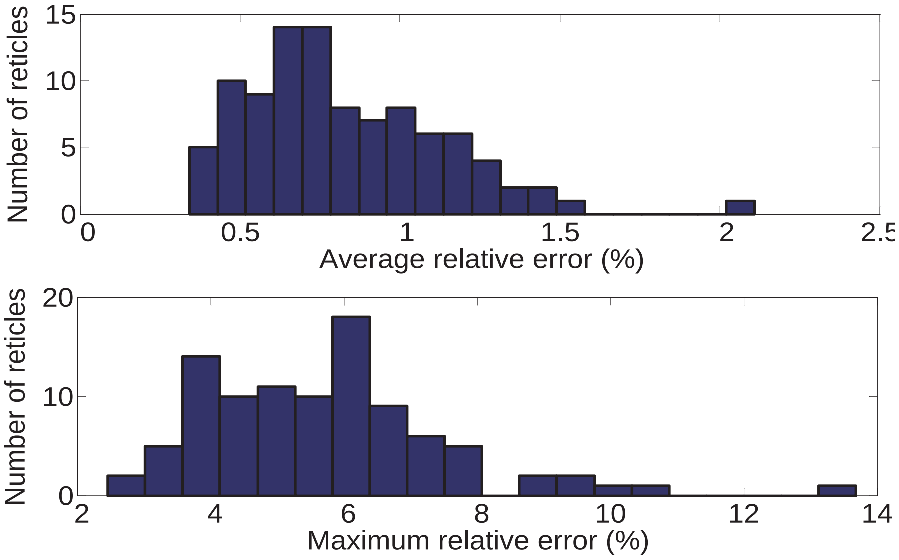
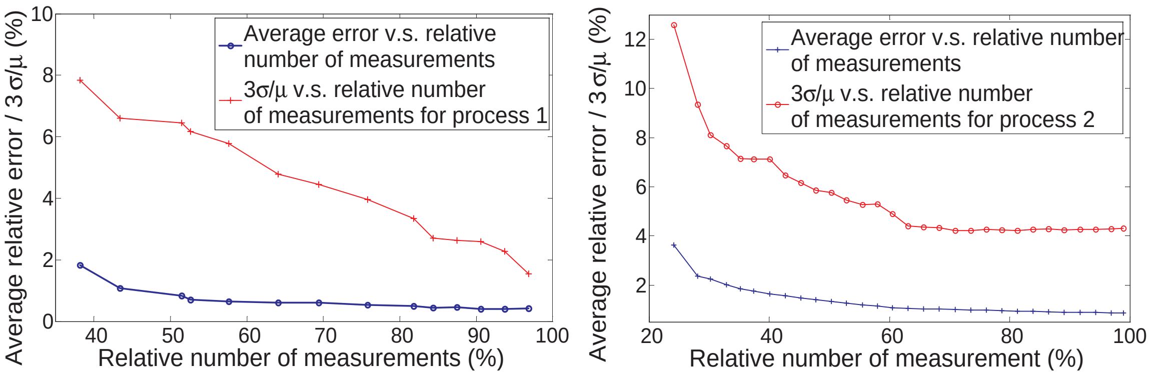


Figure 6.13: Histograms of average (top) and maximum (bottom) relative error for all the reticles in a wafer

error and scaled average variance (=$3\sigma/\mu$) with an increased number of measurements for both 65nm and 130nm processes. With approximately 30% available test structures, the framework can still achieve ∼2-3% average relative errors for two processes. The accuracy of the proposed framework is compared in Figure 6.15 with another two methods, the virtual probe method in [85] and a bilinear interpolation method on the wafers of 130nm process. The proposed framework can achieve a better accuracy with much fewer measurements. For the same accuracy of approximately 2% relative error,



(a)

(b)

Figure 6.14: Trend of average error and scaled average variance ($3\sigma/\mu$) reduction with an increased number of measurements for (a): 65nm and (b): 130nm process

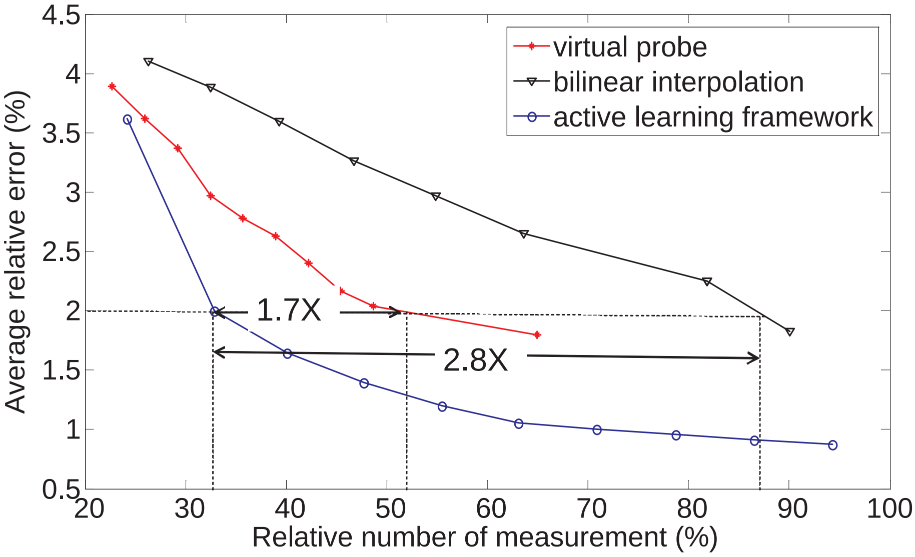


Figure 6.15: Average relative error comparison with increased measurements for the wafers from process 2 using the active learning framework, virtual probe method from [85] and bilinear uniform interpolation

the test cost is reduced by 37% and 75% compared with [85] and bilinear interpolation, respectively.

6.5 Summary

This chapter proposes an active learning framework to extract process variation from measurements and reduce test cost. Several techniques are developed to model the variation. By reusing a priori knowledge from earlier wafers, the partial test can be conducted on the forthcoming wafers to achieve the required accuracy and test cost. Experimental results show that the framework can achieve an accuracy of ∼2-3% relative error using only ∼30% test structures for two industrial processes.

CHAPTER VII

Conclusions and Future Work

This thesis discusses several key problems for nano-meter regime yield enhancement when accounting for the process and environmental variabilities. The topics span both pre- and post-silicon optimization techniques for power, performance and reliability demands. Besides that, we also discuss how to efficiently extract variation from limited measurements for post-silicon use.

In the dissertation research work, we first presented a **pre-silicon optimization** technique that selects body bias voltage for several cluster of gates at design time. By fixing body voltage at design time, the testing overhead to determine the body bias is significantly saved. Meanwhile, it has a higher flexibility than other traditional design-time optimization techniques, like gate sizing and dual threshold voltage assignment. We also proposed a **post-silicon optimization** approach to trade off the oxide breakdown reliability margin for chip performance by using limited measurements. Based on our statistical oxide breakdown reliability analysis, we can achieve far more accurate chip lifetime estimation compared with the conventional guard band approach that assumes the worst case oxide thickness across the chips. The lifetime estimation is then fed to the optimization part and guides the maximum supply voltage to be chosen for the system. Since pre- and post-silicon techniques may target at the same design metrics, performing optimization separately may increase unnecessary overhead or even repeated operations. We then explore the possible **coordination** of adaptive body biasing and gate sizing by aligning the body bias profile distribution for gates at design time. It is also noted that accurate **variation modeling** is essential for efficient yield optimization. We then proposed an active learning framework to extract the post-silicon variation model for the product wafers. The framework helps the model evolve itself by reusing and learning the information from the tested wafers to validate and improve the models.

There may be several potential research challenges for the topics that were addressed in this thesis:

- In Chapter II, when clustering the gates for design-time body bias selection, we accounted for only the similarity of the body bias profiles, *i.e.*, the electrical behavior (candidate voltage selection to minimize leakage). However, due to the extra well spacing for cells with different body bias, it is also important to consider the physical constraints, *e.g.* routing feasibility or physical density when grouping gates together. Thus, an interesting research topic is to determine the clustering policy when considering both the body bias behavior and design physical constraints.

- In the oxide breakdown reliability analysis, the underlying assumption is the oxide thickness variation for any device follows a Gaussian distribution. It is possible to extend to the analysis with arbitrary distribution, which could be simplified using moments analysis.

- There are multiple failure mechanisms in chip designs. Oxide breakdown is only one of them. Most failure analysis are still based on the worst case corner. It is interesting to extend the oxide breakdown statistical framework to incorporate other failure modes, *e.g.* electro-migration.

- In our post-silicon reliability management framework, the measurement sites are uniformly selected. However, it is known that in real designs different location on a chip may provide different information and hence reduce the system uncertainty to a different degree. It is therefore helpful to explore the uncertainty reduction ability each site may have and hence provide designers with an adaptive measurement sites selection strategy.

- The achieved gate sizes in our co-ordination work are continuous and supposed to be snapped to discrete cell sizes. It remains an open question how to efficiently discretize the continuous solution to discrete values while maintaining the optimality of the solution.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] T. Karnik, *et.al.* Sub-90 nm technologies-challenges and opportunities for CAD. In *Proc. ICCAD*, pages 203–206, 2002.

[2] S. Borkar, *et.al.* Parameter variations and impact on circuits and microarchitecture. In *Proc. DAC*, pages 338–342, 2003.

[3] D. Frank, *et.al.* Design and CAD challenges in 45nm CMOS and beyond. In *Proc. ICCAD*, pages 329–333, 2006.

[4] B. Calhoun, *et.al.* Digital circuit design challenges and opportunities in the era of nanoscale CMOS. *Proc. of IEEE*, vol. 96, no. 2:343–365, 2008.

[5] M. Orshansky, S. Nassif, and D. Boning *Design for Manufacturability and Statistical Design: A Constructive Approach.* Springer, 2008.

[6] International technology roadmap for semiconductors, 2007 Edition.

[7] D. Blaauw, *et.al.* Statistical timing analysis: basic principles to state-of-the-art. *IEEE TCAD*, vol. 27, no. 4:589–607, 2008.

[8] J. Xiong, *et.al.* Robust extraction of spatial correlation. In *Proc. ISPD*, pages 2–9, 2006.

[9] M. Pelgrom, *et.al.* Matching properties of MOS transistors. In *IEEE JSSC*, vol. 24, no. 5:1433–1439, 1989.

[10] F. Liu. A general framework for spatial correlation modeling in VLSI design. In *Proc. DAC*, pages 817–822, 2007.

[11] K. Qian and C. J. Spanos. A comprehensive model of process variability for statistical timing optimization. In *Proc. SPIE*, pages 1–11, 2008.

[12] L. Cheng, *et.al.* Physically justifiable die-level modeling of spatial variation in view of systematic across wafer variability. In *Proc. DAC*, pages 104–109, 2009.

[13] A. Gattiker, *et.al.* Data analysis techniques for CMOS technology characterization and product impact assessment. In *Proc. ITC*, pages 1–10, 2006.

[14] A. Gattiker. Using test data to improve ic quality and yield. In *Proc. ICCAD*, pages 771–777, 2008.

[15] C. Chiang and J. Kawa, *Design for Manufacturability and Yield for Nano-Scale CMOS.* Springer, 2007.

[16] B. Shubhankar, *PerformanceModeling and Optimization Techniques in the Presence of Random Process Variations to Improve Parametric Yield of VLSI Circuits.* PhD thesis, University of Cincinnati, 2008.

[17] S. Borkar. Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. *IEEE Micro.*, vol. 25, issue 6:10–16, 2005.

[18] A. Srivastava, *et.al.* A novel approach to perform gate-level yield analysis and optimization considering correlated variations in power and performance. *IEEE TCAD*, vol. 27, no. 2:272–285, 2008.

[19] X. Bai, *et.al.* Uncertainty aware circuit optimization. In *Proc. DAC*, pages 58–63, 2002.

[20] S. Raj, *et.al.* A methodology to improve timing yield in the presence of process variations. In *Proc. DAC*, pages 448–453, 2004.

[21] A. Srivastava, *et.al.* Statistical optimization of leakage power considering process variations using dual-vth and sizing. In *Proc. DAC*, pages 773–778, 2004.

[22] D. Sinha, *et.al.* Statistical gate sizing for timing yield optimization. In *Proc. ICCAD*, pages 1037–1041, 2005.

[23] S. Bhardwaj and S. Vrudhula. Leakage minimization of nano-scale circuits in the presence of systematic and random variations. In *Proc. DAC*, pages 541–546, 2005.

[24] S. Bhardwaj, *et.al.* LOTUS: leakage optimization under timing uncertainty for standard-cell designs. In *Proc. ISQED*, pages 27–29, 2006.

[25] M. Mani, *et.al.* An efficient algorithm for statistical minimization of total power under timing yield constraints. In *Proc. DAC*, pages 309–314, 2005.

[26] J. Singh, *et.al.* Robust gate sizing by geometric programming. In *Proc. DAC*, pages 315–320, 2005.

[27] S. Kulkarni, *et.al.* A statistical framework for post-silicon tuning through body bias clustering. In *Proc. ICCAD*, pages 39–46, 2006.

[28] M. Mani, *et.al.* Joint design-time and postsilicon minimization of parametric yield loss using adjustable robust optimization. In *Proc. ICCAD*, pages 19–26, 2006.

[29] V. Khandelwal and A. Srivastava. Variability-driven formulation for simultaneous gate sizing and post-silicon tunability allocation. In *Proc. ISPD*, pages 17–25, 2006.

[30] J. Tschanz, *et.al.* Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage. *IEEE JSSC*, vol. 37, issue 11:1396–1401, 2002.

[31] K. Ishibashi. Adaptive body bias techniques for low power SOC. In *Presentation at IEEE ISSCC*, 2007.

[32] T. Chen and S. Naffziger. Comparison of adaptive body bias (ABB) and adaptive supply voltage (ASV) for improving delay and leakage under process variations. *IEEE TVLSI*, vol. 11, issue 5:888–899, 2003.

[33] Z. Weinberg. Hole injection and transport in SiO2 films on Si. In *Applied Physics Letters*, vol. 27, no. 8:437–439, 1975.

[34] E. Avni and J. Shappir. A model for silicon-oxide breakdown under high field and current stress. In *Journal of Applied Physics*, vol. 64, no. 2:734–742, 1988.

[35] J. Sune, *et.al.* On the breakdown statistics of very thin SiO2 films. In *Thin Solid Films*, vol. 185, issue 2:347–362, 1990.

[36] R. Degraeve, *et.al.* A consistent model for the thickness dependence of. intrinsic breakdown in ultra-thin oxides. In *Proc. IEDM*, pages 863–866, 1995.

[37] C. Zhuo, *et.al.* Design Time Body Bias Selection for Parametric Yield Improvement. to appear in *Proc. ASPDAC*, 2010.

[38] K. Chopra, *et.al.* A statistical approach for full-chip gate-oxide reliability analysis. In *Proc. ICCAD*, pages 698–705, 2008.

[39] C. Zhuo, *et.al.* Post-fabrication measurement-driven oxide breakdown reliability prediction and management. In *Proc. ICCAD*, pages 441–448, 2009.

[40] C. Zhuo, *et.al.* Variation-aware gate sizing and clustering for post-silicon optimized circuits. In *Proc. ISLPED*, pages 105–110, 2008.

[41] C. Zhuo, *et.al.* Process variation and temperature-aware reliability management. In *Proc. DATE*, pages 580–585, 2010.

[42] C. Zhuo, *et.al.* An active learning framework for post-silicon variation extraction and test cost reduction. In *Proc. ICCAD*, pages 508–515, 2010.

[43] K. Leung and P. Mok, A capacitor-free CMOS low-dropout regulator with damping-factor-control frequency compensation *IEEE JSSC*, vol. 38, no. 10:1691–1702, 2003.

[44] H. Tanaka, *et.al.* A precise on-chip voltage generator for a gigascale DRAM with anegative word-line scheme. *IEEE JSSC*, vol. 34, issue 8:1084–1090, 1999.

[45] I. Guyon, *et.al.*, editors. *Feature Extraction, Foundations and Applications.* Springer, 2006.

[46] A. R. Conn, *et.al.* *LANCELOT:a Fortran package for large-scale. nonlinear optimization (release A).* Springer, 1992.

[47] H. Chang and S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single pert-like traversal. In *Proc. ICCAD*, pages 621–625, 2003.

[48] C. Visweswariah, *et.al.* First order incremental block based statistical timing analysis. In *Proc. DAC*, pages 331–336, 2004.

[49] CPLEX, http://www.ilog.com/products/cplex/.

[50] N. Steen, *et.al.* Gaussian quadratures for the integrals $\int_0^\infty e^{-x^2} f(x)dx$ and $\int_0^b e^{-x^2} f(x)dx$. *Math. of Comp.*, vol. 23, no.107:661–671, 1969.

[51] S. Kotz, *et.al.* *Continuous Multivariate Distributions.* Wiley, 2000.

[52] V. Veetil, *et.al.* Efficient monte carlo based incremental statistical timing analysis. In *Proc. DAC*, pages 676–681, 2008.

[53] J. Roy and I. Markov, ECO-system: embracing the change in placement. University of Michigan, Ann Arbor, MI, Tech. Rep. CSETR-519-06, 2006.

[54] C. Hu. Gate oxide scaling limits and projection. In *Proc. IEDM*, pages 319–322, 1996.

[55] J. Stathis. Physical and predictive models of ultra thin oxide reliability in CMOS devices and circuits. *IEEE Trans. on Devices and Materials Reliability*, vol. 1, no. 1:43–59, 2001.

[56] B. Kaczer, *et.al.* Observation of hot-carrier-induced nFET gate-oxide breakdown in dynamically stressed CMOS circuits. In *Proc. IEDM*, pages 171–174, 2002.

[57] J. Sune and E. Y.Wu. Statistics of successive breakdown events in gate oxides. *IEEE Electron Device Letter*, vol. 24, issue 4:272–274, 2003.

[58] Y. Lee, *et.al.* Prediction of logic product failure due to thin-gate oxide breakdown. In *Proc. IRPS*, pages 18–28, 2006.

[59] J. Sune. New physics-based analytic approach to the thin-oxide breakdown statistics. *IEEE Electron Device Letter*, 22:296–298, 2001.

[60] E. Wu, D. Harmon, and L. Han. Interrelationship of voltage and temperature dependence of oxide breakdown for ultrathin oxides. *IEEE Electron Device Letter*, vol. 21 , issue 7:362–364, 2000.

[61] E. Wu, J. Sune, W. Lai, E. Nowak, J. McKenna, A. Vayshenkerb, and D. Harmon. Interplay of voltage and temperature acceleration of oxide breakdown for ultra-thin gate oxides. *Microelectronic Engineering*, vol. 59, issues 1-4:25–31, 2001.

[62] R. Degraeve, *et.al.* Temperature acceleration of oxide breakdown and its impact on ultra-thin gate oxide reliability. In *Proc. VLSIT*, pages 59 – 60, 1999.

[63] D. Brooks, *et.al.* Power, thermal, and reliability modeling in nanometer-scale microprocessors. *IEEE Micro.*, vol. 27, no. 3:49–62, 2007.

[64] E. Karl, *et.al.* Reliability modeling and management in dynamic microprocessor-based systems. In *Proc. DAC*, pages 1057–1060, 2006.

[65] HotSpot, http://lava.cs.virginia.edu/HotSpot/

[66] K. Skadron, *et.al.* Temperature-aware microarchitecture. In *Proc. ISCA*, pages 2–13, 2003.

[67] Y. Lee, *et.al.* Implant damage and gate-oxide-edge effects on product reliability. In *Proc. IEDM*, pages 481–484, 2004.

[68] E. Avni and J. Shappir. A model for silicon-oxide breakdown under high field and current stress. *Journal of Applied Physics*, 64:734–742, Jun. 1988.

[69] E. Wu, *et.al.* Ultra-thin oxide reliability for ulsi applications. *Semicond. Sci. Technol.*, vol. 15, no. 5:425–435, 2000.

[70] J. P. Imhof. Computing the distribution of quadratic forms in normal variables. *Biometrika*, 48:419–426, 1961.

[71] K.-H. Yuan and P. M. Bentler. Two simple approximations to the distributions of quadratic forms. Paper 2007010106, Department of Statistics, UCLA, 2007.

[72] D. Brooks, *et.al.* Wattch: a framework for architectural-level power analysis and optimizations. In *Proc. ISCA*, pages 83–94, 2000.

[73] S. Reda, *et.al.* Analyzing the impact of process variations on parametric measurements: Novel models and applications. In *Proc. DATE*, pages 375–380, 2009.

[74] W. Meeker and L. Escobar. *Statistical methods for reliability data.* Wiley, 1998.

[75] E. Karl, *et.al.* Analysis of system-level reliability factors and implications on real-time monitoring methods for oxide breakdown device failures. In *Proc. ISQED*, pages 391–395, 2008.

[76] E. Karl, *et.al.* Compact in-situ sensors for monitoring negative-bias-temperature-instability effect and oxide degradation. In *Proc. ISSCC*, pages 410–623, 2008.

[77] J. Keane, *et.al.* An array-based test circuit for fully automated gate dielectric breakdown characterization. In *Proc. CICC*, pages 121–124, 2008.

[78] Q. Liu, *et.al.* Confidence scalable post-silicon statistical delay prediction under process variation. In *Proc. DAC*, pages 496–502, 2007.

[79] A. Agarwal, *et.al.* Statistical timing analysis for intra-die process variations with spatial correlations. In *Proc. ICCAD*, pages 900–907, 2003.

[80] B. Stine, *et.al.* Analysis and decomposition of spatial variation in integrated circuit processes and devices. *IEEE TSM*, vol. 10, no. 1:24–41, 1997.

[81] D. Beece, *et.al*. Transistor sizing of custom high-performance digital circuits with parametric yield considerations. In *Proc. DAC*, pages 781–786, 2010.

[82] M. Ketchen and M. Bhushan. Product-representative at-speed test structures for cmos characterization. *IBM JRD*, vol.50 no.4/5:451–468, 2006.

[83] Q. Liu and S. Sapatnekar. Synthesizing a representative critical path for post-silicon delay prediction. In *Proc. ISPD*, pages 183–190, 2009.

[84] M. Bushnell and V. Agrawal. *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits.* Kluwer Academic Publishers, 2000

[85] X. Li, *et.al*. Virtual probe: a statistically optimal framework for minimum-cost silicon characterization of nanoscale integrated circuits. In *Proc. ICCAD*, pages 433–440, 2009.

[86] R. Warner. *Applied Statistics: From Bivariate Through Multivariate Techniques.* Sage Publication, 2008.

[87] M. Tipping. Sparse bayesian learning and the relevance vector machine. *JMLR*, vol. 1:211–244, 2001.

[88] M. Tipping and A. Faul. Fast marginal likelihood maximisation for sparse bayesian models. In *Proc. AIS*, pages 3–6, 2003.

[89] S. Ji, *et.al*. *Bayesian compressive sensing. IEEE TSP*, vol. 56, no. 6:2346-2356, 2008.