UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of a doctoral thesis by

Hongliang Chang

and have found that it is complete and satisfactory in all aspects,
and that any and all revisions required by final
examining committee have been made.

Professor Sachin S. Sapatnekar

Name of the Faculty Advisor

Signature of the Faculty Advisor

Date

GRADUATE SCHOOL

Circuit Timing and Leakage Power Analysis Under Process Variations

A PhD Dissertation
Submitted to the Faculty of the Graduate School
of The University of Minnesota
By

Hongliang Chang

in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy in Computer and Electrical Engineering

Sachin S. Sapatnekar, Advisor

February 2006

Abstract

The task of analyzing circuit performance has become very challenging in sub-100nm technologies due to the effects of numerous variations, particularly process variations. Traditional multi-process-corner methods, which have been used to account for these variations for many years, cannot be scaled to these technologies as they become computationally expensive and over-pessimistic. Therefore, a statistical approach to performance analysis must be developed in order to predict circuit performance and yield more efficiently and accurately, and to guide circuit optimization.

In this thesis, we focus on the statistical analysis of timing and leakage power. Our approach considers both inter-die and intra-die process variations, as well as the effect of spatial correlations, and develops efficient techniques for handling this problem. We propose a grid-based model for spatial correlations that correctly takes spatial correlations into account, by partitioning the chip areas into grids. Within the same grid, perfect correlations are assumed for process parameters of the same type, and the correlation values between any pair of grids degrades with increasing distance.

A novel statistical static timing analysis (SSTA) method is first proposed. In this method, Gaussian probability distributions are used to approximate all process variations, and linear functions of process variables are used to represent the sensitivities of delays to all process variables. Prior to our work, most previous methods for SSTA were unable to account for the effect of spatial correlations in a computationally efficient way. We present a novel method that deals with the effects of spatial correlations on delays using the principal component analysis method to rotate the set of correlated variables into an independent set. A PERT-like (Program

Evaluation and Review Technique) traversal of the circuit graph is then performed by using *sum* and *max* operations, as in the static timing analysis, but with the *sum* and *max* operations defined analytically on Gaussian random variables. The use of principal components has several benefits: first, it allows SSTA to be carried out in linear-time; second, it permits dimension reduction to remove unimportant process parameters from the analysis; and third, it provides a technique to exactly handle the effects of structural correlations in the circuit due to reconvergent fanout. Since this method involves only analytical formulas for computations, the method has a very efficient computational complexity which is linear in the number of gates and interconnects, as well as the number of varying process parameters and grid partitions that are used to model spatial correlations. We show that the probability distribution of circuit delay and thus yield of timing can be predicted accurately by the proposed method by verifying with Monte Carlo simulations. We also demonstrate that spatial correlations must be considered appropriately to achieve correct results of timing analysis and yield.

Next, we present a general framework for SSTA that can incorporate non-Gaussian-distributed process variables and/or nonlinear delay functions of the variables, by extending the approach for handling Gaussian process variables and linear delay functions. To incorporate the non-Gaussian and nonlinear function variational sources, any delay is expressed in a generalized form by introducing a nonlinear non-Gaussian term to the linear function form in the technique for Gaussian sources of variation and linear delay functions. The *sum* and *max* operations are then extended to handle random variables in generalized forms. The method is fully compatible with the technique for handling Gaussian variation sources and linear delay functions, and preserves its computational efficiency in processing linear Gaussian process parameters. We show that the probability distributions of

circuit delays computed by the new technique are closer to the results of Monte Carlo simulations than a method that approximates non-Gaussian distributions with Gaussians and nonlinear functions with linear functions, especially at high timing yield levels.

Finally, we present an algorithm for analyzing full-chip leakage power under process variations, considering the spatial correlations of intra-die variation. The analysis is input-pattern-independent that computes the leakage power dissipation of each gate as a weighted sum of the leakage over all possible input vectors of the gate, with the weights as the probabilities of input vectors. The approach considers both the subthreshold and gate-tunneling leakage power, as well as their interactions. With process variations, each leakage component is approximated by a lognormal distribution, and the total chip leakage is computed as a sum of the correlated lognormals. Since the lognormals to be summed are large in number and have complicated correlation structures due to spatial correlations and the correlation between the two leakage mechanisms, we propose an efficient method to reduce the number of correlated lognormals for summation to a manageable number by identifying dominant states of leakage currents and taking advantage of the spatial correlation model and input states at the gates. An improved approach utilizing principal components of correlated process parameters is also proposed to further improve run-time efficiency. We show that the proposed methods are effective in predicting the probability distribution of the total chip leakage, and that ignoring spatial correlations can underestimate the standard deviation of full-chip leakage power.

Acknowledgments

First of all, I would like to express my deepest thanks and appreciation to my academic advisor, Prof. Sachin S. Sapatnekar, who guided and supported me throughout my PhD thesis research work. I sincerely appreciate his patient help, precious advice and consistent encouragement during my PhD study. I am especially thankful that he provided me with the opportunities to work on the most cutting-edge and important research topics in VLSI CAD area, and this has been beneficial to my research as well as my career. His loyalty to science, persistence in conquering difficulties in research, enthusiasm for work, as well as his responsive and generous personality have always been excellent examples for me to follow throughout my career and life.

I would also like to thank my PhD committee members, Professor George Karypis, Professor Kia Bazargan, Professor Gerald Sobelman and Professor Larry Kinney, for reviewing my thesis and giving valuable feedback and advice on my work.

I would like to convey my appreciation to Dr. Vladimir Zolotov and Dr. Chandu Visweswariah, who were my mentor and manager, respectively, during my internship at the IBM T.J. Watson Research Center. I would like to thank them for giving me an excellent opportunity and the experience of performing research works at the world's top research center. With their invaluable help and their suggestions, I was able to tackle a difficult research topic and implement the ideas in a real industrial tool, and this work constitutes a chapter of my PhD thesis. I would also like to thank Dr. Kerim Kalafala, Dr. Natesan Venkateswaran and the colleagues in IBM EinsTimer group, and Dr. Sambasivan Narayan in IBM Essex Junction, for their valuable support and collaboration throughout my internship. Overall, the

internship provided me with not only a broader view of research, but also an excellent opportunity to understand the gaps between academic research and industrial needs, and the scope for fruitful interactions between the two.

I am grateful to my colleagues in the VEDA lab at Minnesota, Brent Goplen, Shrirang Karandikar, Vidyasagar Nookala, Haifeng Qian, Rupesh Shelar, Jaskirat Singh, Yong Zhan, and Tianpei Zhang, for their warm and willing assistance at a personal level, and for their valuable advice and discussions that enriched my research.

I owe many thanks to my parents who never stopped helping and encouraging me overcoming difficulties during the pursuit of my PhD degree. I am grateful to my husband who has always been extremely understanding and supportive during my studies. I owe a huge debt to my little daughter who is only 20 months and was under my care for as little as 4 months. I would like to voice my special appreciations to my parents-in-law who took care of my daughter and supported me as I finished my PhD studies. My career and life are more meaningful because of the love and care that I have been privileged to receive from my whole family.

Dedication

To my parents, my husband and our 20-month old daughter Yueyi.

Contents

1 Introduction **1**

1.1 Process Variations Trends 2

1.2 Circuit Performance Analysis under Process Variations 5

1.3 Our Contributions 9

2 Modeling Process Variations **14**

2.1 Inter-die Variation 15

2.2 Intra-die Variation 16

2.3 Spatial Correlations 18

3 Statistical Static Timing Analysis **23**

3.1 Introduction 24

3.2 Problem formulation 27

3.3 SSTA Algorithm 31

3.3.1 Modeling Gate/Interconnect Delay PDFs 32

3.3.2 Orthogonal Transformation of Correlated Variables 37

3.3.3 PERT-like Traversal of SSTA 40
3.3.4 The Utility of Principal Components 45
3.4 Computational Complexity 48
3.5 Extending the Method to Handle Inter-die Variations, Spatially Uncorrelated Parameters, and Min-delay Computations 51
3.5.1 Inter-die Variations 51
3.5.2 Spatially Uncorrelated Parameters 52
3.5.3 Distribution of the Minimum of a Set of Gaussians 54
3.6 Experimental Results 54
3.7 Conclusion 62

4 Incorporating Non-Gaussian Distributed Process Parameters and Nonlinear Delay Functions 64

4.1 Introduction 65
4.2 Framework for Handling Gaussian and Linear Function Parameters 68
4.3 Framework for Handling Non-Gaussian and/or Non-linear Function Parameters 74
4.3.1 A Generalized Canonical Form for the Delay 75
4.3.2 The Computation of the *sum* Function 76
4.3.3 The Computation of the *max* Function 76
4.4 Implementation and Results 84
4.5 Conclusion 90

5 Prediction of Leakage Power Under Uncertainties **92**

5.1 Introduction 92

5.2 Problem Description 95

5.3 Computing the Distribution of Full-chip Leakage Current 97

5.3.1 Distribution of Subthreshold Leakage Current 98

5.3.2 Distribution of Gate Tunneling Leakage Current 100

5.3.3 Distribution of Full-Chip Leakage Current 101

5.4 An Improved Algorithm, Hybridized with the PCA-based Approach 114

5.4.1 PCA-based Method 114

5.4.2 Hybridization with the PCA-based Approach 117

5.5 Experimental Results 119

5.6 Conclusions 126

6 Conclusion **127**

List of Figures

2.1 Decomposition of process variations into inter-die variations and intra-die variations. 15

2.2 Grid model for spatial correlations. 20

2.3 A depth-2 quadtree model for spatial correlations proposed in [5]. . 22

3.1 Overall flow of our statistical timing analysis. 46

3.2 A comparison of *MinnSSTA* and *MC* methods (assuming fixed values of T_{ox} and N_a) for circuit s38417. The curve marked by the solid line denotes the results of *MinnSSTA*, while the plot marked by the starred lines denotes the results of *MC*. Note that the differences between the curves are exaggerated because of the high slopes and the fact that the scale does not include the origin, but the mean and the variance of the two are very close to each other, as are the delay points corresponding to 95% and higher timing yields. 56

3.3 A comparison of *MinnSSTA* and *MC* methods for circuit s38417, considering all sources of variation, some of which are spatially correlated and some of which are not. The curve marked by the solid line denotes the results of *MinnSSTA*, while the plot marked by the starred lines denotes the results of *MC*. 58

3.4 A comparison of SSTA with and without considering spatial correlations, under Monte Carlo analysis, for circuit s38417. The curve marked by the solid line denotes the case where spatial correlations are ignored, while the curve with the starred lines denotes the results of incorporating spatial correlations; this is identical to the curve in Figure 3.3. 61

4.1 Linear approximation of maximum of two canonical forms A and B, where $A = a_0 + a_1\Delta X$ and $B = b_0 + b_1\Delta X$. Since ΔX is Gaussian-distributed, only the range from $[-3\sigma, 3\sigma]$ is illustrated. The two-piece bold solid lines C=max(A,B) shows the exact maximum of A and B. The dotted line pointed to by $C_{appr} = c_0 + c_1\Delta X$ is the approximation of the max function. 74

4.2 Approximation of the maximum of two generalized canonical forms A and B, where $A = a_0 + f_A(\Delta X)$ and $B = b_0 + f_B(\Delta X)$. The figure shows the range of ΔX from $[-3\sigma, 3\sigma]$ as ΔX is Gaussian-distributed. The bold solid curve $C = max(A, B)$ is the exact maximum of A and B. The dotted line pointed to by $C_{appr} = c_0 + f_C(\Delta X)$ is the approximation of the max function. 78

4.3 Comparison of PDFs for maximum of two generalized canonical forms A and B. (a) shows the results on a non-Gaussian distribution, where $A = 10 + 0.5 \cdot \Delta X_1 + \Delta X_2 + 0.5 \cdot \Delta R_a$ and $B = 10 + \Delta X_1 + 0.5 \cdot \Delta X_2 + 0.5 \cdot \Delta R_b$, where all variational sources ΔX_i are lognormal and ΔR_a is Gaussian. (b) shows results on a nonlinear delay function, where $A = 10 + (\Delta X_1)^3/18 + (\Delta X_2)^3/9 + 0.5 \cdot \Delta R_a$ and $B = 10 + (\Delta X_1)^3/9 + (\Delta X_2)^3/18 + 0.5 \cdot \Delta R_b$, and all variational sources ΔX_i and ΔR_a are Gaussian. 86

4.4 Comparison of accuracy versus run-time for Design A, when different numbers of discretized points (5, 10 and 20 points) are used in the computation. 87

4.5 Comparison of PDFs of arrival time at a timing point for design A when different approaches are applied. All global sources of variations are lognormally distributed in the experiments. The proposed technique is shown by the bold solid curve, the original technique using Gaussian approximations by the thin solid curve, and the Monte Carlo results by the dotted bold curve. 88

4.6 Comparison of PDFs of arrival time at a timing point for design A when different approaches are applied. The delay functions at all circuit nodes are nonlinear (cubic) function of the variational sources in the experiments. The proposed technique is shown by the bold solid curve, the original technique using Gaussian approximations by the thin solid curve, and the Monte Carlo results by the dotted bold curve. 89

5.1 Three scenarios of combined I_{sub} and I_{gate} for a three-input NMOS transistor stack [42]. 107

5.2 Comparison of PDFs of average leakage currents using dominant states with that of full input vector states for a 3-input NAND gate, by Monte Carlo simulation with 3σ variations of L_{eff} and T_{ox} 20%. The solid curve shows the result when only dominant states are used, and the starred curve corresponds to simulation with all input vector states. 108

5.3 Distributions of the total leakage using the proposed basic method against Monte Carlo simulation method for circuit c7552. The solid line illustrates the result of the proposed basic method, while the starred line shows the Monte Carlo simulation results. 120

5.4 Scatter plot of full-chip leakage considering spatial correlation for circuit c432 122

5.5 Scatter plot of full-chip leakage ignoring spatial correlation for circuit c432 123

List of Tables

1.1 Trends of total process variabilities from 250nm to 70nm technologies (compiled from [56]). 3

3.1 Parameters used in the experiments. 55

3.2 Comparison results assuming fixed values of T_{ox} and N_a. 57

3.3 Comparison results of the proposed method and Monte Carlo simulation method. 59

3.4 Statistics of ratio of standard deviation of accurate value $\sigma_{d_{max}}$ to s_0 of the linear expression. 59

3.5 Experimental results on a binary tree circuit of depth-10. 60

3.6 Comparison of timing analysis with and without spatial correlations. 60

3.7 Comparison of 99% and 1% confidence point. 62

4.1 Comparison of the run-time as the number of non-Gaussian distributed sources, and the number discretization points, are varied. . . . 87

4.2 Comparison of run-time versus the numbers of non-Gaussian process parameters for various sizes of industrial designs. 90

5.1 Comparison of the proposed basic method with Monte Carlo simulation. 119

5.2 Comparison of leakage by varying L_{eff} and T_{ox} independently . . . 124

5.3 Run-time comparison of the proposed basic, PCA-based, and improved methods for the ISCAS85 benchmarks 125

5.4 Run-time comparison of the proposed basic, PCA-based, and improved methods for the ISCAS89 benchmarks 125

Chapter 1

Introduction

As integrated circuits have continued to scale down further, the manufacturing process has become less predictable. After manufacturing, the process parameters and the dimensions of the fabricated devices and wires can be very different from their designed values. For example, an oxide thickness that is nominally 25Å may turn out to be, after manufacturing, thicker than the designed value at 27Å , or thinner at 24Å. Such variations in the process parameters can induce substantial fluctuations in the performance of VLSI circuits. Performance parameters such as timing and power may be affected either positively or negatively, and the net result of this may be a low manufacturing yield, as a majority of the manufactured dies fail to meet the performance specifications. Therefore, manufacturing process induced variation, or *process variation*, is an important consideration in VLSI circuit design and yield analysis. Other sources of variations can be categorized as *environmental variations*, which are mainly caused by changes in circuit operating conditions, such as fluctuations in the temperature and supply voltage. Generally speaking, process variations are amenable to being handled by probabilistic methods, since

optimizing a probability density function of the yield can ensure that a certain fraction of all chips work correctly, and the remaining chips may be discarded after manufacturing, and the others are guaranteed to work. On the other hand, environmental variations that affect a chip during normal operation are typically worst-cased. This is because it is not enough, for example, to guarantee that a chip works correctly 99% of the time: it must always work correctly under the worst-case operating conditions.

In this thesis, we will mainly focus on the effects of process variations, and propose statistical techniques to analyze circuit performance under these variations. In this chapter, we will first introduce some background on process variations and process-variation-aware circuit performance analysis, and then present our research goals and the contributions of the thesis.

1.1 Process Variations Trends

Process variations have been a long-standing problem, but several recent trends have made the problem more serious in current and future technologies:

- In [56], the total variabilities of nanometer-scale technology process parameters were observed and forecasted, based on the International Technology Roadmap for Semiconductors (ITRS) [7] projections and on insight from IBM technologies. Table 1.1 [56] lists, for five technologies ranging from 250nm to 70nm, the mean values and 3σ values of different process parameters, the effective transistor gate length (L_{eff}), the transistor gate oxide thickness (T_{ox}), the supply voltage (V_{dd}), the transistor threshold voltage (V_{th}), the interconnect width (W_{int}) and thickness (T_{int}), and the metal resistivity (ρ). It is

Table 1.1: Trends of total process variabilities from 250nm to 70nm technologies (compiled from [56]).

Technology	1997		1999		2002		2005		2006	
Parameters	mean	3σ	mean	3σ	mean	3σ	mean	3σ	mean	3σ
L_{eff} (nm)	250	80	180	60	130	45	100	40	70	33
T_{ox} (nm)	5	0.4	4.5	0.36	4	0.39	3.5	0.42	3	0.48
$V_{dd}(V)$	2.5	0.25	1.8	0.18	1.5	0.15	1.2	0.12	0.9	0.09
$V_{th}(V)$	0.5	0.05	0.45	0.045	0.4	0.04	0.35	0.04	0.3	0.04
$W_{int}(\mu m)$	0.8	0.2	0.65	0.17	0.5	0.14	0.4	0.12	0.3	0.1
$T_{int}(\mu m)$	1.2	0.3	1	0.3	0.9	0.27	0.8	0.27	0.7	0.25
$\rho(\Omega m)$	45	10	50	12	55	19	60	19	75	25

clear that the magnitudes of the total process variabilities, as measured by 3σ/mean increase with technology scaling.

- The number of process parameters that show significant variations (i.e., variations that significantly impact circuit performance) increases with technology scaling. In previous technology generations, gates and transistors were primarily subject to substantial variabilities during a typical manufacturing process, but interconnects are now also starting to show large levels of variability, especially as the number of metal layers becomes higher, and the mismatches of metals may be independent in each layer [79].

- Process variations can be decomposed into inter-die variation and intra-die variations. Inter-die variations correspond to the variability of process parameters from one die to another, while intra-die variations are the variations inside a single die. It used to be the case that inter-die variations dominated intra-die variations, and it was sufficient to consider only the effect of inter-die variation on circuit performance, ignoring that of intra-die variation.

However, in the sub-100nm regime, not only do the total process variations increase, but the proportion of the total variability that is attributable to intra-die variations has also increased to a nonnegligible level that can significantly affect the variability of performance parameters on a chip. For instance, the percentage of L_{eff} variations that is within-die increases from 40% to 65% as the technology goes from 250nm to 70nm [56]. As a consequence, simple corner-based worst-casing techniques that have been used for many years, which assume that all process parameters are at their minimum or maximum values all over the chip, are too pessimistic. In principle, this may be overcome by more sophisticated, region-based cornering, but the number of corners to be explored under this scheme grows exponentially with the number of regions, making such an approach impractical [67].

These trends in process variation result mainly from aggressive technology scaling. Specifically, in nanometer technologies, the minimum feature sizes have approached the resolution limits of photolithography systems and etch, and modern CMOS processes are forced to operate in a sub-wavelength lithography regime. For example, 193nm lasers are currently used to fabricate devices with dimensions of 90nm or less [7, 32]. While clever techniques for resolution enhancement (RET) such as optical proximity correction (OPC) [32, 33] and phase shifting masking (PSM) [23, 48] can overcome some of these effects, the level of control over feature sizes and the so-called critical dimensions (CDs) is reduced. While some of these variations can be modeled deterministically, there is a remnant that is either truly random or is modeled as being random due to the difficulty of a full deterministic analysis. This results in a proportional increase in the total variability, as well as the intra-die variability of process parameters that results in mismatches of process parameters within a single die. Device parameters also become more variable: for

instance, as devices grow smaller and the number of dopant atoms per transistor is in the range of 10 to 100, the level of control in the uniformity and number of these atoms decreases [8, 14], and this process parameter affects device parameters such as the threshold voltage, and eventually, the switching speed of the gate that the transistor lies in.

In addition, as the size of a wafer increases, in terms of multiples of the feature size[1], so does the variability of manufacturing process variation. In particular, some of the process variations that were originally at the inter-die, within-wafer level are projected onto the intra-die level, thus further increasing intra-die variations.

In summary, in current and future technologies, with the number and magnitude of process variations both increasing, process variations have become an increasing concern. Besides inter-die variation, intra-die variation also should be taken into account appropriately in order to correctly predict the effects of process variations on circuit performance.

1.2 Circuit Performance Analysis under Process Variations

Since process variations can significantly affect circuit performance parameters such as timing and power, it is important to analyze the relation between these in order to predict their impact on circuit performance, for parametric yield prediction as well as variation-aware circuit design and optimization. We will now overview several classes of analysis techniques.

[1]Note that even if the wafer size remains constant, reductions in the feature size result in an increase in the wafer size, measured in units of the feature size.

Multi-Corner-Based Methodology

In general, the value of a process parameter after manufacturing falls into a bounded range from a minimum to a maximum value. A process corner corresponds to a set of values of process variables in the parameter space where each parameter in the space takes either the minimum or maximum value. A worst-case corner is defined as the corner where the process parameters take their extreme values that can result in the worst behavior for a typical circuit. Traditional circuit analysis deals with process variations by predicting the worst-case circuit behavior evaluated at worst-case corners. Unfortunately, with the number and magnitude of process variables increasing, checking a small set of worst-case corner could be risky if it may not cover the region sufficiently, or excessively conservative, if the corners are chosen to embody a pessimistic worst-case [79, 84]. Therefore, a multi-corner-based method, which predicts the circuit behavior by analyzing the circuit at all enumerative corners, has to be used to evaluate worst-case behavior. However, the multi-process corner based methodology also suffers from the following disadvantages.

- First, the method is too computationally intensive: on the one hand, as the number of varying process parameters increases, the number of process corners to enumerate, which grows exponential with the number of process variables, grows too high; on the other hand, under intra-die variation, the process parameter values of devices [wires] in the same chip can vary differently, and therefore, the number of process corners required must also consider region-based analysis (alluded to in Section 1.1), which worsens the exponential behavior.

- Second, the approach is too conservative and pessimistic in that the process corner corresponding to the worst-case performance may have a very low

probability of occurrence, which results in a over-pessimistic results. As an example, suppose there are two independent sources of variations p_1 and p_2 with Gaussian distribution $N(\mu_1, \sigma_1)$ and $N(\mu_2, \sigma_2)$, respectively. Then, using the corner-based method, the worst-case could be found by inspecting the corners are at $(p_1, p_2) = (\mu_1 \pm 3\sigma_1, \mu_2 \pm 3\sigma_2)$. However, the probability of each of the (p_1, p_2) corners is as low as 1.96×10^{-5}, significantly less than at the 3σ point. This pessimism is liable to become especially severe as the number of varying process parameters grows higher. Amending this procedure so that the corners correspond to 3σ points does not help either: fundamentally, the problem here is that the level sets of the Gaussian are ellipsoids, and worst-casing over the corners of a multidimensional box is doomed to failure.

Monte Carlo Simulation Approach

The effects of process variations on circuit performance can also be predicted by Monte Carlo simulation method [37, 61, 66]. The approach is an iterative process where each iteration consists of two basic steps, sampling and simulation. In each sampling step, a set of sampled values of process parameters are generated according to the distribution of process parameter variations, or samples as delay/power for all circuit nodes generated according to their distributions. The simulation step then simply runs a circuit/timing/power simulation, using the generated sample values. The Monte Carlo method is very accurate in predicting the distribution of circuit performance. However, for an integrated circuit, the number of iterations required for convergence is generally greater than $10,000$. Although smart techniques can be used to reduce the sampling size, it is still a large number so as to achieve desirable accuracy of simulation result. Therefore the approach is highly computationally expensive, and is not practical even for medium size circuits.

Statistical Analysis Method

Statistical performance analysis methods provide a good possibility for analyzing circuit performance with good accuracy and efficient run-time. These approaches directly exploit the statistical information of the process parameters and utilize efficient stochastic techniques [61] to determine the probability distribution of the circuit performance. In these methods, instead of using fixed values of process parameters (as is done in each multi-corner analysis), random variables are used to model the uncertainty of process parameters. In timing analysis, the delays of gates and interconnects and arrival times at intermediate nodes are all random variables. Therefore, unlike conventional deterministic static timing analysis (STA) which computes timing based on deterministic values, the statistical static timing analysis (SSTA) method *stochastically* computes delays and arrival times on a set of random variables. Therefore, probabilistic characteristics, such as the probability density function (PDF) of circuit timing, can be obtained and yield of timing can also be predicted from the computation. Similarly, for statistical leakage power analysis, the leakage power of each gate is modeled as a random variable and the result of computation is probability distribution and yield of full-chip leakage.

It is worth mentioning that under process variations, circuit optimization techniques should be also adapted to be capable of considering the effects the process variations. Therefore, the importance of analyzing circuit performance under process variation is not limited to yield prediction, but also for variation-aware circuit design and optimization. Multiple-process corner based methods are too pessimistic, and may result in over-constrained circuit optimization. Therefore, although more computational effort goes into reoptimizing the circuit to meet the worst-case performance requirement over all corners, this does not significantly contribute to im-

proving the yield of circuit performance. The alternative of using accurate Monte Carlo methods suffers from a different drawback: the expensive run-time prohibits these methods from being used within a circuit optimization algorithm. In contrast to these, statistical methods for circuit performance analysis are computationally efficient and can achieve good accuracies, and therefore, have the potential to be practically be integrated into various steps of the design flows, such as technology mapping, synthesis, and physical design.

1.3 Our Contributions

In modern chip design, circuit performance is greatly constrained by timing and power. In nanometer-scale technologies, leakage power has become a major component of total chip power dissipation, and it is highly sensitive to manufacturing variations due to its exponential dependency on some process parameters. Therefore, in this thesis, we will focus on the analysis of timing and leakage power, and propose efficient statistical performance analysis methods for timing and leakage power dissipation under the effect of inter-die and intra-die variations. As intra-die variations exhibit spatial correlation, i.e., devices [wires] spatially located close to each other tend to experience more similar variations than those placed far away, the effect of spatial correlations are also considered in the analysis using a model proposed in Chapter 2. The major contributions of the thesis are:

- *Model for spatial correlation:* In order to analyze the impact of intra-die variations, spatial correlations in intra-die variation of process parameters must be modeled correctly. In this thesis, we propose a model for spatial correlation by partitioning the die region into $nrow \times ncol = n$ grids. Since devices

[wires] close to each other are more likely to have more similar characteristics than those placed far away, we assume perfect correlations among the devices [wires] in the same grid, high correlations among those in close grids and low or zero correlations in far-away grids. Under this model, if the number of grids partitioned is n, then for each process parameter, a covariance matrix of size $n \times n$ can then be used to represent the spatial correlation of the process parameter among the grids. The advantage of this model is its flexibility in representing spatial correlation by a covariance matrix, where the covariance matrix could be determined from data extracted from manufactured wafers or a test structure methodology can be used to support the evaluation of process parameter variations and spatial correlations as developed in [13]. In this thesis, the model of spatial correlation is used for both timing and leakage power analysis.

- *Statistical static timing analysis:* We propose an algorithm for statistical static timing analysis that computes the distribution of circuit delay while considering inter-die and intra-die variations as well as the effect of spatial correlations. The method approximates probability distributions of all process variations by Gaussians and models the circuit delay as a Gaussian random variable approximated as a maximum of correlated multivariate normal distributions, considering both gate and wire delay variations in the circuit. In order to manipulate the complexities brought about by the correlation structure, the principal component analysis technique is employed to transform the sets of correlated parameters into sets of uncorrelated ones. The statistical timing computation is then performed with a PERT-like circuit graph traversal with computational complexity $O(p \times n \times (N_g + N_I))$, where N_g and N_I are the number of gates and interconnects, respectively, p is the number of

varying process parameters and n is the number of grid squares in the spatial correlation model. Therefore, the cost is, at worst, $p \times n$ times the cost of a deterministic STA. We believe that this is the first method that can fully handle spatially correlated distributions under reasonably general assumptions, with a complexity that is comparable to traditional deterministic STA. This framework can also be applied to analysis, such as computing minimum delay distributions for short-path analysis (to check for hold time violations), for required arrival time (RAT) analysis, etc., by extending the same framework of maximum of delays to compute the distribution of minimum of delays. This work was published in [17].

- *Statistical timing analysis with non-Gaussian distributed process parameters and nonlinear delay functions:* Statistical timing analysis methods which assume process variations to take the form of linear functions of Gaussians, can be very run-time efficient. However, as delay shows nonlinear sensitivities to some process parameters, and some process variations, which show non-Gaussian distributions and cannot be well approximated with Gaussians, it is essential to develop an SSTA technique that can handle non-Gaussian process parameters and nonlinear delay functions to achieve desirable accuracy. For this purpose, we first propose three general frameworks for statistical timing analysis that can be used for handling Gaussian process variations and linear delay functions. Based on one of the framework, we present a novel and efficient technique for handling arbitrary distributed process variations and nonlinear delay functions in parameterized block-based statistical timing analysis. The method is fully compatible with the previous SSTA technique for dealing with Gaussian process parameters and linear delay functions. and the computational efficiency in processing such types of process variations

is preserved. We show that developing SSTA technique that is capable of incorporating non-Gaussian sources of process variations and/or nonlinear delay functions is important to correctly predict the circuit timing, as well as for validating the approximations of process parameters as Gaussians and models of delay functions as linear ones, in order to selectively apply crucial process parameters as non-Gaussian distributed or with nonlinear functions to improve computational efficiency. This work was published in [19].

- *Statistical leakage power analysis:* We propose an input-pattern-independent method for predicting, under process variations, the probability distribution of the total circuit leakage power, considering subthreshold and gate-tunneling leakage powers and their interactions. Spatial correlations of intra-die variations and the correlation between these two leakage mechanisms, which were ignored in most of the previous works, are also considered. In the method, each leakage component is approximated by a lognormal distribution, and the total chip leakage is computed as a sum of the correlated lognormals. However, the lognormals to be summed are large in number and have complicated correlation structures, due to spatial correlations as well as correlations between the subthreshold leakage and gate leakage. To enhance the efficiency of the algorithm, the number of correlated lognormals for summation is reduced to a manageable number by applying dominant states for leakage currents, taking advantage of the spatial correlation model and input states at the gates. An improved algorithm upon this approach by utilizing principal components of spatially correlated process parameters is also proposed. We show that spatial correlations must be considered in order to correctly estimate the full-chip leakage. An early version of this work was published in [18].

The thesis is organized as follows: Chapter 2 introduces the model for process variations and proposes a grid-based model to capture the effect of spatial correlations in intra-die variations. A statistical timing analysis technique for handling Gaussian distributed process variables and linear delay functions will be presented in Chapter 3. The techniques for incorporating arbitrary distributed process variations and arbitrary delay functions are given in Chapter 4, and then a method for statistical leakage power prediction is proposed in Chapter 5. The last chapter concludes the thesis.

Chapter 2

Modeling Process Variations

Process variations can be separated into the following categories as illustrated in Figure 2.1: *inter-die variations* are the variations from lot to lot, wafer to wafer or die to die, while *intra-die variations* correspond to variability within a single die. Inter-die variations affect all the devices on same die in the same way, e.g., making the transistor gate lengths of devices on the same chip all larger or all smaller than their nominal values, while intra-die variations may affect different devices on the same chip in different ways, e.g., causing some transistors to have smaller-than-nominal gate lengths and other transistors have larger-than-nominal gate lengths. Therefore, the total variation of a process parameter p for some device [wire] in a die can be modeled as [4, 73, 81]:

$$\Delta p_{total} = p - p_0 = \Delta p_{inter} + \Delta p_{intra} \tag{2.1}$$

where p_0 is the nominal value of process parameter, Δp_{inter} and Δp_{intra} are random variables for inter-die and the intra-die variation, respectively. Since inter-die variation has a global effect within a single chip, a single random variable Δp_{inter} is used for all devices [wires] on the same chip.

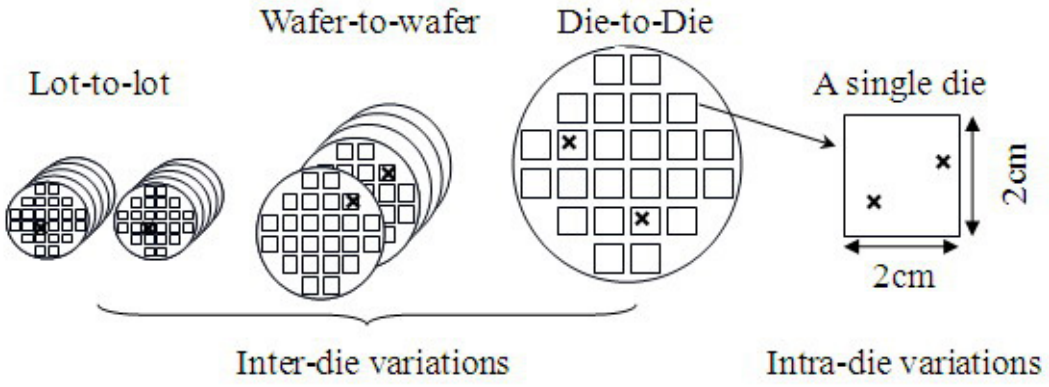


Figure 2.1: Decomposition of process variations into inter-die variations and intra-die variations.

In this chapter, we will introduce models for the inter-die and intra-die variations and for spatial correlations in intra-die variation. The statistical timing analysis and leakage power analysis techniques in later chapters will be based on the proposed models here.

2.1 Inter-die Variation

Inter-die variation refers to the variation of some parametric values across nominally identical manufactured dies spatially located on the same wafer (die-to-die), or different wafer (wafer-to-wafer) or different lots (lot-to-lot). Therefore, inter-die variation has a global effect on the process parameters of all devices/wires on the same chip and is accounted for in circuit design as a shift in the mean of some parameter value across any one chip [12]. Inter-die variation can further be decomposed into systematic and random components, where the systematic component is a deterministic nonrandom term often caused by effects, such as process gradients

across the wafer, while the random term corresponds to unexplained or unmodeled[1] random variations. Although it is possible to deterministically characterize some parts of the systematic component based on some knowledge of the manufacturing process (e.g., the systematic trends may follow a "bowl" shape across the wafer), the unknown or random part is usually modeled by a statistical distribution. After incorporating variations due to the deterministic variations, the total random inter-die variation for a process parameter can be modeled by a Gaussian probability distribution [12]:

$$\Delta p_{inter} \sim N(0, \sigma_{p,inter}) \tag{2.2}$$

where $\sigma_{p,inter}$ is the standard deviation of the inter-die variation.

2.2 Intra-die Variation

Intra-die variation refers to the variation of some parametric values across identically designed devices [wires] spatially located on the same die. It can arise from a number of manufacturing sources, and two sources are of particular importance: the projection of variation trends to the die level from the wafer level, and the interaction between the fabrication process and circuit local layout pattern [12]. The former results in spatially-location dependent variations of process parameters across the die, while the latter proximity and layout dependent. Intra-die variation can also be divided into systematic and random variations. The systematic vari-

[1]In practice, random variations are not necessarily always truly random. Sometimes, one may choose to model a deterministic variation as random, either because the deterministic model is too difficult to develop, or because the cost of performing an analysis with the deterministic model is too high.

ations are those may be modeled deterministically, and the random variations are the remaining unmodeled variations.

A number of models for intra-die variations have been developed in literature. The Pelgrom model [62] characterizes the mismatch of two equally sized transistors by a global systematic component and a local random component, with the global component as a Gaussian random variable whose variance is inversely proportional to the transistor geometry area and the local component as a Gaussian whose variance decreases with the distance between two transistors. The model of intra-die variation of gate CD is proposed in [59, 60]. It first generates the spatial CD maps for all gates categories depending on layout patterns and proximities. Then, from the circuit layout and spatial CD maps, the intra-die variation of each gate CD is modeled with a systematic component which is a category and location dependent deterministic function, and a random residual which is a Gaussian random variable whose variance is proximity dependent. In this thesis, we use the model proposed in [47] described as the following. According to the sources of variation, the intra-die variation Δp_{intra} for a process parameter p of some device [wire] is decomposed into three components, a systematic global component δ_{global}, a systematic local component δ_{local} and a random component ϵ [47]:

$$\Delta p_{intra} = \delta_{global} + \delta_{local} + \epsilon \tag{2.3}$$

The global component, δ_{global}, corresponds to the slowly and smoothly varying global systematic trend spatially across the die. Across a die, it can be modeled by a slanted plane and expressed as a simple linear function of position [12, 28, 47, 57]:

$$\delta_{global}(x, y) = \delta_0 + \delta_x x + \delta_y y \tag{2.4}$$

where (x, y) is the position of device [wire] within the die, δ_x and δ_y are the gradients

of the parameters, indicating the spatial variations of parameters along the x and y direction across the die, respectively.

The local component, δ_{local}, corresponds to the systematic variation caused by the interactions between the fabrication process and die pattern, and is thus proximity-dependent and layout-specific. It can be modeled deterministically from the extracted chip layout pattern and precharacterized spatial maps of process parameters as in [59].

The random residue, ϵ, stands for the remaining uncertainties or unmodeled intra-die variation which is usually modeled as a Gaussian random variable. As will be explained in Section 2.3, since the global systematic process variation can create spatially correlated structure of process variations, the vector of all random components across the chip has a correlated multivariate normal distribution:

$$\vec{\epsilon} \sim N(0, \Sigma) \tag{2.5}$$

where Σ is the covariance matrix [61] of process parameters. The detailed model for this covariance matrix will be described in the next section. For spatially uncorrelated parameters, Σ becomes a diagonal matrix where the entries represent variances. If the variances of the process parameters described by this matrix are assumed to be uniform across the chip, then Σ is a multiple of the identity matrix. On the other hand, in the presence of spatial correlations, Σ is nondiagonal, and captures the correlation structure of the process variations across the die.

2.3 Spatial Correlations

It is observed that, due to the slowly varying global process and operation conditions, the global systematic variations often have a relatively low spatial frequency

and are smooth across the wafer and die [12, 47]. As a result of the smooth spatial variation of process parameter, devices [wires] located close to each other are more likely to have the similar characteristics than those placed far away. Statistically, this behavior translates into the spatial correlation of process parameters.

The model of spatial correlation can be based on the separation distance directly as in [29, 81]. In this thesis, to model the intra-die spatial correlations, we propose a grid-based model by partitioning the die region into $nrow \times ncol = n$ grids. Since devices [wires] close to each other are more likely to have more similar characteristics than those placed far away, we assume perfect correlations among the devices [wires] in the same grid, high correlations among those in close grids and low or zero correlations in far-away grids. For example, in Figure 2.2, gates a and b (whose sizes are shown to be exaggeratedly large) are located in the same grid square, and it is assumed that their parameter variations (such as the variations of their gate length), are always identical. Gates a and c lie in neighboring grids, and their parameter variations are not identical but highly correlated due to their spatial proximity (for example, when gate a has a larger than nominal gate length, it is highly probable that gate c will have a larger than nominal gate length, and less probable that it will have a smaller than nominal gate length). On the other hand, gates a and d are far away from each other, their parameters may be uncorrelated, (e.g., when gate a has a larger than nominal gate length, the gate length for d may be either larger or smaller than nominal).

In this model, it is assumed that nonzero correlations may exist only among the same type of process parameters in different grids, and there is no correlation between different types of process parameters[2]. For example, the values of transistor

[2]In case the assumption is not strictly true [73], the model can be adapted to handle correlations between process parameters of different types, either by decomposing the correlated parameters

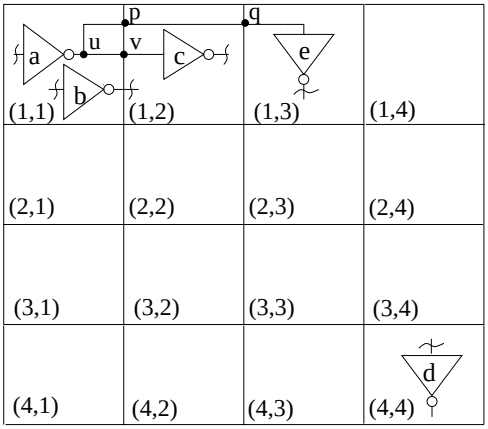


Figure 2.2: Grid model for spatial correlations.

gate lengths for transistors in a grid are correlated with those in nearby grids, but are uncorrelated with other parameters such as interconnect metal width or gate oxide thickness in any grid. Therefore, the parametric variation for a spatially correlated parameter in a single grid at location (x, y) can be modeled using a single random variable $p(x, y)$. In total, this representation requires n random variables, each representing the value of a parameter in one of the n grids, and a covariance matrix of size $n \times n$ representing the spatial correlations among the grids. The covariance matrix could be determined from data extracted from manufactured wafers[3]. Test structure methodologies can be developed to support the evaluation of process parameter variations as in [13, 29, 59]. For example, to construct the covariance matrix for gate critical dimensions, the process parameter values across the die should be measured and extracted [29]. The number and sizes of grid regions divided can then be determined by iteratively computing the process parameter

into an uncorrelated set using an orthogonal transformation via the principal component analysis technique, or by constructing a covariance matrix for all correlated parameters.

[3]Different semiconductor foundries could have different data for process variations and, correspondingly, different spatial correlation structures.

covariance over the separation distance of devices [29] using the measured values and refining the grid size until it converges.

Another grid-based model, the quadtree spatial correlation model, was proposed in [5]. The chip area is first divided into several regions using multi-level quadtree partitioning: at the l^{th} level, the area is partitioned into $2^l \times 2^l$ squares, with the top-most zero level has one region covering the whole die, while the bottom-most k^{th} level has $2^k \times 2^k$ regions if the quadtree has a depth of k. Then, an independent random variable, $\Delta p_{l,r}$ is associated with each square region (l, r) to represent the process variation of parameter p in the r^{th} region at level l. The total variation of parameter p in region (i, j) is modeled as the sum of the independent random variables of all squares at all levels that cover this region:

$$\Delta p_{i,j} = \sum_{0<l<k,\ all\ regions\ (l,r)\ that\ cover\ (i,j)} \Delta p_{l,r} \tag{2.6}$$

In this way, if a quadtree model of depth k is used, a process parametric variation in any region is modeled as a sum of $k+1$ independent random variables, and the total number of such independent random variables is $2^0 + 2^1 + 2^2 + \cdots + 2^k = 2^{k+1} - 1$.

As an example, a quadtree model of depth two is illustrated in Figure 2.3 with the top-most level has one region and bottom-most level $2^2 \times 2^2$ regions. The variation of process parameter p in region $(2, 1)$ is modeled as:

$$\Delta p_{2,1} = \Delta p_{0,1} + \Delta p_{1,1} + \Delta p_{2,1} \tag{2.7}$$

This model can be seen as a special case of our proposed model as shown in Figure 2.2. Our proposed model is more general than the model used in [5], since it is purely based on neighborhood. For example, consider again the case in Figure 2.2, by our model, the parameter in grid $(1, 2)$ has equal correlations with that in grid $(1, 1)$ and $(1, 3)$. While by the model of [5], it will have higher correlation with

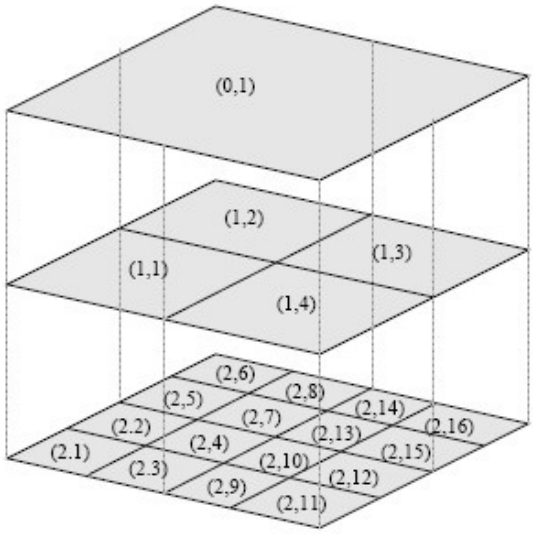


Figure 2.3: A depth-2 quadtree model for spatial correlations proposed in [5].

grid $(1, 1)$ than grid $(1, 3)$, i.e., the correlations are uneven at the two neighbors of grid $(1, 2)$.

It should be noted that not all process parameters exhibit spatial correlation. For example, in manufacturing, due to effects such as random dopant fluctuations, the intra-die variations of some parameters such as T_{ox} and N_a are truly uncorrelated from transistor to transistor.

Chapter 3

Statistical Static Timing Analysis

In this chapter, we present an efficient statistical timing analysis algorithm that predicts the probability distribution of the circuit delay, considering both inter-die and intra-die variations, while accounting for the effects of spatial correlations in intra-die parameter variations. The procedure uses a first-order Taylor series expansion to approximate the gate and interconnect delays. Next, principal component analysis techniques are employed to transform the set of correlated parameters into an uncorrelated set. The statistical timing computation is then easily performed with a PERT-like circuit graph traversal using statistical *sum* and *max* functions. The run-time of the algorithm is linear in the number of gates and interconnects, as well as the number of varying process parameters and grid partitions that are used to model spatial correlations.

3.1 Introduction

As introduced in Chapter 1, conventional static timing analysis techniques handle the problem of variability by analyzing a circuit at multiple process corners. However, it is generally accepted that such an approach is inadequate, since the complexity of the variations in the performance space implies that if a small number of process corners is to be chosen, these corners must be very conservative/pessimistic as well as risky. For true accuracy, this can be overcome by using a larger number of process corners, but then the number of corners that must be considered for an accurate modeling will be too large for computational efficiency, and the method is also over-pessimistic as explained in Chapter 1.

The limitations of traditional static timing analysis techniques lie in their deterministic nature. An alternative approach that overcomes these problems is statistical static timing analysis (SSTA), which treats delays not as fixed numbers, but as probability density functions, taking the statistical distribution of parametric variations into consideration while analyzing the circuit.

In the literature, the statistical timing analysis approaches can be classified into continuous and discrete methods. Continuous methods [5, 9, 58, 78] use analytical approaches to find closed-form expressions for the PDF of the circuit delay. For simplicity, these methods often assume a normal distribution for the gate delay, but even so, finding the closed-from expression of the circuit distribution is still not an easy task. Discrete methods [6, 45, 53] are not limited to normal distributions, and can discretize any arbitrary delay distribution as a set of tuples, each corresponding to a discrete delay and its probability. The discrete probabilities are propagated through the circuit to find a discrete PDF for the circuit delay. However, this method is liable to suffer from the problem of having to propagate an exponential

number of discrete point probabilities. In [27], an efficient method was proposed by modeling arrival times as cumulative density functions and delays as probability density functions and by defining operations of *sum* and *max* on these functions. Alternatively, instead of finding the distribution of circuit delay directly, several attempts have been made to find upper and lower bounds for the circuit delay distribution [6, 11, 58].

Statistical timing analyzers can also be categorized into path-based and block-based techniques. A path-based SSTA method, such as the works in [5, 30, 46, 58], enumerates all signal propagation paths or selective critical paths, finds the probability distribution of each individual path delay and then computes PDF of circuit delay by integration over all paths in space. Although the computation of probability distribution for a single path is not difficult for arbitrarily distributed process parameter or arbitrary delay functions, the integration over all paths requires the joint probability density function of all paths and thus the correlation information among all paths must be computed which is of extremely high complexity. In addition, path-based methods suffer from the requirement that they may require the enumeration of paths: the number of paths can be exponential with respect to the circuit size. Therefore, such methods are not realistic for practical usage. A block-based SSTA method, such as [4, 9, 11, 27, 39, 45, 53, 78, 80], models delays of gates [wires] as random variables, and propagates/computes signal arrival times using *sum* and *max* operations similarly to propagating arrival times by a deterministic STA. Since block-based methods have linear run-times with respect to the circuit size and are good for incremental modes of operation, they are of the most interest.

Although many prior works have dealt with inter-die and intra-die variations, most of them have ignored intra-die spatial correlations by simply assuming zero correlations among devices on the chip [6,9,11,16,25,27,37,44–46,53]. The difficulty

in considering spatial correlations between parameters is that it always results in complicated path correlation structures that are hard to deal with. Prior to our work of this chapter, very few studies have taken spatial correlations into consideration. The authors of [78] consider correlation between delays among the transistors inside a single gate (but not correlations between gates). The work in [46] uses a Monte Carlo sampling-based framework to analyze circuit timing on a set of selected sensitizable true paths. Another method in [58] computes path correlations on the basis of pair-wise gate delay covariances and used an analytic method to derive lower and upper bounds of circuit delay. The statistical timing analyzer in [20] takes into account capacitive coupling and intra-die process variation to estimate the worst case delay of critical path. Two parameter space techniques, namely, the parallelepiped method and the ellipsoid method, and a performance-space procedure, the binding probability method, were proposed in [36] to find either bounds or the exact distribution of the minimum slack of a selected set of paths. The approach in [5] proposes a model for spatial correlation and a method of statistical timing analysis to compute the delay distribution of a specific critical path. However, the probability distribution for a single critical path may not be a good predictor of the distribution of the circuit delay (which is the maximum of all path delays), as will be explained in Section 3.2. Moreover, the method may be computationally expensive when the number of critical paths is too large. In [4], the authors further extend their work in [5, 6] to compute an upper bound on the distribution of exact circuit delay.

In this chapter, we will propose a block-based SSTA method that computes the distribution of circuit delay while considering correlations due to path reconvergence as well as spatial correlations. We will model the circuit delay as a correlated multivariate normal distribution, considering both gate and wire delay variations.

In order to manipulate the complicated correlation structure, the principal component analysis technique is employed to transform the sets of correlated parameters into sets of uncorrelated ones. The statistical timing computation is then performed with a PERT-like circuit graph traversal. The complexity of the algorithm is $O(p \times n \times (N_g + N_I))$, which is linear in the number of gates N_g and interconnects N_I, and also linear in p, the number of spatially correlated random variables, and the number of grid squares, n, that are used to model variational regions. In other words, the cost is, at worst, $p \times n$ times the cost of a deterministic static timing analysis. We believe that this is the first method that can fully handle spatially correlated distributions under reasonably general assumptions, with a complexity that is comparable to traditional deterministic static timing analysis. This work can also be extended, using the same framework of maximum of delays (Section 3.3.3), to find the distribution of minimum of delays which can be applied to the analysis of the worst-case clock skew, required arrival time (RAT) analysis, etc.

The remainder of this chapter is organized as follows. Section 3.2 formally formulates the problem to be solved in this work. The algorithm is presented in Section 3.3 and its run time complexity analysis is given in the following section. The extension to handle inter-die variation and spatially uncorrelated intra-die components is introduced in Section 3.5, and the extension for short path analysis is presented in Section 3.5. Finally, a list of experimental results and their analysis are shown in Section 3.6.

3.2 Problem formulation

Under process variations, parameter values, such as the gate length, the gate width, the metal line width and the metal line height, are random variables. Some of these

variations, such as across-chip linewidth variations (ACLV) which are mainly caused by proximity and local effects [79], are deterministic, while others are random: this work will focus on the effects of random variations, and will model these parameters as random variables. The gate and interconnect delays, as functions of these parameters, also become random variables. Given appropriate modeling of process parameters or gate and interconnect delays, the task of SSTA is to find the PDF of the circuit delay.

The static timing analysis works with the usual translation from a combinational circuit to a timing graph [67]. The nodes in this graph correspond to the circuit primary inputs/outputs and gate input/output pins. The edges are of two types: one set corresponds to the pin-to-pin delay arcs within a gate, and the other set to interconnections from the drivers to receivers. The edges are weighted by the pin-to-pin gate delay, and interconnect delay, respectively. The primary inputs of the combinational circuit are connected to a virtual source node, and the primary outputs to a virtual sink node with directed virtual edges. In the case that primary inputs arrive at different times, the virtual edges from the virtual source to the primary inputs are assigned weights of the arrival times. Likewise, if the required times at the primary outputs are different, the weights of the edges from the outputs to the virtual sink are appropriately chosen.

For a combinational logic circuit, the problem of static timing analysis is to compute the longest path delay in the circuit from any primary input to any primary output, which corresponds to length of the longest path in the timing graph. In static timing analysis, the technique that is commonly referred to in the literature as PERT is commonly used[1]. This procedure starts from the source node to traverse

[1]In reality, this is actually the critical path method (CPM) in operations research. However, we will persist with the term "PERT," which is widely used in the static timing analysis literature.

the graph in a topological order and uses a *sum* operation or *max* operation (at a multi-fanin node) to find the longest path at the sink node. For details, the reader may refer to [41, 67].

Since we will employ a PERT-like traversal to analyze the distribution of circuit delay, we define a statistical timing graph of a circuit, as in the case of deterministic STA.

Definition 3.2.1 *Let $G_s = (V, E)$ be a timing graph for a circuit with a single source node and a single sink node, where V is a set of nodes and E a set of directed edges. The graph G_s is called a statistical timing graph if each edge i is assigned a weight d_i, where d_i is a random variable, where the random variables may be uncorrelated or correlated. The weight associated with an edge corresponds to gate delay or interconnect delay. For a virtual edge, the weight is random variables with mean of its deterministic value and standard deviation of zero and it is independent from any other edges.*

Definition 3.2.2 *Let a path p_i be a set of ordered edges from the source node to the sink node in G_s, and D_i be the path length distribution of p_i, computed as the sum of the weights d_k for all edges k on the path. Finding the distribution of $D_{max} = max(D_1, \ldots, D_i, \ldots, D_{n_{paths}})$ among all paths (indexed from 1 to n_{paths}) in the graph G_s is referred to as the problem of SSTA of a circuit.*

Note that for the same nominal design, the identity of the longest path may change, depending on the random values taken by the process parameters. Therefore, finding the delay distribution of one critical path at a time is not enough, and correlations between paths must be considered in finding the maximum of the PDFs of all paths. Such an analysis is essential for finding the probability of failure

of a circuit, which is available from the cumulative density function (CDF) of the circuit delay.

For an edge-triggered sequential circuit, the statistical timing graph can be constructed similarly by breaking the circuit into a set of combinational blocks between latches, and the analysis includes statistical checks on setup and hold time violations. The former requires the computation of the distribution of the maximum arrival time at the latches, which requires the solution of the SSTA problem as defined above. On the other hand, the latter problem requires the distribution of the minimum arrival time at the latches to be computed, and this can be solved by a trivial extension of the framework for the SSTA problem proposed in this chapter using *min* operators, as will be mentioned in Section 3.5.3, instead of *max* operators.

We will use the models of inter-die and intra-die process variations described in Chapter 2. For intra-die variation, we only consider the impact of global and random components. However, the local component can also be accounted for in the proposed method, given, for instance, the chip layout and precharacterized spatial maps of parameters as in [59]. For transistors, we consider the following process parameters [56] as random variables: transistor length L_{eff} and width W_g, gate oxide thickness T_{ox}, doping concentration density N_a; for interconnect, at each metal layer, we consider the following parameters: metal width W_{int_l}, metal thickness T_{int_l} and interlayer dielectric (ILD) thickness H_{ILD_l}, where the subscript l represents that the random variable is of layer l, where $l = 1 \ldots n_{layers}$. However, the SSTA method presented in this chapter is general enough that it can be applied to handle variations in other parameters as well.

For spatial correlation, we use the grid-based model proposed in Section 2.3. It is assumed that nonzero correlations may exist only among the same type of

process parameters in different grids, and there is no correlation between different types of process parameters. (Note here that we consider interconnect parameters in different layers to be "different types of parameters," e.g., W_{int_1} and W_{int_2} are uncorrelated[2].)

The process parameter values are assumed to be normally distributed random variables. The gate and interconnect delays, being functions of the fundamental process parameters, are approximated using a first-order Taylor series expansion. We will show that as a result of this, all edges in graph G_s are normally distributed random variables. Since we consider spatial correlations of the process parameters, it turns out that some of the delays are correlated random variables. Furthermore, the circuit delay D_{max} is modeled as a multivariate normal distribution. Although the closed form of circuit delay distribution is not normal, we show that the loss of accuracy is not significant under this approximation.

3.3 SSTA Algorithm

The core SSTA method is described in this section, and its description is organized as follows. At first, in Section 3.3.1, we will describe how we model the distributions of gate and interconnect delays as normal distributions, given the PDFs that describe the variations of various parameters. In general, these PDFs will be correlated with each other. In Section 3.3.2, we will show how we can simplify the complicated correlated structure of parameters by orthogonal transformations. Section 3.3.3 will describe the PERT-like traversal algorithm on the statistical tim-

[2]This assumption is not critical to the correctness of our procedure, but is used in our experimental results. Our method is general enough to handle correlations between parameters of different types.

ing graph by demonstrating the procedure for the computation of *max* and *sum* functions. Finally, Section 3.3.4 will explain why orthogonal transformations are important in our method.

For clarity of presentation, the approach in this section is presented to handle intra-die variations. The extension to accounting for inter-die variations will be presented in section 3.5. We here assume that all types of process parameters have spatial correlations. The extension of this work to incorporate the effect of this component will be shown in Section 3.5.

3.3.1 Modeling Gate/Interconnect Delay PDFs

In this section, we will show how the variations in the process parameters are translated into PDFs that describe the variations in the gate and interconnect delays that correspond to the weights on edges of the statistical timing graph.

Before we introduce how the distributions of gate and interconnect delays will be modeled, let us first consider an arbitrary function $d = f(\vec{P})$ that is assumed to be a function on a set of process parameters $\vec{P}$, where each $p_i \in \vec{P}$ is a random variable with a normal distribution given by $p_i \sim N(\mu_{p_i}, \sigma_{p_i})$.

We can approximate the function d linearly using a first order Taylor expansion:

$$d = d_0 + \sum_{\forall \text{ parameters } p_i} \left[\frac{\partial f}{\partial p_i}\right]_0 \Delta p_i \tag{3.1}$$

where d_0 is the nominal value of d, calculated at the nominal values of process parameters in the set $\vec{P}$, $\frac{\partial f}{\partial p_i}$ is computed at the nominal values of p_i, $\Delta p_i = p_i - \mu_{p_i}$ is a normally distributed random variable and $\Delta p_i \sim N(0, \sigma_{p_i})$.

In this approximation, d is modeled as a normal distribution, since it is a linear combination of normally distributed random variables. Its mean μ_d, and variance

σ_d^2 are:

$$\mu_d = d_0 \tag{3.2}$$

$$\sigma_d^2 = \sum_{\forall i} \left[\frac{\partial f}{\partial p_i}\right]_0^2 \sigma_{p_i}^2 + 2\sum_{\forall i \neq j} \left[\frac{\partial f}{\partial p_i}\right]_0 \left[\frac{\partial f}{\partial p_j}\right]_0 cov(p_i, p_j) \tag{3.3}$$

where $cov(p_i, p_j)$ is the covariance of p_i and p_j.

It is reasonable to ask whether the approximation of d as a normal distribution is valid, since the distribution of d may, strictly speaking, not be Gaussian. We can say that when Δp_i has relatively small variations, the first order Taylor expansion is adequate and the approximation is acceptable with little loss of accuracy. This is generally true of intra-die variations, where the process parameter variations are relatively small in comparison with the nominal values. For this reason, as functions of process parameters, the gate and interconnect delays can be approximated as a sum of normal distributions (which is also normal) applying the Equation (3.1).

Computing the PDF of interconnect delay

In this work, we use the Elmore delay model for simplicity to calculate the interconnect delays[3]. Under the Elmore model, the interconnect delay is a function of the resistances $\vec{R}_w$ and capacitances $\vec{C}_w$ of all wire segments in the interconnect tree and input load capacitances $\vec{C}_g$ of the fanout gates, or receivers.

$$d_{int} = d(\vec{R}_w, \vec{C}_w, \vec{C}_g) \tag{3.4}$$

Since the resistances and capacitances above are furthermore decided by the process parameters $\vec{P}$ of the interconnect and the receivers, such as W_{int_l}, T_{int_l}, H_{ILD_l}, W_g,

[3]However, it should be emphasized that any delay model may be used, and all that is required is the sensitivity of the delay to the process parameters. For example, through a full circuit simulation, the sensitivities may be computed by performing adjoint sensitivity analysis.

L_{eff} and T_{ox}, the sensitivities of the interconnect delay to a process parameter p_i can be found by using the chain rule:

$$\frac{\partial d_{int}}{\partial p_i} = \sum_{\forall R_{w_k} \in \vec{R}_w} \frac{\partial d}{\partial R_{w_k}} \frac{\partial R_{w_k}}{\partial p_i} + \sum_{\forall C_{w_k} \in \vec{C}_w} \frac{\partial d}{\partial C_{w_k}} \frac{\partial C_{w_k}}{\partial p_i} + \sum_{\forall C_{g_k} \in \vec{C}_g} \frac{\partial d}{\partial C_{g_k}} \frac{\partial C_{g_k}}{\partial p_i} \quad (3.5)$$

The distribution of interconnect delay can then be approximated on the computed sensitivities.

We will now specifically consider the factors that affect the interconnect delay associated with edges in the statistical timing graph. Recall that under our model, we divide the chip area into grids so that the process parameter variations within a grid are identical, but those in different grids exhibit spatial correlations. Now consider an interconnect tree with several different segments that reside in different grids. The delay variations in the tree are affected by the process parameter variations of wires in all grids that the tree traverses. For example, in Figure 2.2, consider the two segments uv and pq in the interconnect tree driven by gate a. Segment uv passes through the grid $(1, 1)$ and pq through the grid $(1, 2)$. Then the resistance and capacitance of segment uv should be calculated based on the process parameters of grid $(1, 1)$, while the resistance and capacitance of segment pq should be based on those of grid $(1, 2)$. Hence, the distribution of the interconnect tree delay is actually a function of random variables of interconnect parameters in both grid $(1, 1)$ and grid $(1, 2)$, and should incorporate any correlations between these random variables. Similarly, if the gates that the interconnect tree drives reside in different grid locations, the interconnect delay to any sink is also a function of random variables of gate process parameters of all grids in which the receivers are located.

In summary, the distribution of interconnect delay function can be approximated

by:

$$
\begin{aligned}
d_{int} \;=\; & d_{int}^{0} + \sum_{i \subset \Gamma_g} \left[\frac{\partial d}{\partial L_{eff}^{i}} \right]_0 \Delta L_{eff}^{i} + \sum_{i \subset \Gamma_g} \left[\frac{\partial d}{\partial W_{g}^{i}} \right]_0 \Delta W_{g}^{i} \\
& + \sum_{i \subset \Gamma_g} \left[\frac{\partial d}{\partial T_{ox}^{i}} \right]_0 \Delta T_{ox}^{i} + \sum_{l=1}^{n_{layer}} \left\{ \sum_{i \subset \Gamma_{int}} \left[\frac{\partial d}{\partial W_{int_l}^{i}} \right]_0 \Delta W_{int_l}^{i} \right. \\
& \left. + \sum_{i \subset \Gamma_{int}} \left[\frac{\partial d}{\partial T_{int_l}^{i}} \right]_0 \Delta T_{int_l}^{i} + \sum_{i \subset \Gamma_{int}} \left[\frac{\partial d}{\partial H_{ILD_l}^{i}} \right]_0 \Delta H_{ILD_l}^{i} \right\}
\end{aligned}
\tag{3.6}
$$

where d_{int}^{0} is the interconnect delay value calculated with nominal values of process parameters, Γ_g is the set of indices of grids that all the receivers reside in, Γ_{int} is the set of indices of grids that the interconnect tree traverses, and $\Delta L_{eff}^{i} = L_{eff}^{i} - \mu_{L_{eff}^{i}}$ where L_{eff}^{i} is the random variable representing transistor length in the i^{th} grid. The parameters ΔW_{g}^{i}, ΔT_{ox}^{i}, $\Delta W_{int_l}^{i}$, $\Delta T_{int_l}^{i}$ and $\Delta H_{ILD_l}^{i}$ are similarly defined. As before, the subscript "$_0$" next to each sensitivity represents the fact that it is evaluated at the nominal point.

Computing the PDF of gate delay and output signal transition time

The distribution of gate delay and output signal transition time at the gate output can be approximated in a similar manner as described above, given the sensitivities of the gate delay to the process parameters.

Consider a multiple-input gate, let $d_{gate}^{pin_i}$ be the gate delay from the i^{th} input to the output and $S_{out}^{pin_i}$ be the corresponding output signal transition time. In general, both $d_{gate}^{pin_i}$ and $S_{out}^{pin_i}$ can be written as a function of the process parameters $\vec{P}$ of the gate, the loading capacitance of the driving interconnect tree $\vec{C}_w$ and the succeeding gates that it drives $\vec{C}_g$, and the input signal transition time $S_{in}^{pin_i}$ at this

input pin of the gate

$$d_{gate}^{pin_i} = D_{gate}(\vec{P}, \vec{C}_w, \vec{C}_g, S_{in}^{pin_i}), \tag{3.7}$$

$$S_{out}^{pin_i} = S_{gate}(\vec{P}, \vec{C}_w, \vec{C}_g, S_{in}^{pin_i}). \tag{3.8}$$

The distributions of $d_{gate}^{pin_i}$ and $S_{in}^{pin_i}$ can be approximated as Gaussians using linear expressions of parameters, where the mean values of $d_{gate}^{pin_i}$ or $S_{in}^{pin_i}$ can be found by using the mean values of $\vec{P}$, $\vec{C}_w$, $\vec{C}_g$ and $S_{in}^{pin_i}$ in functions D_{gate} or S_{gate}, and the sensitivities of either $d_{gate}^{pin_i}$ or $S_{in}^{pin_i}$ to process parameters can be computed applying the chain rule. The derivatives of $\vec{C}_w$ and $\vec{C}_g$ to the process parameters can be easily computed, as $\vec{C}_w$ and $\vec{C}_g$ are functions of process parameters. The input signal transition time, S_{in}, is a function of the output transition time of the preceding gate and the delay of the interconnect connecting the preceding gates and this gate, where both interconnect delay (as discussed earlier) and output transition time of the preceding gate (as will be shown in the next paragraph) are Gaussian random variables that can be expressed as a linear function of parameter variations. Therefore, at a gate input, the input signal transition time S_{in} is always given as a normally distributed random variable with a mean and first-order sensitivities to the parameter variations.

To consider the effect of the transition time of an input signal on the gate delay, the output signal transition time S_{out} at each gate output must be computed in addition to pin-to-pin delay of the gate. In conventional static timing analysis, S_{out} is set to $S_{out}^{pin_i}$ if the path ending at the output of the gate traversing the i^{th} input pin has the longest path delay d_{path_i}. In SSTA, each of the paths through different gate input pins has a certain probability to be the longest path. Therefore, S_{out} should be computed as a weighted sum of the distributions of $S_{out}^{pin_i}$, where the weight equals the probability that the path through the i^{th} pin is the longest among

all others:

$$S_{out} = \sum_{\forall input\ pin\ i} \{Prob[d_{path_i} > max_{\forall j \neq i}(d_{path_j})] \times S_{out}^{pin_i}\}, \tag{3.9}$$

where d_{path_i} is the random path delay variable at the gate output through the i^{th} input pin. The result of $max_{\forall j \neq i}(d_{path_j})$ is a random variable representing for the distribution of maximum of multiple paths. As will be discussed later in Section 3.3.3, d_{path_i} and $max_{\forall j \neq i}(d_{path_j})$ can be approximated as Gaussians using *sum* and *max* operators, and their correlation can easily be computed. Therefore, finding the value of $Prob[d_{path_i} > max_{\forall j \neq i}(d_{path_j})]$, i.e., $Prob[d_{path_i} - max_{\forall j \neq i}(d_{path_j} > 0)]$ becomes computing the probability of a Gaussian random variable greater than zero, which can easily be found from a look-up table. As each $S_{out}^{pin_i}$ is a Gaussian random variable in linear combination of the process parameter variations, S_{out} is therefore also a Gaussian-distributed random variable and its sensitivities to all process parameters $\frac{\partial S_{out}}{\partial p_i}$ can easily be found from its linear expression.

3.3.2 Orthogonal Transformation of Correlated Variables

In statistical timing analysis without spatial correlations, correlations due to reconvergent paths has long been an obstacle. When the spatial correlation of process parameters is also taken into consideration, the correlation structure becomes even more complicated. To make the problem tractable, we use the Principal Component Analysis (PCA) technique [51] to transform the set of correlated parameters into an uncorrelated set.

PCA is a method that can be employed to examine the relationship among a set of correlated variables. Given a set of correlated random variables $\vec{X}$ with a covariance matrix R, PCA can transform the set $\vec{X}$ into a set of mutually orthogonal

random variables, $\vec{X'}$, such that each member of $\vec{X'}$ has zero mean and unit variance. The elements of the set $\vec{X'}$ are called principal components in PCA, and the size of $\vec{X'}$ is no larger than the size of $\vec{X}$. Any variable $x_i \in \vec{X}$ can then be expressed in terms of the principal components $\vec{X'}$ as follows:

$$x_i = (\sum_j \sqrt{\lambda_j} \cdot v_{ij} \cdot x'_j)\sigma_i + \mu_i, \tag{3.10}$$

where x'_j is a principal component in set $\vec{X'}$, λ_j is the j^{th} eigenvalue of the covariance matrix R, v_{ij} is the i^{th} element of the j^{th} eigenvector of R, and σ_i and μ_i are, respectively, the mean and standard deviation of x_i.

Since we assume that different types of parameters are uncorrelated, we can group the random variables of parameters by types and perform principal component analysis in each group separately, i.e., we compute the principal components for $\vec{L}_{eff}$, $\vec{W}_g$, $\vec{T}_{ox}$, $\vec{N}_a$, $\vec{W}_{int_l}$ and $\vec{T}_{int_l}$ individually. Clearly, not only are the principal components of the same type of parameters independent, but so are the principal components of different type of parameters.

For instance, let $\vec{L}_{eff}$ be a random vector representing transistor gate length variations in all grids and it is of multivariate normal distribution with covariance matrix $R_{L_{eff}}$. Let $\vec{L'}_{eff}$ be the set of principal components computed by PCA. Then any $L^i_{eff} \in \vec{L}_{eff}$ representing the variation of transistor gate length of the i^{th} grid can then be expressed as a linear function of the principal components

$$L^i_{eff} = \mu_{L^i_{eff}} + a_{i1} \times L'^1_{eff} + \cdots + a_{it} \times L'^t_{eff}, \tag{3.11}$$

where $\mu_{L^i_{eff}}$ is the mean of L^i_{eff}, l'^i_{eff} is a principal component in $\vec{L'}_{eff}$, all l'^i_{eff} are independent with zero means and unit variances, and t is the total number of principal components in $\vec{L'}_{eff}$.

In this way, any random variable in $\vec{W}_g$, $\vec{T}_{ox}$, $\vec{N}_a$, $\vec{W}_{int_l}$, $\vec{T}_{int_l}$ and $\vec{H}_{ILD_l}$ can be expressed as a linear function of the corresponding principal components in $\vec{W}'_g$, $\vec{T}'_{ox}$, $\vec{N}'_a$, $\vec{W}'_{int_l}$, $\vec{T}'_{int_l}$ and $\vec{H}'_{ILD_l}$. Superposing the set of rotated random variables of parameters on the original random variables in gate or interconnect delay in Equation (3.6), the expression of gate or interconnect delay is then changed to the linear combination of principal components of all parameters

$$d = d_0 + k_1 \times p'_1 + \cdots + k_m \times p'_m, \tag{3.12}$$

where $p'_i \in \vec{P}'$ and $\vec{P}' = \vec{L}'_{eff} \cup \vec{W}'_g \cup \vec{T}'_{ox} \cup \vec{N}'_a \cup \vec{W}'_{int_l} \cup \vec{T}'_{int_l} \cup \vec{H}'_{ILD_l}$ and m is the size of $\vec{P}'$.

Note that all of the principal components p'_i that appear in Equation (3.12) are independent. Equation (3.12) has the following properties:

Property 1 Since all p'_i are orthogonal, the variance of d can be simply computed as

$$\sigma_d^2 = \sum_{i=1}^{m} k_i^2. \tag{3.13}$$

Property 2 The covariance between d and any principal component p'_i is given by

$$cov(d, p'_i) = k_i \sigma^2_{p'_i} = k_i. \tag{3.14}$$

In other words, the coefficient of p'_i is exactly the covariance between d and p'_i.

Property 3 Let d_i and d_j be two random variables:

$$d_i = d_i^0 + k_{i1} \times p'_1 + \cdots + k_{im} \times p'_m, \tag{3.15}$$

$$d_j = d_j^0 + k_{j1} \times p'_1 + \cdots + k_{jm} \times p'_m. \tag{3.16}$$

The covariance of d_i and d_j, $cov(d_i, d_j)$, can be computed by

$$cov(d_i, d_j) = \sum_{r=1}^{m} k_{ir} k_{jr}. \tag{3.17}$$

In comparison, without an orthogonal transformation, the value of $cov(d_i, d_j)$ must be computed by a more complicated formula as will be described in Section 3.3.4.

3.3.3 PERT-like Traversal of SSTA

Using the techniques discussed up to this point, all edges of the statistical timing graph may be modeled as normally distributed random variables. In this section, we will describe a procedure for finding the distribution of the statistical longest path in the graph.

In conventional deterministic STA, the PERT algorithm can be used to find the longest path in a graph by traversing it in topological order using two types of functions:

- the *sum* function, and
- the *max* function.

In our statistical timing analysis, a PERT-like traversal is employed to find the distribution of circuit delay. However, unlike deterministic STA, the *sum* and *max* operations here are functions of a set of correlated multivariate Gaussian random variables instead of fixed values:
1) $d_{sum} = \sum_{i=1}^{l} d_i$, and
2) $d_{max} = max(d_1, \cdots, d_l)$.
where d_i is a Gaussian random variable representing either gate delay or wire delay expressed as linear functions of principal components in the form of Equation (3.15), and l is the number of random variables that *sum* or *max* function is operating on.

Computing the distribution of the *sum* function

The computation of the distribution of *sum* function is simple. Since the $d_{sum} = \sum_{i=1}^{l} d_i$ is a linear combination of normally distributed random variables, d_{sum} is a normal distribution. The mean μ_{dsum} and variance σ^2_{dsum} of the *sum* are given by

$$\mu_{d_{sum}} = \sum_{i=1}^{l} d_i^0, \tag{3.18}$$

$$\sigma^2_{d_{sum}} = \sum_{j=1}^{m} (\sum_{i=1}^{l} k_{ij})^2. \tag{3.19}$$

Computing the distribution of the *max* function

The *max* function of l normally distributed random variables $d_{max} = max(d_1, \cdots, d_l)$ is, strictly speaking, not Gaussian. However, we have found that, in practice, it can be approximated closely by a Gaussian. This idea is similar in spirit to Berkelaar's approach in [9, 35], although it is more general since Berkelaar's work restricted its attention to delay random variables that were uncorrelated[4]. In this work, we use the Gaussian distribution to approximate the result of a *max* function, so that $d_{max} \sim N(\mu_{d_{max}}, \sigma_{d_{max}})$. We also approximate d_{max} as a linear function of all the principal components $p'_1 \cdots p'_m$

$$d_{max} = \mu_{d_{max}} + a_1 p'_1 + \cdots + a_m p'_m. \tag{3.20}$$

Therefore, determining this approximation for d_{max} is equivalent to finding the values of $\mu_{d_{max}}$ and all a_i's.

From *Property 2* of Section 3.3.2, we know that the coefficient a_r equals $cov(d_{max}, p'_r)$. Then the variance of the expression on the right hand side of Equation (3.20) is

[4]Many researchers in the community were well aware of Berkelaar's results as early as 1997, though his work did not appear as an archival publication.

computed as $s_0^2 = \sum_{r=1}^{m} a_r^2 = \sum_{r=1}^{m} cov^2(d_{max}, p'_r)$. Since this is merely an approximation, there may be a difference between the value s_0^2 and the actual variance $\sigma_{d_{max}}^2$ of d_{max}. To diminish the difference, we can normalize the value of a_r by setting it as

$$a_r = cov(d_{max}, p'_r) \cdot \frac{\sigma_{d_{max}}}{s_0}. \tag{3.21}$$

We can see now that to find the linear approximation for d_{max}, the values of $\mu_{d_{max}}$, $\sigma_{d_{max}}$ and $cov(d_{max}, p_i)$ are required. In the work of [78], similar inputs were required in their algorithm and the results from [22] were applied and seen to provide good results. In this work, we have borrowed the same analytical formula from [22] for the computation of the *max* function.

According to [22], if ξ and η are two random variables, $\xi \sim N(\mu_1, \sigma_1)$, $\eta \sim N(\mu_2, \sigma_2)$, with a correlation coefficient of $r(\xi, \eta) = \rho$, then the mean μ_t and the variance σ_t^2 of $t = \max(\xi, \eta)$ can be approximated by

$$\mu_t = \mu_1 \cdot \Phi(\beta) + \mu_2 \cdot \Phi(-\beta) + \alpha \cdot \varphi(\beta), \tag{3.22}$$

$$\sigma_t^2 = (\mu_1^2 + \sigma_1^2) \cdot \Phi(\beta) + (\mu_2^2 + \sigma_2^2) \cdot \Phi(-\beta) + (\mu_1 + \mu_2) \cdot \alpha \cdot \varphi(\beta) - \mu_t^2, \tag{3.23}$$

where

$$\alpha = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho}, \tag{3.24}$$

$$\beta = \frac{(\mu_1 - \mu_2)}{\alpha}, \tag{3.25}$$

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} exp\left[-\frac{x^2}{2}\right], \tag{3.26}$$

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} exp\left[-\frac{y^2}{2}\right] dy. \tag{3.27}$$

The formula will not apply if $\sigma_1 = \sigma_2$ and $\rho = 1$. However, in this case, the *max* function is simply identical to the random variable with largest mean value.

Moreover, from [22], if γ is another normally distributed random variable and the correlation coefficients $r(\xi, \gamma) = \rho_1$, $r(\eta, \gamma) = \rho_2$, then the correlation between γ and $t = \max(\xi, \eta)$ can be obtained by

$$r(t, \gamma) = \frac{\sigma_1 \cdot \rho_1 \cdot \Phi(\beta) + \sigma_2 \cdot \rho_2 \cdot \Phi(-\beta)}{\sigma_t}. \tag{3.28}$$

Using the formula above, we can find all the values required. As an example, let us see how this can be done by first starting with a two-variable *max* function, $d_{max} = max(d_i, d_j)$. Let d_{max} be of the form of Equation (3.20). We can find the approximation of d_{max} as follows:

1. Given the expressions of d_i and d_j each as linear combinations of the principal components, compute their mean and standard deviation values μ_{d_i}, σ_{d_i} and μ_{d_j}, σ_{d_j}, respectively, as described in *Property 1* of Section 3.3.2.

2. Find the correlation coefficient between d_i and d_j where $cov(d_i, d_j)$, the covariance of d_i and d_j, can be computed using *Property 3* in Section 3.3.2.

   Now if $r(d_i, d_j) = 1$ and $\sigma_{d_i} = \sigma_{d_j}$, set d_{max} to be identical to d_i or d_j, whichever has larger mean value and we can stop here; otherwise, we will continue to the next step.

3. Calculate the mean $\mu_{d_{max}}$ and variance $\sigma^2_{d_{max}}$ of d_{max} using Equations (3.22) and (3.23).

4. Find all coefficients a_r of p'_r. According to *Property 2*, $a_r = cov(d_{max}, p'_r)$, also, $cov(d_i, p'_r) = k_{ir}$ and $cov(d_j, p'_r) = k_{jr}$. Applying Equation (3.28), the values of $cov(d_{max}, p'_r)$ and thus a_r can be calculated.

5. After all of the a_r's have been calculated, determine $s_0 = \sqrt{\sum_{r=1}^{m} {a_r}^2}$. Normalize the coefficient by resetting each $a_r = a_r \frac{\sigma_{d_{max}}}{s_0}$.

The calculation of the two-variable *max* function can easily be extended to a multi-variable *max* function by repeating the steps of the two-variable case recursively.

As mentioned at the beginning of this section, max of two Gaussian random variables is not strictly Gaussian. This approximation can sometimes introduce serious error, e.g., when the two Gaussian random variables have the same mean and standard deviation and correlation value of -1, and the distribution of the maximum is a half Gaussian. During the computation of multi-variable *max* function, some inaccuracy could be introduced since we approximate the *max* function as normal even though it is not really normal, and proceed with further recursive calculations. To the best of our knowledge, there is no theoretical analysis available in literature that quantifies the inaccuracies when a normal distribution is used to approximate the maximum of a set of Gaussian random variables. However, a numerically based analysis was provided in [22] which suggests that in some situations the errors can be great, but for many applications this approximate is quite satisfactory. We will show results in Section 3.6 that suggest that such inaccuracies are not significant in the circuit context, and we will see that our results match very well with the simulation results from a Monte Carlo analysis.

Moreover, recall that we have a "normalization" step to diminish the difference between the variance computed from the linear form of *max* approximation and the real variance of the *max* function. As in the case of approximating the *max* as normal distribution, there is no theoretical proof about how this "normalization" step can affect the accuracy of the approximation. Another option to diminish the difference is to move it into an independent random Gaussian component, and it is difficult to state definitively which of these options is better. In our work, we choose the former option and find that it provides excellent accuracy, as will be

shown in Section 3.6, where the statistics of the "normalization" ratio for several test circuits are provided.

At this point, not only the edges, but also the results of *sum* and *max* functions are expressed as linear functions of the principal components. Therefore, using a PERT traversal by incorporating the computation of *sum* and *max* functions described above, the distribution of arrival time at any node in the timing graph becomes a linear function of principal components, and so the distribution of circuit delay can be computed at the virtual sink node.

The overall flow of our algorithm is shown in Figure 3.1. It is noticed that this work is in some sense parallel to the work of [36]: in [36], delays are represented as linear combinations of global random variables, while in our work, they are linear functions of principal components; in [36], the *max* of delays are reexpressed as linear functions using binding probabilities, while in our work, the linear functions are found by an analytical method from [22].

To further speed up the process, the following technique may be used: During the *max* operation of SSTA, if the value of $\mu + 3 \cdot \sigma$ of one path has a lower delay than the value of $\mu - 3 \cdot \sigma$ of another path, we can simply calculate the *max* function ignoring the former path.

3.3.4 The Utility of Principal Components

The previous sections described our SSTA algorithm. The purpose of this section is to elaborate why the orthogonal transformation is required to transform the set of correlated process parameters to an uncorrelated set, and how it can simplify the problem of SSTA considering spatial correlations.

Let d_i and d_j be the distributions of two gate delays. For simplicity, we assume

Input: *Process parameter variations*

Output: *Distribution of circuit delay*

1. *According to the size of the chip, partition the chip region into* $n = nrow \times ncol$ *grids.*

2. *For each type of parameter, determine the* n *jointly normally distributed random variables and the corresponding covariance matrix.*

3. *Perform an orthogonal transformation to represent each random variable with a set of principal components.*

4. *For each gate and net connection, model their delays as linear combinations of the principal components generated in step 3.*

5. *Map the circuit into a statistical timing graph by adding one virtual-source node, one virtual-sink node and corresponding edges.*

6. *Using* sum *and* max *functions on Gaussian random variables, perform a PERT-like traversal on the graph to find the distribution of the statistical longest path. This distribution achieved is the circuit delay distribution.*

Figure 3.1: Overall flow of our statistical timing analysis.

that the gate lengths $\vec{L}_{eff}$ are the only spatially correlated parameters. We also assume that d_i and d_j are sensitive to the same set of correlated random variables of gate lengths $L^1_{eff}, \ldots, L^n_{eff}$. Using Equation (3.6), d_i and d_j can be expressed as

$$d_i = d^0_i + c_{i1}L^1_{eff} + \cdots + c_{in}L^n_{eff}, \tag{3.29}$$

$$d_j = d^0_j + c_{j1}L^1_{eff} + \cdots + c_{jn}L^n_{eff}. \tag{3.30}$$

Obviously, the covariance of d_i and d_j is decided by the covariance structure of $\vec{L}_{eff}$. The direct calculation of $cov(d_i, d_j)$ is of a complicated form as in the work of [58]

$$cov(d_i, d_j) = \sum_{a=1}^{n}\sum_{b=1}^{n} c_{ia}c_{jb}cov(L^a_{eff}, L^b_{eff}). \tag{3.31}$$

In contrast, in our method, we first perform orthogonal transformations on $\vec{L}_{eff}$. Any element $L^l_{eff} \in \vec{L}_{eff}$ is expressed as

$$L^l_{eff} = \mu_{L^l_{eff}} + a_{l1}l'^1_{eff} + \cdots + a_{lm}l'^m_{eff}. \tag{3.32}$$

Next, by superposition we transform d_i and d_j to:

$$d_i = d^0_i + k_{i1}l'^1_{eff} + \cdots + k_{im}l'^m_{eff}, \tag{3.33}$$

$$d_j = d^0_j + k_{j1}l'^1_{eff} + \cdots + k_{jm}l'^m_{eff}. \tag{3.34}$$

The value of $cov(d_i, d_j)$ can be simply computed using the coefficients of $\vec{L}'_{eff}$ by $cov(d_i, d_j) = \sum_{r=1}^{m} k_{ir}k_{jr}$ in linear time $O(m)$. The advantage in this computation is that we need not know which specific parameters in d_i and d_j are correlated. In fact, consider the coefficients of l'^1_{eff} in both d_i and d_j, $k_{i1} = \sum_{r=1}^{n} c_{ir}a_{r1}$ and $k_{j1} = \sum_{r=1}^{n} c_{jr}a_{r1}$. It can be seen that the covariance of gate lengths have all been incorporated in the coefficient of the principal components $l'^1_{eff}, \ldots, l'^m_{eff}$. For

this reason, we ensure that the computation of $cov(d_i, d_j)$ can actually take the correlations of gate lengths into consideration correctly.

The direct computation of the covariance of path delays is in a similar form. In general, the path delays are correlated when the gate delays on these paths are correlated. As shown in the work of [58], the path covariances can be computed on the basis of pair-wise gate delay covariances; however, the number of paths is numerous which makes it computationally difficult to apply such a path-based method to large circuits.

In our method, with the orthogonal transformation, the covariances of path delays are manifested as the coefficients of the independent principal components as in the case of correlated gate delays. The covariances of the paths can then be simply computed in linear time based on these coefficients only, and it is not necessary to keep track of how the gates on the paths are correlated or which parts are correlated. For the same reason, in this algorithm, besides the spatial correlations, path correlations due to reconvergence (structural correlations) can also be accounted for automatically by using the orthogonal transformation on the spatially correlated parameters. However, when spatially uncorrelated parameters are involved in the computation, the structural correlations due to these independent parameters can not be dealt with by this method easily. The extension of the work for handling spatially uncorrelated parameters will be given in Section 3.5.2.

3.4 Computational Complexity

We present a run time complexity analysis here to show which factors most greatly affect the CPU time of the algorithm.

The flow shown in Figure 3.1 can be divided into two parts: model precharacterization (steps 1, 2 and 3) and SSTA (steps 4, 5 and 6). Model precharacterization consists of construction of parameter variations and grid-based spatial correlation models, and the computation of Principal Components (PC) for spatially correlated parameters. The computation of PCs requires calculations of eigenvectors and eigenvalues of the covariance matrix and its time complexity is $O(p \cdot n^3)$, where n is total number of grids into which the chip is divided and p is the number of spatially correlated parameters considered. While this step may seem to be a bottleneck of the algorithm, it is a only one-time computation for a process. Once the models of parameter variations are constructed, they can be repeatedly used to analyze any design. Meanwhile, for spatial correlated parameters, the PCs computed from the covariance matrix are only model-dependent, so that for different designs analyzed with the same parameter model, the same set of PCs can be applied. In other words, the step of model precharacterization is in fact a one-time library construction at early stage and therefore can be excluded from the run time complexity analysis of the algorithm.

The run-time of the SSTA algorithm can be divided into:

1. The time required to find the delay distribution of the gate and interconnect[5]: This run time depends on how many different grids the interconnect passes through and how many grids the gates are located in, and in general these numbers are bounded by constant numbers. The run time is also proportional to the total number of principal components, since we perform orthogonal transformation at each wire segment of interconnect. For each random variable, the number of principal components is no more than the

[5]The time required to precharacterize the sensitivities of delay on parameter variations is excluded from this analysis, since that task can be carried out offline, rather than *during* SSTA.

total number of grids n partitioned on the chip. The total number of principal components is no more than $p \cdot n$. Thus, the time required to find the distribution of a single gate or wire can be estimated as $O(p \cdot n)$. If N_g is the total number of gates and N_I the number of net connections in the circuit, the time of this part can be estimated as $O(p \cdot n \cdot (N_g + N_I))$.

2. The time required to evaluate the *max* function: The cost of this operation is proportional to the number of random variables involved in the *max* operation and the number of principal components of each random variable. The *max* operation is used at all multi-input gates and at the last level (sink node) where the maximum circuit delay is computed. This number can be upper bounded by the total number of net connections N_I in the circuit. Thus, the run time of this part is $O(p \cdot n \cdot N_I)$.

3. The time required to compute output transition time at each gate output: For a gate with $k > 2$ inputs, it requires k^2 *max* operations and $k - 1$ *sum* operations, which are constant numbers of *max* and *sum* operations. The computation is required for all gates and thus the total cost is $O(p \cdot n \cdot N_g)$.

4. The time required to evaluate the *sum* function: The *sum* operation must be performed at all gates and interconnects encountered during the PERT-like traversal. A single *sum* operation requires $O(n)$, and therefore, the total complexity for this part is $O(p \cdot n \cdot (N_g + N_I))$.

Therefore, the run time complexity of the algorithm is $O(p \cdot n \cdot (N_g + N_I))$, which is $p \cdot n$ times that of deterministic STA.

3.5 Extending the Method to Handle Inter-die Variations, Spatially Uncorrelated Parameters, and Min-delay Computations

In this section, we will first describe how this work can be extended to include the effect of inter-die variations in addition to intra-die variations. Subsequently, we will explain how spatially uncorrelated parameters can be incorporated into the current proposed algorithm. Finally, we will show how minimum delay computations can easily be incorporated into this framework.

3.5.1 Inter-die Variations

As explained in Chapter 2, any process parametric variation can be modeled as

$$\Delta p_{total} = \Delta p_{inter} + \Delta p_{intra}, \tag{3.35}$$

where Δp_{inter} is the inter-die variation and Δp_{intra} is the intra-die variation of the process parameter.

Since inter-die variation has a global effect on all the transistors [wires] within a single chip, and therefore a single random variable, Δp_{inter}, can be applied to all transistors [wires] to model the effect of inter-die variation. Consequently, the covariance matrix for each type of spatially correlated parameter is changed by adding to all entries a value of $\sigma^2_{p_{inter}}$, the variance of inter-die parametric variation. Based on the new covariance matrices, the same SSTA methodology can still be applied to compute distribution of chip delay.

3.5.2 Spatially Uncorrelated Parameters

In practice, it is observed that not all process parameters are spatially correlated. For example, the variations of T_{ox} or N_a are independent from transistor to transistor. To model the intra-die variation of a spatially uncorrelated parameter, a separate random variable must be used for each gate [wire] to represent such independence, instead of a single random variable for all gates [wires] in the same grid for the spatial correlated parameters. Consequently, the timing analysis framework introduced in previous sections must be further extended to accommodate the spatially uncorrelated parameters.

As an example, let us consider the case that gate oxide thickness T_{ox} is the only spatially uncorrelated parameter. The idea described here can easily be extended to the case where there is more than one uncorrelated parameter. With inter- and intra-die variations, the variation of T_{ox} for the i^{th} transistor can be expressed as $\Delta T_{ox}^{inter} + \Delta T_{ox,i}^{intra}$, where ΔT_{ox}^{inter} is the random variable representing for the inter-die variation of gate oxide thickness T_{ox}, and $\Delta T_{ox,i}^{intra}$ the intra-die variation of T_{ox} of the i^{th} transistor. Accordingly, the expressions for device [wire] delays are reformulated by substituting $\Delta T_{ox}^{inter} + \Delta T_{ox,i}^{intra}$ for where the random variable for intra-die variation of gate oxide thickness of the i^{th} transistor appears. Since the orthogonal transformations of parameters are performed only on spatially correlated parameters, the variables ΔT_{ox}^{inter} and $\Delta T_{ox,i}^{intra}$ are preserved in the delay expressions of linear combination of principal components and either variable is independent from the principal components and any other random variables in the delay expressions. The timing propagation using the *sum* and *max* operators remains the same, except that after each *sum* or *max* operation, the random variables for intra-die variations of the spatially uncorrelated $\Delta T_{ox,i}^{intra}$ parameters, are

merged into one random variable, so that, for the arrival time at each circuit node, only one independent random variable is kept for all intra-die variations of spatially uncorrelated parameters, similar to the "residual" variances lumping into the independently random part in [80]. That is, all delays and arrival times are in the following form, with an extra independent random term in addition to Equation (3.12):

$$d = d_0 + k_1 \times p'_1 + \cdots + k_m \times p'_m + k_{m+1} \times r, \tag{3.36}$$

where $k_{m+1} \times r$ is the merged independent term and r is a zero mean and unit variance random variable that is uncorrelated and independent with all other random variables.

Although structural correlations can be automatically taken into account using orthogonal transformation on spatially correlated parameters as explained in Section 3.3.4, the structural correlations due to spatially uncorrelated parameters cannot be efficiently dealt with by this methodology, since directly keeping all uncorrelated random variables in the delay form results in a huge number of variables, and merging of these independent random variables during the propagation lose the correlation information. To reduce the inaccuracies caused, one can appeal to the available literature on handling structural correlations in SSTA [6, 27, 53]. In this work, we have ignored the structural correlations caused by the spatially uncorrelated parameters. However, since the structural correlations from spatially correlated parameters are considered, the inaccuracies introduced from this assumption are not significant, as will be demonstrated in Section 3.6.

3.5.3 Distribution of the Minimum of a Set of Gaussians

In circuit performance analysis, computations such as finding the required arrival time (RAT) for long-path analysis, and minimum delay computations for short-path analysis (to check for hold time violations) require the computation of the minimum of a set of delays, which becomes finding the distribution of the minimum of a set of random variables under process variations.

The procedure for calculation of maximum of a set of Gaussians can be utilized to compute the minimum of a set of Gaussian random variables, $d_1 \cdots d_s$. Specifically, $d_{min} = min(d_1, \cdots, d_s)$ can be computed as

$$d_{min} = -max(-d_1, \cdots, -d_s), \tag{3.37}$$

where d_i is a normally distributed random variable and max is the operator introduced is Section 3.3.3.

3.6 Experimental Results

The proposed algorithm was implemented in C++ as the software package, *MinnSSTA*, and tested on the edge-triggered ISCAS89 benchmark circuits by working on the combinational logic blocks between the latches. All experiments were run on a Linux PC with a 2.0GHz CPU and 256MB memory. We experimented with parameters of 100nm technologies on a 2-metal layer interconnect model. The process parameters (Table 3.1) used here are based on predictions from [24, 56].

Since the computation requires physical information about the locations of the gates and interconnects, all cells in the circuit were first placed using the placement tool, Capo [77]. Global routing was then performed to route all the nets in the

Table 3.1: Parameters used in the experiments.

Parameters	L_{eff} (nm)	W_g (nm)	T_{ox} (nm)	N_a ($\times 10^{17} cm^{-3}$) nmos/pmos	W_{int} (nm)	T_{int} (nm)	H_{ILD} (nm)
$\bar{p}$	60.0	150.000	2.500	9.70000/10.04000	150.0	500.0	300.0
$3\sigma_{inter}$	9.0	11.250	0.250	0.72750	15.0	25.0	22.50
$3\sigma_{intra}$	4.5	5.625	0.125	0.36375	7.5	12.5	11.25
$\delta_x x_{max} + \delta_y y_{max}$	4.5	5.625	0.125	0.36375	7.5	12.5	11.25

circuits. Depending on the size of circuit, we divided the chip area into different sizes of grids, so that each grid contains no more than a hundred cells. Due to the lack of access to real wafer data, the covariance matrix for intra-die variations used in this work were derived from the spatial correlation model used in [5] by equally splitting the variance into all levels.

To verify the results of our method *MinnSSTA*, we used Monte Carlo (*MC*) simulations based on the same grid models for comparison. To balance the accuracy and run time, we chose to run 10,000 iterations for the Monte Carlo simulation.

We first present the experimental results assuming that all process parameters are spatially correlated while using fixed values for the spatially uncorrelated process parameters (T_{ox} and N_a). Table 3.2 shows a comparison of the results of *MC* with those from *MinnSSTA*. For each test case, the mean and standard deviation (SD) values for both methods are listed. The results of *MinnSSTA* can be seen to be very close to the *MC* results: the average error is -0.23% for the mean and -0.32% for the standard deviation. In Figure 3.2, for the largest test case s38417, the plots of the PDF and CDF of the circuit delay for both *MinnSSTA* and *MC* methods are provided. It is observed that the curves almost perfectly match each other. This demonstrates the accuracy of the PCA approach for correlated process parameters, including its ability to account for structural correlations.

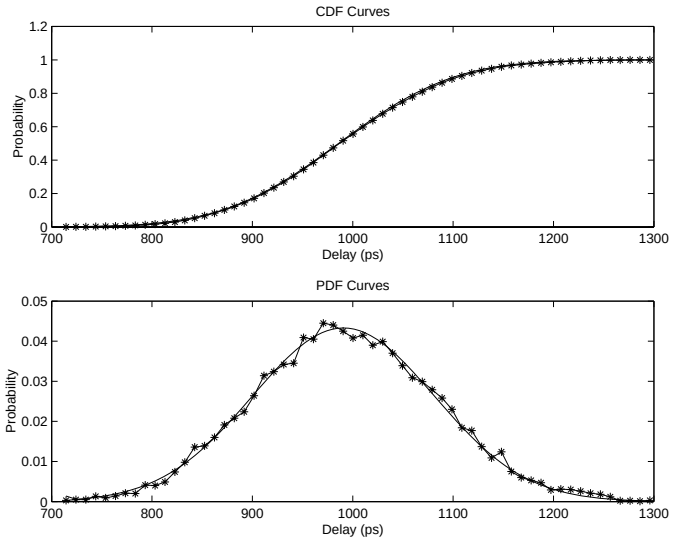


Figure 3.2: A comparison of *MinnSSTA* and *MC* methods (assuming fixed values of T_{ox} and N_a) for circuit s38417. The curve marked by the solid line denotes the results of *MinnSSTA*, while the plot marked by the starred lines denotes the results of *MC*. Note that the differences between the curves are exaggerated because of the high slopes and the fact that the scale does not include the origin, but the mean and the variance of the two are very close to each other, as are the delay points corresponding to 95% and higher timing yields.

Next, the results for considering the variations of the spatially uncorrelated process parameters (T_{ox} and N_a) are given in Table 3.3. On average, the error is 1.06% for the mean value and -4.34% for the standard deviation. In Table 3.7, the 99% and 1% confidence points achieved by *MC* and *MinnSSTA* are also provided and the average errors are -2.46% and -0.99%, respectively. Again, for the largest test case s38417, the PDF and CDF curves of the circuit delay for both *MinnSSTA* and *MC* methods are plotted in Figure 3.3, It can be seen that, at the range of lower and higher circuit delay values, the circuit delay distribution

Table 3.2: Comparison results assuming fixed values of T_{ox} and N_a.

Benchmark	Monte Carlo (MC)		MinnSSTA		$\frac{(MinnSSTA-MC)}{MC}\%$	
Name	Mean(ps)	SD(ps)	Mean(ps)	SD(ps)	Mean	SD
s38417	988.6	91.0	985.8	90.8	-0.28%	-0.22%
s38584	1726.9	153.1	1720.9	151.6	-0.35%	-0.98%
s35932	1165.5	101.6	1162.7	101.3	-0.24%	-0.30%
s15850	1370.2	131.1	1367.2	129.6	-0.22%	-1.14%
s13207	1219.9	116.1	1217.3	116.2	-0.21%	0.09%
s9234	674.6	65.4	673.7	64.8	-0.13%	-0.92%
s5378	413.1	38.5	411.8	38.4	-0.31%	-0.26%
s1196	499.9	45.8	499.3	46.2	-0.12%	0.87%
s27	102.5	9.9	102.3	9.9	-0.20%	0.00%

computed from *MinnSSTA* matches well with that of the Monte Carlo simulation, although there are some deviations in the central portion. As mentioned in Section 3.5.2, some error may be introduced from the structural correlations, which are not handled exactly in the presence of uncorrelated intra-die components. Based on our analysis of the experiments, we find that the cause for the small error that is introduced here is primarily because our implementation does not handle structural correlations between the uncorrelated variables. We believe that, by appending into the existing framework an algorithm that handles structural correlation [6, 27, 53], the error of the results in Table 3.3 can be further reduced.

In Table 3.3, the CPU times for both methods are provided. To show that the PCA steps require very little run time, the run time for this part is also listed; however, as pointed out earlier, this can be considered a preprocessing step that is carried out once for each technology, and its cost need not be considered in the computation. We can see that the CPU time of *MinnSSTA* on all test cases is very fast. The circuit with the longest run time, s35932, was analyzed in only about 500

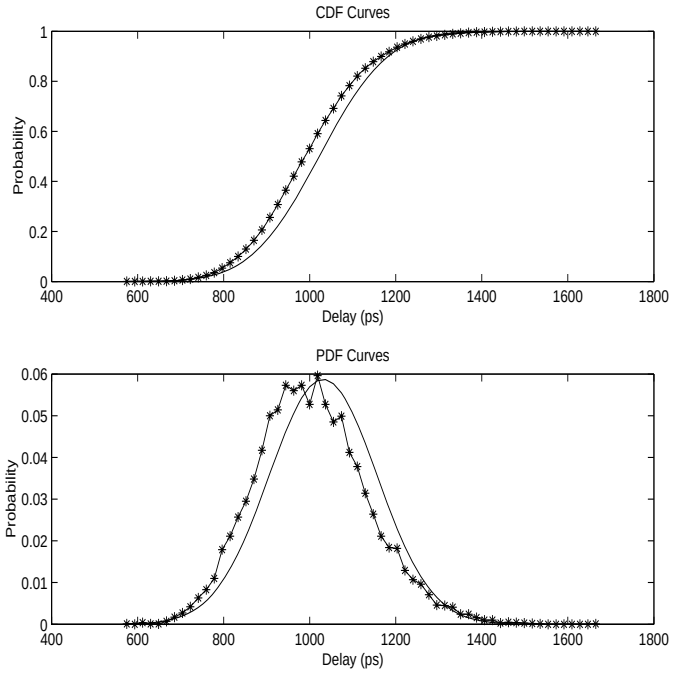


Figure 3.3: A comparison of *MinnSSTA* and *MC* methods for circuit s38417, considering all sources of variation, some of which are spatially correlated and some of which are not. The curve marked by the solid line denotes the results of *MinnSSTA*, while the plot marked by the starred lines denotes the results of *MC*.

seconds, while the *MC* simulation required over 15 hours.

In the proposed approach, in order to make the computed value of standard deviation of d_{max} the same as that of the approximated linear expression, the coefficients of process parameters in the linear expression are normalized by the ratio of the standard deviation of d_{max} (namely, $\sigma_{d_{max}}$) to that of the linear expression s_0. In Table 3.4, the statistics of this ratio for all testcases are listed, including the mean, standard deviation, minimum and maximum values of the ratio and the probability of the ratio falls into each given range. In general, the higher the ratio, the larger the error for estimating d_{max}, and thus the less accurate for estimating

Table 3.3: Comparison results of the proposed method and Monte Carlo simulation method.

Benchmark			Monte Carlo (MC)			MinnSSTA				$\frac{(MinnSSTA-MC)}{MC}$%	
Name	#Cells	#Grids	Mean(ps)	SD(ps)	CPU(s)	Mean(ps)	SD(ps)	CPU(s)	PCA(s)	Mean	SD
s38417	23815	256	995.6	130.3	21005	1022.0	125.4	406.11	0.15	2.65%	-3.76%
s38584	20705	256	1738.4	226.4	24039	1798.2	215.6	460.36	0.15	3.44%	-4.77%
s35932	17793	256	1214.7	161.8	53922	1251.2	144.7	505.71	0.15	3.00%	-10.57 %
s15850	10369	256	1388.2	178.9	8856	1397.8	172.1	175.96	0.15	0.69%	-3.80%
s13207	8260	256	1230.7	158.8	9060	1239.7	154.9	172.62	0.15	0.73%	-2.46%
s9234	5825	64	688.6	90.6	5346	690.6	85.2	32.23	0.02	0.29%	-5.96%
s5378	2958	64	421.1	54.3	3907	420.8	51.8	27.41	0.02	-0.07%	-4.60%
s1196	547	16	505.9	66.0	781	502.7	64.4	1.51	0.01	-0.63%	-2.42%
s27	13	4	103.6	13.7	9	103.0	13.6	0.00	0.00	-0.58%	-0.73%

Table 3.4: Statistics of ratio of standard deviation of accurate value $\sigma_{d_{max}}$ to s_0 of the linear expression.

Circuit	Ratio of $\sigma_{d_{max}}$ to s_0				Probability of the ratio in each range				
Name	mean	stdev	minimum	maximum	< 1	$= 1$	$(1, 1.01)$	$[1.01, 1.1]$	> 1.1
s38417	1.0031	0.0051	≈ 1	1.0262	0.0004	0.3246	0.5582	0.1168	0
s38584	1.0037	0.0054	≈ 1	1.1804	0.0023	0.4124	0.1700	0.0001	0
s35932	1.0120	0.0278	≈ 1	1.1583	0.0022	0.2883	0.4290	0.2350	0.0454
s15850	1.0018	0.0033	≈ 1	1.0233	0.0034	0.4029	0.5538	0.0398	0
s13207	1.0028	0.0048	≈ 1	1.0260	0.0008	0.3256	0.5843	0.0893	0
s9234	1.0017	0.0035	1	1.0209	0	0.3825	0.5636	0.0538	0
s5378	1.0012	0.0023	1	1.0289	0	0.4310	0.5563	0.0126	0
s1196	1.0007	0.0021	≈ 1	1.0150	0.0021	0.7068	0.2764	0.0148	0
s27	1.0006	0.0014	1	1.0030	0	0.8	0.2000	0	0

the circuit delay distribution using the proposed approach. For example, the testcase $s35932$ has the highest probability of 0.045 for the ratio to be greater than 1.1, and also has the largest errors predicting the circuit mean and standard deviation. Over all testcases, the average value of the ratio is 1.003, which is a reasonably small number so that the accuracy of the proposed statistical SSTA should not be affected significantly by this normalization step.

To further verify the applicability of the proposed algorithm, we have demonstrated it on a path-balanced circuit whose topology is a binary tree of depth 10.

Table 3.5: Experimental results on a binary tree circuit of depth-10.

Approach	Mean(ps)	SD(ps)	99% Point(ps)	1% Point(ps)
MC	669.8	86.2	894.8	486.3
MinnSSTA	666.2	80.8	854.0	478.3
$\frac{(MinnSSTA-MC)}{MC}\%$	−0.54%	−6.26%	−4.56%	−1.65%

Table 3.6: Comparison of timing analysis with and without spatial correlations.

Benchmark	Anal. w/ corr. (MC)		Anal. w/o corr. (MCNoCorr)		$\frac{(MC-MCNoCorr)}{MCNoCorr}\%$	
Name	Mean(ps)	SD(ps)	Mean(ps)	SD(ps)	Mean	SD
s38417	995.6	130.3	996.7	98.7	0.11%	-24.25%
s38584	1738.4	226.4	1741.9	180.5	0.20%	-20.27%
s35932	1214.7	161.8	1253.6	140.0	3.20%	-13.47%
s15850	1388.2	178.9	1393.8	121.9	0.40%	-31.86%
s13207	1230.7	158.8	1233.8	110.2	0.25%	-30.60%
s9234	688.6	90.6	691.9	61.9	0.48%	-31.68%
s5378	421.1	54.3	424.7	38.2	0.85%	-29.65%
s1196	505.9	66.0	507.6	48.8	0.34%	-26.06%
s27	103.6	13.7	103.7	10.2	0.10%	-25.55%

Table 3.5 lists the results achieved by *MinnSSTA* and (*MC*). The errors obtained are -0.54% for the mean and -6.26% for the standard deviation; -4.56% and -1.65% for the 99% and 1% confidence point, respectively. This shows that the proposed approach can predict the timing yield well, even for path-balanced circuits.

To show the importance of considering spatial correlations, we consider the difference between performing statistical timing analysis while considering spatial correlation and while ignoring it. Since this is a comparison to determine why spatial correlations are important, the CPU time is not a consideration. Therefore, we run another set of Monte Carlo simulations (*MCNoCorr*) on the same set of benchmarks, this time assuming zero correlations among the devices and wires on the chip. The comparison between the data is shown in Table 3.6. It can be observed that although the mean values are close, the variances of the uncorrelated

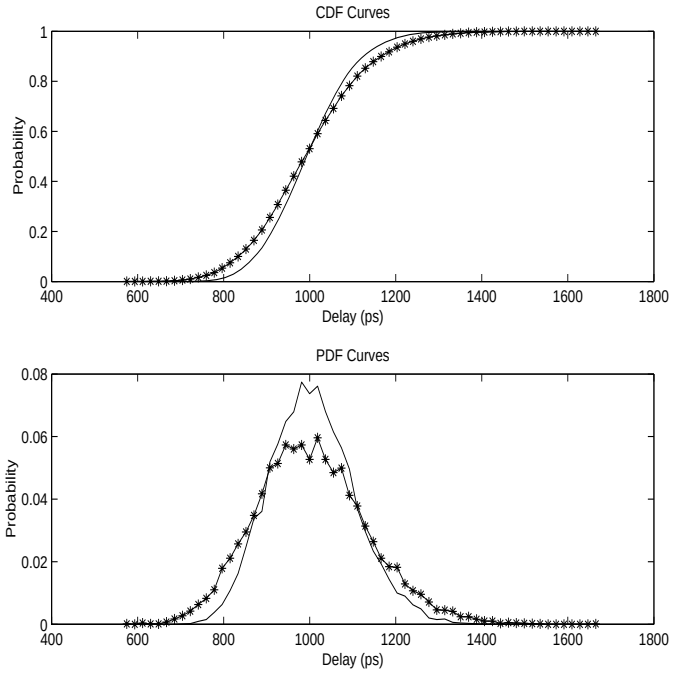


Figure 3.4: A comparison of SSTA with and without considering spatial correlations, under Monte Carlo analysis, for circuit s38417. The curve marked by the solid line denotes the case where spatial correlations are ignored, while the curve with the starred lines denotes the results of incorporating spatial correlations; this is identical to the curve in Figure 3.3.

cases (*MCNoCorr*) are much smaller than the correlated cases (*MC*). On average, the standard deviation of the correlated case increases by 25.93%. Again, we plot the PDF and CDF curves of both simulations for circuit s38417 in Figure 3.4. It is seen that the CDF and PDF curves of *MCNoCorr* deviate significantly from those of *MC*. In other words, statistical timing analysis without considering correlation may incorrectly predict the real performance of the circuit and could even overestimate the performance of the circuit. This underlines the importance of developing efficient SSTA methods that can incorporate spatial correlations.

Table 3.7: Comparison of 99% and 1% confidence point.

Bench.	MC		MinnSSTA		$\frac{(MinnSSTA-MC)}{MC}$%		MPC		$\frac{(MPC-MC)}{MC}$%	
mark Name	99% Pt. (ps)	1% Pt. (ps)	99% Pt. (ps)	1% Pt. (ps)	99% Pt. (ps)	1% Pt. (ps)	Worst Case	Best Case	Worst Case	Best Case
s38417	1333.3	722	1313.6	730.4	-1.48%	1.16%	1758.1	522.1	31.86%	-27.69%
s38584	2310.3	1261.3	2299.5	1296.9	-0.47%	2.82%	3056.0	915.4	32.28%	-27.42%
s35932	1635.2	882.3	1587.6	914.8	-2.91%	3.68%	2051.2	613.0	25.44%	-30.52%
s15850	1844.8	1012.9	1797.9	997.7	-2.54%	-1.50%	2442.9	725.2	32.42%	-28.40%
s13207	1629.9	893.1	1599.8	879.6	-1.85%	-1.51%	2175.4	646.6	33.47%	-27.60%
s9234	922.6	499.7	888.7	492.5	-3.67%	-1.44%	1207.3	359.7	30.86%	-28.02%
s5378	559.9	308.9	541.2	300.4	-3.34%	-2.75%	736.6	219.2	31.56%	-29.04%
s1196	673.4	370.4	652.4	353.0	-3.12%	-4.70%	874.2	265.8	29.82%	-28.24%
s27	138.4	74.9	134.6	71.4	-2.75%	-4.67%	179.3	55.6	29.55%	-25.77%

As an alternative, we consider the option of using multiple process corners (*MPC*) for these experiments, where the circuit delays are evaluated at all possible corners of process parameter values at $\mu \pm 3 \cdot \sigma$, where μ is the mean and σ the standard deviation of the process parameter. Table 3.7 compares the worst-case and best-case delays obtained at exhaustive process corners using the *MPC* method, with the 99% and 1% confidence point delay achieved from the Monte Carlo simulation (*MC*) accordingly. On average, the *MPC* approach overestimates the worst-case delay of circuit by 30.81% and underestimates the best-case delay by 28.08%. These results also emphasize the importance of considering spatial correlations during SSTA, as is done by our algorithm.

3.7 Conclusion

In this chapter, we have proposed an algorithm for performing SSTA, considering spatial correlations related to intra-die process variations. We show that performing statistical timing analysis while ignoring spatial correlations may not be adequate to

predict the circuit performance correctly, and that fast and accurate SSTA methods, such as ours, that incorporate spatial correlations are essential. An analysis of the complexity shows it to be reasonable, and like conventional STA, it is linear in the number of gates and interconnects. The penalty that is paid here is that unlike deterministic STA, it is also linear in the number of grid squares. As a trivial extension of maximum of delays, the computation for the distribution of minimum of delays is also provided.

The current algorithm is limited by the following: it assumes that the probability distributions of all process variations are Gaussian and the distribution of gate [wire] delays have linear dependency on the variations of process parameters. In Chapter 4, we will extend the method to solve the problem of statistical timing analysis that can incorporate non-Gaussian process parameter variations and nonlinear delay dependencies.

Chapter 4

Incorporating Non-Gaussian Distributed Process Parameters and Nonlinear Delay Functions

In this chapter, we present a general framework and an efficient method of block-based SSTA that can deal with process variations with non-Gaussian distributions, and/or delay functions with nonlinear dependencies on process parameter variations. We extend techniques for evaluating the *sum* and *max* functions in SSTA from the linear, Gaussian case of Chapter 3, to the nonlinear, non-Gaussian case. The proposed approach is shown to be accurate and efficient in predicting timing characteristics and yield of circuit.

4.1 Introduction

In Chapter 3, we proposed an efficient method for timing analysis under process variations, under the assumption that where all process variations have or can be approximated by Gaussian distributions, and all delays have linear sensitivities to the process parameters.

There are two limitations to this approach. First, although some types of distributions can be approximated by a Gaussian, others may display asymmetric types of distributions (e.g., lognormal distributions), or symmetric types of non-Gaussian distributions (e.g., uniform distributions) that cannot be well-approximated by a Gaussian. For example, via resistance is known to have an asymmetric probability distribution. A second issue is related to the use of a first-order Taylor expansion to approximate a delay function as a linear function of the variations of process parameters. The linear approximation can only be justified under the assumption that variations are small. With technology scaling, as the percentage change in process variations becomes larger, delays may show nonlinear dependencies on some sources of variations, so that a linear approximation is not accurate enough. For instance, the dependence of delay on the transistor channel length, L_{eff}, is essentially nonlinear, and assuming a linear dependency can result in significant inaccuracies under large variations [43]. Therefore, it is desirable to develop SSTA techniques that can deal with non-Gaussian-distributed process parameters and/or nonlinear effects on gate [wire] delays[1], in order to obtain sufficiently accurate results for analyzing the timing yield.

[1]For conciseness, in the remainder of the thesis, we will use the term "non-Gaussian parameter" to refer to a non-Gaussian-distributed process parameter, and "nonlinear function parameter" to a process parameter whose variation has nonlinear effects on delays.

The task of developing an SSTA technique that is capable of handling arbitrary non-Gaussian and/or nonlinear function parameter is very challenging. As described in Section 3.1, approaches for SSTA can be classified into continuous methods and discrete methods. For contininuous methods, when arbitrary non-Gaussian-distributed or nonlinear function parameters are involved, the task of developing analytic forms for SSTA operations is nontrivial. Discrete methods can represent more general probability distributions, but during SSTA event propagation, it is difficult for them to maintain correlation information due to the global sources of variations among the delays/arrival times. Existing discrete methods are limited to cases when all gate [wire] delays are independent, which is not practical.

In this chapter, we present a parameterized block-based SSTA approach, and one of the challenging tasks herein lies in computing the *sum* and *max* functions while keeping the correlations of delays due to global sources of variations. A parameterized SSTA method models gates or wires delays D as explicit functions of variations of the Δp_i process parameters. Using this representation, parameterized SSTA approach propagates and computes circuit timing characteristics A (such as arrival and required arrival times, delay, timing slack) as functions of the same set of process parameters, and thus the distribution of circuit delay is also expressed as a function of process parameters. Since the parameterized forms of delays/arrival times lend themselves easily to the computation of correlations, delay correlations due to global variation sources can be well preserved during timing propagation. Another advantage of parameterized SSTA is that explicit dependencies of circuit timing characteristics on process parameters can be obtained with the analysis, which can be useful not only for predicting the probability distributions of circuit timing and manufacturing yield, but also for circuit optimization, improving robustness of the design, and manufacturing line tailoring. The procedure described

in Chapter 3 is an example of a parameterized SSTA algorithm, while nonparameterized techniques [11, 27, 45, 58] are less attractive because they do not relate circuit timing variations to changes in the underlying process parameters.

There have been some recent works on parameterized block-based SSTA that are capable of handling non-Gaussian and/or nonlinear function parameters. The method in [4] can handle linear functions, but non-Gaussian process parameters: it proceeds by computing the upper bounds of parameterized arrival times during the propagation, and applying a heuristic method to improve the quality of the bound by propagating multiple arrival times. The works of [82, 83] use quadratic timing models instead of linear models to capture the nonlinear dependencies of gate/wire delays and arrival times on Gaussian-distributed process parameters. In [82], the computation of the *max* function is simplified by converting the quadratic forms of delay functions to those without cross terms, using orthogonalization, and the quadratic approximation of the nonlinear *max* operation is performed via moment matching. In [83], the result of the *max* operation is computed using analytical formulas for handling the case with only Gaussian and linear function parameters, assuming the delays in quadratic forms are Gaussian random variables. The SSTA framework proposed in [40] can handle arbitrary distributed process parameters and nonlinear delay dependencies, by modeling gate delays and arrival times as polynomials using Taylor series expansions on the process parameters, and using regressions to approximate the result of *max* back to a polynomial. However, as regressions are required in the framework, the run-time is rather slow and the method is not scalable to large scale circuits.

In this chapter, we will present an efficient parameterized block-based SSTA method that can handle arbitrarily distributed parameters and arbitrary nonlinear delay dependencies on process parameter, by extending the *sum* and *max* functions

in SSTA approach for Gaussian and nonlinear function parameters. The chapter is organized as follows. The framework for the parameterized SSTA approach for handling Gaussian-distributed process parameters and linear sensitivities of delays to process parameters is first summarized in Section 4.2. A generalized framework for handling arbitrarily distributed parameters and arbitrary nonlinear delay dependencies on process parameters, and an efficient method to implement the framework are then provided in Section 4.3. Finally, the experimental results will be shown in Section 4.4.

4.2 Framework for Handling Gaussian and Linear Function Parameters

In this section, we will summarize the framework for parameterized block-based SSTA that can handle Gaussian and linear function parameters. We will then generalize this framework in the next section to handle arbitrarily distributed process parameters and arbitrary delay functions.

Note that the SSTA approaches proposed in Chapter 3, first published in [17], and the work in [80], are both parameterized block-based methods using similar frameworks. In both works, any gate or wire delay is represented as a linear function of process variations, and this representation is referred to as a first-order canonical form in [80]:

$$A = a_0 + \sum_{i=1}^{n} a_i \cdot \Delta X_i + a_{n+1} \cdot \Delta R_a \tag{4.1}$$

Here, a_0 is the mean or nominal delay, and $\Delta X_i = X_i - \hat{X}_i$ is variation of process parameter X_i, centralized by subtracting its mean value $\hat{X}_i$. Each ΔX_i represents

for a global source of variation that has a global effect on all delays, and is modeled as a Gaussian random variable $N(0, \sigma_{X_i})$; all ΔX_i variables are mutually independent. The coefficient a_i is the sensitivity of delay to X_i, and ΔR_a is the variation of local uncertainty that only affects the delay locally, and is modeled as a normalized Gaussian random variable that is independent of all other sources of variations. The sensitivity of the delay to R_a is given by a_{n+1}.

This first-order canonical form is equivalent to the delays/arrival times expressed by Equation (3.36) that are employed in the proposed SSTA approach in Chapter 3: ΔX_i and ΔR_a correspond to the principal component p'_i and the normalized independent Gaussian random variable r in expression (3.36), respectively. In this chapter, we will follow the notations and terms in Equation (4.1) for presentation.

The circuit timing characteristics are computed by propagating signals from the source node to the sink using two basic operations: propagation of arrival time through a timing edge and computation of the latest arrival time at a node. The first operation requires the computation of the *sum* function $C = A + B$, where A and B are each in the first-order canonical form of (4.1). The resulting sum C can also be written in canonical form, and can be computed as follows:

$$c_0 = a_0 + b_0 \tag{4.2}$$

$$c_i = a_i + b_i \qquad (i = 1, \cdots, n) \tag{4.3}$$

$$c_{n+1} = \sqrt{a_{n+1}^2 + b_{n+1}^2} \tag{4.4}$$

The second operation requires the computation of the max function $C = max(A, B)$. A first-order canonical form C_{app} is used to approximate C, where each coefficient,

c_i, is computed as follows:

$$\begin{aligned} c_0 &= \mu_{max(A,B)} = a_0 \cdot \Phi(\beta) + b_0 \cdot \Phi(-\beta) + \alpha \cdot \varphi(\beta) && (4.5) \\ c_i &= cov(C_{app}, \Delta X_i)/\sigma^2_{X_i} = cov(max(A,B), \Delta X_i)/\sigma^2_{X_i} \\ &= a_i \cdot \Phi(\beta) + b_i \cdot \Phi(-\beta) \qquad (i = 1, \cdots, n) && (4.6) \end{aligned}$$

where α, β, $\Phi(.)$ and $\varphi(.)$ are as defined in Section 3.3.3.

The variance of $max(A, B)$ can be computed by:

$$\sigma^2_{max(A,B)} = (\sigma^2_A + a^2_0) \cdot \Phi(\beta) + (\sigma^2_B + b^2_0) \cdot \Phi(-\beta) + (a_0 + b_0)\theta\varphi(\beta) - c^2_0 \qquad (4.7)$$

As mentioned in Chapter 3, to diminish the difference between the exact values of $\sigma^2_{max(A,B)}$ and $\sum_{i=1}^{n} c_i$, we can either normalize the coefficients $c_1, \cdots, c_n$, or lump the difference to an independent random variable. If the latter option is applied, as in the work of [80], the approximation C_{app} theoretically predicts the results of a linear regression: the exact mean and variance of $max(A, B)$ are matched, and the coefficients $c_1, \cdots, c_n$ are obtained by minimizing the expected value of the squared error. The approximation here also makes statistical sense, since it minimizes the error in regions of higher probability more than in regions of lower probability.

In fact, the approximation C_{app} for $max(A, B)$ can be interpreted in several ways.

- Computing C_{app} so that it is a linear regression to $\Delta X_1, \cdots, \Delta X_n$: $C_{app} = c_0 + c_1\Delta X_1 + \cdots + c_n\Delta X_n + c_{n+1}\Delta R_c$, where each c_i is a constant and $c_{n+1}\Delta R_c$ is the error term, with R_c as a normalized Gaussian random variable.

  Using a least squared regression, the mean of the squared error is:

$$SqrErr = \int_{\Delta\overrightarrow{X}} (max(A,B) - C_{app})^2 \cdot p(\Delta\overrightarrow{X}) d\Delta\overrightarrow{X} \qquad (4.8)$$

where $p(\Delta\overrightarrow{X})$ is the joint PDF function of $\Delta\overrightarrow{X} = \{\Delta X_1, \cdots, \Delta X_n\}$.

The minimum squared error can be achieved when $\frac{d(SqrErr)}{dc_i} = 0$, for $i = 1, \cdots, n$ that gives:

$$E[max(A,B)] = E[C_{app}] \tag{4.9}$$

$$E[max(A,B) \cdot \Delta X_i] = E[C_{app} \cdot \Delta X_i] \quad (\text{for } i = 1, \cdots, n) \tag{4.10}$$

where $E[.]$ is the symbol for the mean of a random variable.

Equation (4.9) matches c_0 to the mean of $max(A,B)$. Given (4.9), expression (4.10) can be written as $cov(max(A,B), \Delta X_i) = cov(C_{app}, \Delta X_i)$, and since $cov(C_{app}, \Delta X_i) = c_i \cdot \sigma^2_{X_i}$, the values of each c_i can directly be computed by $c_i = cov(max(A,B), \Delta X_i)/\sigma^2_{X_i}$.

In the case where each ΔX_i is a Gaussian random variable, the values of $E[max(A,B)]$ and each $cov(max(A,B), \Delta X_i)$ can be analytically computed using Clark's result [22]. Therefore, the approximation C_{app} can be obtained analytically, and the coefficient c_{n+1} of the error term can be obtained by matching the exact value of variance of $max(A,B)$, which, again, can be computed by Clark's result.

- Computing C_{app} as a linear regression to A and B: $C_{app} = s + pA + qB + eR_c$ where eR_c is the error term, with R_c as a normalized Gaussian random variable, and s, p, q and e being constants.

  Similarly, the minimum squared error of the approximation is obtained when:

$$\begin{aligned} E[max(A,B)] &= E[C_{app}] \\ cov(max(A,B), A) &= cov(C_{app}, A) \\ cov(max(A,B), B) &= cov(C_{app}, B) \end{aligned} \tag{4.11}$$

By solving the equations above for p, q and s, we get:

$$\begin{aligned} p &= \frac{cov(max(A,B),B)\cdot cov(A,B) - cov(max(A,B),A)\cdot \sigma_B^2}{cov^2(A,B) - \sigma_A^2\cdot\sigma_B^2} \\ q &= \frac{cov(max(A,B),A)\cdot cov(A,B) - cov(max(A,B),B)\cdot \sigma_B^2}{cov^2(A,B) - \sigma_A^2\cdot\sigma_B^2} \\ s &= E[max(A,B)] \end{aligned} \tag{4.12}$$

If A and B are Gaussian random variables, analytical forms for $cov(max(A,B),A)$ and $cov(max(A,B),B)$ are available, and thus the values p, q and s can be computed for the regression, while the error term eR_c can be obtained by matching the exact value of variance of $max(A,B)$ from Clark's result in [22].

- Using the concept of tightness probability to approximate $max(A,B)$.

  Given random variables A and B, the tightness probability T_A of A is defined as the probability that A is greater than B: $T_A = Prob(A > B)$. The tightness probability T_B of B is similarly defined, and $T_A + T_B = 1$. In the work of [80], the concept of tightness probability is utilized in the linear approximation of $max(A,B)$. If C_{app} is in a first-order canonical form to approximate $max(A,B)$, then the values of each c_i, for $i = 1, \cdots, n$ can be computed by:

$$c_i = T_A \cdot a_i + (1 - T_A)\cdot b_i \tag{4.13}$$

  The values of c_0 and c_{n+1} in C_{app} can be obtained by matching the exact mean and variance of the max function.

  Intuitively, the use of tightness probabilities T_A and $T_B = (1 - T_A)$ as weighting coefficients can be justified by the reasoning that the larger the tightness probability T_A, the more likely that $max(A,B)$ equals A. Figure 4.1 shows the approximation of the maximum of two canonical forms A and B with one

process variable ΔX. In the figure, A and B are shown with thick dashed lines, and the exact maximum is a piecewise linear function $C = max(A, B)$ consisting of two pieces shown with bold solid lines. The first-order approximation C_{appr} is a line with a slope more than the slope of line A and less than the slope of line B. The line C_{appr} is closer to line A than to line B, because the probability of $A > B$ is larger than that of $A < B$ by comparing the corresponding intervals of ΔX. In case of multiple varying process parameters, we have a similar picture but with hyperplanes instead of lines.

Theoretically, if A and B are both Gaussian random variables in first-order canonical forms, the use of tightness probabilities in the approximation of $max(A, B)$ happens to be able to predict the result of a linear regression. This can be verified, for instance, by the formulas for linear regression in (4.12). In (4.12), if Clark's results are used, we get $p = \Phi(\beta)$ and $q = \Phi(-\beta)$, and thus each coefficient c_i of C_{app} can be computed by:

$$c_i = p \cdot a_i + q \cdot b_i = \Phi(\beta) \cdot a_i + \Phi(-\beta) \cdot b_i \tag{4.14}$$

Note that the value of $\Phi(\beta)$ is in fact the probability that the Gaussian random variable $A-B$ is greater than zero, which is the tightness probability of A, and similarly $\Phi(-\beta)$ is the tightness probability of B. Therefore, the expression (4.13) has exactly same form as the linear regression (4.14.)

The interpretations above provide three possible frameworks for statistical approximation of $max(A, B)$ in parameterized SSTA for handling process variables with Gaussian distributions and linear effects on delays. Any of the frameworks can be further extended to handle process variables with non-Gaussian distribution and nonlinear delay effects. In next section, to incorporate non-Gaussian and/or nonlinear function process variations, the ideas of approximating $max(A, B)$ with

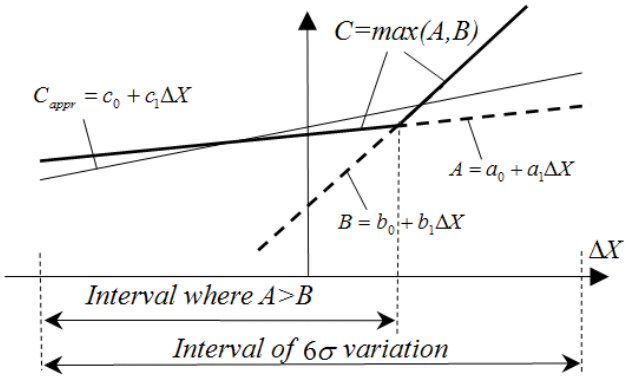


Figure 4.1: Linear approximation of maximum of two canonical forms A and B, where $A = a_0 + a_1\Delta X$ and $B = b_0 + b_1\Delta X$. Since ΔX is Gaussian-distributed, only the range from $[-3\sigma, 3\sigma]$ is illustrated. The two-piece bold solid lines C=max(A,B) shows the exact maximum of A and B. The dotted line pointed to by $C_{appr} = c_0 + c_1\Delta X$ is the approximation of the max function.

the concept of tightness probabilities will be extended. However, in this case, the approximation is a heuristic, rather than an approach that can compute the exact same result of the linear regression of the *max* function.

4.3 Framework for Handling Non-Gaussian and/or Non-linear Function Parameters

In this section, we present a generalized framework and an efficient parameterized SSTA method that can handle arbitrarily distributed process parameters and arbitrary delay functions: The first-order canonical form is first extended to a generalized canonical form in order to incorporate non-Gaussian-distributed parameters and nonlinear delay function parameters. The *sum* and *max* functions are then

extended to variables in the generalized canonical forms, and an efficient method that can compute these functions are described in the following sections.

4.3.1 A Generalized Canonical Form for the Delay

A generalized canonical form of gate or wire delay is defined by extending the form of (4.1) as follows:

$$A = a_0 + \sum_{i=1}^{n_{LG}} a_{LG,i} \cdot \Delta X_{LG,i} + f_A(\Delta X_N) + a_{n+1} \cdot \Delta R_a \tag{4.15}$$

Here a_0 is the mean value of the delay, $\Delta X_{LG} = \{X_{LG,1}, X_{LG,2}, \cdots, X_{LG,n_{LG}}\}$ is the set of random variables for the global sources of variation that are both Gaussian-distributed and have linear effects on delay, and n_{LG} is number of such types of variations. The sensitivity of the delay to $\Delta X_{LG,i}$ is given by $a_{LG,i}$. We also define a set of random variables, of cardinality n_{NLG}, $\Delta X_N = \{\Delta X_{N,1}, \Delta X_{N,2}, \cdots X_{N,n_{NLG}}\}$. The elements of this set correspond to the global sources of variations that are non-Gaussian-distributed or have nonlinear effects on the delay, and f_A is a function describing the dependence of the delay on non-Gaussian and nonlinear function parameters, with a mean value that is normalized to zero. Finally, ΔR_a is a normalized Gaussian parameter that represents local sources of variations, and a_{n+1} is its sensitivity to the delay.

The generalized canonical form differs from the original first-order canonical form of delay only in the term $f_A(\Delta X_N)$ that describes dependencies of A on non-Gaussian and nonlinear function parameters. For convenience, this term is referred to as a non-Gaussian nonlinear term in this chapter. Note that f_A can be either a nonlinear function of non-Gaussian-distributed process parameters, or a linear function of non-Gaussian process parameters, or a nonlinear function of Gaussian

process parameters. The function f_A can be a function of arbitrary type, and the non-Gaussian parameters can have any arbitrary probability density function. For numerical computations, nonlinear functions and non-Gaussian distributions can be specified by tables.

4.3.2 The Computation of the *sum* Function

As in the case for first-order canonical forms, it is straightforward to compute the *sum* function for two random variables, each specified in generalized canonical form. If $C = A + B$, where A and B are both in generalized canonical form, then C can also be expressed in a generalized canonical form, with its coefficients specified by:

$$\begin{aligned} c_0 &= a_0 + b_0 \\ c_{LG,i} &= a_{LG,i} + b_{LG,i} \qquad (1 < i < n_{LG}) \\ f_c(\Delta X_N) &= f_A(\Delta X_N) + f_B(\Delta X_N) \end{aligned} \tag{4.16}$$

The computation of c_0 and each $c_{LG,i}$ is simple. The term $f_c(\Delta X_N)$ is obtained by computing the sum of the non-Gaussian nonlinear terms of A and B. In practice, this can be computed by numerically summing the tables describing $f_A(\Delta X_N)$ and $f_B(\Delta X_N)$.

4.3.3 The Computation of the *max* Function

It is necessary to use an approximation in computing the *max* of two random variables, each specified in generalized canonical form. In order to preserve the correlations of delays, a random variable C_{app} in generalized canonical form is used to approximate $C = max(A, B)$. The framework introduced in Section 4.2 for

computing C_{app} can be applied here, by using the concept of tightness probability:

$$\begin{aligned} c_0 &= E[max(A,B)] \\ c_{LG,i} &= T_A a_{LG,i} + (1-T_A) b_{LG,i}, \quad for\ 1 < i < n_{LG} \\ f_c(\Delta X_N) &= T_A f_A(\Delta X_N) + (1-T_A) f_B(\Delta X_N) \end{aligned} \tag{4.17}$$

As in the case for first-order canonical form, this approximation for the maximum of two generalized canonical forms is a linear approximation: c_0 is matched with the exact mean value of $C = max(A, B)$; C_{app} is a linear combination of A and B using the tightness probabilities, where the coefficient $c_{LG,i}$ is computed as a linear combination of coefficients $a_{LG,i}$ and $b_{LG,i}$, and the non-Gaussian nonlinear term f_C as a linear combination of functions f_A and f_B, weighted by the corresponding tightness probabilities T_A and T_B, respectively. The sensitivity coefficient c_{n+1} for the local independent source of variations is computed so as to make the variance of C_{appr} equal to the variance of the exact maximum $C = max(A, B)$, where the exact variance σ_C^2 is expressed through the mean and the second moment as:

$$\sigma_C^2 = E[max(A,B)^2] - (E[max(A,B)])^2 \tag{4.18}$$

Figure 4.2 graphically shows the interpretation of a linear approximation for the maximum of generalized canonical forms that depend only on one nonlinear function parameter. The canonical forms for A and B are shown using thick dashed curves in the figure, and the exact maximum $C = max(A, B)$ is shown using a bold solid curve. The approximation of the maximum, C_{appr}, is represented by a solid thin curve: here, the curve of C_{appr} is closer to curve A, because, as can be observed in Figure 4.2, $max(A, B)$ is more often equal to A than to B; in other words, A has a higher probability of being the maximum.

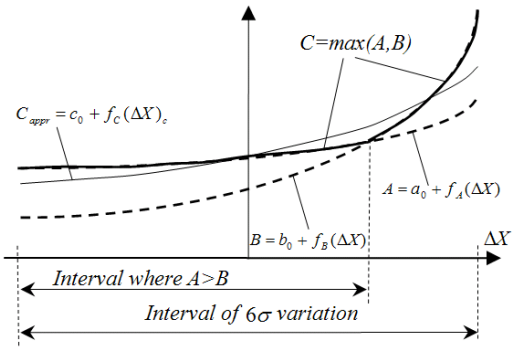


Figure 4.2: Approximation of the maximum of two generalized canonical forms A and B, where $A = a_0 + f_A(\Delta X)$ and $B = b_0 + f_B(\Delta X)$. The figure shows the range of ΔX from $[-3\sigma, 3\sigma]$ as ΔX is Gaussian-distributed. The bold solid curve $C = max(A, B)$ is the exact maximum of A and B. The dotted line pointed to by $C_{appr} = c_0 + f_C(\Delta X)$ is the approximation of the max function.

Finding the approximation for the maximum of two generalized canonical forms requires the computation of the tightness probability T_A, the mean $E[max(A, B)]$ and the second moment $E[(max(A, B))^2]$ of $max(A, B)$ that are defined as follows:

$$\begin{aligned} T_A &= Prob(A > B) \\ &= \int_{A>B} p(\Delta X_N, \Delta X_{LG}, \Delta X_a, \Delta X_b) d\Delta X_N d\Delta X_{LG} d\Delta X_a d\Delta X_b \end{aligned} \tag{4.19}$$

$$E[max(A, B)] = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} max(A, B) p(\Delta X_N, \Delta X_{LG}, \Delta X_a, \Delta X_b) d\Delta X_N d\Delta X_{LG} d\Delta X_a d\Delta X_b \tag{4.20}$$

$$E[(max(A, B))^2] = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} (max(A, B))^2 p(\Delta X_N, \Delta X_{LG}, \Delta X_a, \Delta X_b) d\Delta X_N d\Delta X_{LG} d\Delta X_a d\Delta X_b \tag{4.21}$$

where $p(\Delta X_N, \Delta X_{LG}, \Delta X_a, \Delta X_b)$ is the joint probability density function of all process parameter variations.

If the vector of ΔX_N is empty, then the computations regress to the maximum of two first-order canonical forms, which can be computed analytically in a very

efficient way. However, when there are non-Gaussian probability distributed or nonlinear function parameters, simple analytical formulas may not exist for the maximum of two generalized canonical forms. In the remainder of this section, we will focus mainly on the computation of tightness probability, the mean and the second moment for the *max* function.

Computations of Tightness Probability, Mean and Second Moment

The computations of tightness probability, mean and second moment for the *max* function involve the evaluations of the integrals in (4.19), (4.20) and (4.21) which can be very hard to compute analytically for arbitrary non-Gaussian process parameter PDFs and arbitrary nonlinear functions, f_A. The obvious way to solve this problem is to apply a numerical technique, but this results in losing the desired computational efficiency. In this section, we present a combined approach that processes Gaussian and linear function parameters analytically, and uses a numerical technique only for non-Gaussian or nonlinear function parameters. The method is efficient for realistic cases where most sources of variations can be captured accurately enough by Gaussian distributions and linear delay functions, and only a few of them demonstrate significant nonlinear behavior or non-Gaussian distribution. Therefore, as will be illustrated in the experimental results section, the proposed technique does not reduce the efficiency of dealing with Gaussian and linear function parameters, and can handle additionally up to 7 to 8 non-Gaussian and/or nonlinear function process parameters with reasonable run-times.

There are two equivalent ways of presenting the technique for computating the tightness probability, mean and the second moment. One is based on conditional probability and conditional moments, while the other uses transformation of the integrals defining the tightness probability, mean and the second moment. We begin

with a presentation of the first approach.

The generalized canonical form in expression (4.15) can be reorganized by combining the non-Gaussian nonlinear term and the mean value a_0:

$$A = (a_0 + f_A(X_N)) + \sum_{i=1}^{n_{LG}} a_{LG,i} \cdot \Delta X_{LG,i} + a_{n+1} \cdot \Delta R_a \tag{4.22}$$

Then, for the fixed values of the non-Gaussian and nonlinear function parameters ΔX_N, A can be regarded a first-order canonical form, A_{Cond}, with only Gaussian and linear function parameters and its mean value is $a_0 + f_A(X_N)$. Now, consider two generalized canonical forms A and B represented in the form of Equation (4.22). When all ΔX_N are at fixed values, the conditional tightness probability $T_{A,cond}$, conditional mean $c_{0,cond}$ and conditional second moments $m_{2,cond}$ of $max(A, B)$ become functions of non-Gaussian and nonlinear function parameters ΔX_N:

$$\begin{aligned} T_{A,cond}(\Delta X_N) &= P(A > B|\Delta X_N) \\ c_{0,cond}(\Delta X_N) &= E[max(A, B)|\Delta X_N] \\ m_{2,cond}(\Delta X_N) &= E[(max(A, B))^2|\Delta X_N] \end{aligned} \tag{4.23}$$

Here, we assume that non-Gaussian and nonlinear function parameters ΔX_N are independent of all of the Gaussian and linear function parameters ΔX_{LG}. In fact, this is a rather valid assumption: correlated random variables tend to have similar distributions, and if a linear parameter is correlated with a nonlinear one, independence can be achieved by orthogonal transformation techniques, such as principal component analysis or independent component analysis. Therefore, the joint conditional probability density function of ΔX_{LG}, under the condition of frozen values of ΔX_N, is simply the joint probability density function of the ΔX_{LG}:

$$p(\Delta X_{LG}|\Delta X_N) = p(\Delta X_{LG}) \tag{4.24}$$

Thus, we can use analytical Clark's formulas in [22] for computing the conditional tightness probability, mean and second moments for the maximum of two generalized canonical forms, under the condition that the values of all non-Gaussian and nonlinear function parameters are frozen; however, a_0 and b_0 should be substituted by $a_0 + f_A(\Delta X_N)$ and $b_0 + f_B(\Delta X_N)$. Since this method uses only analytical formulas, the required values can be computed efficiently. The actual values of tightness probability, mean, and second moment of $max(A, B)$ can be computed by integrating the conditional tightness probability, mean and second moment over the space of non-Gaussian and nonlinear function parameters with their joint probability density function:

$$T_A = \int_{-\infty}^{\infty} T_{A,cond}(\Delta X_N) p(\Delta X_N) d\Delta X_N \tag{4.25}$$

$$E[max(A, B)] = \int_{-\infty}^{\infty} c_{0,cond}(\Delta X_N) p(\Delta X_N) d\Delta X_N \tag{4.26}$$

$$E[(max(A, B))^2] = \int_{-\infty}^{\infty} m_{2,cond}(\Delta X_N) p(\Delta X_N) d\Delta X_N \tag{4.27}$$

The integrations in Equations (4.25), (4.26) and (4.27) can be evaluated numerically. In the simplest case, it is performed by integrating numerically in m orthogonal discretized regions of non-Gaussian and nonlinear function parameters. Note that this discretized grid is created solely for the purpose of numerical integration, and it is unrelated to the grid in Chapter 2 that was used to model spatial correlations. Inside each integration grid, we compute the conditional tightness probability, conditional mean and conditional second moment by formulas (4.23). Then the integrals of Equation (4.25), (4.26) and (4.27) can be computed approximately as sums of corresponding values over all the discretization grids. For example, the numerical formula for tightness probability is as follows:

$$T_A = \sum_{k=1}^{m} T_{A,cond,k}(\Delta X_N) \cdot p_k(\Delta X_N) \cdot V_k \tag{4.28}$$

where $T_{A,Cond,k}(\Delta X_N)$ is the conditional tightness probability that $A > B$ under the condition that non-Gaussian and nonlinear function parameters have fixed values inside the k^{th} grid of integration; $p_k(\Delta X_N)$ is the value of the joint probability density function of the non-Gaussian and nonlinear function parameters in k^{th} grid; V_k is volume of the k^{th} grid. The computational complexity of numerical integration, performed by discretizing the integration region, is exponential with respect to the number of nonlinear and non-Gaussian parameters. Our experiments show that for reasonable accuracy it is enough to have as little as 5 to 7 discrete points for each variable. This approach is applicable for cases with up to 7 to 8 nonlinear and non-Gaussian variables. For higher dimensions the integrals can be computed by a Monte Carlo integration technique.

To better understand the technique for computing the required values of tightness probability, mean and standard deviations of $max(A, B)$, we now provide an alternative explanation for an equivalent derivation by a transformation of the integrals. Let us start with the evaluation of tightness probability in Equation (4.19). Given the condition that the ΔX_N variables are independent of the ΔX_{LG} variables, the joint probability density function of all sources of variations can be decomposed into:

$$p(\Delta X_N, \Delta X_{LG}, \Delta X_a, \Delta X_b) = p(\Delta X_N) \cdot p(\Delta X_{LG}, \Delta X_a, \Delta X_b) \tag{4.29}$$

Then the tightness probability T_A can be computed by:

$$T_A = \int_{A>B} p(\Delta X_N) \cdot p(\Delta X_{LG}, \Delta X_a, \Delta X_b) d\Delta X_{LG} d\Delta X_a d\Delta X_b d\Delta X_N \tag{4.30}$$

For fixed values of ΔX_N, the region $A > B$, where A and B are in generalized canonical forms, can be regarded as comparing two Gaussian random variables

$A_G(\Delta X_N)$ and $B_G(\Delta X_N)$, where

$$A_G = (a_0 + f_A(\Delta X_N)) + \sum_{i=1}^{n_{LG}} a_{LG,i}\Delta X_{LG,i} + a_{n+1}\Delta R_a \tag{4.31}$$

$$B_G = (b_0 + f_B(\Delta X_N)) + \sum_{i=1}^{n_{LG}} b_{LG,i}\Delta X_{LG,i} + b_{n+1}\Delta R_b \tag{4.32}$$

If we set

$$Q_0(\Delta X_N) = \int_{A_G(\Delta X_N) > B_G(\Delta X_N)} p(\Delta X_{LG}, \Delta X_a, \Delta X_b) d\Delta X_{LG} d\Delta X_a d\Delta X_b \tag{4.33}$$

then the tightness probability can be computed as:

$$\begin{aligned} T_A &= \int_{A>B} p(\Delta X_N) \cdot p(\Delta X_{LG}, \Delta X_a, \Delta X_b) d\Delta X_{LG} d\Delta X_a d\Delta X_b d\Delta X_N \\ &= \int_{-\infty}^{\infty} p(\Delta X_N) Q_0(\Delta X_N) d\Delta X_N \end{aligned} \tag{4.34}$$

Note that $Q_0(\Delta X_N)$ for fixed values of ΔX_N is in fact the tightness probability of $A_G(\Delta X_N)$ in $max(A_G(\Delta X_N), B_G(\Delta X_N))$, where $A_G(\Delta X_N)$ and $B_G(\Delta X_N)$ are both Gaussians for fixed ΔX_N. Since there is an analytical formula [22] for the tightness probability for Gaussian random variables, for fixed values of ΔX_N, $Q_0(\Delta X_N)$ can be computed efficiently. The tightness probability T_A in (4.34) can then be obtained by numerical integration over the space of non-Gaussian and/or nonlinear process parameters X_N.

Similarly, using the independence between ΔX_N and ΔX_{LG}, the mean and second moment of $max(A, B)$ can be computed as:

$$\begin{aligned} E[max(A,B)] &= \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} max(A,B) \cdot p(\Delta X_N) \cdot p(\Delta X_{LG}, \Delta X_a, \Delta X_b) d\Delta X_{LG} d\Delta X_a d\Delta X_b d\Delta X_N \\ &= \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} p(\Delta X_N) Q_1(\Delta X_N) d\Delta X_N \end{aligned} \tag{4.35}$$

$$\begin{aligned} E[max(A,B)^2] &= \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} max(A,B)^2 \cdot p(\Delta X_N) \cdot p(\Delta X_{LG}, \Delta X_a, \Delta X_b) d\Delta X_{LG} d\Delta X_a d\Delta X_b d\Delta X_N \\ &= \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} p(\Delta X_N) Q_1(\Delta X_N) d\Delta X_N \end{aligned} \tag{4.36}$$

where $Q_1(\Delta X_N)$ and $Q_2(\Delta X_N)$ are defined as:

$$Q_1(\Delta X_N) = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} max(A_G(\Delta X_N), B_G(\Delta X_N)) p(\Delta X_{LG}, \Delta X_a, \Delta X_b) d\Delta X_{LG} d\Delta X_a d\Delta X_b \quad (4.37)$$

$$Q_2(\Delta X_N) = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} (max(A_G(\Delta X_N), B_G(\Delta X_N)))^2 p(\Delta X_{LG}, \Delta X_a, \Delta X_b) d\Delta X_{LG} d\Delta X_a d\Delta X_b \quad (4.38)$$

For fixed values of ΔX_N, $Q_1(\Delta X_N)$ and $Q_2(\Delta X_N)$ are the mean and second moment, respectively, for the maximum of two Gaussian random variables and these can be found using analytical formulas. The mean and second moment of $max(A, B)$ can then be computed by numerical integration over the space of non-Gaussian and/or nonlinear process parameters X_N.

4.4 Implementation and Results

The proposed approach was implemented on top of EinsStat [80], an industrial statistical timing analysis tool. In the implementation, a process variation can have a non-Gaussian distribution and the delay dependence on a process parameter can be a nonlinear function. These are both specified by tables using an appropriately chosen discretization. The integrals for the mean, second moment and tightness probability are computed by numerical integration.

We first tested our implementation on computing $max(A, B)$ of two first-order canonical forms A and B with non-Gaussian parameters:

$$A = 10 + 0.5 \cdot \Delta X_1 + \Delta X_2 + 0.5 \cdot \Delta R_a \quad (4.39)$$

$$B = 10 + \Delta X_1 + 0.5 \cdot \Delta X_2 + 0.5 \cdot \Delta R_b \quad (4.40)$$

where ΔX_1 and ΔX_2 are random variables with lognormal probability distributions, ΔR_a and ΔR_b are Gaussian random variables for the locally independent randomness. Figure 4.3(a) shows the probability density function of $max(A, B)$ computed by the proposed technique, by the original parameterized SSTA technique for linear Gaussian process parameters (where non-Gaussian distributions are approximated with Gaussians having the same mean and standard deviation), and by Monte Carlo simulation. The PDF computed by the proposed technique matches the Monte Carlo results much closer than the PDF computed by the original technique. The proposed technique and Monte Carlo simulation both predict asymmetric PDFs with similar trends especially at the tails of PDFs. The PDF computed by the original technique has a symmetric shape and substantially underestimates the worst-case value.

Next, we tested our technique on $max(A, B)$ with nonlinear (cubic) functions of Gaussian parameters:

$$A = 10 + (\Delta X_1)^3/18 + (\Delta X_2)^3/9 + 0.5 \cdot \Delta R_a \quad (4.41)$$

$$B = 10 + (\Delta X_1)^3/9 + (\Delta X_2)^3/18 + 0.5 \cdot \Delta R_b \quad (4.42)$$

Figure 4.3(b) compares the PDFs computed by the original technique, by the proposed technique and by Monte Carlo simulation. The original technique uses linear approximation of nonlinear functions that passes through the same -3σ and +3σ points. The proposed technique predicts virtually the same result as Monte Carlo, while the original technique significantly over-estimates the standard deviation.

To choose the number of discretization points that provides a good tradeoff between accuracy and run-time, we ran tests on a small industrial design A (3,042 gates and 17,579 timing arcs). Table 4.1 shows the CPU-time of our technique for

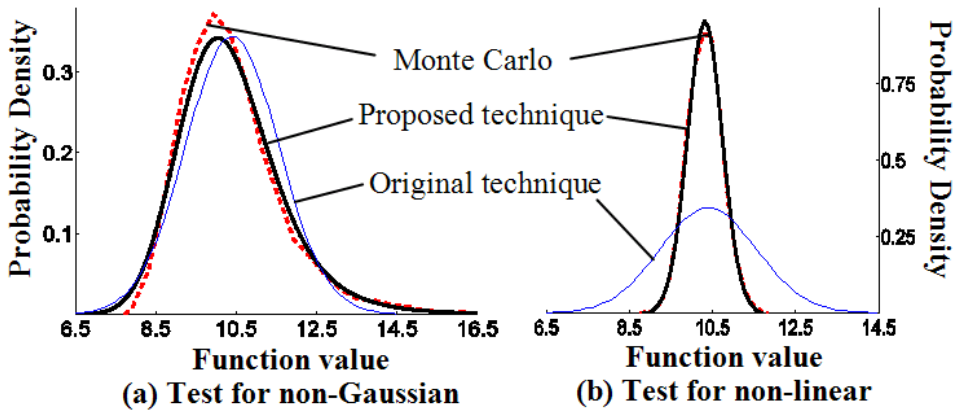


Figure 4.3: Comparison of PDFs for maximum of two generalized canonical forms A and B. (a) shows the results on a non-Gaussian distribution, where $A = 10 + 0.5 \cdot \Delta X_1 + \Delta X_2 + 0.5 \cdot \Delta R_a$ and $B = 10 + \Delta X_1 + 0.5 \cdot \Delta X_2 + 0.5 \cdot \Delta R_b$, where all variational sources ΔX_i are lognormal and ΔR_a is Gaussian. (b) shows results on a nonlinear delay function, where $A = 10 + (\Delta X_1)^3/18 + (\Delta X_2)^3/9 + 0.5 \cdot \Delta R_a$ and $B = 10 + (\Delta X_1)^3/9 + (\Delta X_2)^3/18 + 0.5 \cdot \Delta R_b$, and all variational sources ΔX_i and ΔR_a are Gaussian.

different numbers of non-Gaussian parameters, for 5 and 10 discretization points. The run time was measured on a single processor IBM Risc System 6000 model 43P-681. It is observed that processing three non-Gaussian parameters with 10 discretized points takes about 40 times longer than handling all three parameters as Gaussians, but for 5 discretization points, the run-time is only about 3 times longer. The PDF plots for design A are provided in Figure 4.4 for when 5, 10 and 20 discretized points are used. We observe that as the difference between PDF curves for 10 and 20 points is almost undistinguishable, the curve with 5 points also gives a result that is accurate enough. For nonlinear functions, we saw a similar dependence of run-time on the number of discretization points. Therefore, for our

other experiments, we have used only 5 discretized points.

Table 4.1: Comparison of the run-time as the number of non-Gaussian distributed sources, and the number discretization points, are varied.

Number of non-Gaussians		3	2	1	0
CPU-times (s)	10 points	69.17	7.53	2.14	1.38
	5 points	3.82	1.54	1.40	1.38

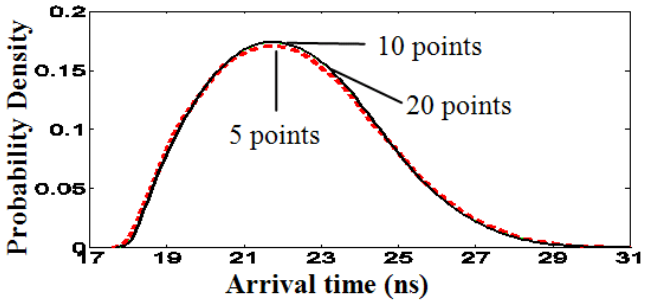


Figure 4.4: Comparison of accuracy versus run-time for Design A, when different numbers of discretized points (5, 10 and 20 points) are used in the computation.

We performed statistical timing analysis of the same design A with linear delay functions of three lognormally distributed global sources of variations and a Gaussian uncorrelated local variation. The average values of delay sensitivities to each global and local variation were set to 2% and 6% of the corresponding nominal delay values, respectively. Figure 4.5 shows the probability density functions of the latest arrival time computed by three different techniques. The proposed technique gives a close match to the Monte Carlo result. In contrast, the PDF computed by the original SSTA technique for linear, Gaussian case deviates substantially from the Monte Carlo result. The PDF computed by Monte Carlo simulation is not Gaussian, but closer to lognormal because all three global sources of variation have

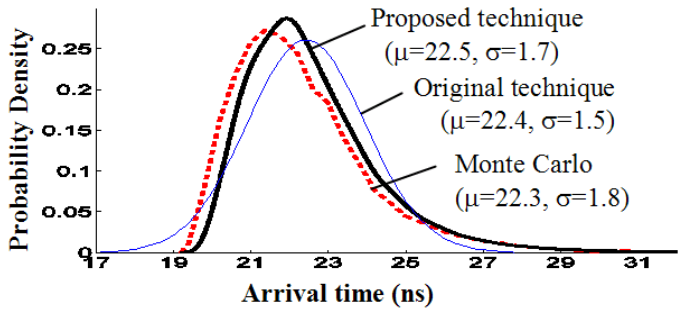


Figure 4.5: Comparison of PDFs of arrival time at a timing point for design A when different approaches are applied. All global sources of variations are lognormally distributed in the experiments. The proposed technique is shown by the bold solid curve, the original technique using Gaussian approximations by the thin solid curve, and the Monte Carlo results by the dotted bold curve.

lognormal distributions. Unlike the proposed method, the original SSTA technique for the linear, Gaussian case approximates all delays with a Gaussian distribution, and therefore, it is hard for it to estimate the PDF well. The Monte Carlo predicts the 0.1% and 99.9% confidence points of path delays as 19.4 ns and 32.0 ns, respectively. The proposed algorithm estimates similar values of 19.6 ns and 31.5 ns, respectively, while the original technique computes these values as 17.8 ns and 27.0 ns, respectively.

In the second set of experiments, the three global sources of variation had Gaussian distributions but the delays of circuit gates and wires were cubic functions of these variations. The values of delay sensitivities to each global source of variation and uncorrelated local variation were set to 2% and 6% of the corresponding nominal delay values, respectively. Figure 4.6 shows PDFs and CDFs of the circuit delay computed by three different techniques. The proposed technique computes the same mean value as Monte Carlo, while the original technique overes-

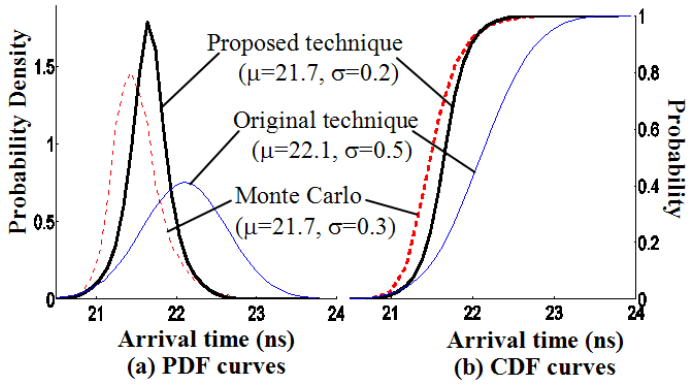


Figure 4.6: Comparison of PDFs of arrival time at a timing point for design A when different approaches are applied. The delay functions at all circuit nodes are nonlinear (cubic) function of the variational sources in the experiments. The proposed technique is shown by the bold solid curve, the original technique using Gaussian approximations by the thin solid curve, and the Monte Carlo results by the dotted bold curve.

timates it. The original technique computes the 99.9% confidence point as 22.7 ns, as against 22.9 ns from Monte Carlo, while the original technique over-estimates it as 23.7 ns. Thus, we can conclude that when parameter variations have non-Gaussian distributions, or gate and wire delay depends on parameters nonlinearly, the proposed technique is essential to correctly predict circuit delay distribution and manufacturing yield.

Table 4.2 shows the run time of statistical timing analysis for five industrial designs when different numbers of non-Gaussian parameters are used in the analysis. In the set of tests, there are three global variational process parameters. In the case when the number of non-Gaussians is zero, the three global sources are set as Gaussian random variables, and in general, when the number of non-Gaussians is set to k ($0 \leq k \leq 3$), the remaining $3-k$ sources remain Gaussians. We see that, as

Table 4.2: Comparison of run-time versus the numbers of non-Gaussian process parameters for various sizes of industrial designs.

Ckt Name	Number of Gates	Timing Arcs	Number of Non-Gaussians			
			3	2	1	0
A	3,042	17,579	3.8 s	1.5 s	1.4 s	1.4 s
B	11,937	57,151	12.3 s	5.53 s	4.3 s	3.07 s
C	53,317	392,097	79.1 s	35.8 s	27.3 s	18.7 s
D	70,216	363,537	93.3 s	41.3 s	30.5 s	19.7 s
E	1,085,034	5,799,545	2,083.1 s	982.0 s	788.5 s	703.6 s

the number of non-Gaussian parameters increases to 3, the run-time is only about 3 to 5 times longer compared to the case without any non-Gaussian parameters. The size of the designs for tests varies from 3,042 up to 1,085,034 gates. For the largest design E, the run-time is only about 35 minutes. In contrast, for the smallest design A, the run-time of Monte Carlo simulation is about 5 hours. However, due to the large size of designs, Monte Carlo simulations cannot be completed in a realistic amount of time, and thus the run-times are not provided in the table. Statistical timing analysis with nonlinear parameters has approximately the same run time.

4.5 Conclusion

In this chapter, we have presented a novel and efficient technique for handling arbitrary non-Gaussian and nonlinear function parameters in parameterized block-based SSTA. Our approach is based on an extension of the first-order canonical form for representing delay and arrival time variations. Therefore this technique is fully compatible with the parameterized SSTA approach for Gaussian and linear function parameters presented in Chapter 3, and preserves its computational efficiency

in processing such types of process parameter variations. The experimental results showed that the probability distributions of circuit delays computed by the new technique are closer to the results of Monte Carlo simulations than the original parameterized SSTA which approximates non-Gaussian distributions with Gaussians and nonlinear functions with linear functions, especially at the 99.9% confidence level. It should be also noted that in many cases non-Gaussian distributions of parameter variations can be approximated with Gaussians with reasonable accuracy, and only significantly asymmetric distributions requires handling as non-Gaussians. This conclusion is very important in practice because it justifies approximating most parameter distributions by Gaussians.

The limitation of the algorithm is that its run-time is exponential to the number of non-Gaussian and/or nonlinear function parameters. To further improve the efficiency, it is possible to develop techniques that can compute the *max* function analytically. In practice, as the number of non-Gaussian and/or nonlinear function parameters is not large, the algorithm is very efficient and provides a general framework for SSTA handling non-Gaussian parameters and nonlinear functions of delays. The method can be used to validate the approximation of process parameters as Gaussians and usage of linear delay functions, and then selectively apply crucial process parameters as non-Gaussian distributed or with nonlinear functions. The method is also important for sign-off timing analysis.

Chapter 5

Prediction of Leakage Power Under Uncertainties

In this chapter, we present a method to analyze the leakage current of a circuit under process variations, considering inter-die and intra-die variations as well as the effect of the spatial correlations of intra-die variations. A lognormal distribution is used to approximate the leakage current of each gate and the total chip leakage is achieved by summing up the lognormals. In this work, both subthreshold leakage and gate tunneling leakage are considered. The proposed method is shown to be effective in predicting the CDF/PDF of the total chip leakage.

5.1 Introduction

Leakage power is increasing drastically with technology scaling, and has already become a substantial contributor to the total chip power dissipation. According to International Technology Roadmap for Semiconductors (ITRS) [7], leakage power is

expected to increase to 50% of the total chip power and to dominate the switching power of a circuit over the next few generations. Consequently, it is important to accurately estimate leakage currents so that they can be accounted for during design, and so that it is possible to effectively optimize the total power consumption of a chip.

The major components of leakage in current CMOS technologies are due to subthreshold leakage and gate tunneling leakage. For a gate oxide thickness, T_{ox}, of over 20Å, the gate tunneling leakage current, I_{gate}, is typically very small [42], while the subthreshold leakage, I_{sub}, dominates other types of leakage in circuit. For this reason, there have been extensive studies on subthreshold leakage over the last ten years [38, 69]. However, the gate tunneling leakage is exponentially dependent on gate oxide thickness, e.g., a reduction in T_{ox} of 2Å will result in an order of magnitude increase in I_{gate}. Therefore, with the continuous scaling of gate oxide thickness, I_{gate} is no longer negligible and is likely to dominate other leakage mechanisms in future generations, at least until new high-K dielectrics are introduced. At this time, it is unclear when these will be introduced, and gate leakage is already seen to be very significant in 90nm, 65nm and 45nm technologies [7], so that its analysis is of profound importance.

In the literature, several research works on the analysis and minimization of total circuit leakage including the effect of I_{gate} have been conducted [42]. The analysis of total leakage power of circuit is complicated by the state dependency of subthreshold and gate tunneling leakage, and the interactions between these two leakage mechanisms.

An added complication, which has been less widely studied, arises due to the increasing importance of process variations in cutting-edge technologies. As a result of this, the values of all process parameters can no longer be considered to be con-

stants, but must be modeled as random variables that are described by probability density functions. These variations translate into uncertainties in circuit performance metrics. Specifically, total circuit leakage also becomes a random variable that depends on the variations of fundamental process parameters that it is most sensitive to parameters such as the transistor effective gate length and the gate oxide thickness.

Under inter-die variations, if the leakage of all gates or devices are sensitive to the process parameters in similar ways, the circuit performance can be analyzed at multiple process corners using deterministic analysis methods. Otherwise, or with intra-die variations, statistical methods must be used to correctly predict the leakage. Specifically, the gate leakage can vary exponentially with these parameters, the simple use of worst-case values for all parameters can result in exponentially larger leakage estimates than are actually obtained. While these will certainly be pessimistic, the inaccuracy in these values makes them practically useless.

Most of the previous works on statistical performance analysis has focused on statistical timing analysis, and only a few works have investigated the variation of leakage power under the effect of process variations [52, 55, 64, 65, 70]. In [55, 70], analytical methods were proposed to estimate the mean and standard deviation of the total chip subthreshold leakage power under intra-die parameter variations. In [52], gate tunneling and the reverse biased source/drain junction band-to-band tunneling (BTBT) leakage, and the correlations among these components were included, in addition to subthreshold leakage, in the analysis of total leakage. In [65], the probability density function of the total chip subthreshold leakage was derived. The authors of [64] presented an analytical framework that provides a closed form expression for the total chip leakage current as a function of process parameters that can be used to estimate yield under power and performance constraints. How-

ever, none of these have considered the effects of spatial correlations in intra-die process variations.

In this chapter, we propose a method for predicting the distribution of total circuit leakage power, including subthreshold and gate tunneling leakage and their interactions, under both inter-die and intra-die variations of parameters. The spatial correlations in intra-die variations and the correlation between these two leakage mechanisms are also considered.

The remainder of the chapter is organized as follows. Section 5.2 formulates the problem that we will solve here. A first method for estimating the distribution of full-chip leakage power is given in Section 5.3, and this is followed by an improved approach, presented in Section 5.4. Finally, a list of experimental results are shown in Section 5.5.

5.2 Problem Description

The total leakage power consumption of a circuit is input-pattern-dependent, i.e., the value differs as the input signal to the circuit changes, because the leakage power consumption, due to subthreshold and gate tunneling leakage, of a gate depends on the input vector state at the gate. As illustrated in [3], the dependency of leakage on process variations is more significant than on input vector states. Therefore, it is sufficient to predict the effects of process variations on total circuit leakage by studying the variation of average leakage current for all possible input patterns to the circuit. However, it is impractical to estimate the average leakage by simulating the circuit at all input patterns, and thus an input pattern-independent approach is more desirable.

In switching power estimation, probabilistic approaches [54] have been used for this purpose. The work of [3] proposed a similar approach that computes the average leakage current of each gate and estimates the total average circuit leakage as a sum of the average leakage currents of all gates:

$$I_{tot}^{avg} = \sum_{k=1}^{N_g} I_{leak,k}^{avg} = \sum_{k=1}^{N_g} \sum_{\forall vec_{i,k}} Prob(vec_{i,k}) \cdot I_{leak,k}(vec_{i,k}) \tag{5.1}$$

where N_g is the total number of gates in the circuit, $I_{leak,k}^{avg}$ is the average leakage current of the k^{th} gate, $vec_{i,k}$ is the i^{th} input vector at the k^{th} gate, $Prob(vec_{i,k})$ is the probability of occurrence of $vec_{i,k}$, and $I_{leak,k}(vec_{i,k})$ is the leakage current of the k^{th} gate when the gate input vector is $vec_{i,k}$.

In this work, we will solve the problem of computing the probability distribution of the average circuit leakage current I_{tot}^{avg}, formulated in Equation (5.1), under process variations. Both subthreshold and gate tunneling leakage currents are taken into account in the computation. We consider process variations of transistor gate length L_{eff} and gate oxide thickness T_{ox}, since the subthreshold and gate tunneling leakage currents are most sensitive to these parameters [52, 75]. We also assume that L_{eff} and T_{ox} are normally distributed, with values of L_{eff} spatially correlated and T_{ox} uncorrelated in different gates inside the chip as in Chapter 3. However, the procedure used herein is not restricted to these assumptions, and can be generalized to other spatially correlated or uncorrelated parameter variations.

5.3 Computing the Distribution of Full-chip Leakage Current

We will now present the methodology used to estimate the distribution of average full-chip leakage current, I_{tot}^{avg}, under process variations. As implied by Equation (5.1), the distribution of I_{tot}^{avg} can be calculated in two steps: first, computing the distribution of each $I_{leak,k}(vec_{i,k})$, the leakage current of the k^{th} gate when the gate input vector is $vec_{i,k}$; and second, finding the distribution of the weighted sum of all $I_{leak,k}(vec_{i,k})$ terms. Since each $I_{leak,k}(vec_{i,k})$ can further be decomposed into $I_{sub,k}(vec_{i,k})+I_{gate,k}(vec_{i,k})$, where $I_{sub,k}(vec_{i,k})$ and $I_{gate,k}(vec_{i,k})$ are the subthreshold and gate tunneling leakage currents, respectively, for the k^{th} gate with input state $vec_{i,k}$, I_{tot}^{avg} can be computed as:

$$I_{tot}^{avg} = \sum_{k=1}^{N_g} \sum_{\forall vec_{i,k}} Prob(vec_{i,k}) \cdot (I_{sub,k}(vec_{i,k}) + I_{gate,k}(vec_{i,k})) \tag{5.2}$$

In the discussion that follows, we will first present how the distributions of subthreshold leakage current, $I_{sub,k}(vec_{i,k})$, and gate tunneling leakage current, $I_{gate,k}(vec_{i,k})$, are estimated in Section 5.3.1 and 5.3.2, respectively. The analytical approach to obtain the probability density function for the total weighted sums of all $I_{sub,k}(vec_{i,k})$ and $I_{gate,k}(vec_{i,k})$ terms will then be presented in Section 5.3.3. As the same framework can be applied for computing the distribution of each $I_{sub,k}(vec_{i,k})$, for conciseness, we will use I_{sub} for $I_{sub,k}(vec_{i,k})$, and similarly, I_{gate} for $I_{gate,k}(vec_{i,k})$, in later sections.

5.3.1 Distribution of Subthreshold Leakage Current

The commonly used model for subthreshold leakage current through a transistor expresses this current as [75]:

$$I_{sub} = I_0 e^{(V_{gs}-V_{th})/n_s V_T}(1 - e^{-V_{ds}/V_T}) \tag{5.3}$$

Here, $I_0 = \mu_0 C_{ox}(W_{eff}/L_{eff})V_T^2 e^{1.8}$, where μ_0 is zero bias electron mobility, C_{ox} is the gate oxide capacitance, W_{eff} and L_{eff} are the effective transistor width and length, respectively, V_{gs} and V_{ds} are gate-to-source voltage and drain-to-source voltage, respectively, n_s is the subthreshold slope coefficient, $V_T = kT/q$ is the thermal voltage, where k is Boltzman constant, T is the operating temperature in Kelvin (K), q is charge on an electron, and V_{th} is the subthreshold voltage.

It is observed that V_{th} is most sensitive to gate oxide thickness T_{ox} and effective transistor gate length L_{eff} due to short-channel effects [75]. Due to the exponential dependency of I_{sub} on V_{th}, a small change on L_{eff} or T_{ox} will have a substantial effect on I_{sub}. From this intuition, we estimate the subthreshold leakage current per transistor width by developing an empirical model through curve-fitting, similarly to [52, 65]:

$$I_{sub} = c \times e^{a_1 + a_2 L_{eff} + a_3 L_{eff}^2 + a_4 T_{ox}^{-1} + a_5 T_{ox}} \tag{5.4}$$

where c and the a_i terms are the fitting coefficients.

In this way, I_{sub} is modeled as an exponential function in the form of $c \times e^U$, where U is an explicit function of L_{eff} and T_{ox}. When L_{eff} and T_{ox} show process variations, the exponent U, and thus I_{sub}, become random variables. Since the magnitude of process variations is observed to be around $10 - 20\%$ in practice, I_{sub} can be well approximated by expanding its exponent U using a first-order Taylor

expansion at the nominal values of the process parameters:

$$I_{sub} = c \times e^{U_0 + \beta_1 \cdot \Delta L_{eff} + \beta_2 \cdot \Delta T_{ox}} \tag{5.5}$$

where U^0 is the nominal value of the exponent U, β_0 and β_1 are the derivatives of U to L_{eff} and T_{ox} evaluated at their nominal values, respectively, and ΔL_{eff} and ΔT_{ox} are random variables representing for the variations in the process parameters L_{eff} and T_{ox}, respectively.

Expression (5.5) for I_{sub} can also be written as $e^{ln(c)+U_0+\beta_1 \cdot \Delta L_{eff}+\beta_2 \cdot \Delta T_{ox}}$. Since ΔL_{eff} and ΔT_{ox} are assumed to be Gaussian-distributed, I_{sub} is seen as an exponential function of a Gaussian random variable, with mean $ln(c) + U_0$ and standard deviation $\sqrt{\beta_1^2 \sigma_{L_{eff}}^2 + \beta_2^2 \sigma_{T_{ox}}^2}$, where $\sigma_{L_{eff}}$ and $\sigma_{T_{ox}}$ are standard deviations of ΔL_{eff} and ΔT_{ox}, respectively.

In general, if x is a Gaussian random variable, then $z = e^x$ is a lognormal distributed random variable and the probability density function of z is given by [61]:

$$f(z) = \frac{1}{z\sqrt{2\pi}\sigma} e^{-(ln(z)-\mu)^2/(2\sigma^2)} \tag{5.6}$$

where μ and σ are the mean and standard deviation of the Gaussian random variable x, respectively. Therefore, it is obvious that I_{sub} can be approximated as a lognormally distributed random variable whose probability density function can be characterized using the values of c, U_0 and β_i's.

Since subthreshold leakage current has a well-known input state dependency due to the stack effect [69], the PDFs of subthreshold leakage currents must be characterized for all possible input states for each type of gate in the library, for which the same approach described in this section can be applied. Once the library is characterized, a simple look-up table (LUT) can then be used to retrieve the

corresponding model characterized given the gate type and input vector state at a gate.

5.3.2 Distribution of Gate Tunneling Leakage Current

In [15], an analytical model was proposed for the gate oxide tunneling current density J_{tunnel}.

$$J_{tunnel} = \frac{4\pi m^* q}{h^3}(kT)^2(1 + \frac{\gamma kT}{2\sqrt{E_B}})e^{\frac{E_{F0,Si/SiO_2}}{kT}}e^{-\gamma\sqrt{E_B}} \tag{5.7}$$

Here m^* is the transverse mass that equals $0.19m_0$ for electron tunneling and $0.55m_0$ for hole tunneling, where m_0 is the free electron rest mass, h is Planck's constant, γ is defined as $4\pi T_{ox}\sqrt{2m_{ox}}/h$, where m_{ox} is the effective electron [hole] mass in the oxide, E_B is the barrier height, $E_{F0,Si/SiO_2} = q\phi_S - q\phi_F - E_G/2$ is the Fermi level at the Si/SiO_2 interface, where ϕ_S is surface potential, ϕ_F is the Fermi energy level potential, either in the Si substrate for the gate tunneling current through the channel, or in the source/drain region for the gate tunneling current through the source/drain overlap, and E_G is the Si band gap energy.

In [15], the gate-tunneling current of PMOS devices is neglected due to the larger effective mass and barrier height for holes compared to electrons at the SiO_2/Si interface. Moreover, only tunneling current in the gate-to-channel region is considered, and edge direct tunneling (EDT) in the gate-to-drain and gate-to-source overlap regions is ignored. This is because these overlap regions are significantly smaller than the gate-to-channel region; moreover, EDT can be further reduced using process technologies [74]. Therefore, in this work, the gate tunneling leakage current is taken into account only for NMOS transistors at logic "1".

Although the formulation (5.7) possesses a high accuracy, it does not lend itself easily to the analysis of the effects of parameter variations. Therefore, we again use

an empirically characterized model to estimate I_{gate} per transistor width through curve-fitting:

$$I_{gate} = c' \times e^{b_1 + b_2 L_{eff} + b_3 L_{eff}^2 + b_4 T_{ox} + b_5 T_{ox}^2} \tag{5.8}$$

where c' and the b_i terms are the fitting coefficients.

Similar to the method for estimating the distribution of I_{sub}, under the variations of L_{eff} and T_{ox}, I_{gate} can be approximated by applying first-order Taylor expansion to the exponent U' of Equation (5.8):

$$I_{gate} = c' \times e^{U'_0 + \lambda_1 \cdot \Delta L_{eff} + \lambda_2 \cdot \Delta T_{ox}} \tag{5.9}$$

where U'_0 is the nominal value of the exponent U', and λ_0 and λ_1 are the derivatives of U' to L_{eff} and T_{ox} evaluated at their nominal values, respectively.

Under this approximation, I_{gate} becomes a lognormal distributed random variable, and its PDF can be characterized through the values of c', U'_0 and λ'_i. Since the gate tunneling leakage current is input state dependent, the PDFs of the I_{gate} variables are characterized for all possible input states for each type of gate in the library, and a simple look-up table (LUT) is used for model retrieval while evaluating a specific circuit.

5.3.3 Distribution of Full-Chip Leakage Current

Sections 5.3.1 and 5.3.2 show that each of $I_{sub,k}(vec_{i,k})$ or $I_{gate,k}(vec_{i,k})$, i.e., the subthreshold or gate tunneling leakage current, respectively, of the k^{th} gate when its input vector is $(vec_{i,k})$, can be modeled as a lognormal random variable under process variations. In this section, we will present the approach to find the distribution of I_{tot}^{avg} as formulated in Equation (5.2), which is a weighted sum of all

$I_{sub,k}(vec_{i,k})$ and $I_{gate,k}(vec_{i,k})$ variables, weighted by $Prob(vec_{i,k})$ terms, the probabilities of input vector $vec_{i,k}$ at the gate. Since the probability of each $vec_{i,k}$ can be computed by specifying signal probabilities at the circuit primary inputs and propagating the probabilities into all gates pins in the circuit, as in [3], in this section, we focus on the computation of the PDF of the weighted sum.

As each of $I_{sub,k}(vec_{i,k})$ or $I_{gate,k}(vec_{i,k})$ has a lognormal distribution, it can easily be seen that any multiplication by a constant maintains this property; specifically, $Prob(vec_{i,k}) \cdot I_{sub,k}(vec_{i,k})$ and $Prob(vec_{i,k}) \cdot I_{gate,k}(vec_{i,k})$ are both lognormally distributed. Therefore, the problem of calculating the distribution of I_{tot}^{avg} becomes that of computing the PDF of the sum of a set of lognormal random variables. Furthermore, the set of lognormal random variables in the summation could be correlated since:

- the leakage current random variables for any two gates may be correlated due to spatial correlations of intra-die variations of process parameters
- within the same gate, the subthreshold and gate tunneling leakage currents are correlated, and the leakage currents under different input vectors are correlated, because they are sensitive to the same process parameters of the same gate, regardless of whether these are spatially correlated or not.

In this section, we will present an efficient approach to predict the probability density function of the full-chip leakage current, by computing the PDF of the sum of correlated lognormal random variables, so that the spatial correlations of process parameters, and correlations between different leakage components can be correctly taken into account. This section is organized as follows. We first describe Wilkinson's method [2] for approximating a sum of correlated lognormal random

variables. Next, a more efficient approach is then proposed to reduce the computational complexity of this calculation. For clarity, the approach described first considers only intra-die variations of process parameters. The extension to handling inter-die variations is trivial, and will be shown briefly in the end of this section.

Finding the Sum of Correlated Lognormals by Wilkinson's Method

Theoretically, the sum of several lognormal distributed random variables is not known to have a closed form. However, it may be well approximated as a lognormal, as is done in Wilkinson's method [2]. That is, the sum of m lognormals, $S = \sum_{i=1}^{m} e^{Y_i}$, where each Y_i is a normal random variable with mean m_{y_i} and standard deviation σ_{y_i}, and the Y_i variables can be correlated or uncorrelated, can be approximated as a lognormal e^Z, where Z is normally distributed, with mean m_z and standard deviation σ_z. In Wilkinson's approach, the values of m_z and σ_z are obtained by matching the first two moments, u_1 and u_2, of e^Z and S as follows:

$$u_1 = E(e^Z) = E(S) = \sum_{i=1}^{m} E(e^{Y_i}) \tag{5.10}$$

$$\begin{aligned} u_2 &= E(e^{2Z}) = E(S^2) = Var(S) + E^2(S) \\ &= \sum_{i=1}^{m} Var(e^{Y_i}) + 2\sum_{i=1}^{m-1}\sum_{j=i+1}^{m} cov(e^{Y_i}, e^{Y_j}) + E^2(S) \\ &= \sum_{i=1}^{m} Var(e^{Y_i}) + 2\sum_{i=1}^{m-1}\sum_{j=i+1}^{m} \left(E(e^{Y_i}e^{Y_j}) - E(e^{Y_i})E(e^{Y_j})\right) + E^2(S) \end{aligned} \tag{5.11}$$

where $E(.)$ and $Var(.)$ are the symbols for the mean and variance values of a random variable, and $cov(.,.)$ represents the covariance between two random variables.

In general, the mean and variance of a lognormal random variable e^{X_i}, where X_i is normal distributed with mean m_{x_i} and standard deviation σ_{x_i}, is computed

by:

$$E(e^{X_i}) = e^{m_{x_i}+\sigma_{x_i}^2/2} \tag{5.12}$$

$$Var(e^{X_i}) = e^{2m_{x_i}+2\sigma_{x_i}^2} - e^{2m_{x_i}+\sigma_{x_i}^2} \tag{5.13}$$

The covariance between two lognormal random variables e^{X_i} and e^{X_j} can be computed by:

$$cov(e^{X_i}, e^{X_j}) = E(e^{X_i} \cdot e^{X_j}) - E(e^{X_i})E(e^{X_j}) \tag{5.14}$$

Superposing Equations (5.12), (5.13) and (5.14) into Equations (5.10) and (5.11) results in:

$$u_1 = E(e^Z) = e^{m_z+\sigma_z^2/2} = E(S) = \sum_{i=1}^{m} (e^{m_{y_i}+\sigma_{y_i}^2/2}) \tag{5.15}$$

$$\begin{aligned} u_2 &= E(e^{2Z}) = e^{2m_z+2\sigma_z^2} = E(S^2) \\ &= \sum_{i=1}^{m} (e^{2m_{y_i}+2\sigma_{y_i}^2} - e^{2m_{y_i}+\sigma_{y_i}^2}) + 2\sum_{i=1}^{m-1}\sum_{j=i+1}^{m} (e^{m_{y_i}+m_{y_j}+(\sigma_{y_i}^2+\sigma_{y_j}^2+2r_{ij}\sigma_{y_i}\sigma_{y_j})/2} \\ &\quad -e^{m_{y_i}+\sigma_{y_i}^2/2}e^{m_{y_j}+\sigma_{y_j}^2/2}) + u_1^2 \end{aligned} \tag{5.16}$$

where r_{ij} is the correlation coefficient between Y_i and Y_j.

Solving (5.15) and (5.16) for m_z and σ_z yields:

$$m_z = 2\ln u_1 - \frac{1}{2}\ln u_2 \tag{5.17}$$

$$\sigma_z^2 = \ln u_2 - 2\ln u_1 \tag{5.18}$$

The computational complexity of Wilkinson's approximation can be analyzed through the cost of computing m_z and σ_z. The computational complexities of m_z and σ_z are determined by those of u_1 and u_2, whose values can be obtained using the formulas in (5.15) and (5.16). It is clear that the computational complexity of u_1 is dominated by that of u_2, since the former involves only one-looped sum, while the

latter also contains a double-looped one: the complexity of calculating u_1 is $O(m)$, while that of u_2 is $O(m \cdot N_{corr})$, where N_{corr} is the number of correlated pairs among all pairs of Y_i variables. The cost of computing u_2 can also be verified by examining the earlier expression of u_2 in (5.11), in which the double-looped sum, in fact, corresponds to the covariance of Y_i and Y_j, which becomes zero when Y_i and Y_j are uncorrelated. Therefore, if $r_{ij} \neq 0$ for all pairs of Y_i and Y_j, the complexity of calculating u_2 is $O(m^2)$; if $r_{ij} = 0$ for all pairs of i and j, the complexity is $O(m)$.

As explained earlier, for full-chip leakage analysis, the number of correlated lognormal distributed leakage components in the summation could be extremely large, which could lead to a prohibitive amount of computation. If Wilkinson's method is applied directly, when the total number of gates in the circuit is N_g, the complexity for computing the sum will be $O(N_g^2)$, which is impractical for large circuits. In the remainder of this section, we will propose to compute the summation in a more efficient way.

Reducing the Number of Correlated Lognormals to be Summed

Since Wilkinson's method has a quadratic complexity with respect to the number of correlated lognormals to be summed, we now introduce mechanisms to reduce the number of correlated lognormals in the summation, to improve the computational efficiency.

First, the number can be reduced by identifying dominant states for subthreshold and gate tunneling leakage currents for each type of gate in the circuit.

Due to state dependencies of subthreshold and gate tunneling leakage currents, the computation of full-chip leakage current must take into account all possible input patterns at all gates in the circuit. In general, for a gate with N_{inpin} input pins,

the number of input states to be considered can be $2^{N_{inpin}}$. However, the leakage currents at some input states may not be as important as at others. It is sufficient to identify the important ones, corresponding to dominant states, and consider the leakage currents only at dominant states without losing much of accuracy.

When only subthreshold leakage current is considered, the dominant states for subthreshold leakage current in a transistor stack correspond to those with only one "off" transistor in the pull-up or pull-down chain [38, 69]. In this way, for a transistor stack of length q, the number of input states to consider is reduced to a much smaller size, q instead of 2^q. However, when gate tunneling leakage current is also considered, the dominant states must be characterized based on both leakage mechanisms and their interactions.

The interaction effects between the two mechanisms are analyzed in [42] by studying three scenarios for the middle transistor t_n in a NMOS transistor stack of length 3, as shown in Figure 5.1: in scenario (a) where t_n has a conducting path to supply and nonconducting path to gate output, I_{gate} does not interact with I_{sub} in the stack and the total leakage in stack is the sum of the two; in scenario (b) where t_n has a nonconducting path to supply and conducting path to gate output, I_{gate} is one order of magnitude smaller than that of case (a) and can be ignored safely; in scenario (c) where t_n has a nonconducting path to supply and gate output, due to the interaction between I_{sub} and I_{gate}, I_{sub} can be ignored safely. For details, the reader is referred to [42].

The analysis shows that a dominant state for subthreshold leakage current may not be one for gate tunneling leakage current, e.g., scenario (b) is a dominant state for I_{sub}, but not I_{gate}, and scenario (c) is a dominant state for I_{gate}, but not I_{sub}. Therefore, one way of identifying the dominant states for leakage current for a gate is to separately determine the set of dominant states for the subthreshold and

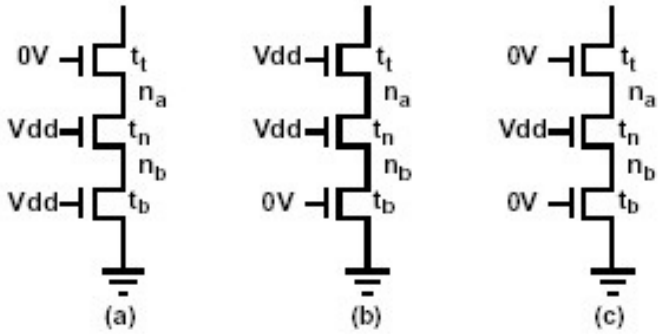
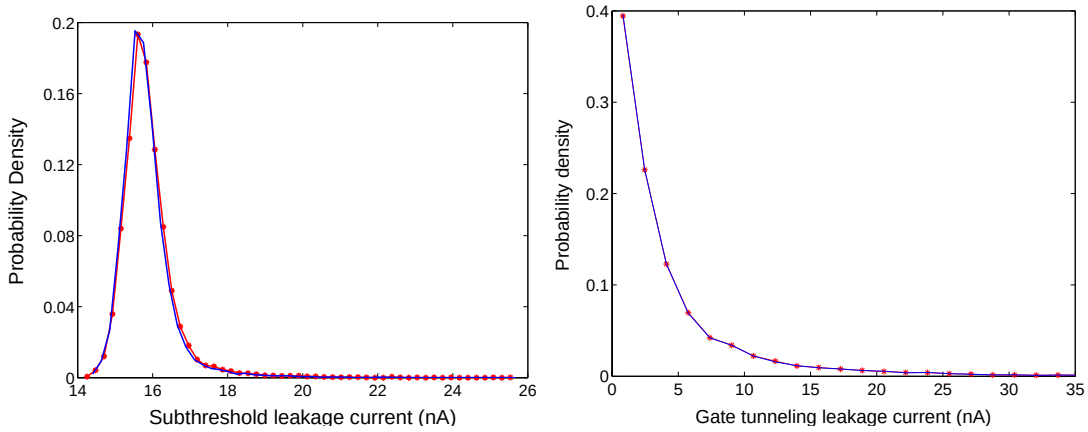


Figure 5.1: Three scenarios of combined I_{sub} and I_{gate} for a three-input NMOS transistor stack [42].

gate tunneling leakage currents. From the analysis above, the dominant states for subthreshold and gate tunneling leakage currents can be identified by the following rules. For a transistor stack, the set of dominant states for subthreshold leakage current remains being those with only one "off" transistor in the pull-up or pull-down chain, since the value of I_{sub} is strongly reduced only when there is more than one "off" transistor in the pull-up or pull-down chain. The determination of dominant states for gate tunneling leakage current is based on the following rule: in a transistor stack, the gate tunneling leakage current of a transistor is negligible if there is a conducting path to the gate output from this transistor.

To show the accuracy of leakage current estimation considering only dominant states under process variations, we compare, by Monte Carlo simulation, the distribution of the average subthreshold leakage current, $\sum_{\forall vec_{i,k}} Prob(vec_{i,k}) \cdot I_{sub,k}(vec_{i,k})$, and the average gate tunneling leakage current, $\sum_{\forall vec_{i,k}} Prob(vec_{i,k}) \cdot I_{gate,k}(vec_{i,k})$, for each type of gate in library using only dominant states with that using a full set of input vectors, assuming all input vectors having equal probabilities of occurrence. Figure 5.2(a) shows as an example the PDF curves of simulations with

dominant states and full set of states for average subthreshold leakage current for a 3-input NAND gate when the 3σ values of L_{eff} and T_{ox} are 20%. A close match is observed between these two PDF curves, and the same observation can be made when we compare the PDF curves of gate leakage for a 3-input NAND gate, using full-simulation and dominant states, as shown in Figure 5.2(b). For all types of gates in our library, the error can be controlled within 2%.



(a) Average subthreshold leakage current (b) Average gate-tunneling leakage current

Figure 5.2: Comparison of PDFs of average leakage currents using dominant states with that of full input vector states for a 3-input NAND gate, by Monte Carlo simulation with 3σ variations of L_{eff} and T_{ox} 20%. The solid curve shows the result when only dominant states are used, and the starred curve corresponds to simulation with all input vector states.

Secondly, instead of directly computing the sum of random variables of all leakage current terms, by grouping leakage current terms by model and grid, and calculating the sum in each group separately first, the computational complexity in the computation of full-chip leakage reduces to quadratic in the number of groups.

This is because, as will be explained in this section, the summation in each

group can be computed in linear time with respect to the number of leakage terms in each group. The results of the sums in all groups are then approximated as correlated lognormal random variables that can be then computed directly using Wilkinson's method. Since the number of groups is relatively small, a calculation that is quadratic in the number of groups is practically very economical.

Consider any dominant state for subthreshold leakage current that has only one "off" transistor in the transistor stack. It is observed that the values of subthreshold leakage currents *per unit width*, and thus their probabilistic distributions under process variations, are almost the same for any two transistor stacks that have the same number of "on" transistors between the drain of the only "off" transistor and the output of the gate. For example, it is observed that the subthreshold leakage current per unit transistor width is the same for the pull-down of a NAND4 in state 0111, a NAND3 in state 011, a NAND2 in state 01, and an INV in state 0. Therefore, this equivalence can be used to compactly store the PDF of the subthreshold leakage current per unit width in an LUT, and different types of gates, with different stack lengths, can be characterized by the same LUT entry. If q is the length of the longest stack in the library, the number of different models is $2q$ in the LUT of I_{sub} (q each for I_{sub} for the PMOS and the NMOS).

For a dominant state of the gate tunneling leakage current, it is observed that if a transistor shows gate tunneling leakage, the value and probability distribution of I_{gate} can be determined by the number of "off" transistors between the leaking transistor and its supply in the transistor stack. In this way, the number of distinct models that store the gate tunneling leakage current per unit width is limited. Specifically, the total number of different models used in the LUT is only $q - 2$, if the length of the longest stack in the library has length q.

Therefore, the total number of distinct models used in the LUT for the PDFs

of the subthreshold and gate tunneling leakage currents is reduced to $2q + q - 2$, where q is the length of the longest stack in the library. Next, we will show that if the leakage current terms to be summed in Equation (5.2) are grouped by the LUT model that they correspond to and their grid location, then the sum in each group can be computed in linear time with respect to the number of leakage terms in the group. For illustration purposes, we only describe the computation of grouped sum for subthreshold leakage current; the computation of gate leakage current proceeds along similar lines.

The subthreshold leakage current term here refers to the term $Prob(vec_{i,k}) \cdot I_{sub,k}(vec_{i,k})$ in I_{tot}^{avg} in Equation (5.2). If $I_{sub,k}(vec_{i,k})$ corresponds to the p^{th} model in the LUT for PDF of subthreshold leakage current and it is located in the l^{th} grid, then $Prob(vec_{i,k}) \cdot I_{sub,k}(vec_{i,k})$ can be written as $\alpha e^{U_{0,p}+\beta_{1,p}\cdot\Delta L_{eff}^{l}+\beta_{2,p}\cdot\Delta T_{ox,k}}$, where the values of $U_{0,p}$, $\beta_{1,p}$ and $\beta_{2,p}$ come from the p^{th} model in the LUT; the coefficient α is $Prob(vec_{i,k}) \cdot W_{eff,k} \cdot c_p$, where c_p is the coefficient from the p^{th} model; ΔL_{eff}^{l} represents the variation of L_{eff} in the l^{th} grid in the spatial correlation model, and $\Delta T_{ox,k}$ the variation of T_{ox} at this gate.

As we write the summation over all these lognormals, we observe that several different gates within the circuit may use the same LUT model: in fact, in general, the number of models is dramatically smaller than the total number of gates, and in practice, can be upper-bounded by a constant. Let $I_{sub,p,l} = \{I_{sub,p,l}^{1}, \cdots, I_{sub,p,l}^{s}\}$, where s is the size of the set, be the group of all subthreshold leakage current terms that use the p^{th} model in the LUT and lie in the l^{th} grid. Obviously, any $I_{sub,p,l}^{j}$ can be expressed in the form of:

$$I_{sub,p,l}^{j} = \alpha_j e^{U_{0,p}+\beta_{1,p}\cdot\Delta L_{eff}^{l}+\beta_{2,p}\cdot\Delta T_{ox,j}} \tag{5.19}$$

Note that each $I_{sub,p,l}^{j}$ has the same values of $U_{0,p}$, $\beta_{1,p}$ and $\beta_{2,p}$ from the p^{th} model,

but the values of α_j may be different for different $I^j_{sub,p,l}$ terms, corresponding to different probabilities of occurrence, or different transistor widths. All $I^j_{sub,p,l}$ terms share the same variable ΔL^l_{eff} since they are in the same l^{th} grid, but each $I^j_{sub,p,l}$ has a different $\Delta T_{ox,j}$ variable, with all such $\Delta T_{ox,j}$ variables being independent of each other (since the values of gate oxide thickness are uncorrelated from gate to gate).

Then, the sum of all terms in $I_{sub,p,l}$ can be then written as:

$$I^{sum}_{sub,p,l} = \sum_{j=1}^{s} I^j_{sub,p,l} = e^{U_{0,p}+\beta_{1,p}\cdot\Delta L^l_{eff}} \cdot \sum_{j=1}^{s} \alpha_j \cdot e^{\beta_{2,p}\cdot\Delta T_{ox,j}} \tag{5.20}$$

Due to the independence of the $T_{ox,j}$ variables, the sum $\sum_{j=1}^{s} \alpha_j \cdot e^{\beta_{2,p}\cdot\Delta T_{ox,j}}$ is in fact a sum of independent lognormal random variables. As explained earlier in the description of Wilkinson's method, the sum of independent lognormal random variables can be approximated by a lognormal random variable with computational complexity linear to the number of independent lognormals. Therefore, the product of the term, $e^{U_{0,p}+\beta_{1,p}\cdot\Delta L^l_{eff}}$, with the lognormal approximation of $\sum_{j=1}^{s} \alpha_j \cdot e^{\beta_{2,p}\cdot\Delta T_{ox,j}}$ is also approximated as a lognormal, and the computational complexity of performing this calculation is $O(s)$.

Now that each $I^{sum}_{sub,p,l}$ is approximated as a lognormal random variable, the full-chip leakage can be calculated as the sum

$$\sum_{p=1}^{N_{models}} \sum_{l=1}^{n} I^{sum}_{sub,p,l}, \tag{5.21}$$

where N_{models} is the total number of models in the library, and n is the number of grid partitions in the spatial correlation model. Note that any two $I^{sum}_{sub,p,l}$ terms may be correlated due to spatial correlations of the process parameter L_{eff}, and thus the computational complexity of the sum is $O(N^2_{models}\cdot n^2)$. Since the number of different

models of a library is upper-bounded by a constant, and the number of grids is substantially smaller than the number of gates in the circuit, the computational complexity for estimating the distribution of full-chip leakage current is reduced from $O(N_g^2)$ to a substantially smaller constant $O(N_{models}^2 \cdot n^2)$.

Handling Correlations Between Leakage Currents in Different Groups

As described in the previous subsection, in order to reduce the number of correlated lognormals to sum, the leakage current terms are summed in groups, where each group is a set of terms that correspond to the same grid and the same model from the LUT. Let $I_{p1,l}^{sum}$ and $I_{p2,l}^{sum}$ be the results of two grouped sums that are both in the same l^{th} grid, and utilizing models $p1$ and $p2$ from the LUT, respectively. According to Equation (5.20), they can be computed as:

$$I_{p1,l}^{sum} = e^{U_{0,p1}+\beta_{1,p1}\cdot\Delta L_{eff}^{l}} \cdot \sum_{j=1}^{s1} \alpha_{j,p1} \cdot e^{\beta_{2,p1}\cdot\Delta T_{ox,j}} = e^{U_{0,p1}+\beta_{1,p1}\cdot\Delta L_{eff}^{l}} \cdot e^{\xi} \quad (5.22)$$

$$I_{p2,l}^{sum} = e^{U_{0,p2}+\beta_{1,p2}\cdot\Delta L_{eff}^{l}} \cdot \sum_{j=1}^{s2} \alpha_{j,p2} \cdot e^{\beta_{2,p2}\cdot\Delta T_{ox,j}} = e^{U_{0,p2}+\beta_{1,p2}\cdot\Delta L_{eff}^{l}} \cdot e^{\gamma} \quad (5.23)$$

where $s1$ and $s2$ are the number of terms in $I_{p1,l}^{sum}$ and $I_{p2,l}^{sum}$, respectively. The term e^{ξ} is the random variable approximating $\sum_{j=1}^{s1} \alpha_{j,p1} \cdot e^{\beta_{2,p1}\cdot\Delta T_{ox,j}}$, and e^{γ} for $\sum_{j=1}^{s2} \alpha_{j,p2} \cdot e^{\beta_{2,p2}\cdot\Delta T_{ox,j}}$, as described in the previous subsection.

It should be noted that $\sum_{j=1}^{s1} \alpha_{j,p1} \cdot e^{\beta_{2,p1}\cdot\Delta T_{ox,j}}$ and $\sum_{j=1}^{s2} \alpha_{j,p2} \cdot e^{\beta_{2,p2}\cdot\Delta T_{ox,j}}$ may be correlated. This is because although $I_{p1,l}^{sum}$ and $I_{p2,l}^{sum}$ correspond to different models in the LUT, they may include leakage currents of the same gate, and obviously leakage currents associated with the same transistors are correlated. Therefore, e^{ξ} and e^{γ} are correlated, and the correlation between ξ and γ must be considered while adding up the sums of all groups for full-chip leakage current calculation.

The correlation between e^{ξ} and e^{γ} can be computed by:

$$\begin{aligned} cov(e^{\xi}, e^{\gamma}) &= E(e^{\xi+\gamma}) - E(e^{\xi})E(e^{\gamma}) \\ &= e^{\mu_\xi+\mu_\gamma+(\sigma_\xi^2+\sigma_\gamma^2)/2}(e^{cov(\xi,\gamma)/2} - 1) \end{aligned} \tag{5.24}$$

where μ_ξ [μ_γ] and σ_γ [σ_γ] are the mean and standard deviation of ξ [γ], respectively.

Thus, the covariance between ξ and γ can be obtained by solving Equation (5.24) for $cov(\xi, \gamma)$:

$$cov(\xi, \gamma) = 2\log\left(1 + \frac{cov(e^{\xi}, e^{\gamma})}{e^{\mu_\xi+\mu_\gamma+(\sigma_\xi^2+\sigma_\gamma^2)/2}}\right) \tag{5.25}$$

In Equation (5.25), the mean and standard deviation of ξ and γ are known values. Since e^{ξ} and e^{γ} are approximations of $\sum_{j=1}^{s1} \alpha_{j,p1} \cdot e^{\beta_{2,p1}\cdot\Delta T_{ox,j}}$ and $\sum_{j=1}^{s2} \alpha_{j,p2} \cdot e^{\beta_{2,p2}\cdot\Delta T_{ox,j}}$, respectively, the value of $cov(e^{\xi}, e^{\gamma})$ can be obtained as:

$$cov(e^{\xi}, e^{\gamma}) = cov\left(\sum_{j=1}^{s1} \alpha_{j,p1} \cdot e^{\beta_{2,p1}\cdot\Delta T_{ox,j}}, \sum_{j=1}^{s2} \alpha_{j,p2} \cdot e^{\beta_{2,p2}\cdot\Delta T_{ox,j}}\right) \tag{5.26}$$

Note that any two $\Delta T_{ox,j}$ variables are independent, and thus the value of the above right hand side can easily be computed as:

$$\sum_j \alpha_{j,p1} \cdot \alpha_{j,p2} \cdot e^{(\beta_{2,p1}^2+\beta_{2,p2}^2)\sigma_{T_{ox,j}}^2/2} \cdot (e^{\beta_{2,p1}\cdot\beta_{2,p2}\cdot\sigma_{T_{ox,j}}^2} - 1) \tag{5.27}$$

where $\sigma_{T_{ox,j}}$ is the standard deviation of $\Delta T_{ox,j}$.

Handling Inter-die Variations

The described framework for statistical computation of full-chip leakage considering spatial correlations in intra-die variations of process parameters can easily be extended to handle inter-die variations. To include the effects of inter-die variations, for each type of process parameter, a global random variable can be applied

to all gates in the circuit to model this effect. For spatially correlated process parameters, this is reflected as an update of the covariance matrix by adding to all entries the variance of the global random variable. For spatially uncorrelated process parameters, it introduces a correlation term between the leakage currents of different gates. However, the same framework of estimating the distribution of full-chip leakage current for handling intra-die variations proposed in Section 5.3 can be applied.

5.4 An Improved Algorithm, Hybridized with the PCA-based Approach

In previous sections, we proposed to improve the computational complexity by reducing the number of correlated lognormals to sum. Another possible approach is to modify the structure of each lognormal random variable so that the summation can be computed efficiently, as was done using a PCA-based method in the work of [71]. In this section, we will first present the method proposed in [71], and an improved method hybridized with the PCA-based approach will be proposed in the following section.

5.4.1 PCA-based Method

The work of [71] proposes a PCA-based method to compute the full-chip leakage considering the effect of spatial correlations of L_{eff}. The principle of the method is very similar to the PCA-based statistical timing analysis introduced in Chapter 3. In this method, our proposed spatial correlation model introduced in Chapter 2 is used. The leakage current of each gate is approximated by a lognormal random

variable in a form similar to expression (5.5) or (5.9)[1], and then the expression is rewritten in a "PCA form" by expanding the variable ΔL_{eff} as a linear function of principal components. For example, let I^i_{sub} be the subthreshold leakage current of the i^{th} gate originally written in a form similar to Equation (5.5) as:

$$I^i_{sub} = e^{U_{0,i}+\beta_{1,i}\cdot\Delta L^l_{eff}+\beta_{2,i}\cdot\Delta T_{ox,i}} \tag{5.28}$$

Here, ΔL^l_{eff} is the random variable for the variation of L_{eff} in the l^{th} grid, and $\Delta T_{ox,i}$ is the variation of T_{ox} at the i^{th} gate. Note that for any $i \neq j$, $\Delta T_{ox,i}$ and $\Delta T_{ox,j}$ are independent since T_{ox} is spatially uncorrelated.

If principal component analysis is performed on the set of correlated variables $\Delta L^1_{eff}, \cdots, \Delta L^n_{eff}$, as explained in Section 3.3.2, then ΔL^l_{eff} can be expressed as a linear function of the set of principal components:

$$\Delta L^l_{eff} = a_{l1} \times L^{'1}_{eff} + \cdots + a_{lN_p} \times L^{'N_p}_{eff} \tag{5.29}$$

where the $L^{'j}_{eff}$ variables are the mutually independent principal components computed from the covariance matrix of $\Delta L^1_{eff}, \cdots, \Delta L^n_{eff}$, the coefficients a_{lj} of each $L^{'j}_{eff}$ are computed from principal component analysis, and N_p is the number of principal components.

Then, the PCA form of I^i_{sub} is:

$$I^i_{sub} = e^{U_{0,i}+\sum_{t=1}^{N_p} k^i_t\cdot L^{'t}_{eff}+\beta_{2,i}\cdot\Delta T_{ox,i}} \tag{5.30}$$

where each $k^i_t = a_{l1}\cdot\beta_{1,i}$ can be computed by comparing this equation with Equation (5.29).

[1]In [71], only process parameter L_{eff} is considered and an independent uncertainty term is introduced for ΔL_{eff}. For convenience, we do not distinguish such differences, since these factors can easily be considered and incorporated in any framework.

In [71], the sum $I^i_{sub} + I^j_{sub}$ is reapproximated again by a lognormal random variable I^h_{sub} in PCA form:

$$I^h_{sub} = e^{U_{0,h} + \sum_{t=1}^{N_p} k^h_t \cdot L'^t_{eff} + \beta^h_r \cdot r} \tag{5.31}$$

where r is a normalized Gaussian random variable generated by merging the two terms ΔT^i_{ox} and ΔT^j_{ox}, and β^h_r is the coefficient of r.

In Equation (5.31), the value of $U^{0,h}$ can be directly computed using Wilkinson's formula (5.17). The other coefficients can be obtained using the following expressions:

$$\begin{aligned} k^h_t &= log \frac{E(I^i_{sub} \cdot e^{L'^t_{eff}}) + E(I^j_{sub} \cdot e^{L'^t_{eff}})}{[E(I^i_{sub}) + E(I^j_{sub})] E(e^{L'^t_{eff}})} \\ \beta^h_r &= \left[log \left(1 + \frac{Var(I^i_{sub}) + Var(I^j_{sub}) + 2cov(I^i_{sub}, I^j_{sub})}{(I^i_{sub} + I^j_{sub})^2} \right) - \sum_{t=1}^{N_p} (k^h_t)^2 \right]^{0.5} \end{aligned} \tag{5.32}$$

Here, $E(.)$, $Var(.)$ and $cov(I^i_{sub}, I^j_{sub})$ can be computed using Equations (5.12), (5.13), and (5.14). Note that all terms in Equation (5.32) are in PCA form. The benefit of using a PCA form is that the mean and variance of a lognormal random variable can be computed in $O(N_p)$, as can the covariance of two lognormal random variables in PCA form. Therefore, the computation of all values and coefficients in I^h_{sub}, and thus the sum of two lognormals in PCA form, can be computed in $O(N_p)$. As mentioned in the description of Wilkinson's method, the computation of full-chip leakage current distribution requires a summation of N_g correlated lognormals. Thus, the PCA-based method has an overall computational complexity of $O(N_p \cdot N_g)$.

5.4.2 Hybridization with the PCA-based Approach

In this section, we will present an improved algorithm by hybridizing the basic approach proposed in Section 5.3 with the PCA-based method in [71].

We summarize the similarities and differences between the basic approach and the PCA-based method as follows. Both methods use Wilkinson's method to approximate sum of lognormal random variables. The basic approach in Section 5.3 improves run-time by reducing the number of correlated lognormals to sum, by first calculating the sum of leakage currents by groups, where each group contains leakage currents in the same grid and using the same LUT model, and then computing full-chip leakage by summing up leakage currents in all groups. The computational complexity of this approach is $O(n^2 \cdot N_{models}^2)$, where n is the number of grids partitioned in the spatial correlation model and N_{model} is the number of models in the LUT. The PCA-based method reexpresses each lognormal random variable in PCA form, and then directly computes the summation of all correlated lognormals using Wilkinson's method in $O(N_g \cdot N_p)$, where N_g is the total number of gates in the circuit and N_p is the number of principal components.

Similar to the basic approach, the improved algorithm proposed will compute the full-chip leakage current hierarchically in groups, and the sum of leakage current terms in each group will be computed in a more efficient way as in the PCA-based approach:

First, the average total leakage current of each gate in the circuit is computed as $\sum_{\forall vec_{i,k}} Prob(vec_{i,k}) \cdot (I_{sub,k}(vec_{i,k}) + I_{gate,k}(vec_{i,k}))$, as defined in Equation (5.1) and (5.2). By using the models from the LUT, the average total leakage current becomes a weighted sum of several leakage current terms, and the number of the terms is no more than N_{models}. In general, if the gate is located in the l^{th} grid,

then any leakage current term can be written in the form $e^{U_0+\beta_1\cdot\Delta L^l_{eff}+\beta_2\cdot\Delta T_{ox,k}}$. If we reapproximate the sum of any two leakage current terms in the same form, Equation (5.32) can be utilized to compute the desired values in the approximation. This is because the process parameters of all transistors in the same gate are fully correlated, so that ΔL^l_{eff} and $\Delta T_{ox,k}$ can be regarded as global random variables in the same gate. Thus, Equation (5.32) can easily be reused by regarding ΔL^l_{eff} and $\Delta T_{ox,k}$ as principal components in the formula. Obviously, the complexity for summing any two leakage current terms in the same gate is $O(1)$, and thus the computation of the average total leakage current of a gate is $O(N_{models})$. If the total number of gates in the circuit is N_g, then the computational complexity of this step is $O(N_{models} \cdot N_g)$.

Next, the total leakage current in each grid is computed separately. Clearly, for all gates in the l^{th} grid, any average leakage current of a gate is expressed as an exponential function of the same random variable ΔL^l_{eff}, while the average leakage current terms for different gates correspond to different $\Delta T_{ox,k}$ variables: note that all $\Delta T_{ox,k}$ variables are independent. The sum of average leakage current of any two gates can be approximated in a manner similar to that used in computing the average leakage current of a single gate, using the formula (5.32) by regarding ΔL^l_{eff} as a principal component. Therefore, the sum has a computational complexity of $O(1)$. Since this step must compute the total leakage current of all gates in all grids, the computation complexity is $O(N_g)$.

Finally, the full-chip leakage is computed by adding up the total leakage currents computed in all grids. If the number of grids is n, n correlated lognormals, with a complicated correlation structure, must be summed up. Therefore, we transform all lognormals in the summation into PCA forms, and the sum can be computed using the same methodology proposed in [71]. The computation complexity of this

Table 5.1: Comparison of the proposed basic method with Monte Carlo simulation.

Circuit Name	Gate Number	Grid Number	Total Circuit Leakage Current (μA)									
			Monte Carlo (MC)		Basic Method		Error%		MCNoCorr		Error%	
			mean	std	mean	std	mean	std	mean	std	mean	std
c7552	5528	64	327.9	106.1	324.3	101.0	-1.1%	-4.9%	327.8	90.7	0.0%	-14.5%
c5315	3887	64	239.0	78.4	235.7	74.3	-1.4%	-5.2%	239.5	67.2	0.2%	-14.3%
c6288	2672	16	229.6	77.3	227.7	78.0	-0.8%	0.8%	229.7	71.8	0.0%	-7.1%
c3540	2606	16	158.9	53.4	156.8	50.9	-1.3%	-4.7%	158.3	44.1	-0.4%	-17.4%
c2670	1925	16	113.7	37.8	112.6	36.6	-1.0%	-3.3%	113.9	31.7	0.2%	-16.3%
c1908	1261	16	73.5	24.9	72.3	23.5	-1.6%	-5.6%	73.2	20.1	-0.4%	-19.1%
c880	594	4	37.4	13.3	36.9	12.7	-1.3%	-4.6%	37.3	10.5	-0.3%	-21.4%
c432	294	4	18.3	6.5	17.9	6.2	-1.8%	-5.0%	18.2	5.1	-0.4%	-21.5%

step is $O(N_p \cdot n)$.

From the analysis above, the total computational complexity of the improved algorithm is $O(N_p \cdot n + (N_{models} + 1) \cdot N_g) = O(N_p \cdot n + N_g)$. This is better than the complexity of $O(N_g \cdot N_p)$ for the PCA-based method, since the number of grids n is substantially smaller than the number of gates N_g in the circuit. If n is a small constant, the basic approach which has a computational complexity of $O(n^2 \cdot N_{models}^2)$ which may outperform the improved approach. However, as n grows to a relatively larger number, the basic approach grows quadratically with n, while improved approach grows linearly which results in a better run-time for the improved approach, as compared to the basic method.

5.5 Experimental Results

In this section, the experimental results for full-chip statistical leakage estimation will be presented. The results using the basic approach proposed in Section 5.3 will be first provided, followed by those using the improved method in Section 5.4.

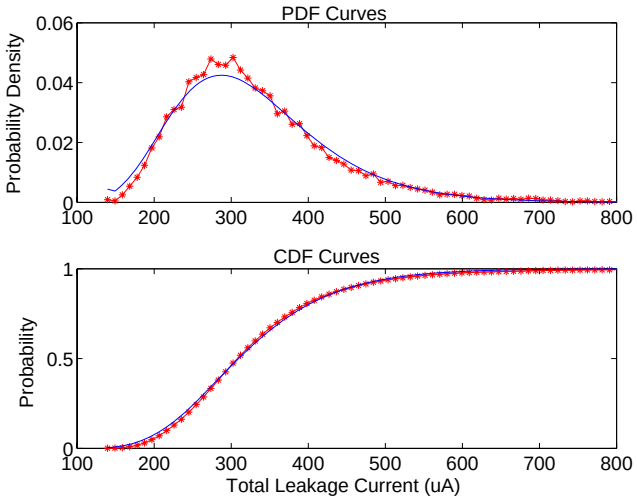


Figure 5.3: Distributions of the total leakage using the proposed basic method against Monte Carlo simulation method for circuit c7552. The solid line illustrates the result of the proposed basic method, while the starred line shows the Monte Carlo simulation results.

Our experiments were performed on the set of circuits in the ISCAS85 benchmark set. The circuits were synthesized with SIS with a cell library consisting of an inverter, and NAND, NOR, AND, and OR gates with 2, 3 and 4 input pins. The designs were placed using Capo [77]. The technology parameters that were used correspond to the 100nm Berkeley Predictive Technology model [76], and the 3σ value of parameter variations for L_{eff} and T_{ox} were set to 20% of the nominal parameter values, of which inter-die variations constitute 40% and intra-die variations 60%. The spatial correlation was modeled so that the correlation coefficient value diminishes equally with the distance between any two grids, as in Chapter 3. The number of grid partitions in the spatial correlation model used for each benchmarks is listed in Table 5.1, and depends on the size of the circuit.

For comparison purposes, we performed Monte Carlo simulations with 10,000

runs on the benchmarks. First, we present the experimental results of the proposed basic method for full-chip leakage estimation introduced in Section 5.3. The results of the comparison of this method with the Monte Carlo approach are shown in Table 5.1. The average errors for the mean and sigma values are -1.3% and -4.1%, respectively. In Figure 5.3, we show the distribution of total circuit leakage current achieved using the proposed basic method and using Monte Carlo simulation for circuit c7552: it is easy to see that the curve achieved by the basic method matches well with the Monte Carlo simulation result. For all testcases, the run-time of the basic method is less than one second, while the Monte Carlo simulation takes considerably longer: for the largest test case, c7552, this simulation takes 3 hours.

To show the importance of considering spatial correlations, we run another set of Monte Carlo simulations (*MCNoCorr*) on the same set of benchmarks, assuming correlation coefficients of zero between the intra-die variations of effective gate length L_{eff} of any two gates on the chip. The comparison data is also shown in Table 5.1. It can be observed that although the mean values are close, on average, the variances of *MCNoCorr*, where spatial correlations are ignored, has a underestimation of 16.5% compared to *MC*, where the spatial correlations are taken into account. This is because the leakage values of different gates are less correlated when spatial correlations are ignored, and thus different gates have lower probabilities of taking larger values of leakage simultaneously, which results in smaller overall variations.

To visualize the difference, in Figures 5.4 and 5.5, for circuit c432, we show the scatter plots for 2000 samples of full-chip leakage current generated by Monte Carlo simulations, with and without consideration of spatial correlations of L_{eff}. The x-axis marks the multiples of the standard deviation value of ΔL_{eff}^{inter}, inter-die variations of effective gate length, ranging from -3 to $+3$, since a Gaussian

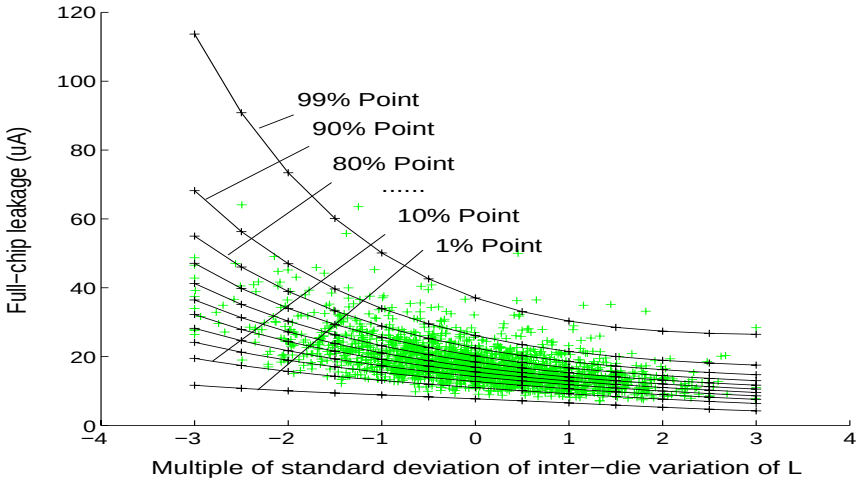


Figure 5.4: Scatter plot of full-chip leakage considering spatial correlation for circuit c432

distribution is assumed. The y-axis are the values of total circuit leakage current. Therefore, at each specific value of ΔL_{eff}^{inter}, the scatter points list the various sampled values of total circuit leakage current due to variations in T_{ox} and intra-die variation of L_{eff}. The plots also show a set of contours lines that correspond to, with the effect of spatial correlation taken into account, a set of percentage points of the CDF of total circuit leakage current at different values of ΔL_{eff}^{inter}. In Figure 5.4, where spatial correlations are considered, nearly all points generated from Monte Carlo simulation fall between the contours of the 1% and 99% lines. However, in Figure 5.5, where spatial correlations are ignored, the spread is much tighter in general: the average value of 90% point of full-chip leakage, with spatial correlation considered, is 1.5 times larger than that without for $\Delta L_{eff}^{inter} \leq -1\sigma$; the same ratio is 1.1 times larger otherwise. Looking at the same numbers in a different way, in Figure 5.5, all points are contained between the 30% and 80% contours if $\Delta L_{eff}^{inter} \leq -1\sigma$. In this range, I_{sub} is greater than I_{gate} by one order of magnitude on average, and thus the variation of L_{eff} can have a large effect on the total

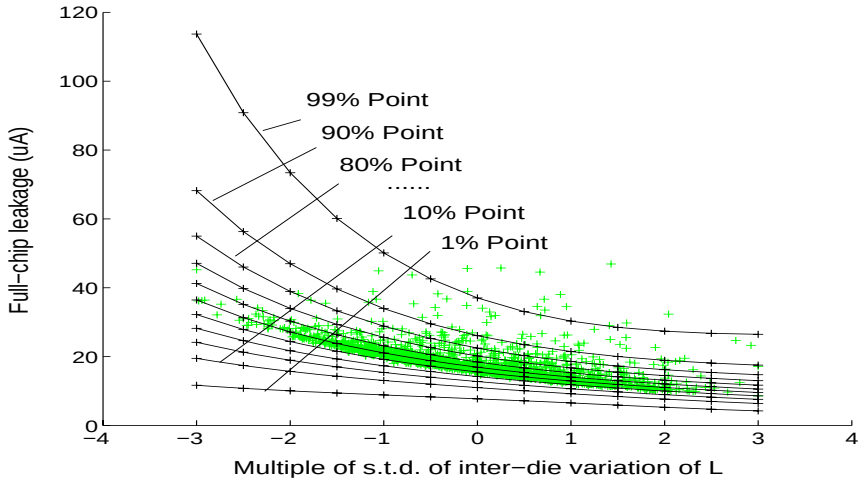


Figure 5.5: Scatter plot of full-chip leakage ignoring spatial correlation for circuit c432

leakage as I_{sub} is exponentially dependent on L_{eff}. Consequently, ignoring spatial correlation results in a substantial underestimation of the standard deviation, and thus the worst-case full-chip leakage. For $\Delta L_{eff}^{inter} > -1\sigma$, I_{sub} decreases to a value comparable to I_{gate} and L_{eff} has a relatively weak effect on the variation of total leakage. In this range, the number of points of larger leakage values is similar to that when spatial correlation is considered. However, a large number of remaining points show smaller variations and are within the 20% and 90% contours, due to the same reasoning given above for $\Delta L_{eff}^{inter} \leq -1\sigma$.

We also study the effect by varying L_{eff} and T_{ox} separately on the variations of full-chip subthreshold and gate-tunneling leakage currents. In Table 5.2, the results by varying L_{eff} only keeping T_{ox} at its nominal value are provided in columns 2 to 7, and the last 6 columns show the reverse. As seen in the table, the variations of L_{eff} and T_{ox} can each individually lead to substantial variations in the full-chip leakage. When only L_{eff} varies, I_{sub} varies substantially (the average ratio of the

Table 5.2: Comparison of leakage by varying L_{eff} and T_{ox} independently

Circuit Name	Leakage by varying effective gate length only (μA)						Leakage by varying gate oxide thickness only (μA)					
	I_{total}		I_{sub}		I_{gate}		I_{total}		I_{sub}		I_{gate}	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
c7552	268.2	81.3	216.2	83.8	52.0	2.7	298.9	63.1	195.1	34.0	103.8	88.2
c5315	194.3	60.6	155.3	62.5	39.0	2.0	217.4	47.6	139.5	24.4	77.9	65.8
c6288	178.5	46.7	131.2	49.1	47.4	2.6	215.0	63.8	120.4	19.6	94.6	79.2
c3540	129.4	42.2	103.3	43.6	26.1	1.5	144.4	31.7	92.9	15.9	51.5	43.7
c2670	92.9	29.9	74.6	30.8	18.3	1.0	103.4	21.9	67.2	11.5	36.2	30.4
c1908	60.4	20.5	49.2	21.1	11.2	0.6	66.5	13.1	44.0	7.6	22.5	18.8
c880	30.6	10.9	24.5	11.2	6.1	0.4	34.1	7.5	22.0	3.8	12.1	10.4
c432	15.1	5.6	12.5	5.8	2.6	0.2	16.4	3.1	11.2	2.0	5.3	4.5
Avg	121.2	37.2	95.9	38.5	25.3	1.4	137.0	31.5	86.5	14.9	50.5	42.6

mean to the standard deviation is 40.2%) and I_{gate} trivially (the corresponding ratio is 5.5%), since I_{sub} is more sensitive to the variation of L_{eff} than T_{ox}, and I_{gate} is a strong exponential function of T_{ox} over L_{eff}. In this case, I_{sub} dominates I_{gate} by 4 to 5 times and the variation of full-chip leakage is mainly due to I_{sub}. In contrast, when only T_{ox} varies, the mean of I_{gate} doubles and standard deviation increases by 40 times, while standard deviation of I_{sub} is about 3 times smaller compared to the former case. In this case, although the mean of I_{gate} is about two times smaller than that of I_{sub}, its standard deviation is 3 times larger than that of I_{sub}. Therefore, in this case, although I_{sub} and I_{gate} are both major contributors to the full-chip leakage, the leakage variations are mainly due to I_{gate}.

Since the proposed basic, improved method, and the PCA-based approach are all based on Wilkinson's approximation, the accuracies of these approaches for total chip leakage estimations are essentially the same. A tabular comparison of accuracies is not provided and only the run-time efficiencies of the three methodologies are compared in this section. In Table 5.3 and 5.4, we show the run-times for different

Table 5.3: Run-time comparison of the proposed basic, PCA-based, and improved methods for the ISCAS85 benchmarks

Benchmark	c432	c880	c1908	c2670	c3540	c6288	c5315	c7552
Number of grids	4	4	16	16	16	16	64	64
Proposed basic method (s)	0.01	0.02	0.04	0.06	0.09	0.10	0.24	0.29
PCA-based method (s)	0.03	0.06	0.18	0.27	0.40	0.57	1.43	1.82
Proposed improved method (s)	0.01	0.03	0.06	0.09	0.12	0.14	0.19	0.25

Table 5.4: Run-time comparison of the proposed basic, PCA-based, and improved methods for the ISCAS89 benchmarks

Benchmark	s5378	s9234	s13207	s15850	s35932	s38584
Number of grids	64	64	256	256	256	256
Proposed basic method (s)	0.22	0.32	5.89	5.91	4.97	10.04
PCA-based method (s)	0.93	1.62	7.58	8.97	17.38	24.28
Proposed improved method (s)	0.16	0.30	0.47	0.56	1.03	1.34

methods for ISCAS85 and ISCAS89 benchmark sets, respectively. In general, the proposed basic method is about 3 to 4 times faster than the PCA-based method. As expected, the proposed improved approach does not show any run-time advantage over the basic method for smaller grid sizes. However, run-time of both the proposed basic and the PCA-based methods grows much faster with the grid size than the improved method. In Table 5.3 and 5.4, when the number of grids grows to greater than 64, the improved approach is about 100 times faster than the other approaches. Therefore, the run-time can be significantly improved by combining the PCA-based with the proposed basic leakage estimation approach.

5.6 Conclusions

We have presented a method for analyzing the leakage current distribution of circuit under process parameter variations considering the spatial correlations among parameters. The proposed method was shown to be effective in predicting the mean, standard deviation and the PDF of the total chip leakage. We have also shown that the spatial correlations of process parameters must be considered appropriately in order to predict yield of chip correctly. We believe that this framework is general to predict the total circuit leakage under other parameter variations. For example, leakage has a strong dependence on temperature and the variation of temperature is also highly spatially correlated. If the correlation statistics are available, this method can easily be extended to capture the effects of temperature variations.

Chapter 6

Conclusion

In current and future technologies, the increasing number and magnitude of process variations make the prediction of circuit performance an important but very challenging task. As the conventional corner-based technique becomes too pessimistic and slow, statistical circuit performance analysis techniques provide a good alternative.

In this thesis, we have focused on the problem of statistical circuit timing and leakage power estimation with inter-die and intra-die variations. The effects of spatial correlations in intra-die variations, which were ignored in most of the previous works, are also considered in our works. We show that spatial correlation is essential in order to correctly predict the probability distributions of circuit timing and leakage power. The statistical timing and leakage power methods presented in the thesis are shown to be both computationally efficient and accurate, and this is demonstrated through comparisons against Monte Carlo simulations. The timing and leakage power estimation techniques are important, both for yield prediction in the post-layout stage, as well as for supporting circuit design and optimization

in all stages of the design flow for shortening the design cycle and saving design costs.

Although in recent years, quite some work has been done in statistical circuit performance analysis for timing and leakage, this area still requires further research. First, statistical performance analysis technique requires proper modeling of process variations including the decomposition and modeling of process variations including spatial correlations. Without an appropriate model, the prediction by statistical analysis could be a "garbage in and garbage out," the result would not make much sense and cannot guide the circuit optimization in the correct direction. Second, the statistical timing analysis technique depends on correct characterization of gate/interconnect delay with respect to process parameter variations. A library that is characterized with worst-case and best-case corners must be recharacterized, such as characterizing with nominal value and sensitivities to process variations, in order to have accurate statistical timing analyzer. Third, although statistical performance analysis methods are more computationally efficient than corner-based methods and Monte Carlo approaches, they also show a tradeoff between accuracy and run-time. This may not be a problem if this is solely for the purpose of performance analysis. However, in order for the method to be integrated into a framework for circuit performance optimization, a good balance is required between the run-time and the accuracy. Finally, variation-aware circuit optimization techniques [10, 21, 34, 35, 50, 63, 72] that can take into account process variations are active fields for research and development. The technique should be applied across the overall flow of circuit design, including steps such as technology mapping [68], synthesis, buffer insertion [26], clock tree [49], physical design [1, 31], to overcome the limitations of traditional deterministic optimization techniques.

Bibliography

[1] C. Ababei and K. Bazargan. Timing minimization by statistical timing hMetis-based partitioning. In *Proceedings of International Conference on VLSI Design*, pages 58–63, New Delhi, India, 2003.

[2] A. A. Abu-Dayya and N. C. Beaulieu. Comparison of methods of computing correlated lognormal sum distributions and outages for digital wireless applications. In *IEEE 44th Vehicular Technology Conference, vol. 1*, pages 175–179, June 1994.

[3] E. Acar, A. Devgan, R. Rao, Y. Liu, H. Su, S. Nassif, and J. Burns. Leakage and leakage sensitivity computation for combinational circuits. In *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 96 – 99, Seoul, Korea, August 2003.

[4] A. Agarwal, D. Blaauw, and V. Zolotov. Statistical timing analysis for intra-die process variations with spatial correlations. In *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, pages 900–907, San Jose, California, USA, November 2003.

[5] A. Agarwal, D. Blaauw, V. Zolotov, S. Sundareswaran, M. Zhao, K. Gala, and R Panda. Statistical delay computation considering spatial correlations.

In *Proceedings of the Asia and South Pacific Design Automation Conference*, pages 271–276, Kitakyushu, Japan, January 2003.

[6] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula. Computation and refinement of statistical bounds on circuit delay. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 348–353, Anaheim, California, USA, June 2003.

[7] Semiconductor Industry Association. International technology roadmap for semiconductors. Available at: `http://public.itrs.net`, 1997-2005.

[8] V. Axelrad and J. Kibarian. Statistical aspects of modern IC designs. In *Proceedings of the 28th European Solid-State Device Research Conference*, pages 309–321, Bordeaux, France, September 1998.

[9] M. Berkelaar. Statistical delay calculation, a linear time method, 1997. (Personal communication).

[10] M. R. C. M. Berkelaar and J. A. G. Jess. Gate sizing in MOS digital circuits with linear programming. In *Proceedings of European Design Automation Conference*, pages 217–221, Glasgow, Scotland, March 1990.

[11] S. Bhardwaj, S. B. K. Vrudhula, and D. Blaauw. τAU: Timing analysis under uncertainty. In *Proceedings of the ACM/IEEE International Conference on Computer Aided Design*, pages 615–620, San Jose, California, USA, November 2003.

[12] D. S. Boning and S. Nassif. Chapter 6: Models of process variations in device and interconnect. In W. Bowhill, A. Chandrakasan, and F. Fox, editors, *Design of High Performance Microprocessor Circuits*. IEEE Press, 2000.

[13] D. S. Boning, J. Panganiban, K. Gonzalez-Valentin, S. Nassif, C. McDowell, A. Gattiker, and F. Liu. Test structures for delay variability, 2002. (Personal communication).

[14] S. Borkar, T. Karnik, and V. De. Design and reliability challenges in nanometer technologies. In *Proceedings of the ACM/IEEE Design Automation Conference*, page 75, San Diego, California, USA, June 2004.

[15] K. A. Bowman, L. Wang, X. Tang, and J. D. Meindl. A circuit level perspective of the optimum gate oxide thickness. *IEEE Transction on Electron Devices*, 48(8):1800 – 1810, August 2001.

[16] R. B. Brawhear, N. Menezes, C. Oh, L. Pillage, and R. Mercer. Predicting circuit performance using circuit-level statistical timing analysis. In *Proceedings of European Design and Test Conference,*, pages 332–337, Paris, France, 1994.

[17] H. Chang and S. S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single PERT-like traversal. In *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, pages 621–625, San Jose, California, USA, November 2003.

[18] H. Chang and S. S. Sapatnekar. Full-chip analysis of leakage power under process variations, including spatial correlations. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 523–528, Anaheim, California, USA, June 2005.

[19] H. Chang, V. Zolotov, S. Narayan, and C. Visweswariah. Parameterized block-based statistical timing analysis with non-gaussian parameters, nonlinear delay functions. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 71–76, Anaheim, California, USA, June 2005.

[20] B. Choi and D. M. H. Walker. Timing analysis of combinational circuits including capacitive coupling and statistical process variation. In *Proceedings of the IEEE VLSI Test Symposium*, pages 49–54, Montreal, Canada, April 2000.

[21] S. H. Choi, B. C. Paul, and K. Roy. Novel sizing algorithm for yield improvement under process variation in nanometer technology. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 454 – 459, San Diego, California, USA, June 2004.

[22] C. E. Clark. The greatest of a finite set of random variables. *Operations Research*, 9:145–162, March-April 1961.

[23] N. Cobb, A. Zakhor, and E. Miloslavsky. A mathematical and CAD framework for proximity correction. In *Proceedings of SPIE Symposium on Optical Microlithography*, pages 208–222, Santa Clara, California, USA, March 1996.

[24] J. Cong. Challenges and opportunities for design innovations in nanometer technologies. In *Semiconductor Research Corporation Design Sciences Concept Paper*, pages 1–15, January 1998.

[25] Y. Deguchi, N. Ishiura, and S. Yajima. Probabilistic CTSS: Analysis of timing error probability in asynchronous logic circuits. In *Proceedings of IEEE/ACM Design Automation Conference*, pages 650–655, San Francisco, CA, USA, 1991.

[26] L. Deng and M. D. F. Wong. Buffer insertion under process variations for delay minimization. In *Proceedings of the IEEE/ACM International Conference on Computer-aided Design*, pages 317–321, San Jose, California, USA, 2005.

[27] A. Devgan and C. Kashyap. Block-based static timing analysis with uncertainty. In *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, pages 607–614, San Jose, California, USA, November 2003.

[28] E. Felt, A. Narayan, and A. Sangiovanni-Vincentelli. Measurement and modeling of MOS transistor current mismatch in analog ICs. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 272–277, San Jose, California, USA, November 1994.

[29] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos. Modeling within-die spatial correlation effects for process-design co-optimization. In *Proceedings of International Society for Quality Electronic Design*, pages 516 – 521, San Jose, CA, USA, March 2005.

[30] A. Gattiker, S. Nassif, R. Dinakar, and C. Long. Timing yield estimation from static timing analysis. In *Proceedings of the International Symposium on Quality Electronic Design*, pages 437–442, San Jose, CA, 2001.

[31] P. Gupta and A. B. Khang. Manufacturing-aware physical design. In *Proceedings of the IEEE/ACM International Conference on Computer-aided Design*, pages 681–687, San Jose, CA, USA, 2003.

[32] P. Gupta, A. B. Khang, D. Sylvester, and J. Yang. Performance-driven OPC for mask cost reduction. In *Proceedings of IEEE International Symposium on Quality Electronic Design*, pages 270–275, San Jose, California, USA, March 2005.

[33] K. Harazaki, Y. Hasegawa, Y. Shichijo, H. Tabuchi, and K. Fujii. High accurate optical proximity correction under the influences of lens aberration in $0.15\mu m$ logic process. In *International Microprocesses and Nanotechnology Conference*, pages 14 – 15, Komaba, Japan, July 2000.

[34] M. Hashimoto and H. Onodera. A performance optimization method by gate sizing using statistical static timing analysis. In *Proceedings of International*

Symposium on Physical Design, pages 111–116, San Diego, California, USA, April 2000.

[35] E. T. A. F. Jacobs and M. Berkelaar. Gate sizing using a statistical delay model. In *Proceedings of Design Automation and Test in Europe*, pages 283–290, Paris, France, March 2000.

[36] J. A. G. Jess, K. Kalafala, S. R. Naidu, R. H. J. Otten, and C. Visweswariah. Statistical timing for parametric yield prediction of digital integrated circuits. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 932–937, Anaheim, California, USA, June 2003.

[37] H.-F. Jyu, S. Malik, S. Devadas, and K. W. Keutzer. Statistical timing analysis of combinational logic circuits. *IEEE Transactions on Very Large Scale Integration Systems*, 1(2):126 – 137, June 1993.

[38] M. Ketkar and S. S. Sapatnekar. Standby power optimization via transistor sizing and dual threshold voltage assignment. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 375 – 378, San Jose, California, USA, November 2002.

[39] V. Khandelwal, A. Davoodi, and A. Srivastava. Efficient statistical timing analysis through error budgeting. In *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, pages 473–477, San Jose, CA, 2004.

[40] V. Khandelwal and A. Srivastava. A general framework for accurate statistical timing analysis considering correlations. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 89–94, Anaheim, California, USA, June 2005.

[41] T. Kirkpatrick and N. Clark. PERT as an aid to logic design. *IBM Journal of Research and Development*, 10(2):135–141, June 1966.

[42] D. Lee, W. Kwong, D. Blaauw, and D. Sylvester. Analysis and minimization techniques for total leakage considering gate oxide leakage. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 175–180, Anaheim, California, USA, June 2003.

[43] X. Li, J. Le, P. Gopalakrishnan, and L. Pileggi. Asymptotic probability extraction for non-normal distributions of circuit performance. In *Proceedings of the ACM/IEEE International Conference on Computer Aided Design*, pages 2–9, San Jose, California, USA, November 2004.

[44] R. B. Lin and M. C. Wu. A new statistical approach to timing analysis of VLSI circuits. In *Proceedings of International Conference on VLSI Design*, pages 507–513, Chennai, India, 1998.

[45] J. J. Liou, K. T. Cheng, S. Kundu, and A. Krstic. Fast statistical timing analysis by probabilistic event propagation. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 661–666, Las Vegas, Nevada, USA, June 2001.

[46] J. J. Liou, A. Krstic, L. C. Wang, and K. T. Cheng. False-path-aware statistical timing analysis and efficient path selection for delay testing and timing validation. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 566–569, New Orleans, Louisiana, USA, June 2002.

[47] Y. Liu, S. R. Nassif, L. T. Pileggi, and A. J. Strojwas. Impact of interconnect variations on the clock skew of a gigahertz microprocessor. In *Proceedings of*

the ACM/IEEE Design Automation Conference, pages 168–171, Los Angeles, California, USA, June 2000.

[48] Y. Liu, A. Zakhor, and M. A. Zuniga. Computer-aided phase shift mask design with reduced complexity. *IEEE Transactions on Semiconductor Manufacturing*, 9(2):170–181, May 1996.

[49] B. Lu, J. Hu, G. Ellis, and H. Su. Process variation aware clock tree routing. In *Proceedings of the ACM International Symposium on Physical Design*, pages 174–181, Monterey, CA, USA, April 2003.

[50] M. Mani, A. Devgan, and M. Orshansky. An efficient algorithm for statistical minimization of total power under timing yield constraints. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 309–314, Anaheim, CA, USA, 2005.

[51] D. F. Morrison. *Multivariate Statistical Methods*. McGraw-Hill, New York, NY, USA, 1976.

[52] S. Mukhopadhyay and K. Roy. Modeling and estimation of total leakage current in nano-scaled CMOS devices considering the effect of parameter variation. In *International Symposium on Low Power Electronics and Design*, pages 172–175, Seoul, Korea, August 2003.

[53] S. Naidu. Timing yield calculation using an impulse-train approach. In *Proceedings of the 15th International Conference on VLSI Design*, pages 219–224, Bangalore, India, January 2002.

[54] F. N. Najm. A survey of power estimation techniques in VLSI circuits. *IEEE Transactions on Very Large Scale Integration Systems*, 2(4):446–455, December 1994.

[55] S. Narendra, V. De, S. Borkar, D. Antoniadis, and A. Chandrakasan. Full-chip sub-threshold leakage power prediction model for sub-0.18μm CMOS. In *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 19–23, Monterey, California, USA, August 2002.

[56] S. R. Nassif. Design for variability in DSM technologies. In *Proceedings of the IEEE International Symposium on Quality of Electronic Design*, pages 451–454, San Jose, California, USA, March 2000.

[57] K. Okada and H. Onodera. Statistical modeling of device characteristics with systematic fluctuation. In *Proceedings of the IEEE/ACM International Symposium on Circuits and Systems*, pages 437–440, Geneva, Switzerland, May 2000.

[58] M. Orshansky and K. Keutzer. A general probabilistic framework for worst case timing analysis. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 556–561, New Orleans, Louisiana, USA, June 2002.

[59] M. Orshansky, L. Milor, P. Chen, K. Keutzer, and C. Hu. Impact of spatial intrachip gate length variability on the performance of high-speed digital circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(5):544–553, May 2002.

[60] M. Orshansky, L. Milor, L. Nguyen, G. Hill, Y. Peng, and C. Hu. Intra-field gate CD variability and its impact on circuit performance. In *Technical Digest of International Electronic Devices Metting*, pages 479–482, Hong Kong, China, December 1999.

[61] A. Papoulis and S. U. Pillai. *Probability, random variables, and stochastic processes*. McGraw-Hill, Boston, USA, 2002.

[62] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers. Matching properties of MOS transistors. *IEEE Journal of Solid-State Circuits*, 24(5), October.

[63] S. Raj, S. B. K. Vrudhula, and J. Wang. A methodology to improve timing yield in the presence of process variations. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 448 – 453, San Diego, California, USA, June 2004.

[64] R. Rao, A. Devgan, D. Blaauw, and D. Sylvester. Parametric yield estimation considering leakage variability. In *Proceedings of Design Automation Conference*, pages 442 – 447, San Diego, California, USA, June 2004.

[65] R. Rao, A. Srivastava, D. Blaauw, and D. Sylvester. Statistical estimation of leakage current considering inter- and intra-die process variation. In *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 84–89, Seoul, Korea, August 2003.

[66] C. P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer, New York, NY, USA, 1999.

[67] S. S. Sapatnekar. *Timing*. Kluwer Academic Publishers, Boston, MA, 2004.

[68] A. K. Singh, M. Mani, and M. Orshansky. Statistical technology mapping for parametric yield. In *Proceedings of the IEEE/ACM International Conference on Computer-aided Design*, pages 511–518, San Jose, CA, USA, 2005.

[69] S. Sirichotiyakul, T. Edwards, C. Oh, J. Zuo, A. Dharchoudhury, R. Panda, and D. Blaauw. Stand-by power minimization through simultaneous threshold voltage selection and circuit sizing. In *Proceedings of the IEEE/ACM Design*

Automation Conference, pages 436–441, New Orleans, Louisiana, USA, June 1999.

[70] A. Srivastava, R. Bai, D. Blaauw, and D. Sylvester. Modeling and analysis of leakage power considering within-die process variations. In *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 64–67, Monterey, California, USA, August 2002.

[71] A. Srivastava, S. Shah, K. Agarwal, D. Sylvester, D. Blaauw, and S. W. Director. Accurate and efficient gate-level parametric yield estimation considering correlated variations in leakage power and performance. In *Proceedings of Design Automation Conference*, pages 535 – 540, Anaheim, California, USA, June 2005.

[72] A. Srivastava, D. Sylvester, and D. Blaauw. Statistical optimization of leakage power considering process variations using dual-Vth and sizing. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 773 – 778, San Diego, California, USA, June 2004.

[73] B. E. Stine, D. S. Boning, and J. E. Chung. Analysis and decomposition of spatial variation in integrated circuit processes and devices. *IEEE Transaction on Semiconductor Manufacturing*, 10(1):24–41, February 1997.

[74] A. Sultania, D. Sylvester, and S. S. Sapatnekar. Tradeoffs between gate oxide leakage and delay for dual Tox circuits. In *Proceedings of Design Automation Conference*, pages 761 – 766, San Diego, California, USA, June 2004.

[75] Y. Taur and T. H. Ning. *Fundamentals of Modern VLSI Devices*. Cambridge Iniverity Press, 1998.

[76] Berkeley predictive technology model (BPTM). Available at: `http://www-device.eecs.berkeley.edu/~ptm`.

[77] Capo: A large-scale fixed-die placer from UCLA. Available at: `http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/Placement`.

[78] S. Tsukiyama, M. Tanaka, and M. Fukui. A statistical static timing analysis considering correlations between delays. In *Proceedings of the Asia and South Pacific Design Automation Conference*, pages 353–358, Yokohama, Japan, January 2001.

[79] C. Visweswariah. Death, taxes and failing chips. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 343 – 347, Anaheim, California, USA, June 2003.

[80] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan. First-order incremental block-based statistical timing analysis. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 331–336, San Diego, California, USA, June 2004.

[81] S. Zanella, A. Nardi, A. Neviani, M. Quarantelli, S. Saxena, and C. Guardiani. Analysis of the impact of process variations on clock skew. *IEEE Transactions on Semiconductor Manufacturing*, 13(4):401 – 407, November 2000.

[82] Y. Zhan, A. J. Strojwas, X. Li, L. T. Pileggi, D. Newmark, and M. Sharma. Correlation-aware statistical timing analysis with non-gaussian delay distributions. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 77–82, Anaheim, California, USA, June 2005.

[83] L. Zhang, W. Chen, Y. Hu, J. A. Gubner, and C. C. Chen. Correlation-preserved non-gaussian statistical timing analysis with quadratic timing model.

In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 83–88, Anaheim, California, USA, June 2005.

[84] P. S. Zuchowski, P. A. Habitz, J. D. Hayes, and J. H. Oppold. Process and environmental variations impacts on ASIC timing. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 336–342, San Jose, California, USA, November 2004.