The Dissertation Committee for Wei-Shen Wang Certifies that this is the approved version of the following dissertation:

Models and Algorithms for Statistical Timing and Power Analysis of Digital Integrated Circuits

Committee:

Michael Orshansky, Supervisor

Nur Touba

David Pan

Sanjay Shakkottai

Frank Liu

Models and Algorithms for Statistical Timing and Power Analysis of Digital Integrated Circuits

by

Wei-Shen Wang, B.S.; M.S.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May 2007

To my parents

Acknowledgements

I am deeply grateful to my advisor, Prof. Michael Orshansky, for his guidance and support throughout these years. I took two courses about CAD/VLSI from Prof. Orshansky in the second year of the Ph.D. program. After that, I joined his research group, and started working on the dissertation. Prof. Orshansky has taught me almost everything about conducting research, and he has always been helpful when I face difficulties. Without his assistance and encouragement, I could not finish this dissertation.

I would like to express my gratitude to the members of my dissertation committee, each of whom has given me guidance and important feedback. They are Prof. Nur Touba, Prof. David Pan, Prof. Sanjay Shakkottai, and Dr. Frank Liu. Their suggestions make this dissertation better. Besides, I took several courses from Prof. Touba, from which I acquired knowledge about VLSI testing and reliable computing. I benefited from Prof. Shakkottai's probability class which played an important role in my research work. I also thank Prof. Pan's assistance during my PhD study, and appreciate Dr. Liu's feedback on my dissertation, and advice on job hunting and career development.

There are several fellow students in the Robust IC Design Laboratory whom I would like to thank for their help. They are Bin Zhang, Murari Mani, Ashish Singh, and Shayak Banerjee. I have benefited greatly from the discussion with them.

Finally, I am grateful to my family for their continuous support and encouragement. They have always been there when I need help. Without them I could not go abroad to pursue the degree at UT. Last but not least, I appreciate my wife, Ming-Chuan, for her love and support during these years.

Models and Algorithms for Statistical Timing and Power Analysis of Digital Integrated Circuits

Publication No._____

Wei-Shen Wang, Ph.D.

The University of Texas at Austin, 2007

Supervisor: Michael Orshansky

The increased variability of process and environmental parameters is having a significant impact on timing and power performance metrics of digital integrated circuits. Traditionally formulated deterministic timing and power analysis algorithms based on worst-case values of parameters often lead to over-pessimistic predictions, and may miss actual worst-case performance corners. As a result, there is an increasing need for statistical algorithms that can take into account the probabilistic nature of parameters. The practical applications of statistical approaches, however, are restricted by the limited availability of parameter distributions, and the idealized modeling of parameters adopted in the statistical frameworks. In some cases, only partial probabilistic descriptions of parameters are available, such as the mean and variance. Thus, designers are in an urgent need for statistical approaches that can handle partially-specified uncertainty.

The objective of this dissertation is to provide robust and accurate timing and power estimates for designers to assess the impact of variability on circuit performance. This dissertation proposes a set of statistical analysis algorithms to estimate circuit timing

and leakage power dissipation based on robust probabilistic approaches and rigorous mathematical modeling of parameter uncertainty. Full and partial probabilistic descriptions of parameters can be incorporated into the developed statistical frameworks. Specifically, the proposed approaches include: 1) a path-based statistical timing analysis algorithm handling path delay correlations; 2) a statistical timing analysis algorithm based on partial probabilistic descriptions of parameters; 3) analytical techniques for assessing the impact of threshold voltage variation on leakage power of dual-threshold voltage designs, and selecting optimal values of the threshold voltages for leakage power reduction; and 4) a robust estimation algorithm for parametric yield and leakage dissipation based on realistic descriptions of parameter uncertainty. The developed algorithms along with the new modeling strategy effectively improve the over-conservatism of the corner-based deterministic algorithms, and also permit assessing the impact of variability on circuit performance in the early design phase, which facilitates fast power and timing verifications in the design process. As the magnitude of variability continues to increase, the developed statistical algorithms and modeling strategy will become increasingly important for the future technology generations.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Technology scaling has been employed by the semiconductor industry for decades to cope with the market-driven demand for the improvement of the integration level (i.e., the number of transistors), manufacturing cost, speed, power, compactness, and functionality [1]. As transistor geometries continue to shrink, however, the increased variability of process and environmental parameters seriously impacts the design and optimization of integrated circuits, which has become a major obstacle in view of scaling [2]. The variability of process parameters, especially the transistor threshold voltage, effective channel length, and oxide thickness, is caused by the fundamental atom-level randomness, and systematic effects of semiconductor fabrication, such as lens aberration, optical proximity effects, and chemical mechanical planarization (CMP). In the recent technology nodes, the variability of these key process parameters causes a large spread for manufactured chips in power dissipation and circuit timing [3]-[5]. Similarly, environmental parameters (e.g., temperature and power supply voltage) may vary widely across the chip due to operating conditions, which also affects timing and power metrics. The International Technology Roadmap for Semiconductors (ITRS) in 2006 indicates that the variability of the parameters could be over 30% of the nominal values, resulting in 40% variation in circuit timing and 50% variation in power dissipation for the current generation (65nm technology) [1]. Besides, the variability of parameters will continue to increase according to the current trend, as shown in Figure 1.1. Due to the severe impact of variability on circuit performance, there is an increasing need for analysis and optimization algorithms that can handle variability of parameters.

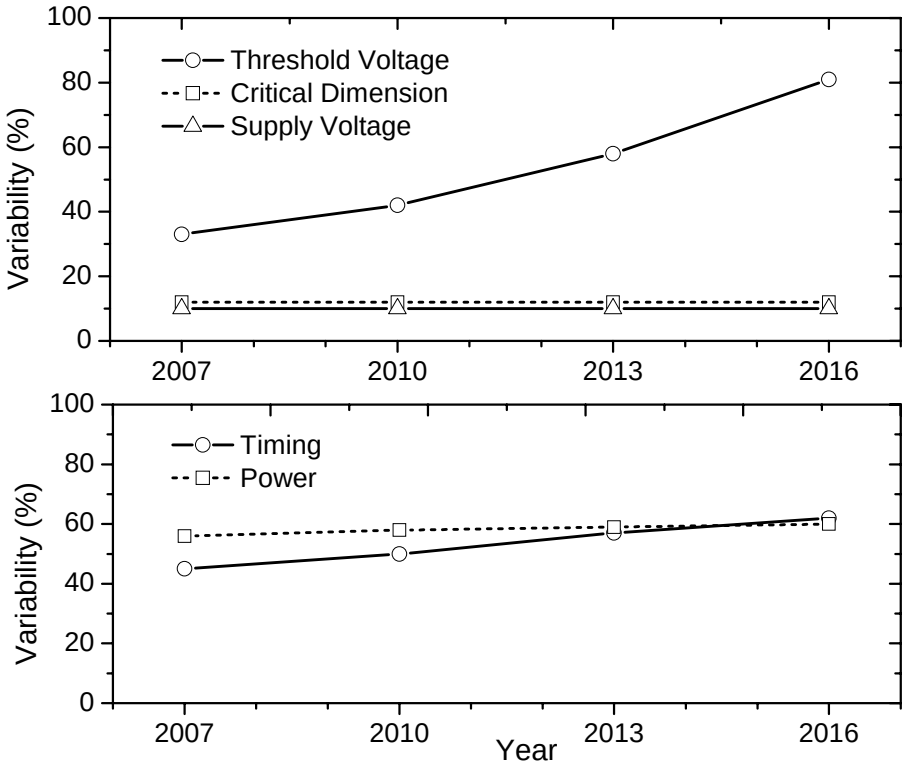

Figure 1.1: Predicted variability of key parameters, circuit timing, and power consumption (Data source: ITRS [1]).

The variability of parameters can be decomposed into two components: die-to-die and within-die components of variation [6]. The within-die, or intra-die component, describes the variation of parameters between distinct devices located on the same die. In contrast, the die-to-die component is due to the wafer-to-wafer, lot-to-lot, and chip-to-chip differences in the semiconductor manufacturing process. Traditionally the die-to-die component has been regarded as the dominant source of process variability; therefore, by neglecting the within-die component, all devices in the circuit are assumed to be equally affected by the die-to-die component of variation [7]. Thus, the feasible range of performance metrics can be determined by assuming all process parameters are

simultaneously at their best- or worst-case values. As for environmental parameters, of which variability is primarily ascribed to the within-die components of variation, intervals of parameters are often used as conservative and convenient estimates. As a result, deterministic algorithms based on Process-Voltage-Temperature (P-V-T) corners have been used for analysis and optimization for VLSI circuits.

For the nanometer-scale VLSI circuits, traditionally formulated deterministic algorithms have encountered a major challenge: using best-/worst-case analysis often leads to over-optimistic or over-pessimistic results because of several reasons [8]. First, the estimated span of performance metrics grows drastically as variability continues to increase. Second, performance corners are determined based on best-/worst-case values of parameters. In practice, however, parameters are not completely correlated due to the growing magnitude and number of uncorrelated within-die sources of variation [9]. Thus, the best-/worst-case performance corners are very unlikely to occur. Furthermore, deterministic timing analysis algorithms may miss some conditions that lead to timing failures because of complex dependencies between timing and sets of parameters. For example, gate delays may exhibit abnormal dependence on temperature at low supply voltages, which is called the inverted temperature dependence [10]. These significant reasons emphasize the importance of adopting a probabilistic framework for circuit performance analysis: a probabilistic framework can rigorously model the probabilistic nature of process and environmental parameters, to avoid the pessimism and ensure the robustness of the design.

Chip designs are mostly driven by the need to maximize the operating frequency that is constrained by circuit delay. For recent technology generations, however, the standby, or leakage, power has also become an important constraint for chips [2], [3]. Because leakage power is extremely sensitive to the variability of process and

environmental parameters, accurate and reliable estimation of leakage power consumption becomes difficult. Without accurate statistical power estimates in the design phase, a large fraction of the fabricated chips may exceed the allowable power limit, thus resulting in yield loss.

Taking the aforementioned concerns into account, the objective of this dissertation is to propose statistical analysis approaches that allow designers to assess the impact of parameter variability on circuit performance. This dissertation aims to develop robust and fast statistical analysis algorithms for circuit timing and power estimation. Meanwhile, the dissertation proposes a new modeling strategy for describing partially-specified uncertainty, and incorporates the proposed strategy into the developed statistical frameworks to handle distinct categories of probabilistic descriptions. The developed statistical algorithms along with the proposed modeling strategy are capable of reducing the over-conservatism of the traditional corner-based approaches. In the beginning of the dissertation we briefly describe the motivations behind statistical timing and power analysis problems, and outline the proposed solution.

1.1 STATISTICAL STATIC TIMING ANALYSIS: MOTIVATIONS AND CHALLENGES

Timing verification seeks to ensure that a chip design meets the given timing specification, i.e., the circuit delay is within a specified range. Traditionally, the event-driven simulation is used for timing verification: a set of input vectors (combinations of 0's and 1's) are applied to the circuit to check whether any input vector causes timing violations [11]. This dynamic approach, however, becomes impractical because of the growing circuit complexity and the large number of input vectors that need to be exercised. Therefore, static timing analysis (STA), which provides a vector-free circuit timing estimate based on the pre-characterized delays of library cells, has emerged as an indispensable technique for full-chip timing verification.

Static timing analysis can be formulated as a problem of computing the task completion time of a PERT network (directed acyclic graph) [12], where the nodes and edges, representing gates and wires in the circuit, are assigned values of delays. The PERT problem can be efficiently solved by algorithms with linear time complexity in the number of edges and nodes [11]. Although static timing analysis may provide pessimistic or optimistic delay estimates due to the conservative nature of the vector-free cell delay estimates, it is extremely efficient compared to the event-driven simulation. Thus, nowadays designers heavily rely on static timing analysis tools for fast timing verification, and also for guidance during timing-power optimization.

With the increased variability, worst-case static timing analysis may result in over-conservative estimates of circuit delay [13], [14]. Although conservative designs guarantee that chips are working at the required frequency, designs may be over-constrained resulting in higher area/power [11]. Statistical static timing analysis (SSTA), which treats delays as random variables, has been proposed to accurately estimate the distribution of circuit delay. To assess the impact of parameters variability on timing, the gate and wire delays are modeled as functions of parameters in SSTA algorithms. For example, the gate delay can be described as:

$$d_i = f_i\left(L_i, V_{th,i}\right) \tag{1.1}$$

where L_i and $V_{th,i}$ represent the effective channel length and threshold voltage of the gate, respectively. The dependency of delay on parameters is often approximated as a linear function of the deviations from the nominal values [15]:

$$d_i \simeq f_i\left(L_{nom}, V_{nom}\right) + \frac{\partial f_i\left(L_{nom}, V_{nom}\right)}{\partial L_i}\Delta L_i + \frac{\partial f_i\left(L_{nom}, V_{nom}\right)}{\partial V_{th,i}}\Delta V_{th,i} \tag{1.2}$$

where $\Delta L_i = L_i - L_{nom}$, and $\Delta V_{th,i} = V_{th,i} - V_{th,nom}$. The first-order derivatives in (1.2) are called delay sensitivities. SSTA algorithms assume the distributions of parameters for

each gate, $\left(L_i, V_{th,i}\right)$, are given, and typically parameters of distinct gates are identically distributed, but may be correlated due to the spatial proximity. With the information of distributions and delay sensitivities, SSTA seeks to estimate the delay distributions of gates, paths, and finally, the circuit timing distribution.

Multiple approaches have been proposed for statistical static timing analysis, including the use of Monte-Carlo simulation [16], [17], the parameter-space integration methods [18], and the analytical approaches [19]-[23]. Monte-Carlo approaches construct the distribution by repeating traditional static timing analysis, with gate and wire delay (or parameter) values sampled from their distributions. Monte-Carlo methods are attractive because they utilize the existing infrastructure and are easily understood by designers. However, they require a large number of full-chip STA runs. Parameter-space integration methods explore the feasible parameter space in which the timing constraints are met, and compute the integral of the joint probability density function over the feasibility set. Although Monte-Carlo based and integration techniques are accurate, both approaches are computationally prohibitive, and become infeasible in the presence of a large number of independent sources and for large circuits. In contrast, the analytical techniques, based on deriving analytical expressions to compute distributions of arrival time and circuit delay, are efficient in terms of run time, but must overcome the major challenge of analytically computing the probabilistic maximum of path delays.

In computing the maximum of path delay distributions, correlations of path delays pose a great difficulty for analytical SSTA approaches. Path delay correlations can be attributed to the dependence of gate delays on the die-to-die variation, spatial correlation of the within-die variation, and path reconvergence, i.e., paths with shared gates. The complexity of the correlation structure seems to make the exact distribution of the maximum path delay to be intractable [24]. Thus, the prior analytical techniques either

approximated the true distribution [15], [22], or produced bounds [8], [24]. Computing bounds for circuit delay distribution has several advantages over the approximation approaches: it avoids the approximation error, and provides greater flexibility in modeling. Thus, this dissertation proposes a statistical timing analysis algorithm of computing bounds for circuit delay distribution. The algorithm first computes delay distributions for paths, and then computes bounds for the maximum path delay distribution. Based on the unique characteristics of the multivariate normal distribution, and the theory of stochastic majorization, the developed algorithm can efficiently construct tight bounds for circuit delay. It also takes into account path delay correlations resulting from path reconvergence, which cannot be handled by most analytical SSTA algorithms. Chapter 2 will describe the details of the algorithm.

1.2 STATISTICAL ANALYSIS BASED ON LIMITED PROBABILISTIC DESCRIPTIONS OF PARAMETERS

Existing statistical analysis algorithms are based on the assumption that the complete probabilistic descriptions (i.e., the cumulative distribution functions) of parameters are available. Several fundamental features of a real-life design process make this assumption unreliable, or invalid. First of all, process characterization is often incomplete due to a limited number of measurements. As a result, there may be a large uncertainty in the statistical properties of process parameters. Secondly, an acute problem is encountered by fab-less design houses targeting their design to be processed by multiple foundries. Since each foundry has its own process technology and characteristics, there is large diversity in the statistical properties due to the multiple populations of process parameters. Finally, it is computationally expensive to fully characterize the distributions of some environmental parameters, such as the on-chip temperature and power supply voltage. For example, the characterization of voltage

fluctuation requires running a large number of input vectors, thus full characterization becomes almost impossible due to the increasing circuit complexity. As a result, the full probabilistic descriptions of parameters may not be available. The interval analysis [25] (worst and best corners) can be used in such cases but it may be quite conservative.

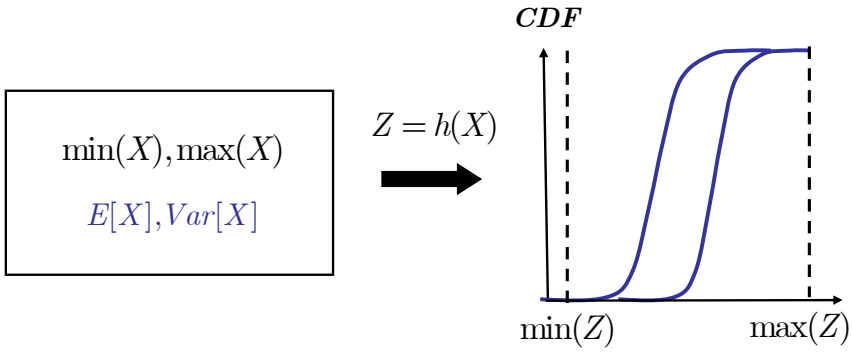

Figure 1.2: The knowledge of the mean, variance, and interval of a variable X permits computing bounds for the distribution of the function, $h(X)$.

In some cases, partial probabilistic descriptions (e.g., the moments of parameters) may be available for process and environmental parameters. Most statistical analysis algorithms, however, cannot utilize this kind of information to predict circuit performance. Thus, this dissertation proposes a new modeling strategy for parameters: parameter uncertainty is described by its mean, variance, and interval. A set of statistical techniques are then developed to handle these partial probabilistic descriptions of process and environmental parameters. Based on a sophisticated generalization of one-sided Chebyshev inequality [26] and the robust Monte Carlo sampling technique [27], the proposed techniques can produce bounds for the distributions of a certain class of functions given the statistical metrics (i.e., the mean and variance), and the interval of the uncertainty (Figure 1.2). A timing analysis algorithm that implements the statistical techniques is developed for computing bounds for path delay and circuit timing; the developed algorithm is also compatible with the existing SSTA algorithms, with the

capacity to handle Gaussian variables and linear delay models. Compared to the traditional interval-based approach, the developed algorithm enables us to reduce the over-conservatism of the timing estimates. Chapter 3 will present the algorithm.

1.3 Analysis of Leakage Power Dissipation of Dual Threshold Voltage Designs

Scaling of device geometries entails the reduction of the transistor threshold voltage to maintain sufficient drain-to-source driving currents; however, it also causes the rapid increase in the subthreshold leakage power dissipation, which is now comparable to the active power, as shown in Figure 1.3 [28]. With the continuous scaling of the threshold voltage, the subthreshold leakage current (I_{sub}) is expected to rise in the future technology nodes because of the exponential dependence on the threshold voltage (V_{th}):

$$I_{sub} \propto \exp(-qV_{th} / mkT) \tag{1.3}$$

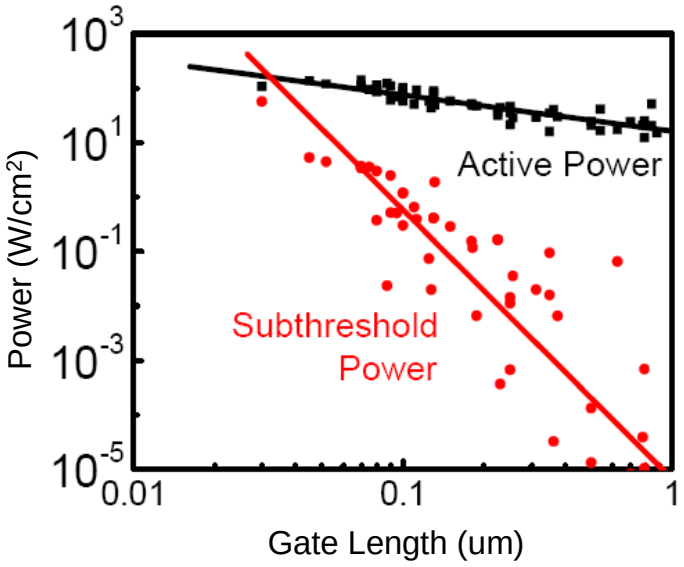

Figure 1.3: Subthreshold leakage power and active power (Source: Ref. [2]).

where q is the electron charge, k is the Boltzmann constant, T is the temperature, and m typically ranges from 1.1 to 1.4 [29]. Thus, the minimization of leakage power becomes the primary concern of the current CMOS designs, and is part of the general need to contain the increase in the overall circuit power consumption.

One of the feasible techniques for suppressing subthreshold leakage current is to assign higher threshold voltages (V_{th}) to gates, but the leakage is reduced at the expense of the increased gate delay, according to the alpha power law [30]:

$$d \propto \frac{1}{\left(V_{dd} - V_{th}\right)^{\alpha}} \tag{1.4}$$

where d is the gate delay, V_{dd} is the power supply voltage, and α is an empirical fitting coefficient about 1.3 [31]. Thus, only the gates with timing slacks (on fast paths) are assigned to high threshold voltages in consideration of the overall circuit timing. Currently dual or multiple threshold voltage designs are widely used to mitigate leakage [32]-[34]. However, in order to provide designers with cell libraries of distinct threshold voltage levels, process engineering faces an important problem: that is, how to select the best (optimal) values of the low and high threshold voltages for leakage minimization. A large separation of threshold voltages substantially reduces leakage current, but incurs a large increase in gate delays. In contrast, if the separation is limited, it only saves a small portion of power dissipation. Previous work has studied this problem of selecting the optimal threshold separation for the variation-free scenario [34], [35].

As transistor dimensions shrink, the magnitude of the threshold voltage variation is increasing, primarily caused by random dopant fluctuations. In the 65nm node and beyond, only tens of dopant atoms are in the transistor channel; thus, the threshold voltage becomes extremely sensitive even to a minor fluctuation of dopant concentrations, which may severely impact the effectiveness of the dual threshold voltage

techniques [36], [37]. Thus, this dissertation proposes a probabilistic analysis framework to estimate leakage power dissipation of dual-threshold voltage designs in the presence of threshold voltage variation. The analysis framework enables us to evaluate the effectiveness of the dual-threshold voltage based on a set of high-level circuit properties and process characteristics, such as the logic depth, total transistor width under distinct threshold voltages, and V_{th} characteristics. This dissertation also presents a strategy of determining the optimal threshold voltages for the dual-V_{th} designs in terms of leakage reduction.

1.4 Estimation of Leakage Power Consumption and Parametric Yield under Realistic Parameter Uncertainty

Parametric yield, which is the fraction of the manufactured chips that meet the performance requirement, is primarily limited by the chip operating frequency in the past. For the nanometer-scale VLSI circuits, however, power consumption has also become a limiting factor for parametric yield due to the tremendous growth of leakage power. The inverse correlation between circuit delay and leakage dissipation substantially affects parametric yield [4]: chips with short channel lengths are fast but leaky, while chips dissipating low leakage power are slow. Additionally, the exponential dependence of leakage on parameters may cause a large spread in leakage dissipation, e.g., 20X variations for 0.18um technology [4], which severely impacts the parametric yield. Thus, an accurate estimation of parametric yield and leakage power dissipation is of paramount importance for current and future technology generations.

Traditionally statistical algorithms for yield prediction and leakage power consumption focus on fully-specified process parameters [38], [39]. Partially-specified parameters are not incorporated in statistical algorithms due to several reasons. First, there are no appropriate representations that can universally describe partially-specified

and fully-specified variables. Second, it is difficult to perform arithmetic operations on partially-specified variables taking into account correlations of variables. As a result, in most cases only the interval information of the partially-specified parameters is used.

This dissertation aims to propose a statistical algorithm for estimating the chip-level parametric yield and leakage power under realistic probabilistic descriptions of parameters. The recently developed mathematical advances in probabilistic interval analysis have been employed in the algorithm to handle fully- and partially-specified variables in arithmetic operations. Two distinct representations of variables, p-boxes and self-validating histograms, can be used to robustly describe full or limited probabilistic descriptions [40]. Besides, robust computation of variables based on linear optimization permits handling correlated variables in arithmetic operations. Thus, the developed algorithm can robustly estimate parametric yield and leakage dissipation. Chapter 4 will describe the algorithm.

1.5 DISSERTATION ORGANIZATION

The remainder of the dissertation is organized as follows. Chapter 2 presents a path-based statistical static timing analysis algorithm based on the theory of stochastic majorization. Chapter 3 describes a statistical timing analysis algorithm that can handle the incomplete probabilistic descriptions of parameters. Analytical probabilistic bounds based on moments and intervals are applied to timing analysis, and a robust Monte-Carlo sampling technique is proposed to estimate the circuit delay distribution. Chapter 4 presents a robust estimation algorithm to compute parametric yield and leakage power consumption under realistic descriptions of parameter uncertainty. Chapter 5 describes an analytical framework for evaluating the effectiveness of the dual-threshold voltage designs. Finally, Chapter 6 concludes this dissertation.

Chapter 2: Path-Based Statistical Static Timing Analysis Handling Delay Correlations

Statistical static timing analysis has emerged to reduce the over-conservatism of static timing analysis; it seeks to compute the distribution of circuit delay given the distributions of gate delays, or parameters. The recent development of SSTA has focused on analytical approaches because of their run-time efficiency compared to Monte Carlo methods [16], [17] and parameter-space integration approaches [18], as described in Chapter 1. Analytical SSTA techniques can be classified into two major categories according to the traversing order of the probabilistic timing graph: path-based [23], [41] and node-based (block-based) approaches [15], [21], [22]. Path-based techniques compute individual path delays, and then compute the maximum path delay distribution, while node-based techniques perform the breadth-first search, compute the node arrival time from the maximum arrival time of fan-in nodes, and propagate the node arrival time to fan-out nodes. Since node-based techniques need to perform the maximum operation at each node, the approximation error resulting from the maximum operation is accumulated, and may cause a large error for the circuit delay distribution.

This chapter proposes a path-based statistical timing analysis to compute the bound for the circuit delay distribution, which can avoid the pitfalls of the node-based approaches. This algorithm can handle path delay correlations due to path reconvergence and dependence on die-to-die components of variation. The main contribution of this work is to compute circuit delay distribution for correlated paths using unique properties of the multivariate normal distribution, and the theory of the stochastic majorization [42]. The developed algorithm is also validated by experiments on a set of combinational

benchmark circuits. The proposed algorithm is very efficient and accurate compared to the Monte Carlo methods.

This chapter is organized as follows. Section 2.1 introduces the mathematical backgrounds for probabilistic bounding, and Section 2.2 describes the computation of probabilistic bounds for circuit delay. Section 2.3 then shows the implementation details as well as experimental results. Section 2.4 develops two techniques to improve the probabilistic bounds for lowly-correlated path delays. Finally, Section 2.5 summarizes this work.

2.1 MATHEMATICAL BASIS FOR PROBABILISTIC BOUNDING

This section first presents the mathematical formulation of circuit timing analysis. Then it proposes a novel strategy of computing bounds for the cumulative probability, based on unique characteristics of the multivariate normal distribution, and the theory of the stochastic majorization.

2.1.1 Problem Formulation

The clock cycle of a chip is constrained by the maximum path delay, $\max\{D_1,...,D_N\} \le T_{clock}$, where D_i denotes the delay of the i^{th} path in the circuit. The delay of each path is a random variable, described by a probability density function. Since the path delay vector $\{D_1,...,D_N\}$ is a random vector, the value of $\max\{D_1,...,D_N\}$ is also a random variable. Then, in order to estimate the statistical properties of the timing performance of the chip, we must find the distribution of $\max\{D_1,...,D_N\}$.

The cumulative distribution function of $\max\{D_1,...,D_N\}$ is given by $F(t) = P\left(\max\{D_1,...,D_N\} \le t\right)$, or equivalently:

$$F(t) = P\left(D_1 \le t, D_2 \le t,..., D_N \le t\right) \tag{2.1}$$

where $F(t)$ is the cumulative probability function defined over the path delay probability space. The contribution of our work is to propose an approach to efficiently deriving bounds on the cumulative distribution function (*cdf*) [43] of the circuit delay for any probabilistic timing graph, which is simply a timing graph with random node delays.

In this work, we assume path delays are Gaussian; this assumption is based on that process variability can be described as normally distributed variables, and gate delays can be modeled as linear functions of process variability when the magnitude of variability is small. We then show how we derive a bound on the *cdf* of the longest path delay that propagates through the probabilistic timing graph. Specifically, we derive upper and lower bounds on

$$F(t) = P\left(\max\left\{D_1,...,D_N\right\} \le t\right). \tag{2.2}$$

We can re-write this equation as a cumulative probability:

$$P(\max\left\{D_1,...,D_N\right\} \le t) = P(\bigcap_{i=1}^{N}\left\{D_i \le t\right\}) \tag{2.3}$$

Assuming the path delay vector is a Gaussian vector, the following theorem can be proved.

Theorem 2.1: For any normal random vector with a given correlation matrix Σ:

$$P_\Sigma\left(\bigcap\left\{D_i \le t\right\}\right) = P_\Sigma\left(\bigcap\left\{Z_i \le t_i'\right\}\right) \tag{2.4}$$

where $t_i' = (t - \mu_{D_i})/\sigma_{D_i}$ and $Z_i \sim N(0,\ 1)$.

Proof:

$$\begin{aligned} & P_\Sigma\left(\bigcap\left\{D_i \le t\right\}\right) \\ &= P_\Sigma\left(\bigcap\left\{\mu_{D_i} + \sigma_{D_i} Z_i \le t\right\}\right). \\ &= P_\Sigma\left(\bigcap\left\{Z_i \le t_i'\right\}\right) \end{aligned}$$

■

This theorem expresses the sought cumulative probability in terms of the distribution of a normal random vector with an arbitrary correlation matrix. Note that the vector t' that determines the set, over which the probability content is being evaluated, is not equicoordinate, i.e., the components of the vector are not equal ($\exists i, j : t_i' \neq t_j'$). Also note that the correlation matrix Σ that characterizes the path delay vector is populated arbitrarily, i.e., it has no special structure. Both of these factors make the efficient numerical evaluation of the probability in (2.4) impossible. Since the evaluation of such cumulative probabilities needs multi-dimensional integrals, we seek ways to pre-characterize cumulative probabilities that can be used as bounds for (2.4). Thus, a set of transformations is described to enable efficient numerical evaluation in the following sections. These transformations will lead to the bounding of the probability of (2.4) by the probabilities expressed in the form of equicoordinate vectors with well-structured correlation matrices. These probabilities can then be numerically evaluated and will provide the bound for the original cumulative probability.

2.1.2 Bounding Circuit Delay Distribution by Restructuring Path Delay Correlation Matrix

The first step is to re-express the cumulative probability of (2.4) by a cumulative probability of the same delay vector with a well structured correlation matrix. This is necessary because it will allow us to later use accurate numerical methods to evaluate the cumulative probability. The following theorem can be used to do that [42]:

Theorem 2.2: (Slepian's Inequality): Let *X* be distributed according to $N(0,\Sigma)$, where Σ is a correlation matrix. Let $R = (\rho_{ij})$ and $T = (\tau_{ij})$ be two correlation matrices. If $\rho_{ij} \geq \tau_{ij}$ holds for all *i*, *j*, then

$$P_{\Sigma=R}\left[\bigcap_{i=1}^{N}\{X_i \leq a_i\}\right] \geq P_{\Sigma=T}\left[\bigcap_{i=1}^{N}\{X_i \leq a_i\}\right] \tag{2.5}$$

holds for all $a = (a_1, \ldots, a_N)^T$.

Example: Consider a vector *X* with *N*=4, and the two correlation matrices given by

$$R = \begin{bmatrix} 1 & \mathbf{0.9} & 0.9 & 0.8 \\ \mathbf{0.9} & 1 & 0.8 & 0.7 \\ 0.9 & 0.8 & 1 & 0.8 \\ 0.8 & 0.7 & 0.8 & 1 \end{bmatrix} \text{ and } T = \begin{bmatrix} 1 & \mathbf{0.8} & 0.9 & 0.8 \\ \mathbf{0.8} & 1 & 0.8 & 0.7 \\ 0.9 & 0.8 & 1 & 0.8 \\ 0.8 & 0.7 & 0.8 & 1 \end{bmatrix}.$$

The only difference in the two correlation matrices is in the value of correlation coefficient ρ_{12} (ρ_{21}). Since *T* has a lower value of that correlation coefficient, the above theorem would ensure that:

$$P_{\Sigma=R}[\bigcap_{i=1}^{4}\{X_i \le a_i\}] \ge P_{\Sigma=T}[\bigcap_{i=1}^{4}\{X_i \le a_i\}].$$

The Slepian's inequality can be applied to the correlation matrix having negative off-diagonal coefficients; however, path delays are mostly positively correlated. Therefore, we only focus on path delays with nonnegative correlations.

By induction we can show that the following corollary also holds:

Corollary 1. Using Theorem 2.2, the cumulative distribution of (2.4), can be bounded by:

$$P_{\Sigma}\left(\bigcap\{Z_i \le t_i'\}\right) \ge P_{\Sigma_{\min}}\left(\bigcap\{Z_i \le t_i'\}\right) \tag{2.6}$$

$$P_{\Sigma}\left(\bigcap\{Z_i \le t_i'\}\right) \le P_{\Sigma_{\max}}\left(\bigcap\{Z_i \le t_i'\}\right) \tag{2.7}$$

where $\Sigma_{\min}$ (and $\Sigma_{\max}$) are generated by setting all their off-diagonal elements to $\rho_{\min} = \min\left(\rho_{ij}\right)$ (and $\rho_{\max} = \max\left(\rho_{ij}\right)$) for all $i \ne j$.

Consider, for example, the probability given by the correlation matrix used in the above example. The lower bound on the probability can be then established via a nicely structured correlation matrix $\Sigma_{\min}$.

$$\Sigma = \begin{bmatrix} 1 & 0.9 & 0.9 & 0.8 \\ 0.9 & 1 & 0.8 & 0.7 \\ 0.9 & 0.8 & 1 & 0.8 \\ 0.8 & 0.7 & 0.8 & 1 \end{bmatrix} \text{ and } \Sigma_{\min} = \begin{bmatrix} 1 & 0.7 & 0.7 & 0.7 \\ 0.7 & 1 & 0.7 & 0.7 \\ 0.7 & 0.7 & 1 & 0.7 \\ 0.7 & 0.7 & 0.7 & 1 \end{bmatrix}.$$

The matrix $\Sigma_{\min}$ has diagonal coefficients equal to 1, and the identical off-diagonal coefficients. This degree of uniformity is essential for the numerical evaluation of the probability that we will use later.

2.1.3 Probabilistic Bounds Based on Stochastic Majorization

At the conclusion of the bounding operations based on the manipulations of the correlation matrices, the cumulative probabilities in (2.6) and (2.7) are now described by correlation matrices with identical off-diagonal elements. Still, they require evaluating the probability content of a multivariate normal distribution over the non-equicoordinate set, which is numerically expensive. To enable a more efficient numerical evaluation of these cumulative probabilities, we now express them in terms of the equicoordinate probability. The reason for the greater attraction of the probabilities of the equicoordinate vectors is that it is easy to pre-characterize the probability of a multi-dimensional vector over a multi-dimensional cube, which is the body with identical coordinates, as opposed to doing that for all possible shapes of the multidimensional parallelepiped.

In this section we show how we can bound the cumulative probabilities of a normal random vector by a partial ordering of the location parameter vectors, using the techniques of the theory of majorization [44]. In order to derive these bounds, however, we will need to ensure that several criteria are met, since the bounding is predicated on several properties of the probability distributions.

The notion of strong and weak majorization can be used to compare random variables and their distributions [42]. First, we introduce a set of formal definitions that

allow comparing random variables and their distributions. Most of the theorems in this section are shown without proof, which can be found in [42].

Let $a = (a_1,...,a_N)^T$ and $b = (b_1,...,b_N)^T$ be two real vectors. Let $a_{[1]} \geq ... \geq a_{[N]}$ and $b_{[1]} \geq ... \geq b_{[N]}$ be the *ordered* values of the components of *a* and *b*.

Definition: *a* is said to majorize *b*, in symbols, $a \succ b$, if $\sum_{i=1}^{N} a_i = \sum_{i=1}^{N} b_i$, and $\sum_{i=1}^{r} a_{[i]} \geq \sum_{i=1}^{r} b_{[i]}$ for $r = 1,...,N-1$.

Definition: *a* is said to weakly majorize *b*, in symbols, $a \succ\succ b$, if $\sum_{i=1}^{r} a_{[i]} \geq \sum_{i=1}^{r} b_{[i]}$ for $r = 1,...,N$.

It is easy to check that a pair of vectors $(3,2,1)^T \succ (2,2,2)^T$ is an example of strong majorization, and a pair of vectors $(3,2,1)^T \succ\succ (1,1,1)^T$ is an example of weak majorization. Additionally, an important fact of the strong majorization is that $a \succ b \Leftrightarrow -a \succ -b$, which is known as the translation invariance property.

The notions of strong and weak majorization can be extended to enable comparing random variables and their distributions for a certain set of functions. This set of functions should be Schur-convex (or Schur-concave) functions which are defined below.

Definition (Schur convexity and Schur concavity): A function ψ of *N* arguments is said to be Schur-convex (Schur-concave) if $a \succ b$ implies that $\psi(a) \geq \psi(b)$ $\left(\psi(a) \leq \psi(b)\right)$.

For a set $A \subset \Re^N$, if its indicator function is Schur-convex (Schur-concave), *A* is said to be a Schur-convex (Schur-concave) set. In addition, an increasing Schur-convex set is a Schur-convex set whose indicator function is *nondecreasing* in each argument; a decreasing Schur-concave set is a Schur-concave set whose indicator function is *nonincreasing* in each argument.

Definition (Strong Stochastic Majorization): Let *X* and *Y* be two *N*-dimensional random variables. *X* is said to stochastically majorize *Y*, in symbols $X \overset{st}{\succ} Y$, if $P[X \in A] \geq (\leq) P[Y \in A]$ for every Borel-measurable Schur-convex (Schur-concave) set *A*.

Definition (Weak Stochastic Majorization): *X* is said to weakly stochastically majorize *Y*, in symbols $X \overset{st}{\succ\succ} Y$, if $P[X \in A] \geq (\leq) P[Y \in A]$ for every Borel-measurable increasing Schur-convex (decreasing Schur-concave) set *A*.

The key idea we are exploiting in deriving bounds on the cumulative probabilities is that for certain distributions, stochastic inequalities can be established on the basis of a partial ordering of the parameter vectors using ordinary (deterministic) majorization. The class of random vectors that can be ordered via the ordering of their location parameter vectors is limited to distributions of which density functions are Schur-convex (Schur-concave). The following theorems formalize this fact [42]:

Theorem 2.3: Let the random variable X_θ have a density $f(x-\theta)$ for $x \in \Re^N, \theta \in \Re^N$ (a location parameter vector). If the density function $f(x)$ is Schur-concave in *x*, then $\xi \succ \theta$ implies that $X_\xi \overset{st}{\succ} X_\theta$.

Theorem 2.4: Let the random variable X_θ have a density $f(x-\theta)$ for $x \in \Re^N, \theta \in \Re^N$. If the density function $f(x)$ is Schur-concave in *x*, then $\xi \succ\succ \theta$ implies that $X_\xi \overset{st}{\succ\succ} X_\theta$.

Thus, if the probability density function of random vector X_θ satisfies the properties of Theorem 2.3 and Theorem 2.4, then we can find a location parameter vector ξ, and the random vector X_ξ that will stochastically majorize X_θ. This is equivalent to saying that (by the definition of stochastic majorization) the probability content of X_ξ over the appropriate set, will bound the probability content of X_θ over this set. Note that this set must satisfy two properties. First, this set must enable

computing the probability content that corresponds to our original purpose: the joint cumulative distribution function of the path delays. Second, by definition, it must be Borel-measurable and Schur-concave (or Schur-convex). For weak stochastic majorization, the set needs to be increasing Schur-convex (or decreasing Schur-concave).

The next theorem provides the final results [42].

Theorem 2.5: (1) Let X_θ have a multivariate normal distribution with the mean vector θ, and with equal variances and equal correlations. The density function $f(x-\theta)$ is Schur-concave.

(2) Let *A* denote the set $\{x \mid x_i \le a, i = 1,\ldots,N\}$. This set is Schur-concave because its indicator function is Schur-concave. Besides, since the indicator function of *A* is nonincreasing in each argument, *A* is a decreasing Schur-concave set.

Because the structured correlation matrices, $\Sigma_{\min}$ and $\Sigma_{\max}$, have identical off-diagonal elements, we know that the density functions of $N(0,\Sigma_{\min})$ and $N(0,\Sigma_{\max})$ are Schur-concave according to Theorem 2.5.

We can now point out the fact that for the set *A* defined above, $P_\theta[X \in A] = P\left(\bigcap_{i=1}^{N}(X_i + \theta_i \le a)\right)$, and putting everything together.

Theorem 2.6: If $\xi \succ \theta$, X_θ and *A* are defined as in Theorem 2.5, then

$$P_\xi[X \in A] \le P_\theta[X \in A], \text{ or}$$

$$P\left(\bigcap_{i=1}^{N}\{X_i + \xi_i \le a\}\right) \le P\left(\bigcap_{i=1}^{N}\{X_i + \theta_i \le a\}\right).$$

Proof: Because the probability density of X_θ is Schur-concave, we know that $\xi \succ \theta$ implies $X_\xi \overset{st}{\succ} X_\theta$ according to Theorem 2.3. Since the set *A* is a Schur-concave set, the definition of the strong stochastic majorization states that $X_\xi \overset{st}{\succ} X_\theta$ implies that

$$P_\xi[X \in A] \le P_\theta[X \in A]. \qquad \blacksquare$$

The similar theorem can be stated for the case of the weak majorization:

Theorem 2.7: If $\xi \succ\succ \theta$, X_θ and *A* are defined as in Theorem 2.5, then

$$P_\xi[X \in A] \le P_\theta[X \in A], \text{ or}$$

$$P\left(\bigcap_{i=1}^{N}\{X_i + \xi_i \le a\}\right) \le P\left(\bigcap_{i=1}^{N}\{X_i + \theta_i \le a\}\right).$$

Proof: Since the probability density of X_θ is Schur-concave, we know that $\xi \succ\succ \theta$ implies $X_\xi \overset{st}{\succ\succ} X_\theta$ according to Theorem 2.4. The definition of weak stochastic majorization states that $X_\xi \overset{st}{\succ\succ} X_\theta$ implies $P_\xi[X \in A] \le P_\theta[X \in A]$ when *A* is a decreasing Schur-concave set. ■

In the next section, we use the above results to compute bounds for cumulative probabilities of the path delay vector. In doing that, it will also become clear why the parallel notions of strong and weak stochastic majorization have to be developed.

Up to this point, we have established the fact that the multivariate normal distribution described here has a density function which is Schur-concave, and the set *A* defined in Theorem 2.5 is a decreasing Schur-concave set. We can now easily find an *equicoordinate* parameter vector that is strongly majorized by the vector of delay coordinates, and then use Theorem 2.6 and Theorem 2.7 to bound the cumulative probability. Using the definition of majorization, the vector of the average values is a sought vector.

Fact 1. If $\underset{\sim}{t} = (t_1,...,t_N)^T$ and $\underset{\sim}{\bar{t}} = (\bar{t},...,\bar{t})^T$, where $\bar{t} = \frac{1}{N}\sum_{i=1}^{N} t_i$, then

$$\underset{\sim}{t} \succ \underset{\sim}{\bar{t}}.$$

Proof: First the sum of the coordinates is equal, i.e., $\sum_{i=1}^{N} t_i = \sum_{i=1}^{N} \bar{t}$. Now we need to prove $\sum_{i=1}^{r} t_{[i]} \ge \sum_{i=1}^{r} \bar{t}$, for *r*=1,…, *N*-1. Consider the term $\sum_{i=1}^{r} t_{[i]} \Big/ r$, which is the average of the *r* largest values among $t_1,...,t_N$. This value must be larger than or equal to the average of all coordinate values. Therefore, we know that

$$\sum_{i=1}^{r} t_{[i]} \Big/ r \geq \bar{t}, \text{ and } \sum_{i=1}^{r} t_{[i]} \geq r\bar{t} = \sum_{i=1}^{r} \bar{t}. \qquad \blacksquare$$

Fact 2. Let Z be the random vector $(Z_1,...,Z_N)$, where $Z_i (i = 1,...,N)$ is defined as in Theorem 2.1. If $\underset{\sim}{t} = (t_1,...,t_N)^T$ and $\underset{\sim}{\bar{t}} = (\bar{t},...,\bar{t})^T$, where $\bar{t} = \frac{1}{N}\sum_{i=1}^{N} t_i$, then

$$P\left(\bigcap \{Z_i \leq t_i\}\right) \leq P\left(\bigcap \{Z_i \leq \bar{t}\}\right).$$

Proof: Consider the Schur-concave set described in Theorem 2.5, $A = \{x \mid x_i \leq a, i = 1,...,N\}$, where x is a N-dimensional vector. According to the translation invariance property, we know that

$$\underset{\sim}{t} \succ \underset{\sim}{\bar{t}} \rightarrow -\underset{\sim}{t} \succ -\underset{\sim}{\bar{t}}.$$

Therefore, from Fact 1 we can infer that $-\underset{\sim}{t} \succ -\underset{\sim}{\bar{t}}$. From Theorem 2.6, because of the partial ordering between the location parameter vectors, i.e., $-\underset{\sim}{t} \succ -\underset{\sim}{\bar{t}}$, we can obtain the probabilistic inequality:

$$P\left(\bigcap_{i=1}^{N} \{Z_i + (-t_i) \leq a\}\right) \leq P\left(\bigcap_{i=1}^{N} \{Z_i + (-\bar{t}) \leq a\}\right).$$

Let $a = 0$, then we prove that

$$P\left(\bigcap_{i=1}^{N} \{Z_i \leq t_i\}\right) \leq P\left(\bigcap_{i=1}^{N} \{Z_i \leq \bar{t}\}\right). \qquad \blacksquare$$

Since we are interested in both the upper and lower bounds on the probability distribution of the path delay vector, we also would like to find an *equicoordinate* parameter that strongly majorizes *t*. This is, however, impossible, and in this case we need to resort to weak majorization. Using Theorem 2.7, we can show that:

Fact 3. If $t_{\min} = \min\{t_1,...,t_N\}$, $\underset{\sim}{t} = (t_1,...,t_N)^T$, $\Delta t = (t_{\min} - t_1,...,t_{\min} - t_N)^T$, and $\underset{\sim}{0} = (0,...,0)^T$, then

$$\underset{\sim}{0} \succ\succ \Delta t .$$

Proof: Since $t_{\min} = \min\{t_1,...,t_N\}$, we know that $t_{\min} - t_i \le 0$, for *i*=1,…,*N*. Then we observe that $\sum_{i=1}^{r} 0 \ge \sum_{i=1}^{r} (t_{\min} - t_i)$ for $r = 1,...,N$. From the definition of the weak majorization, we know that $\underset{\sim}{0} \succ\succ \Delta t$. ■

Fact 4. Let *Z* be the random vector $(Z_1,...,Z_N)$, where $Z_i (i = 1,...,N)$ is defined as in Theorem 1. If $t_{\min} = \min\{t_1,...,t_N\}$, $\underset{\sim}{t} = (t_1,...,t_N)^T$, $\underset{\sim}{0} = (0,...,0)^T$, and $\Delta t = (t_{\min} - t_1,...,t_{\min} - t_N)^T$, then

$$P\left(\bigcap \{Z_i \le t_{\min}\}\right) \le P\left(\bigcap \{Z_i \le t_i\}\right).$$

Proof: Similar to the proof of Fact 2, we consider the decreasing Schur-concave set $A = \{x \mid x_i \le a, i = 1,...,N\}$. From Fact 3, we know that $\underset{\sim}{0} \succ\succ \Delta t$. According to Theorem 2.7 we can infer that

$$P\left(\bigcap_{i=1}^{N} \{Z_i + 0 \le a\}\right) \le P\left(\bigcap_{i=1}^{N} \{Z_i + (t_{\min} - t_i) \le a\}\right).$$

Now let $a = t_{\min}$, then

$$\begin{aligned} P\left(\bigcap_{i=1}^{N} \{Z_i \le t_{\min}\}\right) &\le P\left(\bigcap_{i=1}^{N} \{Z_i + (t_{\min} - t_i) \le t_{\min}\}\right) \\ &= P\left(\bigcap_{i=1}^{N} \{Z_i \le t_i\}\right) \end{aligned} .$$ ■

We have finally bounded the original cumulative probability by cumulative probabilities expressed in terms of an equicoordinate vector, a correlation matrix with identical off-diagonal elements, and the standard multivariate normal vector:

$$P_{\Sigma_{\min}}(\bigcap \{Z_i \le t_{\min}\}) \le P_{\Sigma}(\bigcap Z_i \le t_i) \tag{2.8}$$

$$P_{\Sigma}(\bigcap Z_i \le t_i) \le P_{\Sigma_{\max}}(\bigcap \{Z_i \le \overline{t}\}) \tag{2.9}$$

This is a well-structured object whose probability content is amenable to numerical evaluation.

2.1.4 Numerical Evaluation of the Cumulative Probability

The bounds can be numerically evaluated as long as the correlation matrix and the coordinate vectors are fixed for a random vector of given dimensionality. In order to speed up the run-time evaluation of the probabilities, the numerical evaluation is done using pre-generated lookup tables for a range of vector dimensionalities, coordinates, and correlation coefficients. The table generation, essentially, boils down to numerically evaluating the probability integral, which can be easily done by the Monte-Carlo integration of the cumulative probability of a multivariate normal distribution for the range of: (1) vector dimensionalities (e.g., the number of paths, *N*); (2) the off-diagonal correlation coefficients. Below is the algorithm for evaluating the cumulative probability.

Equicoordinate CDF Computation of Multivariate Normal Distribution

Input: a vector cardinality *N*, a correlation matrix Σ with identical off-diagonal coefficients, and a coordinate *t*.

Output: the cumulative probability $P_{\Sigma}(\bigcap_{i=1}^{N}\{Z_i \le t\})$.

1) Generate *S* sample vectors that follow the multivariate normal distribution $N(0,\Sigma)$.

2) For each sample vector $v = (v_1,...,v_N)$, compute the maximum of the elements:

$$v_{\max} = \max\{v_i\}.$$

3) Count the number of sample vectors S_t subject to the condition that $v_{\max} \le t$. Then,

$$P_{\Sigma}(\bigcap\{Z_i \le t\}) \simeq S_t / S.$$

For a specific number of paths *N*, the table size is $(L+1)M$, assuming the correlation coefficient ρ_{ij} increases from 0 to 1 at the interval of $1/L$, and *M* is the

number of points in the *t* domain. In reality, we do not need too many distinct values of *N*, which helps reduce the table size. Besides, interpolation is used to find the cumulative probability values to reduce data storage. Our pre-generated tables contain cumulative distribution functions for the off-diagonal correlation coefficients ranging from 0 to 1, at the interval of 0.01. The coordinates include values within mean +/- 3 standard deviations. For any coordinate values not in the table, linear interpolation is used to approximate the cumulative probabilities. This strategy can save storage space and reduce the execution time for loading tables.

All the theoretical derivations advanced above were in terms of normal distributions. Therefore, the theory of probabilistic bounds in this work enables computing bounds for circuit delay when path delays follow normal distributions. The developed bounding algorithm can be extended to handle path delays following *truncated* normal distributions. Such an extension is useful because it reflects the reality of the semiconductor manufacturing as a controlled stochastic process. The presence of the human controller prevents the semiconductor process from exhibiting extreme deviations from the mean. For example, a deviation that exceeds 3σ from the mean is not just unlikely (as in the case of normal distribution), but is in fact impossible. The normal model is widely used in the SSTA literature, and more generally, throughout engineering practice, and in most of the cases, is sufficient. Still, under some conditions being able to handle truncated normal distributions directly is important.

The extension to handle truncated normal distributions is treated heuristically in this work. The derived probabilistic inequalities of Section 2.1.3 are used to bound the distribution of a path delay vector. However, when performing numerical evaluation a *table of cumulative probability of truncated normal is used*. In consideration of this, Step 1 of the above algorithm is modified. Our experiments show that this technique is valid

for the range of off-diagonal correlation coefficients and vector dimensionalities taken into account in this work: the empirical distributions of Monte Carlo samples are always bounded by the bound produced by the algorithm.

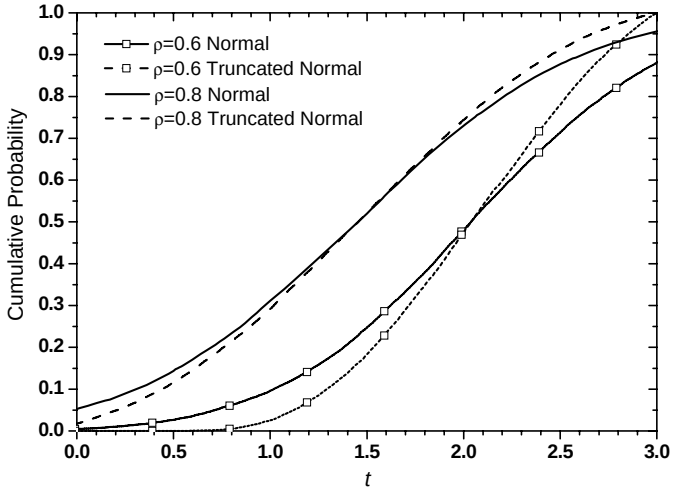

Figure 2.1: Equicoordinate cumulative probabilities of multivariate normal and truncated normal distributions with zero mean and unity variance (N=1000).

Theoretically, the difference between the equicoordinate cumulative probabilities of the normal and truncated normal distributions becomes pronounced for low correlation values, and high dimensionality, i.e., large N. Figure 2.1 illustrates this fact, showing the cumulative probabilities of multivariate normal and truncated normal distributions with zero mean, unity variance, and identical correlation matrices. The off-diagonal correlation coefficients are set to 0.6 and 0.8. Besides, truncation occurs when t=-3 and 3. As illustrated in Figure 2.1, lower correlation of variables results in a larger difference in the cumulative probabilities at the high and low percentiles. Therefore, we may need to take into account the truncated nature of the real-life process variability when dealing with low-correlation multivariate distributions. An experiment on the benchmark circuit is further analyzed in Section 2.3.

The equicoordinate cumulative probability of the truncated normal distribution is higher than that of the normal distribution at the high percentile. That means the lower bound of *cdf* at the high percentile can be improved if we use the truncated normal distribution, while it may also result in a worse upper bound of the cumulative probability. Since the lower bound of the cumulative probability represents the upper bound of circuit delay at a confidence level, using the truncated normal distribution enables us to accurately estimate the upper bound of circuit delay.

2.2 ALGORITHM FOR COMPUTING CIRCUIT DELAY DISTRIBUTION

2.2.1 Computation of Path Delay Vector Covariance Matrices

The exposition above assumes the path-to-path correlation matrix is given. In this section, we derive a set of results that allow computing the correlation matrix and the vector of variances. The linear additive statistical model is a natural result of ascribing the total node delay variation to two components: die-to-die and within-die components of variation. Conceptually, the entire delay variability can be decomposed as:

$$\Delta d_i = \Delta d_{dd} + \Delta d_{i,wd} \tag{2.10}$$

where the first component is due to the die-to-die variation in the processing conditions. The second results from local variations on the same die. Note that the environmental variations can be also captured by the within-die source of variation.

The two-term statistical delay model above does not have enough expressiveness, and also lacks an explicit link to the fundamental causes of variability in the process domain. For this reason, an extended version of such a model has become popular in literature, and is known as the canonical delay model [15]. In this model, delay variability is a weighted sum of the die-to-die components of variation, and an independent random variable which solely represents the uncorrelated within-die components of variation.

Similarly, in our statistical gate delay model, delay variability is represented as a weighted sum of two groups of independent random variables:

$$\Delta d_i = \sum_{j=1}^{n_X} a_{i,j} \left(\Delta x_{j,dd} + \Delta x_{i,j,wd} \right) \tag{2.11}$$

where Δd_i is the overall *delay variation* of gate *i*, $\Delta x_{i,j,wd}$ is a random variable representing the j^{th} within-die component of variation, and $\Delta x_{j,dd}$ denotes the j^{th} die-to-die source of variation. Note that the die-to-die components are global to all gates in the circuit. Uncorrelated local components of variation are not lumped to account for the dependence on multiple within-die components of variation. The within-die and die-to-die components of variation are modeled as independent normal random variables with $E[\Delta x_{j,dd}] = E[\Delta x_{i,j,wd}] = 0$, $Var\{\Delta x_{j,dd}\} = \sigma^2_{j,dd}$, and $Var\{\Delta x_{i,j,wd}\} = \sigma^2_{i,j,wd}$. The coefficients $a_{i,j}$ are interpreted as delay sensitivity values with respect to the variability of the process parameters, which can be obtained by library characterization.

Additionally, this statistical delay model can be expressed concisely by resorting to the matrix expression.

$$\Delta d_i = A_i^T X_{dd} + A_i^T X_{i,wd} \tag{2.12}$$

where the delay sensitivity matrix $A_i = (a_{i,1},...,a_{i,n_X})^T$, $X_{dd} = (\Delta x_{1,dd},...,\Delta x_{n_X,dd})^T$, and $X_{i,wd} = (\Delta x_{i,1,wd},...,\Delta x_{i,n_X,wd})^T$. Again, the die-to-die components, described by X_{dd}, are global to all nodes.

Then, the delay sensitivity matrix A_i and the variance of parameters determine the variance of the gate delay deviation, Δd_i.

$$\begin{aligned} & Var\{\Delta d_i\} \\ &= A_i^T \Sigma_{dd} A_i + A_i^T \Sigma_{i,wd} A_i \\ &= \sum_{j=1}^{n_X} {a_{i,j}}^2 \left(\sigma^2_{j,dd} + \sigma^2_{i,j,wd} \right) \end{aligned} \tag{2.13}$$

where Σ_{dd} and $\Sigma_{i,wd}$ are the covariance matrices of X_{dd} and $X_{i,wd}$, respectively. Distinct kinds of parameters are assumed to be mutually independent. Thus, Σ_{dd} and $\Sigma_{i,wd}$ are actually diagonal matrices, and the matrix multiplications in (2.13) can be done efficiently. More importantly, the above formulation enables us to parsimoniously represent the gate-to-gate delay correlations. It is easy to show by writing out the covariance expression term by term that:

$$\begin{aligned} & corr(\Delta d_i, \Delta d_k) \\ &= Cov(\Delta d_i, \Delta d_k) \Big/ \sqrt{Var\{\Delta d_i\}} \sqrt{Var\{\Delta d_k\}} \\ &= A_i^T \Sigma_{dd} A_k \Big/ \sqrt{A_i^T \Sigma_{dd} A_i + A_i^T \Sigma_{i,wd} A_i} \sqrt{A_k^T \Sigma_{dd} A_k + A_k^T \Sigma_{k,wd} A_k} \end{aligned} \tag{2.14}$$

To compute the path-to-path correlation, we need to compute the variance of the path delay, and the covariance of the delay variation between paths. The delay variance of path P_1 is:

$$\begin{aligned} Var\left\{\sum_{i=1}^{n_{P1}} \Delta d_i\right\} &= Var\left\{\sum_{i=1}^{n_{P1}} \left(A_i^T X_{dd} + A_i^T X_{i,wd}\right)\right\} \\ &= Var\left\{\left(\sum_{i=1}^{n_{P1}} A_i^T\right) X_{dd} + \sum_{i=1}^{n_{P1}} \left(A_i^T X_{i,wd}\right)\right\} \\ &= \left(\sum_{i=1}^{n_{P1}} A_i\right)^T \Sigma_{dd} \left(\sum_{i=1}^{n_{P1}} A_i\right) + \sum_{i=1}^{n_{P1}} \left(A_i^T \Sigma_{i,wd} A_i\right) \end{aligned} \tag{2.15}$$

where n_{P1} is the number of gates on path P_1, and $\sum_{i=1}^{n_{P1}} A_i$ represents the sum of A matrices for gates on path P_1.

However, the correlation of path delays not only results from the dependence on the die-to-die components of variation, but also depends on the structure of the paths. Overlapping paths are correlated due to the common gates of the paths; therefore, the common gates of the paths need to be taken into account when computing the path-to-

path delay correlation. Then the covariance of delays for two paths P_1 and P_2 can be written as:

$$\begin{aligned}
&Cov\left(\sum_{i=1}^{n_{P1}}\Delta d_i,\sum_{k=1}^{n_{P2}}\Delta d_k\right)\\
&=Cov\left(\sum_{i=1}^{n_{P1}}\left(A_iX_{dd}+A_iX_{i,wd}\right),\sum_{k=1}^{n_{P2}}\left(A_kX_{dd}+A_kX_{k,wd}\right)\right)\\
&=Cov\left(\sum_{i=1}^{n_{P1}}A_iX_{dd},\sum_{k=1}^{n_{P2}}A_kX_{dd}\right)+Cov\left(\sum_{i=1}^{n_{P1}}A_iX_{i,wd},\sum_{k=1}^{n_{P2}}A_kX_{k,wd}\right)\\
&+Cov\left(\sum_{i=1}^{n_{P1}}A_iX_{i,wd},\sum_{k=1}^{n_{P2}}A_kX_{dd}\right)+Cov\left(\sum_{i=1}^{n_{P1}}A_iX_{dd},\sum_{k=1}^{n_{P2}}A_kX_{k,wd}\right)
\end{aligned} \tag{2.16}$$

where n_{P1} and n_{P2} denote the numbers of gates on path P_1 and P_2, respectively.

Since the within-die and die-to-die components are assumed to be mutually independent, the delay variations due to the within-die and the die-to-die components are not correlated. Thus, when we compute the covariance of path delays, we only need to consider these terms: $Cov\left(\sum_{i=1}^{n_{P1}}A_iX_{dd},\sum_{k=1}^{n_{P2}}A_kX_{dd}\right)$ and $Cov\left(\sum_{i=1}^{n_{P1}}A_iX_{i,wd},\sum_{k=1}^{n_{P2}}A_kX_{k,wd}\right)$. Besides, for the within-die components, we only need to take into account the delay due to common gates because the within-die components of distinct gates are not correlated. Thus, from (2.16) we obtain

$$\begin{aligned}
&Cov\left(\sum_{i=1}^{n_{P1}}\Delta d_i,\sum_{k=1}^{n_{P2}}\Delta d_k\right)\\
&=Cov\left(\sum_{i=1}^{n_{P1}}A_iX_{dd},\sum_{k=1}^{n_{P2}}A_kX_{dd}\right)+Cov\left(\sum_{i=1}^{n_{P1}}A_iX_{i,wd},\sum_{k=1}^{n_{P2}}A_kX_{k,wd}\right)\\
&=\left(\sum_{i=1}^{n_{P1}}A_i\right)^T\Sigma_{dd}\left(\sum_{k=1}^{n_{P2}}A_k\right)+\sum_{c\in P_1\cap P_2}\left(A_c{}^T\Sigma_{c,wd}A_c\right)
\end{aligned} \tag{2.17}$$

where $P_1\cap P_2$ denotes the common gates for paths P_1 and P_2, A_c is the sensitivity coefficient matrix of a common gate for path P_1 and P_2, and $\Sigma_{c,wd}$ is the covariance

matrix of the within-die component of the shared gate. We can then compute the correlation of path delays:

$$
\begin{aligned}
&corr\left(\sum_{i=1}^{n_{P1}}\Delta d_i,\sum_{k=1}^{n_{P2}}\Delta d_k\right)\\
&=\frac{Cov\left(\sum_{i=1}^{n_{P1}}\Delta d_i,\sum_{k=1}^{n_{P2}}\Delta d_k\right)}{\sqrt{Var\{\sum_{i=1}^{n_{P1}}\Delta d_i\}}\sqrt{Var\{\sum_{k=1}^{n_{P2}}\Delta d_k\}}}\\
&=\frac{\left(\sum_{i=1}^{n_{P1}}A_i\right)^T\Sigma_{dd}\left(\sum_{k=1}^{n_{P2}}A_k\right)+\sum_{c\in P_1\cap P_2}\left({A_c}^T\Sigma_{c,wd}A_c\right)}{\sqrt{\left(\sum_{i=1}^{n_{P1}}A_i\right)^T\Sigma_{dd}\left(\sum_{i=1}^{n_{P1}}A_i\right)+\sum_{i=1}^{n_{P1}}\left({A_i}^T\Sigma_{i,wd}A_i\right)}\sqrt{\left(\sum_{i=1}^{n_{P2}}A_i\right)^T\Sigma_{dd}\left(\sum_{i=1}^{n_{P2}}A_i\right)+\sum_{i=1}^{n_{P2}}\left({A_i}^T\Sigma_{i,wd}A_i\right)}}
\end{aligned}
\tag{2.18}
$$

Finally, we can compute the path-to-path delay correlation by manipulating the sensitivity matrices of path delays. When traversing paths, the sum of sensitivity matrices are updated and propagated to the primary outputs.

2.2.2 Computation of Probabilistic Bounds

The statistical STA algorithm we propose is targeted towards a path-based formulation in which a deterministic STA algorithm is first used to extract a subset of critical paths under the nominal conditions. Algorithmically, given the probabilistic timing graph, we first extract a subgraph G' that contains *N* deterministically (with respect to the mean path delay value) longest paths using efficient path extraction algorithms, e.g., [45] and [46]. Based on the path enumeration algorithm in [46], our implementation can extract the *N* longest paths efficiently.

Here we briefly describe the path enumeration algorithm in [46]. Path extraction is divided into two major phases: 1) computation of maximum delay from a node to sink and ranking successor nodes, and 2) path enumeration of *k* critical paths. First, a source

and a sink node are added to the timing graph, and then a backward traversal from the sink node is performed to compute the maximum delay to the sink from each node.

Besides, for each node, successor nodes are ranked in a descending order according to a cost function, which is the sum of the successor node delay and the maximum delay of the successor node to the sink. Additionally, the slack values from choosing different successor nodes are also computed. After these preprocessing steps, the longest path can be obtained by traversing from the source node, and repeatedly picking the highest-ranked successor as the next node until reaching the sink. Other longest paths can be extracted by using the slack values. In this way, path extraction is quite efficient. From our experiment, it takes less than 1 second to extract 50 paths from a netlist with more than 3,000 nodes. Also, the scalability of this algorithm allows us to take into account more paths for very large-scaled circuits.

We now use the derived path-to-path correlation in (2.18) to compute pair-wise correlation values for the extracted deterministically longest paths. The path delay variance can be directly computed by traversing paths following (2.15). Therefore, the worst-case complexity of computing the variance of a path delay is $O(mn_X)$, where m is the maximum number of gates on the extracted paths, and n_X is the number of parameters. For pair-wise path covariance values, the common nodes of overlapping paths can be identified by pair-wise comparisons. Thus, combining the computation of path delay variance, and the correlation coefficients of N paths, the overall worst-case complexity is $O\left(N \cdot m \cdot (n_X + n_X) + N^2 m^2 n_X\right) = O(N^2 m^2 n_X)$. However, since the objective is to compute the bound for the circuit delay distribution using (2.8) and (2.9), we only need to keep track of the minimum and maximum path-to-path correlation values.

With the maximum and minimum path-to-path correlations, we can then look up the probability tables to determine the bounds for circuit delay at any specific delay values. The flow of the bounding algorithm is shown in Figure 2.2.

1) Generate the cumulative probability table for distinct vector cardinalities (e.g., *N*) and correlation coefficients
2) Extract *N* longest paths using static timing analysis based on the mean path delay values.
3) Traverse the timing graph and compute the variance of path delay for *N* longest paths.
4) Compute the minimum and the maximum path-to-path correlation coefficients ($\Sigma_{\min}^{ij}$ and $\Sigma_{\max}^{ij}$).
5) For a specific delay value *t*
   a) Compute the coordinates of the normalized delay vector

$$t_i^{'} = (t - \mu_{D_i}) / \sigma_{D_i}, \quad i = 1, \ldots, N.$$

   b) Compute the average of the coordinates of the normalized delay vector

$$\bar{t} = \left(1/N\right) \sum_{i=1}^{N} t_i^{'}.$$

   c) Find the minimum coordinate of the normalized delay vector

$$t_{\min} = \min\{t_i^{'}\}.$$

   d) Look up the probability tables using path-to-path correlation coefficients ($\Sigma_{\min}^{ij}$ and $\Sigma_{\max}^{ij}$), $\bar{t}$, and $t_{\min}$, and obtain lower and upper bounds on the circuit delay distribution function at *t*.

$$P_{\Sigma_{\min}}(\bigcap\{Z_i \le t_{\min}\}) \le P_{\Sigma}(\bigcap\{Z_i \le t_i^{'}\}) \le P_{\Sigma_{\max}}(\bigcap\{Z_i \le \bar{t}\}).$$

Figure 2.2: Algorithm flow of computing upper and lower bounds for the circuit delay distribution.

2.3 IMPLEMENTATION AND EXPERIMENTAL RESULTS

The algorithms described above have been implemented in C++, and have been run on a PC with CPU 3.0GHz and 1GB memory. The algorithms have been tested on a set of combinational ISCAS ‘85 benchmark circuits. In the experiments, we take into account several process parameters, including the variability of the effective channel length, threshold voltage, and oxide thickness. The sensitivity values of process variation

are obtained from SPICE simulations for a cell library of 0.13um technology. The 3-sigma values of each parameter are shown in Table 2.1. The exact cumulative distribution function was computed via Monte-Carlo runs of the deterministic static timing analysis algorithm with samples taken from the relevant parameter distributions, where die-to-die and within-die components follow truncated Gaussian. For each simulation, 2000 iterations of Monte Carlo were run.

Table 2.1: The 3-sigma values of process parameters (the percentage of the mean value).

Process Parameters	3-sigma Values
Effective Channel Length	12-15%
Threshold Voltage	8-10%
Oxide Thickness	6-8%

Table 2.2: The run time and prediction error for the proposed algorithm. The run time of Monte Carlo simulation is also included.

Benchmark	Number of nodes	Bounding error RMS (%)				95th percentile error (%)		Run time (sec)		
		Lower bound		Upper bound				Monte Carlo simulations	Bounding algorithm	
		N=50	N=100	N=50	N=100	N=50	N=100		N=50	N=100
c880	456	3.64	4.50	4.14	6.19	2.90	3.25	37	1.97	2.03
c1355	605	2.49	3.10	2.88	3.18	2.44	2.79	56	2.02	2.08
c1908	975	2.84	3.18	4.03	4.86	2.18	2.51	79	2.09	2.24
c2670	1544	3.25	3.84	2.39	3.00	2.28	2.62	112	2.13	2.27
c3540	1787	1.98	2.19	1.61	1.94	1.59	1.59	153	2.22	2.38
c5315	2600	2.93	3.29	2.94	3.53	1.54	1.87	227	2.34	2.53
c6288	2448	1.50	1.72	0.79	0.90	1.10	1.10	246	2.58	3.19
c7552	3874	2.99	3.56	3.93	4.55	1.90	2.23	318	2.53	2.66

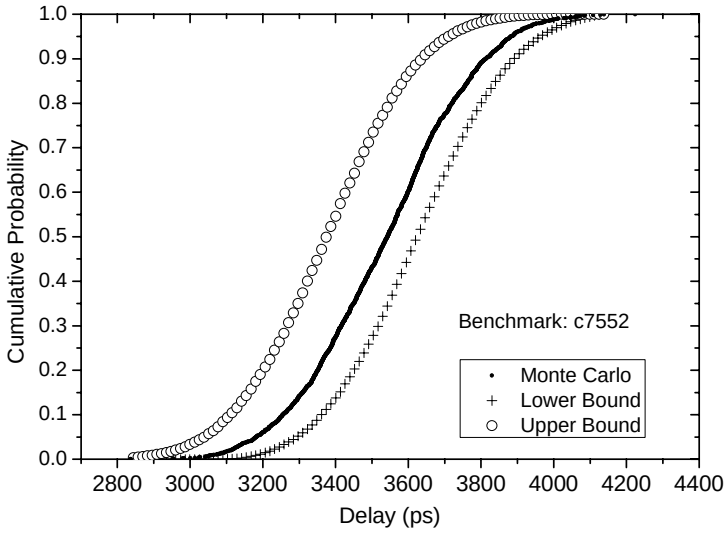

Figure 2.3: Comparison of cumulative probabilities for c7552 (N=100).

The bounds generated by the algorithm follow the Monte Carlo distribution closely. Table 2.2 shows the errors of the upper and lower bounds, as well as the 95th percentile error. The bounds were computed for N=50 and 100 longest paths. Across the benchmarks, the root-mean-square error of the lower bound is 1.72-4.50%, while the error of the upper bound is 0.90-6.19% for N=100. Note that we are specifically interested in the lower bound because it provides us a conservative value of the circuit delay at any confidence level. Importantly, the lower bound becomes tighter at higher confidence levels, giving a more reliable estimate of the parametric yield: at the 95th percentile the error is 1.10-3.25% (N=100). Figure 2.3 shows the cumulative probabilities from the Monte Carlo simulation and the proposed bounding algorithm, for the largest netlist c7552 in ISCAS '85 benchmark circuits. The 95th percentile error of the lower bound for the cumulative probability is 2.23%. Figure 2.4 demonstrates that accurately accounting for gate delay correlation is crucial in predicting the shape of *cdf*. The gate delay correlation is changed by adjusting the ratio of the variance of the die-to-die

component to the total variance. As illustrated in the figure, the mean value of the lowly-correlated case is larger than that of the highly-correlated one, while the span is much smaller.

Table 2.2 also shows the run time of the algorithm. The Monte Carlo simulations (2000 samples) are substantially slower than our algorithm. The implementation is very efficient: the run time is less than 4 seconds for the largest netlist in ISCAS '85 benchmark circuits, which contains more than 3,000 nodes. There exists a close-to-linear growth in the runtime of the algorithm as a function of the circuit size, which enables the practical use of the algorithm for significantly larger circuits. Although the time complexity of the algorithm is quadratic in the number of deterministically longest path, the implementation is still very fast because the path extraction is efficient even for large circuits.

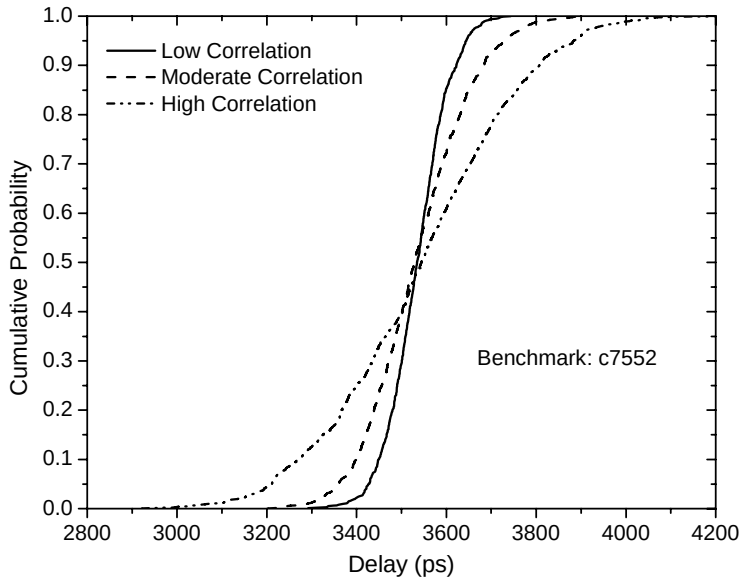

Figure 2.4: Change in the cumulative probabilities for c7552 depending on the level of gate delay correlations.

In the experiments we compare the proposed algorithm to the Monte Carlo method; the cumulative probabilities of the generated samples in the Monte Carlo method are used as the true cumulative probabilities. However, if we take into account the accuracy of the Monte Carlo method, the prediction error of the bounding algorithm actually becomes smaller. Figure 2.5 shows the *cdf* bounds from the proposed algorithm and the Monte Carlo method. The confidence level of the bounds from the Monte Carlo method is 99.7% (for 2000 samples). If we compare the upper bounds for 95th-percentile circuit delay, the prediction error of the bounding algorithm is only 1.15%, compared to the results of the Monte Carlo simulation. Thus, our bounding algorithm can construct very tight *cdf* bounds while achieving the run-time efficiency.

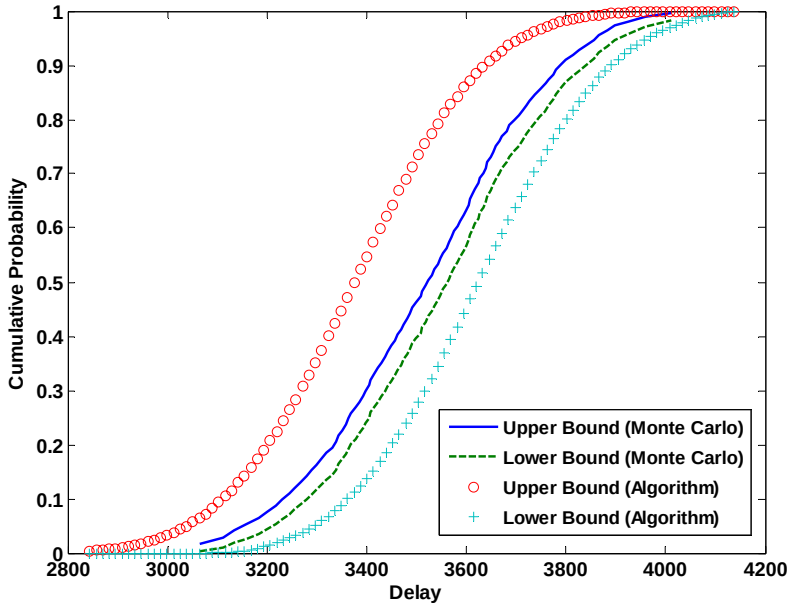

Figure 2.5: Bounds for cumulative probabilities of c7552 from Monte Carlo simulation (2000 samples) and the proposed algorithm (N=100).

Figure 2.6 illustrates the distinction between the upper bounds for circuit delay based on the truncated and pure normal assumptions of variability. The discussion in Section 2.1.4 states that the difference of the cumulative probability is substantial when the dimensionality is high and the correlation is low. Therefore, the number of the extracted paths in this experiment is large (N=1000), and extremely low gate delay correlations are assumed, which causes the minimum path-to-path correlation coefficients among the first N paths in the benchmark circuit c5315 to be very low (0.45 and 0.79, respectively). From Figure 2.6, it can be observed that at high percentiles (e.g., higher than the 90th percentile), the difference between the predicted upper bounds of circuit delay, by assuming pure or truncated normally distributed variability, is less than 3% for both cases. However, there exists a substantial distinction between the ranges of the circuit timing. Here we compare the circuit delay span by computing the difference between the 5th and 95th percentile delays. The results shown in Table 2.3 indicate that using truncated Gaussian distributions reduces the circuit delay spans by 46% and 19%, for $\rho_{\min} = 0.45$ and $\rho_{\min} = 0.79$, respectively. As a consequence, the proposed technique is able to provide a more accurate timing prediction when the effect of truncated normal distributions becomes important.

Table 2.3: The 5%-95% circuit delay span for circuit c5315.

Distribution and Minimum Correlation		**The 5th Percentile Delay (ps)**	**The 95th Percentile Delay (ps)**	**5%-95% Span (ps)**
Normal	$\rho_{\min} = 0.45$	3971	4164	193
Truncated Normal	$\rho_{\min} = 0.45$	4006	4110	104
Normal	$\rho_{\min} = 0.79$	3870	4278	408
Truncated Normal	$\rho_{\min} = 0.79$	3906	4238	332

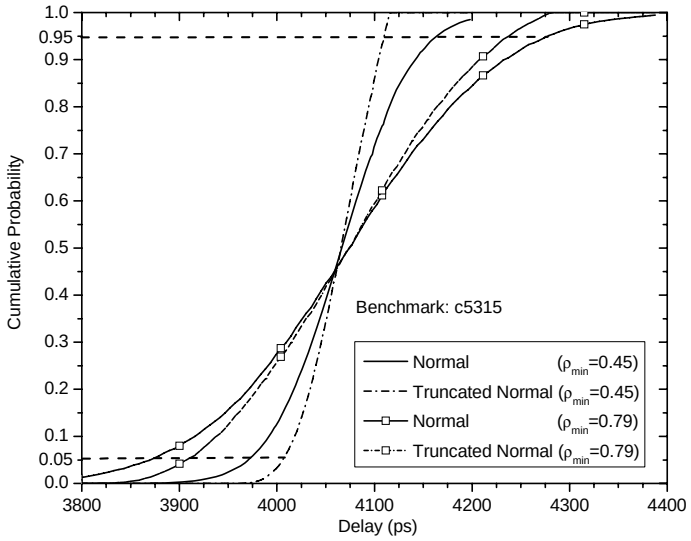

Figure 2.6: Lower bounds for the cumulative probabilities of circuit delay for c5315 based on the normally and truncated normally distributed variations (N=1000).

In the experiments, the number of statically longest paths is fixed (e.g., N=50 or 100). However, since the number of paths grows exponentially as the size of the circuit increases, there may be more paths with delays near the critical path delay for large circuits. Therefore, it is appropriate to increase the number of paths taken into account for large circuits. For example, when the 3σ value of the path delay is about 15% of the mean, we can choose paths with mean delays higher than 87% (~1/1.15) of the critical path delay. This is based on the assumption that it is very unlikely for a path delay to exceed its 3σ value. Besides, when path delays are highly correlated, we can choose much fewer paths because path delays vary similarly; in this condition, paths with lower mean delays are less likely to have the maximum path delay.

2.4 TECHNIQUES FOR DIVERSE CORRELATION MATRICES

In Section 2.1, we use the probability of the well-structured correlation matrix which has identical off-diagonal coefficients as a bound for the sought cumulative probability, based on the Slepian's inequality (Theorem 2.2). Then we use the cumulative probabilities of equicoordinate delay vectors (e.g., $(\bar{t},...,\bar{t})^T$ and $(t_{\min},...,t_{\min})^T$) as the bounds. The experimental results indicate that the proposed algorithm provides tight bounds for the circuit delay across the ISCAS'85 benchmarks. However, it is possible that within the path set $\{D_1,...,D_N\}$ there will be two, or more, groups of paths that are highly mutually correlated (for example, if these paths are all 'logic-heavy' paths) but the correlation between the groups of paths will be small. As an example, consider the matrices below:

$$\Sigma = \begin{bmatrix} 1 & .9 & .9 & .9 & .2 \\ .9 & 1 & .9 & .9 & .2 \\ .9 & .9 & 1 & .9 & .2 \\ .9 & .9 & .9 & 1 & .2 \\ .2 & .2 & .2 & .2 & 1 \end{bmatrix}, \text{ and } \Sigma_{\min} = \begin{bmatrix} 1 & .2 & .2 & .2 & .2 \\ .2 & 1 & .2 & .2 & .2 \\ .2 & .2 & 1 & .2 & .2 \\ .2 & .2 & .2 & 1 & .2 \\ .2 & .2 & .2 & .2 & 1 \end{bmatrix}.$$

In this case, using the matrix $\Sigma_{\min}$ above may lead to the lower bounds that are excessively loose, as illustrated in Figure 2.7. We have explored several ways of reducing such conservatism of lower bounds, which results from the large difference in correlation coefficients. Distinct from the algorithm in Section 2.2, these techniques need all correlation coefficients in addition to the maximum and minimum correlations. Besides, since path delays are mostly positively correlated, the assumption of the proposed techniques is that correlation coefficients of path delays are *nonnegative.*

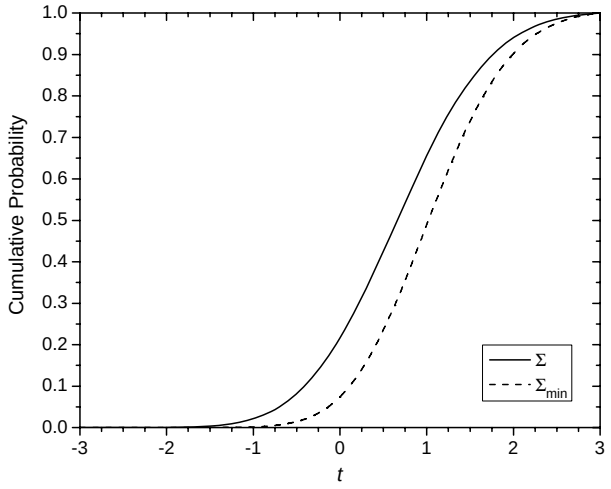

Figure 2.7: Equicoordinate cumulative probabilities, $P(\bigcap Z_i \le t)$, for correlation matrices, Σ and $\Sigma_{\min}$.

Technique 1. If the smallest coefficient of the original correlation matrix $\Sigma^{ij}_{\min} = \min\{\Sigma^{ij}\}$ is very small, the following transformation can be used to improve the quality of the bound. Let *q* be some small value of the positive correlation coefficient (for example, *q*=0.2). Let C be a subset of $\{1,...,N\}$, such that $\rho_{ij} < q$ for all $i \in C$ and $j \notin C$. We can generate a matrix Σ' by setting all the coefficients of Σ that are below *q* to zero, as illustrated in these two matrices:

$$\Sigma = \begin{bmatrix} 1 & .9 & .9 & .2 & .2 \\ .9 & 1 & .9 & .2 & .2 \\ .9 & .9 & 1 & .2 & .2 \\ .2 & .2 & .2 & 1 & .8 \\ .2 & .2 & .2 & .8 & 1 \end{bmatrix}, \text{ and } \Sigma' = \begin{bmatrix} 1 & .9 & .9 & 0 & 0 \\ .9 & 1 & .9 & 0 & 0 \\ .9 & .9 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & .8 \\ 0 & 0 & 0 & .8 & 1 \end{bmatrix}.$$

For jointly Gaussian random variables, zero correlation implies mutual independence. Then, from the Slepian's inequality, we know:

$$P_{\Sigma}\left(\bigcap\{Z_i \le t\}\right) \ge P_{\Sigma'}\left(\bigcap\{Z_i \le t\}\right) = P_{\Sigma_C}(\bigcap_{i\in C}\{Z_i \le t\}) \cdot P_{\Sigma_{C'}}(\bigcap_{i\notin C}\{Z_i \le t\}) \tag{2.19}$$

where Σ_C and $\Sigma_{C'}$ are correlation matrices for random vectors $\{Z_i : i \in C\}$ and $\{Z_i : i \notin C\}$, respectively. Therefore, we can use $P_{\Sigma_C}(\bigcap_{i\in C}\{Z_i \le t\}) \cdot P_{\Sigma_{C'}}(\bigcap_{i\notin C}\{Z_i \le t\})$ as the lower bound for the sought cumulative probability.

To implement this technique, we need to construct a probability table accounting for different sizes of the set *C*. In other words, we need the following cumulative probabilities:

$$P_{\Sigma_C}(\bigcap_{i=1}^{|C|}\{Z_i \le t\}) \tag{2.20}$$

where |*C*| is the size of the set *C*, which ranges from 1 to *N-1*. Note that the correlation coefficients in Σ_C may have many distinct values. To reduce the table size, we need correlation matrices with identical off-diagonal coefficients. From Theorem 2, we know that:

$$P_{\Sigma_C}(\bigcap_{i=1}^{|C|}\{Z_i \le t\}) \ge P_{\Sigma_{C\min}}(\bigcap_{i=1}^{|C|}\{Z_i \le t\}) \tag{2.21}$$

where $\Sigma_{C\min}$ is generated by setting all off-diagonal coefficients to $\min\{\Sigma_C{}^{ij}\}$. Therefore, $P_{\Sigma_{C\min}}(\bigcap_{i=1}^{|C|}\{Z_i \le t\})$ can be used as a lower bound for $P_{\Sigma_C}(\bigcap_{i=1}^{|C|}\{Z_i \le t\})$, which permits reducing the number of data points recorded in the probability table.

As for the size of the table, consider a fixed number of paths *N*. The table size is $(N-1)(L+1)M$, assuming the correlation coefficient ρ_{ij} increases from 0 to 1 at the interval of $1/L$, and M is the number of points in *t* domain. However, the table size can be reduced if we use a small value of *L* (e.g., *L*=10-20). Additionally, if the value of ρ_{ij} does not exist in the table, we can use the largest correlation coefficient that is smaller than ρ_{ij} because of the monotonic properties in Theorem 2.2.

Technique 2. If the smallest coefficient of the original correlation matrix $\Sigma^{ij}_{\min} = \min\{\Sigma^{ij}\}$ is smaller than the majority but is not close to zero, the above technique may not be effective enough. Such would be the case if the correlation matrix was given by:

$$\Sigma = \begin{bmatrix} 1 & .9 & .9 & .5 & .5 \\ .9 & 1 & .9 & .5 & .5 \\ .9 & .9 & 1 & .5 & .5 \\ .5 & .5 & .5 & 1 & .9 \\ .5 & .5 & .5 & .9 & 1 \end{bmatrix}.$$

In this case, we still can *use the monotonic properties* of the cumulative probability with respect to the correlation coefficients, described by Theorem 2.2, to improve the bound. We could use the known probability values for distributions characterized by the tabulated correlation matrices to bound the probability for the matrix of interest.

Below we formalize this idea, and describe a constructive procedure for deriving better bounds for the probability of a path delay vector with an arbitrary correlation matrix.

Let $\Sigma_{(N,\rho),(m,q)}$ be a correlation matrix for a Gaussian random vector, with the off-diagonal correlation coefficient $\rho_{ij} = \begin{cases} \rho & \text{if } i \le N-m \ \wedge \ j \le N-m \\ q & \text{if } i > N-m \ \vee \ j > N-m \end{cases}$. Assume that $\rho > q \ge 0$.

For example, $\Sigma_{(N,\rho),(m,q)} = \begin{bmatrix} 1 & \rho & \rho & \rho & \rho \\ \rho & 1 & \rho & \rho & \rho \\ \rho & \rho & 1 & \rho & \rho \\ \rho & \rho & \rho & 1 & \rho \\ \rho & \rho & \rho & \rho & 1 \end{bmatrix}$, where m=0; $\Sigma_{(N,\rho),(m,q)} = \begin{bmatrix} 1 & \rho & \rho & \rho & q \\ \rho & 1 & \rho & \rho & q \\ \rho & \rho & 1 & \rho & q \\ \rho & \rho & \rho & 1 & q \\ q & q & q & q & 1 \end{bmatrix}$, where m=1, and $\Sigma_{(N,\rho),(m,q)} = \begin{bmatrix} 1 & \rho & \rho & q & q \\ \rho & 1 & \rho & q & q \\ \rho & \rho & 1 & q & q \\ q & q & q & 1 & q \\ q & q & q & q & 1 \end{bmatrix}$, where m=2. In other words, the region of the higher correlation coefficient (ρ) shrinks as m increases.

Let $P(N,\rho,m,q) = P_{\Sigma_{(N,\rho),(m,q)}}\left[\bigcap_{i=1}^{N}\{Z_i \le t\}\right]$. The values of this function are known for a set of pre-characterized tables computed using the Monte-Carlo integration, including combinations of ρ, m, and q.

Since $\rho > q$, from Corollary 1 of Theorem 2.2 we can use $P(N,\rho,m=N,q)$ as the lower bound for $P(N,\rho,m,q)$. Besides, from Theorem 2.2 we can infer that:

$$m_1 \le m_2 \Rightarrow P(N,\rho,m_1,q) \ge P(N,\rho,m_2,q). \tag{2.22}$$

Therefore, if the cumulative probability of a correlation matrix can be bounded by $P(N,\rho,m,q)$ with $m < N$, then a tighter lower bound can be obtained to improve the conservative bound of $P(N,\rho,m=N,q)$. This fact is illustrated in Figure 2.8.

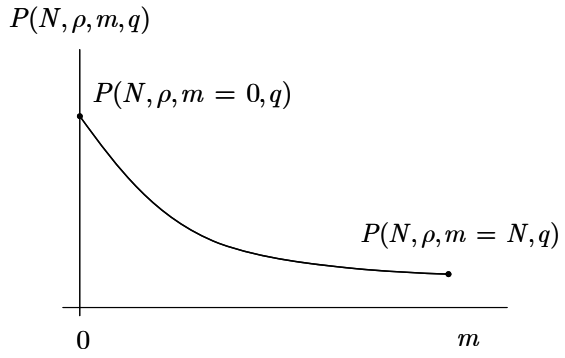

Figure 2.8: The lower bound of the cumulative probability can be improved when $m < N$.

For a specific correlation matrix Σ, first we can use the Slepian's inequality to find a correlation matrix Σ' with a lower cumulative probability, but *such that it only contains two different values of off-diagonal correlation coefficients*. Then, we can use the monotonic properties again to find another matrix in the form of $\Sigma_{(N,\rho),(m,q)}$, of which the probabilities are pre-generated and can be used as a lower bound for $P_{\Sigma'}$, and P_{Σ}. Figure 2.9 shows the pseudo code of the described algorithm.

Let Σ be a correlation matrix, and the maximum and the minimum of the off-diagonal coefficients are $\rho_{\max}$ and $\rho_{\min}$, respectively.

1) Choose an appropriate value ρ between $\rho_{\max}$ and $\rho_{\min}$. Set $q = \rho_{\min}$.

2) Construct a matrix Σ'. For each off-diagonal coefficient ρ_{ij}' in Σ', set $\rho_{ij}' = \begin{cases} q & \text{if } \rho_{ij} < \rho \\ \rho & \text{if } \rho_{ij} \geq \rho \end{cases}$.

3) Transform Σ'.

Starting from $m = N$-1,

a) Switch rows (and corresponding columns for symmetry) of Σ' such that

$$\rho_{ij}' = \rho \ \forall i, j : i \leq N\text{-}m \wedge j \leq N\text{-}m .$$

b) Find the smallest m such that the above condition holds. Then,

$$P_{\Sigma}\left[\bigcap_{i=1}^{N}\{Z_i \leq t\}\right] \geq P_{\Sigma'}\left[\bigcap_{i=1}^{N}\{Z_i \leq t\}\right] \geq P(N,\rho,m,q) .$$

Figure 2.9: Algorithm of Technique 2.

Example: Let $\Sigma = \begin{bmatrix} 1 & 0.9 & 0.3 & 0.9 & 0.3 \\ 0.9 & 1 & 0.3 & 0.9 & 0.3 \\ 0.3 & 0.3 & 1 & 0.3 & 0.8 \\ 0.9 & 0.9 & 0.3 & 1 & 0.3 \\ 0.3 & 0.3 & 0.8 & 0.3 & 1 \end{bmatrix}$; this matrix includes different off-diagonal correlation coefficients (e.g. 0.3, 0.8, and 0.9). Here we choose $\rho = 0.6$ for the purpose of demonstration, and set $q = \min(0.3, 0.8, 0.9) = 0.3$.

Then we generate a correlation matrix having only two distinct values of correlation coefficients:

$$\Sigma' = \{\rho_{ij}'\} = \begin{bmatrix} 1 & 0.6 & 0.3 & 0.6 & 0.3 \\ 0.6 & 1 & 0.3 & 0.6 & 0.3 \\ 0.3 & 0.3 & 1 & 0.3 & 0.6 \\ 0.6 & 0.6 & 0.3 & 1 & 0.3 \\ 0.3 & 0.3 & 0.6 & 0.3 & 1 \end{bmatrix}.$$

From Theorem 2.2, we know that $P_{\Sigma}\left[\bigcap_{i=1}^{5}\{Z_i \le t\}\right] \ge P_{\Sigma'}\left[\bigcap_{i=1}^{5}\{Z_i \le t\}\right]$, so $P_{\Sigma'}\left[\bigcap_{i=1}^{5}\{Z_i \le t\}\right]$ can be used as a lower bound for $P_{\Sigma}\left[\bigcap_{i=1}^{5}\{Z_i \le t\}\right]$. However, this value is not pre-generated because Σ' cannot be described by $\Sigma_{(N,\rho),(m,q)}$. Therefore, we need to transform Σ' into a more appealing form so that it is easy to find a lower bound of Σ'.

Now we transform Σ' such that $\rho_{ij}' = \rho \quad \forall i, j : i \le N\text{-}m \wedge j \le N\text{-}m$, and find the minimum *m*. When *m*=4 and 3, the above condition holds. We can switch the 3rd and 4th rows (and columns) to make *m* = 2, and then obtain the matrix Σ':

$$\Sigma' = \begin{bmatrix} 1 & 0.6 & 0.6 & 0.3 & 0.3 \\ 0.6 & 1 & 0.6 & 0.3 & 0.3 \\ 0.6 & 0.6 & 1 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 & 1 & 0.6 \\ 0.3 & 0.3 & 0.3 & 0.6 & 1 \end{bmatrix}.$$

Note that the value of $P_{\Sigma'}$ does not change after this transformation.

At this step, it is impossible to further reduce *m*; therefore, the minimum value of *m* is 2. Then we can use $P(N, \rho, m, q)$ as a lower bound for $P_{\Sigma'}$ because the correlation matrix of $P(N = 5, \rho = 0.6, m = 2, q = 0.3)$ is:

$$\Sigma_{(6,0.6),(2,0.3)} = \begin{bmatrix} 1 & 0.6 & 0.6 & 0.3 & 0.3 \\ 0.6 & 1 & 0.6 & 0.3 & 0.3 \\ 0.6 & 0.6 & 1 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 & 1 & 0.3 \\ 0.3 & 0.3 & 0.3 & 0.3 & 1 \end{bmatrix}.$$

Comparing Σ' and $\Sigma_{(6,0.6),(2,0.3)}$ we conclude that:

$$P_{\Sigma}\left[\bigcap_{i=1}^{5}\{Z_i \le t\}\right] \ge P_{\Sigma'}\left[\bigcap_{i=1}^{5}\{Z_i \le t\}\right] \ge P(5, 0.6, 2, 0.3).$$

Finally, we find a lower bound, $P(5, 0.6, 2, 0.3)$ of which the cumulative probability is in the pre-characterized table. This bound is better than the cumulative probability computed by setting all off-diagonal elements to the minimum correlation coefficient, i.e., $P(5, 0.6, m = 5, 0.3)$. Therefore, this technique permits improving the conservative lower bound.

In this technique, it is important to determine the appropriate value of ρ because it has a great impact on the quality of the lower bound. In the example, we use $\rho = (\rho_{\max} + \rho_{\min})/2$; however, we can choose any values within $[\rho_{\min}, \rho_{\max}]$. If most of

the correlation coefficients are close to $\rho_{\max}$, we should choose ρ close to $\rho_{\max}$ thus avoid the over-conservatism of using a small ρ.

Another example here quantitatively illustrates the importance of using the proposed technique. Consider a correlation matrix Σ' with *N*=20, $\rho = 0.9$, and *q*=0.5. Apparently, we can use $P(N, \rho, m, q) = P(20, 0.9, 20, 0.5)$ as a lower bound. However, if Σ' can be transformed such that the minimum value of *m* is 10, we have a better lower bound: $P(N, \rho, m, q) = P(20, 0.9, 10, 0.5)$. Figure 2.10 shows the cumulative distribution functions of two different values of *m*, 10 and 20, respectively. These results show that this technique is able to provide a better lower bound for the cumulative distribution function of the multivariate normal distribution when the correlation coefficients have a large span.

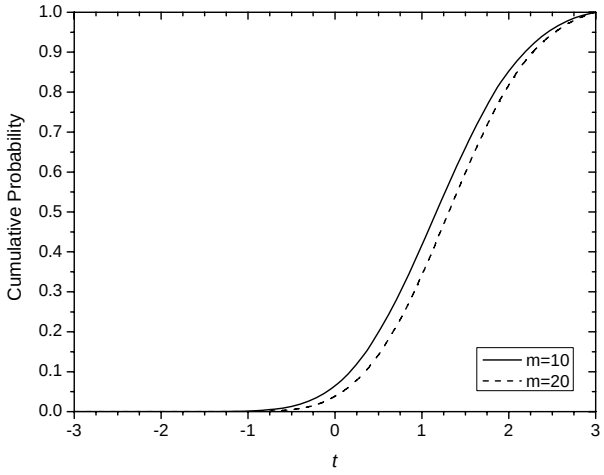

Figure 2.10: Equicoordinate cumulative probabilities of different *m* values ($P(N, \rho, m, q)$, where *N*=20, ρ=0.9, *m*=10 or 20, and *q*=0.5).

Finally, we need to take into the feasibility of the technique, especially the amount of data in the probability table. First, we need to consider the pair (ρ, q) in the algorithm. Suppose the correlation coefficient increases from 0 to 1 at the interval of $1/L$, then we have $(L+1)L/2$ pairs of (ρ, q). Thus, if we only consider a fixed number of paths *N*, the table contains $M(N-1)(L+1)L/2$ data points, where *M* is the number of points in *t* domain. As in Technique 1, a small value of *L* (e.g., 10-20) can be used to prevent the table from taking too much storage space. Additionally, if a specific correlation coefficient ρ (or q) does not exist in the table, we can use the largest correlation coefficient that is smaller than ρ(or q) because of the monotonic properties described in Theorem 2.2.

2.5 SUMMARY

This chapter proposes a path-based statistical timing analysis algorithm based on a new mathematical formulation. Instead of approximating the cumulative distribution function of the circuit, the algorithm computes a tight bound for the circuit delay distribution. Based on the theory of majorization, the equicoordinate random vectors are used to bound the actual cumulative distribution function, enabling efficient numerical evaluation of the cumulative probability.

Path delay correlations are derived based on a statistical timing model that accounts for the die-to-die and the within-die components of variations. This chapter shows that the correlations of path delays result from the dependence on the die-to-die components, as well as the structure of paths. Taking into account multiple process parameters, the proposed statistical timing algorithm can better capture the variability resulting from a number of sources. With the scalable path extraction algorithm, the implementation is very efficient compared to the Monte-Carlo simulations.

Chapter 3: Statistical Static Timing Analysis Based on Incomplete Probabilistic Descriptions of Parameters

The area of statistical static timing analysis has recently made substantial progress in terms of algorithmic and modeling advances. Extensions of the basic framework of SSTA have been investigated to capture non-linear delay response and non-Gaussian parameters [47]-[49]. However, the practical use of SSTA algorithms may be restricted because the fundamental assumption of statistical approaches is not always true: distributions of parameters may be unavailable. In some cases, only mean, or variance can be estimated, but existing approaches cannot handle this kind of information. To tackle this problem, this dissertation proposes a modeling strategy of parameter uncertainty; the variability of parameters is described by partial probabilistic descriptions (i.e., mean and variance), in addition to the interval. Specifically, this dissertation proposes analytical techniques and a robust Monte-Carlo sampling method for bounding the distributions based on incomplete probabilistic descriptions. A path-based timing algorithm implementing the proposed modeling strategy, as well as handling the traditional Gaussian variability descriptions, has been developed. The proposed timing algorithm can reduce the over-conservatism of the interval-based delay estimates.

This chapter is organized as follows. Section 3.1 describes the proposed strategy of handling partially-specified uncertainty. Section 3.2 presents the statistical timing analysis algorithm implementing the developed strategy, and Section 3.3 shows the experimental results. Finally, Section 3.4 summarizes this work.

3.1 NEED FOR NEW UNCERTAINTY MODEL: PARTIAL PROBABILISTIC DESCRIPTIONS OF PARAMETERS

3.1.1 Limited Availability of Full Characterization Data

The basic assumption of statistical approaches is that the detailed statistical information about parameters is accessible; in practice, this assumption is not always valid. Such information may be unavailable due to the limited measurements, or remains confidential in consideration of business competitions. Furthermore, the characteristics of manufacturing processes may change on a day-to-day or week-to-week basis, which poses difficulties for process characterizations. Additionally, it is even more difficult to extract variability descriptions of environmental parameters due to the dependence on input vectors. Thus, typically the intervals of hard-to-measure parameters are used to estimate circuit performance [50].

However, in many instances, some but not full probabilistic information is available; for example, the mean and, possibly, the variance of process parameters can be easily estimated, as compared to the distributions. For environmental parameters, although the joint spatial- and time-dependent characterization is virtually impossible [51], the variability can be considered separately in spatial and temporal domains. Variability in the temporal domain is primarily caused by the input vectors; thus, to ensure a robust timing estimate, it is necessary to take into account the entire range of the parameters. In contrast, when we consider the spatial domain, not all the circuit nodes have identical worst-/best-case values of environmental parameters. To illustrate this, let $V_{dd}(\boldsymbol{x},t)$ denote the power supply voltage of a node that is a function of the spatial location and the time, where $\boldsymbol{x}$ denotes the spatial location, and t denotes the time. Then the worst-case supply voltage of a node is defined as:

$$V_{dd}{}^{wc}(\boldsymbol{x}) = \min V_{dd}(\boldsymbol{x},t) \text{ for all } t\,. \tag{3.1}$$

If all circuit nodes are assumed to have identical intervals of environmental parameters, it may lead to over-pessimistic estimates. If the worst-case voltage drop of each node, $V_{dd}{}^{wc}(\boldsymbol{x})$, can be estimated, the expected value of $V_{dd}{}^{wc}(\boldsymbol{x})$ across the spatial domain ($\boldsymbol{x}$) can be also characterized:

$$E\left[V_{dd}{}^{wc}(\boldsymbol{x})\right] = \sum_{i=1}^{N} V_{dd}{}^{wc}(\boldsymbol{x_i}) \Big/ N \tag{3.2}$$

where N is the number of nodes. In fact, the worst-case voltage drop of each circuit node and the expected value in (3.2) can be estimated by power-grid verification techniques [52] or Monte Carlo sampling procedures. Thus, it is possible to estimate partial probabilistic descriptions (e.g., mean and variance) of process and environmental parameters; however, existing statistical timing analysis algorithms cannot utilize this kind of incomplete probabilistic information. Thus, in addition to the existing techniques, a new way of treating uncertain variables with partial probabilistic information is needed to enable practical design under uncertainty.

3.1.2 Proposed Strategy of Handling Partially-Specified Uncertainty

This dissertation develops a solution of timing analysis under uncertainty based on the principles of probabilistic interval models. The modeling strategy is based on the generalization of classical random variables to variables described by families of distributions. Conceptually, the most general description of an uncertain variable is an interval. Such descriptions form the basis of interval arithmetic and its enhancement in terms of affine arithmetic [25], [53]. An interval description does not have the notion of probability; therefore it does not permit making statements about which values of the variable are more likely. If , the statistical metrics, such as mean and variance, are known in addition to the range, the interval methods are incapable of utilizing this additional information in computing the arithmetic operations (+, -, *, /, max, min).

Probabilistically-enhanced interval analysis is a natural synergy of pure interval arithmetic and probabilistic analysis. It permits the use of partial statistic information (e.g., range, mean, and variance) to quantify the likelihood of the variable. The estimates are guaranteed to be conservative regardless of the precise form of the distribution. For the fully specified random variable (e.g., Gaussian), the most general representation is its cumulative distribution function. For a partially-specified random variable, the most general representation is a bound for the *cdf*, forming a p-box [26].

Following the above philosophy, this dissertation proposes a timing analysis algorithm that produces reliable timing estimates even if the characterization data is incomplete. Compared to prior work on estimating the probabilistic delay bound for an arbitrary PERT network [24], the essential contribution is in handling incomplete uncertainty description. Compared to affine methods [54], the developed strategy can handle both the interval and probabilistic descriptions formally and consistently, without resorting to heuristic assumptions about distributions within the intervals. Additionally, the proposed strategy is compatible with the existing SSTA tools, with the capability of handling the Gaussian variability and the first-order delay modeling.

3.2 TIMING ANALYSIS UNDER PARTIAL PROBABILISTIC DESCRIPTIONS

This section introduces an application of the new probabilistic interval techniques to timing analysis. Section 3.2.1 describes the construction of p-boxes for path delays. Section 3.2.2 constructs the bound of the circuit delay distribution, and presents a method to combine the results of the traditional SSTA with the above derivations.

3.2.1 Path Delay Computation

The timing model used in this framework is based on the additive delay model containing both the uncertainty due to classical random variables and the newly

introduced probabilistic interval variables. The probabilistic interval variables (as opposed to random variables) are variables for which only partial statistical metrics, mean and variance, are available in addition to the known range, or interval. The delay model can be expressed as:

$$d_i = \mu_i + \sum_{j=1}^{n} a_{i,j}\Delta x_{i,j} + \sum_{k=1}^{m} b_{i,k}\Delta y_{i,k} \tag{3.3}$$

where μ_i is the mean value of the gate delay, $\Delta x_{i,j}$ is a zero-mean Gaussian random variable, and $\Delta y_{i,k}$ is a zero-mean probabilistic interval variable. The coefficients $a_{i,j}$ and $b_{i,k}$ are sensitivities of gate delays, representing the first-order derivatives of delays with respect to the variables.

A concise representation of the gate delay model can be obtained by resorting to the matrix form:

$$d_i = \mu_i + A_i^T X_i + B_i^T Y_i \tag{3.4}$$

where the matrices $A_i = [a_{i,1} \cdots a_{i,n}]^T$, $B_i = [b_{i,1} \cdots b_{i,m}]^T$, $X_i = [\Delta x_{i,1} \cdots \Delta x_{i,n}]^T$, and $Y_i = [\Delta y_{i,1} \cdots \Delta y_{i,m}]^T$. The variation of parameters can be further decomposed into the linear sum of die-to-die (X_{dd}, Y_{dd}) and independent within-die components $\left(X_{i,wd}, Y_{i,wd}\right)$:

$$d_i = \mu_i + A_i^T X_{i,wd} + A_i^T X_{dd} + B_i^T Y_{i,wd} + B_i^T Y_{dd} \tag{3.5}$$

The delay of path P_j can be represented by:

$$\begin{aligned} D^j &= \sum_{i \in P_j} (\mu_i + A_i^T X_{i,wd} + A_i^T X_{dd} + B_i^T Y_{i,wd} + B_i^T Y_{dd}) \\ &= \sum_{i \in P_j} \mu_i + \sum_{i \in P_j} g_i + \sum_{i \in P_j} u_i \end{aligned} \tag{3.6}$$

where $g_i = A_i^T X_{i,wd} + A_i^T X_{dd}$, and $u_i = B_i^T Y_{i,wd} + B_i^T Y_{dd}$.

It is convenient to separate the contributions of random delay uncertainty (D_R) and probabilistic interval uncertainty (D_{PI}): $D_R^j = \sum_{i \in P_j} g_i$ and $D_{PI}^j = \sum_{i \in P_j} \mu_i + \sum_{i \in P_j} u_i$.

Computing the path delay distribution when the gate delays are Gaussian is straightforward. Therefore, we focus on the *delay variation resulting from probabilistic interval variables*, i.e., D_{PI}^{j}. The range of the gate delay variation, u_i, is

$$u_i \in \left[\sum_{k=1}^{m}\left(\mid b_{i,k} \mid \underline{\Delta y_{i,k}}\right), \sum_{k=1}^{m}\left(\mid b_{i,k} \mid \overline{\Delta y_{i,k}}\right)\right] \tag{3.7}$$

where $\underline{\Delta y_{i,k}}$ and $\overline{\Delta y_{i,k}}$ are the lower and upper bounds of $\Delta y_{i,k}$, and $\mid b_{i,k} \mid$ denotes the absolute value of $b_{i,k}$. Then we can compute the range of D_{PI}^{j}.

Because the mean values of probabilistic interval variables are zero, the mean of the path delay is:

$$E[D_{PI}^{j}] = \sum \mu_i, i \in P_j \tag{3.8}$$

The variance of the path delay can be computed by:

$$Var\left\{D_{PI}^{j}\right\} = \sum_{i \in P_j} B_i^T \Sigma_{i,wd} B_i + \left(\sum_{i \in P_j} B_i^T\right) \Sigma_{dd} \left(\sum_{i \in P_j} B_i\right) \tag{3.9}$$

where $\Sigma_{i,wd}$ and Σ_{dd} are the covariance matrices of $Y_{i,wd}$ and Y_{dd}, respectively. Since different kinds of parameters are uncorrelated, the covariance matrices are actually diagonal matrices, with the diagonal elements equal to the variance of variables.

While the ultimate objective is to derive the circuit delay distribution, being able to describe individual path delay distributions is also essential. Now that the range, the mean and the variance of D_{PI}^{j} are known, the challenge is to compute the p-box that contains the family of distributions satisfying the given partial statistical information. Since we seek a fast analytical solution, we prefer to use an inequality, which is a sophisticated generalization of the one-sided Chebyshev inequality [43] and the Cantelli inequality [26], [55]. This inequality applies when, in addition to the first two moments

of the variable, its range is also known, resulting in a much tighter bound on the *cdf*. The upper bound for the cumulative probability of a variable X is given in [26]:

$$\begin{aligned}
&P(X \le x) = 0 && x < \underline{X} \\
&P(X \le x) \le 1\Big/(1 + (\mu - x)^2\Big/\sigma^2) && \underline{X} \le x < \mu + \sigma^2\Big/(\mu - \overline{X}) \\
&P(X \le x) \le 1\Big/(1 + (\mu - x)^2\Big/\sigma^2) && \mu + \sigma^2\Big/(\mu - \overline{X}) \le x < \mu + \sigma^2\Big/(\mu - \underline{X}) \\
&P(X \le x) \le 1 - (m^2 - my + s^2)\Big/(1 - y) && \mu + \sigma^2\Big/(\mu - \underline{X}) \le x
\end{aligned} \tag{3.10}$$

where μ denotes the mean, σ^2 denotes the variance, $\underline{X}$ is the lower bound, $\overline{X}$ is the upper bound, $y = (x - \underline{X})\Big/(\overline{X} - \underline{X})$, $m = (\mu - \underline{X})\Big/(\overline{X} - \underline{X})$, and $s^2 = \sigma^2\Big/(\overline{X} - \underline{X})^2$.

Similarly, the lower bound of the cumulative probability is:

$$\begin{aligned}
&P(X \le x) = 0 && x < \mu + \sigma^2\Big/(\mu - \overline{X}) \\
&P(X \le x) \ge 1 - (m(1 + y) - s^2 - m^2)\Big/y && \mu + \sigma^2\Big/(\mu - \overline{X}) \le x < \mu + \sigma^2\Big/(\mu - \underline{X}) \\
&P(X \le x) \ge 1\Big/(1 + \sigma^2\Big/(x - \mu)^2) && \mu + \sigma^2\Big/(\mu - \underline{X}) \le x < \overline{X} \\
&P(X \le x) = 1 && \overline{X} \le x
\end{aligned} \tag{3.11}$$

Then we can use (3.10) and (3.11) to compute the bound for the path delay distribution.

The same analytical structure can be used when the mean and variance are known only with certain accuracy [56]. First, the maximum of the variance should be used in the generalized Chebyshev inequality because it primarily determines the span of the *cdf*. Second, the upper bound of the mean should be used when computing the lower bound of the probability using (3.11), because it leads to the *worst* lower bound of the probability. Similarly, the lower bound of the mean should be used in (3.10).

Having computed the distribution of path delay variation due to probabilistic interval variables, now we combine it with the delay variation resulting from Gaussian variables. Since parameters of different categories are independent, it means that the

delay variations D_R^j and D_{PI}^j are independent, and the bound for the *cdf* of the sum can be computed by convolution:

$$CDF(D^j) = CDF(D_{PI}^j) \otimes f(D_R^j) \tag{3.12}$$

where $f(D_R^j)$ is the probability density function of D_R^j. We use the lower and upper bounds of $CDF(D_{PI}^j)$ in convolution respectively, and then obtain the bounds of $CDF(D^j)$. Finally, we have the bound for the path delay distribution, which enables computing the bound of path delay at any percentile.

3.2.2 Circuit Timing Computation

In this section a new technique is developed for efficient construction of p-boxes on the distribution of circuit delay, i.e., the maximum of all path delays. From (3.6), the bound of the circuit delay can be computed by:

$$\begin{aligned} D_{\max} &= \max(D^1,...,D^N) \\ &= \max\left(D_R^1 + D_{PI}^1,...,D_R^N + D_{PI}^N\right) \\ &\le \max\left(D_R^1,...,D_R^N\right) + \max\left(D_{PI}^1,...,D_{PI}^N\right) \end{aligned} \tag{3.13}$$

Let $D_{R\max} = \max\left(D_R^1,...,D_R^N\right)$ be the term due to random probabilistic variability, and the second term $D_{PI\max} = \max\left(D_{PI}^1,...,D_{PI}^N\right)$ be the term due to interval-probabilistic variability. In deriving the p-box for circuit delay, we adopt a strategy in which the sources of uncertainty described probabilistically are separated from interval probabilistic uncertainty. The distribution of $D_{R\max}$ can be computed by the statistical timing analysis algorithm based on the first-order delay models (e.g., the algorithm described in Chapter 2). Therefore, we concentrate on the computation of $D_{PI\max}$. The two terms are then combined to generate the bounds for the circuit delay distribution.

In constructing the probability box for the circuit delay distribution, ideally, we would like to use analytical means as was done in the previous section. The generalized Chebyshev inequality can be used to find the bounds on the distribution of $D_{PI\max}$, once the mean, the variance, and the range are known. However, for general functions of probabilistic interval variables, finding the bounds on the variance is NP-hard [57]. We show below that for *convex* functions the exact bound on the variance can be computed. Let us first establish the convexity of the term $D_{PI\max}$. The path delay D_{PI}^{j} is a linear and thus convex function of the die-to-die and within-die components of $\Delta y_{i,k}$. The circuit delay is given by $D_{PI\max} = \max(D_{PI}^{1},...,D_{PI}^{N})$ which is also a convex function of probabilistic interval variables [58]. Convexity is essential to our efficient analysis strategy, since as the theorem below shows determining the probability bound and moments of distributions of convex functions is much easier.

The proposed strategy is essentially based on the development of the robust (guaranteed) approach to Monte Carlo sampling from an unknown distribution. The Monte-Carlo simulation is a widely-used technique to solve complex numerical problems [59]. It can be used to estimate the timing performance of integrated circuits when the distributions are known [16], [17]. Without the full distributional knowledge of the parameters, a possible way to perform the simulation is to heuristically generate a variety of distributions that correspond to the given partial information. However, this method is not mathematically robust because it is impossible to enumerate all possible distributions. Besides, the high computational cost accounting for a number of distributions prevents this method from practical use. We show that for convex functions the *robust Monte Carlo* simulation can be rigorously and efficiently performed. Compared to the traditional approach to Monte-Carlo simulation, the selection of distribution is *justified* in our simulation strategy; only distributions that cause the extreme value of the target function

need to be considered. Therefore, this selective strategy is guaranteed to produce a bounding distribution, and achieves high efficiency in terms of the run time. Theorem 3.1 defines the algorithm for such robust Monte Carlo simulation [27].

Theorem 3.1: Let $\{v_1,...,v_M\}$ be a set of *independent* random variables, where $v_i \in [\underline{v_i}, \overline{v_i}]$, and $E[v_i] = E_i$ for *i*=1 to *M*. Let $f(v_1,...,v_M)$ be a non-negative convex function of the random variable v_i, for *i*=1 to *M*. Then, among all possible *cdf*s of $\{v_i : i = 1..M\}$ that correspond to the partial statistical information of the range and the mean, the k^{th} moment of the function, $E[y^k]$, where $y = f(v_1,...,v_M)$, achieves the maximum value when each random variable v_i follows a specific two-point distribution ,

$$\begin{aligned} P(v_i = \underline{v_i}) &= \underline{p_i} \\ P(v_i = \overline{v_i}) &= \overline{p_i} \end{aligned} \tag{3.14}$$

where $\underline{p_i} = \dfrac{\overline{v_i} - E_i}{\overline{v_i} - \underline{v_i}}$, and $\overline{p_i} = \dfrac{E_i - \underline{v_i}}{\overline{v_i} - \underline{v_i}}$. Furthermore, $E[y^k]$ achieves the minimum when $P(v_i = E_i) = 1$ [27].

Using the above sampling procedure guarantees the bounds of $E\left[f\left(v_1,...,v_M\right)\right]$ can be estimated. The detailed descriptions and applications of the two-point distribution can be found in [27], [60], and [61].

Theorem 3.1 effectively reduces the number of possible distributions that must be considered in order to find the bound of the moments. Thus, we develop a fast *hybrid* approach, *the fast robust Monte Carlo simulation*, in which robust Monte Carlo is used to get a quick estimate of the moments and then analytical techniques are used for establishing bounds. In the fast Monte Carlo simulation, a limited number of random samples are drawn following Theorem 3.1. It guarantees that we will get a robust estimate of the range of the mean circuit delay. As for the variance of the circuit delay, it can also be bounded by the sample variance because the two-point distribution in (3.14)

results in the maximum variance of gate delays, thus maximizing the variance of path delays and the circuit delay. Therefore, the generalized Chebyshev inequality can be then used to compute the bound of the distribution analytically.

Figure 3.1 describes the algorithm of the fast Monte Carlo simulation. This proposed strategy estimates the upper bound of sample mean and sample variance with only a limited number of runs. In practice, a few hundred runs are sufficient to generate an estimate with reasonable accuracy. This can be verified by considering the standard error of the sample mean and the confidence level of the true mean, i.e., the mean of the population. From [62], the 99% confidence interval of the true mean (μ) for a variable X is $\overline{X} \pm 2.575\,\sigma_X \big/ \sqrt{N}$, where $\overline{X}$ is the sample mean, σ_X is the true standard deviation, and N is the number of samples. For example, consider a circuit with extremely large span in the delay domain: the 3σ value of circuit delay is 45% of the mean. Then we estimate the confidence level:

$$P(\left|\overline{X} - \mu\right| \leq 2.575 \cdot 0.15\mu \big/ \sqrt{N}) = 0.99 \tag{3.15}$$

for $i = 1 \ldots N$
  Generate a sample for each die-to-die component of parameter.
  for each gate
    Generate a sample for each within-die component of parameter.
    Compute gate delay.
  end
  Use static timing analysis to compute the circuit delay, D_i.
end
Compute the mean and the variance of samples:

$$D_{avg} = \sum_{i=1}^{N} D_i \Big/ N$$

$$s_D^2 = \sum_{i=1}^{N} (D_i - D_{avg})^2 \Big/ N - 1$$

With D_{avg}, s_D^2, and the range of the circuit delay, use (3.11) to compute the lower bound of the *cdf*.

Figure 3.1: Algorithm for the fast robust Monte Carlo simulation.

The error of the sample mean for N = 1000 is less than 1.2% with probability equal to 0.99, which has a very limited impact on the result of using the generalized Chebyshev inequality. Thus, the accuracy of Monte Carlo for such a sample size is acceptable.

Once the lower bound on the distribution of $D_{PI\max}$ is generated, the overall circuit delay distribution can be bounded by combining $D_{PI\max}$ and $D_{R\max}$. Since these two components of delay variation are independent, the distribution of the sum can be computed by convolution, similar to (3.12). The lower bound of the *cdf* is used in the convolution because it represents the upper bound of the delay, which is an important metric for timing analysis.

3.3 EXPERIMENTAL RESULTS

The timing analysis algorithms based on partial description of uncertainty in Section 3.2 have been implemented in C++, and have been tested on a set of combinational ISCAS '85 benchmark circuits. The variability of process parameters (L, V_{th}, and T_{ox}) and the environmental fluctuation (V_{dd}) can be taken into account. The 3σ values for process parameters are set at 20% of the mean, with the die-to-die component contributing 50% of the total variance. The standard deviation of V_{dd} is 4% of the maximum, the mean is 96% of the maximum, and the range is 84-100% of the maximum value. In the experiments, V_{th}, T_{ox} and V_{dd} are modeled as probabilistic interval variables. The range of V_{th} and T_{ox} is 80-120% of the mean. Sensitivities of parameters are from SPICE simulations for a cell library of BPTM 0.13um technology [63].

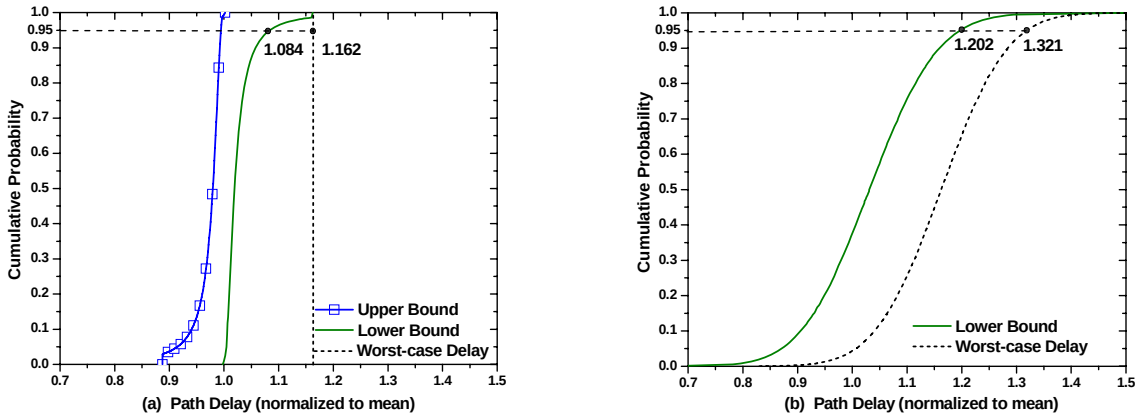

Figure 3.2: The path delay analysis algorithm improves the worst-case delay by 9.0% at the 95th percentile: a) delay due to probabilistic interval variables; b) total path delay.

The proposed timing analysis algorithms separately handle the contributions of the Gaussian uncertainty and the probabilistic interval uncertainty. Thus, the comparison of our algorithms and the worst-case timing analysis, i.e., only using the range (interval) of the probabilistic interval uncertainty, should be done in two steps. We first compare the bounds of D_{PI}^{j} computed by the proposed algorithm and the interval-based (worst-case) timing analysis, then compare the bound of the total delay, which is the sum of D_{PI}^{j} and D_{R}^{j}. Note that the sum of the bound from the worst-case timing analysis for probabilistic interval uncertainty and D_{R}^{j} can be computed by simply shifting the *cdf* of D_{R}^{j} by the interval-based (worst-case) delay value. A similar comparison is also made for the bounds on circuit delay distribution.

Figure 3.2(a) illustrates the importance of probabilistic interval analysis in path delay analysis. The upper bound of the 95th- percentile path delay (D_{PI}^{j}) from the proposed algorithm for the critical path of circuit c6288 is only 8.4% over the mean path delay, while the worst-case timing estimate is 16.2% over the mean. Therefore, the

proposed path timing analysis algorithm reduces the worst-case timing estimate by 6.7%. Similarly, the 95th-percentile total path delay ($D_R^j + D_{PI}^j$) is 20.2% over the mean for the proposed algorithm, which is superior to the worst-case delay (32.1% over the mean) in Figure 3.2(b). Thus, the proposed strategy improves the worst-case estimate by 9.0% for the overall path delay at the 95th percentile.

For circuit delay distribution, the fast robust Monte Carlo simulation (FRMC) has been run on a Sun workstation with 1280 MHz CPU and 8GB memory. We estimated the sample mean and the variance using 1000 samples, and then analytically computed the lower bound of the cumulative probability. The run time of the fast robust Monte Carlo ranges from 12 to 114 seconds. Figure 3.3 shows the circuit delay variation due to probabilistic interval variables of circuit c7552, from the proposed statistical strategy and the worst-case timing analysis. FRMC is able to provide a superior bound to the worst-case delay at lower than the 87th percentile.

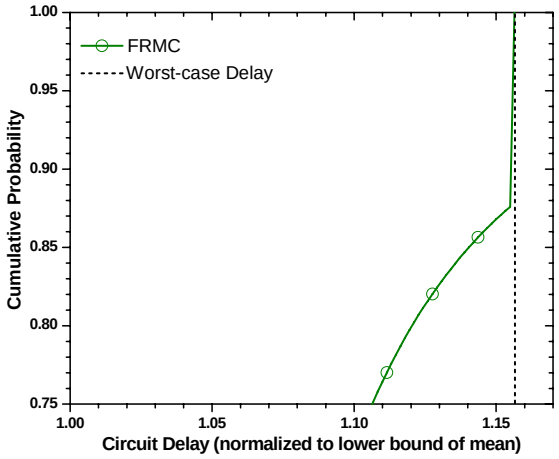

Figure 3.3: Upper bounds for circuit delay due to probabilistic interval variables for circuit c7552.

Table 3.1: Upper bounds for total circuit delay at the 90th and 95th percentiles and run time of fast robust Monte Carlo simulation.

Circuit	Number of Gates	Fast Robust Monte Carlo Simulation					Worst-case Delay	
		90th Percentile		95th Percentile		Run Time (s)	90th Percentile Delay (ps)	95th Percentile Delay (ps)
		Delay (ps)	Reduction (%)	Delay (ps)	Reduction (%)			
c880	456	2383	5.62	2467	4.97	12	2525	2596
c1355	605	2264	4.59	2335	4.26	18	2373	2439
c1908	975	2820	5.56	2919	4.89	26	2986	3069
c2670	1544	3124	5.65	3232	5.08	38	3311	3405
c3540	1787	4097	5.49	4237	4.94	52	4335	4457
c6288	2448	17547	5.28	18081	4.82	87	18526	18996
c5315	2600	3579	5.49	3703	4.88	79	3787	3893
c7552	3874	3136	4.88	3236	4.46	114	3297	3387

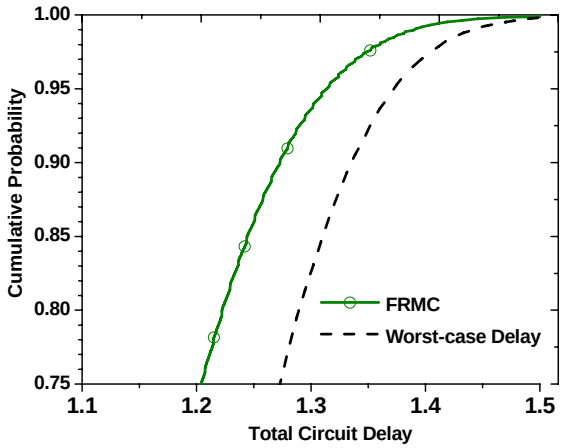

Figure 3.4: Upper bounds for the total circuit delay of c7552.

For the total circuit delay ($D_{\max}$), FRMC improves the estimates from the worst-case timing analysis by 5.3% and 4.8% across the benchmark circuits, for the 90th and 95th percentile delays, respectively. Table 3.1 shows the upper bound of the total circuit delay at the 90th and 95th percentiles for FRMC, and the worst-case timing analysis. Figure 3.4 shows an example of the total circuit delay for the circuit c7552, in which FRMC reduces the worst-case delay estimate by 4.5% at the 95th percentile. Indeed, the

joint use of SSTA and our statistical technique for probabilistic interval variables is a promising synergy, and it can be easily extended to incorporate more circuit parameters, to fully assess the impact on timing performance.

An important feature of the proposed technique is the capability of handling skewed distributions. Some environmental parameters are not symmetrically distributed (e.g., V_{dd}); however, the normal assumption implies the distribution is symmetrical to the mean, which may cause inaccurate estimation of the circuit delay. Figure 3.5(a) compares path delay distributions of two cases with the same interval and variance of V_{dd} uncertainty: the right-skewed V_{dd} uncertainty and the symmetrical case. Because the voltage drop increases gate delays, the right-skewed V_{dd} uncertainty decreases the upper bound of delays, compared to the symmetrical V_{dd} distribution. From Figure 3.5(b), the same trend can be observed in the distribution of the total circuit delay. Thus, our timing analysis algorithm can be used to handle asymmetrical distributions (e.g., non-Gaussian), and provide a more accurate timing estimate.

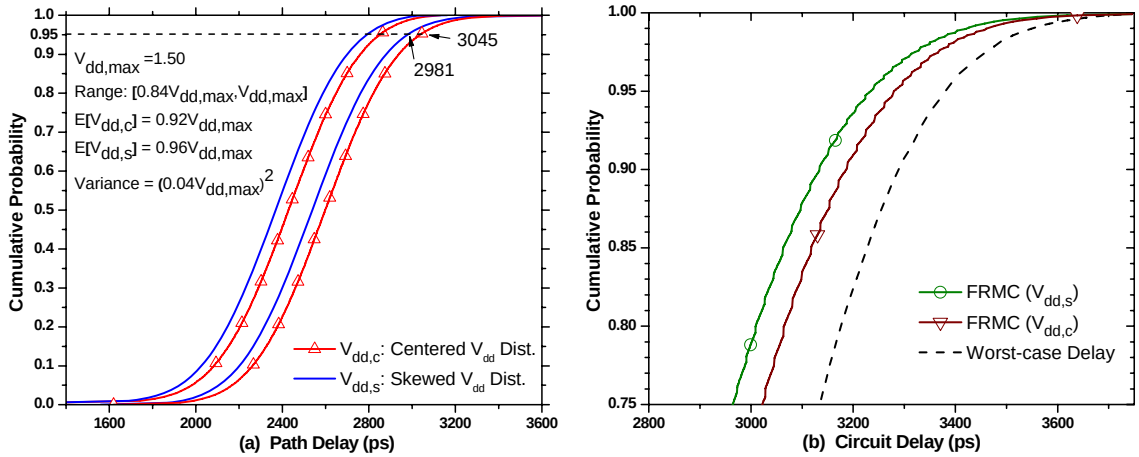

Figure 3.5: The right-skewed V_{dd} distribution decreases bounds: a) path delay; b) circuit delay of the symmetrical V_{dd} distribution.

3.4 SUMMARY

This chapter develops a set of statistical techniques for estimating path and circuit delay distributions. Given partial statistical metrics of the uncertainty, the proposed algorithm is able to analytically compute the bounds of the path delay. Besides, a fast robust Monte Carlo simulation technique is proposed to assess the impact of the uncertainty, and estimate probabilistic bounds for the circuit delay. With the justified selection of the distribution used in the simulation, the proposed technique can efficiently construct a guaranteed bound of the circuit delay distribution.

Chapter 4: Estimation of Leakage Power Dissipation and Parametric Yield Based on Realistic Probabilistic Descriptions of Parameters

The substantial increase of leakage power dissipation is one of the most important concerns in recent technology nodes. As device geometries continue to shrink, power supply voltage also scales to contain the increase of power dissipation. Scaling of supply voltage, however, also necessitates the reduction of threshold voltage to improve the gate delay performance, thus resulting in the rapid growth of subthreshold leakage current [64]. Meanwhile, scaling of the vertical dimension of transistors also increases the gate tunneling leakage current; in the nanometer regime the gate tunneling leakage through the insulating oxide layer becomes comparable to the subthreshold leakage current [64], [65]. Because both gate and subthreshold leakage are extremely sensitive to variability of process and environmental parameters, quantifying the impact of variability on leakage is important for circuit designers.

Parametric yield, which is the percentage of the manufactured chips meeting the performance constraints, is traditionally determined by circuit timing. Nowadays the tremendous growth of leakage power also becomes a limiting factor. Techniques for joint timing- and power-limited parametric yield estimation [38], [66], [67] relied on the fact that yield is limited both by leakage power consumption and chip frequency. Leakage power usually exhibits inverse correlations with chip frequency: slow dies have low leakage, while fast dies have high leakage. Several papers separately studied the statistical leakage estimation problem. A gate-level full-chip leakage analysis algorithm taking into account spatial correlations of within-die process variations was proposed in [39]. A probabilistic approach was proposed to estimate subthreshold leakage distribution

accounting for within-die and die-to-die variations of process parameters, temperature, and supply voltage [68]. All the above techniques, however, rely on idealized assumptions about variability of process and environmental parameters.

The practical application of statistical approaches has to account for the limited availability of parameter distribution. Chapter 3 considers this limitation imposed on timing analysis, and proposes a robust strategy for handling partial probabilistic information. This chapter copes with the similar problem in the context of statistical leakage power analysis and yield estimation. First we discuss several practical concerns about existing leakage estimation approaches, and describe the mathematical basis to address these concerns. We then propose a robust algorithm for the estimation of leakage and parametric yield to handle full and partial probabilistic descriptions of parameters. Experimental results show that the proposed algorithm permits reducing the over-conservatism of traditionally formulated worst-case analysis.

This chapter is organized as follows. Section 4.1 investigates several practical issues on leakage estimation, and Section 4.2 presents the mathematical formulation of probabilistic interval methods. Section 4.3 describes the full-chip leakage modeling and yield estimation, and Section 4.4 shows the experimental results. Finally, Section 4.5 summarizes this work.

4.1 PRACTICAL CONCERNS ON LEAKAGE ESTIMATION

4.1.1 Simplified Modeling of Leakage

An important concern for statistical leakage analysis algorithms is the reliance on idealized models of the leakage power dissipation. Working with idealized models often requires adopting unreasonable assumptions about the dependence on process parameters. In the literature, an empirical model of the following form was found to accurately model

leakage dependence on the key process parameters that are subject to substantial variability [69].

$$I_{sub} = I_o W \exp[a_1 + a_2 L + a_3 L^2 + a_4 T_{ox}^{-1} + a_5 T_{ox}] \tag{4.1}$$

where L is the effective channel length, and T_{ox} is the oxide thickness. Then, in order to handle the leakage current using analytical expressions, the distribution of I_{sub} is approximated as a log-normal distribution. It means that (4.1) is simplified into:

$$I_{sub} = I_{nom} W \exp[b_1 \Delta L + b_2 \Delta T_{ox}] \tag{4.2}$$

where I_{nom} is the nominal value of the subthreshold leakage current.

Given that the die-to-die variation of L is significant, and because of the exponential effect that an approximation will have on the result, this transformation may not be acceptable. The cost of not performing such a transformation is that I_{sub} is not characterized by a lognormal distribution, which poses difficulties for existing leakage analysis methods. We will solve this problem by adopting self-verifying robust methods of estimation. Besides, the current process technology is well-controlled; the outlier of parameters is unlikely to exist in fabricated chips. Therefore, truncated distributions are more appropriate to represent process variability because it avoids the erroneous estimation resulting from the infinite tails of Gaussian variables.

4.1.2 Idealized Modeling of Process Parameters

In some cases, the real-life distributions of parameters may exhibit non-Gaussian, mixture [70], or multi-modal behavior, i.e., the probability density function has multiple peaks. Algorithms based on parametric techniques are notoriously poor at handling these distributions; the distributions can only be approximated as parametric distributions (e.g., normal) for convenient manipulations. However, the robust estimation framework can naturally handle a variety of distributions including non-Gaussian, mixture, and multi-modal variables, which will be demonstrated in Section 4.2.

An extreme case for the mixture distribution is when a fabless company works with two or more foundries in manufacturing a design. The design must be robust under distinct fab-specific parameter distributions. The formal statistical model to analyze this case is finite mixture distribution. Suppose chips are fabricated by n (e.g., n=2) fabs. The probability density function of the effective channel length, $f(L)$, can be computed by:

$$f(L) = \sum_{i=1}^{n} f(L \mid F_i) \cdot p(F_i) \tag{4.3}$$

where $p(F_i)$ is the probability that a chip is fabricated in fab i, and $f(L \mid F_i)$ is the conditional probability density function for fab i. The mean (μ) and variance (σ^2) of the mixture distribution can be computed by:

$$\mu = \sum_{i=1}^{n} p(F_i)\mu_i \qquad \sigma^2 = \sum_{i=1}^{n} p(F_i)\left(\mu_i^2 + \sigma_i^2\right) - \mu^2 \tag{4.4}$$

where μ_i and σ_i^2 are the mean and variance of L from fab i.

Figure 4.1 shows the mixture of the effective channel length distributions due to statistically distinct populations, assuming each foundry contributes half of the manufactured chips. For the purpose of demonstration, process variability of each foundry follows the normal distribution. From Figure 4.1, approximating the mixture distribution as a single normal distribution causes inaccurate modeling of channel length variations, which overestimates the subthreshold leakage of a transistor by 13% and 48%, at the 95th and 99th percentile, respectively. In this example, although it is possible to use parametric techniques to separately estimate the leakage distribution for each foundry and then compute the distribution of all chips, the need for multiple estimations of leakage distribution would be cumbersome. In contrast, the proposed robust estimation strategy permits conveniently handling the mixture distribution, in a single flow of leakage analysis. Besides, some process parameters may follow non-Gaussian distributions, e.g.,

via resistance [71]. If non-Gaussian process parameters are taken into account, approximation is inevitably required for parametric techniques to manipulate non-Gaussian variables. Thus, this example points out the limitation of the analytical approaches based on parametric techniques and the idealized assumptions about parameter variability.

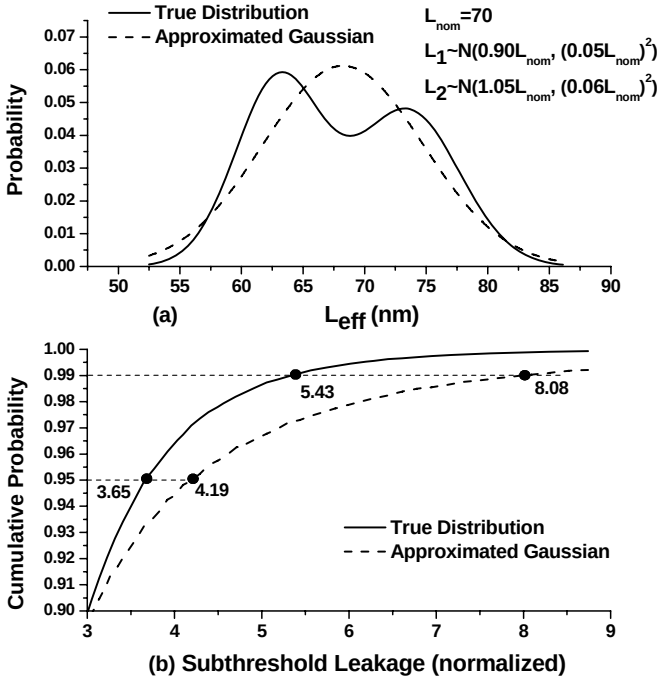

Figure 4.1: Approximating uncertainty as a Gaussian variable may lead to a large error in leakage estimation: a) channel length distribution; and, b) subthreshold leakage distribution.

4.1.3 Strategy of Handling Limited Probabilistic Descriptions

Taking into account all the concerns described above, in this chapter we address the limitations of the existing methods by introducing a new mathematical formulation that enables robust prediction of timing- and power-limited parametric yield. The fundamental theory behind the work is probabilistic interval analysis that extends the representation of a random variable to a family of distributions, i.e., bounds for

cumulative distribution functions, and thus can work with a wider class of uncertainty models. This strategy of handling the partially-specified uncertainty has been applied to timing analysis, as described in Chapter 3. It effectively reduces the over-conservatism of the interval-based prediction. In this chapter, we further exploit other advances in probabilistic interval analysis for convenient and robust manipulations of partially-specified random variables.

The proposed algorithm is based on non-parametric robust statistics, which permits using statistical metrics (e.g., the mean and variance) to describe the partially-specified environmental fluctuation in chips. Arithmetic operations based on linear programming [40] are used to compute probabilistic bounds for functions of random variables. This strategy, along with realistic modeling of process variability, is able to assess the impact of process and environmental variations on leakage dissipation and parametric yield. To evaluate the effectiveness of the proposed estimation approach, we compare the leakage estimates that are based on the partial probabilistic descriptions, with those based on the interval information. The experiments compare the chip-level parametric yield and leakage dissipation estimated based on: 1) fully-specified process parameters and partial probabilistic descriptions of environmental parameters, and 2) fully-specified process parameters and intervals of environmental parameters. The results indicate that through the application of the proposed algorithm the conservatism is reduced by 5-13% for the total leakage dissipation at the 99^{th} percentile across frequency bins. Thus, the proposed algorithm can utilize the partial probabilistic descriptions to effectively reduce the conservatism caused by interval-based analysis.

4.2 PRINCIPLES OF ROBUST COMPUTATION OF RANDOM VARIABLES

This section presents a formal description of the robust estimation procedure. Its purpose is to enable *reliable* and *assumption-free* generation of distributions of functions of random variables. The adopted framework can be seen as a probabilistic interval method. It supplements the estimates of intervals and affine methods with the partial probabilistic information enabling a new type of analysis. The framework requires the development of two distinct sets of mathematical tools for robust representation of random variables, and robust operations with random variables.

4.2.1 Robust Representations of Random Variables

In a robust estimation framework, we need an appropriate representation of uncertainty. A general representation for a random variable is a p-box [26].

Definition: $\overline{F}$ and $\underline{F}$ are non-decreasing functions from $\Re$ into [0, 1], and $\underline{F}(x) \le \overline{F}(x), \forall x \in \Re$. A p-box, denoted by $[\underline{F}, \overline{F}]$, is defined as a set of imprecisely known cumulative distribution functions, $F(x) = P(X \le x)$, where $\underline{F}(x) \le F(x) \le \overline{F}(x), \forall x \in \Re$ [26].

A p-box represents upper and lower bounds for the cumulative distribution function of a random variable. It is a basic notion for the robust computation, and can be used to robustly describe a random variable. Because the p-box representation is parametric-free, it can be used to describe a variety of distributions including the non-Gaussian, multi-modal, and mixture distributions. For instance, given the cumulative distribution function (i.e., the full probabilistic description) of a random variable, $F(x)$, we can sample it at a series of non-decreasing values, x_i, where $F(x_0) = 0$ and $F(x_n) = 1, 0 \le i \le n$. Then a p-box can be constructed as:

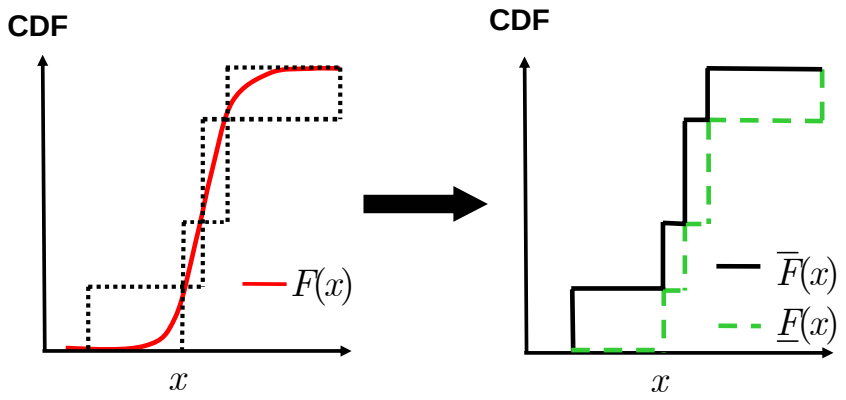

Figure 4.2: Construction of a p-box from the cumulative distribution function.

$$\begin{aligned} &\overline{F}(x) = F(x_{i+1}) \\ &\underline{F}(x) = F(x_i) \qquad\qquad x_i \le x < x_{i+1},\ i = 0...n-1. \\ &\overline{F}(x) = \underline{F}(x) = F(x) \quad x \ge x_n \text{ or } x < x_0 \end{aligned} \tag{4.5}$$

Figure 4.2 illustrate the construction of the p-box. Thus, the p-box representation allows us to incorporate realistic distributions of process variability into the estimation framework, instead of resorting to the idealized assumption of parameters.

Another useful description of uncertainty is in terms of intervals with *partial probabilistic information.* In addition to the ranges, often limited information, such as the mean or the variance, is available. In this situation, it seems wasteful not to use this partial information. From Section 4.1, these statistical metrics of environmental parameters can be estimated during the early design phase; therefore, the environmental parameters are modeled as *probabilistic intervals* in our framework.

The *probabilistic interval* description needs to be converted into a p-box representation for further manipulations. This is done using a sophisticated generalization of the one-sided Chebyshev inequality [43] and Cantelli inequality [26], [55] which enables computing bounds of the cumulative probability. Basically the Cantelli inequality

gives bounds for the cumulative probability of a *non-negative* random variable X with mean μ and variance σ^2. The upper bound of the probability is:

$$\begin{aligned} &P(X \le x) = 0 && x \le 0 \\ &P(X \le x) \le 1 \Big/ \left(1 + (\mu - x)^2 \Big/ \sigma^2\right) && 0 \le x \le \mu \\ &P(X \le x) = 1 && \mu < x \end{aligned} \tag{4.6}$$

Also, the lower bound of the probability is given by:

$$\begin{aligned} &P(X \le x) = 0 && x < \mu \\ &P(X \le x) \ge 1 - \mu / x && \mu \le x \le \mu + \sigma^2 \Big/ \mu \\ &P(X \le x) \ge 1 \Big/ (1 + \sigma^2 \Big/ (x - \mu)^2) && \mu + \sigma^2 \Big/ \mu < x \end{aligned} \tag{4.7}$$

This set of inequalities provides a p-box based on the mean, variance, and the lower bound (0). It can be generalized to handle real-valued random variables, and be combined with one-sided Chebyshev inequality [26]. Then it can compute the p-box if the mean, variance, and the interval of a random variable are known. This set of inequalities are introduced in (3.10) and (3.11). For convenience, we describe these inequalities again. The upper bound of the cumulative probability is:

$$\begin{aligned} &P(X \le x) = 0 && x < \underline{X} \\ &P(X \le x) \le 1 \Big/ (1 + (\mu - x)^2 \Big/ \sigma^2) && \underline{X} \le x < \mu + \sigma^2 \Big/ (\mu - \overline{X}) \\ &P(X \le x) \le 1 \Big/ (1 + (\mu - x)^2 \Big/ \sigma^2) && \mu + \sigma^2 \Big/ (\mu - \overline{X}) \le x < \mu + \sigma^2 \Big/ (\mu - \underline{X}) \\ &P(X \le x) \le 1 - (m^2 - my + s^2) \Big/ (1 - y) && \mu + \sigma^2 \Big/ (\mu - \underline{X}) \le x \end{aligned} \tag{4.8}$$

where μ denotes the mean, σ^2 denotes the variance, $\underline{X}$ is the lower bound, $\overline{X}$ is the upper bound, $y = (x - \underline{X}) \Big/ (\overline{X} - \underline{X})$, $m = (\mu - \underline{X}) \Big/ (\overline{X} - \underline{X})$, and $s^2 = \sigma^2 \Big/ (\overline{X} - \underline{X})^2$. Similarly, the lower bound of the cumulative probability is:

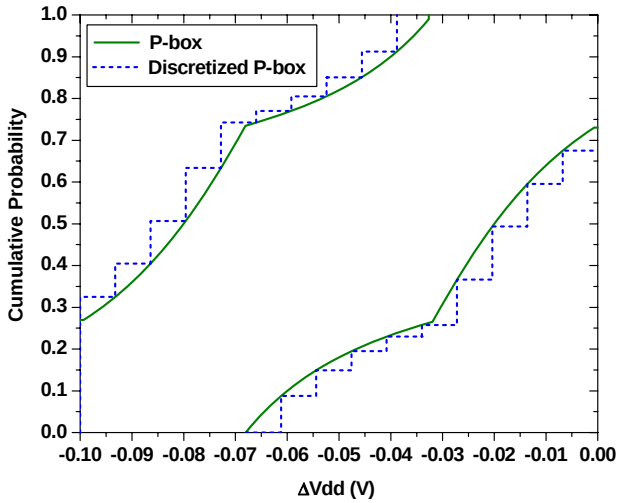

Figure 4.3: The knowledge of range, mean and variance permits constructing a p-box for a variable.

$$\begin{aligned}
&P(X \le x) = 0 && x < \mu + \sigma^2/(\mu - \overline{X}) \\
&P(X \le x) \ge 1 - (m(1+y) - s^2 - m^2)/y && \mu + \sigma^2/(\mu - \overline{X}) \le x < \mu + \sigma^2/(\mu - \underline{X}) \\
&P(X \le x) \ge 1/(1 + \sigma^2/(x-\mu)^2) && \mu + \sigma^2/(\mu - \underline{X}) \le x < \overline{X} \\
&P(X \le x) = 1 && \overline{X} \le x
\end{aligned} \tag{4.9}$$

For example, when the partial probabilistic description of the supply voltage is available, its p-box can be constructed by (4.8) and (4.9). Figure 4.3 shows an example: the mean and the variance values of the variable are -0.05 and $(0.03)^2$, respectively. Besides, the upper and lower bounds for the variable are 0.0 and -0.10. This p-box in fact represents all distributions with the same mean, variance, and range, and it permits estimating the uncertainty at any confidence level. For example, in Figure 4.3 when the cumulative probability is 0.50, the right-side p-box falls at -0.02V, which means at least 50% of the samples have ΔV_{dd} less than or equal to -0.02V, i.e., $P(\Delta V_{dd} \le -0.02\text{V}) \ge 0.50$. Thus, we can easily estimate the percentage of the samples meeting a specific requirement using p-boxes. Compared to interval-based analysis, in

which only lower and upper bounds are specified, analysis approaches based on probabilistic intervals and p-boxes can utilize the partial probabilistic descriptions, and provide less conservative estimates.

The p-box representations described above are useful for describing process and environmental parameters. The robust estimation framework seeks to perform numerical operations on the p-box descriptions of variables, and provides guaranteed computational results (e.g., total leakage dissipation) that are also described by p-boxes. In order to implement computation with p-boxes, an intermediate and numerically tractable representation is needed during robust computations. This is based on the notion of *self-validating histograms* [40].

Definition: A self-validating histogram of a random variable X is:

$$X = \bigcup_i X_i \ , \quad X_i = \left[\underline{X_i}, \overline{X_i}\right].$$
$$P\left(X \in X_i\right) = p_i \quad \text{for all } i, \text{ and } \sum_i p_i = 1$$

where X_i is an interval associated with the probability p_i.

This histogram representation describes a random variable as a set of intervals associated with probabilities, as in Figure 4.4. As a *two-valued* histogram, in which lower and upper endpoints of the intervals are recorded, the histogram is self-validating because it is able to keep track of the accuracy (error) of the computed quantities [72]. Besides, the intervals, X_i, may overlap, which provides great flexibility of describing random variables. Before numerical computations of p-boxes, we need to transform p-boxes into histograms [73], which will be clear in Section 4.2.2. This transformation requires two phases: first, the p-box needs to be conservatively discretized as in Figure 4.3, which means that the discretized p-box forms an envelope of the original p-box. The discretization can be done with arbitrary granularity depending on the required accuracy.

Then the discretized p-box is transformed into the two-valued histogram, as shown in Figure 4.5. Having transformed random variables into the self-validating histograms, we can then perform arithmetic operations on the histograms.

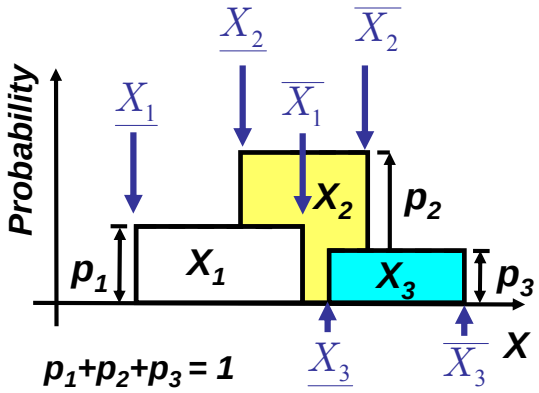

Figure 4.4: An illustration of the self-validating histogram.

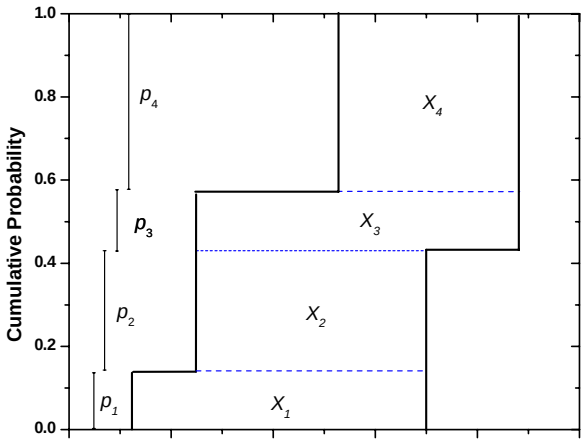

Figure 4.5: Transformation of a discretized p-box into a histogram representation.

4.2.2 Robust Operations on Variables

Various arithmetic operations can be performed on variables described by self-validating histograms, establishing the general probabilistic arithmetic, which permits computing the p-box for functions of random variables. In this section, we demonstrate the arithmetic operations on single, and multiple random variables described by self-validating histograms.

First we describe how to evaluate functions of a random variable *X*, given the histogram representation. This computation can be done by creating a table including all intervals, as in Figure 4.6. For each interval of *X*, we compute upper and lower bounds of the function $Z = f(X)$ when $X \in [\underline{X_i}, \overline{X_i}]$. The bounds are:

$$\underline{Z_i} = \min f\left(X \in [\underline{X_i}, \overline{X_i}]\right),\ \overline{Z_i} = \max f\left(X \in [\underline{X_i}, \overline{X_i}]\right) \tag{4.10}$$

where $\underline{Z_i}$ and $\overline{Z_i}$ denotes the lower and upper bounds of the function, respectively, when $X \in [\underline{X_i}, \overline{X_i}]$.

In the constructed probability table, the probability of the derived interval $\left(Z \in [\underline{Z_i}, \overline{Z_i}]\right)$ is equal to the probability of the original interval, i.e., $P(Z \in [\underline{Z_i}, \overline{Z_i}]) = P(X \in [\underline{X_i}, \overline{X_i}])$. Then a histogram of $Z = f(X)$ can be constructed. The final step is to compute the p-box of *Z*. The cumulative probability of the function at a specific value z, $P(Z \le z)$, is bounded by:

$\boldsymbol{X}$	$\boldsymbol{Z = f(X)}$
⋮	⋮
$X \in [\underline{X_i}, \overline{X_i}]$	$Z \in [\underline{Z_i}, \overline{Z_i}],\ p_i = P(Z \in [\underline{Z_i}, \overline{Z_i}])$
⋮	⋮

Figure 4.6: Probability table for a function of a random variable.

$$\begin{aligned} P(Z \le z) &\le \sum_i p_i \quad \forall i : \underline{Z_i} \le z \\ P(Z \le z) &\ge \sum_i p_i \quad \forall i : \overline{Z_i} \le z \end{aligned} \tag{4.11}$$

That is, when computing the upper bound of $P(Z \le z)$, the intervals with the lower bound $\underline{Z_i}$ no larger than *z* need to be considered. In the histogram representation, the probability mass of an interval is specified, i.e., $p_i = P(Z \in [\underline{Z_i}, \overline{Z_i}])$; however, how the probability mass is distributed within the interval is not constrained. When the probability mass p_i of an interval entirely falls at the lower bound of the interval $\underline{Z_i}$, i.e., $p_i = P(Z = \underline{Z_i})$, it results in the largest increase in the *cdf*, which determines the upper bound of the *cdf*. Similarly, the lower bound of the *cdf* can be determined when the probability falls at the upper bound of the interval. As a result, we can use (4.11) to compute the p-box of *f*(*X*). Besides, the bounds for statistical metrics can be also computed from the histogram representation. For instance, the bounds for the expected value of *Z* are given by:

$$\underline{E}[Z] = \sum_i p_i \underline{Z_i} \ , \quad \overline{E}[Z] = \sum_i p_i \overline{Z_i} \tag{4.12}$$

where $\underline{E}[Z]$ and $\overline{E}[Z]$ denote the lower and upper bounds for $E[Z]$.

For operations on multiple variables, the computation of p-boxes is an optimization problem because the distribution of the result depends on the correlation of variables. The operation of multiple random variables described by histograms requires solving a linear optimization problem [40]. We briefly describe how to compute the probabilistic bound for a function of two variables, *X* and *Y*, with unknown dependency.

$\boldsymbol{Z = f(X,Y)}$	…	$Y \in [\underline{Y_j}, \overline{Y_j}]$	…
⋮	⋱	⋮	⋰
$X \in [\underline{X_i}, \overline{X_i}]$	…	$Z \in [\underline{Z_{ij}}, \overline{Z_{ij}}],\ p_{ij} = P(Z \in [\underline{Z_{ij}}, \overline{Z_{ij}}])$	…
⋮	⋰	…	⋱

Figure 4.7: Probability table for a function of multiple random variables.

First a discrete two-dimensional table is constructed to include all combinations of intervals from *X* and *Y*. For each cell we compute the bounds of the function, $f(X,Y)$.

$$\begin{aligned} \underline{Z_{ij}} &= \min f\left(X \in \left[\underline{X_i}, \overline{X_i}\right], Y \in \left[\underline{Y_j}, \overline{Y_j}\right]\right), \text{ and} \\ \overline{Z_{ij}} &= \max f\left(X \in \left[\underline{X_i}, \overline{X_i}\right], Y \in \left[\underline{Y_j}, \overline{Y_j}\right]\right). \end{aligned} \tag{4.13}$$

Then we assign a variable, p_{ij}, as the probability mass of the cell. It represents the probability that the variable *Z* falls within the interval $[\underline{Z_{ij}}, \overline{Z_{ij}}]$:

$$p_{ij} = P(Z \in [\underline{Z_{ij}}, \overline{Z_{ij}}]) \tag{4.14}$$

The constructed table is shown in Figure 4.7. Similar to (4.11), the cumulative probability of the function, $P(Z \le z)$, is bounded by:

$$\begin{aligned} P(Z \le z) &\le \sum\nolimits_{i,j} p_{ij} \quad \forall i,j : \underline{Z_{ij}} \le z \\ P(Z \le z) &\ge \sum\nolimits_{i,j} p_{ij} \quad \forall i,j : \overline{Z_{ij}} \le z \end{aligned} \tag{4.15}$$

The expressions in (4.15) can be used as the objective functions of the optimization problem. Now we describe the constraints of the problem. First the cell probability should be nonnegative. Second, the sum of probabilities of cells in the same row (column) should be equal to the marginal probability of *X* (*Y*). Suppose we consider the i^{th} row in the probability table, the cells in this row includes all intervals of *Y*. Thus, the sum of the cell probabilities only depends on *X*:

$$\begin{aligned}
&\sum_j p_{ij} \\
&= \sum_j P\left(X \in \left[\underline{X_i}, \overline{X_i}\right], Y \in \left[\underline{Y_j}, \overline{Y_j}\right]\right) \\
&= P\left(X \in \left[\underline{X_i}, \overline{X_i}\right]\right)
\end{aligned} \tag{4.16}$$

For each row and column in the probability table we can derive a constraint for the variable, p_{ij}. The p-box of the multivariate function can be then computed by solving the optimization problems below for distinct values of z.

(i) The upper bound of the cumulative probability:

$$\max \quad \sum_{i,j} p_{ij} \quad \forall i, j : \underline{Z_{ij}} \le z \tag{4.17}$$

(ii) The lower bound of the cumulative probability:

$$\min \quad \sum_{i,j} p_{ij} \quad \forall i, j : \overline{Z_{ij}} \le z \tag{4.18}$$

The constraints of the optimization problems are:

$$\begin{aligned}
\sum_i p_{ij} &= P(Y \in [\underline{Y_j}, \overline{Y_j}]) \quad \text{for all } j. \\
\sum_j p_{ij} &= P(X \in [\underline{X_i}, \overline{X_i}]) \quad \text{for all } i. \\
p_{ij} &\ge 0 \quad \text{for all } i, j.
\end{aligned} \tag{4.19}$$

Since the objective function and all constraints are linear functions of the cell probability, p_{ij}, the optimization problem is a linear programming problem that can be solved efficiently. Besides, this probabilistic arithmetic can be extended to handle multiple variables by manipulating a multi-dimensional probability table. Consequently, we are able to compute the p-box of any arbitrary function of variables described by the histogram representations.

The complexity of the above p-box computation based on linear programming is determined by several parameters of the problem, including the number of intervals in a histogram, and the number of the observed points in the *Z* domain. If the histograms of *X*

and Y both include N intervals, then there exist N rows and N columns in the probability table, resulting in $2N$ linear constraints (the nonnegative constraints of cell probabilities are not included). Suppose the optimization problem is solved using the interior-point method [58], the bound for computation complexity is $O\left(n^{3.5}L\right)$, where n is the number of inequalities (constraints), and L is the number of bits in the binary representation for expressing variables [76]. Thus, the optimization problem can be solved in polynomial time of the number of intervals in the histogram, i.e., N. If we compute the p-box at M points in the Z domain, then the bound for the complexity is $O\left(MN^{3.5}L\right)$.

The robust computation described above requires solving linear optimization problems to compute p-boxes of the computation result. A special case for operations on random variables is that variables are mutually independent. In this situation, the result of arithmetic operations can be efficiently evaluated without solving the optimization problem. For example, if we consider two independent random variables X and Y, the joint table is a two-dimensional grid, in which the probability of the entries is generated by:

$$P(X \in X_i, Y \in Y_j) = P(X \in X_i)P(Y \in Y_j) \tag{4.20}$$

Once the associated probability of each cell is computed, we are able to construct the histogram of any function of probabilistic interval variables. With the histogram, the probability bound can be evaluated using (4.15).

4.3 Robust Estimation of Chip Leakage Power and Parametric Yield

Reliable estimation of chip leakage power and parametric yield is extremely important for chip designers. The two-sided squeeze on yield means that yield estimation essentially requires reliable frequency and leakage prediction. We apply the robust estimation framework developed above to the problem of reliably evaluating the chip-

level leakage power and parametric yield. Since the input of the chip-level problem is relatively small, the computational cost is not a concern. All factors that affect the robustness of leakage estimation, described in Section 4.1, are included in the analysis.

Frequency binning is a commonly used scheme for yield estimation [38], [75]. First the manufactured chips are classified into groups (bins) according to the maximum allowable operating frequency. Because of the correlation between leakage and frequency, distinct characteristics of leakage power dissipation can be observed across bins. Thus, yield estimation is done for individual bins. Frequency binning is convenient for yield estimation because the binning itself has resolved the problem of computing timing-limited yield. For chips in the same bin, we only need to consider the variability in power. Thus, this work also follows the frequency binning scheme; we estimate the power-limited yield for chips in each bin, and then compute the leakage power distribution for all chips.

4.3.1 Parametric Yield Estimation for Frequency-Binning Scheme

Prior work has shown that the chip frequency is largely determined by the die-to-die channel length variation [38]. Thus, for a specific frequency bin, we can assume that the die-to-die channel length variation is a fixed value, which simplifies the derivation.

The variability of three process parameters is considered in our analysis: effective channel length (L), threshold voltage (V_{th}), and oxide thickness (T_{ox}). The subthreshold leakage current models adopted here, describes it as an exponential function of the effective channel length, subthreshold voltage, and power supply voltage (V_{dd}). The dependency to on-chip temperature (T) is super-linear [74]; however, the leakage can be well approximated as an exponential function according to SPICE simulations. Therefore, similar to prior work [38], [68], the subthreshold leakage current of a unit-width transistor, is modeled as:

$$I_{sub} = I_{sub,nom} \exp\left(a\Delta L^2 + b\Delta L + c\Delta V_{th} + d\Delta V_{dd} + e\Delta T\right) \tag{4.21}$$

where $I_{sub,nom}$ is the nominal value of the subthreshold leakage current. This model can effectively describe the quadratic dependency of the exponent on L variability.

The variability of process parameters can be then decomposed as the sum of the within-die and the die-to-die components of variation.

$$\begin{aligned} I_{sub}(i) = I_{sub,nom} &\exp(a\Delta L_g^{\,2} + b\Delta L_g + c\Delta V_{th,g}) \\ &\times \exp[a\Delta L_l(i)^2 + \left(2a\Delta L_g + b\right)\Delta L_l(i) + c\Delta V_{th,l}(i)] \\ &\times \exp\left(d\Delta V_{dd}(i) + e\Delta T(i)\right) \end{aligned} \tag{4.22}$$

where $\left(\Delta L_l(i), \Delta V_{th,l}(i)\right)$ and $(\Delta L_g, \Delta V_{th,g})$ denote within-die (local) and die-to-die (global) components. For environmental parameter we focus on the within-die components of variation because the die-to-die variation is typically negligible.

Now we can compute the cumulative probability of I_{sub}. In this framework, environmental fluctuations (V_{dd} and T) are represented by probabilistic intervals, described in Section 4.2.1. For the purpose of demonstration, process variability (L and V_{th}) is modeled as a truncated Gaussian variable; however, our robust estimation is able to handle various distributions because fully specified uncertainty can be described by the p-box representation.

The variability due to distinct categories of parameters is assumed to be independent, as in the previous work [38], [66]. We can then construct a multidimensional table for I_{sub} using (4.20), and then compute the interval and the probability for each cell. Since I_{sub} is a monotonic function of all the parameters of interest, its range can be efficiently computed by considering the combinations of endpoints for parameters within each cell. Finally, we obtain a histogram representation of I_{sub}. The p-box representation can be then computed using (4.11).

The total subthreshold leakage can be computed by summing up the contribution of all transistors on the chip:

$$I_{sub,total} = \sum_i \left[W_i I_{sub,nom} \exp\left[a\Delta L_l(i)^2 + (2a\Delta L_g + b)\Delta L_l(i) + c\Delta V_{th,l}(i) + d\Delta V_{dd}(i) + e\Delta T(i) \right] \right] \times \exp(a\Delta L_g^2 + b\Delta L_g + c\Delta V_{th,g}) \quad (4.23)$$

where W_i denotes the equivalent device width accounting for complex gates and stack effects [64] of leakage. Because all transistors share the die-to-die component of variation, we can assess the impact of the within-die variation first. Assuming the within-die variability of a specific parameter for all devices is independently and identically distributed, and the based on the Central Limit Theorem [43] we can approximate the sum of the subthreshold leakage using the mean, as in [38], [66]:

$$I_{sub,total} \approx I_{sub,nom} \lambda_{sub} \sum_i W_i \times \exp(a\Delta L_g^2 + b\Delta L_g + c\Delta V_{th,g}) \quad (4.24)$$

where $\lambda_{sub} = E\left[\exp\left(a\Delta L_l^2 + \left(2a\Delta L_g + b\right)\Delta L_l + c\Delta V_{th,l} + d\Delta V_{dd} + e\Delta T \right) \right]$. The computation of the within-die factor λ_{sub} can be done by creating a multidimensional probability table and a histogram, which accounts for the uncertainty of $\left(\Delta L_l, \Delta V_{th,l}, \Delta V_{dd}, \Delta T\right)$. The lower (upper) bound for the mean of the histogram representation can be computed using (4.12). Thus, we can obtain the bounds of λ_{sub}.

In a chip-level analysis, the impact of environmental parameters can be evaluated using the statistical metrics (e.g., mean and variance) across the entire chip. If the statistical metrics of individual blocks in the chip design are available, our robust estimation framework can utilize these block-specific descriptions, and provide accurate leakage estimates. For example, a chip design based on the voltage island paradigm [77] may have distinct profiles of environmental parameters for blocks. Suppose the block-level statistical metrics of environmental parameters are available, our robust estimation

framework is able to evaluate $\lambda_{sub}\sum_i W_i$ for each block, and sum up this term for all blocks on the chip. After evaluating the within-die factor and the equivalent width in (4.24), we can then construct a histogram for $\Delta V_{th,g}$, compute the histogram of $I_{sub,total}$, and obtain the p-box of the total subthreshold leakage current.

Similarly, we can compute bounds on the within-die gate tunneling leakage distribution over T_{ox} and V_{dd}. The total gate tunneling leakage current is expressed as:

$$I_{gate,total} \approx I_{gate,nom} \exp(h\Delta T_{ox,g}) E\left[\exp(h\Delta T_{ox,l} + k\Delta V_{dd})\right] \sum_i W_i \qquad (4.25)$$

The above model describes the dependency of gate leakage on T_{ox} and V_{dd}, and insensitivity to the on-chip temperature [78].

The final step of parametric yield evaluation is to find the distribution of the total leakage current which is the sum of the gate tunneling and subthreshold leakage sources. The previous method [38] has assumed independence of subthreshold and gate leakage power. In our model, however, these sources of leakage power are correlated due to the dependency on the power supply voltage. As a result, the probability of the sum cannot be computed by convolution of individual probability density functions; this sum of leakage currents must be computed by the probabilistic arithmetic described in Section 4.2.2, which can handle correlated random variables.

For every fixed value of the die-to-die channel length variation (ΔL_g), we use the abovementioned algorithm to compute the p-box of the leakage current. Then for each frequency bin, we have the probabilistic bounds of leakage dissipation; also, given a specific power limit, we can compute the bounds for the parametric yield of the bin.

4.3.2 Leakage Power Dissipation of Entire Population

To compute the leakage distribution of all chips, we need to take into account chips across all frequency bins. Thus, the die-to-die channel length ΔL_g is no longer a

fixed value; it should be described by a histogram representation. Suppose the full probabilistic description of ΔL_g is available, we can construct a histogram for ΔL_g in which all intervals are disjoint.

$$\begin{aligned} \Delta L_g &= \cup_j \Delta L_{g,j}, \text{and} \\ \Delta L_{g,j} &= \left[\underline{\Delta L_g}(j), \overline{\Delta L_g}(j)\right]. \end{aligned} \tag{4.26}$$

where $\underline{\Delta L_g}(j)$ and $\overline{\Delta L_g}(j)$ are endpoints of the *j*-th interval, $\Delta L_{g,j}$, of the histogram.

For each interval ($\Delta L_{g,j}$), we use the robust leakage algorithm to compute the histogram representing the total leakage dissipation. The only difference is that $\Delta L_{g,j}$ is an interval when computing the subthreshold gate leakage current. After we obtain the histogram representing the total leakage current for each interval, the cumulative probability of the leakage dissipation for all intervals can be computed by:

$$\begin{aligned} &P\left(I_{total} \le i\right) \\ &= \sum\nolimits_j \left[P\left(I_{total} \le i \mid \Delta L_g \in \left[\underline{\Delta L_g}(j), \overline{\Delta L_g}(j)\right]\right) \times P\left(\Delta L_g \in \left[\underline{\Delta L_g}(j), \overline{\Delta L_g}(j)\right]\right)\right] \end{aligned} \tag{4.27}$$

This is based on that all intervals of ΔL_g are disjoint. Finally, the p-box of the leakage distribution for all chips can be computed.

4.4 EXPERIMENTAL RESULTS

The robust nature of the proposed strategy allows us to compute bounds for the cumulative probability of the leakage current, instead of an approximated value. Since our primary objective is to estimate the guaranteed chip leakage dissipation and parametric yield, we focus on the lower bound of the cumulative probability, i.e., the upper bound of the leakage power at any percentile. Thus, only the right-side p-box is shown in the figures. In the experiments, coefficients of parameters in subthreshold and gate leakage modeling are obtained from SPICE simulations for devices of PTM 70nm

technology [63], [79]. The 3σ values of L, V_{th}, and T_{ox} parameters are 20%, 10%, and 8% of the nominal values, respectively, with 50% of the variance contributed by the die-to-die components. Modeled as fully-specified random variables, these process parameters follow truncated Gaussian distribution, with the truncation occurring at μ±3σ values. Besides, distinct categories of parameters (e.g., L and V_{th}) are assumed to be mutually independent. The target chip is divided into 16 blocks with distinct ranges of environmental parameters; the maximum voltage drop is about 10-12% of the nominal value, and the standard deviation is about 3%. The range of on-chip temperature spans about 20°C, with the standard deviation about 3°C.

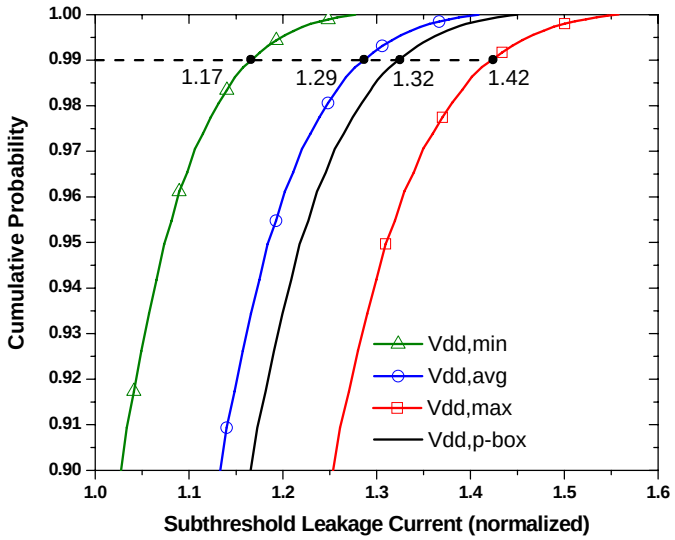

Figure 4.8: Total subthreshold leakage considering process variability (L and V_{th}) and V_{dd} uncertainty ($\Delta L_g = 0$).

To evaluate the effectiveness of the proposed strategy, in the experiments the chip-level parametric yield and leakage are estimated based on: 1) fully-specified process parameters and limited probabilistic descriptions of environmental parameters, and 2) fully-specified process parameters and intervals of partially-specified environmental parameters. In other words, we assess the improvement of using p-boxes over intervals for partially-specified parameters.

In Figure 4.8 we show the estimated subthreshold leakage dissipation for a specific bin, which illustrates the importance of using robust modeling for power supply voltage. We compute the lower bound for the *cdf* of the subthreshold leakage for a frequency bin, in which chips have zero inter-chip channel length variation $(\Delta L_g = 0)$. We also compute the probabilistic bounds assuming V_{dd} are described by intervals, or fixed values. The intervals of V_{dd} and average V_{dd} value of blocks are used. From the experiments the robust estimation strategy predicts that the total subthreshold leakage is 1.32X of the nominal value at the 99th percentile, which means in this specific frequency bin at least 99% of the samples have leakage current no larger than 1.32X of the nominal case, i.e., $P(I_{sub,total} \leq 1.32 I_{sub,total,nom}) \geq 0.99$. In contrast, the leakage estimates based on the maximum V_{dd} is 1.42X at the 99th percentile. Thus, the robust strategy improves the conservatism of the leakage estimate by 7.0%, compared to using the maximum V_{dd}. Similarly, the robust strategy reduces the conservatism of the total tunneling leakage consumption by 15.5% at the 99th percentile (Figure 4.9). Consequently, the robust method can utilize the partial probabilistic descriptions to enhance pure interval analysis of environmental parameters. Besides, Figure 4.8 illustrates that it is inappropriate to use the average value of power supply voltage because it predicts a lower estimate of leakage consumption, thus results in an over-optimistic prediction of the parametric yield.

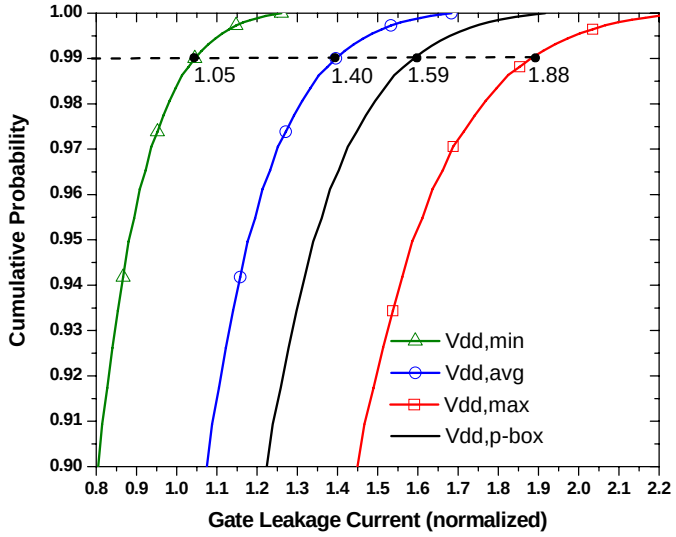

Figure 4.9: Total gate tunneling leakage considering process variability (T_{ox}) and V_{dd} uncertainty.

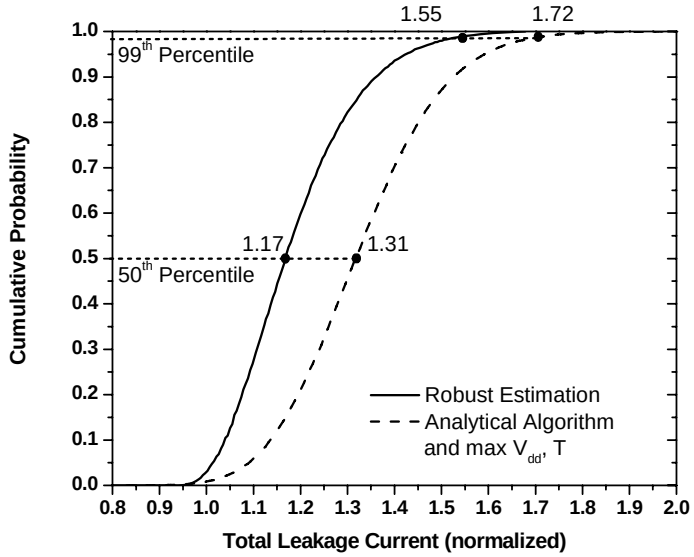

Figure 4.10: Total leakage current for a specific bin ($\Delta L_g = 0$).

We also compare our robust methodology with prior work [38] on leakage analysis. Because the algorithm in [38] does not take into account supply voltage and on-chip temperature, these parameters are assumed to be fixed values. Now the maximum values of supply voltage and temperature are used in the algorithm described in [38] because the objective is to provide a guaranteed estimation of the parametric yield. Figure 4.10 shows the total leakage dissipation computed by both approaches for a specific frequency bin $(\Delta L_g = 0)$. Compared to the algorithm in [38], our estimation approach provides less conservative leakage estimates at any percentile due to robust modeling of environmental parameters; the robust strategy reduces the conservatism of the leakage dissipation by 11.0% and 9.5% at the 50^{th} and 99^{th} percentiles, respectively. Besides, the robust estimation method predicts a higher parametric yield for a given limit of leakage current. Figure 4.11 shows the equi-yield contours for total leakage dissipation computed by both approaches. As for the 99^{th}-percentile leakage consumption, the difference ranges from 5.3% to 13.4%, within the $\pm 3\sigma$ range of the die-to-die channel length variation, as shown in Table 4.1. Note that the difference between the contours of the same yield becomes pronounced for chips with the negative die-to-die channel length variation. Therefore, for chips with large leakage currents, the robust approach can provide a more accurate estimation of parametric yield. It helps designers save extra efforts to apply additional leakage reduction techniques, and validates the necessity of adopting a robust estimation approach.

Table 4.1: Normalized leakage current at the 99^{th} percentile.

Die-to-die *L* variation (σ_{Lg})	-3	-2	-1	0	1	2	3
Algorithm in [38] with intervals of V_{dd} and *T*	4.04	2.94	2.21	1.72	1.39	1.16	1.00
Robust modeling of V_{dd} and *T*	3.50	2.57	1.97	1.55	1.27	1.08	0.94
Reduction of conservatism (%)	13.4	12.6	10.5	9.5	8.1	6.9	5.3

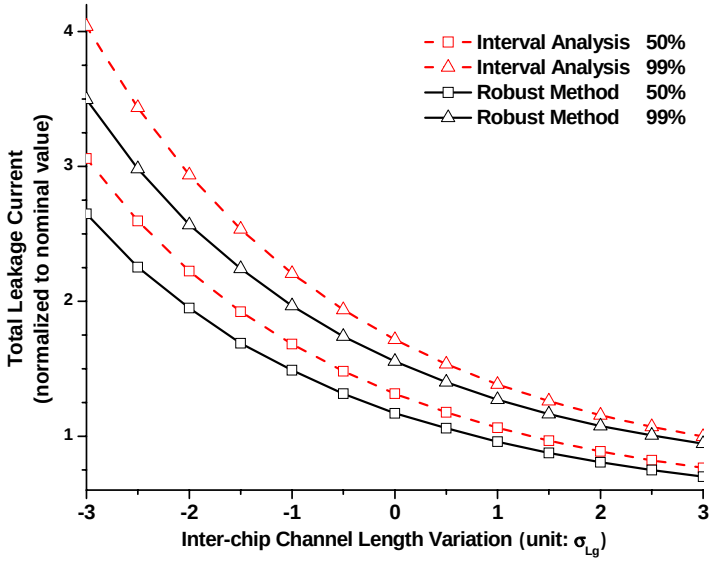

Figure 4.11: Equi-yield contours across bins.

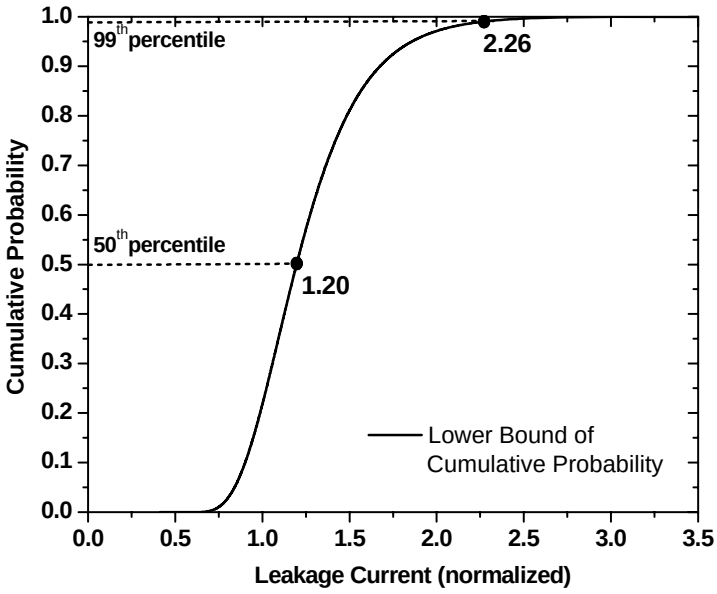

Figure 4.12: Leakage distribution for all chips.

The probabilistic bounds of the leakage power for all chips are shown in Figure 4.12. Since the previous work does not have the closed-form expression for the total leakage distribution of all frequency bins, Figure 4.12 only shows the right-side p-box for the leakage distribution computed by the proposed robust algorithm. The leakage power is 1.20X of the nominal value at the 50^{th} percentile, and 2.26X at the 99^{th} percentile. Thus, the robust strategy permits estimating the leakage distribution for all chips, which enables the chip-level leakage estimation in the early design phase.

4.5 SUMMARY

This chapter proposes a robust estimation approach to computing chip-level leakage dissipation, and parametric yield for the frequency binning scheme. Based on robust representations and operations of random variables, the proposed strategy is able to manipulate a variety of distributions that cannot be handled by analytical techniques, and also takes into account the correlation of variables. Given statistical metrics of partially-specified parameters, the robust estimation methodology is able to provide guaranteed probabilistic bounds for leakage dissipation and parametric yield, thus reducing the conservatism of the interval-based estimates.

Chapter 5: Analysis of Leakage Power Reduction for Dual Threshold Voltage Technologies in the Presence of Large Threshold Voltage Variation

The minimization of leakage power becomes the dominant concern of the nanometer scale CMOS designs, and is part of the general struggle to mitigate the increase in the overall circuit power consumption. The primary reason for the increase in leakage power is the reduction of threshold voltage (V_{th}) of devices, which is causing an exponential increase in leakage current, as described earlier in this dissertation.

Multiple circuit design techniques have been developed to cope with the tremendous growth of leakage. One of the effective and feasible approaches to suppressing leakage is the dual threshold voltage methodology. Introducing a second V_{th} allows maintaining the overall high performance, while reducing leakage current by setting transistors with timing slacks to high V_{th}. While performance difference between the high and low V_{th} transistors is roughly 2X, the leakage current differs by almost 30X [80]. The possibility of setting some transistors to high V_{th} is a potent way to reduce leakage. Today cell libraries of dual threshold voltage levels are provided by foundries as part of the standard service. An important challenge for process engineering is, however, to determine the optimal values of threshold voltages in terms of leakage power reduction, prior to releasing the dual-V_{th} cell libraries to designers.

This chapter proposes an analysis framework for the dual-V_{th} design methodology in the presence of the threshold voltage variation. Random variation in V_{th} is caused by the fluctuation in the number and location of the dopant atoms in the channel of the MOS transistor. The magnitude of the variation in V_{th} is growing as devices shrink [37]. Even

small fluctuations in doping concentration will cause a large change in V_{th}. In order to improve the effectiveness of the dual V_{th} method in reducing leakage power, we must treat V_{th} probabilistically.

We derive a set of analytical models that allow the probabilistic analysis of leakage power reduction within the dual-V_{th} design methodology. The equations that we derive can also be used to probabilistically describe the leakage power, in general. A specific issue that we seek to answer is the way in which the values of the two threshold voltages can be selected in the presence of variability, that is, we seek to find the optimal separation between the nominal values of low and high threshold voltages such that the overall power savings are optimized. With large V_{th} variability, the separation must be large enough to overcome the statistical noise. Without the loss of generality in our work, we assume that the value of lower V_{th} is fixed by the timing requirements [81], thus we focus on the optimal selection of the high V_{th}. Some prior work [34], [35] considered the problem non-statistically. We are the first to introduce a rigorous probabilistic description of the problem.

From this analysis we find that the dual-V_{th} technique may be significantly less effective in reducing power in the presence of variability. The model shows that a higher V_{th}, compared to previous models, is needed to achieve a substantial reduction in leakage power in the presence of large V_{th} variation. The optimal value of the second V_{th} is typically higher compared to the variation-free scenario.

This chapter is organized as follows. Section 5.1 describes the probabilistic framework for circuit delay and leakage power, and Section 5.2 presents the analysis and experimental results. Section 5.3 summarizes this work.

5.1 MINIMIZING LEAKAGE POWER UNDER A PROBABILISTIC THRESHOLD VOLTAGE MODEL

This section develops a probabilistic model to describe leakage power reduction using a dual V_{th} design under V_{th} variability. The objective is to seek a model and an optimization strategy that will allow us to minimize leakage power, P_{leak}, while ensuring that the clock frequency satisfies certain requirements. The traditional way is to formulate the power reduction problem as a constrained optimization problem:

$$\min\ P_{leak}, \text{ s.t. } f_{clock} \geq f^* \tag{5.1}$$

However, in contrast to the above model, we will treat both P_{leak} and delay (frequency) probabilistically. Also, in our actual formulation, we normalize the power of a dual-V_{th} design to that of the single-V_{th} design, and use it as a measure of the effectiveness of leakage power reduction enabled by a dual V_{th} technology. We call this normalized quantity the dual-to-single leakage power *ratio* (R), and it is formally defined as:

$$R = P_{dual} / P_{single} \tag{5.2}$$

It is obvious that the dual-to-single power ratio is less or equal than one. Now, considering the probabilistic nature of our formulation, we will seek to minimize the expected value of the power ratio, $E[R]$, under the constraint that with a certain probability the frequency target will be met:

$$\min\ \ E[R],\ \ s.t.\ P\left\{f_{clock} \geq f^*\right\} \geq \alpha \tag{5.3}$$

The following sections will derive the appropriate probabilistic models for power ratio and delay.

5.1.1 Model of Leakage Power Optimization

To calculate the minimum leakage power achievable in a dual-V_{th} design, we first model the dual-to-single leakage power ratio. We rely on the known modeling strategy for static power reduction [34], [35], and then modify the formulation to include a probabilistic V_{th} model taking into account die-to-die and within-die components of variation. In the model of [34], [35], the circuit is modeled as a collection of non-crossing combinational logic paths. Leakage power is reduced by assigning a subset of gate stages to higher V_{th}. The model assumes that an arbitrary fraction of the total circuit capacitance, which is assumed to be proportional to the total gate width, can be assigned to higher V_{th}.

We start by modeling the leakage power dissipation at the device level. In [82], the subthreshold leakage current of a unit-width transistor can be described by:

$$I_{leak} = I_0 e^{-\frac{\ln 10}{s} V_{th}} \tag{5.4}$$

where *s* is the subthreshold swing.

The threshold voltage is modeled as a random variable to allow the probabilistic treatment of the threshold voltage variation. Empirical evidence suggests that variability of threshold voltage can be modeled by the normal distribution [83]. Variability of threshold voltage can be decomposed into the sum of the die-to-die component ($\Delta V_{th,dd}$), and within-die component ($\Delta V_{th,wd}$):

$$V_{th} = \overline{V}_{th} + \Delta V_{th,dd} + \Delta V_{th,wd} \tag{5.5}$$

where $\overline{V}_{th}$ is the mean of the threshold voltage, $\Delta V_{th,dd} \sim N(0, \sigma^2_{V_{th,dd}})$, and $\Delta V_{th,wd} \sim N(0, \sigma^2_{V_{th,wd}})$.

We can rewrite (5.4) as:

$$I_{leak,i} = I_0 e^{-\frac{1n10}{s}\left(\overline{V}_{th} + \Delta V_{th,dd} + \Delta V_{th,wd,i}\right)} \tag{5.6}$$

Since all devices on the chip have the identical die-to-die component, we can first evaluate the impact of the within-die component on the leakage current. Assuming the within-die variations of distinct gates are identically and independently distributed (i.i.d.), and the number of gates on the chip is large, the sum of the leakage current due to within-die components can be approximated by the expected value of the leakage current, according to the Central Limit Theorem [43]. Under the assumption that V_{th} is a normal random variable, the leakage current of a single gate follows a lognormal distribution. The expected value of a log-normally distributed function can be found in closed form [43]. Thus, the total chip leakage current can be modeled as:

$$\begin{aligned} I_{leak,total} &= \sum_i W_i I_{leak,i} = \sum_i W_i I_0 e^{-\frac{\ln 10}{s}(\overline{V}_{th} + \Delta V_{th,dd})} e^{-\frac{\ln 10}{s}\Delta V_{th,wd,i}} \\ &\simeq I_0 e^{-\frac{\ln 10}{s}(\overline{V}_{th} + \Delta V_{th,dd})} \sum_i W_i E\left[e^{-\frac{\ln 10}{s}\Delta V_{th,wd,i}}\right] \\ &= \left(\sum_i W_i\right) I_0 e^{-\frac{\ln 10}{s}(\overline{V}_{th} + \Delta V_{th,dd})} e^{\frac{1}{2}\left(\frac{\ln 10}{s}\right)^2 \sigma^2_{V_{th,wd}}} \end{aligned} \tag{5.7}$$

where W_i is the gate width of gate i.

We now compute the total leakage power of a circuit with dual threshold voltages, V_{th}^h and V_{th}^l. The leakage power of a dual threshold voltage design, P_{dual}, can be computed by summing up the leakage current contributed by gates under V_{th}^h and V_{th}^l:

$$P_{dual} = V_{dd} I_0 \times \begin{pmatrix} W_h \cdot e^{-\frac{\ln 10}{s}(\overline{V}_{th}^h + \Delta V_{th,dd}^h)} e^{\frac{1}{2}\left(\frac{\ln 10}{s}\right)^2 \sigma^2_{V_{th,wd}^h}} \\ + (W - W_h) \cdot e^{-\frac{\ln 10}{s}(\overline{V}_{th}^l + \Delta V_{th,dd}^l)} e^{\frac{1}{2}\left(\frac{\ln 10}{s}\right)^2 \sigma^2_{V_{th,wd}^l}} \end{pmatrix} \tag{5.8}$$

where W_h is the total gate width in the entire circuit set to the high threshold voltage, W is the total gate width in the circuit, and V_{dd} is the power supply voltage.

The leakage power of a single threshold voltage design, P_{single}, can be obtained by setting W_h in (5.8) to zero. Since the total transistor width and gate stages are largely proportional, the ratio of the total width under different threshold voltages can be approximated as the ratio of gate stages:

$$\frac{W_h}{W} \approx \frac{N_h}{N} \tag{5.9}$$

where N_h is the number of gate stages in the entire circuit set to the high threshold voltage, and N is the total number of gates stages in the circuit. Then the power ratio can be computed by

$$\begin{aligned} R &= \frac{P_{dual}}{P_{single}} \\ &= 1 - \frac{N_h}{N}\left(1 - \exp\left(-\frac{\ln 10}{s}\left(\overline{V}_{th}^{h} - \overline{V}_{th}^{l} + \Delta V_{th,dd}^{h} - \Delta V_{th,dd}^{l}\right)\right) f(\sigma^2_{V^h_{th,wd}}, \sigma^2_{V^l_{th,wd}})\right) \end{aligned} \tag{5.10}$$

where $f(\sigma^2_{V^h_{th,wd}}, \sigma^2_{V^l_{th,wd}}) = \exp\left(\frac{1}{2}\left(\frac{\ln 10}{s}\right)^2 \left(\sigma^2_{V^h_{th,wd}} - \sigma^2_{V^l_{th,wd}}\right)\right)$. Finally, we can compute the expected value of the power ratio, $E[R]$:

$$E[R] = 1 - \frac{N_h}{N}\left(1 - g\left(V_{os}\right)\right) \tag{5.11}$$

where $V_{os} = V_{th}^{h} - V_{th}^{l}$, and

$$g\left(V_{os}\right) = \exp\left(-\frac{\ln 10}{s}\overline{V}_{os} + \frac{1}{2}\left(\frac{\ln 10}{s}\right)^2 \left(\sigma^2_{V^h_{th,wd}} - \sigma^2_{V^l_{th,wd}} + \sigma^2_{V^h_{th,dd}} + \sigma^2_{V^l_{th,dd}}\right)\right) \tag{5.12}$$

where $\overline{V}_{os} = E[V_{os}]$.

The model above thus captures the ratio of the static power of a dual-V_{th} design to that of a single-V_{th} design taking into account the statistical nature of the threshold

voltage. In the next section we derive a probabilistic model of path delay that allows us to express the ratio of gate stages set to high V_{th} and thereby allow us to calculate the expected value of power ratio. Finally, we will analyze the impact of high V_{th} and σ_{Vth} on the power ratio to identify the optimal values of the separation of two threshold voltages.

5.1.2 Probabilistic Circuit Delay Modeling

In a dual-V_{th} design methodology, the static power is minimized by setting a fraction of gates on paths with timing slacks to a higher V_{th}. This makes the paths slower but less leaky. In this section, we provide a probabilistic description of path delay degradation as a result of setting a portion of the gate stages on a path to a higher V_{th}.

Let D be the delay of a path with a mixture of gates at low and high threshold voltages. We rely on the observation that, to a good approximation, path delay is proportional to the total path capacitance [35], which, we in turn approximate by gate stages. Therefore, the path delay can be described as:

$$D = \sum_{i=1}^{n_h} d_h + \sum_{j=1}^{n-n_h} d_l \tag{5.13}$$

where d_h and d_l correspond to the gate delay at high and low threshold voltages, n is the number of gate stages along a path, and n_h is the number of gate stages assigned to high V_{th}.

Due to the uncertainty of threshold voltage, the path delay needs to be modeled probabilistically. Although several process parameters have an impact on the gate delay, our analysis mainly focuses on the variability of threshold voltage. We start to explore the relation between the variability of the threshold voltage and the gate delay. From the alpha power law, we know the dependency of gate delay on the threshold voltage.

$$d_h \propto \frac{1}{\left(V_{dd} - V_{th}^h\right)^\alpha} \tag{5.14}$$

We can write the gate delay as:

$$d_h = k \cdot \left(V_{dd} - V_{th}^h\right)^{-\alpha} \tag{5.15}$$

Using the statistical delta models, the gate delay can be described as:

$$d_h \approx \overline{d_h} + \left.\frac{\partial d_h\left(V_{th}^h\right)}{\partial V_{th}^h}\right|_{V_{th}^h = \overline{V}_{th}^h} \Delta V_{th}^h \tag{5.16}$$

where $\overline{d_h}$ is the mean of the high-V_{th} gate delay, and $\Delta V_{th}^h = V_{th}^h - \overline{V}_{th}^h$. For low-$V_{th}$ gates, the derivation is similar; thus, here we only show the formulae for high-V_{th} gates.

Let s_h denote the first-order derivative of the gate delay with respect to the threshold voltage, which is also known as the delay sensitivity. It can be computed from (5.15):

$$s_h = \frac{\partial}{\partial V_{th}^h} d_h\left(V_{th}^h = \overline{V}_{th}^h\right) = \frac{-\alpha k}{\left(V_{dd} - \overline{V}_{th}^h\right)^{\alpha+1}} = \frac{-\alpha \overline{d}_h}{V_{dd} - \overline{V}_{th}^h} \tag{5.17}$$

From (5.5), the proposed framework takes into account die-to-die and within-die components of threshold voltage variation. Combining (5.5), (5.16), and (5.17), the gate delay can be represented by:

$$d_h = \overline{d_h} + s_h\left(\Delta V_{th,dd}^h + \Delta V_{th,wd}^h\right) \tag{5.18}$$

Following (5.13) and (5.18), the path delay can be then computed by summing the gate delays along the path:

$$\begin{aligned} D = {} & n_n \overline{d}_h + (n - n_n)\overline{d}_l + n_h s_h \Delta V_{th,dd}^h + (n - n_h) s_l \Delta V_{th,dd}^l \\ & + \sum_{i=1}^{n_h} s_h \Delta V_{th,wd,i}^h + \sum_{j=1}^{n-n_h} s_l \Delta V_{th,wd,j}^l \end{aligned} \tag{5.19}$$

The variance of the path delay can be computed by:

$$\begin{aligned} Var\{D\} = n_h{}^2 s_h{}^2 \sigma^2_{V^h_{th,dd}} + \left(n - n_h\right)^2 s_l{}^2 \sigma^2_{V^l_{th,dd}} \\ + n_h s_h{}^2 \sigma^2_{V^h_{th,wd}} + (n - n_h) s_l{}^2 \sigma^2_{V^l_{th,wd}} \end{aligned} \tag{5.20}$$

Additionally, the mean of the path delay is:

$$\begin{aligned} E[D] &= n_h \bar{d}_h + \left(n - n_h\right)\bar{d}_l \\ &= n\bar{d}_l + n_h\left(\bar{d}_h - \bar{d}_l\right) \end{aligned} \tag{5.21}$$

The above equation implies that the mean of path delay is a linear function of n_h, the number of the high-V_{th} gates on the path. Therefore, the mean of path delay increases with the number of high-V_{th} gates on the path. Having derived the mean and variance of the path delay, we continue to specify the delay constraints in the next section.

5.1.3 Finding Optimal Threshold Voltage Separation under the Probabilistic Models

In this section we derive the analytical framework for finding the optimal value of the high V_{th} under the statistical description of path delays and power consumption. The objective is to minimize the expected power ratio, $E[R]$, under the timing constraint that no path delay exceeds the critical path delay. In order to make the analysis tractable, we use a simplified model in which the circuit consists of a collection of *M* non-crossing paths [35]. In contrast to earlier approaches, this circuit delay constraint is formulated probabilistically, in terms of the probability of meeting the timing constraint:

$$P\{\text{ckt delay} \le T\} = P\left\{\max\{D_1,...,D_M\} \le T\right\} = \alpha_d \tag{5.22}$$

Because we assume that the paths are non-crossing,

$$P\{\text{ckt delay} \le T\} = \prod_{i=1}^{M} P\{D_i \le T\} = \alpha_d \tag{5.23}$$

For a specified confidence level α_d, we can compute the probability that every path does not violate the timing constraint. This probability is given by:

$$P\{D_i \le T\} = \alpha_d^{1/M}. \tag{5.24}$$

The timing constraint that we enforce is that no path delay can be greater than the critical path delay. To ensure that no path delay is greater than the critical path delay, we must satisfy for a given confidence level α_c:

$$P\{D_i \le T\} = \alpha_c. \tag{5.25}$$

Since we assume that the path delays are described by the normal distribution, we can re-write (5.25) as:

$$D_i^{\alpha} = E[D_i] + \phi^{-1}(\alpha_c)\sigma_{D_i} \le T \tag{5.26}$$

where ϕ is the cumulative distribution function of the standard normal distribution, and σ_{D_i} is the standard deviation of the path delay. Any value of α_c can be used, for example, a convenient value is $\alpha_c = 99.7\%$, such that $\phi^{-1}(\alpha_c) = 3$.

With the constraint on delay given in (5.26) we can now find the number of gates along a path to set to the high V_{th}. As followed from (5.11), the power ratio *R* is linearly proportional to the number of gate stages that we can set to high V_{th} in the entire circuit. Note that

$$N_h = N \cdot N_{ave}^h \tag{5.27}$$

where N_{ave}^h is the average ratio of gates set to high V_{th} per path, which can be found by evaluating:

$$N_{ave}^h = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{n_h}{n}\right)_i \tag{5.28}$$

where *M* is the number of paths in the circuit. We can compute the value of $\left(\frac{n_h}{n}\right)_i$ per path by maximizing the number of high V_{th} gates on the path, while satisfying the timing

constraint. Based on the expressions of the path delay mean and variance, (5.20) and (5.21), we may re-write (5.26) as:

$$D_i^{\alpha} = n\overline{d}_l + n_h\left(\overline{d}_h - \overline{d}_l\right) + \phi^{-1}(\alpha_c)\cdot\sqrt{Var\left\{D_i\right\}} \le T \tag{5.29}$$

where $Var\left\{D_i\right\} = n_h{}^2 s_h{}^2 \sigma^2_{V^h_{th,dd}} + \left(n - n_h\right)^2 s_l{}^2 \sigma^2_{V^l_{th,dd}} + n_h s_h{}^2 \sigma^2_{V^h_{th,wd}} + (n - n_h) s_l{}^2 \sigma^2_{V^l_{th,wd}}$.

In the above expression, only the values of n and n_h are not determined. Note that the number of gates in the path, n, can be computed given the path delay before the optimization, i.e., all gates on the path are assigned low threshold voltage. Given the mean delay of a single path with all low-V_{th} gates, D_L, the number of gates in the path, n, can be computed by:

$$n = \frac{D_L}{\overline{d}_l}. \tag{5.30}$$

We can now set an analytical quadratic equation to find n_h per path as a function of D_L.

To quantitatively evaluate the average ratio of gates set to the higher V_{th}, we assume a specific shape of the initial path delay distribution, $p(D_L)$. First we use the triangular distribution that has been shown to be characteristic of many circuits [84]. To make the analysis simpler, the model normalizes all path delays to critical path delay. Then, by integrating $n_h\left(D_L\right)$ for individual paths over the entire path delay distribution, N^h_{ave} can be computed:

$$N^h_{ave} = \frac{\int_0^1 n_h(D_L)p(D_L)\partial D_L}{\int_0^1 n_i\left(D_L\right)p(D_L)\partial D_L} \tag{5.31}$$

where n_i is the number of gate stages per path. In our implementation, we use numerical integration to compute the value of N^h_{ave} for specific distributions. Then, we can link it

back to the original equation for the expected power ratio,(5.11), such that the result is only a function of the V_{th} separation.

$$E[R] = 1 - N_{ave}^{h} \cdot \left(1 - g(V_{os})\right) \tag{5.32}$$

This equation can now be used to explore the dependence of the expected power ratio on V_{th} separation, and for finding the optimal value of the higher V_{th}.

5.2 Analysis and Experimental Results

We have implemented the analytical results developed in the earlier sections in a MATLAB-based analysis environment. This is a fast implementation that allows us to rapidly consider multiple scenarios with respect to the magnitude and characteristics of V_{th} variation, and other circuit parameters, such as the power supply voltage (V_{dd}) and the confidence level (α). As for the magnitude of the threshold voltage variation, our assumption is that the variance is equal for low and high V_{th} gates, with the die-to-die and within-die components contributing 60% and 40% of total variance, respectively. The subthreshold voltage swing is 90mV/dec, and the constant for the alpha power law is 1.3. In the experiments the power supply voltage (V_{dd}) is 1.0V, and the low-V_{th} is 0.30V.

Figure 5.1 shows the expected power ratio for a range of variance values. The minimum $E[R]$ and the optimal high threshold voltages for different values of variances are also shown in Table 5.1. Higher variation of threshold voltage results in substantial leakage power dissipation. It can be seen that the power reduction enabled by the use of the dual-V_{th} design techniques gets smaller as the amount of variability increases. For example, the minimum value of $E[R]$ is 0.106 and 0.332 for the cases with σ_{Vth} = 0mV and 50mV, respectively. It may be easier to see the significance of this result if we consider the inverse of the power ratio that can be interpreted as the gain in power efficiency enabled by the dual-V_{th} design. Thus, while the gain is about 10X with no

variability, the gain is only 3X when σ_{Vth} = 50mV. In a sense, the efficiency of the dual-V_{th} technique has become 3X lower.

Furthermore, the probabilistic model shows that under a dual-V_{th} approach, the variation-free scenario skews the actual gains, and does not allow one to pick the truly optimal values of V_{th}. The V_{th} value that minimizes the expected static power is approximately $V_{th}^{h} = V_{th}^{l} + 0.135V_{dd}$ with σ_{Vth}=50mV. For the variation-free scenario, the model predicts that the minimum static power occurs at a high V_{th} value of about $0.124V_{dd}$ greater than low V_{th}.

Table 5.1: The minimum $E[R]$ vs. the optimal high V_{th} for different values of variances.

σ_{Vth} (mV)	0	10	20	30	40	50
***E*[*R*]**	0.106	0.138	0.177	0.222	0.274	0.332
Optimal High V_{th}	0.424	0.419	0.418	0.421	0.426	0.435

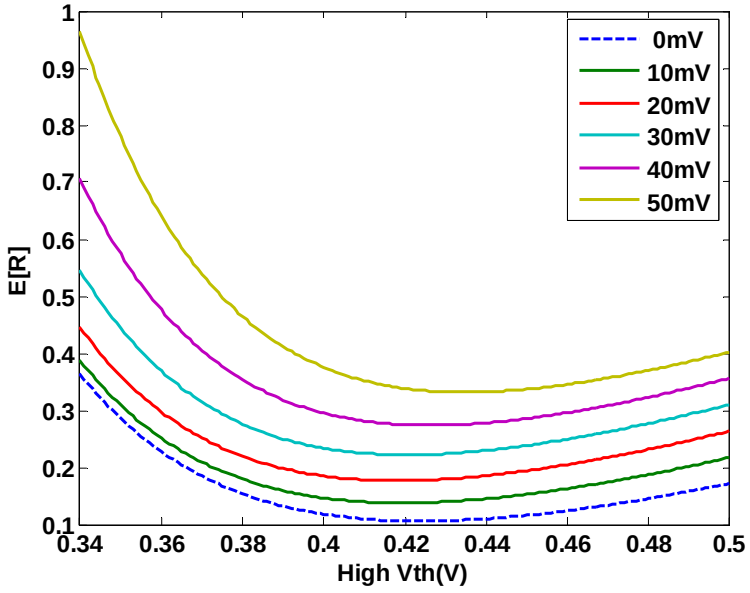

Figure 5.1: The $E[R]$ vs. the value of the higher V_{th} for different values of σ_{Vth}.

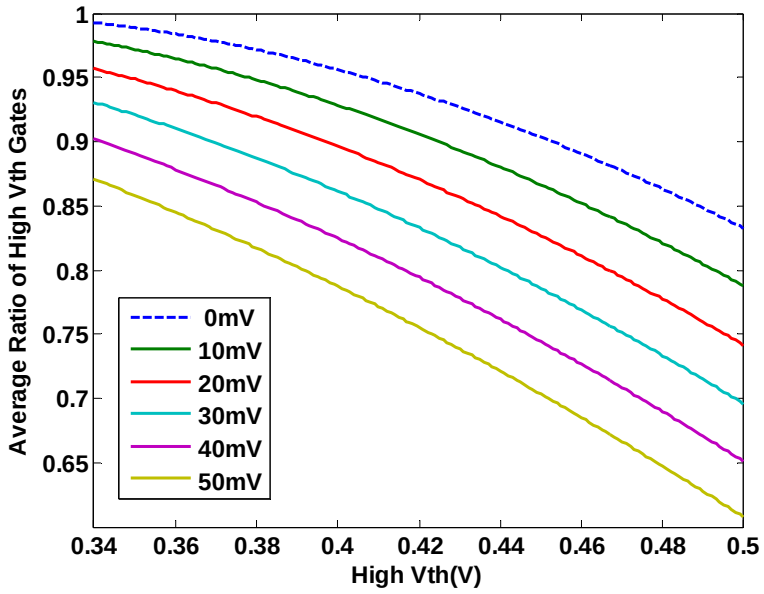

Figure 5.2: Average ratio of high V_{th} gates vs. optimal high V_{th} for different values of σ_{Vth}.

Figure 5.2 shows the average ratio of high-V_{th} gates (N_{ave}^{h}) under different V_{th} variations for the triangular path delay distribution. As expected, high V_{th} variation leads to low N_{ave}^{h}, thus resulting in high static power dissipation. Hence, the probabilistic model for V_{th} becomes more important as the variation in V_{th} gets larger, which is predicted to occur given the current scaling trends.

From Figure 5.1 and Table 5.1, it can be observed that for low magnitudes of V_{th} variation (e.g., σ_{Vth}=10-30mV), the optimal value of high V_{th} is lower than that of the variation-free case. The optimal high V_{th} decreases until the standard deviation of V_{th} increases to a specific value (e.g., $\sigma_{V_{th}}$=20mV in Figure 5.1). Beyond that value, the greater the variation in V_{th}, the higher the value of high V_{th} has to be to minimize static power. From our experiments, it appears that the specific value of V_{th} variance ($\sigma_{\min Vth}$), after which the optimum value of high V_{th} monotonically grows, is affected primarily by

the subthreshold voltage swing (s). Figure 5.3 illustrates the dependency of $\sigma_{\min Vth}$ on the threshold voltage swing. The value of $\sigma_{\min Vth}$ grows monotonically with the subthreshold voltage swing.

Figure 5.4 shows the dependence of $E[R]$ on the quantile point of the probabilistic path distribution. Our basic approach is to take the 3-sigma point of ΔD^{α}. Clearly, the higher the confidence level that the timing constraints are not violated, the fewer the gates that can be assigned to high V_{th}, and the higher is the power dissipation. From Figure 5.4, the values of the minimum $E[R]$ for using the 50th percentile and the 3-sigma point are 0.148 and 0.332, respectively. However, the optimal high V_{th} value appears to be dependent on the confidence level. Comparing the use of the 50th percentile (mean delay) to the 3-sigma point, we find that the optimal V_{th} value changes by 19 mV.

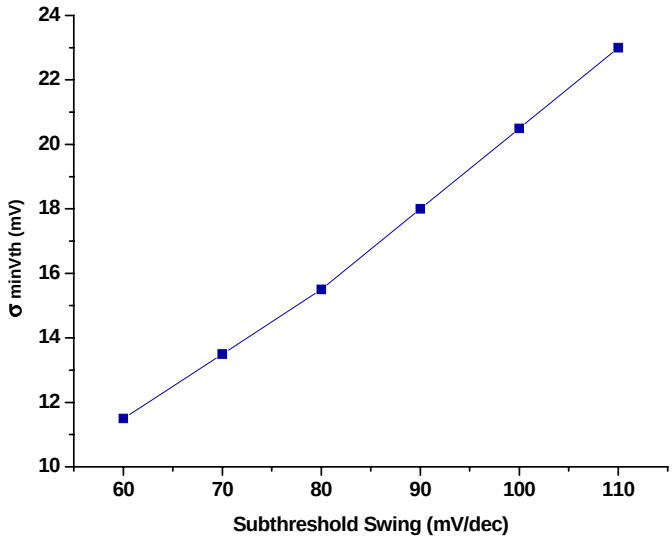

Figure 5.3: The value of V_{th} variance after which the optimum value of high V_{th} monotonically grows is a function of subthreshold voltage swing.

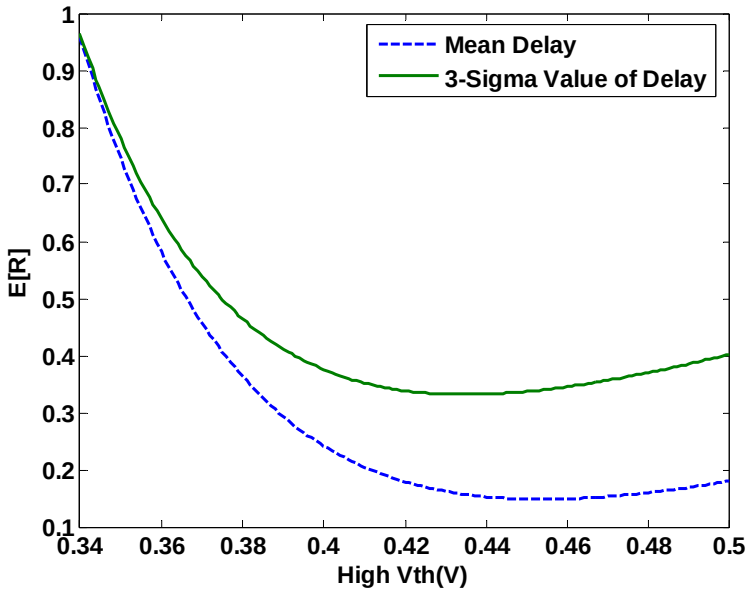

Figure 5.4: $E[R]$ vs. the value of higher Vth for the mean delay and 3-sigma point (σ_{Vth}=50mV).

Table 5.2: Values of optimal higher V_{th} for different initial path delay distributions (σ_{Vth}= 50mV).

Distribution Type	**Optimal Higher V_{th}**	**Minimum $E[R]$**	N^{h}_{ave}
Triangle	$Vth^{l} + 0.13 \cdot V_{dd}$	0.33	0.72
Uniform	$Vth^{l} + 0.13 \cdot V_{dd}$	0.53	0.52
Sloped	$Vth^{l} + 0.12 \cdot V_{dd}$	0.65	0.40

Not all circuits can be approximated by the triangular path delay distribution and here we also include the results for a sloped path distribution, where most of the paths have delays close to the critical path delay, and a uniform path delay distribution. As expected, the achievable power savings in a dual V_{th} approach for both distributions is smaller due to the greater number of paths with the delay near the critical path. Table 5.2 gives a summary of the optimal V_{th} value for each case.

We find from the analysis of different distributions that although the amount of power savings is different, the optimal value of the higher V_{th} changes only slightly. In addition, since the dependence of E[R] on V_{th} is rather weak for a range of V_{th} values near the optimum, any V_{th} value in this range provides about the same amount of power savings. That is, there is a range in which raising V_{th} provides diminishing returns in terms of power savings. As a rule of thumb, when the high threshold voltage is equal to $V_{th}^{l} + 0.13V_{dd}$, it provides the highest amount of savings for most typical path distributions.

Figure 5.4 would seem to suggest that any value of high V_{th} from $V_{th}^{l} + 0.13V_{dd}$ to $V_{th}^{l} + 0.18V_{dd}$ would be a suitable choice, as all high V_{th} values within this range result in approximately the same amount of power dissipation. However, there is a cost of further V_{th} increase: removing slack from the sub-critical path prevents using other circuit optimization techniques for power reduction, such as, transistor sizing. The key is to use the technique that best trades slack for lower power, similar to work done in [84]. If the high V_{th} value is calculated on a pre-optimized path delay distribution, then a value for high V_{th} can be chosen that represents the best power-delay tradeoff.

To show this, we plot the degradation of $E[R]$ in Figure 5.5, as a function of the increased gate stage delay. Let $\gamma = d_h / d_l$ be the degradation of delay per gate stage, with d_h and d_l corresponding to the gate delay at high and low V_{th}. Using the alpha-power law model, we can show that this ratio can be described as:

$$\gamma = \left(\frac{V_{dd} - V_{th}^{l}}{V_{dd} - V_{th}^{h}} \right)^{\alpha} \tag{5.33}$$

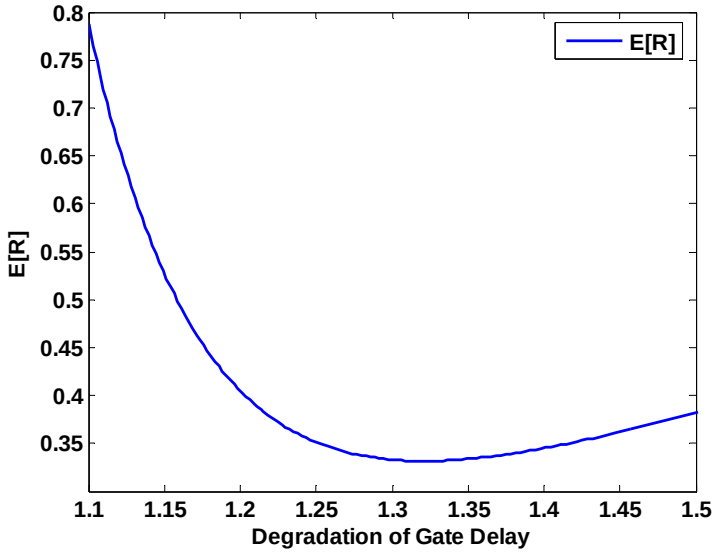

Figure 5.5: Degradation in gate delay (γ) vs. $E[R]$ (σ_{Vth}= 50mV).

The graph is similar in shape to Figure 5.4, because the degradation of gate delay is nearly linear for small values of the high V_{th}. It is clear that the value of V_{th} that minimizes static power is situated at a point of very unfavorable power-delay tradeoff. Comparing Figure 5.4 and Figure 5.5, we can see that the minimum power is achieved at a point at which $\gamma = 1.32$. A better value for the high V_{th} would instead be at a slightly lower value for the high V_{th}. For example, at a high V_{th} equal to $V_{th}^{l} + 0.10V_{dd}$ there is still a 62% savings in power with a gate stage delay of only $\gamma = 1.22$, which would leave considerably more slack for other circuit optimization techniques.

5.3 SUMMARY

This chapter derives a probabilistic analytical framework for selecting the optimal high threshold voltage to minimize the expected static power for the dual-V_{th} designs. From this analysis we find that the dual-V_{th} technique is significantly less effective in reducing power in the presence of variability. We observe that under large variability a

larger separation between the lower and higher threshold voltages is required to achieve optimal leakage power reduction. These findings highlight the importance of using a probabilistic approach to circuit analysis as fundamental process variability continues to increase.

Chapter 6: Conclusions

In the nanometer-scale regime the traditionally formulated deterministic timing and power analysis algorithms may result in the over-conservatism, which entails the development of statistical analysis approaches to accounting for the growing variability of parameters and performance metrics. The objective of this dissertation is to develop robust statistical analysis algorithms for designers to evaluate the impact of parameter variability on circuit timing and power consumption, which facilitates power and timing verification, and guides later circuit optimization. This dissertation has investigated: 1) fast statistical timing analysis handling path delay correlations; 2) statistical timing analysis based on incomplete parameter uncertainty; 3) estimation of parametric yield and leakage power consumption under realistic probabilistic descriptions of parameters, and 4) analysis of leakage power for dual threshold voltage designs in the presence of threshold voltage variation.

Timing verification is one of the important steps in the circuit design process. Statistical static timing analysis has been proposed to account for variability of parameters, and reduce the over-conservatism of the corner-based estimates. Among SSTA approaches, Monte-Carlo and parameter-space integration methods are accurate, but computationally prohibitive. This dissertation develops a fast path-based timing analysis algorithm to compute bounds for the circuit delay distribution, based on the theory of stochastic majorization and characteristics of the multivariate normal distribution. In computing the circuit delay distribution, we adopt the path-based traversal scheme that permits accounting for delay correlations due to path reconvergence.

Compared to the Monte-Carlo simulation results, the proposed algorithm can construct tight probabilistic bounds, and also achieves run-time efficiency.

The limited availability of parameter distributions restricts the practical use of statistical approaches. In some cases, only partial statistical information is accessible; however, this kind of information cannot be incorporated into the existing statistical framework. The alternative approach is to use the intervals of parameters, but it may result in the over-conservatism. To address this real-life concern, this dissertation proposes a new modeling strategy for handling partial probabilistic descriptions, and develops a SSTA algorithm based on this strategy. Specifically, we apply an analytical bound for the cumulative distribution function to compute path delays, and propose a robust Monte Carlo sampling technique for circuit delay. Compared to timing estimated based on intervals, the proposed algorithm can reduce the over-conservatism, thus avoiding the over-design to save power and area.

The tremendous growth of leakage power is a significant menace to technology scaling. The subthreshold leakage and gate tunneling leakage power dissipation has become comparable to dynamic power consumption. Besides, variability of leakage and its inverse correlation with chip frequency causes a serious loss in parametric yield. Thus, this dissertation proposes a robust estimation strategy for parametric yield and leakage dissipation. The developed algorithm can handle full and partial probabilistic descriptions of parameters, and provide guaranteed bounds for yield and leakage power. Based on robust representations and computation of variables, the algorithm can handle correlated random variables and reduce the over-conservatism of the leakage estimates from worst-case analysis.

This dissertation also investigates one of the effective approaches to mitigating the subthreshold leakage: the dual-threshold voltage design methodology. We focus on

the effectiveness of the dual-V_{th} designs in the presence of the threshold voltage variation. We develop an analytical framework to compare the power dissipation of the single-V_{th} and dual-V_{th} designs. We also propose an algorithm for selecting the optimal threshold voltage values for power reduction. The analysis shows that under large V_{th} variations the dual-V_{th} technique becomes less effective in reducing power.

In the future technology generations, the variability of parameters will continue to pose difficulties for circuit performance analysis and optimization. There will be a larger number of variability sources, and the magnitude of variability will increase, due to the growing circuit complexity and technology scaling. The circuit designers will need statistical analysis algorithms that can accomplish two major objectives: scalability and robustness. Statistical algorithms must have the capacity to handle a huge number of sources of variability in the circuit, and also provide guaranteed bounds for circuit performance even under limited information of variability. As a consequence, statistical timing and power analysis will become increasingly important in the future, and there are still many important and challenging problems ahead for us.

Bibliography

[1] *International Technology Roadmap for Semiconductors*, 2005 Edition & 2006 Update.

[2] T.-C. Chen, "Where CMOS is going: trendy hype vs. real technology," in *Proc. ISSCC*, 2006, pp. 1-18.

[3] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol. 19, no. 4, pp. 23-29, Jul./Aug., 1999.

[4] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, and A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitectures," in *Proc. DAC*, 2003, pp. 338-342.

[5] O. Unsal, J. Tschanz, K. Bowman, V. De, X. Vera, A. Gonzalez, and O. Ergin, "Impact of parameter variations on circuits and microarchitecture," *IEEE Micro*, vol. 26, no. 6, pp. 30-39, Nov./Dec., 2006.

[6] S. Nassif, "Delay variability: sources, impact and trends," in *Proc. ISSCC*, 2000, pp. 368-369.

[7] K. Bowman, S. Duvall, and J. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE J. Solid State Circuits*, vol. 37, no. 2, pp. 183-190, 2002.

[8] A. Agarwal, D. Blaauw, and V. Zolotov, "Statistical timing analysis for intra-die process variations with spatial correlations," in *Proc. ICCAD*, 2003, pp. 900-907.

[9] 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 Trans. Computer Aided Design of Integrated Circuits and Systems*, vol. 21, no. 5, pp. 544-553, May 2002.

[10] A. Dasdan and I. Hom, "Handling inverted temperature dependence in static timing analysis," *ACM Trans. Design Automation of Electronic Systems*, vol. 11, no. 2, pp.306-324, Apr. 2006.

[11] S. Hassoun and T. Sasao, *Logic Synthesis and Verification*, Kluwer Academic Publisher, 2002.

[12] D. G. Malcolm, J. H. Roseboom, C. E. Clark, and W. Fagar, "Application of a technique for research and development program evaluation," *Operations Research*, vol. 7, no. 5, pp. 646-669, 1959.

[13] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Computational and refinement of statistical bounds on circuit delay," in *Proc. DAC*, 2003, pp. 348-353.

[14] M. Orshansky and A. Bandyopadhyay, "Fast statistical timing analysis handling arbitrary delay correlations," in *Proc. DAC*, 2004, pp. 337-342.

[15] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," in *Proc. DAC*, 2004, pp. 331-336.

[16] R. Hitchcock, "Timing verification and the timing analysis program," in *Proc. DAC*, 1982, pp. 594-604.

[17] H.-F. Jyu, S. Malik, S. Devadas, and K. W. Keutzer, "Statistical timing analysis of combinational logic circuits," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol.1, no.2, pp. 126-137, 1993.

[18] J. A. G. Jess, K. Kalafala, S. R. Naidu, R. H. J. M. Otten, and C. Visweswariah, "Statistical timing for parametric yield prediction of digital integrated circuits," in *Proc. DAC*, 2003, pp. 932-937.

[19] M. Berkelaar, "Statistical delay calculation", in *Proc. International Workshop on Logic Synthesis*, 1997, pp. 15-24.

[20] A. Gattiker, S. Nassif, R. Dinakar, and C. Long, "Timing yield estimation for static timing analysis," in *Proc. ISQED*, 2001, pp. 437-442.

[21] A. Devgan and C. Kashyap, "Block-based static timing analysis with uncertainty," in *Proc. ICCAD*, 2003, pp. 607-614.

[22] H. Chang and S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal", in *Proc. ICCAD*, 2003, pp. 621-625.

[23] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst case timing analysis," in *Proc. DAC*, 2002, pp. 556-561.

[24] A. Nadas, "Probabilistic PERT," *IBM J. Research and Development*, vol. 23, no.3, pp. 339-347, 1979.

[25] R. E. Moore, *Interval Analysis*, Prentice-Hall, 1966.

[26] S. Ferson, V. Kreinovich, L. Ginzburg, D. Myers, and K. Sentz, "Constructing probability boxes and Dempster-Shafer structures," Sandia Report, SAND2002-4015, Extended Version, 2002.

[27] M. Ceberio, M. Orshansky, G. Xiang, and W.-S. Wang, "Interval-based robust statistical techniques for non-negative convex functions, with application to timing analysis of computer chips", in *Proc. ACM Symp. Applied Computing*, 2006, pp. 1645-1649.

[28] E. J. Nowak, "Maintaining the benefits of CMOS scaling when scaling bogs down," *IBM J. Research and Development.*, vol. 46, no. 2/3, pp. 169-180, 2002.

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

[30] T. Sakurai, and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584-594, April 1990.

[31] T. Sakurai, "Alpha power-law MOS model," *IEEE Solid State Circuits Quarterly Newsletter*, vol. 9, no. 4, pp. 4-5, Oct. 2004.

[32] S. Sirichotiyakul, T. Edwards, C. Oh, J. Zuo, A. Dharchoudhury, R. V. Panda, and D. Blaauw, "Stand-by power minimization through simultaneous threshold voltage selection and circuit sizing," in *Proc. DAC*, 1999, pp. 436-441.

[33] P. Pant, R. Roy and A. Chatterjee, "Dual-threshold voltage assignment with transistor sizing for low power CMOS circuits," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 2, pp. 390-394, 2001.

[34] A. Srivastava, and D. Sylvester, "Minimizing total power by simultaneous Vdd/Vth assignment," in *Proc. ASPDAC*, pp. 400-406, 2003.

[35] M. Hamada, Y. Ootaguro, and T. Kuroda, "Utilizing surplus timing for power reduction," in *Proc. CICC*, 2001, pp. 89-92.

[36] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," *IEEE Micro*, vol. 25, no. 6, pp. 10-16, Nov./Dec., 2005.

[37] A. Asenov, S. Kaya, and J. H. Davies, "Intrinsic threshold voltage fluctuations in decanano MOSFETs due to local oxide thickness variations," *IEEE Trans. Electron Devices*, vol. 49, no. 1, pp.112-119, 2002.

[38] R. Rao, A. Devgan, D. Blaauw, and D. Sylvester, "Parametric yield estimation considering leakage variability," in *Proc. DAC*, 2004, pp. 442-447.

[39] H. Chang and S. Sapatnekar, "Full-chip analysis of leakage power under process variations, including spatial correlations," in *Proc. DAC*, 2005, pp. 523-528.

[40] D. Berleant and J. Zhang, "Using Pearson correlation to improve envelopes around the distributions of functions," *Reliable Computing*, vol. 10, no.2, pp. 139-161, 2004.

[41] A. Agarwal, D. Blaauw, S. Sundareswaran, V. Zolotov, K. Gala, M. Zhou, and R. Panda, "Path-based statistical timing analysis considering inter- and intra-die correlations," in *Proc. ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, 2002, pp. 16-21.

[42] Y. L. Tong, *Probability Inequalities in Multivariate Distributions*, Academic Press, 1980.

[43] W. Feller, *An Introduction to Probability Theory and Its Applications*, Wiley and Sons, 3rd Edition, 1968.

[44] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and Its Applications*, Academic Press, 1979.

[45] S. Yen, D. Du, and S. Ghanta, "Efficient algorithms for extracting the k most critical paths in timing analysis," in *Proc. DAC*, 1989, pp. 649-652.

[46] Y.-C. Ju and R. Saleh, "Incremental techniques for the identification of statically sensitizable critical paths," in *Proc. DAC*, 1991, pp. 541-546.

[47] Y. Zhan, A. Strojwas, X. Li, L. Pileggi, D. Newmark, and M. Sharma, "Correlation-aware statistical timing analysis with non-Gaussian delay distributions," in *Proc. DAC*, 2005, pp. 77-82.

[48] H. Chang, V. Zolotov, S. Narayan, and C. Visweswariah, "Parameterized block-based statistical timing analysis with non-Gaussian parameters and nonlinear delay functions," in *Proc. DAC*, 2005, pp. 71-76.

[49] L. Zhang, W. Chen, Y. Hu, J. Gubner, and C.-P. Chen, "Correlation-Preserved Non-Gaussian Statistical Timing Analysis with Quadratic Timing Model," in *Proc. DAC*, 2005, pp. 83-88.

[50] Dan Ernst, S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, N. S. Kim, and K. Flautner, "Razor: circuit-level correction of timing errors for low-power operation," *IEEE Micro*, vol. 24, no. 6, pp. 10-20, 2004.

[51] D. Kouroussis, I. A. Ferzli, and F. N. Najm, "Incremental partitioning-based vectorless power grid verification," in *Proc. ICCAD*, 2005, pp. 358-364.

[52] M. Nizam, F. Najm, and A. Devgan, "Power grid voltage integrity verification," in *Proc. ISLPED*, 2005, pp. 239-244.

[53] J. Stolfi and L.H. de Figueiredo, "An introduction to affine arithmetic," *TEMA Tend. Mat. Apl. Comput.*, vol .4, no. 3, pp. 297-312, 2003.

[54] J. D. Ma and R. A. Rutenbar, "Interval-valued reduced order statistical interconnect modeling," in *Proc. ICCAD*, 2004, pp. 460-467.

[55] H. Godwin, Inequalities on Distribution Functions, Hafner, 1964.

[56] S. Ferson, *RAMAS Risk Calc 4.0 Software: Risk Assessment with Uncertain Numbers*, CRC Press, 2002.

[57] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpre, and M. Aviles, "Computing variance for interval data is NP-hard," *ACM SIGACT News*, vol. 33, no. 2, pp. 108-118, Jun. 2002.

[58] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[59] G. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*, Springer-Verlag, 1995.

[60] M. Glesner, J. Schuck and R. B. Steck, "SCAT - a new statistical timing verifier in a silicon compiler system," in *Proc. DAC*, 1986, pp. 220-226.

[61] D. E. Wallace and C. H. Séquin, "Plug-in timing models for an abstract timing verifier," in *Proc. DAC*, 1986, pp. 683-689.

[62] J. Rice, *Mathematical Statistics and Data Analysis*, Wadsworth & Brooks, 1988.

[63] Y. Cao, T. Sato, M. Orshansky, D. Sylvester, and C. Hu, "New paradigm of predictive MOSFET and interconnect modeling for early circuit design," in *Proc. CICC*, 2000, pp. 201-204.

[64] S. Narendra and A. Chandrakasan, *Leakage in Nanometer CMOS Technologies*, Springer, 2006.

[65] D. Lee, D. Blaauw, and D. Sylvester, "Gate oxide leakage current analysis and reduction for VLSI circuits," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 2, pp. 155-166, Feb. 2004.

[66] H.-Y. Wong, L. Cheng, Y. Lin, and L. He, "FPGA device and architecture evaluation considering process variations", in *Proc. ICCAD*, 2005, pp. 19-24.

[67] A. Srivastava, S. Shah, K. Agarwal, D. Sylvester, D. Blaauw, and S. Director, "Accurate and efficient gate-level parametric yield estimation considering correlated variations in leakage power and performance," in *Proc. DAC*, 2005, pp. 535-540.

[68] S. Zhang, V. Wason, and K. Banerjee, "A probabilistic framework to estimate full-chip threshold leakage power distribution considering within-die and die-to-die P-T-V variations," in *Proc. ISLPED*, 2004, pp. 156-161.

[69] S. Mukhopadhyay and K. Roy., "Modeling and estimation of total leakage current in nano-scaled CMOS devices considering the effect of parameter variation," in *Proc. ISLPED*, 2003, pp. 172-175.

[70] G. McLachlan and D. Peel, *Finite Mixture Models*, Wiley, 2000.

[71] M. Gregoire, S. Kordic, M. Ignat, X. Federspiel, P. Vannier, and S. Courtas, "New stress voiding observations in Cu interconnects," in *Proc. International Interconnect Technology Conference*, 2005, pp. 36-38.

[72] D. Berleant and C. Goodman-Strauss, "Bounding the results of arithmetic operations on random variables of unknown dependency using intervals," *Reliable Computing*, vol. 4, no. 2, pp. 147-165, 1998.

[73] H. Reagan. S. Ferson, and D. Berleant., "Equivalence of methods for uncertainty propagation of real-valued random variables," *International J. Approximate Reasoning*, vol. 36, no. 1, pp. 1-30, 2004.

[74] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif., "Full-chip leakage estimation considering power supply and temperature variations," in *Proc. ISLPED*, 2003, pp. 78-83.

[75] B. D. Cory, R. Kupar, and B. Underwood, "Speed binning with path delay test in 150-nm technology," *IEEE Design and Test of Computers*, vol. 20, no. 5, pp. 41-45, Sep./Oct. 2003.

[76] M. J. Todd, "The many facets of linear programming," *Mathematical Programming*, vol. 91, no. 3, pp. 417-436, Feb. 2002.

[77] D. Lackey, P. Zuchowski, T. Bednar, D. Stout, S. Gould, and J. Cohn, "Managing power and performance for System-on-Chip designs using voltage islands," in *Proc. ICCAD*, 2002, pp. 195-202.

[78] A. Raychowdhury, S. Mukhopadhyay, and K. Roy, "Modeling and estimation of leakage in sub-90nm devices," in *Proc. Int. Conf. on VLSI Design*, 2004, pp. 65-70.

[79] Predictive technology model. Available: http://www.eas.asu.edu/~ptm.

[80] S. Sirichotiyakul, T. Edwards, C. Oh, J. Zuo, A. Dharchoudhury, R. V. Panda, and D. Blaauw, "Stand-by power minimization through simultaneous threshold voltage selection and circuit sizing," in *Proc. DAC*, 1999, pp. 436-441.

[81] M. Hirabayashi, K. Nose, T. Sakurai, "Design methodology and optimization strategy for dual-VTH scheme using commercially available tools," in *Proc. ISLPED*, 2001, pp. 283 – 286.

[82] J. Kao, S. Narendra, and A. Chandrakasan, "Subthreshold leakage modeling and reduction techniques," in *Proc. ICCAD*, 2002, pp. 141-148.

[83] S. Narendra, D. Blaauw, A. Devgan, and F. Najm, "Leakage issues in IC design: Trends, estimation, and avoidance," Tutorial, *ICCAD*, 2003.

[84] R. W. Brodersen, M. A. Horowitz, D. Markovic, B. Nikolic and V. Stojanovic, "Method for true power optimization," in *Proc. ICCAD*, 2002, pp. 35-42.

Vita

Wei-Shen Wang was born in Taipei, Taiwan on January 12th, 1976, the son of Chai-Fu Wang and Shu-Chu Chen. He received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taiwan, in 1998 and 2000, respectively. From 2000 to 2002, he was a telecommunication officer in the Republic of China (Taiwan) Army. In 2002, he entered the Ph.D. program in the Department of Electrical and Computer Engineering, the University of Texas, Austin. His research interests include statistical static timing analysis, leakage power minimization, and design for manufacturability of digital integrated circuits.

Permanent address: 8 F., No. 150, Songqin St., Taipei City, 110, Taiwan.

This dissertation was typed by the author.